

Cytopede: A Three-Dimensional Tool for Modeling Cell Motility on a Flat Surface

MARC HERANT and MICAH DEMBO

ABSTRACT

When cultured on flat surfaces, fibroblasts and many other cells spread to form thin lamellar sheets. Motion then occurs by extension of the sheet at the leading edge and retraction at the trailing edge. Comprehensive quantitative models of these phenomena have so far been lacking and to address this need, we have designed a three-dimensional code called Cytopede specialized for the simulation of the mechanical and signaling behavior of plated cells. Under assumptions by which the cytosol and the cytoskeleton are treated from a continuum mechanical perspective, Cytopede uses the finite element method to solve mass and momentum equations for each phase, and thus determine the time evolution of cellular models. We present the physical concepts that underlie Cytopede together with the algorithms used for their implementation. We then validate the approach by a computation of the spread of a viscous sessile droplet. Finally, to exemplify how Cytopede enables the testing of ideas about cell mechanics, we simulate a simple fibroblast model. We show how Cytopede allows computation, not only of basic characteristics of shape and velocity, but also of maps of cell thickness, cytoskeletal density, cytoskeletal flow, and substratum tractions that are readily compared with experimental data.

Key words: cytoskeleton, fibroblast, finite element, sessile drop, thin film, traction force microscopy.

1. INTRODUCTION

ANIMAL CELLS PLATED ON A FLAT CULTURE DISH, as is the case in the vast majority of laboratory experiments, take on an archetypal “fried egg” appearance. During migration, they spontaneously assume a polarized shape that varies according to cell type (e.g., hand-mirror for fibroblasts, or crescent for fish keratocytes). A thin sheet of cytoplasm extends at the leading edge (the lamellipodium) and undergoes cycles of protrusion, adhesion, and traction. Most of the time, this morphology is robustly preserved under perturbations but occasionally, certain maneuvers have drastic effects, in particular those that directly affect cytoskeletal organization.

The experimental methods of molecular and cell biology (imaging, biochemistry, genetics) have revealed a large number of the key chemical players and pathways involved in generating and controlling cell motility (Bakal et al., 2007; Jaffe and Hall, 2005). This information has been supplemented by biophysical

approaches that include speckle imaging, FRAP, FLAP, atomic force microscopy and traction microscopy (Harris et al., 1980; Dembo et al., 1996; Dembo and Wang, 1999; Zicha et al., 2003; Waterman-Storer and Danuser, 2002; Danuser and Waterman-Storer, 2006). Unfortunately, analysis and integration of the massive datasets generated by these methods presents major conceptual, mathematical and numerical difficulties. To help bridge the gap we offer here a prototypical tool for cytomechanical modeling and computation, which we have named Cytopede. Cytopede is based on and extends an earlier computational environment for cytomechanics—the RIF, or “reactive interpenetrative flow” method (Dembo and Harlow, 1986).

The overarching scheme of Cytopede is derived from a sense that the issues and challenges involved in understanding cell motility are numerous and difficult but also highly modular and separable. Therefore, Cytopede starts by a process of deconstruction and analysis to enumerate and computationally implement fundamental physical components such as “viscosity” or “protrusive force” that drive a cell from one mechanical state to the next. These building blocks can then be combined and modulated in an infinite variety of ways.

In addition, Cytopede was designed to incorporate two key sets of attributes: first to automatically enforce fundamental laws of mass and momentum transport and conservation which, like everything else, cells must obey; second to allow straightforward contact with laboratory experiments. For the latter, compatibility with certain types of datasets is especially important. These are: (i) cell geometry which is simply given by the cell contour, and which may occasionally be complemented by information on cell thickness within that contour, (ii) cytoskeletal concentration, composition, reactivity and flow as revealed by fluorescent markers, and (iii) traction forces which can be determined via the deformable substratum method.

Cytopede is a software package dedicated to the simulation of free-boundary multiphase low-Reynolds number three-dimensional films in general and to the simulation of spread cells in particular. It uses the finite element method, distinguishes cytosolic from cytoskeletal dynamics (hence the label multiphase), and allows a wide range of shape changes as long as the ventral side of the cell that is modeled remains in contact with a flat substratum. Computationally innovative aspects include a new kind of adaptive mesh with specialized volume elements, surface elements, and line elements to allow for flexible implementation of free boundaries dynamics and of adhesion and detachment dynamics between a cell and its substrate.

As it matures, Cytopede is intended to be freely available to the community at large of researchers interested in modeling cell motility. The main objectives of this article are thus to some extent pedagogical and encompass the following themes: a discussion of the physical concepts that inform the approach used in Cytopede to address problems of cell motility; a description of the implementation of these concepts through the algorithms used by Cytopede; testing of the code through a numerical experiment performed on a viscous drop with surface tension spreading under the influence of gravity; and a biological example in which Cytopede is used to formulate and solve a simple model of a locomoting cell.

2. PHYSICAL CONCEPTS

Compared to other instances of locomotion, the most remarkable feature of a migrating cell is the rapid chemical turnover of its structural elements. Cytoskeletal polymers are assembled at the front of the cell from components floating in the cytosol (actin monomers, cross-linkers, adhesion proteins, motor proteins, etc.). They are then disassembled, reassembled, and so forth through some unknown number of cycles, before reaching the rear. The simple fact that the reactivity of the cytoskeleton occurs on the same time scale as its motion means that the two phenomena cannot be separated and that a cell truly “makes itself anew” as it moves forward. Indeed, the importance and close connection between the chemistry and physics of the cytoskeleton has long been recognized in biophysical and modeling circles and has received overwhelming experimental confirmation (Watanabe and Mitchison, 2002). For the purposes of a physical analysis, the first consequence of this extreme reactivity is that it precludes models in which the cytoplasm is reduced to a single continuum. At minimum, it is necessary to consider interpenetrating cytosolic and cytoskeletal phases, and one must focus on the structural rules that govern the definition of the phases and the dynamical rules that govern their interconversion. Thus, actin monomers will contribute to the

cytoskeletal phase if they are in the filamentous form and to the cytosolic phase when in the globular diffusible form but they cannot be part of both phases simultaneously.

With this in mind, consider the three principal structural components of animal cells:

- *The cytoskeleton* is a porous continuum consisting mainly of actin filaments that resists deformation through viscoelastic properties and is driven by molecular motors (e.g., myosin), polymerization forces, and thermodynamic (colloid) effects (e.g., electrostatic or steric).
- *The cytosol* flows passively through the cytoskeleton; it is a medium for the propagation of diffusible signals. It is incompressible and therefore conducts pressure. Finally, it can be converted to cytoskeleton via the polymerization of dissolved monomers (e.g., G-actin \rightarrow F-actin) and vice versa.
- *The plasmalemma (or cortical membrane)* defines the boundary of the cell by controlling (and often preventing) electrical, chemical and volumetric exchanges with the external medium. It is furthermore highly flexible to bending motions, highly fluid to shearing deformations, and yet very resistant to area expansions. Together, these three properties make it a good conductor of stress in the form of cortical tension.

Note that this classification ignores organelles, such as granules, or the cell nucleus. One could of course construct models that include such elements, but in most circumstances, the cost in terms of complexity becomes so large as to dominate the scientific return. In addition, the ability of cell fragments to migrate autonomously (Verkhovsky et al., 1999) lends further credibility to this approximation.

If one makes the key assumption that at the mesoscopic scale—i.e., at a scale small compared to the whole cell but large compared to individual molecules—the properties of the cell can be represented by continuous fields, then the general framework of continuum mechanics can be applied to animal cells just as it is done with any other materials. Given a cell occupying a simply connected domain $\Omega(t)$ with boundary $\partial\Omega(t)$, Cytopede assembles an internally consistent set of coupled differential equations, boundary conditions and initial conditions that govern the temporal evolution of the domain itself and of various scalar and vector fields defined in its interior and on its surface. If $\mathbf{x} \in \Omega$ is a position vector and t time, then the most basic and important fields are:

- $\theta_n(\mathbf{x}, t)$ the network phase (cytoskeleton) volume fraction,
- $\theta_s(\mathbf{x}, t)$ the solvent phase (cytosol) volume fraction,
- $\mathbf{v}_n(\mathbf{x}, t)$ the network velocity field,
- $\mathbf{v}_s(\mathbf{x}, t)$ the solvent velocity field, and
- $P(\mathbf{x}, t)$ the cytoplasmic pressure.

The evolution equations for the quantities θ_n , θ_s , \mathbf{v}_n , \mathbf{v}_s , and P are then determined by the laws of mass and momentum conservation.

Of note is that since it was developed in the context of cell motility in the 1980s (Dembo and Harlow, 1986), the two-phase fluid description is becoming more prevalent with several recent computational implementations (Rubinstein et al., 2005; Zajac et al., 2008).

2.1. Mass conservation

The fact that we only consider two phases (cytoskeleton and cytosol) mandates that the sum of network and solvent volume fractions is unity:

$$\theta_n + \theta_s = 1. \quad (1)$$

Net cytoplasmic volume flow is given by the sum of the flow of cytosolic volume and cytoskeletal volume, i.e. $\mathbf{v} = \theta_n \mathbf{v}_n + \theta_s \mathbf{v}_s$. Since both components of the cytoplasm are condensed phases, it is an excellent approximation to regard the combined flow as incompressible ($\nabla \cdot \mathbf{v} = 0$):

$$\nabla \cdot (\theta_n \mathbf{v}_n + \theta_s \mathbf{v}_s) = 0. \quad (2)$$

Finally, conservation of cytoskeleton implies that the rate of change of network concentration at a given point in space (Eulerian derivative) is the sum of an advective transport term describing the net inflow of network, and a source term \mathcal{J} which represents the net rate of *in situ* cytoskeletal production by polymerization:

$$\frac{\partial \theta_n}{\partial t} = -\nabla \cdot (\theta_n \mathbf{v}_n) + \mathcal{J}. \quad (3)$$

Obviously \mathcal{J} depends on a prescription for local chemical activity that needs to be provided separately. Eq. 3 has a counterpart for the solvent

$$\frac{\partial \theta_s}{\partial t} = -\nabla \cdot (\theta_s \mathbf{v}_s) - \mathcal{J}. \quad (4)$$

which, when taken together with Eq. 1 unsurprisingly reduces to Eq. 2. As a result, only Eqs. 1, 2, and 3 are needed and Eq. 4 is redundant.

2.2. Momentum conservation

The momentum equations for the solvent and network phases are simplified by two observations. Because of the small dimensions and velocities involved, inertial terms can be neglected. Second, the essentially aqueous nature of the cytosol implies that its characteristic viscosity is not very different from that of water (0.02 poise). Since this is much less than typical cytoplasmic viscosities (~ 1000 poise), the viscous stress is carried by the cytoskeletal (network) phase while the cytosolic (solvent) phase is nearly inviscid. Given these approximations, the only two forces that act on the solvent are pressure gradients and solvent-network drag—that is, the drag force due to solvent flow through the network when velocities are mismatched. In the spirit of Darcy's law, the solvent momentum equation can be written

$$-\theta_s \nabla P + \mathcal{H} \theta_s \theta_n (\mathbf{v}_n - \mathbf{v}_s) = 0. \quad (5)$$

P is the cytoplasmic pressure and it is assumed that, as for the partial pressures of a mixture of gases, it is shared by the cytosolic and cytoskeletal phases according to concentration (volume fraction). \mathcal{H} is the solvent-network drag coefficient more familiar as the product $\theta_n \mathcal{H}$ representing the hydraulic resistance that appears in the usual form of Darcy's equation. Theory (Scheidegger, 1960) as well as experiments on polymer networks (Tokita and Tanaka, 1991) give estimates of \mathcal{H} that lead to small drag forces compared to other forces acting within the cytoplasm, chief among them the cytoskeletal viscosity. This is not surprising since \mathcal{H} is roughly proportional to the solvent viscosity which is small compared to the network viscosity. The smallness of \mathcal{H} in turn implies that pressure gradients are small, or that pressure is close to uniform inside the cell; although, for a different view point, see Mitchison et al. (2008) and Charras et al. (2009). Thus, from the point of view of overall cell shape and motion as determined by cytoskeletal dynamics (Eq. 6 below), the precise value \mathcal{H} does not matter as long as it is sufficiently small. However, from the point of view of internal cytosolic flow, which can play an important transport role, the value of \mathcal{H} does matter, and pressure gradients, even though small are not negligible.

It is in the network (cytoskeleton) momentum equation that the rich complexity of cell mechanics becomes evident. Aside from pressure gradients and solvent-network drag, the network is also subject to viscous, elastic, and interaction forces, and the network momentum equation can therefore be written

$$-\theta_n \nabla P - \mathcal{H} \theta_s \theta_n (\mathbf{v}_n - \mathbf{v}_s) + \nabla \cdot [\nu (\nabla \mathbf{v}_n + (\nabla \mathbf{v}_n)^T)] - \nabla \cdot \Psi + \mathbf{F}_{\text{ext}} = 0. \quad (6)$$

Here, ν is the network viscosity, Ψ is the part of the network stress tensor remaining under static conditions, and \mathbf{F}_{ext} is an external body force. The term Ψ can include inter-filament interactions (such as contractility due to myosin activity or swelling due to colloid osmotic effects), filament-membrane interactions, elastic forces due to deformations, etc.

2.3. Remarks on protrusive forces due to cytoskeleton-membrane interactions

Current ideas about cellular protrusion give a central role to the interaction of polymerizing filaments with the plasma membrane (Hill and Kirschner, 1982; Peskin et al., 1993; Dickinson and Purich, 2002; Kovar and Pollard, 2004). A comprehensive discussion of how such mechanisms can be described within a continuum mechanical framework is beyond the scope of this outline and will be the subject of a separate article, but we briefly mention key principles. As is always the case in the low Reynolds number regime, we require no net forces, so that any force on the membrane (e.g., for protrusion) must be balanced by an equal and opposite force on the network. This can be enforced automatically if the force is derived from the

divergence of a tensor field Ψ^{nM} . In addition, because the membrane is fluid, cytoskeleton-membrane interactions cannot generate a shear stress in a patch of free membrane (this may change at adherent membrane). Then, at a patch of free membrane, the stress will be typically of the form:

$$\Psi^{nM} = \psi^{nM} \mathbf{nn}, \quad (7)$$

where \mathbf{nn} is the dyadic of the unit vector outward normal to the membrane and where ψ^{nM} is a scalar which may, for instance, be a function of local network density and/or polymerization at the membrane. In this work and elsewhere, we have essentially used a δ -function at the boundary which embodies the stress times its range d_M away from the membrane (Herant et al., 2003; Herant and Dembo, 2006) and set $\Psi^{nM} = 0$ elsewhere inside the cell. In a forthcoming work we will describe a more complete approach to the cytoskeleton-membrane interaction stress fields where the definition Ψ^{nM} at the boundary of the cell with condition Eq. 7 can be extended to define a more physical generalized membrane-network stress field in the intracellular space.

2.4. Boundary conditions

As partial differential equations, the evolution equations must be complemented by boundary conditions and this is where characteristics of the plasma membrane come into play. For mass conservation, we assume that the membrane remains impermeable to the cytoskeleton (which may even be anchored to the membrane) so that:

$$\mathbf{v}_M \cdot \mathbf{n} = \mathbf{v}_n \cdot \mathbf{n} \quad (8)$$

where \mathbf{v}_M is the velocity of the boundary and \mathbf{n} is the outward normal unit vector. If we also assume that the membrane is impermeable to the cytosol (which appears to be true in some cases and not in others) we also have

$$\mathbf{v}_M \cdot \mathbf{n} = \mathbf{v}_s \cdot \mathbf{n} \quad (9)$$

but this condition can be relaxed to allow a net volume flux through the boundary.

For momentum conservation, there are two main possibilities: (1) the boundary is constrained by interaction with a solid surface as in the case of the cell/substratum interface, or (2) the boundary is free membrane bathed by an inviscid external medium. In (1), the boundary condition reduces to a constraint on the normal component of the velocity (which must be 0), and, if a no slip condition also applies, there are additional constraints on tangential components of the velocity. In (2), the boundary condition amounts to a stress continuity requirement:

$$\nu(\nabla \mathbf{v}_n + (\nabla \mathbf{v}_n)^T) \cdot \mathbf{n} - \Psi \cdot \mathbf{n} - P\mathbf{n} = -2\gamma\kappa\mathbf{n} - P_{\text{ext}}\mathbf{n} + \sigma, \quad (10)$$

where γ is the surface tension, κ is the mean curvature of the membrane, \mathbf{n} the outward normal to the membrane, and σ an externally imposed traction vector. This kind of boundary condition typically applies to the free back surface of the cell.

