City Research Online

City St George's, University of London

Citation: Blazakis, K. N., Madzvamuse, A., Reyes-Aldasoro, C. C., Styles, V. & Venkataraman, C. (2015). Whole cell tracking through the optimal control of geometric evolution laws. Journal of Computational Physics, 297, pp. 495-514. doi: 10.1016/j.jcp.2015.05.014

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/8689/

Link to published version: https://doi.org/10.1016/j.jcp.2015.05.014

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

arXiv:1504.05399v1 [math.NA] 21 Apr 2015

Whole cell tracking through the optimal control of geometric evolution laws

Konstantinos N. Blazakis[a], Anotida Madzvamuse[a], Constantino-Carlos Reyes-Aldasoro[b], Vanessa Styles[a] and Chandrasekhar Venkataraman*[a]

[a]*University of Sussex, Department of Mathematics, Brighton, UK, BN1 9QH*

[b]*City University London, School of Mathematical Sciences, Computer Science and Engineering, Northampton Square, London, UK, EC1V OHB*

Abstract

Cell tracking algorithms which automate and systematise the analysis of time lapse image data sets of cells are an indispensable tool in the modelling and understanding of cellular phenomena. In this study we present a theoretical framework and an algorithm for whole cell tracking. Within this work we consider that "tracking" is equivalent to a dynamic reconstruction of the whole cell data (morphologies) from static image datasets. The novelty of our work is that the tracking algorithm is driven by a model for the motion of the cell. This model may be regarded as a simplification of a recently developed physically meaningful model for cell motility. The resulting problem is the optimal control of a geometric evolution law and we discuss the formulation and numerical approximation of the optimal control problem. The overall goal of this work is to design a framework for cell tracking within which the recovered data reflects the physics of the forward model. A number of numerical simulations are presented that illustrate the applicability of our approach.

Keywords. Cell tracking, geometric evolution law, optimal control, phase field, finite elements.

1 Introduction

Cell migration is a fundamental process in cell biology and is tightly linked to many important physiological and pathological events such as the immune response, wound healing, tissue differentiation, metastasis, embryogenesis, inflammation and tumour invasion [1]. Experimental advances provide techniques to observe migrating cells both *in vivo* and *in vitro*. Inferring dynamic quantities from this static data is an important task that has many applications in biology and related fields. The field of cell tracking arose from this need and is concerned with the development of methods to track and analyse dynamic cell shape changes from a series of still images captured within a time frame (see for example [2, 3] for reviews).

On the other hand, a major focus of current research is the derivation of mathematical models for cell migration based on physical principles, e.g., [4]. Furthermore, such models appear to show good qualitative and quantitative agreement with experimental observations of migrating cells. Despite this, very little research has focused on incorporating these mathematical

*Corresponding author. Email : C.Venkataraman@sussex.ac.uk

modelling advances into appropriate cell tracking algorithms. In a related work, we investigated fitting parameters in models for cell motility to experimental image data sets of migrating cells where observations of both the position of the cells and the concentrations of cell-resident proteins related to motility were available [5].

In this study we present a first step towards the development of a framework for cell tracking based on novel models of cell motility. Specifically, we propose a cell tracking algorithm which can be thought of as fitting a simplified, yet physically meaningful, model for cell migration to experimental observations and data. We focus on the setting, prevalent in cell tracking problems, where only the position of the cell at a series of discrete times is available and no further biological information is given. We present a mathematical model based on physical principles for the cell movement that consists of a geometric evolution equation. We then formulate an inverse problem, which takes the form of a PDE constrained optimisation problem, for fitting the model to the static experimental observations. To solve the optimisation problem we propose an algorithm based on the optimal control of geometric evolution laws [6, 7].

The objective of this study is to serve as a useful first step in the development of cell tracking algorithms in which the underlying model for the evolution is based on physical principles, rather than purely geometric considerations. In this setting, one hopes to attain estimates of motility-related features such as trajectories, velocities, persistence lengths, circularity, etc., which reflect the physics underlying the model. We illustrate the fact that the tracking procedure we propose allows us to incorporate physically important aspects of cell migration by including volume conservation in the model for the evolution. This is motivated by the observation that, for many cells, while the surface area of the cell membrane may change significantly during migration the volume enclosed by the cell remains roughly constant [8]. Of course other physical aspects of the migration could be included in the model, such as a spontaneous curvature of the membrane which is relevant for more complex models of cell motility involving the Helfrich model [9].

The remainder of our discussion proceeds as follows. In §2 we briefly describe the problem of cell tracking and introduce our approach to cell tracking, which may be regarded as fitting a mathematical model to experimental image data sets. We present the geometric evolution law model we seek to fit, which is a simplification of recently developed models in the literature that show good agreement with experiments [4,8–13]. We finish §2 by reformulating our model into the phase field framework, which appears more suitable for the problem in hand, and we formulate the cell tracking problem as a PDE constrained optimisation problem. In §3 we propose an algorithm for the resolution of the PDE constrained optimisation problem and we discuss some practical aspects related to the implementation. In particular we note that the theoretical and computational framework may be applied directly to multi-cell image data sets and raw image data sets (of sufficient quality) without segmentation. In §4 we present some numerical examples for the case of $2d$ single and multi-cell image data sets. Finally in §5 we present some conclusions of our study and discuss future extensions and applications of the work.

2 Problem Formulation

2.1 Approaches to cell tracking

Our focus is on developing whole cell tracking algorithms which track the morphology of the cell rather than particle tracking in which particles such as the cell centroid or cell resident proteins or (macro-)molecules are tracked [14]. A number of approaches have proved successful in cell tracking with level-set [15] or electrostatic based methods among the most widely used [16]. One feature of such methods is that the trajectories they generate are not physical in nature rather they are designed with the goal of achieving nice geometric properties, e.g., equidistribution of vertices, smoothness of the trajectories and so on. Our approach differs to

these purely geometric approaches in that we start with a model derived from physical principles and it is this model for the evolution that drives the tracking algorithm. In this sense our approach is similar in spirit to the parameter identification procedure described in [5] as in both studies the goal may be regarded as fitting a mathematical model to experimental image data sets.

We now summarise the main problems that must be addressed by a cell tracking algorithm. In general cell tracking consists of three main steps

1. Segmentation: In this step the raw image data set is processed and the cells are separated from the background in each frame.
2. Matching: The cells segmented in the first step must then be associated from frame to frame (note this is only relevant in the case of multiple cell image data sets) such that where possible (in practice cells may disappear or spontaneously appear in images) there is a one-to-one map that uniquely associates individual cells from one frame to the next.
3. Linking: Finally the linking step consists of estimating dynamic data from the associated segmented static cells.

In this work we will largely neglect the segmentation step. We assume either that we have segmented image data to work with or that the image data is of sufficient quality that the contrast between the cell and the background is clear and a simple thresholding step is sufficient to label the different cells. In the case of segmented image data, we assume this data consists of closed surfaces (or curves in $2d$) that describe the boundaries of each individual cell. We briefly comment on the role of segmentation in our approach and the fact that it may be unnecessary to extract the sharp contours of the cell boundaries in Remark 3.2. For ease of presentation we will also describe the algorithm in the case of single cell image data and thus the matching step is redundant. However the theoretical aspects of our approach apply equally to multiple cell data and one potentially attractive feature of our cell tracking procedure is that the matching problem may be resolved implicitly without the need to associate cells form one frame to the next, c.f., Remark 3.4. Our main focus is on the tracking step and in the remainder of this study we outline a theoretical framework, a practical implementation and numerical examples, for linking data at a series of discrete times which allows the recovery of the whole cell morphologies in time.

2.2 Model

As mentioned above in contrast to many of the existing approaches for cell tracking, the framework we propose in this study is based on fitting a model, derived from physical principles, for the motion of the cell to experimental image data. The general class of models to which our approach is applicable are PDE based models for the motion, where the movement of the cell membrane is described by a geometric evolution law. We introduce some notation for the formulation of the model.

We denote by Γ the cell membrane, which is assumed to be a closed smooth oriented $d-1$ dimensional hypersurface in $\mathbb{R}^d$, $d = 2, 3$, with outward pointing unit normal $\boldsymbol{\nu}$. Given a function η defined in a neighbourhood of Γ, the tangential or surface gradient of η denoted by ∇_Γ is defined as

$$\nabla_\Gamma \eta := \nabla \eta - \nabla \eta \cdot \boldsymbol{\nu} \boldsymbol{\nu}, \tag{2.1}$$

where ∇ denotes the Cartesian gradient in $\mathbb{R}^d$. The Laplace-Beltrami operator Δ_Γ is defined as the tangential divergence of the tangential gradient, i.e.,

$$\Delta_\Gamma \eta := \nabla_\Gamma \cdot (\nabla_\Gamma \eta)\,. \tag{2.2}$$

The mean curvature H of Γ with respect to the normal $\boldsymbol{\nu}$ is defined as

$$H := \nabla_\Gamma \cdot \boldsymbol{\nu}. \tag{2.3}$$

In this study we model the evolution of the cell membrane as being governed by volume conserved mean curvature flow with forcing, given by

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x},t) = (-\sigma H(\boldsymbol{x},t) + \eta(\boldsymbol{x},t) + \lambda_V(t))\,\boldsymbol{\nu}(\boldsymbol{x},t) \quad \text{on } \Gamma(t), t \in (0,T], \\ \Gamma(0) = \Gamma^0, \end{cases} \tag{2.4}$$

where Γ is the closed surface that represents the cell membrane, $\boldsymbol{V}$ is the material velocity of Γ, σ is the surface tension and $\lambda_V(t)$ is a spatially uniform force accounting for volume conservation, physically this may be thought of as an interior pressure. The forcing function η is the main driver of the directed migration. The model we present is phenomenological and hence it is difficult to directly relate η to biophysical processes. However, as positive values of η correspond to protrusive forces and negative values of η correspond to contractile forces one interpretation of the forcing function η is that it accounts for both protrusive forces generated by polymerisation of actin at the leading edge of the cell and contractile forces generated by the action of myosin motors at the rear of the cell.

The evolution law (2.4) is a simplification of a large class of models that arise in the modelling of cell motility which take the following form

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x},t) = \Big(g_1(H(\boldsymbol{x},t)) + g_2(\boldsymbol{a}(\boldsymbol{x},t)) + \lambda_V(t) \Big) \boldsymbol{\nu}(\boldsymbol{x},t) \quad \text{on } \Gamma(t), t \in (0,T], \\ \Gamma(0) = \Gamma^0, \end{cases} \tag{2.5}$$

where g_1 models the dependence of the evolution on geometric quantities, such as resistance of the membrane to stretching which could be modelled by mean curvature terms as in (2.4) or bending which can be modelled through the inclusion of Willmore or Helfrich flow type terms. The function g_2 appearing in (2.5) captures the dependence of the evolution on a vector of bulk and/or surface resident species $\boldsymbol{a}$. The surface resident species $\boldsymbol{a}$ could, for example, satisfy another PDE such as a surface reaction-diffusion system

$$\begin{cases} \partial^\bullet_{\boldsymbol{V}} \boldsymbol{a} + \boldsymbol{a} \nabla_{\Gamma(t)} \cdot \boldsymbol{V} - \boldsymbol{D} \Delta_{\Gamma(t)} \boldsymbol{a} = \boldsymbol{f}(\boldsymbol{a}) \quad \text{on } \Gamma(t), t \in (0,T], \\ \boldsymbol{a}(\cdot,0) = \boldsymbol{a}^0(\cdot) \quad \text{on } \Gamma(0), \end{cases} \tag{2.6}$$

where $\boldsymbol{a} = (a_1, \ldots, a_{n_a})^T$, n_a is the number of chemical species involved, a_i denotes the density of the ith chemical species, $\boldsymbol{V}$ is the material velocity of the surface,

$$\partial^\bullet_{\boldsymbol{V}} \boldsymbol{a} := \partial_t \boldsymbol{a} + \boldsymbol{V} \cdot \nabla \boldsymbol{a}, \tag{2.7}$$

is the material derivative with respect to the velocity $\boldsymbol{V}$, $\boldsymbol{D}$ is a diagonal matrix of positive diffusion coefficients and $\boldsymbol{f}(\boldsymbol{a})$ is the nonlinear reaction. Models of the form (2.5)-(2.6) have been used successfully to model cell motility in [4, 11, 12, 17] while models coupling evolution laws of the form (2.5) to bulk PDEs (i.e., equations posed in the cell interior) have been considered in [9, 10, 13].

Despite its simplicity the evolution law (2.4) may be regarded as a prototype of the more complex models for cell motility of the form (2.5)-(2.6). The geometric evolution component (2.5) is often the most challenging component of the model to solve numerically and developing an understanding of how to construct cell tracking algorithms assuming a geometric evolution law based model for the motion is an important first step towards developing tracking algorithms based on more realistic physical models. In many applications it is also the case that the only information available from the data is the position of the cell membrane and no adequate model for the biochemistry of the motility related species involved is available. Without any knowledge of the relevant biochemistry it is difficult to identify which motility related species should influence the evolution let alone propose how the evolution depends on their distribution (i.e., a g_2 in (2.5)) or a model for the species dynamics (i.e., an equation such as (2.6)). Nevertheless one may still wish to extract dynamic quantities from static image

data sets, in this setting it may be reasonable to consider the evolution law (2.4) as a stand alone model for the motion as at least the mechanical aspects of the membrane evolution are accounted for through a physical model derived from basic physical principles.

