

Review

Implementing vertex dynamics models of cell populations in biology within a consistent computational framework



Alexander G. Fletcher ^{a,*¹}, James M. Osborne ^{b,c,1}, Philip K. Maini ^a, David J. Gavaghan ^b

^a Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford OX2 6GG, UK

^b Computational Biology Group, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK

^c Oxford Centre for Integrative Systems Biology, Department of Biochemistry, University of Oxford, South Parks Road, Oxford OX1 3QU, UK

ARTICLE INFO

Article history:

Available online 9 October 2013

Keywords:

Off-lattice

Cell-based models

Epithelial monolayers

Model comparison

Chaste

ABSTRACT

The dynamic behaviour of epithelial cell sheets plays a central role during development, growth, disease and wound healing. These processes occur as a result of cell adhesion, migration, division, differentiation and death, and involve multiple processes acting at the cellular and molecular level. Computational models offer a useful means by which to investigate and test hypotheses about these processes, and have played a key role in the study of cell–cell interactions. However, the necessarily complex nature of such models means that it is difficult to make accurate comparison between different models, since it is often impossible to distinguish between differences in behaviour that are due to the underlying model assumptions, and those due to differences in the *in silico* implementation of the model. In this work, an approach is described for the implementation of vertex dynamics models, a discrete approach that represents each cell by a polygon (or polyhedron) whose vertices may move in response to forces. The implementation is undertaken in a consistent manner within a single open source computational framework, Chaste, which comprises fully tested, industrial-grade software that has been developed using an agile approach. This framework allows one to easily change assumptions regarding force generation and cell rearrangement processes within these models. The versatility and generality of this framework is illustrated using a number of biological examples. In each case we provide full details of all technical aspects of our model implementations, and in some cases provide extensions to make the models more generally applicable.

© 2013 Elsevier Ltd. All rights reserved.

Contents

1. Introduction	300
1.1. Aim of work	300
2. Overview of individual cell-based modelling approaches	301
2.1. Vertex dynamics approaches	302
2.1.1. Honda and co-workers	302
2.1.2. Weliky and co-workers	302
2.1.3. Other models	303
3. Technical details of implementation of vertex dynamics models of epithelial monolayers	303
3.1. Honda and co-workers	303
3.2. Weliky and co-workers	304
3.3. Modifications to the model by Weliky and co-workers	305
3.4. Mesh restructuring operations	306
3.4.1. T1 swap – edge rearrangement	306
3.4.2. T2 swap – element removal	306

* Corresponding author.

E-mail address: alexander.fletcher@maths.ox.ac.uk (A.G. Fletcher).

¹ Joint first authors.

3.4.3. T3 swap – element intersection	306
3.4.4. Node switch – internal element intersection	306
3.5. Element division and removal	307
3.6. Numerical implementation and details of timestepping algorithm	307
3.6.1. Implementation within the Chaste framework	308
4. Example simulations	308
4.1. Monolayer growth	308
4.1.1. Previous work	309
4.1.2. Our implementation	309
4.2. Wound healing	312
4.2.1. Previous work	312
4.2.2. Our implementation	312
4.3. Active migration	314
4.4. Constrained geometries	315
4.5. Cell sorting	318
5. Discussion	321
Acknowledgements	324
Area and length computations	324
Shortest axis calculation for cell division	325
References	325

1. Introduction

Epithelial tissues form one of the major tissue types in vertebrates (Alberts et al., 2002). In epithelial tissue, cells are tightly bound together into sheets called epithelia, usually resting on a supporting bed of connective tissue. Strong intracellular protein filaments cross the cytoplasm of each cell and attach to specialized junctions in the plasma membrane. These junctions tie the surfaces of adjacent cells either to each other or to the underlying basal lamina. Epithelial cell sheets line all the cavities and free surfaces of the body. The specialized junctions between cells enable epithelia to form barriers that inhibit the movement of water, solutes, and cells from one body compartment to another.

The spreading of epithelial cell sheets and self-assembly of cells into tissues play a central role in morphogenesis, wound healing, tumour growth, tissue engineering and other biological processes. These phenomena occur as a result of cell division, differentiation, programmed cell death, adhesion and migration, and involve multiple processes acting on both a cellular and subcellular scale. Mathematical and computational models offer a useful means to investigate and test hypotheses about these processes, and have played a key role in the study of cell–cell interactions. To this end, a variety of different cell-based modelling approaches have been developed, which treat cells as individual objects, and thus allow for the study of cellular processes such as motility, adhesion, mitosis and apoptosis and their effect on the dynamic behaviour of epithelia.

To date very little comparative study of different cell-based approaches has been undertaken. While there are clear strengths and weaknesses with each approach, and therefore cases in which it is clear which approach is valid, there are other cases in which it is not clear. This raises the question of to what extent the results of model simulations may be artifacts of the modelling approach used or method of solution. It is difficult to make an accurate comparison between different models, in order to evaluate the impact of various assumptions on the behaviour of a model. This difficulty is exacerbated in recent multiscale models, which may couple the mechanical behaviour of a cell aggregate or epithelial monolayer to other processes such as the production and transport of signalling molecules (Smith et al., 2012; van Leeuwen et al., 2009). When comparing different constitutive assumptions using the same overall modelling approach, a computational framework is

required, which allows one to easily change the fine details of a model and its implementation.

1.1. Aim of work

In this work we focus on a class of cell-based models of epithelial populations termed vertex dynamics models (Hardin and Walston, 2004). These models share the central assumption that a cell may be approximated by a polygon (or polyhedron) whose vertices may move in response to forces. However the precise nature of the forces considered to act on cells, and the method of implementation and simulation, differs in each model. The implementation of algorithms of this type is technically challenging, and since the focus in much of the available literature has been primarily on the scientific outputs of such models, the technical details of their implementation have typically not been included. This makes it very difficult to reproduce the results of such studies (particularly in the absence of the source code), and, as a result, to compare directly competing modelling paradigms. In this paper we therefore give full details of all technical aspects of the implementation of the various algorithms, have made all of our code freely available to the community (see <http://www.cs.ox.ac.uk/chaste/download.html>), and have included code fragments to explain in detail how the various models can be implemented straightforwardly in a consistent manner.

This approach allows us to describe a fully consistent method of simulation of the two main classes of vertex dynamics models proposed to date. This is implemented in a consistent manner within the Chaste framework, a multi-purpose software library that we have developed to support computational simulations for a wide range of biological problems. Chaste is developed using an agile approach and comprises fully tested, industrial-grade software (Mirams et al., 2013). Current functionality has arisen from applications in the fields of cancer, cardiac physiology and soft-tissue mechanics. The project began in 2005, motivated by the observation that the modelling of a variety of physiological phenomena required both a generic mathematical modelling framework, and a generic simulation framework (Pitt-Francis et al., 2009).

The implementation of the main classes of published vertex dynamics model within the same computational framework allows a systematic study and comparison of the models to be undertaken.

The same solution methods underlie all of the component libraries within Chaste, hence we can be confident that observed differences in model simulations are due to different constitutive assumptions rather than different methods of solution. Our aim in this work is to provide full details of all technical aspects of the implementation of vertex dynamics models, as well as free access to the open source code, to allow others to understand, use and extend the models and implementation. Using a series of well-defined biological processes as case studies, we demonstrate the generality and versatility of vertex dynamics models. In each case we highlight the effect of different modelling assumptions on system behaviour.

The remainder of this paper is structured as follows. We start in Section 2 by reviewing previous work in the area of individual cell-based modelling, focussing on the vertex dynamics approach. We go on to describe the governing equations and cell rearrangement rules for two types of vertex dynamics model, and discuss how these models are implemented. In Section 4 we consider a number of biological examples, which illustrate how increasingly complex models may be implemented in a straightforward manner within our computational framework. We conclude in Section 5 by summarising our work and discussing further avenues for investigation.

2. Overview of individual cell-based modelling approaches

In this section we provide a brief overview of individual cell-based modelling approaches, focussing on the different types of vertex dynamics model that have been developed to describe epithelial cell populations.

The majority of existing spatial models of tissue growth and development treat a tissue as a continuum. Such models average over length scales that are much larger than the typical diameter of a cell. Cellular processes such as motility and adhesion can therefore only be considered in an averaged sense in a continuum framework. It can therefore be difficult to incorporate subcellular processes, such as signalling pathways or protein-level descriptions of cell–cell adhesion, into such models. In addition, it is often unclear whether an epithelial sheet comprises enough cells to justify a continuum approach; the intestinal crypt for example, which has been the focus of much theoretical investigation (De Matteis et al., 2012), contains around 300 cells in mice and 2000 cells in humans. In some circumstances, a discrete approach may therefore be more appropriate when modelling a system of interacting cells.

In individual cell-based models, the microscopic state of each cell is characterized by a state vector that includes information on the cell's position and velocity as well as its internal biological state (for example, the cell's progress through the cell cycle). Such models provide natural candidates for modelling the behaviour of multi-cellular systems. A number of discrete approaches have been developed for modelling cell populations. These vary in complexity from fixed-lattice cellular automata and the cellular Potts model to off-lattice centre dynamics and vertex dynamics models.

In cellular automata, physical space is discretized into a fixed lattice, on which cells can occupy sites. The rules for cell movement are formulated in terms of cells moving between lattice sites. Cell interactions are assumed to be spatially localized, so that a cell may only interact with other cells that are within some small neighbourhood. The current state of a cell is found using automaton rules and the state of the cell's neighbourhood at the previous time step. The system may be updated using a fixed time step algorithm, using synchronous or asynchronous updating (Lee et al., 1995), or else using an event-driven approach based on the Gillespie algorithm (Block et al., 2007). Although cellular automata replace laws of motion with much simplified rules for cell movement, they can exhibit remarkably realistic behaviour and sophisticated self-

organized structures on mesoscopic length scales (Deutsch and Dormann, 2005).

The simplicity of the cellular automaton approach renders it possible to simulate the interactions of large numbers of cells. This approach can thus be used to investigate the smaller-scale structures of tissues that more sophisticated but computationally intensive models cannot study. Despite these benefits, cellular automata have some limitations. First, one must choose how to partition physical space into automaton elements. In many cellular automata a fixed regular lattice is used, which can result in artificial spatial anisotropy and unrealistic cell behaviour; this may be overcome through the use of fixed irregular lattices, generated by Delaunay triangulation of random points (Block et al., 2007; Kansal et al., 2000). Second, due to the use of a fixed lattice, difficulties can arise in modelling cell mitosis and motility: for example, each cell can only move a multiple of its 'diameter' at each time step, leading to a discontinuous and unrealistic motion when considering individual cell tracks. It is also difficult to incorporate processes such as cell size changes, rigid body motions and mechanical deformations of cells into cellular automata.

In contrast to cellular automata, the cellular Potts model (Graner and Glazier, 1992) considers each cell to cover several lattice sites, and so allows for more realistic cell shapes. The movement of each cell is determined by some form of energy minimization using Monte Carlo simulation. The cellular Potts model can incorporate more detailed mechanical processes than cellular automata, such as cell membrane tension, cell–cell and cell–substrate adhesion, chemotaxis and cell volume conservation. It is also more computationally intensive than cellular automata, and thus more suitable for the detailed modelling of smaller populations of cells. As with fixed lattice cellular automata, it is difficult to relate experimental quantities to parameters within the cellular Potts model. Most notably, there is not a clear association between real time and the time step used when simulating the cellular Potts model by the Monte Carlo method; often the effective temperature of the system must be fitted so that a model evolves on the 'correct' timescale. Cell surface-energy simulations can also encounter problems due to the artificial anisotropies of a lattice geometry.

One can overcome some of the problems associated with cellular automata and the cellular Potts model by using off-lattice models. These allow continuous cell positions and enable the study of mechanical effects on cell populations in a straightforward manner. Off-lattice models offer a more physically realistic view of cell population dynamics, and may be parameterized by biophysical and kinetic parameters that can be determined experimentally.

Off-lattice models generally take one of three approaches to describing cell shape. Cells can be modelled either as quasi-spherical particles (Drasdo and Höhme, 1995) or deformable ellipsoids (Dallon and Othmer, 2004), which deform during mitosis and cell interactions, or else as Voronoi polygons or polyhedra (Meineke et al., 2001). Monte Carlo methods or Langevin equations are then used to simulate cell dynamics. Cell shape is not explicitly described in either the quasi-spherical particle or deformable ellipsoid formulations, and there is no explicit control of cell volume in the former. Thus details such as the elongation of cell shapes, as is observed when epithelial cells switch to fibroblast-like migration, cannot be accounted for using these approaches. These details are important in some circumstances. In the Voronoi approach, the spatial position of each cell is identified by its centre. The shape of the cell is defined to be the set of points in space that are nearer to the centre of the cell than the centres of any other cell. Voronoi diagrams generally provide a good fit to epithelial sheets with packed, polygonal-shaped cells. A cell's Voronoi contact surface is likely to be more realistic than its quasi-spherical contact

surface, especially in dense tissues with many neighbouring cells. Unfortunately, the Voronoi approach does not control cell volume.

To be physically realistic, a discrete cell population model must include an explicit representation of cell shape and explicit volume control, as well as cell growth regulation and control mechanisms. In vertex dynamics models, each cell is modelled as a polygon (or polyhedron in three dimensions), which represents the cell's membrane. Rules are set up to define how each vertex moves, based for example on the location of connected vertices and the surface area and volume of neighbouring cells. Thus it is the cell vertices, rather than the cell centres, which are allowed to move. Further rules may be employed for cell birth, cell death and cell rearrangement processes. Vertex dynamics models are particularly suited to the modelling of differential cell–cell adhesion. This is because connected vertices define the contact area between neighbouring cells. As the cell is defined by its boundary it is straightforward to couple cell volume to subcellular processes, such as mitosis, allowing processes such as contact inhibition to be considered. It is not as easy to include differential cell-substrate adhesion, however, since any change in drag terms on vertex movement would depend on all cells surrounding the vertex.

2.1. Vertex dynamics approaches

Vertex dynamics models have been used extensively to investigate the mechanisms and mechanics of autonomous epithelial monolayer deformations, which are of considerable importance in development, wound healing and tumour growth (Hardin and Walston, 2004). We now summarize previous work done in this field. A more comprehensive review of vertex dynamics models (up to 2004) and their relation to lattice-based and cell-centre models can be found in Brodland (2004). Hardin and Walston (2004) review the mechanisms and mechanics of rearrangements of epithelial sheets and compare them to those of non-epithelial ‘deep’ cells.

In this study we choose to focus on the models developed by Honda and co-workers and Weliky and co-workers, as these share the key aspects of most other vertex dynamics formulations, notably the assumption that cells move in response to mechanical forces in an overdamped manner. The evolution of each cell in these models is governed by motion of its vertices, which are typically assumed to obey deterministic equations of motion. It is standard to make the simplifying assumption that the motion of vertices is overdamped (Drasdo, 2000), and inertial terms are small compared to dissipative terms. This leads to first-order dynamics, with the evolution of the position \mathbf{r}_i of vertex i determined by

$$\eta_i \frac{d\mathbf{r}_i}{dt} = \mathbf{F}_i, \quad (1)$$

where $\mathbf{F}_i(t)$ denotes the total force acting on vertex i at time t and η_i denotes its drag coefficient.

The main difference between the models by Honda and co-workers and Weliky and co-workers lies in the definition of the force \mathbf{F}_i . In the work of Honda and co-workers (Honda and Eguchi, 1980), a free energy function is specified, whose gradient is assumed to exert a force on each vertex. In contrast, in the work of Weliky and Oster (1990) the forces acting on each vertex are given explicitly. Further details on these contrasting approaches are given in Sections 3.1 and 3.2.

2.1.1. Honda and co-workers

One of the earliest studies of vertex dynamics models was undertaken by Honda and Eguchi (1980), who consider a simplified two-dimensional cell aggregate with no gaps or overlaps. Their

model has been applied to wound healing (Nagai and Honda, 2006) and has also been extended to three dimensions using polyhedra (Honda et al., 2004). This model was used for determining how a two-dimensional cell aggregate might change into another one whose total boundary length is shorter than that of the original, without reducing the area of each cell. In further work, (Honda et al., 1982) study the process of corneal cell rearrangement as it proceeds without cell division. The model requires the assumption of boundary shortening, in which each cell evolves to minimize its total boundary length. The authors interpret this process as a function of the cellular contractile machinery, including actin.

In other work, (Honda et al., 1986) consider an example of checker-board formation: the luminal surface of the oviduct epithelium of an adult Japanese quail. The model includes two cell types: ciliated cells and gland cells. During sexual maturation, the cell pattern develops from a kagome-like pattern (hexagons surrounded by triangles) to a checker-board pattern. Assuming that adhesion is stronger between unlike cells than between like cells results in an ideal checker-board pattern, because all cell boundaries are edges along which unlike cells meet. In contrast, assuming that the boundary length of the cell pattern is minimized (caused by contraction of bundles of microfilaments running along lateral boundaries of the columnar epithelial cells while keeping contact between neighbouring cells) results in a honeycomb pattern. In reality, these two processes balance.

More recently (Nagai and Honda, 2001) use computational simulations to assess the ability of their vertex dynamics model of an epithelial monolayer to reproduce the ‘honeycomb’ pattern that is characteristic of such tissues. The authors have also investigated how cells within a three-dimensional aggregate are able to move and rearrange themselves despite the absence of free space (Honda et al., 2004). The authors examine the viscoelastic properties of the aggregate, in which an external force flattening a cell aggregate (e.g. centrifugation) quickly causes component cells to flatten, but after a delay the cells slowly rearrange to recover their original shape; although the cell aggregate remains flat. When a cell aggregate undergoes a mechanical deformation, it relaxes like an elastic material on short timescales, and like a viscous fluid on long timescales.

2.1.2. Weliky and co-workers

A distinct vertex dynamics approach has been developed by Weliky and co-workers, in which intra- and inter-cellular forces are modelled explicitly (Weliky and Oster, 1990). The authors consider the network of vertices to be attached by ‘zero-stiffness’ rods, and assume that the cytocortical pressure is the same at each cell node, and varies inversely with the cell area. Nodal pressure is assumed to drive cell protrusion, while nodal tensions drive contraction. This model has been used to study the behaviour of the epidermal enveloping layer during epiboly, the developmental process in which an embryo is encompassed by the ectoderm, in the teleost fish *Fundulus*. As cell rearrangements only are considered, this study did not incorporate cell proliferation.