2.5. Constitutive equations

As presented, the mass and momentum conservation equations are cast in a general framework that must be defined by further prescriptions to provide closure of the system. Thus, the constitutive equations embody most of what one usually thinks of as the biological and regulatory aspects of cellular dynamics. These can be made complicated and detailed or simple and schematic depending on the specific modeling objective. As an example, it is plausible that the cytoskeletal viscosity ν (appearing in Eq. 6) should approach zero when the network volume fraction approaches zero. A simple scheme to satisfy this requirement would suggest that viscosity be determined by a constitutive law such as

$$\nu = \nu_0 \theta_n \quad (11)$$

prescribing a linear relation between viscosity and network concentration. A more complex idea designed to account for the possibility of network gelation may be expressed by

$$\nu = \nu_0 \theta_n \exp\left(\frac{\theta_n}{\theta_{\text{gel}}}\right), \quad (12)$$

where θ_{gel} is a parameter that stipulates the volume fraction at which the probability of inter-filament crosslinking exceeds a percolation threshold for gelation. In principle, the form of the constitutive law for cytoskeletal viscosity could be verified empirically by investigating the rheology of the cytoplasm at various cytoskeletal concentrations; however, in general such experimental evidence is sparse and often difficult to interpret so that constitutive laws such as Eq. 12 are educated guesses.

The considerable uncertainties and complexities of physiological feedback and regulation make it likely that for the foreseeable future, progress in modeling cell motility will generally require a process of trial and error. It will be necessary to formulate various plausible constitutive relations, carry out computations to work out their consequences, and compare these consequences with experimental observations. It is in large measure to enable this process that we have created Cytopede, and it is also the reason why Cytopede is designed to allow maximum latitude for constitutive laws of every possible sort. Thus, the code allows the user freedom to set the constitutive laws that govern \mathcal{J} (the network formation or cytoskeletal polymerization rate), \mathcal{H} (the resistance to solvent flow through the network), Ψ (the network stress due to elasticity and static interactions), ν (the network viscosity), γ (the tension of the cortical membrane), along with other parameters that may govern behavior such as diffusion and transport of reactants, etc. In addition, it is also possible to define rules for attachment to and detachment from the substratum, processes that clearly play a key role in cell spreading and migration. The main advantage of this open and modular formalism is that one can start with a minimal and schematic approach and test the results and then gradually build complexity and detail step by step to virtually unlimited levels.

3. COMPUTATIONAL ALGORITHMS

Although we have tried to make it as self-contained as possible, this section presupposes a passing familiarity with the finite element method for which many textbooks exist; our favorite is Hughes (2000). In what follows, we outline the ideas behind Cytopede as well as key computational components. Algebraic developments are found in the Appendix. The Cytopede code itself is written in Fortran 90, and can be compiled and run on standard linux desktop PC workstations. For most of our visualization needs, we use gmV (General Mesh Viewer, freely available).

3.1. Numerical method and mesh structure

We use a Galerkin finite element method for several reasons. First, it is a well-established method for solving partial differential equation with solid theoretical underpinnings and with a considerable track record in many practical applications. We demand a conservative method that does not in itself add additional layers of uncertainty to computational results. Second, it is well-suited for free boundary problems and capable of accommodating the complex shapes and deformations observed in migrating cells. Third, it allows a straightforward and precise way of implementing the range of boundary conditions encountered in studies of cell motility despite a complex and dynamic boundary geometry.

For three-dimensional finite element simulations on an evolving computational domain, the management of the mesh represents the most arduous challenge, mainly because of the necessity to perform reliable automatic rezoning over a volume whose evolving shape cannot be predicted in advance. In addition, visualization and debugging difficulties increase exponentially with mesh complexity. This is why the overriding consideration in the selection of a mesh structure is for the maximum simplicity still compatible with a reasonably accurate simulation. This is also the reason why prior efforts to model spread cells have previously generally been limited to two-dimensional computations (Bottino et al., 2002; Rubinstein et al., 2005) or to semi-analytical calculations in the lubrication approximation (Oliver et al., 2005).

A spread cell on a substratum essentially represents a thin flat body. For this configuration, a pavement consisting of a single layer of 8-node brick elements is probably the simplest mesh structure possible but it has a drawback. In classic viscous film flow over a surface, the lower contact boundary has $v = 0$ while the upper free boundary has $\partial v / \partial z = 0$, leading to a classic parabolic (Poiseuille) profile with height. The same probably holds true for certain cytoskeletal flows, in particular for centripetal flow where there may be

anchoring at the substratum and retrograde flow higher up. For these types of flow, the linear resolution afforded by two nodes in height is inadequate. Instead we make use of a single layer of 12-node elements with quadrilateral ventral and dorsal faces and with 4 intermediate nodes located between the ventral and dorsal nodes (Fig. 1). This leads to quadratic accuracy vertically (sufficient to reproduce a Poiseuille type flow) and linear accuracy horizontally. However, horizontal resolution can be increased by mesh refinement while vertical resolution cannot.

We define as a *stack* a series of three nodes connected vertically. Thus each element is made up of four stacks, and in a given stack, the first node is the ventral node (usually with $z = 0$), the third node is the dorsal node, and the second node is the middle node. A stack is termed an edge stack if it lies at the edge of the mesh structure. The ventral nodes of edge stacks make up the contact line. The ventral surface of the domain consists of all the ventral faces of the elements. The back surface of the domain consists of all the dorsal faces of the elements together with the all the edge faces linking two edge stacks. Both these surfaces meet and are bounded by the contact line (Fig. 2).

3.2. Shape functions

We use standard isoparametric shape functions corresponding to the mapping of an element from real coordinates (x, y, z) to natural coordinates (ξ, η, ζ) ; (Fig. 1):

$$H_{kl} = \frac{1}{8}(1 \pm \xi)(1 \pm \eta)\zeta(\zeta \pm 1), \tag{13}$$

for ventral ($l = 1, \zeta = -1$) or dorsal ($l = 3, \zeta = +1$) nodes belonging to stack k where the signs are chosen such that the shape function is unity at node kl and vanishes at all others, and:

$$H_{k2} = \frac{1}{4}(1 \pm \xi)(1 \pm \eta)(1 - \zeta^2), \tag{14}$$

for middle nodes ($l = 2, \zeta = 0$), and again the signs are chosen such that the shape function is unity at node $k, 2$ and vanishes at all other nodes of the element. Any field Φ defined at the nodes can be interpolated to an arbitrary position $|\xi|, |\eta|, |\zeta| \leq 1$ within an element as

$$\Phi(\xi, \eta, \zeta) = \sum_{k=1}^4 \sum_{l=1}^3 \Phi_{kl} H_{kl}(\xi, \eta, \zeta) = \sum_{i \in e} \Phi_i H_i(\xi, \eta, \zeta), \tag{15}$$

where i runs over all the nodes of a given element e . In particular, the mapping from natural coordinates to real coordinates is given by expressions of the type:

$$x(\xi, \eta, \zeta) = \sum_{i \in e} x_i H_i(\xi, \eta, \zeta). \tag{16}$$

The Jacobian of the coordinate transformation is given by:

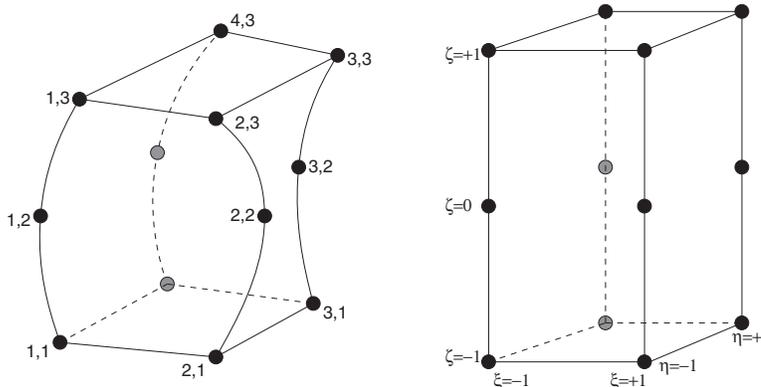
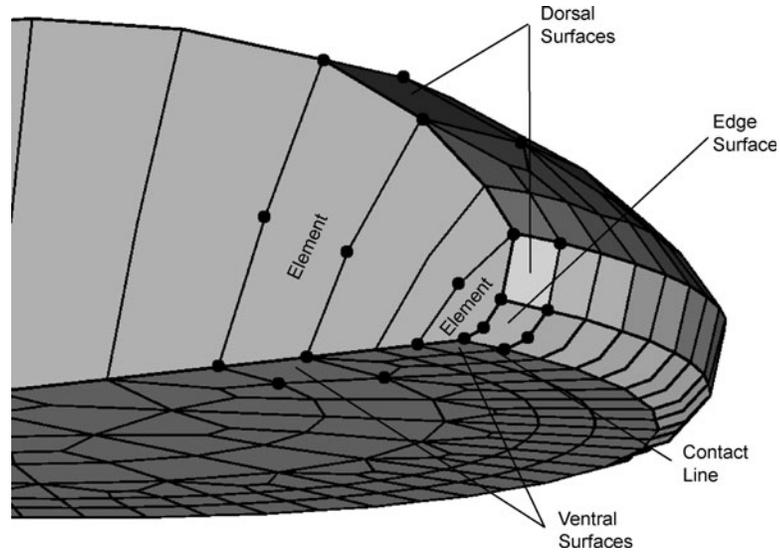


FIG. 1. An element in real (left) and natural (right) space. On the left, the nodes are labeled by stack and level.

FIG. 2. A slice through a Cytoskeleton simulation. Two elements are marked with those of their nodes that are visible. The internal element has a dorsal and ventral surface. The edge element has a dorsal and ventral quadrilateral surface, and an edge hexalateral surface.



$$\mathbf{J} = \begin{pmatrix} \partial x / \partial \zeta & \partial x / \partial \eta & \partial x / \partial \zeta \\ \partial y / \partial \zeta & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial z / \partial \zeta & \partial z / \partial \eta & \partial z / \partial \zeta \end{pmatrix}, \quad (17)$$

with for instance

$$\frac{\partial x}{\partial \xi} = \sum_{i \in e} x_i \frac{\partial H_i}{\partial \xi}. \quad (18)$$

Partial derivatives of the field Φ can be computed, e.g.:

$$\begin{aligned} \frac{\partial \Phi}{\partial x} &= \sum_{i \in e} \Phi_i \frac{\partial H_i}{\partial x} \\ &= \sum_{i \in e} \Phi_i \left(\frac{\partial H_i}{\partial \xi} \frac{\partial \xi}{\partial x} + \frac{\partial H_i}{\partial \eta} \frac{\partial \eta}{\partial x} + \frac{\partial H_i}{\partial \zeta} \frac{\partial \zeta}{\partial x} \right) \end{aligned} \quad (19)$$

where terms such as $\partial \xi / \partial x$ are computed by inverting the Jacobian \mathbf{J} .

Surface shape functions S are defined in a way analogous to the volume shape function H . However, the form of an S depends on whether a quadrilateral (ventral or dorsal) or a hexalateral (edge) surface is considered. In the first case

$$S_i = \frac{1}{4} (1 \pm \xi)(1 \pm \eta) \quad (20)$$

where the signs are chosen such that the shape function is unity at node i and vanishes at the other 3 nodes that belong to the quadrilateral. In the second case,

$$S_i = \frac{1}{4} (1 \pm \xi)\zeta(\zeta \pm 1), \quad (21)$$

if node i is a ventral or dorsal node and

$$S_i = \frac{1}{2} (1 \pm \xi)(1 - \zeta^2), \quad (22)$$

if node i is a middle node. Signs are determined as before.

For a quadrilateral surface with natural coordinates ξ, η , the surface mapping from natural coordinates to real coordinates is then given by expressions of the type

$$x(\xi, \eta) = \sum_{i \in e4} x_i S_i(\xi, \eta) \quad (23)$$

where i runs over the 4 nodes of the quadrilateral $e4$. The Jacobian of the coordinate transformation (which is not square) is given by

$$\mathbf{J}_S = \begin{pmatrix} \partial x / \partial \xi & \partial x / \partial \eta \\ \partial y / \partial \xi & \partial y / \partial \eta \\ \partial z / \partial \xi & \partial z / \partial \eta \end{pmatrix}, \quad (24)$$

with for instance

$$\frac{\partial x}{\partial \xi} = \sum_{i \in e4} x_i \frac{\partial S_i}{\partial \xi}. \quad (25)$$

Similar expressions are obtained for edge surfaces by substituting ζ for η and running i over the 6 nodes of the hexalateral $e6$.

3.3. Quadratures

In our evaluation of volume integrals we use Gaussian quadrature with 12 Gauss points per element located at natural coordinates $\xi = \pm \sqrt{3}/3$, $\eta = \pm \sqrt{3}/3$, and $\zeta = -\sqrt{3}/5, 0, +\sqrt{3}/5$:

$$\begin{aligned} \int_{\text{element}} dV \Phi(x, y, z) &= \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta \int_{-1}^{+1} d\zeta \Phi(\xi, \eta, \zeta) \det(\mathbf{J}) \\ &\sim \sum_{\text{GP} \in e} w_{\text{GP}} \det(\mathbf{J}_{\text{GP}}) \Phi(\xi_{\text{GP}}, \eta_{\text{GP}}, \zeta_{\text{GP}}) \\ &= \sum_{\text{GP} \in e} w_{\text{GP}} \det(\mathbf{J}_{\text{GP}}) \sum_{i \in e} \Phi_i H_i(\xi_{\text{GP}}, \eta_{\text{GP}}, \zeta_{\text{GP}}) \end{aligned} \quad (26)$$

where w_{GP} is the weight of the Gauss points (5/9 for $\zeta = \pm \sqrt{3}/5$ and 8/9 for $\zeta = 0$) and where $\det(\mathbf{J}_{\text{GP}})$ is the determinant of the Jacobian for the map $xyz(\xi, \eta, \zeta)$ evaluated at the Gauss points. The quantity

$$V_{\text{GP}} = w_{\text{GP}} \det(\mathbf{J}_{\text{GP}}) \quad (27)$$

can be assimilated to the volume associated with a Gauss point, and the total volume of an element is therefore the sum of the volumes of its 12 Gauss points.

The evaluation of volume integrals of gradients is as follows:

$$\begin{aligned} \int_{\text{element}} dV \nabla_{xyz} \Phi(x, y, z) &= \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta \int_{-1}^{+1} d\zeta \nabla_{xyz} \Phi(\xi, \eta, \zeta) \det(\mathbf{J}) \\ &\sim \sum_{\text{GP} \in e} V_{\text{GP}} \sum_{i \in e} \Phi_i \nabla_{xyz} H_i(\xi_{\text{GP}}, \eta_{\text{GP}}, \zeta_{\text{GP}}) \end{aligned} \quad (28)$$

where the real coordinate derivatives of the shape function H_i are evaluated at a Gauss point by inversion of the Jacobian \mathbf{J}_{GP} (see Eqs. 17–19).

To compute surface integrals, we use 4 Gauss points for ventral or dorsal quadrilaterals, and 6 Gauss points for edge hexalaterals. The surface normal at a Gauss point is given (in the case of a quadrilateral) by

$$\mathbf{N}_{\text{GP}} = \begin{pmatrix} \partial x / \partial \xi \\ \partial y / \partial \xi \\ \partial z / \partial \xi \end{pmatrix} \wedge \begin{pmatrix} \partial x / \partial \eta \\ \partial y / \partial \eta \\ \partial z / \partial \eta \end{pmatrix}. \quad (29)$$

The area A_{GP} associated with a surface Gauss point is then given by

$$A_{\text{GP}} = w_{\text{GP}} \|\mathbf{N}_{\text{GP}}\| = w_{\text{GP}} \sqrt{N_x^2 + N_y^2 + N_z^2} \quad (30)$$

where $w_{\text{GP}} = 1$ for quadrilaterals. For hexalaterals, $w_{\text{GP}} = 5/9$ for the four $\zeta = \pm \sqrt{3}/5$ Gauss points and $w_{\text{GP}} = 8/9$ for the two $\zeta = 0$ Gauss points.

The integral of a surface field is then

$$\int_{\text{surface}} d\sigma \Phi(x, y, z) \sim \sum_{\text{GP} \in e4} A_{\text{GP}} \sum_{i \in e4} \Phi_i S_i(\zeta_{\text{GP}}, \eta_{\text{GP}}) \quad (31)$$

where the index i runs over the four nodes of the quadrilateral $e4$ (or the hexalateral $e6$ with substitution of ζ_{GP} for η_{GP}).

The integral of the gradient of a scalar surface field can similarly be evaluated:

$$\int_{\text{surface}} d\sigma \nabla_S \Phi(x, y, z) \sim \sum_{\text{GP} \in e4} A_{\text{GP}} \sum_{i \in e4} \Phi_i \nabla_S S_i(\zeta_{\text{GP}}, \eta_{\text{GP}}) \quad (32)$$

Details on how the surface gradient of a shape function ($\nabla_S S_i$) is computed are provided in the Appendix.