2.3 An optimal control approach to cell tracking

The cell tracking approach we consider in this study corresponds to the following problem.

2.4 Problem (Cell tracking)**.** Given an initial cell membrane position Γ^0 and an observation of the position Γ_{obs}, find a space-time distributed forcing η such that the evolution of the cell membrane, $\Gamma(t), t \in [0,T]$ satisfies (2.4) with $\Gamma(0) = \Gamma^0$ and $\Gamma(T)$ the position of the cell membrane at time $t = T$, is close to Γ_{obs}.

As the volume enclosed by the cell may vary over the images it is inappropriate to enforce conservation of a constant volume. Instead we enforce, with the help of a Lagrange multiplier $\lambda_V(t)$, that the volume enclosed by the cell is given by $\widetilde{V}(t) = V^0 + \frac{t}{T}(V_{obs} - V^0)$, i.e. that the volume of the cell is a time-dependent linear interpolant of the volumes of the data.

Problem 2.4 is an optimal control of a free boundary problem, where the free moving boundary problem is that of forced mean curvature flow and the control variable is the space time distributed forcing. The theory of optimal control of geometric evolution laws is in its infancy, in fact only recently has progress been made on the optimal control of parabolic equations on evolving surfaces even in the case of prescribed evolution [18]. On the other hand the theory for the optimal control of semilinear parabolic equations is more mature (see, for example, [19]). We wish to exploit this fact and to this end we consider the phase field approximation of (2.4) given by the Allen-Cahn equation;

$$\begin{cases} \partial_t \varphi(\boldsymbol{x},t) & = \Delta\varphi(\boldsymbol{x},t) - \frac{1}{\varepsilon^2} G'(\varphi(\boldsymbol{x},t)) - \frac{1}{\varepsilon}\left(c_G \eta(\boldsymbol{x},t) - \lambda(t)\right) \text{ in } \Omega \times (0,T], \\ \nabla\varphi \cdot \boldsymbol{\nu}_\Omega & = 0 \text{ on } \partial\Omega \times (0,T], \\ \varphi(\cdot,0) & = \varphi^0(\cdot) \text{ in } \Omega, \end{cases} \tag{2.8}$$

where $\Omega \subset \mathbb{R}^d$ is a bulk domain, with normal $\boldsymbol{\nu}_\Omega$, that contains Γ, φ^0 is a diffuse interface representation of Γ^0 and $\varepsilon > 0$ is a small parameter which governs the width of the diffuse interface. For details on the asymptotic analysis of (2.8) and the convergence (as $\varepsilon \to 0$) to a solution of (2.4) we refer the reader, for example, to [20–23] and references therein. The function G appearing in (2.8) is a double well potential, for example the quartic potential

$$G(\varphi) = \frac{1}{4}\left(\varphi^2 - 1\right)^2 \tag{2.9}$$

which has minima at ± 1. The constant $c_G = \frac{1}{\sqrt{2}} \int_{-1}^{1} G(r)^{1/2} \mathrm{d}r$ appearing in (2.8) is a scaling constant that depends on the double well potential. We enforce the time-dependent volume constraint following the approach of [21]. Specifically our diffuse interface formulation of the constraint on the enclosed volume is given by a constraint on $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$, where $[a]_+ = max(a,0)$. We define M_φ, the linear interpolant of $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$ of the initial and target diffuse interface data by

$$M_\varphi(t) := \int_\Omega [\varphi^0]_+ + \frac{t}{T}\left([\varphi_{obs}]_+ - [\varphi^0]_+\right) \mathrm{d}\boldsymbol{x},$$

and determine $\lambda(t)$ in (2.8) such that $M_\varphi(t) = \int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$. We have used λ (rather than λ_V) for the Lagrange multiplier in (2.8) to reflect the fact that our constraint is on $\int_\Omega [\varphi(\boldsymbol{x},t)]_+ \mathrm{d}\boldsymbol{x}$. However we shall refer to this constraint as a volume constraint in order to highlight the physical feature the constraint is intended to model. We also investigated an alternative approach to enforcing the volume constraint via penalising deviations from a target volume following [24]

(see also [9]), in our numerical tests this strategy proved less robust than the volume constraint proposed above.

To formulate the cell tracking problem as a PDE constrained optimal control problem we define the objective functional we shall seek to minimise as follows

$$J(\varphi,\eta)=\frac{1}{2}\int_\Omega\left(\varphi(\boldsymbol{x},T)-\varphi_{obs}(\boldsymbol{x})\right)^2\mathrm{d}\boldsymbol{x}+\frac{\theta}{2}\int_0^T\int_\Omega\eta(\boldsymbol{x},t)^2\mathrm{d}\boldsymbol{x}\mathrm{d}t, \tag{2.10}$$

where φ_{obs} is a diffuse interface representation of the observation Γ_{obs} and $\theta>0$ is a regularisation parameter. The first term on the right of (2.10) is the so called fidelity term that measures the distance between the solution to the model and the target data and the second term is the regularisation which is necessary to ensure a well-posed problem (for example see [19]).

Our optimal control approach to the cell tracking problem may now be stated as the following minimisation problem.

2.5 Problem (Optimal control problem)**.** Given an initial diffuse interface representation of the cell membrane position φ^0 and an observation of the position φ_{obs}, find a space-time distributed forcing $\eta^*:\Omega\times[0,T]\to\mathbb{R}$ such that with φ a solution of (2.8) with initial condition $\varphi(\cdot,0)=\varphi^0$, the forcing η^* solves the minimisation problem

$$\min_\eta J(\varphi,\eta),\quad\text{with } J \text{ given by (2.10).} \tag{2.11}$$

2.6 Optimality conditions

To apply the theory of optimal control of semilinear PDEs for the solution of the tracking problem, we briefly outline the derivation of the optimality conditions, for further details see for example [19, 25]. Introducing the Lagrange multiplier (adjoint state) p, we define the Lagrangian functional

$$\begin{aligned}\mathcal{L}(\varphi,\eta,p)=J(\varphi,\eta)-\int_0^T\int_\Omega\Big(&\partial_t\varphi(\boldsymbol{x},t)-\Delta\varphi(\boldsymbol{x},t)\\&+\frac{1}{\varepsilon^2}G'(\varphi(\boldsymbol{x},t))+\frac{1}{\varepsilon}\big(c_G\eta(\boldsymbol{x},t)-\lambda(t)\big)\Big)p(\boldsymbol{x},t)\mathrm{d}\boldsymbol{x}\mathrm{d}t.\end{aligned} \tag{2.12}$$

Requiring stationarity of the Lagrangian with respect to the adjoint state yields the state equation (2.8) and requiring stationarity of the Lagrangian, at the optimal control η^* and associated optimal state φ^*, with respect to the state and the control, yields the (formal) first order optimality conditions

$$\delta_\varphi\mathcal{L}(\varphi^*,\eta^*,p)\varphi=0,\quad\forall\varphi:\varphi(\boldsymbol{x},0)=0, \tag{2.13}$$

$$\delta_\eta\mathcal{L}(\varphi^*,\eta^*,p)\eta=0,\quad\forall\eta. \tag{2.14}$$

Condition (2.13) yields the adjoint equation, which is the following linear parabolic PDE for the adjoint state p,

$$\begin{cases}\partial_t p(\boldsymbol{x},t) &= -\Delta p(\boldsymbol{x},t)+\frac{1}{\varepsilon^2}G''(\varphi(\boldsymbol{x},t))p(\boldsymbol{x},t)\quad\text{in }\Omega\times(0,T],\\ \nabla p\cdot\boldsymbol{\nu}_\Omega &= 0\quad\text{on }\partial\Omega\times(0,T],\\ p(\boldsymbol{x},T) &= \varphi(\boldsymbol{x},T)-\varphi_{obs}(\boldsymbol{x})\quad\text{in }\Omega.\end{cases} \tag{2.15}$$

Note that equation (2.15) is posed backwards in time and hence is equipped with terminal conditions. Condition (2.14) together with the Riesz representation theorem yields the optimality condition (c.f., [19])

$$\delta_\eta\mathcal{L}(\varphi^*,\eta^*,p)=\theta\eta^*+\frac{c_G}{\varepsilon}p=0. \tag{2.16}$$

2.7 Remark (Choice of the potential)**.** We note that our approach to the optimal control problem involving the formulation of the adjoint problem appears to require a smooth potential G (c.f., (2.9)). The formulation of the adjoint problem is to our best knowledge an open problem for other widely used, but non smooth or unbounded, potentials such as the obstacle or logarithmic potential.

3 Practical considerations, implementation and algorithm

As is standard, we use the optimality conditions to construct an iterative optimisation loop to solve the optimal control problem, Problem 2.5. The basic idea is that in each step of the loop we first solve the state equation (2.8) with a given control, then solve the adjoint equation (2.15) with the computed states and then update the control using the optimality condition (2.16). For this initial study to ensure robustness of the algorithm and to aid in the clarity of the exposition we employ a simple gradient based update of the control [26]. Given η^k and p we compute the updated control η^{k+1} via steepest descent. That is we choose as an update direction the negative gradient, the formula for the update of the control is

$$\eta^{k+1}(\boldsymbol{x},t) = \eta^k(\boldsymbol{x},t) - \alpha\Big(\theta\eta^k(\boldsymbol{x},t) + \frac{c_G}{\varepsilon}p(\boldsymbol{x},t)\Big), \quad (\boldsymbol{x},t) \in \Omega \times [0,T), \tag{3.1}$$

where α is a step size. For simplicity in this study we take a constant step size of $\alpha = 0.01$.

For the termination criteria for the algorithm we stop if the objective functional J is less than a given tolerance tol_J, the update in the control is less than a given tolerance, i.e., if $\left\|\alpha(\theta\eta^k + \frac{c_G}{\varepsilon}p)\right\|_{\mathrm{L}_2(\Omega\times[0,T))} < tol_\eta$ or if a maximum number of iterations K_{max} is reached. In practice the forward (2.8) and adjoint equations (2.15) must be approximated and to do this we utilise a finite element method, using continuous piecewise linear elements. The details of the numerical method for the approximation of the forward and adjoint equations are given in A.

The cell tracking algorithm we propose may now be stated in pseudocode as follows (the notation employed is as in A);

3.1 Optimal control based cell tracking algorithm

Require: Data: $\varphi_h^0, (\varphi_{obs})_h$ the initial and target (discrete) diffuse interface data.
Numerical Parameters: $T > 0$ end-time and $M > 0$ number of timesteps.
Optimisation Parameters: tolerances tol_J, tol_η, and K_{max}.
Initial guess for the control: $(\eta_h)^0 := (\eta_h^i)^0 \in \mathbb{V}, i = 0, \ldots, M$.

$k := 0$
while $\Big(\left\|\alpha(\theta(\eta_h)^k + \frac{c_G}{\varepsilon}p_h^{k+1})\right\|_{\mathrm{L}_2(\Omega\times[0,T))} > tol_\eta$, $J > tol_J$ and $k < K_{max}\Big)$ **do**

Solve state equation for $\{(\varphi_h^i)^{k+1}, (\lambda^i)^{k+1}\}, i = 1, \ldots, M$ with $(\eta_h^i)^k$ and initial data $(\varphi_h^0)^{k+1} = \varphi_h^0$. ▷ c.f., §A

Solve adjoint equation for $(p_h^i)^{k+1}, i = M-1, \ldots, 0$ with computed $(\varphi_h^i)^{k+1}$ and with terminal data $(p_h^M)^{k+1} = (\varphi_h^M)^{k+1} - (\varphi_{obs})_h$.

Update control $(\eta_h^i)^{k+1} = (\eta_h^i)^k - \alpha(\theta(\eta_h^i)^k + \frac{c_G}{\varepsilon}(p_h^i)^{k+1}) \quad i = 0, \ldots, M$.

Compute J according to (2.10).

$k := k + 1$.

end while

3.2 Remark (Segmentation and image data)**.** An important aspect of any cell tracking algorithm is its ability to extract data suitable for the tracking algorithm from the experimental image data set. In many cases the experimental image data set is grayscale data with the intensity (brightness) indicating whether a point is in the interior of a cell, i.e., points inside the cell appear bright for example and points outside appear dark. For many tracking algorithms this intensity data is then post processed via a segmentation algorithm (e.g., active contour methods [27, 28]) to yield sharp interface representations of the cell membrane. Assuming a sharp interface representation of the cell membrane is available, diffuse interface representations may be easily initialised (see for example [5]). We note however that the raw intensity data produced by many imaging procedures may already be close to a diffuse interface representation of the cell, this is typically the case when the data is relatively free of noise and the contrast between the cell and the background is high. In this case one may wish to exploit this fact in the algorithm and work with the raw image data set itself (or a post processed e.g., thresholded version), thus circumventing the extra error induced by segmentation.