In further work, (Weliky et al., 1991) extended their model to investigate cell behaviour during notochord morphogenesis in *Xenopus laevis*. The authors test whether a number of hypothesised cell behaviours are required for a vertex dynamics model of notochord morphogenesis in *Xenopus laevis* to reproduce the qualitative features observed experimentally. The authors show that only the combination of refractory boundaries, contact inhibition and polarized protrusive activity reproduces normal notochord development. Hence, these phenomena may cooperate to produce the coordinated cell movements witnessed within the developing notochord. In addition to vertices at each three-cell junction (where each vertex, not on the boundary, is contained in three cells, as is

the case in alternative models) the authors introduce vertices along the common boundaries between cells, to allow for ‘curved’ surfaces. Protrusions are absent in notochord cells where they contact adjacent somitic mesoderm. This is modelled by inhibiting protrusive activity at vertices on the cell sheet boundary. Motile cells are modelled as either unidirectionally or bidirectionally protrusive. Unidirectionally protrusive cells are modelled by randomly selecting a single cell vertex to protrude; these cells generally tend to remain isodiametric. Bidirectionally protruding cells are modelled by restricting protrusive activity to two new vertices located at opposite ends of the cell. In the absence of cell–cell interactions, cells exhibit directional persistence such that they tend to maintain their direction of motion. Cells interact with each other by contact inhibition of protrusive activity. Persistence of motion is modelled by assigning probabilities to each cell vertex, which allows a cell to change direction at each time step.

2.1.3. Other models

In addition to the various approaches described above, a number of other studies have been undertaken using vertex dynamics models. Brodland and co-workers have developed a vertex dynamics model based upon a finite element formulation, which has been used to describe the sorting of different cell types due to the cells’ different adhesive properties (Brodland and Chen, 2000a, 2000b). In further work, (Brodland and Veldhuis, 2002) consider the interactions between mitosis and cell shape and epithelium deformations. Unlike other vertex dynamics models, these authors’ model calculates the mechanical effect of the cytoplasm based on strain fields within the cell. A drawback with this model is a stiffening artifact associated with short cell edges (Brodland, 2007). In other work, (Farhadifar et al., 2007) use a vertex dynamics model to investigate the mechanism(s) by which cells in the *Drosophila* wing epithelium convert from an irregular to a hexagonal array shortly before hair formation. The authors take into account cell elasticity and junctional forces arising from cortical contractility and adhesion. The authors find that two different types of relaxed configurations occur: a solid-like hexagonal network and a liquid-like soft network.

As well as being used to gain insight into self-organization in epithelial tissues, vertex dynamics models have been used in the context of plant development. (Rudge and Haseloff, 2005) develop a computational model of plant morphogenesis at the cellular level. The authors include cell–cell signalling by diffusion of morphogens, as well as genetic regulation. Polar cell behaviour is included. The model is shown to reproduce plant-like cell proliferation, coordination of growth zones, and specification of cell behaviour by lineage and position.

3. Technical details of implementation of vertex dynamics models of epithelial monolayers

In this section we describe the specific constitutive assumptions made in each of the main classes of vertex dynamics model described above. We also provide full technical details of how such models are simulated in Chaste, and how various cell rearrangement processes associated with movement, cell division and cell death are implemented in practice.

In a vertex dynamics model of an epithelial monolayer, each cell is modelled as a polygon (in 2D) or polyhedron (in 3D), whose vertices may move. It is common to make the simplifying assumption that all cells in a monolayer are the same height, which allows one to treat the cells as two-dimensional, thereby reducing the complexity of the problem. We refer to the collection of polygonal cells as a ‘mesh’, which comprises a set of ‘elements’ and ‘vertices’. Each vertex is a point in space, defined by its location and

a unique identifying index; while each element is a polygon, defined by an ordered list of vertex indices and a unique identifying index. We refer to a pair of neighbouring vertices within an element as an ‘edge’. Fig. 1 illustrates the structure of the vertex mesh. We require that each vertex that is not on the edge of the mesh is contained in exactly three elements and vertices on the boundary of the mesh are in either one or two elements. This is analogous to Plateau’s equilibrium rules in foams (Taylor, 1976). If the mesh is in this initial state then it will remain so as it evolves. The evolution of the mesh is defined by the following processes:

1. equations of motion are defined to specify how vertices move in relation to each other, as described in Section 2.1;
2. elements may move past each other as a result of the mesh undergoing local rearrangement (see Section 3.4);
3. elements may divide into two as a result of cell division (see Section 3.5);
4. elements may be removed from the mesh as a result of cell death (see Section 3.5).

Each element in this mesh is associated with a cell, which can influence the evolution of the monolayer through the above processes by changing parameters and/or inducing cell division or death. These processes are discussed in the following sections.

We now discuss in more detail the methods used to define \mathbf{F}_i in each of the three classes of vertex dynamics model outlined in Section 2.1. We note that for specific, well-defined experimental protocols it is possible to calibrate complex off-lattice models (Sandersius et al., 2011). In general however, the extent to which each of these models have been validated against data or biological knowledge varies, and the extent to which model parameters are experimentally accessible also varies. In all three classes of model we adopt the notation that at time t , the monolayer comprises $N(t)$ cells of constant height (thus allowing us to treat the aggregate as two-dimensional), and that each cell k is associated with $n_k(t)$ vertices.

3.1. Honda and co-workers

The approach developed by Honda and co-workers, as discussed in Section 2.1.1, is to neglect the detailed biomechanical mechanisms underlying force generation with cells. Instead the authors propose phenomenological expressions describing constraints associated with the limited ability of each cell to undergo elastic

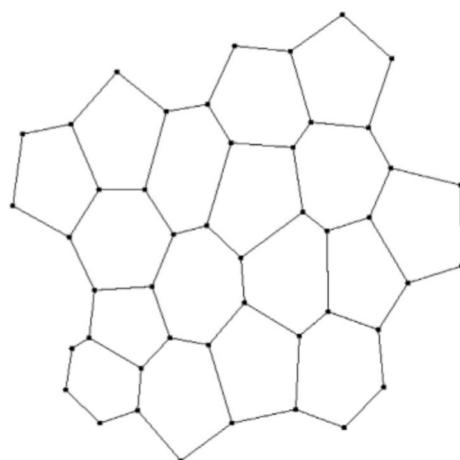


Fig. 1. Example vertex mesh. The monolayer is comprised of connected non-overlapping polygons with varying numbers of vertices.

deformations, volumetric changes and other movements due to adhesion to other cells. The authors define a ‘work function’ or ‘free energy’ U , which is intended to capture the main contributions to the potential energy of the system. This is then used to derive the force acting on each vertex. The total free energy is comprised of three parts: a deformation energy, U_D , a membrane surface tension energy, U_S and a cell–cell adhesion energy, U_A ,

$$U = U_D + U_S + U_A = \sum_{k=1}^N (U_D^k + U_S^k + U_A^k). \quad (2)$$

Here the sum is taken over all cells in the monolayer. The deformation energy term, U_D^k , aims to ensure that each cell k attains a target volume. It is defined by

$$U_D^k = \lambda (A_k - A_{0_k})^2, \quad (3)$$

where λ is a positive constant analogous to an elastic constant, A_{0_k} is the target area for cell k (which can change over time, for example as a result of cell growth and division), and A_k is the current area of cell k . For a non-self-intersecting (simple) polygon whose $n_k(t)$ vertices have positions $\{(x_j, y_j)\}_{j=0}^{n_k-1}$ (at time t) ordered counter-clockwise, the area A_k is given by

$$A_k = \frac{1}{2} \left| \sum_{j=0}^{n_k-1} (x_j y_{j+1} - x_{j+1} y_j) \right|. \quad (4)$$

Note that the sign of the area expression (4), without the absolute value, can be used to determine the ordering of the vertices of the polygon. If the sign is positive, then the polygon vertices are ordered counter-clockwise about the normal; otherwise they are ordered clockwise.

The membrane surface energy, U_S^k , aims to conserve cell membrane length and acts to drive each cell k to a circular/spherical shape. It is defined by

$$U_S^k = \beta (C_k - C_{0_k})^2, \quad (5)$$

where β is a positive constant, C_{0_k} is the target perimeter for cell k , and C_k is the perimeter of cell k . The natural perimeter is the circumference of a circular cell of area A_{0_k} ,

$$C_{0_k} = 2\sqrt{\pi A_{0_k}}. \quad (6)$$

This functional form is chosen as cells in free suspension will tend to a circular shape. The cell–cell adhesion energy, U_A^k , represents the free energy associated with bonds between each cell k and its neighbours. It is defined by

$$U_A^k = \sum_{j=0}^{n_k-1} \gamma_{kj} d_{kj}, \quad (7)$$

where γ_{kj} is a positive constant dependent on the cell types in contact and d_{kj} is the distance between the j th vertex of cell k and the next in an anticlockwise direction. Different cell types can have selective cell–cell adhesion, due to the different types of cadherin junctions and other cell–cell bonds in contact. A lower value of γ_{kj} represents a more stable or favourable contact between cell types.

The gradient of free energy ∇U is assumed to exert a force on each vertex of the monolayer. Therefore we can define the force applied to vertex i as

$$\begin{aligned} \mathbf{F}_i &= -\nabla_i U \\ &= -\nabla_i \sum_{k \in \mathcal{N}_i} \left(\lambda (A_k - A_{0_k})^2 + \beta (C_k - C_{0_k})^2 + \sum_{j=0}^{n_k-1} \gamma_{kj} d_{kj} \right), \end{aligned} \quad (8)$$

where $\nabla_i U = \nabla U|_{\mathbf{r}=\mathbf{r}_i}$ and \mathcal{N}_i denotes the set of element indices corresponding to cells that contain vertex i . In practice, we compute the right hand side of Equation (8) as described in Appendix A. An advantage of this model over the model proposed by Weliky and co-workers is that it permits contact-area dependent cell–cell adhesion as well as cell-volume control.

3.2. Weliky and co-workers

The approach developed by Weliky and co-workers, discussed in Section 2.1.2, is based on the assumption that deformation of the cortical actomyosin cytoskeleton is the primary mechanism underlying force generation. This involves cellular protrusion, arising from the hydrostatic and osmotic pressure within the actin gel and elastic forces due to actin polymer fibres; and cellular contraction, due to the movement of circumferential actin microfilament bundles and adherens junctions with neighbouring cells.

Based on these considerations, the authors assume that three components contribute to the force on each vertex from each cell containing that vertex. The first two terms describe the tensions in the cell wall, which act in the direction of the cell membrane and are proportional to the perimeter of the cell. The third term describes the cytocortical pressure, which is the difference between the osmotic pressure tending to expand the cytoskeleton and the restraining elastic pressure generated by the actin polymer fibres, and which is assumed for simplicity to be inversely proportional to the total area of the cell, and directed such that it bisects the internal angle at the vertex.

Let $m_i(t)$ be the number of elements containing vertex i at time t , then the total force acting on the vertex, $\mathbf{F}_i(t)$, is the sum of net forces \mathbf{f}_k originating from each cell k containing the vertex, given by

$$\mathbf{F}_i = \sum_{k=1}^{m_i} \mathbf{f}_k = \sum_{k=1}^{m_i} [\mathbf{p}_i^k + \hat{\mathbf{t}}_{il}^k + \hat{\mathbf{t}}_{ir}^k], \quad (9)$$

where

$$\mathbf{p}_i^k = \frac{\rho \hat{\mathbf{p}}_i^k}{A_k}, \quad \hat{\mathbf{t}}_{il}^k = \kappa \hat{\mathbf{t}}_{il} C_k, \quad \text{and} \quad \hat{\mathbf{t}}_{ir}^k = \kappa \hat{\mathbf{t}}_{ir} C_k, \quad (10)$$

are the components of force due to the internal pressure of the cell and the surface tension in each adjacent edge, respectively. The vector $\hat{\mathbf{p}}_i^k$ is a unit vector, outwards from the cell, bisecting the angle between adjacent edges, and $\hat{\mathbf{t}}_{il}$ and $\hat{\mathbf{t}}_{ir}$ are unit vectors parallel to the edge connecting adjacent vertices. These unit

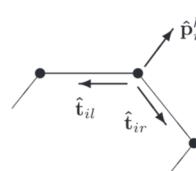


Fig. 2. Direction of $\hat{\mathbf{p}}_i^k$, $\hat{\mathbf{t}}_{il}^k$ and $\hat{\mathbf{t}}_{ir}^k$ for vertex i , contained in element k . The adjacent vertices are labelled l and r .

vectors are illustrated in Fig. 2. As before, the cross-sectional area and perimeter of each cell k are denoted by A_k and C_k respectively. The parameter ρ is used to represent a cell's resistance to changes in cross-sectional area and κ denotes the elastic modulus of the cell boundary (Weliky and Oster, 1990). These parameters are analogous to the parameters λ and β , respectively, in the model of Honda and co-workers described above. However, the effect of varying these parameters is different in each model. For example, increasing λ causes cells to be more resistant to changes in cross-sectional area, while increasing ρ leads to cells equilibrating to a smaller size; the latter is controlled by the parameter A_{0k} in the model of Honda and co-workers.

3.3. Modifications to the model by Weliky and co-workers

In this section we propose two modifications of the model proposed by Weliky and co-workers, detailed in Section 3.2. These allow the model to be applied to the types of example considered in Section 4, and which enable us to more fully compare the two approaches.

We first note that the model by Weliky and co-workers leads to elements whose areas vary as the number of vertices are increased, as shown in Fig. 3. Therefore, the model as presented in (Weliky and Oster, 1990) is not suitable for modelling cells with a specified target area. We therefore propose the following modification to the Weliky–Oster force laws (9)–(10) so that the equilibrium element area no longer depends on the number of vertices in the element. In order to deduce how the equilibrium area of an element depends on the number of vertices we consider an individual cell comprised of n_k vertices, arranged as a regular polygon. When the element is in an equilibrium state the forces exerted on each vertex are balanced. Balancing the forces \mathbf{p}_i^k , \mathbf{t}_{il}^k and \mathbf{t}_{ir}^k as defined in Equation (10) at each vertex in the element gives

$$\frac{\rho}{A_k} = 2\kappa C_k \sin\left(\frac{\pi}{n_k}\right). \quad (11)$$

The area, A_k , and perimeter, C_k , of a regular polygon can be related by the following relationship (Coxeter, 1973):

$$A_k = \frac{C_k^2 \cot\left(\frac{\pi}{n_k}\right)}{4n_k}. \quad (12)$$

Combining (11) and (12) to remove C_k yields the resting area of the element as

$$A_k = \left(\frac{\rho \sqrt{\cot\left(\frac{\pi}{n_k}\right)}}{4\kappa \sqrt{n_k} \sin\left(\frac{\pi}{n_k}\right)} \right)^{\frac{2}{3}}. \quad (13)$$

Fig. 3(a) demonstrates that there is exact agreement between this theoretical equilibrium area and simulations up to 20 vertices. This observation of area dependence for each element k leads to a modified force model, in which the parameter ρ in (10) is modified to depend on the number of vertices n_k by

$$\rho = 4A_{0k} \kappa \sin\left(\frac{\pi}{n_k}\right) \sqrt{n_k \tan\left(\frac{\pi}{n_k}\right)}, \quad (14)$$

where A_{0k} is the target area of element k . From here on we refer to this model as the modified Weliky–Oster model.

Fig. 3(b) shows how the equilibrium area varies with the number of vertices for both the modified Weliky–Oster model and the Nagai–Honda model (with $A_{0k} = 1.0$). We see that in both models the equilibrium area is equal to one for a sufficiently large value of n_k . The equilibrium area is less than one in the Nagai–Honda model when using a relatively small value of n_k , because the target perimeter is chosen assuming the element approximates a circle, as seen in Equation (6).

Our second modification involves the introduction of a force component representing membrane stiffness. This modification is required because in the case of the model by Weliky and co-workers, under conditions of high proliferation or compression, elements of the mesh can become non-convex and even self-intersect. To avoid these problems, here we describe an additional component for the Weliky–Oster force law, which represents resistance of the cell membrane to deformations from equilibrium. Let $m_i(t)$ be the number of elements containing vertex i at time t , then the total force acting on the vertex, $\mathbf{F}_i(t)$, is the sum of the net forces \mathbf{f}_k originating from each cell k containing the vertex, given by

$$\mathbf{F}_i = \sum_{k=1}^{m_i} \mathbf{f}_k = \sum_{k=1}^{m_i} \gamma_k (\theta_i^k - \theta_0^k)^{n_\theta} \hat{\mathbf{p}}_i^k. \quad (15)$$

Here θ_0^k denotes the target angle and is given by $\theta_0^k = 2\pi/n_k$, where n_k is the number of vertices in element k ; the vector $\hat{\mathbf{p}}_i^k$ is a unit vector, outwards from the cell, as used for the Weliky–Oster model in Section 3.2; and n_θ is a positive integer (taken to be 3 in all examples in this study). This extra force term is intended to represent the reaction of the cytoskeleton and other cellular

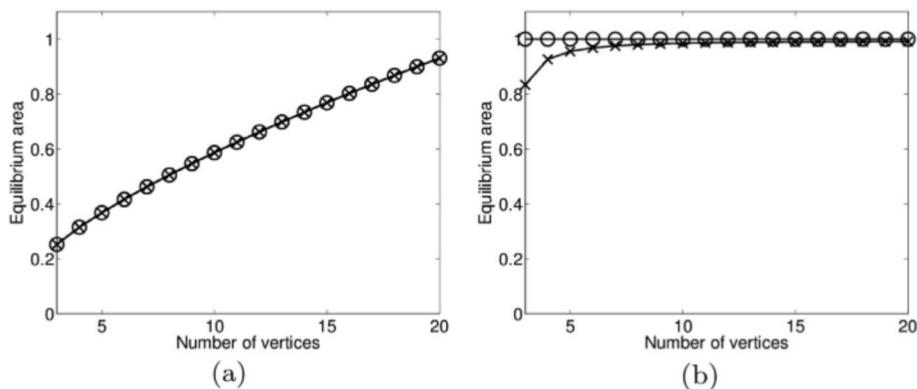


Fig. 3. How the equilibrium area of a single cell depends on its number of vertices, in the absence of any interactions with neighbouring cells. (a) The Weliky–Oster model; crosses represent the results of simulation and circles represent the analytical equilibrium area given by Equation (13). (b) The modified Weliky–Oster model (circles) and the Nagai–Honda model (crosses).

components to a large external force. The force acts to maintain the equilibrium angle at element vertices and thus preserve element convexity. In biological systems where natural elongation of epithelial cells occurs, the target angle θ_0^k may be modified accordingly. Note that while the precise form of the bending force defined in Equation (15) is chosen by us phenomenologically, a bending force with the same functional form but with $n_\theta = 1$ was proposed in a recent theoretical study of cell migration in the mouse embryo (Trichas et al., 2012). This is equivalent to the standard bond bending force associated with the angle between each pair of successive bonds in models of polymer chains (Allen and Tildesley, 1989). Evaluation of the effect of the precise functional form of such a bending force on the tissue-level behaviour will be a focus of future work.

3.4. Mesh restructuring operations

Independent of the chosen model for vertex movement, we need to ensure that the elements are always non-intersecting and to allow cells (or their associated elements) to form and break bonds. In order to achieve this we consider three types of elementary operations (Nagai and Honda, 2001, 2006). These operations are: edge rearrangement (a T1 swap); element removal (a T2 swap); and element intersection (a T3 swap). We emphasize that the need for these operations does not arise simply as a result of taking too large a time step when solving the equations of motion numerically. Rather, such rearrangements are necessary because each of the vertex dynamics models considered results in finite forces acting on a cell's vertices arbitrarily far from equilibrium; we do not consider 'hard body' interactions.