3.4. Basic computational sequence

Given a network velocity field at time t , the simulation is advanced to $t + \Delta t$ by means of five sequential operations:

1. *Mesh Advection*: An advected domain geometry is calculated by moving all mesh nodes with the local network velocity over the time interval Δt . If this motion causes a contact constraint to be violated (e.g., if the mesh motion causes a currently free surface to contact the substrate), then the contact line is advanced or retracted accordingly.
2. *Rezoning*: Nodes of the advected mesh are repositioned so that the distortion of volume and surface elements is minimized. The node movements during rezoning are constrained so as to preserve domain geometry (e.g., the contact line and the mesh surface contour and volume). The rezoning also maintains topology and number of elements.
3. *Material Advection*: The network, and all optional node attributes representing mass concentration of one kind or another are advected with the appropriate Lagrangian motion over the time interval Δt . The resulting fields are then interpolated to the new mesh generated by steps 1 and 2 (this approach is sometimes called a general Euler-Lagrangian advection scheme).
4. *Diffusion and Reaction*: All user-supplied constitutive rules controlling diffusion and reactions are evaluated for whatever chemical species may be included in the model. This usually involves determination of reaction rates, diffusion coefficients, and surface flux rates. Concentration of the various chemical species is then evolved on the new mesh over the interval Δt according to diffusive transport, chemical reactions and boundary fluxes.
5. *Momentum Transport*: All user-supplied constitutive rules controlling momentum transport are evaluated. This may involve the network viscosity, the surface tension of the plasma membrane, and many other parameters. The momentum equations, the incompressibility condition and applicable boundary conditions are solved to determine the pressure, network velocity and solvent velocity on the new mesh.

The above computational cycle is repeated until the desired termination condition is reached (i.e., a pre-specified evolution time, or a pre-specified behavior endpoint). Note that a simulation typically begins without an initial velocity solution, so that operation 5 must be performed first before entering the sequence 1–5.

The time step Δt is determined by the strongest of two constraints. First a Courant condition; given a velocity field the time step should not be more than 5% of the element crossing time. Second, a chemistry/diffusion constraint set by the monitoring of numerically estimated second time derivatives of the evolved species, and requiring 1% accuracy.

The overall approach is first-order accurate in time. We note that a higher order integration method would be an obstacle to the modularization of the individual tasks listed above that enables a relatively tractable computational process.

3.5. Step 1. Mesh advection

The basic requirement for the mesh is that its boundaries must follow the evolution of the boundaries of the cell, which is itself determined by the prior configuration and the cytoskeletal velocity field. For this reason it is desirable to have the mesh nodes move with the network velocity. However, without further intervention, such an approach would rapidly lead to severe distortion of the mesh with misshapen ele-

ments, and eventually, a computationally fatal folding of the mesh structure. Taking this into account, the mesh is evolved in a sequence of steps:

1. The mesh nodes move following the network flow, e.g., for node i

$$\mathbf{x}_i^a = \mathbf{x}_i^o + \Delta t \mathbf{v}_{ni}, \quad (33)$$

where the superscript a designates the advected mesh, and the superscript o designates the old mesh.

2. The ventral node positions are readjusted vertically to have $z = 0$. This is necessary because ventral boundary conditions on the velocity are implemented with a penalty method that leads to very small, but nonzero vertical velocities ($< 10^{-6}$ of the typical velocities in the problem). The readjustment to the $z = 0$ plane therefore has minimal impact on the calculation but makes life simpler.
3. The nodes of the contact line are shifted forward (and occasionally backward) if necessary. This is discussed in detail below, but the idea is that when the contact angle becomes $> 180^\circ$, the contact line advances. Similarly, it is possible to set a minimum contact angle below which the contact line retreats.

3.5.1. Forward motion of the contact line when the boundary condition is stick. When the velocity boundary condition at the substratum allows slippage, the contact line is simply advected with the matter as per Eq. 33. However when the velocity boundary condition at the substratum is no slip ($\mathbf{v} = 0$), the situation demands more care. Our approach consists in letting the leading edge simply pivot down around the contact line and make contact with the substratum in a natural way. Note that this is only possible because the elements are quadratic in height.

Consider an edge boundary stack of three nodes 1, 2, and 3 (Fig. 3). The parametrized equation of the edge curve is given by:

$$x = x_1 L_1(\zeta) + x_2 L_2(\zeta) + x_3 L_3(\zeta), \quad (34)$$

and similarly for y and z where the L 's are one-dimensional shape functions: $L_1(\zeta) = \frac{1}{2}\zeta(\zeta - 1)$, $L_2(\zeta) = 1 - \zeta^2$, and $L_3(\zeta) = \frac{1}{2}\zeta(\zeta + 1)$. Let \mathbf{t} be the tangent to this edge curve. We have

$$\begin{aligned} t_x &= x_1 \frac{\partial L_1}{\partial \zeta} + x_2 \frac{\partial L_2}{\partial \zeta} + x_3 \frac{\partial L_3}{\partial \zeta} \\ &= x_1 \left(\zeta - \frac{1}{2}\right) + x_2 (-2\zeta) + x_3 \left(\zeta + \frac{1}{2}\right) \end{aligned} \quad (35)$$

and similarly for t_y and t_z . At the contact line (which coincides with node 1), we have $\zeta = -1$, and $z_1 = 0$:

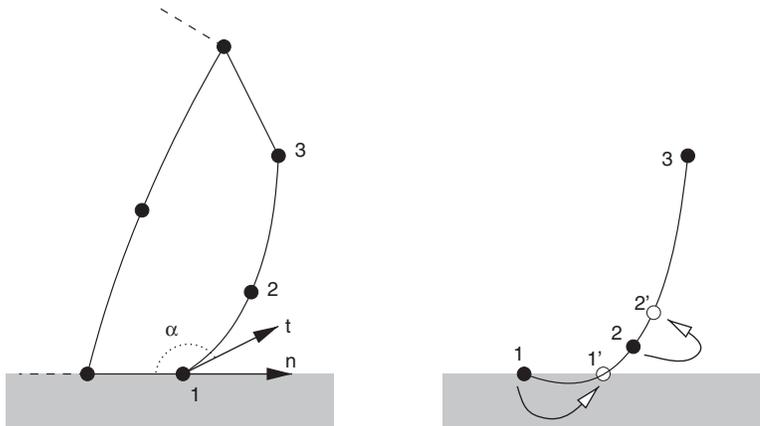


FIG. 3. Side view of the contact line. \mathbf{t} is the leading edge tangent at the contact point, α is the contact angle (left), and \mathbf{n} is the outward normal to the contact line. When the middle node 2 descends below one quarter of the height of the dorsal node 3, the contact angle become larger than 180° , and the leading edge effectively penetrates the substratum. Node 1 is then moved to position $1'$, and node 2 is moved up along the leading edge to one quarter of the height of the dorsal node 3 so as to restore a contact angle of 180° exactly.

$$\begin{aligned}
t_x &= -\frac{3}{2}x_1 + 2x_2 - \frac{1}{2}x_3 \\
t_y &= -\frac{3}{2}y_1 + 2y_2 - \frac{1}{2}y_3 \\
t_z &= 2z_2 - \frac{1}{2}z_3.
\end{aligned} \tag{36}$$

We now define \mathbf{n}_{CL} , the outward normal to the contact line. The contact angle α is then given by:

$$\begin{aligned}
&\text{if } \mathbf{n}_{CL} \cdot \mathbf{t} \geq 0 \text{ then} \\
&\quad \alpha = \pi - \arctan \frac{t_z}{(t_x^2 + t_y^2)^{1/2}} \\
&\text{else} \\
&\quad \alpha = \arctan \frac{t_z}{(t_x^2 + t_y^2)^{1/2}}.
\end{aligned} \tag{37}$$

It is clear that if $t_z < 0$ or equivalently $z_2 < \frac{1}{4}z_3$, the contact angle is $< 0^\circ$ (rare) or $> 180^\circ$ (common in spreading) and the contact line node needs to move.

The procedure is thus as follows (Fig. 3): if for an edge stack, we have $z_2 < \frac{1}{4}z_3$ this means that there exists $\zeta_1 > -1$ such that $z(\zeta_1) = 0$ (i.e., the edge curve crosses the substratum twice, once at $\zeta = -1$ which corresponds to node 1, and once at $\zeta = \zeta_1$). We therefore displace node 1 to position 1':

$$\begin{aligned}
x'_1 &= x_1 H_1(\zeta_1) + x_2 H_2(\zeta_1) + x_3 H_3(\zeta_1) \\
y'_1 &= y_1 H_1(\zeta_1) + y_2 H_2(\zeta_1) + y_3 H_3(\zeta_1) \\
z'_1 &= 0.
\end{aligned} \tag{38}$$

We also move 2 to 2' at the position determined by ζ_2 such that $z(\zeta_2) = \frac{1}{4}z_3$:

$$\begin{aligned}
x'_2 &= x_1 H_1(\zeta_2) + x_2 H_2(\zeta_2) + x_3 H_3(\zeta_2) \\
y'_2 &= y_1 H_1(\zeta_2) + y_2 H_2(\zeta_2) + y_3 H_3(\zeta_2) \\
z'_2 &= z_1 H_1(\zeta_2) + z_2 H_2(\zeta_2) + z_3 H_3(\zeta_2) \\
&= \frac{1}{4}z_3.
\end{aligned} \tag{39}$$

This returns the contact angle to 180° .

3.6. Step 2. Rezoning

1. The nodes are repositioned to optimize the mesh configuration without changing the boundaries.
 - (a) The nodes on the contact line are rezoned to be equidistant from one another. The procedure is described in Figure 4.
 - (b) The dorsal edge and middle edge nodes are shifted along their tangent planes so as to lie in the plane perpendicular to the substrate $z = 0$ and containing the contact line normal. The displacement keeps the elevation of the edge nodes above the substratum (z) constant and prevents distortions of the edge of the mesh.
 - (c) The interior ventral nodes are rezoned to minimize the two-dimensional Winslow functional (see below).
 - (d) The dorsal nodes and middle edge nodes are rezoned to minimize the two-dimensional Winslow functional (see below).
 - (e) The interior middle nodes are rezoned to minimize the three-dimensional TTM functional (see below).

The end-result of this sequence is a new rezoned mesh \mathbf{x}'_i .

2. The nodes belonging to the back surface of the rezoned mesh \mathbf{x}'_i are projected onto the surface of the advected mesh \mathbf{x}^d_i to obtain a "projected mesh" \mathbf{x}^p_i (for all other nodes, $\mathbf{x}^p_i = \mathbf{x}'_i$).
3. The final new mesh is obtained by a linear combination of the rezoned mesh and projected mesh:



FIG. 4. Accordion effect during rezoning. Left: top down view of three ventral contact line nodes labeled 1,2,3. Node 2 needs to move closer to node 3. Let m be the median between node 1 and 3, and \mathbf{t} the

volume tangent at node 2. A volume conserving rezone would move node 2 along \mathbf{t} towards the intersection with m (solid arrow). A boundary rezone would be to move node 2 along the boundary line (2,3), but this is not volume conserving. In practice we use a weighted average of the two (90% volume rezone and 10% boundary rezone). The boundary rezone contribution is to avoid an accordion instability as described on the right, where tangent node motion to even out inter-nodal distances only worsens the situation.

$$\mathbf{x}_i^f = 0.9 \mathbf{x}_i^r + 0.1 \mathbf{x}_i^p \quad (40)$$

The reason we do not use the rezoned mesh directly is to attempt to avoid the accordion effect described in Figure 4.

3.6.1. Rezoning of the ventral surface. Once the positions of the ventral edge nodes of the contact line have been set, the interior ventral nodes of the mesh (but not the edge ventral nodes which make up the contact line!) are rezoned to optimize the ventral mesh. Our general approach follows that described in Knupp and Steinberg (1993). It is variational in that the ventral nodes interior to the contact line are repositioned to minimize a functional integral I over the entire ventral surface:

$$I = \sum_{\text{ventral quads}} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta W \quad (41)$$

where W is the Winslow functional, a local measure of two-dimensional mesh distortion:

$$W = \frac{1}{\det(\mathbf{J}_{2D})} [g_{11} + g_{22}] \quad (42)$$

where g_{kl} are the components of the Riemann metric tensor of the ventral surface;

$$g_{kl} = \begin{pmatrix} \partial_k x \\ \partial_k y \end{pmatrix} \cdot \begin{pmatrix} \partial_l x \\ \partial_l y \end{pmatrix} \quad (43)$$

and the indices k and l run over the natural surface coordinates ξ and η . The determinant of the two-dimensional surface Jacobian $\det(\mathbf{J}_{2D})$ (see Eq. A-3) essentially corresponds to an elementary surface area (see Eq. 30) so that the functional is dimensionless. Details on the computation of W and derivatives are provided in the Appendix.

We minimize I by Newton's method. Let

$$\frac{\partial I}{\partial x_i} = I_{,x} = \sum_{\text{ventral quads}} \int \int d\xi d\eta W_{,x} \quad (44)$$

denote the derivative of I with respect to displacement δx of ventral node i . Note that in the sum, only the few quadrilaterals to which node i actually belongs make a nonzero contribution. An estimate of the displacements of Δx_i and Δy_i of node i that minimize I is given by the system

$$\begin{cases} I_{,xx} \Delta x_i + I_{,xy} \Delta y_i = -I_{,x}, \\ I_{,xy} \Delta x_i + I_{,yy} \Delta y_i = -I_{,y}. \end{cases} \quad (45)$$

In practice however, all the nodes are coupled so it is prudent to damp the motion while iterating to minimize I (we use a factor of 1/2 and further constrain the motion to 10% of the local element length scale).

3.6.2. Rezoning of the back surface. Rezoning of the back nodes follows the same principles as for the ventral surface. Once again, edge nodes that belong to the contact line are not moved, but there are some additional complexities from the fact that the surface is not planar and that we have to deal with dorsal quadrilaterals and edge hexalaterals. Thus, the functional integral I is now

$$I = \sum_{\text{dorsal quads}} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta W + \sum_{\text{edge hexalaterals}} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\zeta W \quad (46)$$

where, as before, W is the Winslow functional defined in Eq. 42 and the components of the metric tensor are:

$$g_{kl} = \begin{pmatrix} \partial_k x \\ \partial_k y \\ \partial_k z \end{pmatrix} \cdot \begin{pmatrix} \partial_l x \\ \partial_l y \\ \partial_l z \end{pmatrix} \quad (47)$$

where the indices k and l run over the natural surface coordinates (ξ, η for dorsal quadrilaterals and ξ, ζ for edge hexalaterals).

Minimizing I by Newton's method with respect to the displacement of node i involves a 3×3 system:

$$\begin{cases} I_{,xx}\Delta x_i + I_{,xy}\Delta y_i + I_{,xz}\Delta z_i = -I_{,x} \\ I_{,xy}\Delta x_i + I_{,yy}\Delta y_i + I_{,yz}\Delta z_i = -I_{,y} \\ I_{,xz}\Delta x_i + I_{,yz}\Delta y_i + I_{,zz}\Delta z_i = -I_{,z} \end{cases} \quad (48)$$

This time however we need to add constraints. First the displacement of a node must be restricted to the plane tangent to the surface, or more precisely, to the plane along which node motion preserves volume. Let \mathbf{N}_{Vi} be the normal to the volume tangent plane at a node i (see Appendix for the calculation of \mathbf{N}_{Vi} , Eq. A-13), the rezoning of node i must be in that plane and so must satisfy

$$\mathbf{N}_{Vi} \cdot \Delta \mathbf{x}_i = 0. \quad (49)$$

Second, to preserve the contact angle during the motion of the middle and dorsal edge nodes (here labeled 2 and 3) the displacement must satisfy:

$$\frac{\Delta z_2}{\Delta z_3} = \frac{z_2}{z_3}. \quad (50)$$

These two constraints can be implemented via the method of Lagrange multipliers.

3.6.3. Rezoning of the interior nodes. Once all the surface nodes have been rezoned, the middle nodes belonging to interior stacks need to be repositioned. This is also accomplished via a variational principle on a functional:

$$I = \sum_{\text{elements}} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta \int_{-1}^{+1} d\zeta T \quad (51)$$

where T is a three-dimensional extension of the Winslow functional generally known at the TTM functional Knupp and Steinberg (1993):

$$T = \frac{1}{\det(\mathbf{J})} [g_{11}g_{22} + g_{11}g_{33} + g_{22}g_{33} - g_{12}^2 - g_{13}^2 - g_{23}^2] \quad (52)$$

where the g_{kl} 's are as before (Eq. 47) and the indices k and l run over the intrinsic coordinates ξ, η , and ζ , and $\det(\mathbf{J})$ is simply the determinant of the three-dimensional Jacobian (Eq. 17).

As for surface nodes, we compute the derivatives of I with respect to the motion of individual nodes and minimize I using Newton's method and solving the system given by Eq. 48 (the necessary derivatives of T are provided in the Appendix). There are no constraints on the displacement of interior nodes.

Here an attentive reader may ask why not simply use the TTM functional for all nodes with appropriate constraints for the boundary nodes? We have attempted this and found it to lead to poor mesh structure at the edge of the mesh, an obviously critical region in studies of thin films.

3.7. Step 3. Material advection

We use a Lagrangian-Eulerian method to update the network concentrations θ_n from t to $t + \Delta t$. First comes the Lagrangian step which corresponds to the advection of the mesh with the flow of network (Eq. 33):

$$\theta_n^a = \theta_n^o - \Delta t \nabla \cdot \mathbf{v}_n \quad (53)$$

where θ_n^o is the original network concentration at a given node, and θ_n^a is the network concentration at that same node after the node has been advected with the network velocity. If one considers a reference volume V^o around the original node and advects it with the network flow to V^a , then obviously

$$\theta_n^a = \theta_n^o \frac{V^o}{V^a} \quad (54)$$

(which is the same as saying $\Delta t \nabla \cdot \mathbf{v}_n = (V^o - V^a)/V^o$). A given node belongs to several elements, and for each of these elements, it has a nearest Gauss point (each element has 12 nodes and 12 ‘‘matching’’ Gauss points). We choose as our reference volume the sum of the V_{GP} corresponding to the Gauss points associated with a node so that

$$\theta_n^a = \theta_n^o \frac{\sum V_{GP}^o}{\sum V_{GP}^a}, \quad (55)$$

where the volumes V_{GP} are computed for the original and advected mesh \mathbf{x}_i^o and \mathbf{x}_i^a through Eq. 27.