3.3 Remark (Observations at multiple points in time)**.** For clarity of exposition we focus on the case of fitting to a single observation. The approach generalises straightforwardly to multiple observations with the first term in (2.10) simply replaced by a sum over the distinct times at which the observations are taken of the difference between the solution (at the appropriate time) and the target data.

3.4 Remark (Multiple cells and matching problems)**.** As mentioned above a major focus of many cell tracking algorithms is to track multiple cells in the same image and the resolution of the so called matching problem. Our approach can be applied to multi-cell image data. Here φ^0 and φ_{obs} would be diffuse interface representations of the multi-cell image data set and the diffuse interfaces would consist of multiple disjoint phases. The remaining aspects of the approach remain unchanged and the matching problem is solved implicitly in the computation of the optimal control.

There are however multiple practical issues which arise in this setting related to the separation between distinct cells, which affects the choice of ε, and the fact that the evolution law (2.8) allows changes in the topology of the phases which may lead to cell splitting, the annihilation of a phase (which would correspond to the disappearance of a cell) or the nucleation of a phase (i.e., the spontaneous appearance of a cell). We intend to comment on practical approaches to multi-cell tracking elsewhere.

4 Numerical examples

We now present some benchmark numerical examples illustrating the application of the algorithm to artificial image data sets. For all the simulations we report on in this section, in the state equation (2.8) we set $\varepsilon = 0.1$, and we took the end-time $T = 0.4$. As mentioned previously the parameter ε governs the width of the diffuse interface and should in general be taken as small as is computationally feasible, smaller values of ε necessitate a finer grid, in this initial study with uniform grids we set $\varepsilon = 0.1$ as the CPU times become prohibitive for smaller values of ε. The end time T corresponds to the nondimensionalised time between snapshots and could in principle be related to an acquisition time between images given real biological data. For each of the experiments, apart from those of §4.4, we set the initial guess for the control to be constant in space and time (zero in the single cell case and one for the multi-cell examples). For the approximation of the forward and adjoint equations we used triangulations with 8321 DOFs in all the simulations, apart from those of §4.3, and selected a uniform timestep $\tau = 1 \times 10^{-3}$. The same numerical parameters for the optimisation algorithm were used for all the experiments and are given in Table 1. In every example we report on the algorithm terminated due to the update of the control being less than the prescribed tolerance.

α	θ	tol_J	tol_η	K_{max}
0.01	0.01	1×10^{-4}	1×10^{-4}	3500

Table 1: Parameter values used for the numerical simulations.

The technical details of the hardware used to carry out the simulations is given in Remark 4.1.

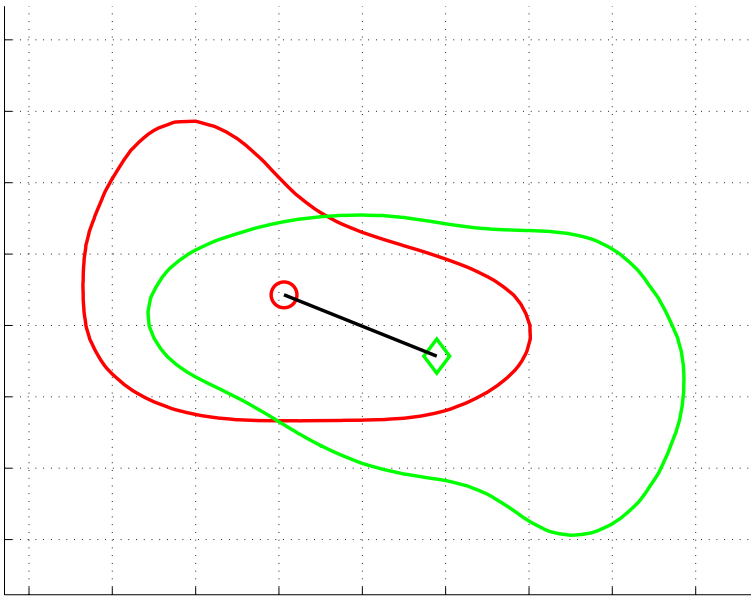
4.1 Remark (Hardware details)**.** All the numerical experiments have been performed on the high performance cluster (HPC) at the University of Sussex. Each of the simulations was carried out in serial using a single core of the cluster. The HPC cluster currently consists of 3140 cores with an even mixture of Intel and AMD CPUs. The majority of the cluster are 64 core AMD nodes with 256GB RAM per node, and a smaller number of 512GB RAM nodes. The cluster uses the high-performance Lustre clustered-filesystem for I/O, and currently stands at 298TB of storage for research use.

4.2 Application to synthetic data

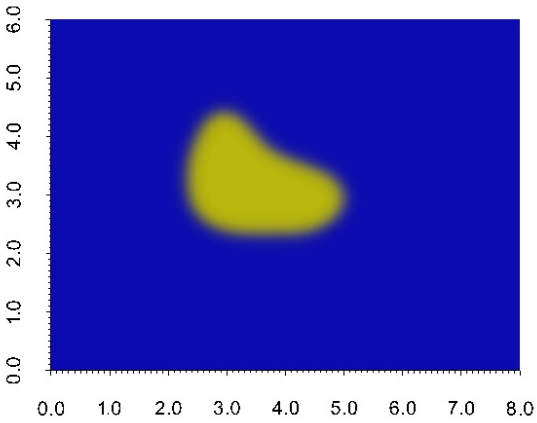
Here we apply the algorithm to a single synthetic cell data set taken from the *PhagoSight* website `http://www.phagosight.org/synData.php`. The synthetic cell was generated as a mixture of Gaussians with Poisson noise that varied over time to simulate the displacement and change of shape of a neutrophil as observed in a Zebrafish embryo. The data for analysis consisted of points on the synthetic cell membrane at a series of times (for simplicity we used $2d$ data, i.e., the cell membrane was a $1d$ curve embedded in $\mathbb{R}^2$). The initial and target curves we took as test data for the algorithm are shown in Figure 1(a). To apply our algorithm, based on diffuse interface representations, we define the domain Ω:=$[0, 8] \times [0, 6]$ which was such that both the initial and target curves were contained in the domain. We then constructed diffuse interface representations of the target data following the procedure described in [5]. Figures 1(b) and 1(c) show the diffuse interface representations of the initial and target data respectively. In order to investigate the influence of the volume constraint on the computed cell morphologies we performed two experiments, in the first we simply considered the forced Allen-Cahn model for the evolution with no volume constraint, i.e., (2.8) with $\lambda = 0$, and in the second we included the volume constraint as described in §2. The algorithm took 1996 iterations to meet the stopping criteria with no volume constraint and 2479 iterations with the volume constraint, corresponding to CPU times of 25238 and 119216 seconds respectively.

Figure 2 shows the value of the objective functional against the number of iterations of the optimisation algorithm with and without the volume constraint. We observe similar behaviour in both cases with an initial rapid decrease in the objective functional followed by a more gradual reduction with each iteration as we approach the minimum. Figure 3 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint. The curve corresponding to the zero level set is shaded by the value of the computed optimal control. The background shading corresponds to the target data. In both cases the position of the zero level-set of the computed solution shows good agreement with the target data. Qualitatively we observe cells with a clearly defined "front" and "rear", with the computed control corresponding to protrusive forces at the front and contractive forces at the rear.

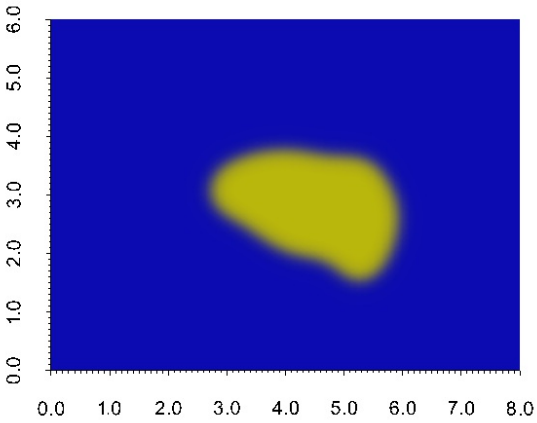
Figure 4 shows the area enclosed by the zero level-set of the solution with the optimal control with and without the volume constraint together with the linear interpolant of the areas of the data. We see that without the volume constraint the area initially decreases then rapidly increases as we approach the final time whilst with the volume constraint (note the constraint is actually on the mass rather than the volume) the area is close to the linear interpolant of the areas of the data. In terms of the computed cell morphologies, Figure 5 shows snapshots of the computed cell membranes (zero level-sets) for the two different cases. We clearly observe that



(a) Initial (red curve) and target (green curve) synthetic data. The cell centroids are shown together with the trajectory of the linear interpolant of the cell centroids (black line).



(b) Initial data (φ^0).

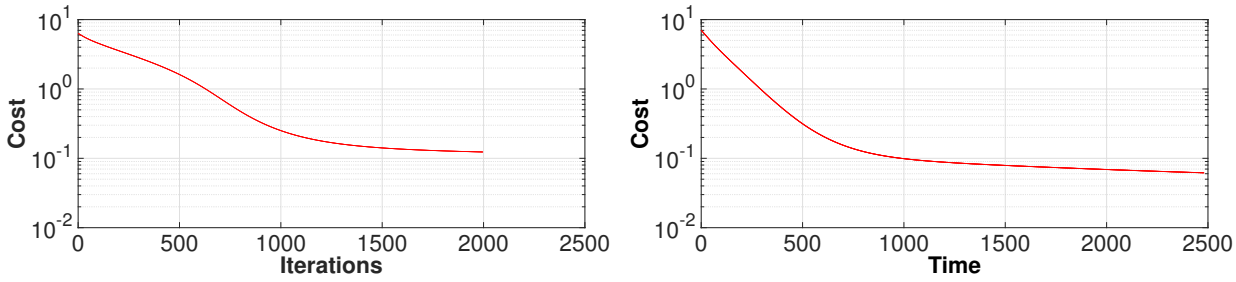


(c) Target data (φ_{obs}).

Figure 1: Initial and target data for the example of §4.2.

the intermediate snapshot (blue curve) encloses a much smaller area if the volume constraint is not included in the algorithm. In Figure 6 we report on the trajectory and speed (magnitude of the velocity) of the centroid (center of mass) of the zero level-set of the computed solution with the optimal control, with and without the volume constraint. We observe similar trajectories with and without the volume constraint and in both cases we observe an increase in the speed as we approach the end-time. However in the case of no volume constraint this increase is more marked with a sharp spike in the centroid velocity observed close to the final time. The increasing centroid speed we observe may be unphysical and if a (roughly) constant centroid velocity is desired one strategy may be to impose pointwise constraints on the control, this would prevent the large increase in the maximum and minimum values of the control observed during the simulations as we approach the final time as shown in Figure 7. Another possible strategy would be to modify the regularisation in (2.10).

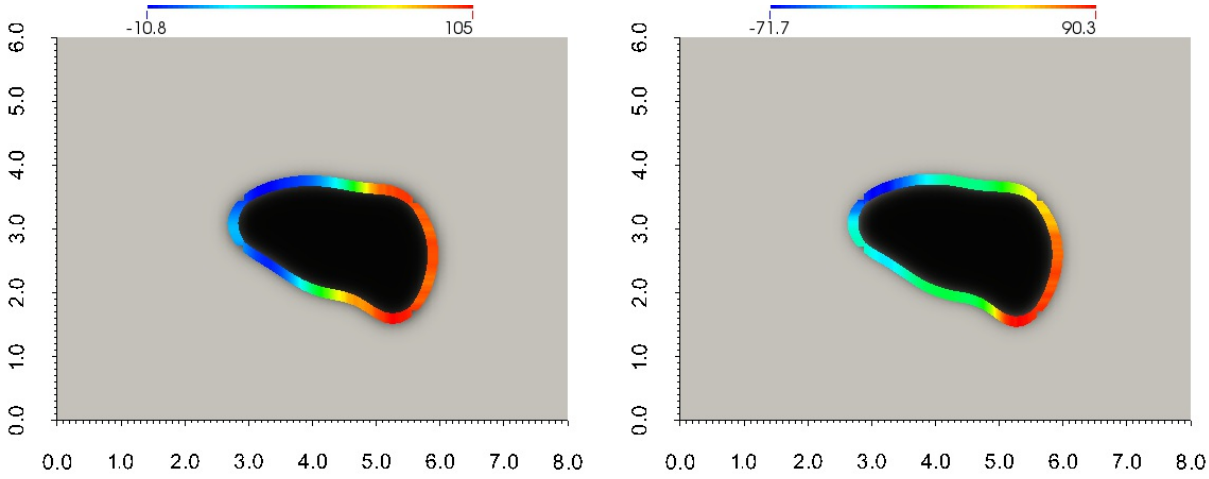
Figure 8 shows the fidelity term $\left\|\varphi_{obs}(\boldsymbol{x}) - \varphi(\boldsymbol{x}, T)^k\right\|_{\mathrm{L}_2(\Omega)}$ with and without the volume constraint versus the number of iterations (where k corresponds to the optimisation iteration number). The fidelity term may be considered as a quantitative measure for the "goodness of



(a) Without the volume constraint

(b) With the volume constraint

Figure 2: The value of the cost functional versus the number of iterations for the experiments of §4.2 with and without the volume constraint. We observe a rapid decrease in the cost initially followed by a much more gradual decrease as we approach the minimum, this is as expected since the steepest descent algorithm is used for the update of the control.