3.4.1. T1 swap – edge rearrangement

Element rearrangement, also known as a T1 swap, occurs when two vertices i, j are located less than a minimum threshold distance d_{\min} apart (Nagai and Honda, 2001), as shown in Fig. 4(a) and (b). This distance is much smaller than a typical element size; we undertake a full investigation of the effect of varying this and other model parameters on mechanical behaviour in a forthcoming study. The vertices are moved and placed a distance d_{sep} apart where $d_{\text{sep}} = k_{\text{sep}}d_{\min}$, where k_{sep} is known as the separation ratio. This rearrangement ensures that any vertex that is not on the boundary of the mesh is always contained in exactly three elements. If the vertex is on the boundary of the mesh then it is contained in one or two elements. The rearrangement also occurs when the short edge is on the boundary of the mesh, as shown in Fig. 4(c) and (d). This process allows elements on the boundary of the mesh to maintain a realistic polygonal shape. A special case for this type of rearrangement is if the two neighbouring vertices are only contained in a single element: in this case the two vertices are merged, as shown in Fig. 5.

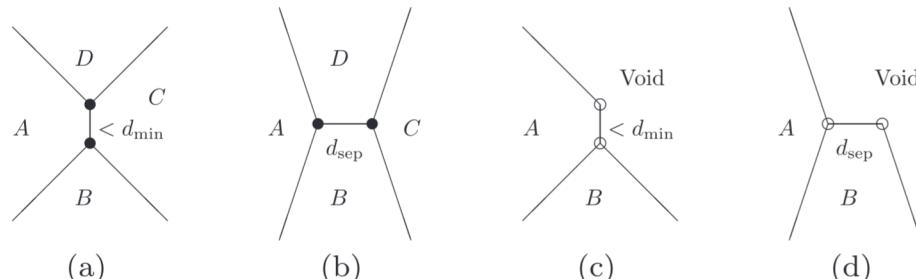


Fig. 4. Illustration of element rearrangement. Connections are broken between elements A and C and a new connection is formed between elements B and D. (a), (b): The case where all elements are present. (c), (d): The same rearrangement happens if any of the elements A–D are missing, for example if the short edge is on the boundary of the mesh. Areas not contained in the mesh are labelled as 'Void'. Vertices on the boundary of the mesh are denoted by open circles while internal vertices are denoted by filled circles.

3.4.2. T2 swap – element removal

If a triangular element becomes smaller than a given threshold, A_{\min} , it is removed from the mesh and the cell associated with the element is removed from the simulation, as shown in Fig. 5(c) and (d). In order for the element to become small enough to be removed, during the course of a simulation, the target area for the element must be small. For example, this would be the case if the cell associated with the element were apoptotic. A general element (with any number of vertices) will become triangular through a series of T1 swaps as it shrinks in size. This is illustrated in Fig. 6. Note that a void in the mesh may also be removed through a T2 swap. The values of the threshold parameters A_{\min} and d_{\min} must be chosen so that this progression can occur.

3.4.3. T3 swap – element intersection

To allow the simulation of monolayers with voids (in applications such as wound healing) and free moving boundaries we need to define what happens when elements intersect. If a vertex intersects the edge of another element then the vertex is incorporated into the element. This process is known as element intersection or a T3 swap. There are four main classes of element intersections, depending on the number of elements containing the intersecting element and the connectivity of the intersecting and intersected elements. The intersections for each of these classes are shown in Fig. 7. New vertices created in the rearrangements are always at least d_{sep} away from existing vertices. If an intersecting vertex is within a distance d_{\min} of another vertex then it is moved to be a distance d_{sep} away from the existing vertex before the rearrangement occurs. If an edge is too short to accommodate the new vertices then the vertices at the ends of the edge are moved apart (extending equally along the edge) to make space for them. The inclusion of this cell rearrangement process allows for the simulation of genuine free and moving boundary problems, in contrast to most published examples of vertex dynamics models.

3.4.4. Node switch – internal element intersection

For some force laws it is possible that an element becomes concave. Moreover this element could become self-intersecting, if this happens then the model breaks down. In order to deal with this situation one could impose that all force laws ensure that elements remain concave. However, this is not practical as all published models may exhibit convex elements under certain conditions, such as hyper-proliferation or compression. Here we propose a new kind of rearrangement known as a node switch which reorders the nodes in an element (and the connected elements) once the element becomes self-intersecting. Fig. 8 illustrates how internal element intersections may be resolved. It is important to note that for a complex intersection, as illustrated in Fig. 8(d), any restructuring operation would result in two elements so are not considered here.

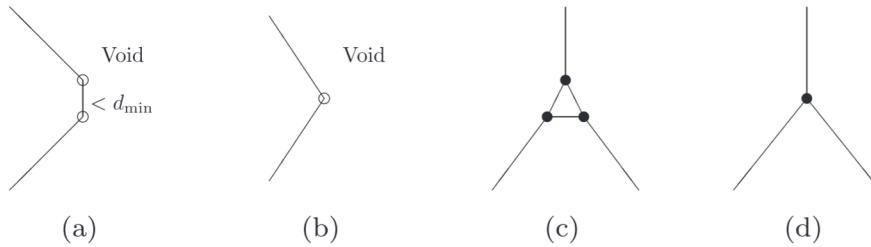


Fig. 5. Illustration of vertex merging and element removal. (a)–(b): Merging vertices. Areas not contained in the mesh are labelled as ‘Void’. (c)–(d) Element removal. A triangular element is removed from the mesh if its area is below a given threshold. The removal is such that each vertex will still be contained in exactly three elements.

3.5. Element division and removal

It is usual to assume that when cells undergo mitosis, they split into two equal areas (Brodland and Veldhuis, 2002). To implement cell division, first an angle of mitosis must be chosen. This may be drawn from a uniform distribution for isotropic cell division, or in the direction of the shortest axis through the cell’s centroid (see Appendix B), or biased in a certain direction if directed proliferation and cell polarity are considered. Two new vertices are placed at the intersection between the dividing line that passes through the cell’s centroid and its cell perimeter, thereby creating two daughter cells from a single parent cell. The element is divided into two new elements using this division axis, as shown in Fig. 9.

In addition to the removal of small triangular elements via a T2 swap (as described in Section 3.4.2), general elements can be removed from the mesh, causing a void to be formed. An example of this is shown in Fig. 9(c) and (d). There are therefore two possible mechanisms for modelling the death of cells in an epithelial layer: either the cell dies instantaneously via necrosis and the element associated with the cell is removed, forming a void; or the cell becomes apoptotic and shrinks until it is destroyed. In the latter case, the target area of the associated element is decreased until the element becomes small and triangular (see Fig. 6) and is removed by a T2 swap, leaving a continuous monolayer.

3.6. Numerical implementation and details of timestepping algorithm

The composition and structure of the monolayer is updated using the following algorithm. In practice we discretize time with time step Δt so that time $t^m = m\Delta t$, and we let $\mathbf{r}_i^m = \mathbf{r}_i(t^m)$, $\mathbf{F}_i^m = \mathbf{F}_i(t^m)$ etc. Then, starting from an initial configuration \mathbf{r}_i^0 , for each time step we:

1. update cell properties from any subcellular model;
2. implement any required mesh restructuring operations;
3. loop over elements/cells to calculate the forces applied at each vertex, \mathbf{F}_i , using chosen force law;
4. update the positions of all vertices simultaneously.

At each time step, any required mesh restructuring operations (T1, T2 or T3 swaps or element divisions) are performed

sequentially, looping over each element and node in order of ascending index. Since each type of rearrangement is a local restructuring operation, the choice of implementation is unlikely to significantly affect the configuration at the next time step, as long as a sufficiently small time step is chosen such that cells are unlikely to require multiple restructuring operations occurring in a single time step. We solve the equations of motion (1) numerically using a simple forward Euler discretization. The displacement of each vertex i within a small time interval Δt is thus given by

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \frac{\Delta t}{\eta_i} \mathbf{F}_i(t). \quad (16)$$

Equation (16) is solved for all vertices in parallel in order to avoid any bias associated with sequential updating of vertex positions. As this numerical method is explicit, we must choose a sufficiently small time step to maintain the stability of the algorithm. If Δt is too large, then the numerical solution may exhibit spurious oscillations or grow without limit and eventually fail. For each example presented in Section 4, we provide the time step used, but we have also run each simulation with a smaller value of Δt to verify stability.

It is possible to use more accurate higher order methods to solve the equations of motion, such as the Adams–Bashforth or the Runge–Kutta methods (Iserles, 2009), a variant of which is used by Honda and co-workers (Nagai and Honda, 2001). An alternative approach is to implement adaptive timestepping, in which Δt is allowed to vary depending on the state of the simulation: one would expect that a large time step would be reasonable close to equilibrium. In either case, the usual benefit of using a more complex updating scheme is the ability to use a larger time step. However, in the case of vertex dynamics simulations far from equilibrium, this benefit is largely negated because the size of time step is constrained by the need to capture the dynamics of cell rearrangement processes and cell division or death. This issue may be compounded by the inclusion of detailed models of subcellular processes, which may be updated at a greater temporal resolution than the cell locations, and hence would need to be interpolated when using higher order methods. We therefore consider a simple forward Euler scheme to be sufficient for the present study.

In order to avoid elements intersecting each other, as illustrated in Fig. 7, the movement of vertices may need to be restricted. At a single time step the maximum distance a vertex may move is $d_{\min}/2$, where d_{\min} is as described in Section 3.4.1. If this threshold is

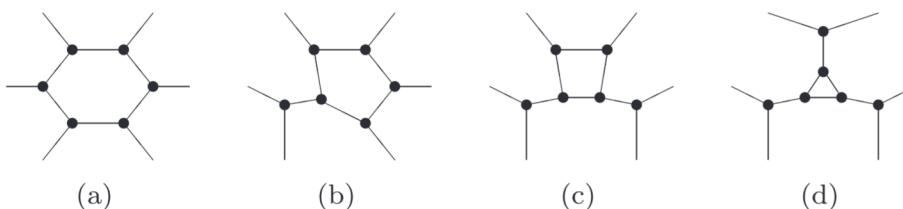


Fig. 6. Progression of a general element to a triangular element through a series of T1 swaps. The resulting triangular element may now be removed through a T2 swap.

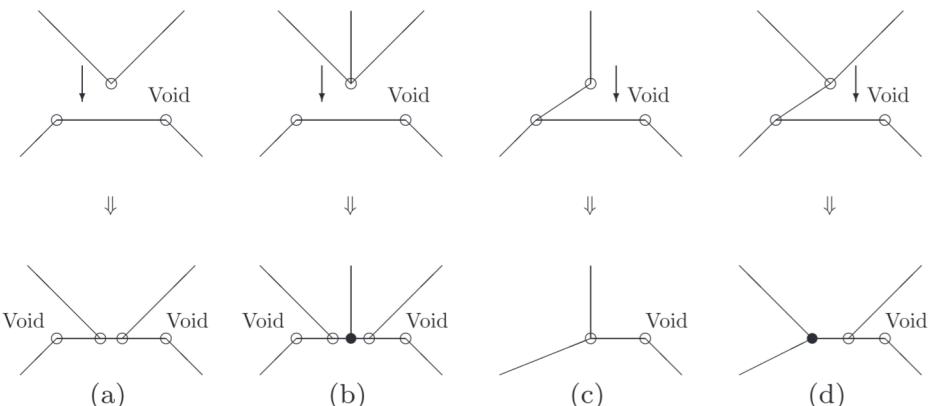


Fig. 7. Illustration of methods for element intersection. Intersecting vertices are shown by open circles. All new vertices created are placed a distance d_{sep} apart, using the intersection of the vertex and the edge as a midpoint where necessary. Areas not contained in the mesh are labelled as 'Void'. Vertices on the boundary of the mesh are denoted by open circles while internal vertices are denoted by filled circles.

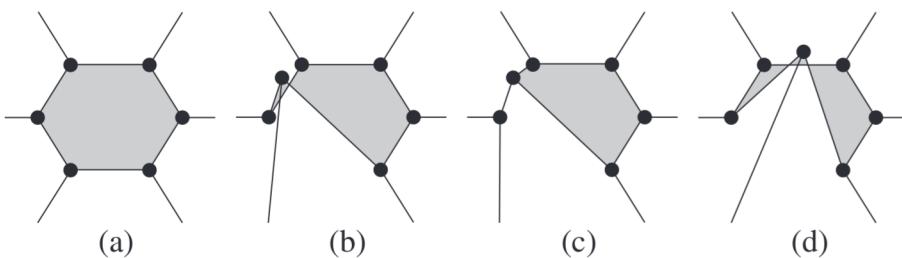


Fig. 8. The possible cases for a self-intersecting element. (a) A simple vertex element. (b) The vertices have moved such that the bottom left vertex is intersecting a neighbouring element; here the vertex has crossed a neighbouring edge, so a node switch may be undertaken, (c), to resolve the issue. In the case where the vertex crosses an edge more than one vertex away from the intersecting node, (d), the intersection cannot be resolved without splitting the element into two new elements.

reached then a smaller time step must be used to ensure accurate solution of the equations of motion (1).

3.6.1. Implementation within the Chaste framework

The models described in Sections 3.1 and 3.2 have been implemented within Chaste. The Chaste framework employs a test-driven approach to software development (Pitt-Francis et al., 2009), and thus cell-based model simulations are implemented as test suites written using the C++ library CxxTest,² which is a light-weight portable testing framework. Chaste is open source software and available for free download from <http://www.cs.ox.ac.uk/chaste/download.html>. Versions 3.0 and earlier used the GNU LGPL 2.1 licence, while versions 3.1 and later are being made available under the 3-clause BSD licence.

The main Chaste code has been written in C++ because it is a fast, object-oriented language, which allows for testing, generality and extensibility. In the next section we include a number of code fragments that illustrate the structure of cell-based simulations in Chaste. Further documentation on the codebase is available on the Chaste developers' wiki at <https://chaste.cs.ox.ac.uk/cgi-bin/trac.cgi/wiki>, sections of which are available for public browsing. All code used to implement vertex dynamics models and to generate results presented in this work is based on release 3.1 of Chaste. This code may be downloaded as the user project VertexComparison2012, which comprises a number of C++ classes and associated test suites, from <http://www.cs.ox.ac.uk/chaste/download.html>.

4. Example simulations

In this section we demonstrate the implementation of vertex dynamics models within our computational framework using a number of case studies, chosen to illustrate the flexibility and generality of this modelling approach. We begin by comparing the behaviour of the two models (Nagai–Honda and Weliky–Oster) in three examples: the growth of a planar epithelial monolayer; a simple wound healing assay; and active cell migration. We then illustrate how further complexity may be implemented in such models, using two further examples: the growth and movement of cells in restricted geometries; and autonomous sorting of heterotypic cell populations.

4.1. Monolayer growth

As a first example, we consider a vertex dynamics model of a monolayer cell culture. This example demonstrates the basic properties of such a model and comprises a population of cells that undergo mitosis in a stochastic manner and interact mechanically via each of the force laws described in Section 3.

Monolayers grown from tumour cell lines are a common *in vitro* cell culture system used to study the early stages of tumour growth. Monolayers are formed when cells are cultured on a flat substrate under nutrient-rich conditions. Once seeded, the cell population initially grows at an exponential rate, as cells experience optimal growth conditions. Growth then slows as cells come into contact with one another and form colonies, which grow more slowly, at a rate dependent on the cell type and culture conditions. Eventually a confluent monolayer approximately one cell high is formed, in which each cell is attached via focal adhesions to the substrate and

² <http://cxxtest.tigris.org>.

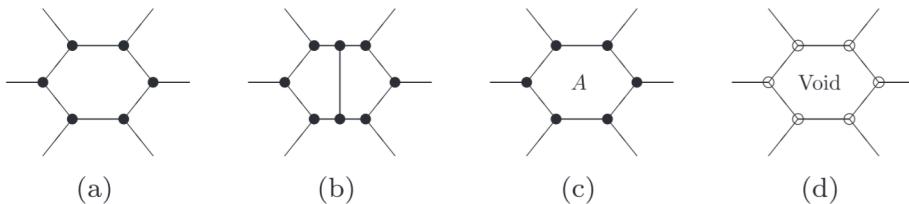


Fig. 9. Example of element division and removal. (a)–(b) Element division. (c)–(d) An element, marked A in (c), is removed from the mesh in (d). Vertices on the boundary of the mesh are denoted by open circles while internal vertices are denoted by filled circles.

to its neighbouring cells. At this stage normal cells stop proliferating as a result of contact inhibition. Many tumours accumulate mutations that allow cells to escape growth control signals, resulting in reduced cell–cell adhesion and hyper-proliferation (Hirohashi, 1998). Despite their relative simplicity, monolayers have several advantages over other *in vitro* cell culture systems, such as ease of culture, low expense and reproducibility.

4.1.1. Previous work

A number of off-lattice cell-based models have been developed to simulate the growth of monolayers. Such models have mostly employed a cell-centre dynamics description. Drasdo and Höhme (2003) develop a cell-centre dynamics model of monolayer growth that takes account of limited compressibility and deformability and cell–cell adhesion. The authors simulate their model using a Metropolis algorithm (Metropolis et al., 1953) to integrate a master equation for the cell configuration. They find initial exponential growth, reducing to power-law growth with the number of cells increasing quadratically for large times. In the model this transition occurs as a result of the excluded volume effect, a form of contact inhibition. In other work, (Galle et al., 2005) use a cell-centre dynamics model to study the effect of normal and malfunctioning growth regulation and control on the spatiotemporal organization of epithelial cell populations. They model the dynamics of each cell by Langevin equations using an ‘overlapping spheres’ approach to define cell neighbours. To model cell–substrate contact-dependent regulation of proliferation and anoikis, the authors assume that a cell cannot proliferate and enters a quiescent state if its cell–substrate contact area is less than some threshold, and that a cell undergoes anchorage-dependent anoikis with a defined rate if its cell–substrate contact area is less than another threshold. In further work, (Galle et al., 2009) use their model to show that intracellular feedback on cell contact formation can be a source of epithelial monolayer stability and tissue homeostasis.

4.1.2. Our implementation

In this example we consider a simple vertex dynamics model of a growing monolayer culture, which couples either the Nagai–Honda or modified Weliky–Oster (with additional membrane deformation force) model of cell mechanics with a simple stochastic model of cell proliferation that takes into account contact inhibition of cell division under high levels of compression. We make the simplifying approximation that a cell’s state of compression, which may be interpreted as the intracellular ‘pressure’, is inversely proportional to its area. We therefore implement contact inhibition of cell proliferation as follows. Each newly divided cell is assigned a random variable T_{cycle} , drawn from the uniform distribution $U[10, 14]$, which dictates the length of its cell cycle under normal conditions. At each time step $t_n = n\Delta t$, after solving the equations of motion numerically and updating vertex positions, we compute the area A_i of each cell i in the monolayer. If $A_i < \phi A_0$, where $\phi \in (0, 1)$, then we increment the value of T_{cycle} for

cell i by Δt . Here the threshold parameter ϕ is a measure of a cell’s sensitivity to its state of compression. The cell must thus spend an additional duration Δ in its cell cycle before undergoing mitosis. In this way, we allow for cell proliferation to be affected by transient periods of high stress, followed by relaxation. The process of mitosis is assumed to last a constant duration T_{mitosis} for all cells.