Second comes an Eulerian step which corresponds to the rezoning of the mesh from \mathbf{x}_i^a to \mathbf{x}_i^f . Here, the procedure is simply to interpolate θ_n^f from θ_n^a . Given an interior node in the final mesh, the element e_a of the advected mesh that contains it is determined along with the natural coordinates ξ_a, η_a, ζ_a of the node in e_a . We then have

$$\theta_n^f = \sum_{12 \text{ nodes} \in e_a} \theta_n^a(\text{node}) H_{\text{node}}(\xi_a, \eta_a, \zeta_a). \quad (56)$$

For boundary nodes in the final mesh, the closest point on the surface of the advected mesh is determined with natural coordinates ξ_a, η_a or ξ_a, ζ_a as the case may be. We then have:

$$\theta_n^f = \sum_{4/6 \text{ nodes} \in s_a} \theta_n^a(\text{node}) S_{\text{node}}(\xi_a, \eta_a/\zeta_a). \quad (57)$$

It is possible to advect chemical species in the solvent phase by using the same procedure as outlined above, except that the advected mesh is now the result of node motion according to the solvent flow field \mathbf{v}_s . However, we have found that advection of chemical species in the solvent (cytosolic) phase is most often negligible compared to diffusion (i.e., the Peclet number is small) so that usually, the only important step is interpolation from the original to the final mesh.

3.8. Step 4. Diffusion and reaction

Diffusion-reaction problems are standard fare and we will limit ourselves to a brief outline of an implicit backward Euler scheme with finite element treatment of spatial derivatives and boundary conditions. For a chemical species c , we need to evolve

$$\frac{\partial c}{\partial t} = \nabla \cdot D \nabla c + \dot{c} \quad (58)$$

between t and $t + \Delta t$, where D is the diffusion coefficient, and \dot{c} is the rate at which the species is created by chemical reaction. We will assume that at time t , we know D , c^o , \dot{c}^o , and $\partial \dot{c}^o / \partial c$ (evaluated via appropriate constitutive laws when necessary). Discretizing in time, we can then write:

$$\frac{c - c^o}{\Delta t} = \nabla \cdot D \nabla c + \dot{c}^o + \frac{\partial \dot{c}^o}{\partial c} (c - c^o), \quad (59)$$

where c is the species concentration at the end of the time step. Rearranging the terms we obtain

$$\left(1 - \Delta t \frac{\partial \dot{c}^o}{\partial c} - \Delta t \nabla \cdot D \nabla\right) c = c^o + \Delta t \dot{c}^o - \Delta t \frac{\partial \dot{c}^o}{\partial c} c^o. \quad (60)$$

Following the canonical Galerkin finite element approach—which is beyond the scope of this work and is described in standard texts (Hughes, 2000)—Eq. 60 can be recast in a weak (variational) form while simultaneously expanding c over a set of trial functions (the shape functions). This leads to a linear system determining the vector c_i where i runs over the all the nodes (and corresponding shape functions):

$$\mathbf{Q} \mathbf{c} = \mathbf{f} \quad (61)$$

where \mathbf{Q} is a square “stiffness” matrix, and \mathbf{f} is a “load” or “force” vector. Two nodes i and j belonging to the same element e contribute to the stiffness matrix and to the load vector via their associated shape functions H_i and H_j :

$$Q_{ij,e} = \left(1 - \Delta t \frac{\partial \dot{c}^o}{\partial c}\right) \int_e dV H_i H_j + \Delta t D \int_e dV \nabla H_i \cdot \nabla H_j \quad (62)$$

and

$$f_{j,e} = \sum_{i \in e} \int_e dV H_i H_j \left(c_i^o + \Delta t \dot{c}_i^o - \Delta t \frac{\partial \dot{c}_i^o}{\partial c} c_i^o \right) \quad (63)$$

So that in the end $Q_{ij} = \sum_e Q_{ij,e}$ and $f_j = \sum_e f_{j,e}$.

Dirichlet and Neumann boundary conditions lead to modifications of the diffusion equation Eq. 58 that look like:

$$\frac{\partial c}{\partial t} = \dots + \Pi(c - c_{\text{ext}}) + \dot{s} \quad (64)$$

where c_{ext} is an external reference concentration and Π is an effective permeability (the higher Π , the stricter the Dirichlet boundary condition $c = c_{\text{ext}}$), and where \dot{s} is a source term corresponding to a Neumann boundary condition. For a Dirichlet condition, both the stiffness matrix and the load vector must be modified:

$$Q_{ij,e} \rightarrow Q_{ij,e} + \Pi_e \int_{e4/e6} d\sigma S_i S_j \quad (65)$$

$$f_{j,e} \rightarrow f_{j,e} + \Delta t c_{\text{ext}} \Pi_e \int_{e4/e6} d\sigma S_j. \quad (66)$$

To take into account an external chemical gradient, one can easily make c_{ext} a function of coordinates (e.g., to model chemotaxis).

For a Neumann condition, only the load vector changes

$$f_{j,e} \rightarrow f_{j,e} + \Delta t \int_{e4/e6} d\sigma \dot{s}_j S_j. \quad (67)$$

In practice, we solve Eq. 61 through the conjugate gradient method with preconditioning (simple diagonal matrix of $(1/Q_{ii})$'s), although for problems with a small number of nodes, direct LU decomposition is feasible.

Two-dimensional surface diffusion is implemented in a similar manner except that elements are now quadrilaterals and hexalaterals, surface shape functions are used instead of volume shape functions, and boundary conditions apply to lines (e.g., the contact line) instead of surfaces.

3.9. Step 5. Momentum transport

Because of the multiphase nature of the flow (one has to solve the triplet \mathbf{v}_n , \mathbf{v}_s , and P rather than just for \mathbf{v} and P) some modifications are required compared to the usual single phase viscous flow finite element treatment.

Recall the solvent and network momentum equations

$$-\theta_s \nabla P + \mathcal{H} \theta_s \theta_n (\mathbf{v}_n - \mathbf{v}_s) = 0 \quad (68)$$

and

$$-\theta_n \nabla P - \mathcal{H} \theta_s \theta_n (\mathbf{v}_n - \mathbf{v}_s) + \nabla \cdot [\nu (\nabla \mathbf{v}_n + (\nabla \mathbf{v}_n)^T)] - \nabla \cdot \Psi + \mathbf{F}_{\text{ext}} = 0. \quad (69)$$

By adding the solvent and network momentum equation together, \mathbf{v}_s can be eliminated to obtain a “bulk” cytoplasmic momentum equation

$$-\nabla \cdot [\nu (\nabla \mathbf{v}_n + (\nabla \mathbf{v}_n)^T)] = -\nabla \cdot \Psi - \nabla P + \mathbf{F}_{\text{ext}} \quad (70)$$

which we call the “velocity equation.” The boundary conditions can be free or contact. For free membrane, stress continuity requires

$$[\nu (\nabla \mathbf{v}_n + (\nabla \mathbf{v}_n)^T)] \cdot \mathbf{n} - \Psi \cdot \mathbf{n} - P \mathbf{n} = -2\gamma \kappa \mathbf{n} - P_{\text{ext}} \mathbf{n} + \sigma \quad (71)$$

where P_{ext} is the external pressure, γ and κ the surface tension and mean curvature, and σ is the boundary traction. For contact boundaries, Dirichlet conditions are imposed on the velocity.

The solvent momentum equation (Eq. 68) gives an expression for \mathbf{v}_s :

$$\mathbf{v}_s = \mathbf{v}_n - \frac{\nabla P}{\mathcal{H} \theta_n} \quad (72)$$

which can then be substituted in the incompressibility condition to yield:

$$\nabla \cdot \frac{1}{\mathcal{H} \theta_n} \nabla P = \nabla \cdot \mathbf{v}_n \quad (73)$$

or, if we relabel $\theta_s / (\mathcal{H} \theta_n) = 1/\Phi$

$$\nabla \cdot \frac{\nabla P}{\Phi} = \nabla \cdot \mathbf{v}_n \quad (74)$$

which we call the “pressure equation.” In situations where there is zero membrane permeability (i.e., no trans-membrane solvent flow), the boundary condition simplifies to:

$$\nabla P = 0. \quad (75)$$

The general strategy to obtain a solution follows Uzawa’s algorithm: an initial guess for the pressure field (which is usually good since obtained from the previous time step) allows the computation of the network velocity field by Eq. 70. This velocity field can then be used to update the pressure field by Eq. 74, and so on through iterations between the two equations. Once the network velocity field \mathbf{v}_n and pressure field P have converged to a self-consistent solution, the solvent velocity field \mathbf{v}_s can be trivially extracted through the use of Eq. 72 with automatic enforcement of the incompressibility condition.

To solve the velocity equation (Eq. 70), we wish to compute the vector \mathbf{u} which assembles the three velocity components of every node (i.e., \mathbf{u} has $3N$ components where N is the number of nodes):

$$\mathbf{K} \mathbf{u} = \mathbf{f} \quad (76)$$

where \mathbf{K} is a square stiffness matrix, and \mathbf{f} is usually known as the “force” vector. Expressions for the contributions by individual elements to the stiffness matrix and the force vector can be derived from Eq. 70. If i and j are two nodes that belong to element e , they make the following contributions to the stiffness matrix:

$$K_{ijx,e} = \int_e dV \nu \left(2 \frac{\partial H_i}{\partial x} \frac{\partial H_j}{\partial x} + \frac{\partial H_i}{\partial y} \frac{\partial H_j}{\partial y} + \frac{\partial H_i}{\partial z} \frac{\partial H_j}{\partial z} \right) \quad (77)$$

$$K_{ixjy,e} = \int_e dV \nu \frac{\partial H_i}{\partial x} \frac{\partial H_j}{\partial y} \quad (78)$$

and to the load vector

$$f_{jx,e} = \sum_{i \in e} \int_e dV (\Psi \cdot \nabla H_i)_x + \sum_{i \in e} \int_e dV P \frac{\partial H_i}{\partial x} + \sum_{i \in e} \int_e dV F_{\text{ext}}^x H_j \quad (79)$$

where the last term really amounts to $V_e F_{\text{ext}}^x$, where V_e is the volume of the element. In the end, the components of \mathbf{K} and \mathbf{u} are given by $K_{ijx} = \sum_e K_{ijx,e}$ and $f_{jx} = \sum_e f_{jx,e}$.

Once new nodal network velocities have been estimated for a given guess of the pressure field P^o (obtained from the previous cycle of iteration between velocity and pressure equations, or from the prior time step if this is the first pass), we solve the pressure equation (Eq. 74) to obtain a new pressure field P . Unfortunately, direct solution of Eq. 74 leads to severe instabilities so that the following damped evolution equation is solved instead:

$$r(P - P^o) = \nabla \cdot \left[\frac{\nabla P}{\Phi} \right] - \nabla \cdot \mathbf{v}_n \quad (80)$$

where $r > 0$ is a relaxation coefficient. Note that since, within a given time step, we iterate back and forth between the velocity and pressure equations until the velocity and pressure fields have converged to a stable value, the LHS of Eq. 80 tends to 0 (P and P^o will be the same) as we get closer to the solution so that in the final analysis, it is Eq. 74 that is solved. The relaxation coefficient is taken to be proportional to the perturbation of the velocity field by a perturbation in pressure:

$$r = \beta \left| \frac{\delta(\nabla \cdot \mathbf{v}_n)}{\delta(P)} \right|. \quad (81)$$

In the code, r is evaluated numerically using the velocity equation (Eq. 70) and β is set to 10^2 for small Φ (typical cytoplasmic condition) and 10^3 for large Φ (single phase viscous flow condition). We can rewrite Eq. 80 as:

$$\left[r - \nabla \cdot \frac{\nabla}{\Phi} \right] P = rP^o - \nabla \cdot \mathbf{v}_n \quad (82)$$

which is to be cast by the finite element approach into a linear system of equations

$$\mathbf{Q} \mathbf{P} = \mathbf{g} \quad (83)$$

where \mathbf{Q} is a square stiffness matrix, \mathbf{P} is the vector of nodal pressures that we seek, and \mathbf{g} is a load vector. Individual element contributions from element e to \mathbf{Q} and \mathbf{g} are given:

$$Q_{ij,e} = \int_e dV \left(rH_i H_j - \frac{1}{\Phi} \nabla H_i \cdot \nabla H_j \right) \quad (84)$$

and

$$g_{j,e} = \int_e dV (rP^o H_j - \mathbf{v}_n \cdot \nabla H_j) \quad (85)$$

so that $Q_{ij} = \sum_e Q_{ij,e}$ and $g_j = \sum_e g_{j,e}$.

Boundary conditions to these equations can take a multiplicity of forms that cannot all be covered here. The most important cases are the contact velocity boundary conditions on the ventral surface (substratum), and the contribution of the surface tension to free boundary dynamics.

For instance, if the x component of \mathbf{v}_n must be constrained at a surface containing node i , as would be the case for a ventral quadrilateral with a no-slip velocity boundary condition at the substratum, we add a diagonal term to the velocity equation stiffness matrix;

$$K_{ixi,e} \rightarrow K_{ixi,e} + \nu_p \int_{e4/e6} d\sigma S_i \quad (86)$$

where $\nu_p = 10^7 \bar{\nu}$ is a ‘‘penalty’’ viscosity set many orders of magnitude greater than the average problem viscosity $\bar{\nu}$. If a viscous drag between network and substrate is desired instead of a no-slip condition this is easily implemented by the use of an appropriate viscosity coefficient.

The load on a given surface node due to the surface tension γ is $-\nabla_i(\gamma A)$, where A is the surface area and where the gradient is taken with respect to motion of node i . We therefore add a term to the velocity equation vector load:

$$f_{j,e} \rightarrow f_{j,e} - \nabla_i(\gamma A_e). \quad (87)$$

Details on how to compute this gradient are given in the Appendix.

In practice, we solve both equations through the conjugate gradient method with preconditioning (for the pressure equation with the diagonal matrix $(1/Q_{ii})$, for the velocity equation with the block diagonal matrix obtained by local inversions of the 3×3 nodal stiffness matrices (K_{ii}^{-1})). Convergence is judged to be sufficient for relative changes $< 10^{-4}$ in pressure and velocities. For an ongoing simulation where the initial guess is close to the solution, this typically occurs within 10–20 iterations, but for a new problem, it may take several hundred iterations. Even when convergence is rapid, solving the momentum equations is almost always the most computationally intensive part of the simulation.

4. TESTING: THE SESSILE DROP

Consider a hemispheric drop with radius and height $a_0 = h_0 = 1$, viscosity $\nu = 10^3$, and surface tension $\gamma = 1$ sitting in equilibrium on a non-wetting surface with no-slip boundary condition ($\mathbf{v} = 0$). At time $t = 0$, “gravity” is turned on in the form of an external body force $F_z = -10^2$ so that the Bond number becomes large: $B = |F_z|a_0^2/\gamma = 100$. The drop flattens, and after the contact angle becomes 180° (the substratum is non-wetting), the contact line begins to advance. Eventually, the gravitational work gained by further flattening is balanced by the surface tension work of area expansion, and the sessile drop settles in a new equilibrium shape, still with contact angle 180° . Note that we do not give units—the reader is free to assume cgs or SI or any other self-consistent units.

We have chosen this problem as a benchmark for several reasons. First it is simply posed and simply explained. Second, it has received attention for more than a century (Bashforth and Adams, 1883), and although it cannot be solved analytically, several useful approximations are available. Third, because of the large Bond number, it leads to a very flat drop with an aspect ratio ~ 20 : this is representative of our intended application of Cytopede to model flattened cells as thin films. Fourth, it has cylindrical symmetry which allows comparisons with reference two-dimensional numerical simulations performed with another finite element code that has been previously validated (Dembo, 1994; Drury and Dembo, 1999).

This single phase flow application requires phase locking between the solvent and the network, which can be achieved by setting a large network-solvent hydraulic resistance \mathcal{H} (see Eq. 68). We use

$$\mathcal{H} = \frac{\nu}{L^2} \quad (88)$$

where L is the local length scale of an element which we have found to be a safe choice, large enough so that network-solvent slippage is insignificant, but not so big so as to lead to the spurious pressure modes that appear in finite element calculations when the Babuska-Brezzi condition is violated (Drury and Dembo, 1999).

The results of the simulations are presented in Table 1 and in Figures 5–7. It is apparent that both two- and three-dimensional calculations have converged numerically. For the two-dimensional calculations, a low resolution simulation with 641 nodes (562 elements) gives essentially the same result as a high resolution simulation with 2405 nodes (2248 elements). For the three-dimensional calculations, a simulation with 1035 nodes (308 elements) gives essentially the same result as a simulation with 3051 nodes (944 elements).