(a) Without the volume constraint

(b) With the volume constraint

Figure 3: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control (η^*) with and without the volume constraint for the experiments of §4.2. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

fit" of the computed data to the observations. We observe a steady decay in the fidelity term as we approach the optimal control in both cases.

4.3 The effect of mesh refinement

In this section we investigate the effect of the mesh-size on the results by refining the mesh whilst keeping the time step τ constant. We report on the value of the fidelity term computed using ϕ^*, i.e., the forward state computed with the optimal control. The initial and target data are taken to be the same as in §4.2 and we employ the algorithm with the volume constraint. The initial guess for the control is taken to be zero in each case. The results from the mesh refinements are presented in Table 2. We observe a reduction in the fidelity term as we refine the mesh which implies an improved fit to the observed data. Although in principle it would

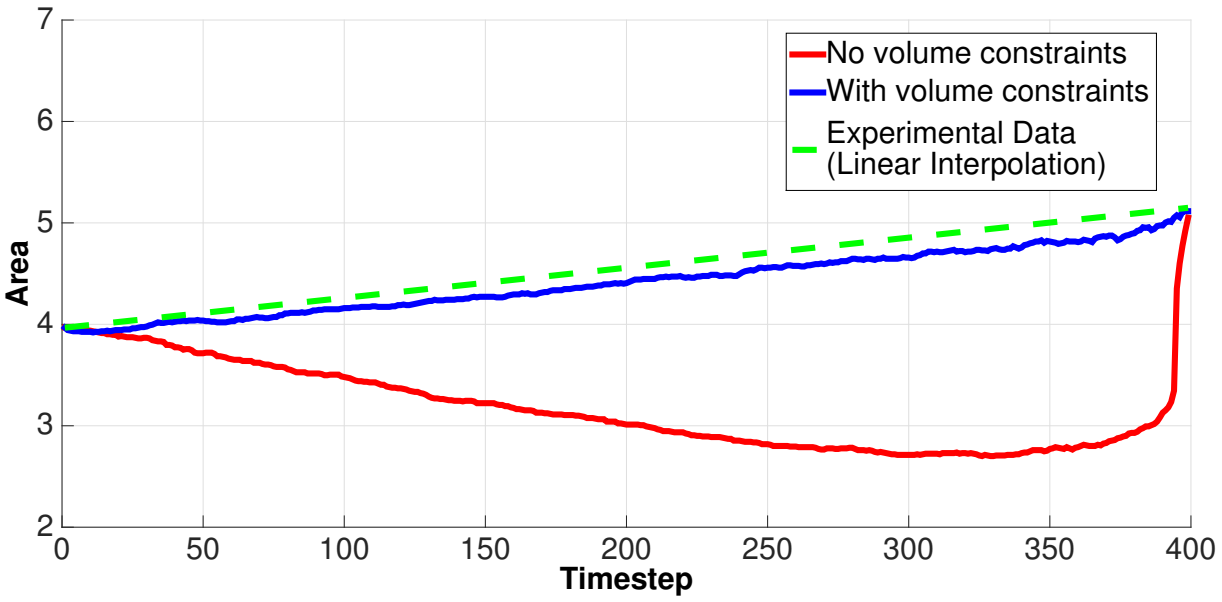
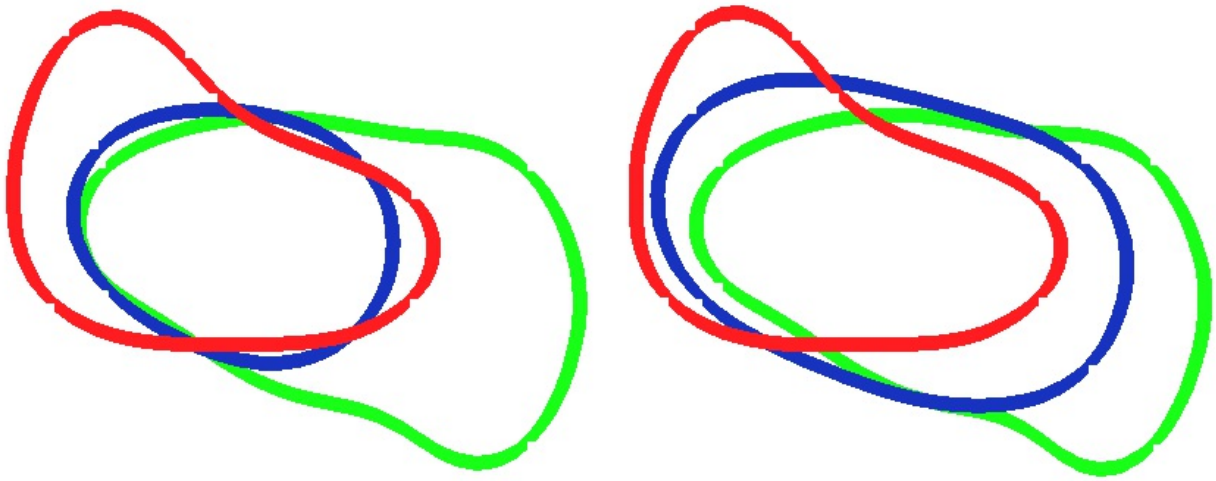


Figure 4: Area enclosed by the cell for the experiments of §4.2 with and without the volume constraint. The cell shrinks considerably during the evolution without the volume constraint whilst a good fit to the linear interpolant of the area enclosed by the data is observed with the volume constraint.

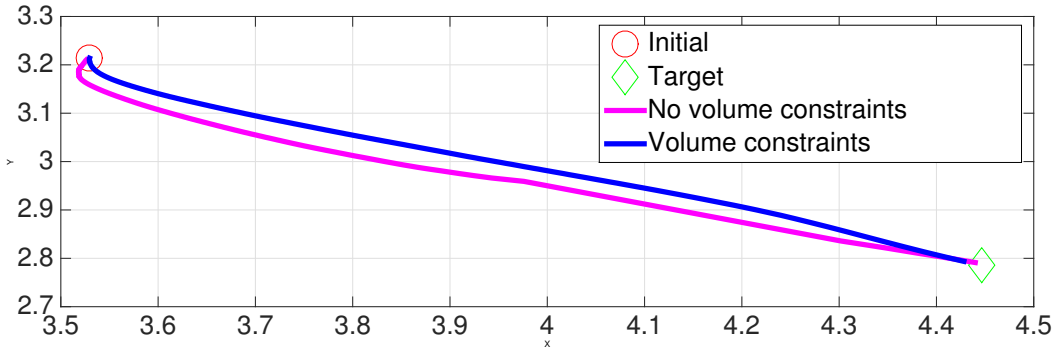


(a) Without the volume constraint

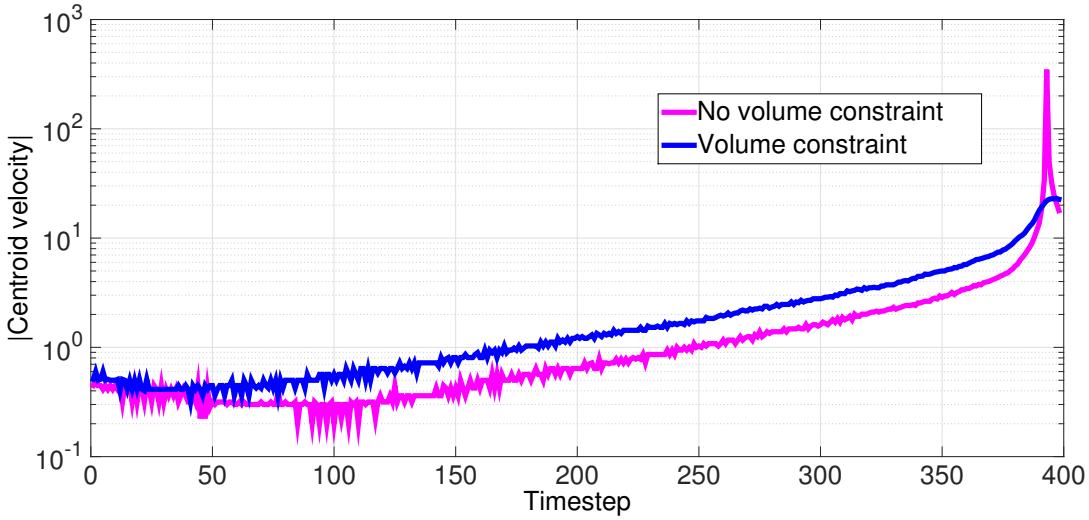
(b) With the volume constraint

Figure 5: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments of §4.2 with and without the volume constraint at $t = 0$ (red), $t = 0.35$ (blue) and $t = 0.4$ (green). We observe that the volume enclosed by the blue curve is significantly smaller than the volumes enclosed by the red and green curves without the volume constraint whilst this is not observed if the volume constraint is included.

be interesting to investigate the influence of refining both the timestep and mesh-size on the computed results, our tests indicate that the algorithm breaks down for time steps significantly larger than 0.01 and hence, refinement of the timestep and mesh-size together becomes computationally prohibitive.



(a) Centroid trajectories together with centroids of the data.



(b) Centroid speed (log scaling on the $y-$axis).

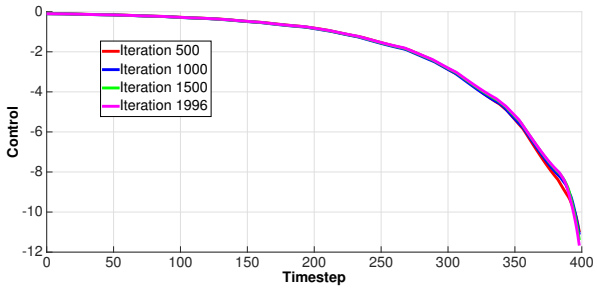
Figure 6: Trajectories and speeds of the centroid of the zero level-sets of the solution with the optimal control with and without the volume constraint for the example of §4.2.

$DOFs$	$\|\|\varphi(\boldsymbol{x},t)-\varphi_{obs}\|\|_{L_2(\Omega)}$	CPU time (secs)
545	0.348165	3142
2113	0.176357	26050
8321	0.154305	119216
33025	0.135178	291597

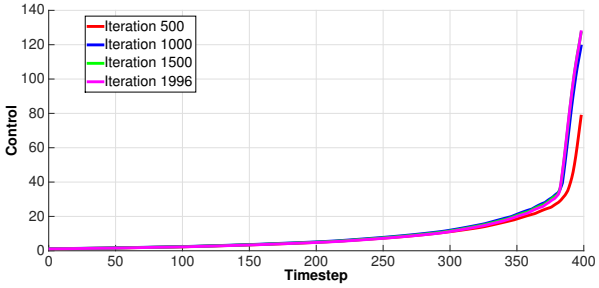
Table 2: Mesh refinement.

4.4 The influence of the initial guess for the control

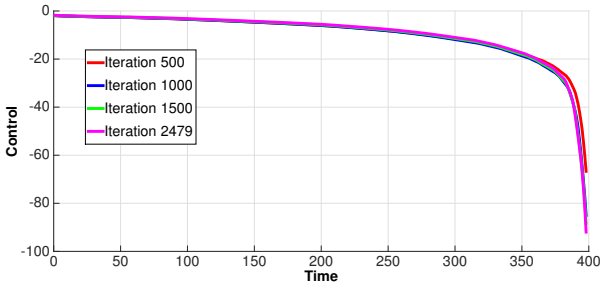
Here we apply the algorithm with the volume constraint on the simple example of a translated circle to illustrate the effect that the choice of the initial guess for the control η has on the solution of the problem. To apply our algorithm we define the domain Ω to be $[-3,6]\times[-3,3]$ with a triangulation of 8321 grid points. We selected a uniform timestep $\tau = 1\times10^{-3}$ and set the interfacial thickness $\varepsilon = 0.1$. We took the end-time $T = 0.8$. The remaining numerical parameters for the optimisation algorithm are as given in Table 1. The initial data was taken to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function taking the value 1 inside $B_1(0,0)$ (a circle of radius 1 centred at the origin) and -1 in $\Omega/B_1(0,0)$. The target data was taken to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function taking the value 1 inside $B_1(3,0)$ and -1 in $\Omega/B_1(0,0)$. Figure 9 shows



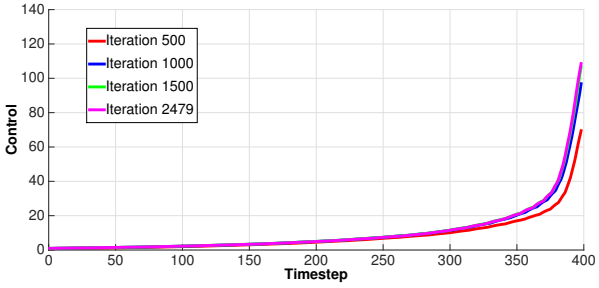
(a) Minimum value of the control for a series of iterations against time for the algorithm without volume constraints.



(b) Maximum value of the control for a series of iterations against time for the algorithm without volume constraints.