A free boundary is implemented on the monolayer, with no modification being made in the force on each boundary vertex (easily identified as any vertex belonging to a maximum of two cells). Note that in other studies a specialised ‘boundary force’ has been proposed for boundary vertices in order to ensure a smooth and circular boundary and avoid boundary edges from shrinking (Canela-Xandri et al., 2011; Trichas et al., 2012).

The model parameter values used in these simulations are provided in Tables 1 and 2. Note that in the case of d_{\min} , a value of 0.2 is used by Honda and Eguchi (1980) and Nagai and Honda (2006); we have chosen a smaller value in order to postpone the occurrence of T1 swaps. Varying the value of d_{\min} does not qualitatively alter results. In the case of η , a value of 0.1 is used by Nagai and Honda (2006); we use a different time and space scaling here, which leads to a different value of $\eta = 1.0$. No parameter values are given by Weliky and Oster (1990), so we have chosen order-of-magnitude estimates for ρ , κ and γ to ensure that the forces on cells are of the same order as when using the Nagai–Honda model. For both models, we use typical values for the duration of the cell cycle T_{cycle} for proliferative cells in the colonic crypt (Meineke et al., 2001). As long as this duration is significantly longer than the timescale associated with relaxation to mechanical equilibrium, altering the value of T_{cycle} does not qualitatively alter results. A

Table 1
Table of parameter values used in the simulation shown in Figs. 10 and 11.

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	–
η	Drag coefficient	1.0	Time (Length) $^{-1}$	(Nagai and Honda, 2006)
λ	Deformation energy coefficient	55	Force (Length) $^{-3}$	(Nagai and Honda, 2006)
A_0	Mature cell target area	1	(Length) 2	–
β	Membrane surface energy coefficient	0	Force (Length) $^{-1}$	(Nagai and Honda, 2006)
γ_{cell}	Cell–cell adhesion energy coefficient	5	Force	(Nagai and Honda, 2006)
γ_{boundary}	Cell–boundary adhesion energy coefficient	10	Force	(Nagai and Honda, 2006)
ϕ	Contact inhibition threshold	0.9	Non-dimensional	–
T_{cycle}	Cell cycle duration	$U[10, 14]$	Time	(Meineke et al., 2001)
T_{mitosis}	Mitosis duration	1	Time	(Meineke et al., 2001)
Δt	Time step	0.001	Time	–

Table 2

Table of parameter values used in the simulation shown in Figs. 12 and 13. Note that the value of ϕ is obtained by multiplying the desired threshold parameter (in this case, 0.9) by the equilibrium area of a mature cell (in this case, 0.812261).

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	—
η	Drag coefficient	1.0	Time ($\text{Length})^{-1}$	(Nagai and Honda, 2006)
ρ	Area force coefficient	1.0	Force ($\text{Length})^2$	—
κ	Perimeter force coefficient	1.0	Force ($\text{Length})^{-1}$	—
γ	Angle force coefficient	10.0	Force	—
n_θ	Angle force exponent	3	Non-dimensional	—
ϕ	Contact inhibition threshold	0.731	Non-dimensional	—
T_{cycle}	Cell cycle duration	$U[10, 14]$	Time	(Meineke et al., 2001)
T_{mitosis}	Mitosis duration	1	Time	(Meineke et al., 2001)
Δt	Time step	0.001	Time	—

value of Δt is chosen to ensure that the numerical scheme converges and that a further reduction in the time step does not change simulation results.

The results of example simulations using the Nagai–Honda and modified Weliky–Oster models are shown in Figs. 10–13. Figs. 10 and 12 show snapshots of the monolayers using the Nagai–Honda and modified Weliky–Oster models respectively, each of which were simulated for 200 h. We observe that the Nagai–Honda model results in a monolayer that is close to being radially symmetric. In contrast, the modified Weliky–Oster model gives rise to a boundary that is not smooth but that features ‘kinks’ at characteristic spacings. To further investigate this interesting phenomenon, we performed additional simulations with the modified Weliky–Oster model, in which the starting cell has a different number of vertices (6, 12, 20 or 30). Snapshots of these simulations are shown in Fig. 14, suggesting a sensitivity of the boundary structure on the initial configuration. Based on these preliminary results, we suspect that this behaviour is an artefact of the force model, which includes a resistance of the cell membrane to deformations from equilibrium as described in (15). However, more extensive numerical analysis would be required for a comprehensive investigation of this behaviour. It would also be interesting to compare these results with experimental data on the spatial structure of *in vitro* monolayer boundaries. As previously mentioned, for the chosen parameter values cells in isolation will relax toward a circular equilibrium configuration. While this remains the case for tissues without significant division or movement, in cases of high proliferation cells become compressed and

the delicate balance between forces is unable to prevent elements becoming concave.

Figs. 11 and 13 show various outputs for the two simulations depicted in Figs. 10 and 12 respectively. The first is the number of cells in each monolayer N as a function of time t . Log–log plots of N against t are shown in Fig. 11(a) and Fig. 13(a), while plots of \sqrt{N} against t are shown in Fig. 11(b) and Fig. 13(b). In each case, a dashed line is overlaid, indicating the straight line fit to the data for the last 50 time units. These figures indicate that over time, both simulations approach a power-law behaviour with $N \sim t^2$. However, once the population size is sufficiently large for contact inhibition of cell division to occur, the rate of growth slows to a linear rate. It is clear from Fig. 11(b) and Fig. 13(b) that the rate of growth differs significantly between the two mechanical models. In the Nagai–Honda model, at $t = 200$ there are nearly 25,000 cells, while in the Weliky–Oster case the number is closer to 1500. This discrepancy in timescales arises from the choice of parameter values used in the two models: the typical timescale of relaxation to mechanical equilibrium is longer in the Weliky–Oster case than in the Nagai–Honda case, resulting in contact inhibition and a subsequent limitation in proliferation rate occurring earlier in the simulation.

The second output is the mean radial distribution of cell areas, taken at the end time of each simulation, and shown in Fig. 11(c) and Fig. 13(c). We note that the modified Weliky–Oster model appears to exert a tighter control on cell area than the Nagai–Honda model, resulting in a more uniform radial distribution of mean cell areas. The third output is the mean radial distribution of fraction of cells undergoing mitosis, again taken at the end time of each simulation, and shown in Fig. 11(d) and Fig. 13(d). We see that, for the chosen parameter values, contact inhibition of cell division results in the fraction of mitotic cells being reduced to zero a finite depth into the centre of the monolayer.

In this example, we have demonstrated the basic properties of vertex dynamics models by considering a population of cells that undergo mitosis in a stochastic manner and interact mechanically. As discussed in Section 1.1, our aim is to provide full details of all technical aspects of the implementation of vertex dynamics models. Therefore, to illustrate how this example is implemented within Chaste, we conclude with a brief description of the code used to generate the results visualized in Fig. 10. A more detailed overview of the structure of the code-base may be found in Pitt-Francis et al. (2009). As previously mentioned, the Chaste code is object-oriented, and a number of primary classes are involved in the type of cell-based simulation used in these examples. Many of these are abstract classes, with several concrete implementations, such as those specialized for vertex dynamics models detailed below. Different implementations of the abstract classes can be provided to investigate different model assumptions.

As shown in Fig. 15, a `VertexBasedCellPopulation` object is used to link a collection of `Cell` objects (which do not know their spatial location) with geometric information, provided by a

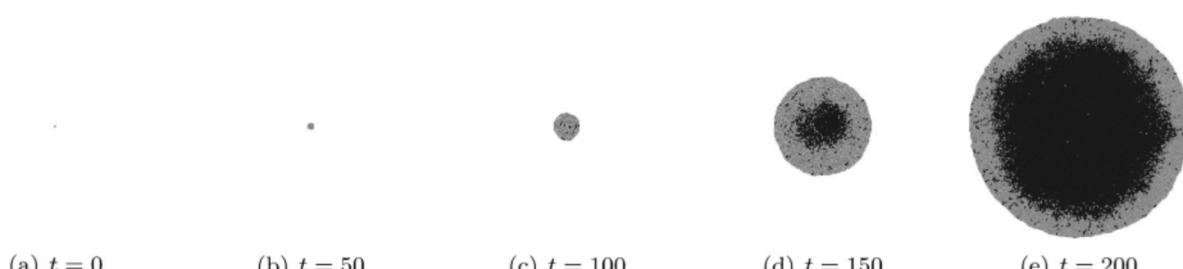


Fig. 10. Snapshots of a simulation of a growing monolayer using the Nagai–Honda vertex dynamics model. Cells coloured dark grey are experiencing contact inhibition of cell division.

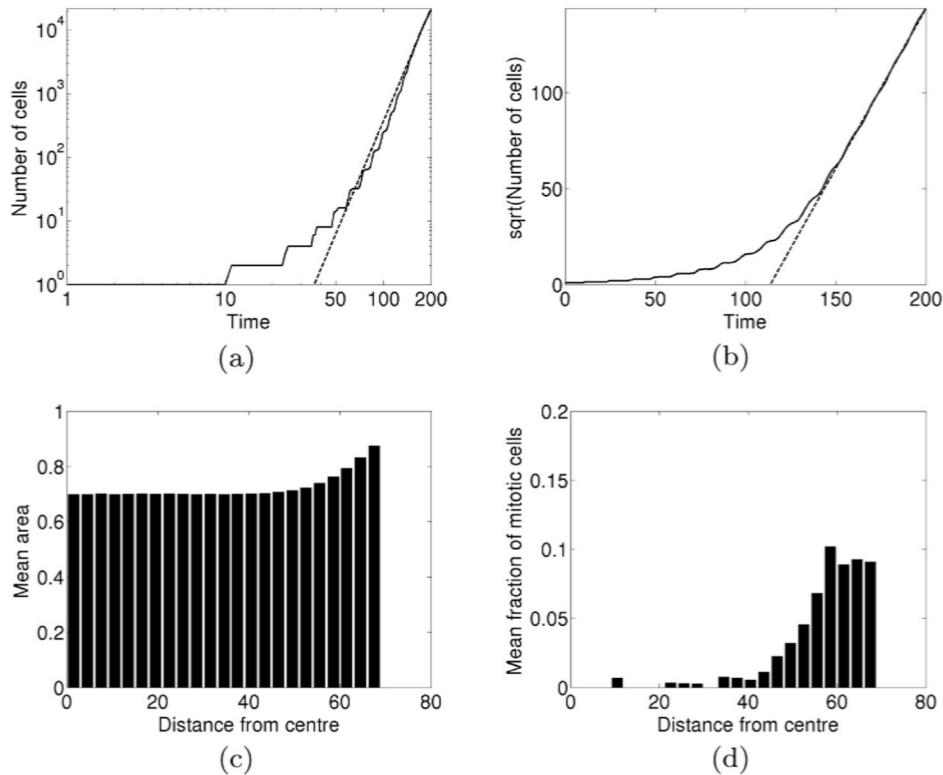


Fig. 11. Growth statistics for the simulation of the growing monolayer shown in Fig. 10. (a) Log–log plot of the number of cells in the monolayer as a function of time (solid line), overlaid with the straight line fit to the data over the last 50 time units (dashed line). (b) The square root of the number of cells in the monolayer as a function of time (solid line), overlaid with the straight line fit to the data over the last 50 time units (dashed line). (c) Mean radial distribution of cell areas in the monolayer at time $t = 200$, computed by averaging over concentric rings about the centre of mass of the monolayer. (d) Mean radial distribution of fractions of cells undergoing mitosis in the monolayer at time $t = 200$, computed by averaging over concentric rings about the centre of mass of the monolayer.

MutableVertexMesh object. A method `ReMesh()`, defined on the MutableVertexMesh class, includes all the code necessary to implement the various mesh restructuring operations defined in Section 3.4. Certain helper classes and methods may be used to generate meshes and cells. In this case we use the methods `HoneycombVertexMeshGenerator` and `SetUpCellsWithStochasticAreaDependentCellCycleModel()`, the latter being defined elsewhere in the test suite. The OffLatticeSimulation class combines the VertexBasedCellPopulation object with a NagaiHondaForce object, which specifies the force balances which determine how cells move as described in Section 3.1 in order to perform a simulation. A further object, VolumeTrackingModifier, is passed to the simulation to allow each Cell object to access its area at each time step (and hence decide whether to cease proliferating). The simulation time is provided by a singleton class, which is automatically set up

at the start of each test. Each Cell has a corresponding cell-cycle model, in this case given by the stochastic area-dependent model defined above. This is specified in `SetUpCellsWithStochasticAreaDependentCellCycleModel()`, which takes the parameter ϕ as its third input argument. The simulation is run by calling the `Solve()` method on the OffLatticeSimulation object.

Note that in order to generate the results shown in Fig. 12, we need only change the type of the force object, from NagaiHondaForce to ModifiedWelikyOsterForce. For the results shown we have used the default value for each model parameter, which is set in each force object's constructor. To use different parameter values, we call an appropriate `Set` method on the force object. For example, to change the value of λ to 100 in the Nagai-Honda force law, we would call `p_force->SetNagaiHondaDeformationEnergyParameter(100)`.

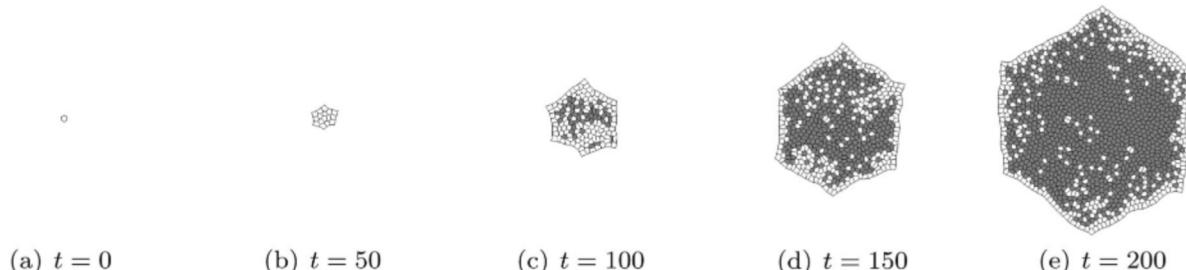


Fig. 12. Snapshots of a simulation of a growing monolayer using the modified Weliky–Oster vertex dynamics model with the additional membrane force. Cells coloured dark grey are experiencing contact inhibition of cell division.

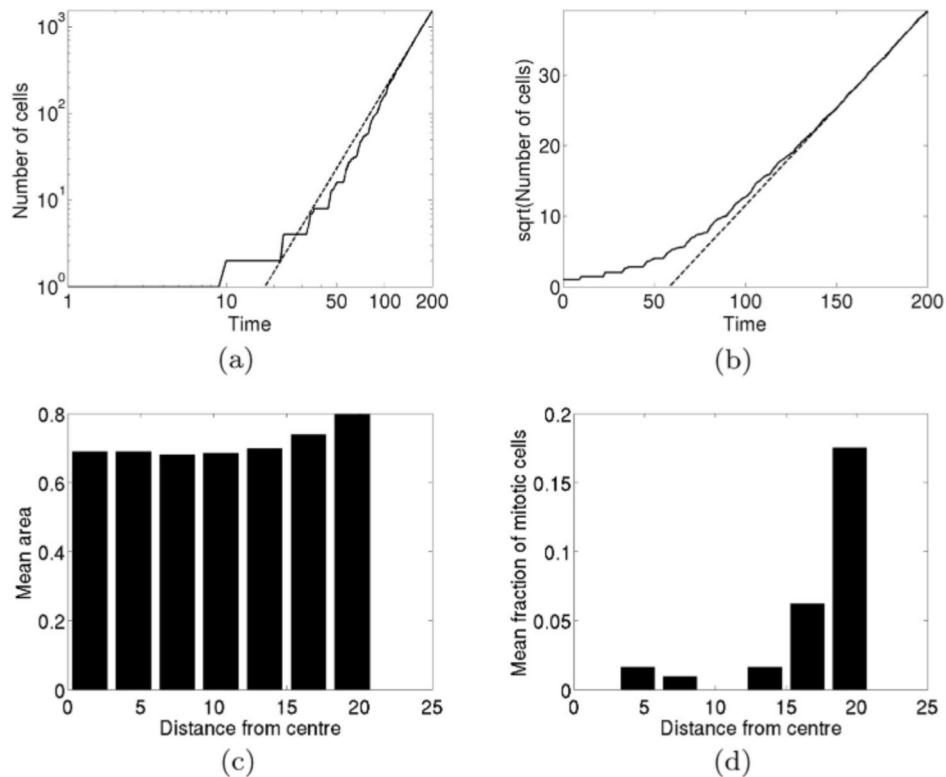


Fig. 13. Growth statistics for the simulation of a growing monolayer shown in Fig. 12. (a) Log–log plot of the number of cells in the monolayer as a function of time (solid line), overlaid with the straight line fit to the data over the last 50 time units (dashed line). (b) The square root of the number of cells in the monolayer as a function of time (solid line), overlaid with the straight line fit to the data over the last 50 time units (dashed line). (c) Mean radial distribution of cell areas in the monolayer at time $t = 200$, computed by averaging over concentric rings about the centre of mass of the monolayer. (d) Mean radial distribution of fractions of cells undergoing mitosis in the monolayer at time $t = 200$, computed by averaging over concentric rings about the centre of mass of the monolayer.

4.2. Wound healing

For our second example, we reproduce a simple model of wound healing developed by Nagai and Honda (2006) to study possible mechanisms for the observed migration of epithelial cells into an area left after some cells are removed from a tissue. This example demonstrates how voids in a tissue may be removed via repeated T2 and T3 swaps, and how fixed boundary conditions may be imposed in a vertex dynamics model.

The wound-healing cell migration assay provides a valuable *in vitro* tool for studying the combined processes of cell migration and proliferation and the role of cell interaction during these processes. This procedure involves the incubation of cells to near confluence and the removal of a patch of cells to create an artificial wound. As cells invade the wound site, the cell population typically exhibits travelling wave-like behaviour. Cells in a wound-healing scrape assay maintain close contact with each other and their motility has been observed to decrease with increasing local density.

4.2.1. Previous work

A large number of mathematical models have been proposed to account for various aspects of the wound healing process. These include continuum models based on Fisher's equation (Maini et al., 2004) and discrete models (Dallon et al., 2000). More recent multiscale models couple discrete descriptions of cells with continuum descriptions of key extracellular matrix components (Cai et al., 2007).

Nagai and Honda (2006) use a vertex dynamics model to study which mechanical processes drive wound healing at the cellular

level in epithelial tissues. In their model, each cell is represented by a collection of vertices, which move to minimize the total free energy of the system. This free energy is taken to comprise three terms: the interfacial energies of cell–cell and cell-wound boundaries, a cellular deformation energy, and a cell-basal-lamina interfacial energy. This model is the first to include a rearrangement process to cope with element intersections associated with adhesion (the T3 swap). As a result new cell boundaries are established when two margins of the wound meet and adhere to each other during wound contraction. After investigating the effect of a variety of different forces acting on cells, the authors conclude that the principal mechanism behind migration of cells into a wound is an adhesive force between marginal cells and the basal lamina. They propose that this accounts for the experimental observation that as a wound heals, it contracts approximately keeping its shape (Radice, 1980).