The equilibrium state ($t \rightarrow \infty$ in the present set up) has been studied extensively and expression for the ultimate height and radius derived: $h_\infty = 2(\gamma/F_z)^{1/2} = 0.2$; (although there can be a potential correction factor $< 5\%$ as is noted in Padday and Pitt (1972), and $a_\infty^2 = (V/2\pi)(F_z/\gamma)^{1/2}$ (Rienstra, 1990) so that $a_\infty = 1.83$). The three-dimensional (and two-dimensional) simulations converge asymptotically to similar limits, with a very slightly larger drop size in the 3D simulation which is in part due to a small upward drift in volume ($\sim 2\%$) over ~ 5000 time steps of the calculation (Table 1).

Looking at the dynamic evolution from the initial configuration toward the equilibrium state, analytic guidance is unfortunately restricted to an investigation of the dynamic motion of the contact line before reaching equilibrium which gives $\dot{a} \propto a^{1/7}$ as an approximation (Hocking, 1983); this is close to our

TABLE 1. THE SPREADING DROP: THEORY VERSUS NUMERICAL EXPERIMENTS

	<i>Theory</i>	<i>2D calculation</i>	<i>3D calculation</i>
V_0	2.094	2.085	2.084
V_∞	2.094	2.078	2.128
h_∞	0.2	0.203	0.205
a_∞	1.83	1.82	1.86

V , h , and a are, respectively, the drop volume, maximum height, and average radius of the contact line. For the numerical simulations, $t = 10,000$ was taken as t_∞ . Simulation results in 2D and 3D correspond to the high-resolution calculations.

calculations. For further insights, we are therefore limited to a comparison between the two-dimensional and three-dimensional simulations. In both the two-dimensional and three-dimensional calculations, the contact line radius a initially equal to $a_0 = 1$ begins to change at $t \sim 18$; this is the time it takes for the contact angle to increase from 90° at $t = 0$ to 180° . There is however an early deviation between the time-dependence of the height and contact line radius as computed by the two-dimensional and three-dimensional codes. Close inspection reveals that, in the early evolution, the height of the drop decreases much faster in the three-dimensional calculation than in the two-dimensional calculation (Fig. 6). This inaccurate velocity solution in the three-dimensional calculation is due to limited vertical resolution using only 3 node (one element) interpolation—we are here far from the thin film regime with an initial aspect ratio of only 2. Because mass initially flows faster from the top to the sides of the drop, the radius of the contact line a takes an early lead in the three-dimensional compared to the two-dimensional calculation. As the drop thins, the three-dimensional calculation becomes more accurate, but the shift in the contact line evolution subsists. For instance, the drop profile obtained at $t = 100$ with a three-dimensional simulation is significantly different from that obtained with the two-dimensional calculation at the same time, but exactly matches the two-dimensional solution at $t = 160$ (Fig. 7). Improvements to the early part of the three-dimensional calculation would require additional layers of elements. However, in most conditions that pertain to cells on a substratum, aspect ratios are large and the three-dimensional code should perform adequately.

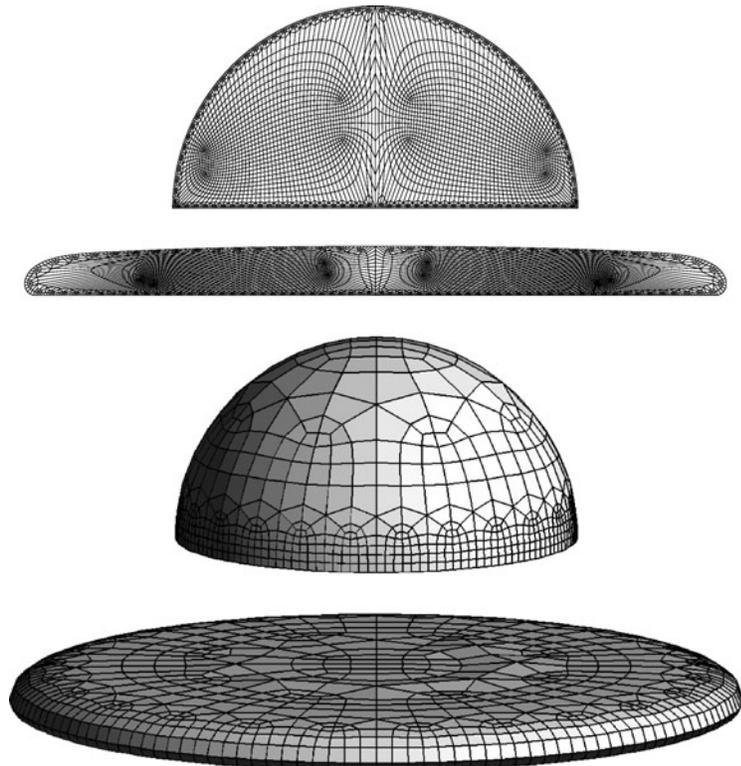


FIG. 5. Meshes for two- and three-dimensional high-resolution simulations at time $t = 0$ and $t = 2000$. Note that in the two-dimensional simulation the actual mesh used in the computation is half of what is shown. Note also the three-dimensional mesh consists of a dorsal layer (shown), a ventral layer, and an intermediate layer (not shown).

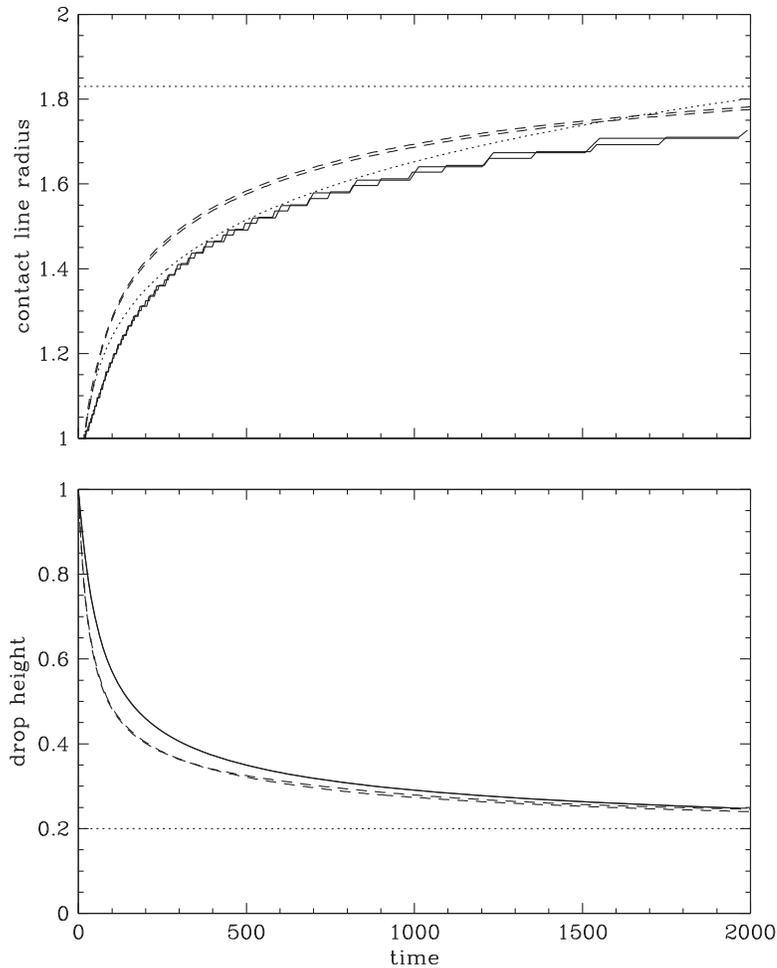


FIG. 6. Evolution of the contact line radius a (top) and the drop height h (bottom) with time. Dotted lines represent theoretical estimates, $a_\infty = 1.83$, $a \propto a^7$ and $h_\infty = 0.2$ (see text). Dashed lines represent output from the low- and high-resolution three-dimensional simulations. Solid lines represent output from the low- and high-resolution two-dimensional simulations (indistinguishable for h). The step-like nature of the a curve for two-dimensional simulations comes from the fact that the contact line advances by a discrete jump when a new element comes into contact with the substratum.

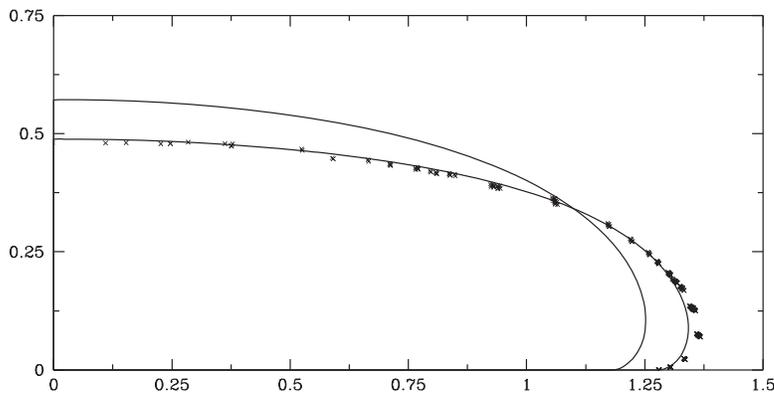


FIG. 7. Profile of the surface of the drop. Points correspond to the three-dimensional high-resolution simulation at $t = 100$. Solid lines correspond to the two-dimensional high-resolution simulation at $t = 100$ and $t = 165$.

5. APPLICATION: A MODEL OF THE FIBROBLAST

We now turn to the intended purpose of Cytopede—the simulation of locomoting cells, and present a basic cytomechanical model consisting in a set of rules (constitutive relations and boundary conditions) that leads to fibroblast-like behavior. This example is then used to show how contact with quantitative experimental observations may be achieved. Note that rather than striving for verisimilitude, we have attempted to develop a pedagogical and informative scenario. Nevertheless, while this model represents a gross simplification of what must be a very complicated collection of physico-chemical processes in a real cell, it still represents a large increase in complexity over the simple spreading drop.

5.1. The model

The key parameters used in the model are listed in Table 2 and fall into two broad categories: diffusion-reaction parameters that determine in space and time the biochemistry of the cytoskeleton, and mechanical parameters that determine the dynamical behavior of the cytoskeleton. Choices for many of these parameters were inspired from previous two-dimensional models of neutrophils (Herant et al., 2003, 2005, 2006).

5.1.1. Biochemical parameters. The baseline volume fraction of network θ_n is set to $\theta_0 = 10^{-3}$, but must increase several fold near the leading edge (up to $\theta_n \sim 2 \times 10^{-2}$). To achieve this, we set (see Eq. 3)

$$J = \frac{\theta_{\text{eq}} - \theta_n}{\tau_n} \times \max(m, 1) \quad (89)$$

where θ_{eq} is the local equilibrium network concentration, $\tau_n = 20$ s is a network turnover or equilibration timescale modulated by the local (dimensionless) concentration m of a polymerization messenger:

$$\theta_{\text{eq}} = \theta_0(1 + m). \quad (90)$$

The polymerization messenger species is generated at activated portions of the plasma membrane and diffuses into the cytoplasm with diffusion coefficient $D_m = 10^{-8}$ cm² s⁻¹ and lifetime $\tau_m = 1$ s:

$$\frac{\partial m}{\partial t} = -\frac{m}{\tau_m} + D_m \nabla^2 m \quad (91)$$

where we neglect advection by the local flow (the Peclet number is small). The Neumann boundary condition at the membrane is

$$\mathbf{n} \cdot \nabla m = \frac{\epsilon_m}{D_m} \quad (92)$$

where ϵ_m is the local emissivity of the messenger. In the simulations, ϵ_m is set to be maximum near activated portions of the contact line and zero further in, so that $m \sim 20$ near the leading edge, and rapidly decays into the cytoplasm over the penetration depth $d_m = (D_m \tau_m)^{1/2} = 1$ μm .

TABLE 2. PARAMETERS USED FOR THE MODEL FIBROBLAST

Parameters	Symbol	Value
Cytoplasmic volume	V_c	$1080 \mu\text{m}^3$
Biochemical parameters		
Baseline network density	θ_0	10^{-3}
Network turnover time	τ_n	20 s
Messenger concentration	m	—
Equilibrium network	$\theta_{\text{eq}} = \theta_0(1 + m)$	—
Messenger diffusion coefficient	D_m	$1 \mu\text{m}^2 \text{s}^{-1}$
Messenger decay time	τ_m	1 s
Messenger penetration depth	$d_m \sqrt{D_m \tau_m}$	1 μm
Messenger emissivity ^a	ϵ_m	$0-20 \mu\text{m} \text{s}^{-1}$
Mechanical parameters		
Specific network viscosity ^b	ν_0	$1 \times 10^5 \text{ Pa s}$
Disjoining force strength ^c	$\psi_0^{nM} d_M m \theta_n$	$(10 \times m \theta_n) \text{ mN m}^{-1}$
Flattening force strength ^c	$\psi_0^{ns} d_s \epsilon_m$	$(-400 \times \epsilon_m) \text{ mN m}^{-1}$
Network-solvent drag	\mathcal{H}	$100 \text{ pN s } \mu\text{m}^{-4}$
Slip contact angle	α_s	60°
Surface Tension	$\gamma_0 (A/A_0)^3$	$0.01 (A/A_0)^3 \text{ mN m}^{-1}$

^aUnits are length/time instead of 1/(area time) because m is dimensionless instead of 1/volume.

^bBaseline viscosity is thus $\theta_0 \nu_0 = 100 \text{ Pa s}$.

^cThe dynamically relevant term is the network-membrane potential energy times its range d_M or d_s .

The messenger is of course just a way to encode positional information as a driver of cytoskeletal chemistry and activity. In the real cell, a host of biochemical intermediates perform this task, and so, this simple “messenger” is not intended to represent any single molecule.

5.1.2. Mechanical parameters. We assume that the viscosity of the cytoskeleton is proportional to its density so that in the momentum equation (Eq. 6):

$$\nu = \nu_0 \theta_n \quad (93)$$

with $\nu_0 = 10^5$ Pa s, so that the baseline viscosity is $\nu_0 \theta_0 = 100$ Pa s, and reaching maximal values of 1000 Pa s at the leading edge.

To drive protrusion, we implement a network-membrane repulsive stress term which has the form:

$$\Psi^{nM} = \psi_0^{nM} m \theta_n \mathbf{nn} \quad (94)$$

where \mathbf{nn} is the dyadic of the unit vector outward normal to the membrane. For $\psi_0^{nM} > 0$, a normal stress pushes the membrane outward and the network experiences an equal and opposite reaction driving it inward, so that this stress corresponds to a disjoining force that expands the cortical network layer. In a polymerization force model such as is used here, there is a linear dependence on the local polymerization rate (driven by the messenger concentration m) and the network density θ_n . Note that ψ_0^{nM} is set to zero at the ventral surface of the cell because the network is assumed to be anchored to the substratum.

The actual relevant dynamical parameter is the stress times its range d_M away from the membrane (this is a somewhat subtle point; see discussions in Herant et al., 2003; Herant and Dembo, 2006). With $\psi_0^{nM} d_M = 10$ mN m⁻¹, the messenger concentration at the leading edge $m \sim 20$, and taking $d_M = 1$ μ m, the stress energy within the layer is ~ 1.8 k_BT per actin monomer. The physical origin of this stress is left unspecified but may involve buckling of filaments against the membrane (Kovar and Pollard, 2004), long-range electrostatic interactions, or entropic constraints, which transform the chemical energy of polymerization into a stress capable of producing mechanical work.

To prevent the protrusive stress from bulging the membrane upward rather than forward, it is necessary to postulate a compensatory force directed down toward the substratum, or else to assume a planar geometry of the cytoskeleton that only allows growth in the horizontal direction (so that most of the z terms in the stress-tensor vanish), something we find implausible. We therefore implement a network-substratum attractive stress term which has the form

$$\Psi^{ns} = \psi_0^{ns} \epsilon_m \mathbf{nn} \quad (95)$$

where \mathbf{nn} is the dyadic of the unit vector downward normal to the substratum. For $\psi_0^{ns} < 0$, the stress pulls the network down towards the substratum. Again the dynamically relevant quantity is the stress times its range d_s (Herant and Dembo, 2006). With $\psi_0^{ns} d_s \epsilon_m = -0.8$ mN m⁻¹ at the leading edge ($\epsilon_m \sim 2 \cdot 10^{-3}$ cm s⁻¹), and assuming $d_s = 1$ μ m, this leads to a maximum downward force at the leading edge of 800 pN per μ m² of ventral membrane. Again, the physical origin of this force is left unspecified, but may involve molecular motors such as unconventional myosins which are often detected at the leading edge of locomoting cells (Fukui et al., 1989; Yonemura and Pollard, 1992; Wagner et al., 1992; Sousa and Cheney, 2005).

We finally note that aside from the flattening force described above, this model of the fibroblast does not include contractile elements of the kind that might represent the activity of myosin II bundles often detected in the cell body. Such components could readily be incorporated but for the sake of simplicity, we do not do so here.

5.1.3. Boundary conditions. We assume impermeability of the plasma membrane to both cytosol and cytoskeleton (Eqs. 8, 9) so that total volume is conserved. For the ventral interface in contact with the substratum, the boundary condition on the network velocity is no slip ($\mathbf{v}_n = 0$): we postulate that adhesion via transmembrane proteins locally immobilizes the cytoskeleton rigidly (for simplicity, viscous slippage is not considered but could be implemented easily to match behaviors of the type described by Leibler and Huse 1993). Thus, when new membrane at the leading edge comes into contact with the substratum we assume immediate adhesion. The only exception is the trailing edge of the cell where we allow movement when the contact angle decreases to less than 60°.