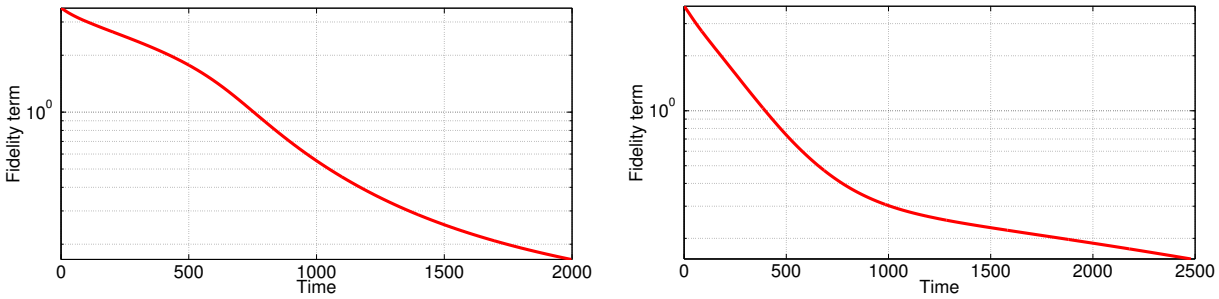


(c) Minimum value of the control for a series of iterations against time for the algorithm with volume constraints.



(d) Maximum value of the control for a series of iterations against time for the algorithm with volume constraints.

Figure 7: Minimum and maximum values of the control η with and without the volume constraint for the example of §4.2.



(a) Without the volume constraint

(b) With the volume constraint

Figure 8: The value of the fidelity term versus the number of iterations for the experiments of §4.2 with and without the volume constraint.

the initial and target diffuse interface data. To illustrate the effect of the choice of initial guess

(a) Initial data (φ^0).

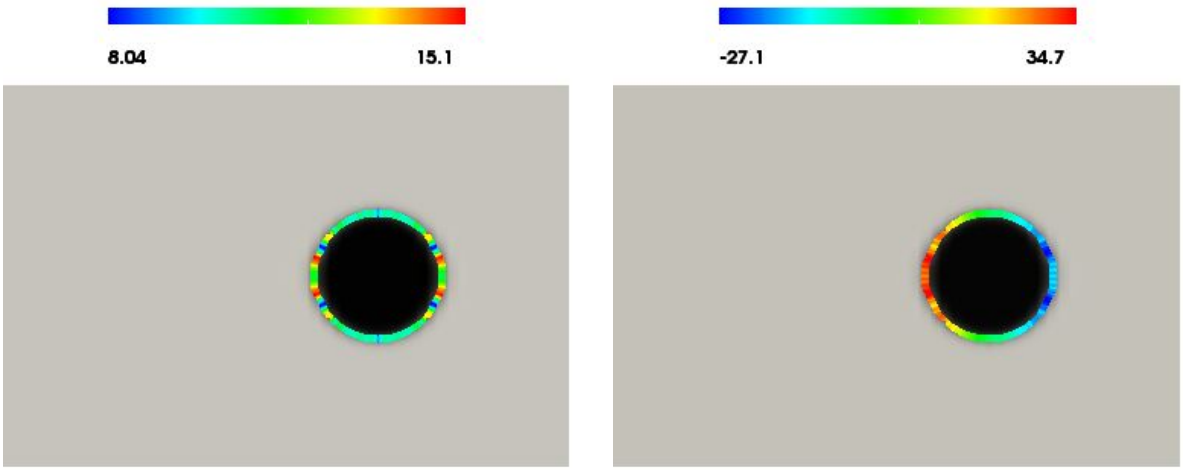
(b) Target data (φ_{obs}).

Figure 9: Initial and target data for the examples of §4.4.

on the algorithm, we consider two different values for the initial guess, firstly we set $\eta = 0$ and secondly we set $\eta = \boldsymbol{c} \cdot \nabla\varphi$, where $\boldsymbol{c} = (2.5, 0)$, i.e., in the latter case the initial guess depends on the solution to the Allen-Cahn equation. In both cases we used the algorithm with the volume constraints. With the zero initial guess the algorithm took 3262 iterations to meet the stopping criteria corresponding to a CPU time of 320433 seconds. With the second choice of initial guess the algorithm took 2056 iterations to meet the stopping criteria corresponding to a CPU time of 228173 seconds respectively.

Figure 10 shows the zero level-set of the computed solution using the optimal control at the final time. The curve corresponding to the zero level-set is shaded by the value of the control with the background shading corresponding to the target data. In both cases the position of the computed curve (zero level-set) with the optimal control shows good agreement with the target data. Figure 11 shows snapshots of the computed zero level-sets with the two different initial guesses. For the case with the initial value of $\eta = 0$, we observe in Figure 11(a) that the interface remains close to the initial position for most of the time of the simulation, and at the very last moment it shrinks to a point with a new phase nucleated at the position of the target data corresponding to a change in topology. With the second choice of initial guess ($\eta = \boldsymbol{c}\cdot\nabla\varphi$) we observe in Figure 11(b) that there is a gradual motion towards the target position with no

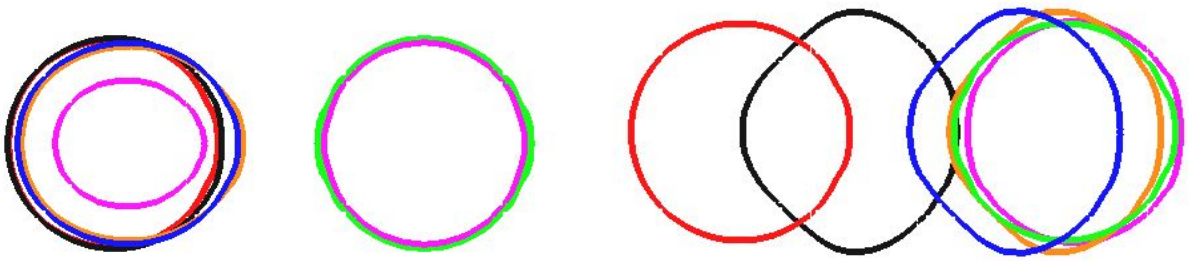
changes in topology. Figure 12 shows the area enclosed by the zero level-set of the computed solution with the optimal control with the two different initial guesses together with the linear interpolant of the areas of the data. We observe a sharp increase in area towards the end of the time interval with the zero initial guess as the new phase is nucleated. With the second choice of initial guess, the area of the computed curve exhibits a good fit to the linear interpolant of the areas of the data.



(a) With initial guess $\eta = 0$

(b) With initial guess $\eta = \boldsymbol{c} \cdot \nabla\varphi$

Figure 10: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments of §4.4. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.



(a) With initial guess $\eta = 0$

(b) With initial guess $\eta = \boldsymbol{c} \cdot \nabla\varphi$

Figure 11: Zero level-sets of the solutions computed ($\varphi(\boldsymbol{x}, t)$) with the optimal control ($\eta^*(\boldsymbol{x}, t)$) for the experiments of §4.4 at $t = 0$ (red), $t = 0.2$ (black), $t = 0.6$ (blue), $t = 0.7$ (orange), $t = 0.789$ (pink) and $t = 0.8$ (green). We observe the nucleation of a phase and a change in topology with the zero initial guess whilst there are no evident changes in topology and the zero level-set maintains a fixed topology in the case of the nonzero initial guess.

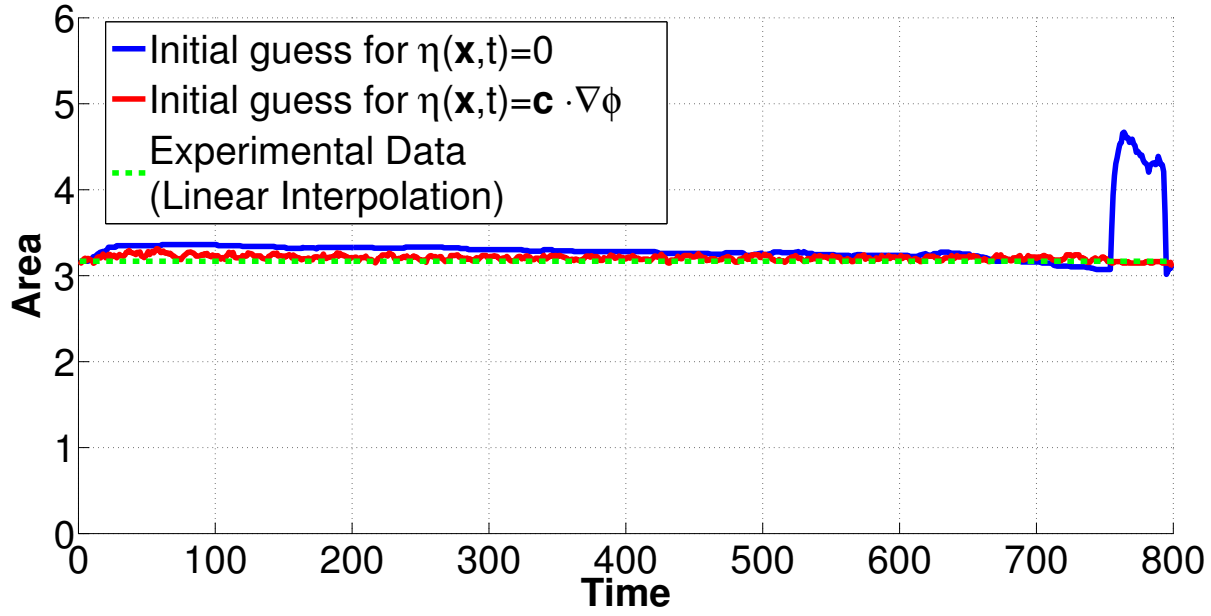


Figure 12: Area enclosed by the curve for the experiments of §4.4. A good fit to the linear interpolant of the areas is only observed with the nonzero initial guess. We observe a rapid increase in the area near the end time for the zero initial guess, this corresponds to the time at which a new phase is nucleated, c.f., Figure 11(a).

4.5 Application to multi-cell image data sets

We now apply the algorithm to the case of multi-cell image data sets. As a proof-of-concept we consider the simplest possible scenario where we have an initial and desired dataset both consisting of two cells that are well separated.

For the first experiment we defined the initial data and target data as follows. Defining the domain Ω to be $[-2,8]\times[-2,2]$ we defined the subdomains Ω_1, Ω_2, Ω_3 and Ω_4 to be the simply connected bounded domains with boundary curves $\Gamma_1,\Gamma_2,\Gamma_3$ and Γ_4 defined by (the curves Γ_1,Γ_2 and Γ_3 and Γ_4 are the zero level-sets of the diffuse interfaces shown in Figures 13(a) and 13(b) respectively).

$$\Gamma_1 := \left\{x\in\Omega \,|\, x_1^2+x_2^2-0.8^2+0.1\sin(4x_1)+0.1\sin(3x_2)=0\right\},$$
$$\Gamma_2 := \left\{x\in\Omega \,|\, \left(\frac{x_1}{2}-2\right)^2+(x_2-0.6)^2-0.7^2+0.1\sin\left(\frac{5x_1}{2}\right)+0.3\sin(2x_2)=0\right\},$$
$$\Gamma_3 := \left\{x\in\Omega \,|\, (x_1-0.4)^2+(x_2-0.5)^2-0.8^2+0.1\sin(6x_1)+0.1\sin(7x_2)=0\right\},$$
$$\Gamma_4 := \left\{x\in\Omega \,|\, \left(\frac{x_1}{2}-2.5\right)^2+(x_2-1)^2-0.7^2+0.1\sin\left(\frac{7x_1}{2}\right)+0.1\sin(1.5x_2)=0\right\}.$$

We then set the initial and target data to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function

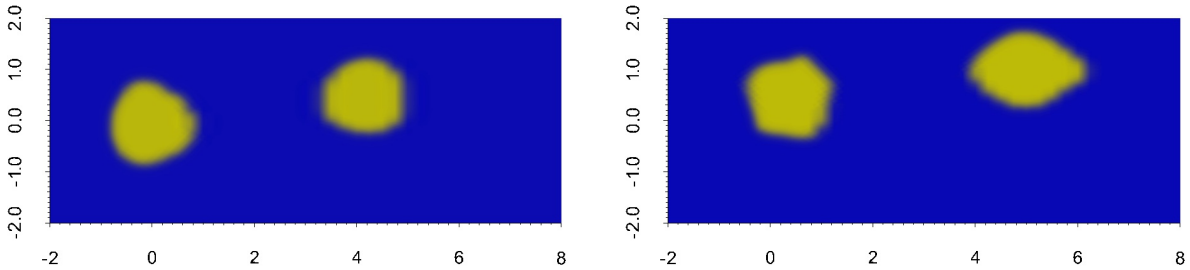
$$\varphi^0 = \begin{cases} 1 \text{ for } \boldsymbol{x}\in\Omega_1\cup\Omega_2, \\ -1 \text{ for } \boldsymbol{x}\in\Omega/\left(\Omega_1\cup\Omega_2\right), \end{cases} \quad \text{and} \quad \varphi_{obs} = \begin{cases} 1 \text{ for } \boldsymbol{x}\in\Omega_3\cup\Omega_4, \\ -1 \text{ for } \boldsymbol{x}\in\Omega/\left(\Omega_3\cup\Omega_4\right). \end{cases}$$

Figure 13 shows the initial and target diffuse interface data.