Note that earlier models, such as (Nagai and Honda, 2006), do not include cell proliferation or the observed effects of contact inhibition of migration in the wound-healing assay. Cai et al. (2007) have demonstrated that this effect is in fact significant and should be included when modelling individual cell behaviour. They present a multi-scale modelling approach to analyse the population-scale behaviour and the individual cell-scale behaviour in a wound-healing assay. The authors apply their model to a real experimental system using mouse 3T3 fibroblast cells.

4.2.2. Our implementation

In this example we consider a simulation of wound healing using the Nagai–Honda and modified Weliky–Oster models, based upon the study by Nagai and Honda (2006). We model a planar

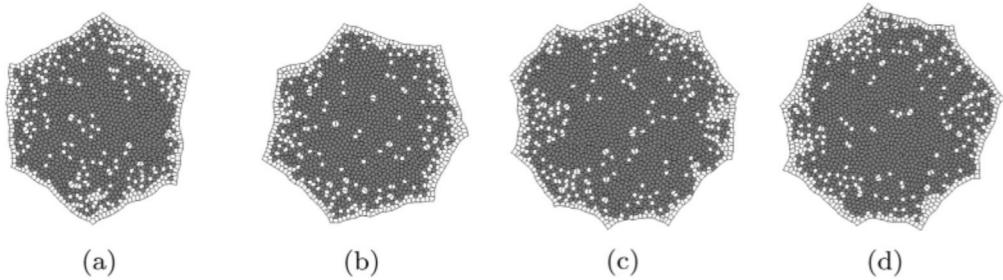


Fig. 14. Snapshots of simulations of growing monolayers using the modified Weliky–Oster vertex dynamics model with the additional membrane force in which the starting cell contains (a) 6, (b) 12, (c) 20 and (d) 30 vertices. Each simulation is visualised at time $t = 200$. Cells coloured dark grey are experiencing contact inhibition of cell division.

domain comprising a regular array of 11×12 cells, each cell initially compressed to 0.8 times its equilibrium area A_0 . The cells are confined to reside in a box of size 10×10 relaxed cell widths (this is shown by a solid box in subsequent results). At the start of each simulation cells are allowed to relax into the box; note that as the Nagai–Honda and modified Weliky–Oster models have different equilibrium structures, the relaxed configuration will be slightly different for each model. Once the cells have reached equilibrium, a hole of width 1.75 is created by removing a central zone of cells from the domain. We consider this to be the start of the simulation. We then track the evolution of the tissue as cells on the boundary of the wound move to fill it as a result of relaxation to mechanical equilibrium. We neglect cell proliferation in the simulation, as we do not expect significant proliferation to occur on the timescale over which wound closure occurs. Confinement of cells within the bounding box is implemented at each time step by checking if any

vertex has moved beyond the boundary of the box, and moving any such vertex back, along the direction it has travelled over that time step, onto the boundary. Such vertices are able to move freely along the boundary if required, and may move away from it as a result of forces experienced from neighbouring vertices. Note that this boundary condition is not purely reflective and thus does not conserve energy over each time step; we suppose that cell edges ‘press’ against the bounding box rather than bounce ballistically off it. Upon removal of the central zone of cells, the boundary of the wound is treated as a free boundary, as described in the monolayer example in Section 4.1. The parameter values used in these simulations are provided in Tables 3 and 4.

Figs. 16 and 17 present a sequence of snapshots of simulations using the Nagai–Honda and modified Weliky–Oster models, respectively. Fig. 18 presents a comparison of superimposed wound margins using the two models. We find that in each model, the

```

void TestNagaiHondaMonolayer()
{
    // Create a simple 2D vertex-based mesh with a single element
    HoneycombVertexMeshGenerator mesh_generator(1, 1);
    MutableVertexMesh<2,2>* p_mesh = mesh_generator.GetMesh();

    // Create a cell, with stochastic area-dependent cell-cycle model with phi=0.9*0.812261
    std::vector<CellPtr> cells;
    SetUpCellsWithStochasticAreaDependentCellCycleModel(cells, 1, 0.9*0.812261);

    // Create a vertex-based cell population
    VertexBasedCellPopulation<2> population(*p_mesh, cells);

    // Set up a cell-based simulation, time step, output directory and end time
    OffLatticeSimulation<2> simulator(population);
    simulator.SetOutputDirectory("TestMonolayer/NagaiHonda");
    simulator.SetDt(0.001);
    simulator.SetEndTime(200);

    // Add volume-tracking modifier
    MAKE_PTR(VolumeTrackingModifier<2>, p_modifier);
    simulator.AddSimulationModifier(p_modifier);

    // Create a force law with default parameter values and pass it to the simulation
    MAKE_PTR(NagaiHondaForce<2>, p_force);
    simulator.AddForce(p_force);

    // Run simulation
    simulator.Solve();
}

```

Fig. 15. Sample Chaste code used to generate the results visualized in Fig. 10.

Table 3

Table of parameter values used in the simulations shown in Fig. 16 and Fig. 18(a).

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	—
η	Drag coefficient	1.0	Time (Length) ⁻¹	(Nagai and Honda, 2006)
λ	Deformation energy coefficient	55	Force (Length) ⁻³	(Nagai and Honda, 2006)
A_0	Mature cell target area	1	(Length) ²	—
β	Membrane surface energy coefficient	0	Force (Length) ⁻¹	(Nagai and Honda, 2006)
γ_{cell}	Cell–cell adhesion energy coefficient	5	Force	(Nagai and Honda, 2006)
γ_{boundary}	Cell-boundary adhesion energy coefficient	10	Force	(Nagai and Honda, 2006)
Δt	Time step	0.005	Time	—

speed at which cells invade the wound is roughly uniform over time. We see that the speed of healing is much slower in the modified Weliky–Oster model than the Nagai–Honda model. This difference in timescales arises from the choice of parameter values in each model, and highlights the difficulties associated with attempts to compare quantitatively different constitutive models when there is no clear timescale. In addition, when using the Nagai–Honda model (Fig. 16) cells remain in contact with the boundary of the domain, due to the initial compression of cells. However, when using a Weliky–Oster model the cells respond more slowly to compression; instead of expanding to fill the void they primarily move to attempt to fill the void, as indicated by the movement of cells away from the corners of the domain observed in Fig. 17(d) and Fig. 18(b). This is due to the model trying to minimise the perimeter of the void. The response of the model to compression is not explicitly imposed as it is in the Nagai–Honda model, hence the slower reaction. While the quantitative details of these two mechanical models differ, their qualitative behaviour does not. Indeed, any vertex dynamics model that caused cells to relax to an equilibrium size following compression, and which incorporated T1 and T3 swaps that facilitate the removal of small holes from a tissue, would result in this type of closure of this type of wound.

In this example we have demonstrated how voids in a tissue may be removed via repeated T1 and T3 swaps, and how fixed boundary conditions may be imposed in a vertex dynamics model. In order to show how this example is implemented in the Chaste framework, we conclude with a brief description of the code used to generate this simulation, with sample code presented in Fig. 19. We note that much of this code is identical to that used in the previous example. As before we associate a collection of Cells with a MutableVertexMesh in a VertexBasedCellPopulation. Note that in this case, we begin by rescaling the mesh by a factor of 0.8 in each direction to simulate the initial compression of the cells. We also use a new helper class, CellsGenerator, with associated method GenerateBasicRandom(), to construct the cells. We specify in this method that each cell should be differentiated so that

no proliferation occurs in the simulation. In order to impose no-flux boundary conditions, we create a VertexPlaneBoundaryCondition object specifying each edge of the domain. For brevity, we only include the condition for the top of the domain in Fig. 19, which specifies this straight edge with a point on the line and outward normal from it. Any VertexPlaneBoundaryCondition objects are then passed to the OffLatticeSimulation object using the method AddCellPopulationBoundaryCondition(). Two further helper methods, SmoothVertexMeshEdges() and CreateHoleInCellPopulation(), are used to smooth the edges of the vertex mesh to avoid overlaps and to create the wound in the domain, respectively. These methods are defined elsewhere in the test suite. The input arguments in the latter method specify, in order, the width of the wound and its dimensions (left, right, bottom and top boundaries).

As in the previous example, we may modify the code shown in Fig. 19 to implement the Weliky–Oster model in a straightforward manner. In order to generate the results shown in Fig. 17, we need only change the type of the force object, from NagaiHondaForce to ModifiedWelikyOsterForce. Different parameter values are used for the force than in the previous example as here we are using the unmodified model WelikyOsterForce object. The specific parameter values are chosen so that a cell with six vertices has a resting area of approximately one.

4.3. Active migration

In the previous two examples, we considered systems in which passive cell motion occurs as a result of cell division and relaxation to mechanical equilibrium. For our third example, we consider a simple model of a tissue in which a cell undergoes active migration, independently of its proliferative status. This example demonstrates how active movement may be implemented in a vertex dynamics model.

While this example is not based on a specific biological system, sustained directional movement of subsets of cells within epithelia is a hallmark of several developmental processes. These include the ordered migration of anterior cells within the visceral endoderm, a simple epithelium that forms the outer layer of the egg-cylinder stage mouse embryo, which is responsible for correctly orientating the anterior–posterior axis (Trichas et al., 2012).

We consider a fixed planar domain of width 10 and length 20 comprising a contiguous population of differentiated cells. The boundary of the domain is fixed such that no vertices may cross it, as described in the wound healing example in Section 4.2. Initially all cells are in mechanical equilibrium. A single cell at the bottom of the domain is selected and ‘dragged’ up the domain as a result of an additional component to the force law, which only affects vertices associated with this cell. For these vertices, the equation of motion (1) takes the modified form

$$\eta_i \frac{d\mathbf{r}_i}{dt} = \mathbf{F}_i + \alpha \hat{\mathbf{e}}_y, \quad (17)$$

where α is a constant, $\hat{\mathbf{e}}_y$ denotes the unit vector parallel to the y axis and the index i runs over all vertices associated with the actively migrating cell.

Fig. 20 shows a sequence of snapshots of this simulation. The model parameter values used in this simulation are provided in Table 5. In the case of the parameter α , an appropriate value is chosen so as to balance the motile force with forces due to the Nagai–Honda model such that active migration occurs. We see that neighbouring cells undergo a series of rearrangement operations to accommodate the active cell. Further measures of this process are

Table 4

Table of parameter values used in the simulations shown in Fig. 17 and Fig. 18(b).

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	—
η	Drag coefficient	1.0	Time (Length) ⁻¹	(Nagai and Honda, 2006)
ρ	Area force coefficient	2.5	Force (Length) ²	—
κ	Perimeter force coefficient	$1/\sqrt{2.5}$	Force (Length) ⁻¹	—
Δt	Time step	0.005	Time	—

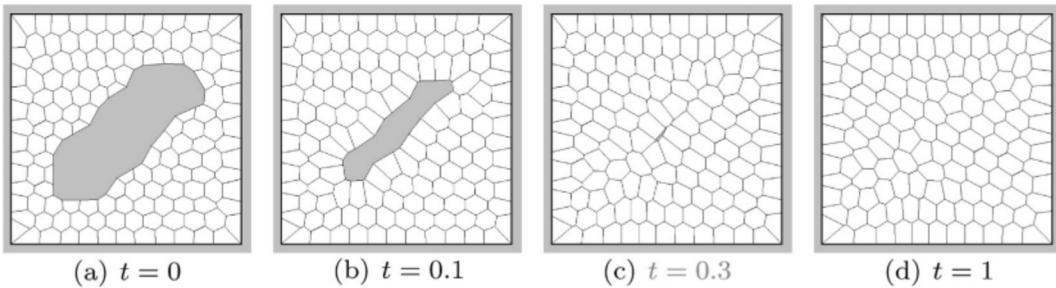


Fig. 16. Demonstration of wound healing using the Nagai-Honda model. The void, representing a wound, is coloured grey. At time $t = 1$ the monolayer has almost reached an equilibrium state.

recorded in Fig. 21. Fig. 21(a) shows the location of T1 swaps during the course of the motion. Fig. 21(b)–(d) show the absolute displacement fields in total, in the y component and in the x component respectively.

Sample code used to generate the results shown in Figs. 20 and 21 is provided in Fig. 22. We note that much of this code is identical to that in the previous examples, and thus we focus on what has changed. As before, we set up an OffLatticeSimulation object using a VertexBasedCellPopulation object, and specify boundary conditions using one or more VertexPlaneBoundaryCondition objects. We employ a new helper method, LabelMotileRegionInCellPopulation(), to label all cells within a radius 0.4472 of the point (1.5, 5.0). The chosen radius is small enough for us to label a single cell using this method. This cell is labelled with a CellLabel object. The modified force law for this cell is implemented through adding another force object, of type MotileCellForce, to the simulation.

If we attempt to use a Weliky–Oster model in this situation (and later situations where cells are also confined) then an incompatible internal intersection, of the form shown in Fig. 8(d), will occur and we are unable to resolve the situation. In this simulation of active migration, the vertices associated with the motile cell are moving through the domain, interacting with other vertices. As a result the surrounding elements may become concave under a Weliky–Oster style model, which eventually leads to such an intersection. The addition of a vertex angle constraint as described in Section 3.3 causes the elements to remain concave for longer, but incompatible intersections still occur eventually. In contrast, due to the volume and surface area constraints present in the Nagai–Honda model, the conditions for incompatible intersections to occur are extreme: the tissue would need to be subjected to massive compression or internal forces much larger than any found in biology. We infer that the Weliky–Oster model is unsuitable for modelling situations of high compression, excess proliferation or restricted geometries. For this reason, in the next example we consider only the Nagai–Honda model.

4.4. Constrained geometries

For our fourth example, we consider a simple model of a tissue in which cells are constrained to lie in some prescribed domain. Many biological systems consist of populations of cells evolving in constrained geometries. Aside from biology, this example also has relevance to Stokes experiments for quasi-two-dimensional foams, which are used to study the effect of the presence of an obstacle on a flowing foam (Dollet and Graner, 2007). The example demonstrates how such boundary conditions may be implemented in a vertex dynamics model.

In this example we consider a planar domain of width 12 that is periodic in the x direction. Cells are prohibited from passing below the line $y = 0$ and are removed from the simulation if they enter the region $y > 12$. Initially a population of 12 cells is seeded at the bottom of the domain, as shown in Fig. 23(a). We define a proliferative domain, $y < y_{\text{prolif}}$, in which cells can proliferate. For the purpose of defining this domain, the position of a cell is taken to be the location of its centroid. Once cells leave this region they no longer proliferate.

As in the monolayer example presented in Section 4.1, each cell in the proliferative region is assigned a random variable T_{cycle} that dictates the length of its cell cycle under normal conditions, and the process of mitosis is assumed to last a constant duration T_{mitosis} for all cells. At each time step $t_n = n\Delta t$, after solving the equations of motion numerically and updating vertex positions, we compute the area A_i of each cell i in the monolayer. If $A_i < \phi A_0$, where $\phi \in (0, 1)$, then we increment the value of T_{cycle} for cell i by Δt . Here the threshold parameter ϕ is a measure of a cell's sensitivity to its state of compression. The cell must thus spend an additional duration Δt in its cell cycle before undergoing mitosis. In this way, we allow for cell proliferation to be affected by transient periods of high stress, followed by relaxation.

We impose an obstruction in the interior of the domain, a solid disc of radius 3 centred at $(x, y) = (6, 5)$. At each time step, once the cell positions have been updated according to the Nagai-Honda

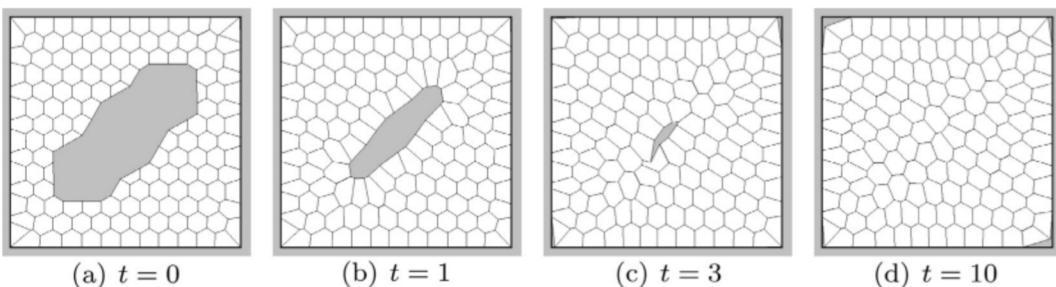


Fig. 17. Demonstration of wound healing using the modified Weliky–Oster model. The void, representing a wound, is coloured grey. At time $t = 10$ the monolayer still has a void.

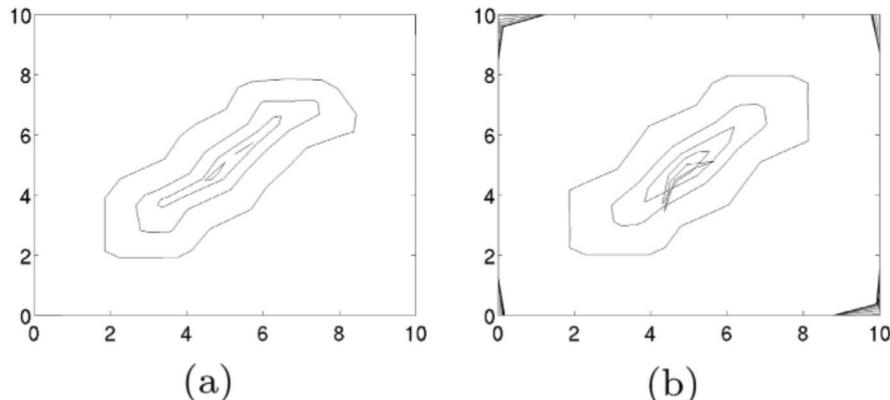


Fig. 18. Superimposed wound margins using the (a) Nagai–Honda and (b) modified Weliky–Oster models, indicating the uniform speed of healing around the wound over time. In each case wound margins are drawn every $\delta t = 0.5$ from the start of the simulation.

```

void TestNagaiHondaMonolayerWithHole()
{
    // Create a regular compressed 2D vertex-based mesh, of size 12 by 13 elements
    HoneycombVertexMeshGenerator mesh_generator(12, 13);
    MutableVertexMesh<2,2>* p_mesh = mesh_generator.GetMesh();
    p_mesh->Scale(0.8, 0.8);

    // Set up a vector of differentiated cells
    std::vector<CellPtr> cells;
    MAKE_PTR(DifferentiatedCellProliferativeType, p_diff_type);
    CellsGenerator<StochasticDurationGenerationBasedCellCycleModel, 2> generator;
    generator.GenerateBasicRandom(cells, p_mesh->GetNumElements(), p_diff_type);

    // Create a vertex-based cell population and set up a cell-based simulation
    VertexBasedCellPopulation<2> population(*p_mesh, cells);
    OffLatticeSimulation<2> simulator(population);
    simulator.SetOutputDirectory("TestWoundHealing/NagaiHonda");
    simulator.SetEndTime(10.0);

    // Create a force law with default parameter values and pass it to the simulation
    MAKE_PTR(NagaiHondaForce<2>, p_force);
    simulator.AddForce(p_force);

    // Create the bottom boundary condition and pass it to the simulation
    c_vector<double, 2> point = zero_vector<double>(2);
    c_vector<double, 2> normal = zero_vector<double>(2);
    normal(1) = -1.0;
    MAKE_PTR_ARGS(VertexPlaneBoundaryCondition<2>, p_bc, (&population, point, normal));
    simulator.AddCellPopulationBoundaryCondition(p_bc);

    // Other boundary conditions are omitted here, but are defined similarly

    // Smooth out edges to get nice box domain, then run simulation
    SmoothVertexMeshEdges(population);
    simulator.Solve();

    // Now remove cells in a given region, and run simulation for a further duration
    CreateHoleInCellPopulation(1.75, 2.0, 8.0, 2.0, 8.0);
    SmoothVertexMeshEdges(population);
    simulator.SetEndTime(20);
    simulator.Solve();
}

```

Fig. 19. Sample Chaste code used to generate the results visualized in Fig. 16.