For the back surface of the cell, the motion of the free boundary is determined per Eq. 10 with the surface tension

$$\gamma = \gamma_0 \left(\frac{A}{A_0} \right)^3 \quad (96)$$

where A is the total surface area of the cell, A_0 is the area of a volume-equivalent sphere (a constant since volume is conserved), and $\gamma_0 = 0.01 \text{ mN m}^{-1}$ is a baseline surface tension. We have found that the exact form does not matter as long as the tension rises steeply with area; what counts is that the tension become sufficiently strong to limit cell spreading to a plausible extension. In the simulations, once the cell is spread, $A/A_0 \sim 2\text{--}3$ so that $\gamma \sim 0.1\text{--}0.4 \text{ mN m}^{-1}$.

5.2. The simulation

The initial condition is that of a hemispherical cell with cytoplasmic volume $V_c = 1080 \mu\text{m}^3$ (a reasonable estimate of fibroblastic volume without nucleus (Uhal et al., 1998)). The entire cell circumference is activated so that protrusion due to network polymerization drives the contact line out. As the cell flattens (Fig. 8), the surface area increases and so does surface tension (Eq. 96) thus resisting further extension. After ~ 100 s, the cell takes on a stable disk shape with diameter of $\sim 33 \mu\text{m}$, and height of $\sim 1.5 \mu\text{m}$. The surface tension at which protrusion stalls is $\sim 0.34 \text{ mN m}^{-1}$ and can be readily identified with a force of protrusion of $\sim 700 \text{ pN}$ per μm of leading edge (the doubling is due to the fact that both dorsal and ventral folds of membrane are pulling back).

From this equilibrium disk shape, we break the symmetry by abrogating contact line activation over 85% of the circumference. Messenger production and therefore network polymerization continues over the remaining 15% of the contact line. Note that as the overall perimeter length of the cell changes because of deformation of the contact line, activation is maintained over 15% of that length. In addition, when the contact angle becomes $< 60^\circ$, we force a return of the contact angle to 60° by shifting the contact line which enables retraction of the trailing edge.

As a result of these changes, a leading edge stretches out ahead while the remainder of the perimeter slowly retracts and forms a tail (Fig. 8). Eventually, a longitudinally elongated, approximately triangular cell emerges with length of $\sim 40 \mu\text{m}$ and leading edge migration velocity of order $0.06 \mu\text{m s}^{-1}$.

Because the shape of the model fibroblast is not stable, distortion eventually grows beyond the abilities of the rezoning algorithm to maintain an adequate mesh. For this particular model, this occurs at $t = 600$ s, by which time the code has taken $\sim 7,000$ time steps. The calculation takes approximately 3 hours on a PC workstation (running on a single CPU).

5.3. Contact with experimental data sets

Four principal aspects of the mechanical characteristics of cells are experimentally accessible and available in the general literature.

1. The basic contour and motion of the cell,
2. The thickness h of the cell as a function of position x , y and time t ,
3. The cytoskeletal density, flow, and turnover,
4. The traction forces exerted by the cell on the substratum.

In our past experience constructing cytomechanical models (He and Dembo, 1997; Herant et al., 2003, 2006), we have found that while it is relatively easy to fit the data from a single type of experiment, dealing with two data sets is much harder, and matching more is essentially impossible unless and until one is able to develop insights into the fundamental processes that determine the behavior under study. The development of such, yet to be determined, insights in the mechanics of locomoting cells is therefore the principal motivation for wanting to elaborate models that satisfy (1), (2), (3), and (4) simultaneously.

Using the model fibroblast as an example, we show in the following sections how the output of Cytopede can be compared to each of the four data sets listed above. Note that the objective here is not to attempt to create an optimal model of the fibroblast, but rather to discuss how the strengths and weaknesses of a given cytomechanical model may be interpreted in light of experimental observations.

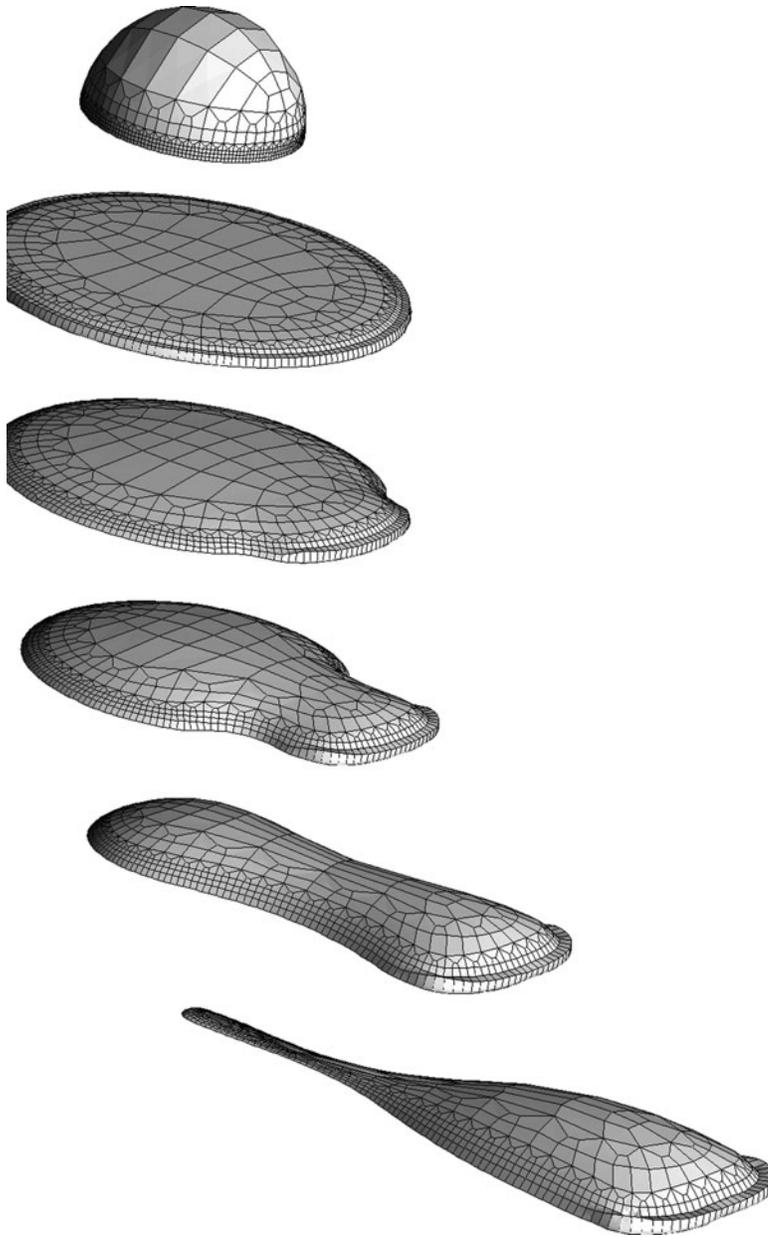


FIG. 8. Time series of evolution of a fibroblast model. Hemispherical cell ($t = -100$ s) flattened to an equilibrium disc shape ($t = 0$ s) by activation of the entire contact line. After 85% of the contact line is rendered inactive, an elongated fibroblast like shape emerges ($t = 20, 100, 300, 600$ s).

5.3.1. Contour and motion of the cell. Determination of the contour and motion of the cell only requires a time-series of microscopic images and is therefore almost always available. It is also a direct output of Cytopede simulations (Figs. 8 and 10).

Although the model fibroblast eventually takes on a generally triangular shape, this shape is somewhat unstable and is in addition sensitive to small perturbations in the conditions of the simulations. This is mainly due to conditions at the rear: due to slow cytoskeletal disassembly, a viscous plug of swept up cytoskeleton develops at the rear of the model fibroblast causing the growth of a tail (Fig. 10). Feedback between rear detachment and frontal progression is provided by the surface tension. However, because of its elongated nature, the tail is vulnerable to the dripping faucet instability whereby surface tension causes pinching.

In real fibroblasts several behaviors of the tail are observed (Chen, 1981a,b). Fibroblasts may repeatedly leave part of their tails behind as they advance over the substrate in a way similar to what would occur in our model if the mesh were able to follow through such a change in topology. Alternatively, the tail may

undergo a major detachment event with abrogation of adhesion over a large area (this is interpretable as a catastrophic chain-reaction by which failure of a few bonds redistributes the load to the remaining bonds causing further failures, etc.). Such detachment allows retraction of the tail in the cell body and could be modeled in a straightforward manner. Finally, what is also frequently observed is a filiform tail which is fairly stable, and which may represent a configuration where the cytoskeleton forms a static bundle of cables.

5.3.2. Thickness of the cell. Thickness maps of the cell impose important constraints on mechanical models but unfortunately, experimental data are somewhat sparse, and mostly based on confocal microscopy which has limited resolution in height. An alternative way to determine thickness is to measure fluorescence intensity of a freely diffusible fluorophore which essentially acts as a volume marker. Unless somehow calibrated, this only yields relative thickness, but can still be useful, especially coupled with an estimate of the total cell volume.

A map of the thickness of the fibroblast model is shown in Figure 9 and exposes one of its shortcomings: it does not generate a thin lamellipodium at the leading edge. Instead of a 0.1–0.2- μm -thick ruffling membrane, the model shows a height of 0.5–1.0 μm very close to the contact line. We note however that recent evidence has shown that at least in some situations, the lamellipodium is unnecessary for locomotion and that its role may be more sensory than mechanical (Gupton et al., 2005; Galbraith et al., 2007). In the simulation, the thickness of the leading edge is determined by the balance of a flattening force (possibly mediated by unconventional myosins) and a polymerization-driven protrusion force which while pushing out, also pushes up. Different parameter settings could certainly produce different leading edge geometries. We note however that the assumed non-nuclear cytoplasmic volume for our model fibroblast is plausible, and that the model also generates an appropriate spread cell surface area. Therefore in some average sense the thickness predicted by the model cannot be a gross overestimate.

5.3.3. Cytoskeletal density, flow, and turnover. By replacing somewhat imprecise photo-bleaching methods, the invention of speckle microscopy has allowed an unprecedented look at cytoskeletal dynamics (Vallotton et al., 2004). Again, such observations are readily compared with the output of Cytospede through the network density field θ_n and the network velocity field \mathbf{v}_n .

Figure 10 shows the ventral network volume fraction. Enhanced cytoskeletal density is evident at the leading edge extending approximately 5 μm back, a length scale approximately set by the product of the retrograde flow velocity in the leading edge frame of $v_n \sim 0.2 \mu\text{m s}^{-1}$ with the cytoskeletal turnover time scale $\tau_n = 20 \text{ s}$. Cytoskeleton also accumulates to a density ~ 5 times baseline at the rear of the cell as it is swept up by the retracting tail.

Figure 11 shows the characteristic centripetal flow of network from the leading edge. Velocities of order $v_n \sim 0.1 \mu\text{m s}^{-1}$ are in agreement with observations, although the retrograde flow does not persist very far into the cell body. Figure 12 shows the same flow in the frame of reference of the advancing leading edge.

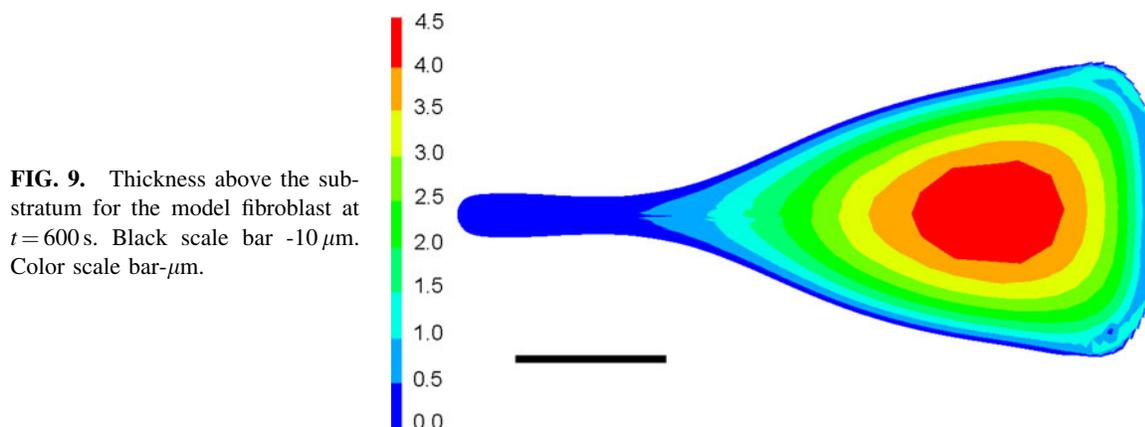


FIG. 9. Thickness above the substratum for the model fibroblast at $t = 600 \text{ s}$. Black scale bar $-10 \mu\text{m}$. Color scale bar $-\mu\text{m}$.

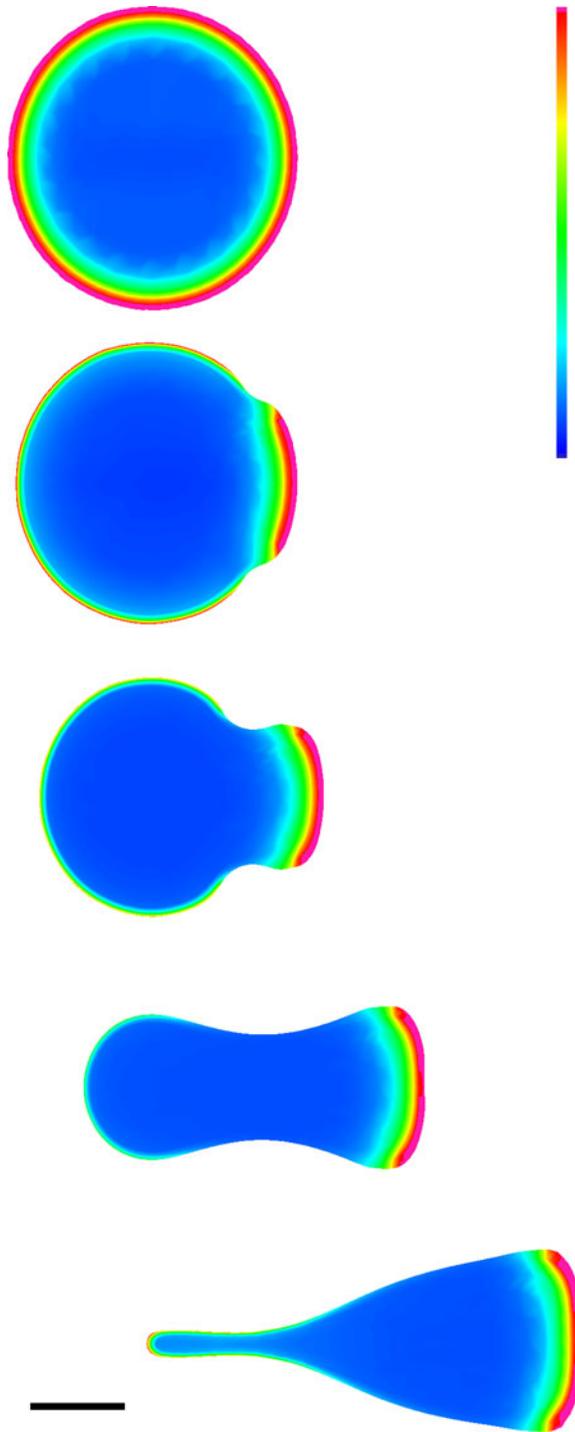
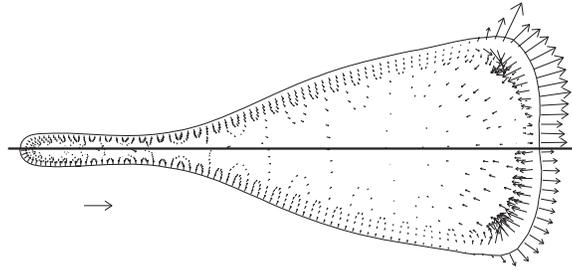


FIG. 10. Ventral cytoskeletal volume fraction for the model fibroblast at $t = 0, 20, 100, 300, 600$ s. Black scale bar $-10 \mu\text{m}$. Color scale bar $-0-1.5\%$ volume fraction of network.

5.3.4. Substratum tractions. Cells exert forces on the substratum on which they adhere. On a compliant substratum, these cellular forces cause the substratum to deform. Conversely, measurement of the deformations allows the recovery of information about cellular forces. This is the basic idea of traction microscopy by which cellular traction maps can be obtained (Dembo and Wang, 1999). Likewise, it is possible to derive a traction map from a Cytopede simulation.