As previously, we compare the results of the algorithm with and without the volume constraint. For this experiment, the algorithm took 2035 iterations to meet the stopping criteria with no volume constraint and 2199 iterations with the volume constraint, corresponding to CPU times of 28608 and 105750 seconds respectively.

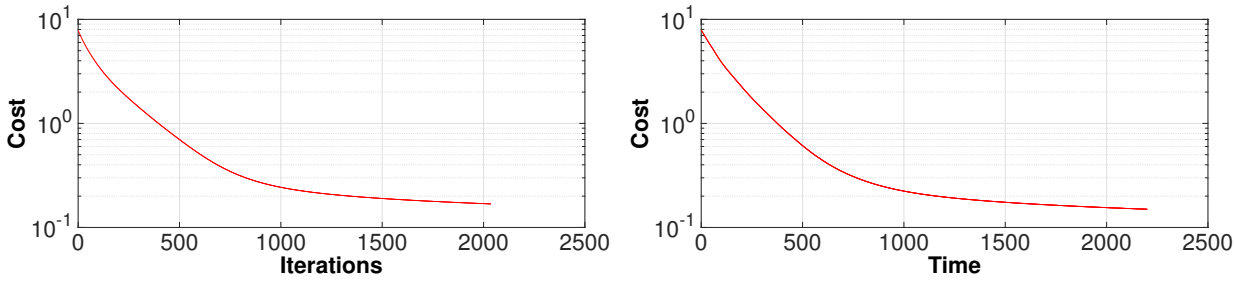
Figure 14 shows the value of the objective functional against the number of iterations of the optimisation algorithm with and without the volume constraint. Figure 15 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint shaded by the value of the control with the background shading corresponding to the target data. The results are similar to the single cell simulations of §4.2 with an initial rapid decrease in the cost followed by a subsequent gradual decrease. The cells (zero level-sets) computed with the optimal control show good agreement with the target data for both versions of the algorithm and for both cells. For each of the versions of the algorithm, both of the computed cells again posses a clearly defined "front" and "rear" similar to the single cell case.

Figure 16 shows the area enclosed by the zero-level set of the computed solution with the optimal control with and without the volume constraint together with the linear interpolant of the areas of the data. We observe analogous behaviour to the single cell. In terms of the computed cell morphologies, Figure 17 shows snapshots of the computed zero level-sets for the two different versions of the algorithm. We see that in this multi-cell setting the algorithm has implicitly solved the matching problem by generating two disjoint cells whose topology remains fixed throughout the evolution. We observe that the loss of volume in the case of no volume constraint corresponds to one of the cells in the intermediate snapshot (blue curve) enclosing a much smaller area.



(a) Initial data (φ^0). (b) Target data (φ_{obs}).

Figure 13: Initial and target data for the examples of §4.5.

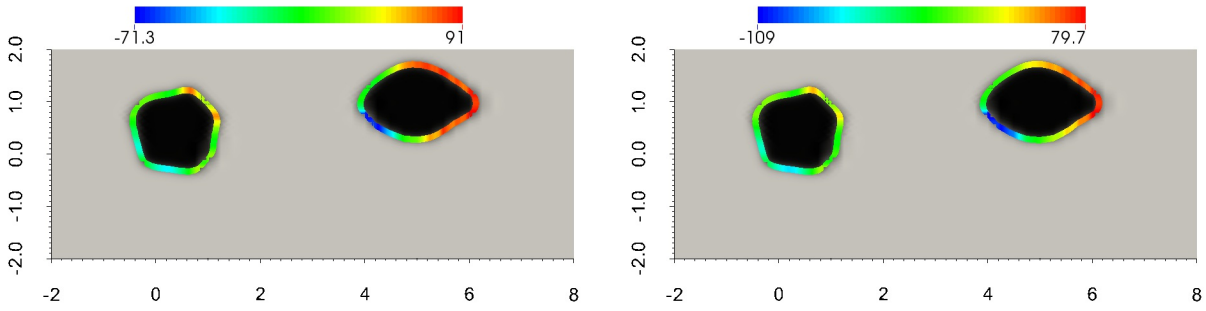


(a) Without the volume constraint (b) With the volume constraint

Figure 14: Cost functional versus the number of iterations for the examples of §4.5.

4.6 An example with topological change

Of course, in general our algorithm may generate cells whose topology is not fixed as in §4.4. To this end we report on another experiment.



(a) Without the volume constraint

(b) With the volume constraint

Figure 15: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) with and without the volume constraint for the examples of §4.5. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

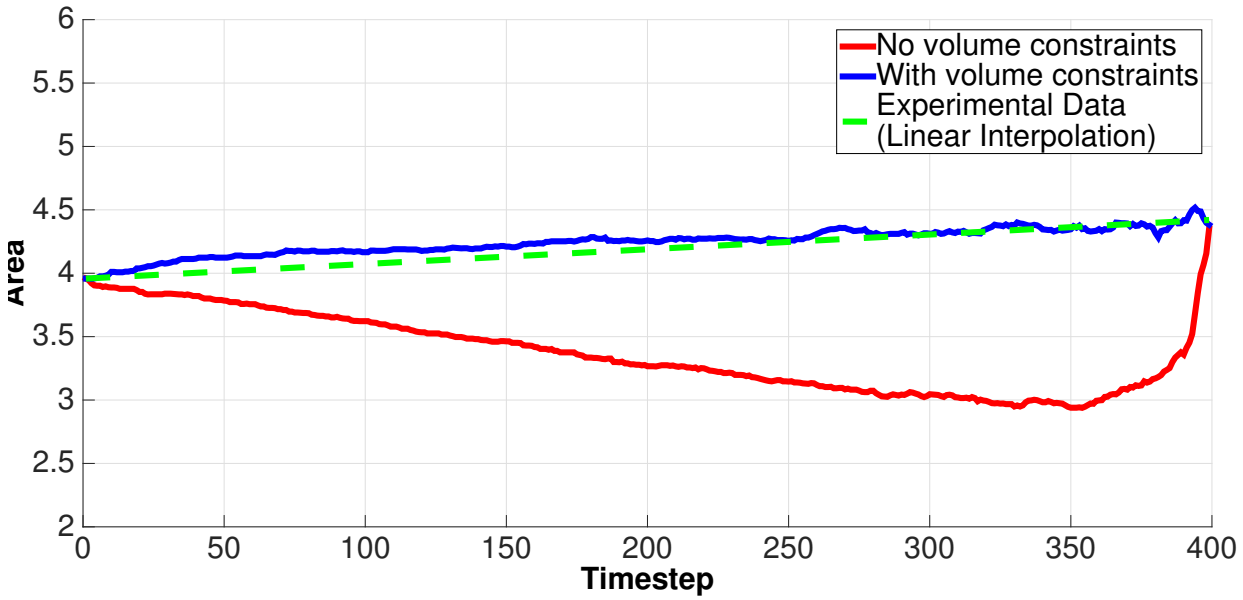
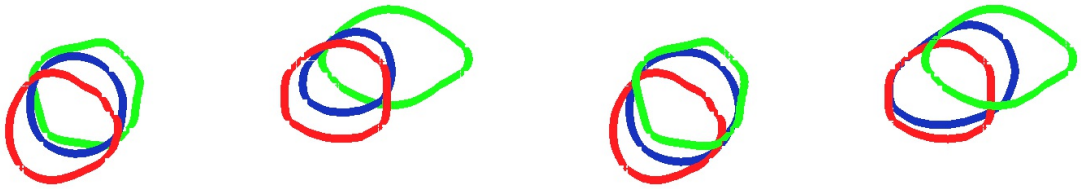


Figure 16: Area enclosed by the cell for the experiments of §4.5 with and without the volume constraint. As with the single cell data, the area (now the sum of the area of the two cells) shrinks considerably during the evolution without the volume constraint whilst a good fit to the linear interpolant of the area enclosed by the data is observed with the volume constraint.

Defining the domain Ω to be $[-2, 6.3] \times [-2.5, 2.5]$ we defined the subdomains $\Omega_1, \Omega_2, \Omega_3$ and Ω_4 to be the simply connected bounded domains with boundary curves $\Gamma_1, \Gamma_2, \Gamma_3$ and Γ_4



(a) Without the volume constraint

(b) With the volume constraint

Figure 17: Zero level-sets of the solutions computed with the optimal control for the examples of §4.5 with and without the volume constraint at $t = 0$ (red), $t = 0.35$ (blue) and $t = 0.4$ (green). The volume enclosed by both cells shrinks during the evolution without the volume constraint whilst this is not observed if the volume constraint is included. Both with and without the volume constraint, the implicit solution of the matching problem in this case generates two disjoint cells which do not change in topology.

defined by (see Figure 18)

$$\Gamma_1 := \left\{x \in \Omega \,|\, x_1^2 + x_2^2 - 0.9^2 + 0.1\sin(4.5x_1) + 0.11\sin(3x_2)) = 0\right\},$$
$$\Gamma_2 := \left\{x \in \Omega \,|\, (x_1 - 5)^2 + x_2^2 - 0.7^2 + 0.1\sin\left(\frac{5x_1}{2}\right) + 0.3\sin(2x_2) = 0\right\},$$
$$\Gamma_3 := \left\{x \in \Omega \,|\, (x_1 - 0.35)^2 + (x_2 - 0.7)^2 - 0.8^2 + 0.1\sin(6x_1) + 0.1\sin(7x_2) = 0\right\},$$
$$\Gamma_4 := \left\{x \in \Omega \,|\, (x_1 - 0.3)^2 + (x_2 - 1.1)^2 - 0.7^2 - 0.1\sin\left(\frac{7x_1}{2}\right) + 0.1\sin(1.5x_2) = 0\right\}.$$

We then set the initial and target data to be a smoothed (by running a few steps of the Allen-Cahn solver) version of the function

$$\varphi^0 = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_1 \cup \Omega_2, \\ -1 \text{ for } \boldsymbol{x} \in \Omega / \left(\Omega_1 \cup \Omega_2\right), \end{cases} \quad \text{and} \quad \varphi_{obs} = \begin{cases} 1 \text{ for } \boldsymbol{x} \in \Omega_3 \cup \Omega_4, \\ -1 \text{ for } \boldsymbol{x} \in \Omega / \left(\Omega_3 \cup \Omega_4\right). \end{cases}$$

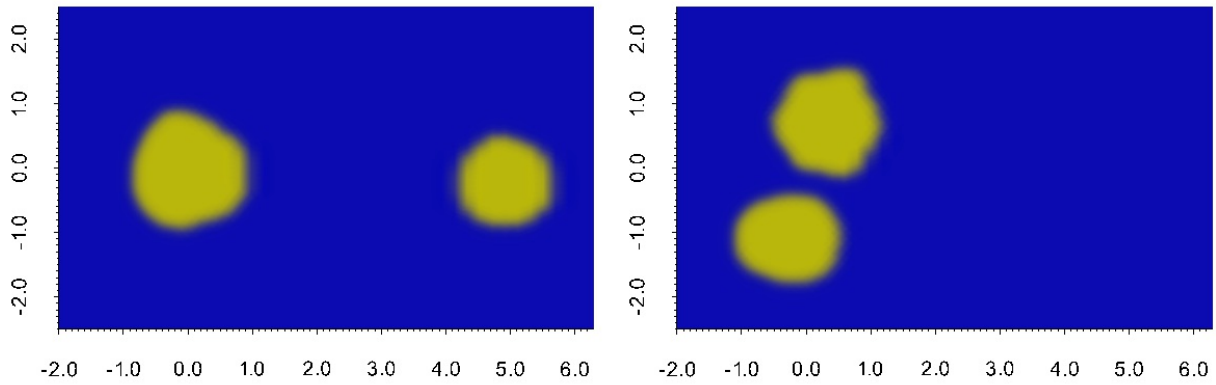
Figure 18 shows the initial and target diffuse interface data.

As previously, we compare the results of the algorithm with and without the volume constraint. For this experiment, the algorithm took 1960 iterations to meet the stopping criteria with no volume constraint and 1937 iterations with the volume constraint, corresponding to CPU times of 27553 and 93150 seconds respectively.

Figure 19 shows the value of the objective functional against the number of iterations of the optimisation algorithm with and without the volume constraint. Figure 20 shows the zero level-set of the computed solution using the optimal control at the final time with and without the volume constraint shaded by the value of the control with the background shading corresponding to the target data. The results are similar to the previous simulations with an initial rapid decrease in the cost followed by a subsequent gradual decrease and good agreement with the target data for both versions of the algorithm and for both cells. For each of the versions of the algorithm, both of the computed cells again posses a clearly defined "front" and "rear".