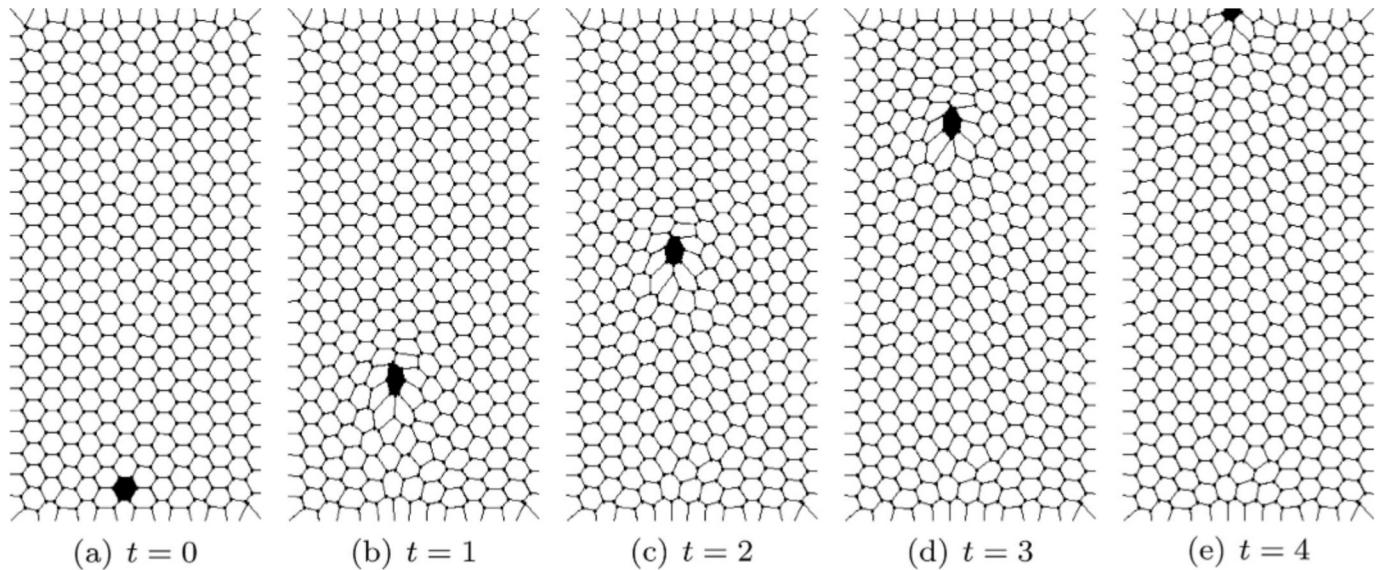


Fig. 20. Demonstration of active migration. Vertices associated with the cell coloured dark grey evolve according to the modified equation of motion (17).

model, all boundary vertices are checked. If a vertex is within the obstruction then it is moved to the boundary of the obstruction. We are interested in how the obstruction affects the evolution of this cell population and how the cells influence the obstruction.

The boundary conditions in this example are implemented as follows. The boundary of the obstacle and the bottom edge of the domain are treated in the same way as the walls of the bounding box in the wound example in Section 4.2. If at a given time step all vertices associated with a cell have height greater than the top edge of the domain, then that cell and any of its vertices not shared with other cells are removed from the simulation; this may be considered analogous to a Robin boundary condition, since the rate at which cells are lost from the top of the domain is proportional to the density of cells there. Finally, periodicity is imposed at the left- and right-hand edges of the domain by correctly associating vertices with cells in a periodic manner in the initial configuration and by using a modified metric that takes periodicity into account when measuring distances and areas.

A simulation with $y_{\text{prolif}} = 1.0$ is shown in Fig. 23, while a simulation with $y_{\text{prolif}} = 12.0$ is shown in Fig. 24. We refer to these simulations as the restricted and total proliferation cases, respectively. The model parameter values used in these simulations are provided in Table 6. We see that cell proliferation at the base of the domain forces all cells upwards. Once cells reach the obstruction they move around it, and continue to move up the domain as a result of mitotic pressure. In the case of restricted proliferation there is a void in the population above the obstruction, as there is nothing to force the cells to reconnect on the other side of the obstruction (Fig. 23(d)). In contrast, Fig. 24(c) and (d) show that in the case of total proliferation the two sides of the cell population join together. This is due to the proliferation of cells in the upper regions of the domain forcing the cells together. Moreover, as there are no attractive forces between the cells and the obstruction, the only force keeping them together is the mitotic pressure from the division of surrounding cells. Hence, as shown in Fig. 24(d), cells may move away from the obstruction at times between stochastic division events.

The force exerted on the obstacle may be calculated by summing the force applied by each vertex on the boundary of the obstacle. This force is compared for the cases of restricted and total proliferation in Fig. 25. In both cases, the vertical force initially increases

once cells come into contact with the obstruction, and there is also a slight reduction as cells begin to move around the obstacle. Once the free space on the sides of the obstacle are occupied, at around $t = 20$, the force settles on a relatively steady state. The mean vertical component of the force for $t \geq 20$ is given by 7.6×10^{-3} for restricted proliferation and by the slightly lower value of 6.5×10^{-3} for total proliferation. The corresponding values for the horizontal component of the force are 6.0450×10^{-5} and -4.8169×10^{-5} respectively.

Due to the fact that cells are proliferating above the obstruction in the case of total proliferation, the force fluctuates more and at some times the vertical component is less than the case of restricted proliferation. To assess fluctuations in the force we may compare the standard deviation of the vertical component of the force for $t \geq 20$, which is given by 1.2×10^{-3} for restricted proliferation and 1.3×10^{-3} for total proliferation. The corresponding values for the horizontal component of the force are given by 3.4178×10^{-4} and 4.0560×10^{-4} respectively.

Sample code used to generate the results shown in Figs. 23–25 is provided in Fig. 26. We now summarise briefly what has changed

Table 5
Table of parameter values used in the simulation shown in Fig. 20.

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	–
η	Drag coefficient	1.0	Time (Length) $^{-1}$	(Nagai and Honda, 2006)
λ	Deformation energy coefficient	55	Force (Length) $^{-3}$	(Nagai and Honda, 2006)
A_0	Mature cell target area	1	(Length) 2	–
β	Membrane surface energy coefficient	0	Force (Length) $^{-1}$	(Nagai and Honda, 2006)
γ_{cell}	Cell–cell adhesion energy coefficient	5	Force	(Nagai and Honda, 2006)
γ_{boundary}	Cell–boundary adhesion energy coefficient	10	Force	(Nagai and Honda, 2006)
α	Magnitude of force on motile cell	0.1	Force	–
Δt	Time step	0.01	Time	–

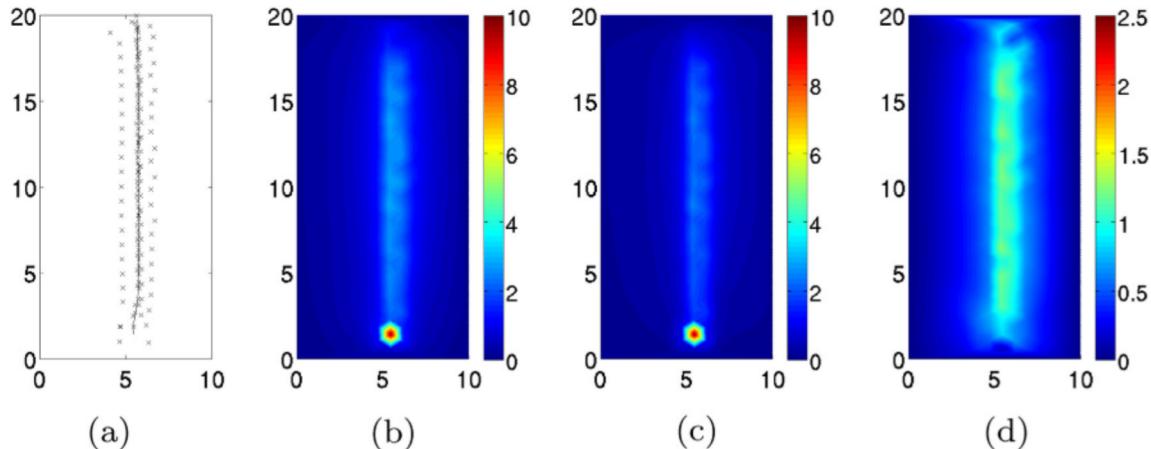


Fig. 21. Measures of the cell rearrangement processes that occur during the simulation of active migration visualized in Fig. 20. (a) The path of the centre of the moving cell (solid line) and the location of the T1 swaps (crosses). (b) Total absolute displacement of all cell centres. (c) Total absolute displacement of all cell centres in the y direction. (d) Total absolute displacement of all cell centres in the x direction. Note that in (b) and (c) the scale for displacement is fixed at 10 (cell lengths), since the migrating cell will move approximately 20 (cell lengths) and bias the plots.

in this code compared to the previous example. In order to impose periodicity at the left and right boundaries of the domain, we use a subclass of `MutableVertexMesh` named `Cylindrical2dVertexMesh`. This has an associated helper class, `CylindricalHoneycombVertexMeshGenerator`, for mesh generation. The functionality for height-dependent cell proliferation is implemented using a new cell-cycle model object, `SimpleWntCellCycleModel`, for each cell. The name of this class reflects its origin as a model for the dependence of cell proliferation in the colonic crypt on the presence of a gradient of Wntmorphogen (van Leeuwen et al., 2009). A helper singleton object, `WntConcentration`, is used to impose height-dependent proliferation. The removal of cells from the top of the domain is handled by a new object called `PlaneBasedCellKiller`, whose construction has a similar form to the `VertexPlaneBoundaryCondition` encountered in previous examples. Lastly, the obstacle is represented by the object `ObstructionBoundaryCondition`, which is treated by the simulation as an ‘interior’ boundary condition.

4.5. Cell sorting

For our fifth example, we consider a simple model of cell sorting. This example demonstrates how heterogeneous mechanical properties can give rise to non-trivial configurations of cell populations. The Weliky–Oster model cannot accommodate variable adhesion between different cell types, as there is no force component which depends on the length of cell edges. We therefore again restrict our attention to the Nagai–Honda model in this example.

If embryonic cells of two or more histological types are placed into contact with each other, they can undergo spontaneous reproducible patterns of rearrangement. This process of cell sorting can generate checkerboard patterns, engulfment of one cell type by another, or other more complex patterns (Steinberg, 1970). To investigate the cause of these rearrangements, a number of models have been developed that describe the mechanics of cell–cell interactions. These models have been used to study the validity of several theories that attempt to explain cell sorting. One such theory is the differential adhesion hypothesis, which states that cells tend to prefer cell–cell contact with certain cell types rather than others due to type-specific differential intercellular adhesion (Foty and Steinberg, 2005). An alternative theory is the differential surface contraction hypothesis, which states that each cell exhibits

surface contraction and cells of different types exert different degrees of surface contraction when in contact with other cell types or any surrounding medium (Harris, 1976).

Vertex dynamics models allow tests of these hypotheses to be performed *in silico*, and comparisons to be made in circumstances in which the hypotheses’ predictions diverge. A much more detailed review of the biophysics underlying cell sorting, and past modelling work done in this area, is provided by Brodland (2004).

In this example we simulate the proliferation of a heterotypic population of cells. As discussed in Section 3.1, the cell–cell adhesion energy in the Nagai–Honda model represents the free energy associated with bonds between each cell k and its neighbours:

$$U_A^k = \sum_{j=0}^{n_k-1} \gamma_{kj} d_{kj}. \quad (18)$$

Here we consider two cell types, termed ‘light’ and ‘dark’, and allow the parameter γ_{kj} to take different values depending on the nature of a cell–cell contact,

$$\gamma_{kj} = \begin{cases} \gamma_{ll} & \text{if cells } k \text{ and } j \text{ are both ‘light’,} \\ \gamma_{dd} & \text{if cells } k \text{ and } j \text{ are both dark,} \\ \gamma_{ld} & \text{if cell } k \text{ is light and cell } j \text{ is dark, or vice versa.} \end{cases} \quad (19)$$

Similarly, we allow for different values of this parameter for interactions between ‘light’ cells and the boundary (γ_{lb}) and ‘dark’ cells and the boundary (γ_{db}). As in previous examples, each cell is assigned a random variable T_{cycle} that dictates the length of its cell cycle under normal conditions, and the process of mitosis is assumed to last a constant duration T_{mitosis} for all cells. The model parameter values used in this simulation are provided in Table 7. Note that we use a larger value for d_{\min} than in previous examples. This is in order to promote the occurrence of the cell rearrangements that are necessary for cell sorting: a larger value of d_{\min} means that for two neighbouring cells moving apart from each other, their shared edge does not have to shrink as much before a T1 swap is triggered, which allows the cells to more easily reach an energetically favourable configuration (i.e. undergo sorting). A free boundary is implemented at the edge of the cell population, as described in the monolayer example in Section 4.1.

The results of this simulation are shown in Fig. 27. As expected, we see that partial cell sorting occurs rapidly, with the formation of

```

void TestNagaiHondaMonolayerWithActiveMigration()
{
    // Create a regular compressed 2D vertex-based mesh, of size 24 by 14 elements
    HoneycombVertexMeshGenerator mesh_generator(24, 14);
    MutableVertexMesh<2,2>* p_mesh = mesh_generator.GetMesh();
    p_mesh->Scale(0.8, 0.8);

    // Set up a vector of differentiated cells
    std::vector<CellPtr> cells;
    MAKE_PTR(DifferentiatedCellProliferativeType, p_diff_type);
    CellsGenerator<StochasticDurationGenerationBasedCellCycleModel, 2> generator;
    generator.GenerateBasicRandom(cells, p_mesh->GetNumElements(), p_diff_type);

    // Create pointer to a labelled cell state
    boost::shared_ptr<AbstractCellProperty> p_state(CellPropertyRegistry::Instance()->Get<CellLabel>());

    // Create a vertex-based cell population
    VertexBasedCellPopulation<2> population(*p_mesh, cells);

    // Now label cells in a given region as motile using a helper method
    LabelMotileRegionInCellPopulation(0.4472, 1.5, 5.0);

    // Set up a cell-based simulation
    OffLatticeSimulation<2> simulator(population);
    simulator.SetOutputDirectory("TestVertexCellMoving");
    simulator.SetEndTime(30);

    // Create a force law with default parameter values and pass it to the simulation
    MAKE_PTR(NagaiHondaForce<2>, p_force);
    simulator.AddForce(p_force);

    // Boundary conditions are omitted here, but are defined as in the previous example

    // Smooth out edges to get nice box domain, then run simulation
    SmoothVertexMeshEdges(population);
    simulator.Solve();

    // Now flag a cell as motile by adding a special force object to the simulation
    MAKE_PTR(MotileCellForce<2>, p_motile_force);
    simulator.AddForce(p_motile_force);

    // Reset end time for simulation and run for a further duration
    simulator.SetEndTime(40);
    simulator.Solve();
}

```

Fig. 22. Sample Chaste code used to generate the results visualized in Fig. 20.

islands of dark cells surrounded by light cells. We note that the final configuration shown in Fig. 27(c) remains stable as the simulation proceeds. This is in contrast to the behaviour of such a simulation if the cellular Potts model were used, since our simulation does not

include the random motion that is required in the latter approach. This raises the question of to what extent we would expect 'complete' cell sorting to occur in practice, for a given biological system in which there is significant cell proliferation. It would appear that

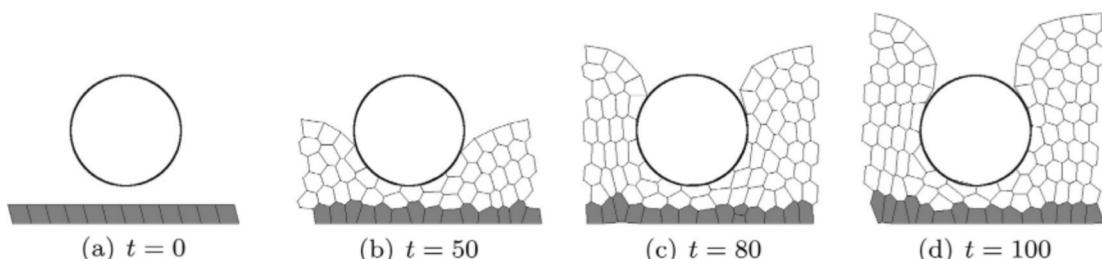


Fig. 23. Movement around an obstacle in the case where proliferation occurs only in the bottom 10% of the domain. Proliferating cells are coloured dark grey, non-proliferating cells are coloured white and voids are coloured light grey.

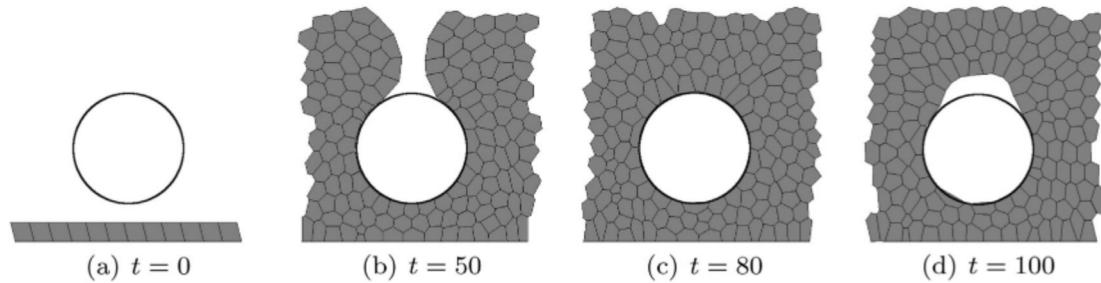


Fig. 24. Movement around an obstacle in the case where all cells in the domain are proliferating. Proliferating cells are coloured dark grey and voids are coloured light grey.

the degree of random fluctuations associated with cell movement dictates whether a partially sorted configuration remains stable. In order to pursue this further, we present in Fig. 28 the results of a cell sorting simulation in which, in addition to the differential adhesion described above, a small random force is applied to each vertex i at each time step. This additional force takes the form

$$\mathbf{F}_i^{\text{random}} = \eta_i \sqrt{\frac{2D}{\Delta t}} \xi_i, \quad (20)$$

where D denotes a macroscopic diffusion coefficient and each component of the vector ξ_i is an independent standard normal random number. Note that the dependence of $\mathbf{F}_i^{\text{random}}$ on the time step Δt is required to ensure that when the equation of motion for each vertex is solved numerically, the rate of diffusion is independent of the size of time step. The value of D used in the simulation shown in Fig. 28 is listed in Table 7.