In the model fibroblast described here, substratum tractions have two contributors: viscous stresses and surface tension force. The viscous tractions on the substratum are due to the shear stress imparted by the flow of the cytoskeleton:

FIG. 11. Network (cytoskeletal) velocity field in the laboratory frame for the model fibroblast at $t = 600$ s. The top and bottom half correspond to the dorsal and intermediate height velocity fields respectively (ventral velocities vanish because of the no-slip boundary condition). The bottom left reference arrow corresponds to a velocity of $0.1 \mu\text{m s}^{-1}$.



$$\begin{aligned} T_x^\nu &= \nu \frac{\partial v_x}{\partial z} \\ T_y^\nu &= \nu \frac{\partial v_y}{\partial z}, \end{aligned} \quad (97)$$

where \mathbf{T} has the dimension of a force per unit area. For a given segment of the contact line [CL], the surface tension force load is given by

$$\mathbf{F}^\gamma = - \int_{[\text{CL}]} dl (1 + \cos \alpha) \mathbf{n}_{\text{CL}} \quad (98)$$

where α is the contact angle, and \mathbf{n}_{CL} is the outward normal to the contact line (in the substratum plane). There is also a z component of the surface tension force load, but since it is never measured, we ignore it here and only consider the load in the substratum plane.

Combining viscous and surface tension contributions to the traction field is not completely straightforward because it depends on assumptions about the surface distribution of the surface tension load. For example, in Figure 13, it is assumed that the force load \mathbf{F}_e^γ on the contact line contained by a single edge element e results in a traction $\mathbf{T}_e^\gamma = \mathbf{F}_e^\gamma / A_{e,v}$, where $A_{e,v}$ is the area of the ventral face of the element. The problem here is that the magnitude of the traction depends on the ratio $l_{e,\text{CL}} / A_{e,v}$ ($l_{e,\text{CL}}$ is the length of the contact line in element e); this depends on the mesh, and in particular, on the resolution (higher resolution means higher traction because the tension force is applied to a smaller area). An alternative is to plot point forces which describe the total force due to each element as a δ -function load applied to the centroid of its ventral face. This is shown in Figure 14.

Because the Reynolds number is very small, the sum of the forces exerted on the substratum must vanish (no inertial acceleration). Indeed, we find that

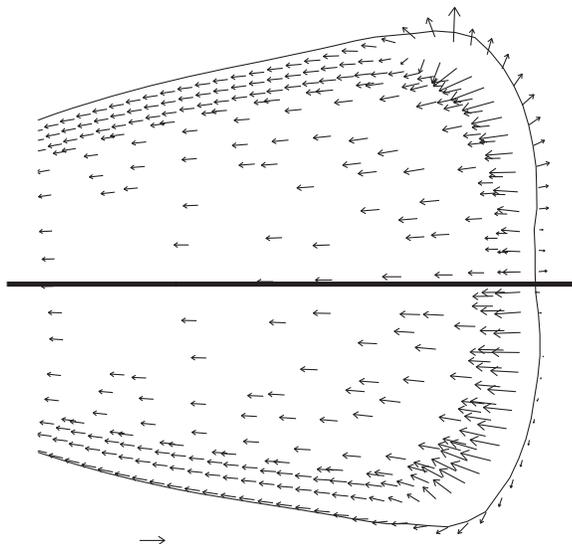


FIG. 12. Network (cytoskeletal) velocity field in the leading edge frame for the front of the model fibroblast at $t = 600$ s. The top and bottom half correspond to the dorsal and intermediate height velocity fields respectively. The bottom left reference arrow corresponds to a velocity of $0.1 \mu\text{m s}^{-1}$.

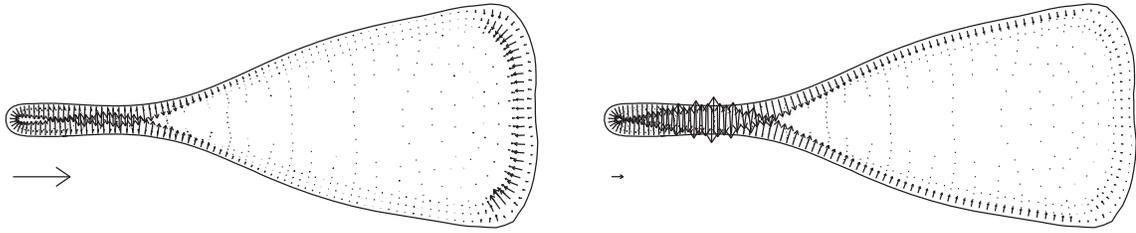


FIG. 13. Tractions exerted by the model fibroblast on the substratum. Left panel shows tractions due only to shear viscosity. Right panel shows total tractions including surface tension effects. Reference arrows correspond to a traction of $10^3 \text{ pN } \mu\text{m}^{-2}$ (10^3 Pa or 10^4 dyn cm^{-2})—note that the scaling (but not the reference arrow) changes between the two panels.

$$\frac{\left| \int dA \mathbf{T} \right|}{\int dA |\mathbf{T}|} \sim 4\%, \quad (99)$$

where the integrals run over the entire ventral surface. The small residual force is likely the result of inaccurate integration of the contact angle over the contact line in the surface tension load calculation.

Turning to the actual substratum traction and force maps, we note the following. There are no tractions or forces due to surface tension at the advancing leading edge (Fig. 14): this is simply because the contact angle there is 180° . At the narrowing of the tail of the model fibroblast, there are strong pinching tractions due to surface tension which are a reflection of the dripping faucet instability mentioned above (Fig. 13). Such tractions are not typically seen experimentally although this does not mean that they are not there; because these tractions tend to be equal and opposite over short distances, a resolution of a few μm as typical for traction microscopy will not show a net effect. Alternatively, it is possible that the waist of the tail is detached from the substratum forming a bridge between the rearmost portion of the cell and its body, in which case no traction would register.

The magnitude of the tractions at the leading edge are of order a few $10^2 \text{ pN } \mu\text{m}^{-2}$ (10^2 Pa or 10^3 dyn cm^{-2}) which is about a factor of two or three smaller than the maximum tractions observed in fibroblasts (Dembo and Wang, 1999). Since inhibition of the contractile apparatus in fibroblasts significantly decreases tractions (Beningo et al., 2006), a possible cause for this discrepancy is that we do not consider myosin mediated (longitudinal) contractility. The total force generated by the cell onto the substratum is of order 20,000 pN, a quantity that is reasonable from the molecular point of view if one considers that the typical force developed by a molecular engine is 1 pN.

The traction and force maps also illustrate something that should be intuitively obvious, that the tail and the leading edge are intimately connected; if the tail is sticky, further advance will require large protrusive tractions. In the present model, this cohesion is ensured by the surface tension, but it is also possible to conceive part of this role borne by active contractility due to myosin filaments in the cell body.

6. DISCUSSION

The essential motivation for developing Cytopede was to allow the modeling of whole cells crawling on a surface. This involves two main themes: first, the mechanical characteristics that determine the basic

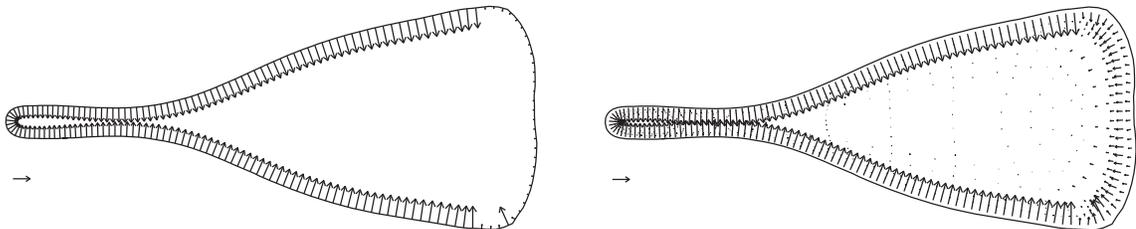


FIG. 14. Force exerted by the model fibroblast on the substratum displayed as point loads. Left panel shows loads on the contact line due to surface tension alone. Right shows total load on each ventral element due to viscous stress and surface tension. Reference arrows correspond to a force of 100 pN.

velocities and forces implicated in cellular migration; second, the chemical signaling that controls cellular spatial organization (i.e., where is the front, where is the back, and which way to move). In this article, we have mainly focused on the former because that is where the greatest technical difficulties in modeling lie and also because in the absence of a physical understanding of underlying mechanical effector processes of cell shape and motion, we felt it futile to attempt to create persuasive models of morphogenetic signaling. Thus, our strategy so far has been to lump the details of the true signal into the simplest diffusion-reaction models involving only one catch-all messenger species. While it is true that in such an approach, one is guided by what one wants to get (e.g., a hand-mirror shape for a fibroblast), such minimal models have the advantage that they can capture the essence of spatial organization without adding in the complexity of realistic biochemistry. We have recently applied this *modus operandi* by using Cytopede to develop an understanding of the key elements that make a motile cell look like a keratocyte versus a fibroblast (Herant and Dembo, 2010).

The next stage however will be to create models that incorporate signaling in a way that allows for the natural emergence of morphological properties through mechanisms such as spontaneous symmetry breaking and stable patterning (Nishimura et al., 2009). Here the difficulty is not technical; diffusion-reaction networks are comparatively easy to implement numerically. Rather the challenge becomes that the complexity of the models grows exponentially with the number of biochemical agents and with the number of interactions between them; this is associated with an enormous expansion of the parameter space that must be explored and pruned. A probable explanation for this complexity is that it mirrors the noisy, highly varied circumstances (much noisier and more varied than conditions typically encountered in the laboratory) in which cells have to operate in a predictable fashion. Thus, many signaling molecules are simultaneously involved to produce robust deterministic cellular responses. (For reviews, Wedlich-Soldner and Li see (2004) and Parent and Devreotes (1999); for theoretical investigations of specific spatial signaling, see Rappel et al. (2002) and Postma and Haastert (2001).

One has the choice between attempts at models aiming for full blown realism and encompassing dozens of species and hundreds of rates (Slepchenko et al., 2003; Gilman et al., 2002), or intermediate approaches in which various functional subsystems of the biochemistry are lumped into simplified representations (Mitchison et al., 2008). Our bias is for the latter for we tend to favor illustrative approaches that nourish physical intuition about the underlying mechanisms involved. In either case, however, Cytopede has the basic capabilities to perform simulations that meet requirements for varying degrees of signaling detail.

7. APPENDIX

7.1. Evaluation of the gradient of a surface shape function

We will address the case of a quadrilateral with natural coordinates ξ, η . At each Gauss point (or any other point of the surface) one can define two tangent vectors:

$$\partial_{\xi} = \begin{pmatrix} \partial x / \partial \xi \\ \partial y / \partial \xi \\ \partial z / \partial \xi \end{pmatrix}, \quad \partial_{\eta} = \begin{pmatrix} \partial x / \partial \eta \\ \partial y / \partial \eta \\ \partial z / \partial \eta \end{pmatrix} \quad (\text{A-1})$$

with for instance

$$\frac{\partial x}{\partial \xi} = \sum_i x_i \frac{\partial S_i}{\partial \xi}. \quad (\text{A-2})$$

These two vectors are in general neither normalized nor orthogonal but they can be used to construct two unit vectors \mathbf{e}_1 and \mathbf{e}_2 that are tangent and orthonormal. For instance $\mathbf{e}_1 = \partial_{\xi} / |\partial_{\xi}|$, $\mathbf{e}_3 = \partial_{\xi} \wedge \partial_{\eta} / |\partial_{\xi} \wedge \partial_{\eta}|$ (the unit vector normal to the surface), and $\mathbf{e}_2 = \mathbf{e}_3 \wedge \mathbf{e}_1$.

Within the $(\mathbf{e}_1, \mathbf{e}_2)$ reference frame one can construct a two-dimensional square Jacobian for the mapping between ξ, η and x, y, z :

$$\mathbf{J}_{2D} = \begin{pmatrix} e_{1x} & e_{1y} & e_{1z} \\ e_{2x} & e_{2y} & e_{2z} \end{pmatrix} \begin{pmatrix} \partial x / \partial \xi & \partial x / \partial \eta \\ \partial y / \partial \xi & \partial y / \partial \eta \\ \partial z / \partial \xi & \partial z / \partial \eta \end{pmatrix}. \quad (\text{A-3})$$

Note that $\det(\mathbf{J}_{2D})$ evaluated at a Gauss point is the same as A_{GP} in Eq. 30. This Jacobian can then be inverted to obtain the gradient of a shape function in the reference frame of $\mathbf{e}_1, \mathbf{e}_2$:

$$\begin{pmatrix} \partial_1 S \\ \partial_2 S \end{pmatrix} = \mathbf{J}_{2D}^{-1} \begin{pmatrix} \partial S / \partial \xi \\ \partial S / \partial \eta \end{pmatrix}. \quad (\text{A-4})$$

Finally, all that is left is to recover the surface gradient in the real coordinates (x, y, z) :

$$\nabla_{\text{Surface}} S = \begin{pmatrix} e_{1x} & e_{2x} \\ e_{1y} & e_{2y} \\ e_{1z} & e_{2z} \end{pmatrix} \begin{pmatrix} \partial_1 S \\ \partial_2 S \end{pmatrix} \quad (\text{A-5})$$

7.2. Volume conserving rezoning of back surface nodes

The volume of an element e is approximated by (see Eq. 27):

$$V_e = \sum_{GP \in e} w_{GP} \det(\mathbf{J}_{GP}) = \sum_{GP \in e} V_{GP} \quad (\text{A-6})$$

where w_{GP} is the weight associated with a Gauss point and $\det(\mathbf{J}_{GP})$ is the determinant of the Jacobian matrix

$$\mathbf{J}_{GP} = \begin{pmatrix} \partial x / \partial \xi & \partial x / \partial \eta & \partial x / \partial \zeta \\ \partial y / \partial \xi & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial z / \partial \xi & \partial z / \partial \eta & \partial z / \partial \zeta \end{pmatrix} \quad (\text{A-7})$$

evaluated at the Gauss point. Once again recall that for instance

$$\frac{\partial x}{\partial \xi} = \sum_i \frac{\partial H_i}{\partial \xi} x_i \quad (\text{A-8})$$

where i runs over the nodes of the element.

The change in volume of element e due to the change of position of a node i is given by

$$\delta V_e = \delta \mathbf{r}_i \cdot \nabla_i V_e \quad (\text{A-9})$$

with

$$\nabla_i V_e = \sum_{GP \in e} \begin{pmatrix} \partial V_{GP} / \partial x_i \\ \partial V_{GP} / \partial y_i \\ \partial V_{GP} / \partial z_i \end{pmatrix} \quad (\text{A-10})$$

where (x_i, y_i, z_i) are the coordinates of node i . The partial derivatives of V_{GP} can be computed explicitly. For instance we have

$$\frac{\partial V_{GP}}{\partial x_i} = w_{GP} \begin{vmatrix} \partial H_i / \partial \xi & \partial H_i / \partial \eta & \partial H_i / \partial \zeta \\ \partial y / \partial \xi & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial z / \partial \xi & \partial z / \partial \eta & \partial z / \partial \zeta \end{vmatrix}_{GP}. \quad (\text{A-11})$$

The total volume change in the computational domain due to the motion of a node i is then given by

$$\delta V = \delta \mathbf{r}_i \cdot \sum_e \nabla_i V_e \quad (\text{A-12})$$

where e runs over all elements. However, only the elements to which node i belongs make a nonzero contribution to the sum. For a surface node i we thus define

$$\mathbf{N}_V = \nabla_i V = \sum_e \nabla_i V_e \quad (\text{A-13})$$

where \mathbf{N}_V is the normal to the volume tangent plane along which motion of a boundary node does not change the total computational volume.

7.3. Minimization of the Winslow functional for rezoning of the ventral and back surfaces

Here we will give general expressions for the rezoning of the back surface. They are applicable to the ventral surface as a special case where all the z -labeled quantities disappear. Recall that we would like to reposition a set of surface boundary nodes so as to minimize the functional I over the total surface:

$$I = \sum_{e4} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta W_{e4} + \sum_{e6} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\zeta W_{e6} \quad (\text{A-14})$$

where the sum runs over all the relevant quadrilaterals ($e4$) and hexalaterals ($e6$). We therefore need to know the change in I with respect to motion dx_i of a node i . Node i only belongs to a few quadrilaterals or hexalaterals so

$$\frac{\partial I}{\partial x_i} = \sum_{e4_i} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\eta \frac{\partial W_{e4}}{\partial x_i} + \sum_{e6_i} \int_{-1}^{+1} d\xi \int_{-1}^{+1} d\zeta \frac{\partial W_{e6}}{\partial x_i} \quad (\text{A-15})$$

where the sums now run only on the element surfaces containing i . We will also need the second derivatives to implement Newton's method (Eq. 48). Since the integrals will be evaluated by Gaussian quadrature what is needed is to compute explicitly the derivatives of W at the Gauss points with respect to the motion of node. This is straightforward, but the algebra is somewhat tedious and must be organized carefully.

Setting $J = \det(\mathbf{J}_{2D})$ in Eq. 42, the two-dimensional Winslow functional W is

$$W = \frac{1}{J} [g_{11} + g_{22}] \quad (\text{A-16})$$

where the components of the metrics are as in Eq. 47. In what follows, the two intrinsic coordinates of the surface will be ξ and η but one only has to change η to ζ to recover the hexalateral case.