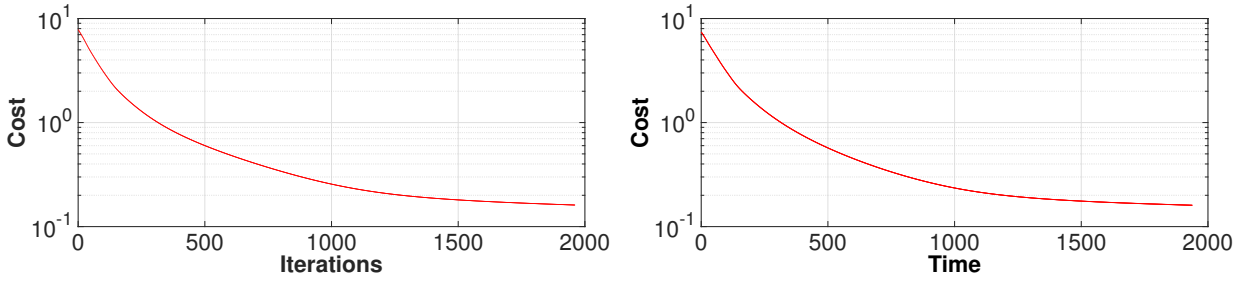
Figure 21 shows the area of the domain in which the computed solution is positive with and without the volume constraint together with the linear interpolant of the areas of the data. For this experiment we observe that the area enclosed by the computed cells differs significantly from the linear interpolant areas of the data both with and without the volume constraint. This may be due to the change in topology of the interface during the evolution. Figure 22 shows snapshots of the computed zero level-sets for the two different versions of the algorithm.

Unlike the previous examples we see that for this particular choice of initial and target data, the algorithm yields cells which change in topology with one of the curves shrinking until it disappears whilst the other splits eventually becoming two disjoint curves. Thus our algorithm generates trajectories corresponding to the annihilation (via shrinking) of one cell whilst the other cell splits to form the two cells observed in the image data set.



(a) Initial data (φ^0). (b) Target data (φ_{obs}).

Figure 18: Initial and target data for the examples of §4.6.

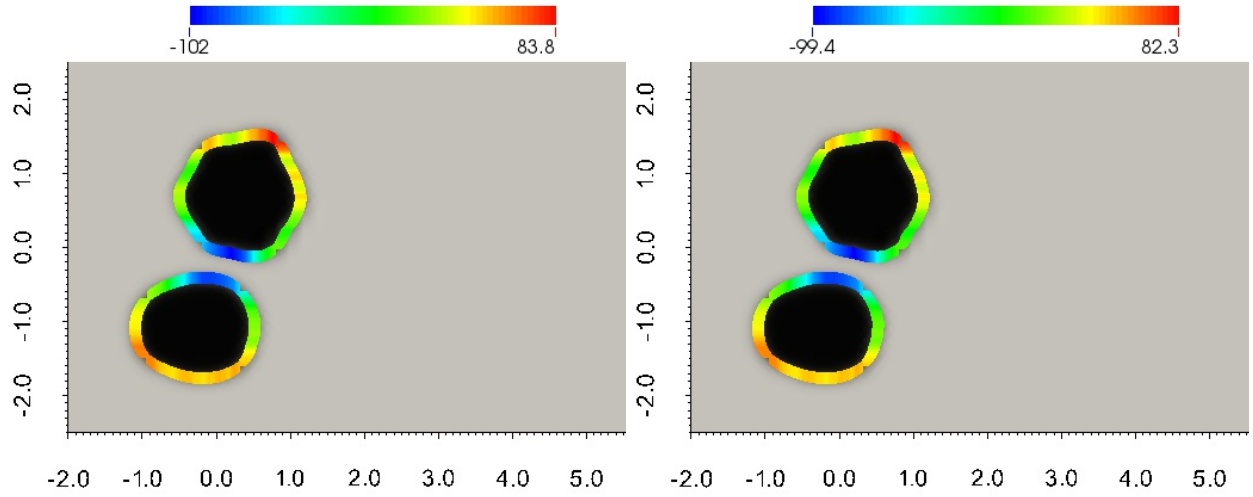


(a) Without the volume constraint (b) With the volume constraint

Figure 19: The value of the cost functional versus the number of iterations for the examples of §4.6 with and without the volume constraint. We observe a rapid decrease in the cost initially followed by a much more gradual decrease as we approach the minimum, this is as expected since the steepest descent algorithm is used for the update of the control.

4.7 Comments on the numerical experiments

The CPU times for each of the experiments is of the order of hours. For all the experiments the number of iterations required before the stopping criteria is met are similar, however this leads to simulations with the volume constraint taking approximately four times as long (in terms of CPU time) as those without the volume constraint. This is due to the iterative nature of the algorithm used to compute the Lagrange multiplier c.f., §A which necessitates multiple solves per timestep. We note that the CPU times of the algorithms may be too large for many applications. In light of this we mention that the stopping criteria we have used may be too strict for many applications and that a significant decrease in the cost function together with



(a) Without the volume constraint

(b) With the volume constraint

Figure 20: Zero level-set of the solutions ($\varphi(\boldsymbol{x}, T)$) computed using the approximated optimal control ($\eta^*(\boldsymbol{x}, t)$) with and without the volume constraint for the examples of §4.6. The curve (zero level-set of $\varphi(\boldsymbol{x}, T)$) is shaded by the approximated optimal control ($\eta^*(\boldsymbol{x}, T)$) and the background by the target data ($\varphi_{obs}(\boldsymbol{x})$). The color-bar corresponds to the scale for $\eta^*(\boldsymbol{x}, T)$. We see good agreement between the zero level-set of the data computed with the optimal control and the target data in both cases.

a reasonable fit to the data is observed after as few as 50 iterations (which reduces the CPU time by a factor of around 40) and that for many applications this level of fit may be sufficient hence the stopping criteria could be relaxed. Finally we mention that the current solution procedure based on uniform grids and serial solution of the forward and adjoint problems may be improved and we are currently investigating combining adaptive grids with a parallel solver for the forward and adjoint problems which gives a significant speed up but presents new technical challenges which we wish to avoid in this paper to maintain clarity of exposition.

5 Conclusion

In this study we presented a first step towards the development of cell tracking algorithms based on physical models for cell migration. The presented algorithm seeks to track whole cell morphologies and is applicable to single cell or multi-cell image data sets. Our approach may be regarded as a model fitting procedure in which a physically derived model for the evolution of the cell or cells is fitted to experimental image data sets. The algorithm is based on the theory of optimal control of PDEs and full details of the derivation and implementation of the algorithm are given. We also present a number of numerical experiments illustrating the performance of the algorithm with synthetic representative single cell and multi-cell image data sets.

The key novelty of our approach is that the model for the evolution of the cell (or cells), which drives the tracking procedure, is based on a relevant simplification of existing physically derived models for cell motility that reproduce many experimentally observed aspects of cell migration (e.g., [4]). Thus this study is an important step towards the development of cell tracking algorithms in which the recovered trajectories are physically meaningful. This is in contrast to the majority of existing algorithms for whole cell tracking in which the models for the evol-

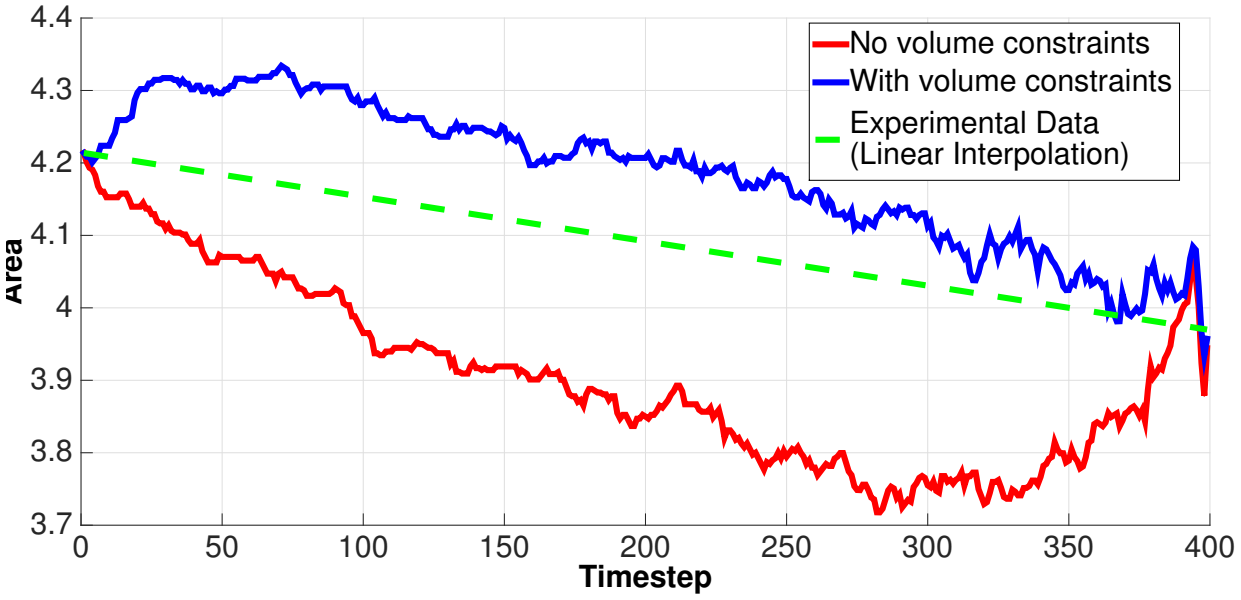
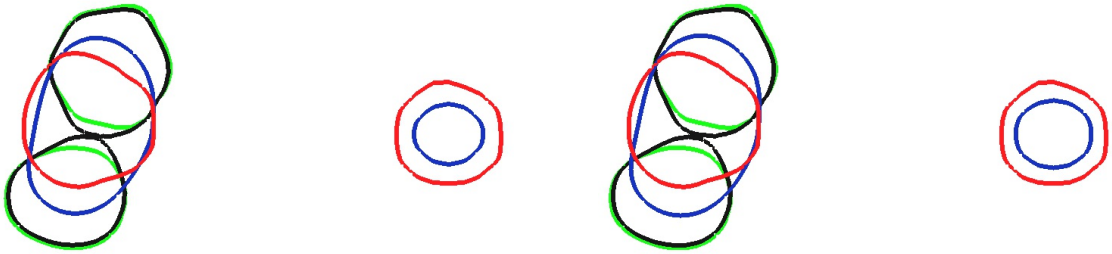


Figure 21: Area enclosed by the cell for the experiments of §4.6 with and without the volume constraint. As with the single cell data, the area (now the sum of the area of the two cells) shrinks considerably during the evolution without the volume constraint, whilst the incorporation of the volume constraint yields a better fit to the linear interpolant of the areas. Unlike the previous examples however, even with the volume constraint the fit to the linear interpolant of the areas of the data is poor.



(a) Without the volume constraint

(b) With the volume constraint

Figure 22: Zero level-sets of the solutions computed with the optimal control for the examples of §4.6 with and without the volume constraint at $t = 0$ (red), $t = 0.3$ (blue), $t = 0.396$ (black) and $t = 0.4$ (green). We observe that in this case the difference between the two schemes is less pronounced. It is also clear that both with and without the volume constraint, the implicit solution of the matching problem in our algorithm in this case leads to the annihilation of one cell (as it shrinks to a point) while the other cell splits with the zero level-set changing in topology from a single closed curve to two disjoint closed curves.

ution of the cell that underly the tracking procedure are purely geometric in nature neglecting completely the physics of cell migration [15,16,29]. One significant advantage of the approach to cell tracking we propose, is that the physics of the model driving the evolution of the cell are reflected in the recovered dynamic data. Thus it is possible to encode physical features of cell migration into the tracking procedure. We illustrate this fact by including volume conservation

in the model. Comparing the results of the tracking algorithm with and without volume conservation, we observe, that in a number of simulations neglecting volume conservation leads to physically unrealistic cell morphologies with a significant reduction of the cell volume in the recovered morphologies whilst this undesirable effect is no longer evident if volume conservation is included. We note that volume conservation is more physically relevant in three dimensions. This is since large changes in volume in two-dimensional imaging data of cells (i.e., the area of the projections of the cell on to a two-dimensional plane) are often observed in experimental data despite the cells conserving their enclosed (three-dimensional) volume.

Of course volume conservation is only an example of the kind of biology or physics one may wish to encode in the algorithm. A number of models that fit into our framework have been proposed incorporating more complex biophysics such as spontaneous curvatures [9], for Helfrich type models, adhesion for the migration of cells on substrates or in the ECM [13], cell-cell or cell-obstacle interactions [4] and chemotaxis [12], these models are thus potential candidates for the model driving the evolution in our tracking algorithm.

We work with diffuse interface representations of the cell membrane to make use of the mature theory for the optimal control of semilinear PDEs. One attractive aspect of this approach is that, as we do not require sharp interface representations of the cell membrane, it may be possible therefore to work directly with the raw experimental image data set without any need for segmentation. However the diffuse interface or phase field framework we employ does make the algorithm computationally intensive as evidenced by the relatively large CPU times for our experiments and a key area for future work is to investigate improvements in the computational efficiency of the algorithm. This need is especially evident if one wishes to track cells in $3d$, as although our theoretical framework applies equally to this setting the computational cost becomes prohibitive. Computational aspects under investigation include

- Spatial and temporal adaptivity which is challenging in this setting as the solution of the state equation enters the adjoint equation.
- Alternative update schemes for the control to the simple yet robust gradient based update considered in this study.
- Parallelisation and the development of fast solvers for the solution of the state and adjoint equations.