By comparing Figs. 27 and 28 we find that the addition of stochasticity to the force applied to each vertex accelerates cell sorting. While both simulations begin from the same configuration of labelled and non-labelled cells, in the case of added stochasticity there are fewer clusters of labelled cells by time $t = 70$. This is because some of the smaller clusters merged between times $t = 0$ and $t = 20$. While we would expect further sorting to occur at higher noise levels, note that for large values of D vertices may move more than the maximum allowable distance $d_{\min}/2$ over a single time step, and hence extremely small time steps are required to avoid element overlaps and ensure accurate solution of the equations of motion.

In this example, we have demonstrated how heterogeneous mechanical properties can give rise to non-trivial configurations of cell populations. We conclude by again presenting, in Fig. 29, sample code used to generate the simulation, which is shown in

Table 6
Table of parameter values used in the simulations shown in Figs. 23 and 24.

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	—
η	Drag coefficient	1.0	Time (Length) $^{-1}$	(Nagai and Honda, 2006)
λ	Deformation energy coefficient	55	Force (Length) $^{-3}$	(Nagai and Honda, 2006)
A_0	Mature cell target area	1	(Length) 2	—
β	Membrane surface energy coefficient	0	Force (Length) $^{-1}$	(Nagai and Honda, 2006)
γ_{cell}	Cell-cell adhesion energy coefficient	5	Force	(Nagai and Honda, 2006)
γ_{boundary}	Cell-boundary adhesion energy coefficient	10	Force	(Nagai and Honda, 2006)
γ_{prolif}	Maximum proliferation height	1 or 12	Length	—
T_{cycle}	Cell cycle duration	U [10, 14]	Time	(Meineke et al., 2001)
T_{mitosis}	Mitosis duration	1	Time	(Meineke et al., 2001)
Δt	Time step	0.001	Time	—

Fig. 27. This code contains three notable differences compared to the code for the first two examples. First, to randomly label cells to be of a different type, we re-use the `CellLabel` object. We then call a helper method, `RandomlyLabelCells()`, which uses an `Iterator` structure to loop over the `VertexBasedCellPopulation` object and assigns each cell a label with probability 0.5. Second, we make use of the force object `NagaiHondaDifferentialAdhesionForce`, which allows for cell-type dependent adhesion. Third, we include an additional force object

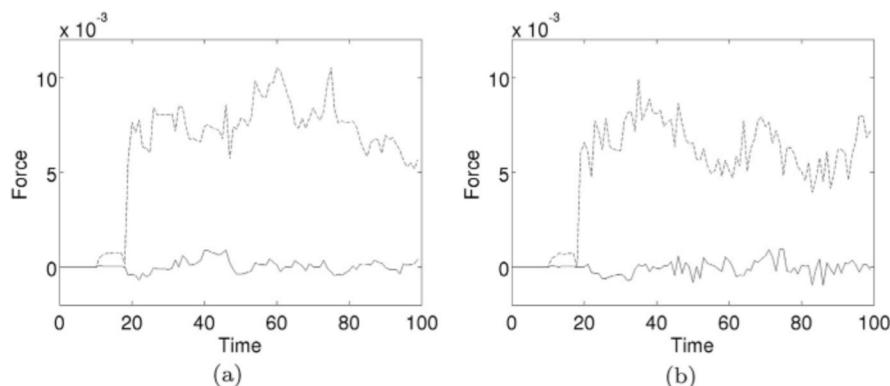


Fig. 25. Force exerted on the obstruction in the case where (a) proliferation occurs only in the bottom 10% of the domain and (b) all cells are proliferating. The solid line shows the force in the x direction and the dashed line shows the force in the y direction. All values are non-dimensionalised.

```

void TestNagaiHondaMonolayerWithObstruction()
{
    // Create a regular compressed 2D vertex-based mesh, of size 12 by 1 elements
    CylindricalHoneycombVertexMeshGenerator generator(12, 1, true, 0.6);

    // Impose periodicity at the left- and right-hand boundaries of the domain
    Cylindrical2dVertexMesh* p_mesh = generator.GetCylindricalMesh();
    p_mesh->SetCellRearrangementThreshold(0.01);

    // Set up a vector of cells with given cell-cycle model
    std::vector<CellPtr> cells;
    MAKE_PTR(TransitCellProliferativeType, p_transit_type);
    CellsGenerator<SimpleWntUniformDistCellCycleModel, 2> cells_generator;
    cells_generator.GenerateBasicRandom(cells, p_mesh->GetNumElements(), p_transit_type);

    // Create a vertex-based cell population and set up a cell-based simulation
    VertexBasedCellPopulation<2> population(*p_mesh, cells);
    OffLatticeSimulation<2> simulator(population);
    simulator.SetOutputDirectory("TestVertexObstruction");
    simulator.SetEndTime(100.0);

    // Create a force law and pass it to the simulation
    MAKE_PTR(NagaiHondaForce<2>, p_force);
    simulator.AddForce(p_force);

    // Create cell killer for y = 12 and pass to the simulation
    c_vector<double, 2> point = zero_vector<double>(2);
    point(1) = 12.0;
    c_vector<double, 2> normal = zero_vector<double>(2);
    normal(1) = 1.0;
    MAKE_PTR_ARGS(PlaneBasedCellKiller<2>, p_killer, (&population, point, normal));
    simulator.AddCellKiller(p_killer);

    // Create a boundary condition representing an obstruction in within the domain
    MAKE_PTR_ARGS(ObstructionBoundaryCondition<2>, p_bc, (&population));
    simulator.AddCellPopulationBoundaryCondition(p_bc);

    // Other boundary conditions are omitted here, but are as in previous examples

    // Create an instance of a Wnt concentration to impose height-dependent proliferation
    WntConcentration<2>::Instance()->SetType(LINEAR);
    WntConcentration<2>::Instance()->SetCellPopulation(population);
    WntConcentration<2>::Instance()->SetCryptLength(3);

    // Run simulation
    simulator.Solve();

    // Tidy up
    WntConcentration<2>::Instance()->Destroy();
}

```

Fig. 26. Sample Chaste code used to generate the results visualized in Fig. 23.

`VertexDiffusionForce`, which implements the random motion of vertices described in (20).

5. Discussion

The spreading of epithelial cell sheets and self-assembly of cells into tissues play a central role in morphogenesis, wound healing, tumour growth, tissue engineering and other biological processes. These phenomena occur as a result of cell division, differentiation, programmed cell death, adhesion and migration, and involve multiple processes acting on both a cellular and subcellular scale.

Mathematical and computational models offer a useful means by which to investigate and test hypotheses about these processes, and have played a key role in the study of cell–cell interactions.

In this study we have developed a robust, open-source computational framework for the simulation of vertex dynamics models, within which further complexity may be incorporated in a straightforward manner. This work enables the comparison of different constitutive assumptions, and indeed modelling paradigms, for the theoretical study of epithelial cell populations. Establishing how to perform such comparisons systematically remains an open question. We have included full details of the

Table 7

Table of parameter values used in the simulations shown in Figs. 27 and 28.

Parameter	Description	Value	Dimensions	Reference
d_{\min}	Cell rearrangement threshold	0.1	Length	—
η	Drag coefficient	1.0	Time (Length) ⁻¹	(Nagai and Honda, 2006)
λ	Deformation energy coefficient	55	Force (Length) ⁻³	(Nagai and Honda, 2006)
A_0	Mature cell target area	1	(Length) ²	—
β	Membrane surface energy coefficient	0	Force (Length) ⁻¹	(Nagai and Honda, 2006)
γ_{ll}	Light/light cell adhesion energy coefficient	1	Force	(Brodland, 2004)
γ_{dd}	Dark/dark-cell adhesion energy coefficient	3	Force	(Brodland, 2004)
γ_{ld}	Light/dark cell adhesion energy coefficient	6	Force	(Brodland, 2004)
γ_{db}	Dark cell-boundary adhesion energy coefficient	40	Force	(Brodland, 2004)
γ_{lb}	Light cell-boundary adhesion energy coefficient	12	Force	(Brodland, 2004)
T_{cycle}	Cell cycle duration	U [10, 14]	Time	(Meineke et al., 2001)
$T_{mitosis}$	Mitosis duration	1	Time	(Meineke et al., 2001)
D	Diffusion coefficient	0 or 0.02	(Length) ² Time ⁻¹	—
Δt	Time step	0.001	Time	—

technical implementation of the algorithms, filling in some of the gaps left by previous authors whose primary focus was on the scientific findings. Our intention is that readers can either reimplement the methods described themselves relatively straightforwardly using the detailed descriptions of the algorithms, or they can use and fully understand our implementation, which is freely available within the open source Chaste framework.

From a more scientific perspective, an important question regarding the use of off-lattice cell-based models is how robust they are to changes in model assumptions. In this study, we have reviewed the ‘vertex dynamics’ approach to modelling the evolution of a collection of epithelial cells. We discussed the variety of biological applications for which this approach has been employed, the different constitutive assumptions regarding force generation, and issues associated with implementation of cell rearrangement processes within vertex dynamics models. A number of case studies were used to illustrate the flexibility and generality of this modelling approach.

We compared the behaviour of two vertex dynamics models, a model developed by Honda and co-workers and a modified form of a model developed by Weliky and co-workers, when used to simulate three examples: the growth of a planar epithelial monolayer; a simple wound healing assay; and active cell migration. In the case of the Weliky–Oster model, we first introduced two modifications that allow the model to be applied to the types of example considered in Section 4, and thus enable us to more fully compare the two approaches. The first modification resulted in an equilibrium cell area that is independent of the number of vertices associated with the cell. The second modification involved the introduction of a force component representing membrane stiffness, which helps to prevent elements of the mesh associated with the cell population becoming non-convex or self-intersecting. We also illustrated how further complexity may be implemented in such models, using two further examples: the growth and movement of cells in restricted geometries; and autonomous sorting of heterotypic cell populations.

Based on the five biological examples considered in this study, we may evaluate the strengths and limitations of the two vertex dynamics model approaches. The results of our growing monolayer and wound healing examples highlighted the difficulty in equating timescales of mechanical relaxation between the two models, due in part to the difference in whether limited cell compressibility is modelled explicitly. We also found qualitative differences in the two models in the case where significant cell proliferation occurs, such as on the boundary of a growing monolayer. Conceptually, we may consider an explicit force-based approach as favoured by Weliky and co-workers to be more appropriate for dynamics far from equilibrium, while for a system relaxing toward mechanical equilibrium the approach of Honda and co-workers may seem more intuitive. Of course, in practice each approach culminates in the solution of equations of motion, and the two approaches differ only in the directions and relative magnitudes of the forces used to describe the hydrostatic pressure within each cell and tension at cell–cell interfaces. Our active migration and constrained geometry examples indicated that the Weliky–Oster model is more likely to exhibit concave cells and unresolvable cell intersections in cases of significant compression, proliferation or geometric constraints. While such findings are preliminary, and a more systematic survey remains to be undertaken, nevertheless the present work highlights the need for assessing sensitivity of model predictions to constitutive assumptions.

In this study we chose to focus on two types of vertex dynamics model, one by Honda and co-workers and a modified form of a model developed by Weliky and co-workers, as these reflect the majority of published models. Our choice of models and simple biological examples allowed us to illustrate the key constitutive assumptions and provide a clear, comprehensive description of the

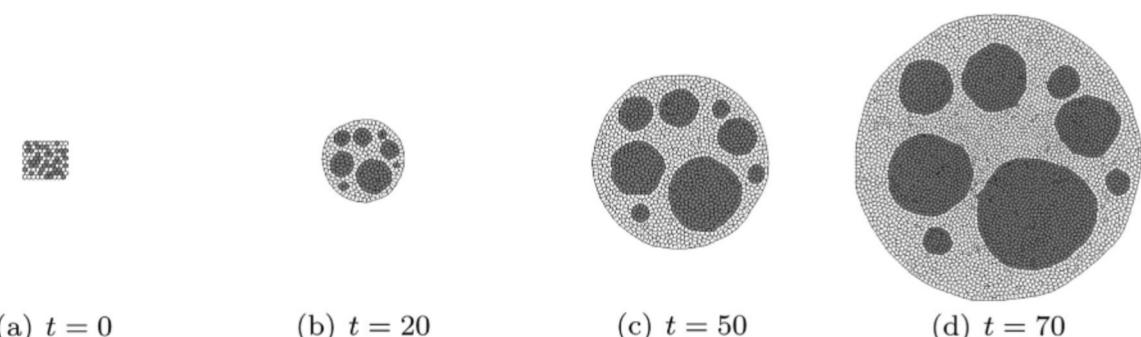


Fig. 27. Snapshots of a simulation of cell sorting using the Nagai-Honda vertex dynamics model with a population of light and dark cells.

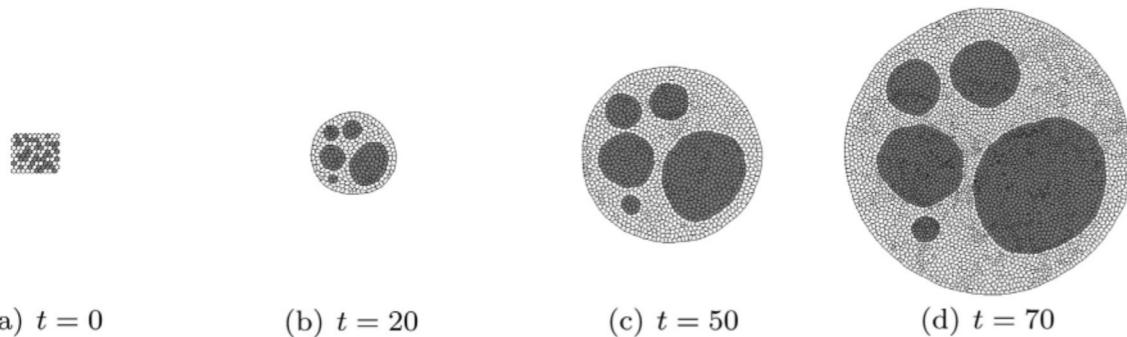


Fig. 28. Snapshots of a simulation of cell sorting using the Nagai–Honda vertex dynamics model with a population of light and dark cells including random motion of cell vertices.

implementation of such models. The limitations of the model of Weliky and co-workers in the latter two examples, and the need to provide modifications to the model in these cases, highlights the need to carefully consider the applicability of a given model when addressing a particular application.

By implementing these models within a consistent computational framework, we have ensured that the same method of simulation is used in each case, so that any observed differences in behaviour may be traced to differing constitutive model assumptions. A

complementary question relates to how robust off-lattice cell-based models are to changes in model assumptions and method of simulation. Drasdo et al. (2007) address this question by comparing the use of the Metropolis method of simulation with numerical integration of Langevin equations, and show that there is reasonable agreement between these two methods. They also investigate the effect of using different constitutive laws for cells' mechanical behaviour, and find that the qualitative results of simulations are very robust to such differences. Galle et al. (2006) review the use of off-lattice cell-based

```

void TestVertexMonolayerCellSorting()
{
    // Create a regular 2D vertex-based mesh, of size 10 by 10 elements
    HoneycombVertexMeshGenerator mesh_generator(10, 10);
    MutableVertexMesh<2,2>* p_mesh = mesh_generator.GetMesh();
    p_mesh->SetCellRearrangementThreshold(0.05);

    // Set up a vector of cells with given cell-cycle model
    std::vector<CellPtr> cells;
    MAKE_PTR(TransitCellProliferativeType, p_transit_type);
    CellsGenerator<StochasticDurationCellCycleModel, 2> cells_generator;
    cells_generator.GenerateBasicRandom(cells,p_mesh->GetNumElements(), p_transit_type);

    // Randomly label some cells
    boost::shared_ptr<AbstractCellProperty> p_state(CellPropertyRegistry::Instance()->Get<CellLabel>());
    RandomlyLabelCells(cells, p_state, 0.5);

    // Create cell population
    VertexBasedCellPopulation<2> population(*p_mesh, cells);

    // Set up cell-based simulation
    OffLatticeSimulation<2> simulator(population);
    simulator.SetOutputDirectory("TestVertexCellSorting");
    simulator.SetEndTime(50.0);

    // Set up force law and pass it to the simulation
    MAKE_PTR(NagaiHondaDifferentialAdhesionForce<2>, p_force);
    simulator.AddForce(p_force);

    // Add some random motion to vertices
    MAKE_PTR_ARGS(VertexDiffusionForce<2>, p_random_force, (0.01));
    simulator.AddForce(p_random_force);

    // Run simulation
    simulator.Solve();
}

```

Fig. 29. Sample Chaste code used to generate the results visualized in Fig. 27.

models in a similar vein to Drasdo et al. (2007), focussing on robustness to changes in model assumptions.

We have restricted our attention in the present study to vertex dynamics models of planar two-dimensional cell populations. While the two-dimensional approximation is reasonable in many cases, in some instances it is necessary to consider the full three-dimensional structure of a population, for example when modelling the growth of cell populations in fluid media. While a few three-dimensional vertex dynamics models have been developed (Honda et al., 2004; Viens and Brodland, 2007), the additional computational cost of moving to three dimensions has so far restricted simulations of such models to small numbers of cells. The main computational burden in the simulations presented in this study lies in recognising and performing cell rearrangement operations. When considering rearrangements of polyhedral cells in three dimensions, edges, faces and vertices of the polyhedra may be all interchanged to maintain non-intersecting elements. This leads to a much larger set of elementary operations to allow for than those detailed in Section 3.4 for the two-dimensional case. Moreover, for a given cell population size, the number of equations of motion to solve at each time step increases dramatically as we move to three dimensions. This is because in three dimensions the equilibrium shape of a cell is the orthic tetrakaidecahedral tessellation (Honda et al., 2004), which comprises 24 vertices, compared to six vertices per cell at equilibrium in two dimensions. The efficient simulation of three-dimensional vertex dynamics models, as well as two-dimensional approximations on curved, non-planar surfaces, remains an avenue of future work.

An important question associated with the use of vertex dynamics models is to what extent they may be parameterised or validated against experimental data. Although to date most vertex dynamics models have been used to gain qualitative insights into the behaviour of epithelial dynamics, some recent studies have attempted to place such models on a quantitative footing, most notably through the use of infra-red laser ablation of individual junctions in order to infer the tension from the initial speed of vertex recoil (Bonnet et al., 2012; Landsberg et al., 2009; Ma et al., 2009). A variety of inverse problem frameworks have also been developed for the estimation of model parameters based on the output of less invasive experimental measurements (Brodland et al., 2010; Chiou et al., 2012; Ishihara and Sugimura, 2012). In cases where significant levels of cell intercalation or proliferation occur, the ability to gather sufficient experimental data for models to be properly calibrated remains hindered in part by the lack of established methods for semi-automated tracking of cell division and rearrangement events in time-lapse data.

One of the strengths of a discrete modelling approach, such as the vertex dynamics approach discussed in this paper, is that it allows the simulation of observed tissue-level behaviours, such as cell heterogeneity and the occurrence of mutated cells. However, two limitations are associated with this modelling approach. First, it can be difficult to gain insight into a biological process using complex stochastic models, since many simulations are required to determine mean behaviour. Second, as more biophysical detail is incorporated into a model of cell dynamics, the size of the parameter space to be explored increases combinatorially. One possible method to circumvent these limitations is to employ some form of model reduction. For example, Murray et al. (2009) have shown that the spatial dynamics of cells connected via overdamped linear springs, as in Meineke et al. (2001) and van Leeuwen et al. (2009), can be well represented by a nonlinear diffusion equation for cell number density. In other work, (Fozard et al., 2010) and (Alber et al., 2007) have derived coarse-grained continuum models to describe the evolution of epithelial monolayers of cells starting from vertex and cellular Potts descriptions in certain simplified

cases, respectively. A number of computational studies have also sought to characterise the mechanical behaviour of various types of individual cell-based models under loading, unloading and shearing, both with and without the complicating presence of cell rearrangements (Davit et al., 2013; Pathmanathan et al., 2009). Such techniques could help refine the search for suitable parameter values when attempting to fit such discrete models to data, either qualitatively or quantitatively.

Acknowledgements

The authors gratefully acknowledge funding through the OCISB project (BB/D020190/1). AGF is supported by EPRSC (EP/I017909/1) and Microsoft Research, Cambridge. JMO is supported by the Life Sciences Interface and Systems Biology Doctoral Training Centres (EP/E501605/1 and EP/G50029/1 respectively) and Microsoft Research, Cambridge.

Appendix A. Area and length computations

Note that the movement of vertex i only affects the free energy of the cells containing it. The gradient of free energy can be written explicitly in terms of the cells surrounding the vertex,

$$\mathbf{F}_i = - \sum_{l \in \mathcal{N}_i} \left(2\lambda(A_l - A_{0_l}) \nabla_i A_l + 2\beta(C_l - C_{0_l}) \nabla_i C_l + \sum_{j=0}^{n_l-1} \gamma_{l,j} \nabla_i d_{l,j} \right).$$

Noting that the circumference C_k can be written as

$$C_k = \sum_{j=0}^{n_k-1} d_{k,j},$$

we have

$$\mathbf{F}_i = - \sum_{l \in \mathcal{N}_i} (2\lambda(A_l - A_{0_l}) \nabla_i A_l + 2\beta(C_l - C_{0_l}) (\nabla_i d_{l,l_{i-1}} + \nabla_i d_{l,l_i}) + \gamma_{l,l_{i-1}} \nabla_i d_{l,l_{i-1}} + \gamma_{l,l_i} \nabla_i d_{l,l_i})$$

where l represents the l th cell containing vertex i , ordered counter-clockwise, and l_i is the local index of node i in element l . When a vertex i is moved, it will affect two terms in the above summation, the $(i-1)$ th and i th terms. Using the area expression (4) for a simple polygon, we can compute the gradient of A_l as

$$\nabla_i A_l = \frac{1}{2} \left(\begin{array}{c} y_{l+1}^l - y_{l-1}^l \\ x_{l-1}^l - x_{l+1}^l \end{array} \right), \quad (A.1)$$

where (x_{l-1}^l, y_{l-1}^l) are the coordinates of the vertex one position further clockwise than the l_i th vertex of cell l and (x_{l+1}^l, y_{l+1}^l) are the coordinates of the vertex one position further counter-clockwise than the l_i th vertex of cell l . For the energy constraints on cell-cell adhesion and membrane shape, it is necessary to calculate the rate a length changes when moving a vertex. This can be computed as

$$\nabla_i d_{l,j} = \frac{1}{d_{l,j}} \left(\begin{array}{c} x_i - x_j \\ y_i - y_j \end{array} \right), \quad (A.2)$$

where (x_i, y_i) are the coordinates of vertex i and (x_j, y_j) are the coordinates of the neighbouring vertex j .

Appendix B. Shortest axis calculation for cell division

When considering cell division in vertex dynamics models, it may be necessary to calculate the shortest axis of the element associated with a cell undergoing division. Let this element have M vertices with positions (x_m, y_m) for $i = 1, \dots, M$, where vertices are ordered anti-clockwise relative to the centroid of the element. Then the shortest axis is given by the line passing through the centroid of the element in the direction of the eigenvector corresponding to the larger eigenvalue of the inertial tensor

$$\begin{pmatrix} I_{xx} & -I_{xy} \\ -I_{xy} & I_{yy} \end{pmatrix}, \quad (\text{B.1})$$

where the moments of inertia I_{xx} , I_{yy} , and I_{xy} are defined by

$$I_{xx} = \frac{1}{12} \sum_{m=1}^M (x_m y_{m+1} - x_{m+1} y_m) (y_m^2 + y_m y_{m+1} + y_{m+1}^2), \quad (\text{B.2})$$

$$I_{yy} = \frac{1}{12} \sum_{m=1}^M (x_m y_{m+1} - x_{m+1} y_m) (x_m^2 + x_m x_{m+1} + x_{m+1}^2), \quad (\text{B.3})$$

$$I_{xy} = \frac{1}{24} \sum_{m=1}^M (x_m y_{m+1} - x_{m+1} y_m) (x_m y_{m+1} + 2x_m y_m + 2x_{m+1} y_{m+1} + x_{m+1} y_m). \quad (\text{B.4})$$

Note that if $I_{xy} = 0$, then the element is regular, so every line passing through its centroid is a shortest axis. In this case a random axis for division is chosen.

References

- Alber, M., Chen, N., Lushnikov, P., Newman, S., 2007. Continuous macroscopic limit of a discrete stochastic model for interaction of living cells. *Phys. Rev. Lett.* 99, 168102.
- Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P., 2002. *Molecular Biology of the Cell*, fourth ed. Garland Science.
- Allen, M., Tildesley, D., 1989. *Computer Simulation of Liquids*. Oxford University Press.
- Block, M., Schöll, E., Drasdo, D., 2007. Classifying the expansion kinetics and critical surface dynamics of growing cell populations. *Phys. Rev. Lett.* 99, 248101.
- Bonnet, I., Marcq, P., Bosveld, F., Fetler, L., Bellaïche, Y., Graner, F., 2012. Mechanical state, material properties and continuous description of an epithelial tissue. *J. R. Soc. Interface* 9, 2614–2623.
- Brodland, G., 2004. Computational modeling of cell sorting, tissue engulfment, and related phenomena: a review. *Appl. Mech. Rev.* 57, 47–77.
- Brodland, G., 2007. A new cell-based FE model for the mechanics of embryonic epithelia. *Comput. Meth. Biomech. Biomed. Eng.* 10, 121–128.
- Brodland, G., Chen, H., 2000a. The mechanics of cell sorting and envelopment. *J. Biomech.* 33, 845–851.
- Brodland, G., Chen, H., 2000b. The mechanics of heterotypic cell aggregates: insights from computer simulations. *J. Biomech. Eng.* 122, 402–407.
- Brodland, G., Conte, V., Cranston, P., Veldhuis, J., Narasimhan, S., Hutson, M., Jacinto, A., Ulrich, F., Baum, B., Miodownik, M., 2010. Video force microscopy reveals the mechanics of ventral furrow invagination in drosophila. *Proc. Natl. Acad. Sci. U.S.A.* 107, 22111–22116.
- Brodland, G., Veldhuis, J., 2002. Computer simulations of mitosis and interdependences between mitosis orientation, cell shape and epithelia reshaping. *J. Biomech.* 35, 673–681.
- Cai, A., Landman, K., Hughes, B., 2007. Multi-scale modeling of a wound-healing cell migration assay. *J. Theor. Biol.* 245, 576–594.
- Canela-Xandri, O., Sagués, F., Casademunt, J., Buceta, J., 2011. Dynamics and mechanical stability of the developing dorsoventral organizer of the wing imaginal disc. *PLoS Comput. Biol.* 7, e1002153.
- Chiou, K., Hufnagel, L., Shraiman, B., 2012. Mechanical stress inference for two dimensional cell arrays. *PLoS Comput. Biol.* 8, e1002512.
- Coxeter, H., 1973. *Regular Polytopes*. Dover.
- Dallon, J., Othmer, H., 2004. How cellular movement determines the collective force generated by the *Dictyostelium discoideum* slug. *J. Theor. Biol.* 231, 203–222.
- Dallon, J., Sherratt, J., Maini, P., Ferguson, M., 2000. Biological implications of a discrete mathematical model for collagen deposition and alignment in dermal wound repair. *Math. Med. Biol.* 17, 379–393.
- Davit, Y., Osborne, J., Byrne, H., Gavaghan, D., Pitt-Francis, J., 2013. Validity of the Cauchy–Born rule applied to discrete cellular-scale models of biological tissues. *Phys. Rev. E* 87, 042724.
- De Matteis, G., Graudenzi, A., Antoniotti, M., 2012. A review of spatial computational models for multi-cellular systems, with regard to intestinal crypts and colorectal cancer development. *J. Math. Biol.*, 1–54.
- Deutsch, A., Dormann, S., 2005. *Cellular Automaton Modeling of Biological Pattern Formation: Characterization, Applications, and Analysis*. Springer.
- Dollet, B., Graner, F., 2007. Two-dimensional flow of foam around a circular obstacle: local measurements of elasticity, plasticity and flow. *J. Fluid Mech.* 585, 181–211.
- Drasdo, D., 2000. Buckling instabilities of one-layered growing tissues. *Phys. Rev. Lett.* 84, 4244–4247.
- Drasdo, D., Hoehme, S., Block, M., 2007. On the role of physics in the growth and pattern formation of multi-cellular systems: what can we learn from individual-cell based models? *J. Stat. Phys.* 128, 287–345.
- Drasdo, D., Höhme, S., 1995. Monte Carlo approach to tissue–cell populations. *Phys. Rev. E* 52, 6635–6657.
- Drasdo, D., Höhme, S., 2003. Individual-based approaches to birth and death in avascular tumors. *Math. Comput. Model.* 37, 1163–1175.
- Farhadifar, R., Röper, J., Aigouy, B., Eaton, S., Jülicher, F., 2007. The influence of cell mechanics, cell-cell interactions, and proliferation on epithelial packing. *Curr. Biol.* 17, 2095–2104.
- Foty, R., Steinberg, M., 2005. The differential adhesion hypothesis: a direct evaluation. *Dev. Biol.* 278, 255–263.
- Fozard, J., Byrne, H., Jensen, O., King, J., 2010. Continuum approximations of individual-based models for epithelial monolayers. *Math. Med. Biol.* 27, 39–74.
- Galle, J., Aust, G., Schaller, G., Beyer, T., Drasdo, D., 2006. Individual cell-based models of the spatio-temporal organisation of multicellular systems – achievements and limitations. *Cytometry Part A* 69A, 704–710.
- Galle, J., Loeffler, M., Drasdo, D., 2005. Modeling the effect of deregulated proliferation and apoptosis on the growth dynamics of epithelial cell populations *in vitro*. *Biophys. J.* 88, 62–75.
- Galle, J., Preziosi, L., Tosin, A., 2009. Contact inhibition of growth described using a multiphase model and an individual cell based model. *Appl. Math. Lett.* 22, 1483–1490.
- Graner, F., Glazier, J., 1992. Simulation of biological cell sorting using a two-dimensional extended Potts model. *Phys. Rev. Lett.* 69, 2013–2016.
- Hardin, J., Walston, T., 2004. Models of morphogenesis: the mechanisms and mechanics of cell rearrangement. *Curr. Opin. Genet. Dev.* 14, 399–406.
- Harris, A., 1976. Is cell sorting caused by differences in the work of intercellular adhesion? A critique of the Steinberg hypothesis. *J. Theor. Biol.* 61, 267–285.
- Hirohashi, S., 1998. Inactivation of the E-cadherin-mediated cell adhesion system in human cancers. *Am. J. Pathol.* 153, 333–339.
- Honda, H., Eguchi, G., 1980. How much does the cell boundary contract in a monolayered cell sheet? *J. Theor. Biol.* 84, 575–588.
- Honda, H., Ogita, Y., Higuchi, S., Kani, K., 1982. Cell movements in a living mammalian tissue: long-term observation of individual cells in wounded corneal endothelia of cats. *J. Morphol.* 174, 25–39.
- Honda, H., Tanemura, M., Nagai, T., 2004. A three-dimensional vertex dynamics cell model of space-filling polyhedra simulating cell behavior in a cell aggregate. *J. Theor. Biol.* 226, 439–453.
- Honda, H., Yamanaka, H., Eguchi, G., 1986. Transformation of a polygonal cellular pattern during sexual maturation of the avian oviduct epithelium: computer simulation. *Development* 98, 1–19.
- Iserles, A., 2009. *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press.
- Ishihara, S., Sugimura, K., 2012. Bayesian inference of force dynamics during morphogenesis. *J. Theor. Biol.* 313.
- Kansal, A., Torquato, S., Harsh, G., Chicca, E., Deisboeck, T., 2000. Simulated brain tumor growth dynamics using a three-dimensional cellular automaton. *J. Theor. Biol.* 203, 367–382.
- Landsberg, K., Farhadifar, R., Ranft, J., Umetsu, D., Widmann, T., Bittig, T., Said, A., Jülicher, F., Dahmann, C., 2009. Increased cell bond tension governs cell sorting at the Drosophila anteroposterior compartment boundary. *Curr. Biol.* 19, 1950–1955.
- Lee, Y., Kouvroukoglou, S., McIntire, L., Zygourakis, K., 1995. A cellular automaton model for the proliferation of migrating contact-inhibited cells. *Biophys. J.* 69, 1284–1298.
- Ma, X., Lynch, H., Scully, P., Hutson, M., 2009. Probing embryonic tissue mechanics with laser hole drilling. *Phys. Biol.* 6, 036004.
- Maini, P., McElwain, D., Leavesley, D., 2004. Travelling waves in a wound healing assay. *Appl. Math. Lett.* 17, 575–580.
- Meineke, F.A., Potten, C.S., Loeffler, M., 2001. Cell migration and organization in the intestinal crypt using a lattice-free model. *Cell Prolif.* 34, 253–266.
- Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., 1953. Combinatorial minimization. *J. Chem. Phys.* 21, 1087–1092.
- Mirams, G., Arthurs, C., Bernabeu, M., Bordas, R., Cooper, J., Corrias, A., Davit, Y., Dunn, S.J., Fletcher, A., Harvey, D., Marsh, M., Osborne, J., Pathmanathan, P., Pitt-Francis, J., Southern, J., Zemzemi, N., Gavaghan, D., 2013. Chaste: an open source C++ library for computational physiology and biology. *PLoS Comput. Biol.* 9, e1002970.

- Murray, P., Edwards, C., Tindall, M., Maini, P., 2009. From a discrete to a continuum model of cell dynamics in one dimension. *Phys. Rev. E* 80, 031912-1.
- Nagai, T., Honda, H., 2001. A dynamic cell model for the formation of epithelial tissues. *Philos. Mag. B* 81, 699–719.
- Nagai, T., Honda, H., 2006. Wound healing mechanism in epithelial tissues cell adhesion to basal lamina. In: Proceedings of the 2006 WSEAS Int. Conf. on Cellular & Molecular Biology, Biophysics & Bioengineering. WSEAS Press, pp. 111–116.
- Pathmanathan, P., Cooper, J., Fletcher, A., Mirams, G., Murray, P., Osborne, J., Pitt-Francis, J., Walter, A., Chapman, S., 2009. A computational study of discrete mechanical tissue models. *Phys. Biol.* 6, 036001.
- Pitt-Francis, J., Pathmanathan, P., Bernabeu, M., Bordas, R., Cooper, J., Fletcher, A., Mirams, G., Murray, P., Osborne, J., Walter, A., Chapman, S., Garny, A., van Leeuwen, I., Maini, P., Rodriguez, B., Waters, S., Whiteley, J., Byrne, H., Gavaghan, D., 2009. Chaste: a test-driven approach to software development for biological modelling. *Comput. Phys. Commun.* 180, 2452–2471.
- Radice, G., 1980. The spreading of epithelial cells during wound closure in *Xenopus* larvae. *Dev. Biol.* 76, 26–46.
- Rudge, T., Haseloff, J., 2005. A computational model of cellular morphogenesis in plants. In: Advances in Artificial Life. Springer, pp. 78–87.
- Sandersius, S., Weijer, C., Newman, T., 2011. Emergent cell and tissue dynamics from subcellular modeling of active biomechanical processes. *Phys. Biol.* 8, 045007.
- Smith, A., Baker, R., Kay, D., Maini, P., 2012. Incorporating chemical signalling factors into cell-based models of growing epithelial tissues. *J. Math. Biol.* 65, 441–463.
- Steinberg, M., 1970. Does differential adhesion govern self-assembly processes in histogenesis? Equilibrium configurations and the emergence of a hierarchy among populations of embryonic cells. *J. Exp. Zool.* 173, 395–433.
- Taylor, J., 1976. The structure of singularities in soap-bubble-like and soap-film-like minimal surfaces. *Ann. Math.*, 489–539.
- Trichas, G., Smith, A., White, N., Wilkins, V., Watanabe, T., Moore, A., Joyce, B., Sugnaseelan, J., Rodriguez, T.A., Kay, D., Baker, R., Maini, P., Srinivas, S., 2012. Multicellular rosettes in the mouse visceral endoderm facilitate the ordered migration of ave cells. *PLoS Biol.* 10, e1001256.
- van Leeuwen, I., Mirams, G., Walter, A., Fletcher, A., Murray, P., Osborne, J., Varma, S., Young, S., Cooper, J., Doyle, B., Pitt-Francis, J., Momtahan, L., Pathmanathan, P., Whiteley, J., Chapman, S.J., Jensen, O., King, J., Maini, P., Waters, S., Gavaghan, D., Byrne, H., 2009. An integrative computational model for intestinal tissue renewal. *Cell Prolif.* 42, 617–636.
- Viens, D., Brodland, G., 2007. A three-dimensional finite element model for the mechanics of cell-cell interactions. *J. Biomech. Eng.* 129, 651–657.
- Weliky, M., Minsuk, S., Keller, R., Oster, G., 1991. Notochord morphogenesis in *Xenopus laevis*: simulation of cell behavior underlying tissue convergence and extension. *Development* 113, 1231–1244.
- Weliky, M., Oster, G., 1990. The mechanical basis of cell rearrangement. I. Epithelial morphogenesis during Fundulus epiboly. *Development* 109, 373–386.