Our initial numerical investigations suggest that with a combination of the techniques outlined above it is possible to efficiently track $3d$ cell migration and we report on this elsewhere. Other potentially attractive directions for future work would be to consider higher order finite element spaces for the discretisation of the forward and adjoint problems or the use of spectral element methods, both of which may allow a more accurate solution of the forward and adjoint problem with fewer degrees of freedom, hence reducing the memory requirements.

Investigating the performance of the algorithm with real biological data for different cell types and in different environments is an important and worthwhile task. We are currently applying the algorithm to the tracking of in vivo neutrophil migration and intend to report on this elsewhere. As mentioned previously one interpretation of the forcing η^* is that it accounts for both protrusive forces generated by polymerisation of actin at the leading edge of the cell together with contractile forces generated by the action of myosin motors at the cell rear. Thus a potential avenue for assessing the plausibility of the cell tracks computed with our algorithm would be to compare the computed η^* with experimental imaging data on the location of polymerised actin and myosin-II on the cell membrane with the expectation being that regions in which the computed forcing η^* is positive would correspond to regions rich in polymerised actin and regions in which the computed forcing η^* is negative would correspond to regions rich in myosin-II. There are also many extensions of our approach which are likely to prove useful in applications. Our algorithm could equally be applied to the identification of (possibly time-dependent) parameters in models for cell migration (e.g., a spatially constant forcing or material parameters such as surface tension or bending rigidity) however in this case it is likely

that the sharp interface approach we propose in [5] will be more efficient. As observed in some of the experiments we report on, the framework we employ allows changes in topology of the cells. Whilst this may be desirable for some applications, e.g., tracking cells beyond cell division or cell fusion, in many biological experiments the topology of the cells is fixed. Our experiments suggest that topological changes arise primarily in the case of multi-cell image data sets. In this setting it should be possible to track the evolution of certain topological invariants (or more specifically diffuse interface representations of such invariants) and use these as an indicator for when the computed cells are changing in topology. The user could then manually reduce the multi-cell tracking problem to multiple single cell (or smaller scale multi-cell) tracking problems by specifying the correspondence between cells in different frames, with the hope that changes in topology do not occur for these new problems. The model we propose for the evolution in this study is a simplification of more general physically relevant models in which bulk or surface PDEs for the biochemistry are coupled to a geometric evolution law for the motion. An important area for future work is the extension of the framework to this more general setting. We note that the phase field approach we employ makes it computationally straightforward to couple the geometric evolution law for the motion to bulk PDEs (posed either within the cell or in the extra-cellular matrix) [8, 10, 13].

We hope that our optimal control-model fitting based framework is a useful first step towards incorporating advances in the modelling of cell migration into cell tracking algorithms.

Acknowledgments

This work (AM, VS and CV) is supported by the Engineering and Physical Sciences Research Council, UK grant (EP/J016780/1) and the Leverhulme Trust Research Project Grant (RPG-2014-149). KB was partially supported by the Embirikion Foundation Grant (2011-2014)-Greece. The computations were carried out using the computational resources of the School of Mathematics and Physical Sciences at the University of Sussex.

A Numerical solution of the forward and adjoint problems

A.1 Discretisation of the state equation

We introduce the variational form for the forward problem (2.8) defined as follows. Find $(\varphi, \lambda) \in \mathrm{L}_2([0,T];\mathrm{H}^1(\Omega)) \times L^2(0,T)$ such that

$$\int_\Omega \partial_t \varphi \psi d\mathbf{x} + \int_\Omega \nabla\varphi \cdot \nabla\psi d\mathbf{x} = \frac{1}{\varepsilon}\int_\Omega (c_G\eta - \lambda)\psi d\mathbf{x} - \frac{1}{\varepsilon^2}\int_\Omega G'(\varphi)\psi d\mathbf{x} \quad \forall \psi \in \mathrm{H}^1(\Omega).$$

Let $\mathcal{T}$ be a decomposition of Ω into simplexes (for simplicity we assume Ω is polygonal). We define the finite element space

$$\mathbb{V} := \{\psi_h \in \mathrm{H}^1(\Omega) \cap C^0(\Omega) : \psi_h|_k \in \mathbb{P}^1 \quad \forall k \in \mathcal{T}\}. \tag{A.1}$$

For the time discretisation employ an implicit-explicit method where the diffusive term is treated implicitly and the reaction terms explicitly. Introducing the shorthand for a time discrete sequence $f^n := f(t^n)$ and a uniform timestep τ with $T = M\tau$, $M \in \mathbb{N}$, the fully discrete scheme reads, for $n = 0, \ldots, M-1$, given $\varphi_h^n, \eta_h^n \in \mathbb{V}$ find $(\varphi_h^{n+1}, \lambda^{n+1}) \in \mathbb{V} \times \mathbb{R}$ such that

$$\frac{1}{\tau}\int_\Omega (\varphi_h^{n+1} - \varphi_h^n)\psi_h d\mathbf{x} + \int_\Omega \nabla\varphi_h^{n+1} \cdot \nabla\psi_h d\mathbf{x} = \frac{1}{\varepsilon}\int_\Omega (c_G\eta_h^n - \lambda^{n+1})\psi_h d\mathbf{x} - \frac{1}{\varepsilon^2}\int_\Omega \Lambda^h(G'(\varphi_h^n))\psi_h d\mathbf{x} \quad \forall \psi_h \in \mathbb{V},$$

where $\Lambda^h : C^0(\Omega) \to \mathbb{V}$ denotes the Lagrange interpolant.

We solve the above problem using the iterative technique introduced and studied in [21], which uses a bisection method for the Lagrange multiplier. In particular we seek an iterative sequence $\{\phi_h^{n+1,k}, \lambda^{n+1,k}\}_{k\geq 1}$ where $\phi_h^{n+1,k}$ solves

$$\frac{1}{\tau}\int_\Omega (\varphi_h^{n+1,k} - \varphi_h^n)\psi_h d\mathbf{x} + \int_\Omega \nabla\varphi_h^{n+1,k}\cdot\nabla\psi_h d\mathbf{x} = \\ \frac{1}{\varepsilon}\int_\Omega (c_G\eta_h^n - \lambda^{n+1,k})\psi_h d\mathbf{x} - \frac{1}{\varepsilon^2}\int_\Omega \Lambda^h(G'(\varphi_h^n))\psi_h d\mathbf{x} \quad \forall \psi_h \in \mathbb{V},$$

with $\lambda^{n+1,1} = -\frac{2\varepsilon}{\tau} + 1$, $\lambda^{n+1,2} = \frac{2\varepsilon}{\tau} - 1$ and $\{\lambda^{n+1,k+1}\}_{k\geq 2}$ satisfying

$$\lambda^{n+1,k+1} = \lambda^{n+1,k} + \frac{\left(\lambda^{n+1,k} - \lambda^{n+1,k-1}\right)\left(M_\varphi^{n+1} - \int_\Omega [\varphi_h^{n+1,k}]_+\right)}{\left(\int_\Omega [\varphi_h^{n+1,k}]_+ - \int_\Omega [\varphi_h^{n+1,k-1}]_+\right)},$$

where we recall

$$M_\varphi^{n+1} := \int_\Omega [\varphi_h^0]_+ + \frac{(n+1)\tau}{T}\left([\varphi_{obs}]_+ - [\varphi_h^0]_+\right) \mathrm{d}\boldsymbol{x}.$$

We deem this iteration to have converged when $|\lambda^{n+1,k+1} - \lambda^{n+1,k}| < \text{tol}$.

The discretisation of the forward problem without the volume constraint is as above with $\lambda = 0$. The discretisation of the adjoint problem without the volume constraint is the same as above as the Lagrange multiplier $\lambda(t)$ does not enter the adjoint problem (2.15).

References

[1] D. Bray, Cell movements: from molecules to motility, Routledge, 2001.

[2] Y. Xiong, P. A. Iglesias, Tools for analyzing cell shape changes during chemotaxis, Integrative Biology 2 (11-12) (2010) 561–567.

[3] E. Meijering, O. Dzyubachyk, I. Smal, Methods for cell and particle tracking, Methods in enzymology 504 (2012) 183.

[4] C. M. Elliott, B. Stinner, C. Venkataraman, Modelling cell motility and chemotaxis with evolving surface finite elements, Journal of The Royal Society Interface 9 (76) (2012) 3027–3044.

[5] W. Croft, C. M. Elliott, G. Ladds, B. Stinner, C. Venkataraman, C. Weston, Parameter identification problems in the modelling of cell motility, Journal of Mathematical Biology (2014) 1–38.

[6] F. Haußer, S. Rasche, A. Voigt, The influence of electric fields on nanostructures—simulation and control, Mathematics and Computers in Simulation 80 (7) (2010) 1449–1457.

[7] F. Haußer, S. Janssen, A. Voigt, Control of nanostructures through electric fields and related free boundary problems, in: Constrained Optimization and Optimal Control for Partial Differential Equations, Springer, 2012, pp. 561–572.

[8] D. Shao, W. Rappel, H. Levine, Computational model for cell morphodynamics, Physical review letters 105 (10) (2010) 108104.

[9] W. Marth, A. Voigt, Signaling networks and cell motility: a computational approach using a phase field description, Journal of mathematical biology (2013) 1–22.

[10] F. Ziebert, S. Swaminathan, I. Aranson, Model for self-polarization and motility of keratocyte fragments, Journal of The Royal Society Interface.

[11] M. Neilson, J. Mackenzie, S. Webb, R. Insall, Modelling cell movement and chemotaxis pseudopod based feedback, SIAM Journal on Scientific Computing 33 (3) (2011) 1035–1057.

[12] M. Neilson, D. Veltman, P. van Haastert, S. Webb, J. Mackenzie, R. Insall, Chemotaxis: A feedback-based computational model robustly predicts multiple aspects of real cell behaviour, PLoS biology 9 (5) (2011) e1000618.

[13] D. Shao, H. Levine, W.-J. Rappel, Coupling actin flow, adhesion, and morphology in a computational cell motility model, Proceedings of the National Academy of Sciences 109 (18) (2012) 6851–6856.

[14] K. M. Henry, L. Pase, C. F. Ramos-Lopez, G. J. Lieschke, S. A. Renshaw, C. C. Reyes-Aldasoro, Phagosight: an open-source matlab® package for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model, PloS one 8 (8) (2013) e72636.

[15] L. Bosgraaf, P. van Haastert, T. Bretschneider, Analysis of cell movement by simultaneous quantification of local membrane displacement and fluorescent intensities using quimp2, Cell motility and the cytoskeleton 66 (3) (2009) 156–165.

[16] R. A. Tyson, D. Epstein, K. Anderson, T. Bretschneider, High resolution tracking of cell membrane dynamics in moving cells: an electrifying approach, Mathematical Modelling of Natural Phenomena 5 (01) (2010) 34–55.

[17] 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, Integrative Biology 2 (11-12) (2010) 687–695.

[18] M. Vierling, Parabolic optimal control problems on evolving surfaces subject to pointwise box constraints on the control–theory and numerical realization, Interfaces and Free Boundaries 16 (2) (2014) 137–173.

[19] F. Tröltzsch, Optimal control of partial differential equations: Theory, methods and applications, Vol. 112, AMS Bookstore, 2010.

[20] X. Chen, Generation and propagation of interfaces in reaction-diffusion systems, Transactions of the American Mathematical Society 334 (2) (1992) 877–913.

[21] J. Blowey, C. Elliott, Curvature dependent phase boundary motion and parabolic double obstacle problems, in: Degenerate Diffusions, Springer, 1993, pp. 19–60.

[22] G. Bellettini, M. Paolini, Anisotropic motion by mean curvature in the context of finsler geometry, Hokkaido Mathematical Journal 25 (3) (1996) 537–566.

[23] M. Brassel, E. Bretin, A modified phase field approximation for mean curvature flow with conservation of the volume, Mathematical Methods in the Applied Sciences 34 (10) (2011) 1157–1180.

[24] Q. Du, C. Liu, X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions, Journal of Computational Physics 212 (2) (2006) 757–777.

[25] M. Hinze, R. Pinnau, M. Ulbrich, S. Ulbrich, Optimization with pde constraints., Mathematical Modelling: Theory and Applications. 23.

[26] J. Snyman, Practical mathematical optimization: an introduction to basic optimization theory and classical and new gradient-based algorithms, Vol. 97, Springer, 2005.

[27] T. F. Chan, L. A. Vese, Active contours without edges, Image processing, IEEE transactions on 10 (2) (2001) 266–277.

[28] D. Dormann, T. Libotte, C. J. Weijer, T. Bretschneider, Simultaneous quantification of cell motility and protein-membrane-association using active contours, Cell motility and the cytoskeleton 52 (4) (2002) 221–230.

[29] V. Kadirkamanathan, S. R. Anderson, S. A. Billings, X. Zhang, G. R. Holmes, C. C. Reyes-Aldasoro, P. M. Elks, S. A. Renshaw, The neutrophil's eye-view: Inference and visualisation of the chemoattractant field driving cell chemotaxis in vivo, PloS one 7 (4) (2012) e35182.