We wish to compute the derivatives of W with respect to displacement of node i . For this it is convenient to write $W = F/J$. First, we compute the various derivatives of the metric. We will use the notation:

$$\frac{\partial g_{kl}}{\partial x_i} = g_{kl,x} \quad (\text{A-17})$$

to denote the derivative of g_{kl} with respect to displacement dx of node i . Finally, recall once again that:

$$\frac{\partial x}{\partial \xi} = \sum_{j \in e4} x_j \frac{\partial S_j}{\partial \xi} \quad (\text{A-18})$$

where S_j is the surface shape function contribution from node j .

Metric terms: g 's.

$$g_{11} = \left(\frac{\partial x}{\partial \xi} \right)^2 + \left(\frac{\partial y}{\partial \xi} \right)^2 + \left(\frac{\partial z}{\partial \xi} \right)^2; \quad g_{22} = \left(\frac{\partial x}{\partial \eta} \right)^2 + \left(\frac{\partial y}{\partial \eta} \right)^2 + \left(\frac{\partial z}{\partial \eta} \right)^2 \quad (\text{A-19})$$

First derivatives:

$$g_{11,x} = 2 \frac{\partial S_i}{\partial \xi} \frac{\partial x}{\partial \xi}; \quad g_{11,y} = 2 \frac{\partial S_i}{\partial \xi} \frac{\partial y}{\partial \xi}; \quad g_{11,z} = 2 \frac{\partial S_i}{\partial \xi} \frac{\partial z}{\partial \xi} \quad (\text{A-20})$$

$$g_{22,x} = 2 \frac{\partial S_i}{\partial \eta} \frac{\partial x}{\partial \eta}; \quad g_{22,y} = 2 \frac{\partial S_i}{\partial \eta} \frac{\partial y}{\partial \eta}; \quad g_{22,z} = 2 \frac{\partial S_i}{\partial \eta} \frac{\partial z}{\partial \eta} \quad (\text{A-21})$$

Second derivatives:

$$g_{11,xx} = g_{11,yy} = g_{11,zz} = 2 \left(\frac{\partial S_i}{\partial \xi} \right)^2 \quad (\text{A-22})$$

$$g_{22,xx} = g_{22,yy} = g_{22,zz} = 2 \left(\frac{\partial S_i}{\partial \eta} \right)^2 \quad (\text{A-23})$$

$$g_{11,xy} = g_{11,xz} = g_{11,yz} = g_{22,xy} = g_{22,xz} = g_{22,yz} = 0 \quad (\text{A-24})$$

Numerator terms: F 's.

$$F = g_{11} + g_{22} \quad (\text{A-25})$$

First derivatives:

$$F_{,x} = g_{11,x} + g_{22,x}; \quad F_{,y} = g_{11,y} + g_{22,y}; \quad F_{,z} = g_{11,z} + g_{22,z} \quad (\text{A-26})$$

Second derivatives:

$$F_{,xx} = g_{11,xx} + g_{22,xx}; \quad F_{,yy} = g_{11,yy} + g_{22,yy}; \quad F_{,zz} = g_{11,zz} + g_{22,zz} \quad (\text{A-27})$$

$$F_{,xy} = F_{,xz} = F_{,yz} = 0 \quad (\text{A-28})$$

Denominator terms: J 's. We have previously found (Eqs. 30 and A-3) that J evaluated as a Gauss point is given by:

$$J = A_{\text{GP}} = \|\mathbf{N}_{\text{GP}}\| = \sqrt{N_x^2 + N_y^2 + N_z^2} \quad (\text{A-29})$$

where

$$\begin{aligned} N_x &= \frac{\partial y}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi} \frac{\partial y}{\partial \eta} \\ N_y &= \frac{\partial z}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi} \frac{\partial z}{\partial \eta} \\ N_z &= \frac{\partial x}{\partial \xi} \frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi} \frac{\partial x}{\partial \eta}. \end{aligned} \quad (\text{A-30})$$

First derivatives of N :

$$\begin{aligned} N_{x,x} = N_{y,y} = N_{z,z} &= 0 \quad (\text{A-31}) \\ N_{x,y} &= \frac{\partial S_i}{\partial \xi} \frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi} \frac{\partial S_i}{\partial \eta} = -N_{y,x} \\ N_{x,z} &= \frac{\partial y}{\partial \xi} \frac{\partial S_i}{\partial \eta} - \frac{\partial S_i}{\partial \xi} \frac{\partial y}{\partial \eta} = -N_{z,x} \\ N_{y,z} &= \frac{\partial S_i}{\partial \xi} \frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi} \frac{\partial S_i}{\partial \eta} = -N_{z,y} \end{aligned} \quad (\text{A-32})$$

and all the second derivatives of N vanish.

The first derivatives of J are given by:

$$\begin{aligned} J_{,x} &= \frac{1}{J} [N_y N_{y,x} + N_z N_{z,x}] \\ J_{,y} &= \frac{1}{J} [N_x N_{x,y} + N_z N_{z,y}] \\ J_{,z} &= \frac{1}{J} [N_x N_{x,z} + N_y N_{y,z}]. \end{aligned} \quad (\text{A-33})$$

The second derivatives of J are:

$$\begin{aligned} J_{,xx} &= \frac{1}{J} [N_{y,x}^2 + N_{z,x}^2 - J_{,x}^2] \\ J_{,yy} &= \frac{1}{J} [N_{x,y}^2 + N_{z,y}^2 - J_{,y}^2] \\ J_{,zz} &= \frac{1}{J} [N_{x,z}^2 + N_{y,z}^2 - J_{,z}^2] \\ J_{,xy} &= \frac{1}{J} [N_{z,x} N_{z,y} - J_{,x} J_{,y}] \\ J_{,xz} &= \frac{1}{J} [N_{y,x} N_{y,z} - J_{,x} J_{,z}] \end{aligned} \quad (\text{A-34})$$

$$J_{,yz} = \frac{1}{J} [N_{x,y}N_{x,z} - J_{,y}J_{,z}] \quad (\text{A-35})$$

Winslow functional terms: W 's.

$$W = \frac{F}{J} \quad (\text{A-36})$$

The first derivatives of W are:

$$W_{,x} = \frac{1}{J^2} [F_{,x}J - FJ_{,x}] \quad (\text{A-37})$$

with the y and z derivatives obtained by substituting x with y and z .

The second derivatives of W are:

$$W_{,xx} = \frac{1}{J^3} [F_{,xx}J^2 - 2F_{,x}JJ_{,x} + 2FJ_{,x}^2 - FJJ_{,xx}] \quad (\text{A-38})$$

with the yy and zz derivatives obtained by substituting x with y and z . We also have

$$W_{,xy} = \frac{1}{J^3} [-(F_{,x}J_{,y} + F_{,y}J_{,x})J + 2FJ_{,x}J_{,y} - FJJ_{,xy}] \quad (\text{A-39})$$

with $W_{,xz}$ and $W_{,yz}$ obtained by the substitutions $(x, y) \rightarrow (x, z)$ and $(x, y) \rightarrow (y, z)$ respectively.

7.4. Minimization of the TTM functional for rezoning of the interior nodes

Recall that we would like to reposition the interior middle nodes so as to minimize the functional I over the total computational volume:

$$I = \sum_e \int_{-1}^{+1} d\zeta \int_{-1}^{+1} d\eta \int_{-1}^{+1} d\zeta T_e \quad (\text{A-40})$$

where the sum is carried out over all the elements e . We therefore need to find the change in I with respect to a motion dx_i of interior middle node i . Node i only belongs to a few elements so

$$\frac{\partial I}{\partial x_i} = \sum_{e_i} \int_{-1}^{+1} d\zeta \int_{-1}^{+1} d\eta \int_{-1}^{+1} d\zeta \frac{\partial T_e}{\partial z_i} \quad (\text{A-41})$$

where the sum runs only on the elements containing node i . We will also need the second derivatives to implement Newton's method (Eq. 48). Since the integrals will be evaluated by Gaussian quadrature, what is needed is to compute explicitly the derivatives of T at the Gauss points with respect to the motion of a node. As for the two-dimensional Winslow case, this is straightforward but the algebra demands care.

The three-dimensional TTM functional is given by:

$$T = \frac{1}{J} [g_{11}g_{22} + g_{11}g_{33} + g_{22}g_{33} - g_{12}^2 - g_{13}^2 - g_{23}^2] \quad (\text{A-42})$$

where

$$g_{kl} = \begin{pmatrix} \partial_k x \\ \partial_k y \\ \partial_k z \end{pmatrix} \cdot \begin{pmatrix} \partial_l x \\ \partial_l y \\ \partial_l z \end{pmatrix} \quad (\text{A-43})$$

and the indices k and l run over the intrinsic coordinates ξ , η , and ζ . Further J is the determinant of the Jacobian (Eq. 17).

We wish to compute the derivatives of T with respect to displacement of node i . For this it is convenient to write $T = F/J$. First, we compute the various derivatives of the metric. We will use the notation:

$$\frac{\partial g_{kl}}{\partial x_i} = g_{kl,x} \quad (\text{A-44})$$

to denote the derivative of g_{kl} with respect to displacement δx of node i .

Metric terms: g 's.

$$g_{11} = \left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2 + \left(\frac{\partial z}{\partial \xi}\right)^2 \quad (\text{A-45})$$

$$g_{12} = \frac{\partial x}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial y}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial z}{\partial \xi} \frac{\partial z}{\partial \eta} \quad (\text{A-46})$$

and similarly for other permutations of the indices 1, 2, and 3.

First derivatives:

$$g_{11,x} = 2 \frac{\partial H_i}{\partial \xi} \frac{\partial x}{\partial \xi}; \quad g_{11,y} = 2 \frac{\partial H_i}{\partial \xi} \frac{\partial y}{\partial \xi}; \quad g_{11,z} = 2 \frac{\partial H_i}{\partial \xi} \frac{\partial z}{\partial \xi}; \quad (\text{A-47})$$

and

$$g_{12,x} = \frac{\partial H_i}{\partial \xi} \frac{\partial x}{\partial \eta} + \frac{\partial H_i}{\partial \eta} \frac{\partial x}{\partial \xi} \quad (\text{A-48})$$

$$g_{12,y} = \frac{\partial H_i}{\partial \xi} \frac{\partial y}{\partial \eta} + \frac{\partial H_i}{\partial \eta} \frac{\partial y}{\partial \xi} \quad (\text{A-49})$$

$$g_{12,z} = \frac{\partial H_i}{\partial \xi} \frac{\partial z}{\partial \eta} + \frac{\partial H_i}{\partial \eta} \frac{\partial z}{\partial \xi} \quad (\text{A-50})$$

with other indicial permutations obtained by permutations of ξ , η , and ζ .

Second derivatives:

$$g_{11,xx} = g_{11,yy} = g_{11,zz} = 2 \left(\frac{\partial H_i}{\partial \xi}\right)^2 \quad (\text{A-51})$$

and

$$g_{12,xx} = g_{12,yy} = g_{12,zz} = 2 \frac{\partial H_i}{\partial \xi} \frac{\partial H_i}{\partial \eta}. \quad (\text{A-52})$$

Once again the other indices are obtained by permutations of ξ , η , and ζ .

All the cross derivative terms (e.g. $g_{11,xy}$ or $g_{12,xy}$) vanish.

Numerator terms: F 's.

$$F = g_{11}g_{22} + g_{11}g_{33} + g_{22}g_{33} - g_{12}^2 - g_{13}^2 - g_{23}^2 \quad (\text{A-53})$$

First derivatives:

$$F_{,x} = g_{11,x}g_{22} + g_{11}g_{22,x} + g_{11,x}g_{33} + g_{11}g_{33,x} + g_{22,x}g_{33} + g_{22}g_{33,x} \\ - 2g_{12,x}g_{12} - 2g_{13,x}g_{13} - 2g_{23,x}g_{23} \quad (\text{A-54})$$

with $F_{,y}$ or $F_{,z}$ obtained by replacing x with y or z .

Second derivatives:

$$F_{,xx} = g_{11,xx}g_{22} + 2g_{11,x}g_{22,x} + g_{11}g_{22,xx} + g_{11,xx}g_{33} + 2g_{11,x}g_{33,x} + g_{11}g_{33,xx} \\ + g_{22,xx}g_{33} + 2g_{22,x}g_{33,x} + g_{22}g_{33,xx} \\ - 2g_{12,x}^2 - 2g_{12,xx}g_{12} - 2g_{13,x}^2 - 2g_{13,xx}g_{13} - 2g_{23,x}^2 - 2g_{23,xx}g_{23} \quad (\text{A-55})$$

with $F_{,yy}$ or $F_{,zz}$ simply obtained by substituting x with y or z

$$F_{,xy} = g_{11,x}g_{22,y} + g_{11,y}g_{22,x} + g_{11,x}g_{33,y} + g_{11,y}g_{33,x} + g_{22,x}g_{33,y} + g_{22,y}g_{33,x} \\ - 2g_{12,x}g_{12,y} - 2g_{13,x}g_{13,y} - 2g_{23,x}g_{23,y} \quad (\text{A-56})$$

with $F_{,xz}$ and $F_{,yz}$ obtained by the substitutions $(x, y) \rightarrow (x, z)$ and $(x, y) \rightarrow (y, z)$, respectively.

Denominator terms: J 's.

$$J = \begin{vmatrix} \partial x / \partial \zeta & \partial y / \partial \eta & \partial z / \partial \zeta \\ \partial y / \partial \zeta & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial z / \partial \zeta & \partial z / \partial \eta & \partial z / \partial \zeta \end{vmatrix} \quad (\text{A-57})$$

First derivatives:

$$\begin{aligned} J_{,x} &= \begin{vmatrix} \partial H_i / \partial \zeta & \partial H_i / \partial \eta & \partial H_i / \partial \zeta \\ \partial y / \partial \zeta & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial z / \partial \zeta & \partial z / \partial \eta & \partial z / \partial \zeta \end{vmatrix}, \\ J_{,y} &= \begin{vmatrix} \partial x / \partial \zeta & \partial x / \partial \eta & \partial x / \partial \zeta \\ \partial H_i / \partial \zeta & \partial H_i / \partial \eta & \partial H_i / \partial \zeta \\ \partial z / \partial \zeta & \partial z / \partial \eta & \partial z / \partial \zeta \end{vmatrix}, \\ J_{,z} &= \begin{vmatrix} \partial x / \partial \zeta & \partial x / \partial \eta & \partial x / \partial \zeta \\ \partial y / \partial \zeta & \partial y / \partial \eta & \partial y / \partial \zeta \\ \partial H_i / \partial \zeta & \partial H_i / \partial \eta & \partial H_i / \partial \zeta \end{vmatrix}. \end{aligned} \quad (\text{A-58})$$

Second derivatives all vanish:

$$J_{,xx} = J_{,yy} = J_{,zz} = J_{,xy} = J_{,xz} = J_{,yz} = 0. \quad (\text{A-59})$$

TTM functional terms: T 's.

$$T = \frac{F}{J} \quad (\text{A-60})$$

First derivatives:

$$T_{,x} = \frac{1}{J^2} [F_{,x}J - FJ_{,x}] \quad (\text{A-61})$$

with $T_{,y}$ and $T_{,z}$ simply obtained by substituting y and z for x .

Second derivatives:

$$T_{,xx} = \frac{1}{J^3} [F_{,xx}J^2 - 2F_{,x}JJ_{,x} + 2FJ_{,x}^2] \quad (\text{A-62})$$

with $T_{,yy}$ and $T_{,zz}$ simply obtained by substituting y and z for x .

$$T_{,xy} = \frac{1}{J^3} [F_{,xy}J^2 - (F_{,x}J_{,y} + F_{,y}J_{,x})J + 2FJ_{,x}J_{,y}] \quad (\text{A-63})$$

with $T_{,xz}$ and $T_{,yz}$ obtained by the substitutions $(x, y) \rightarrow (x, z)$ and $(x, y) \rightarrow (y, z)$, respectively.

7.5. Surface tension load at a surface node

When a boundary node i is displaced by δx , the work done against the surface tension is given by $\delta W = \delta(\gamma A)$ where δA is the change in surface area of the boundary. By the principle of virtual work, this is equivalent to having a force $\mathbf{F}_\gamma = -\nabla_i(\gamma A)$ acting on node i where ∇_i involves derivatives with respect to displacements of i . The force \mathbf{F}_γ is added to the load in the velocity momentum equation (Eqs. 76, 79, 87).

Assume i belongs to a quadrilateral surface $e4$ (all that follows works also for a hexalateral surface $e6$ with ζ replacing η), the area is approximated by:

$$A_{e4} = \sum_{\text{GP} \in e4} w_{\text{GP}} A_{\text{GP}} \quad (\text{A-64})$$

where w_{GP} is the weight associated with a Gauss point. Let

$$\mathbf{N}_{\text{GP}} = \begin{pmatrix} \partial x / \partial \xi \\ \partial y / \partial \xi \\ \partial z / \partial \xi \end{pmatrix} \wedge \begin{pmatrix} \partial x / \partial \eta \\ \partial y / \partial \eta \\ \partial z / \partial \eta \end{pmatrix} \quad (\text{A-65})$$

where for instance

$$\frac{\partial x}{\partial \xi} = \sum_{j \in e4} \frac{\partial S_j}{\partial \xi} x_j. \quad (\text{A-66})$$

We have

$$A_{\text{GP}} = \|\mathbf{N}_{\text{GP}}\| = \sqrt{N_x^2 + N_y^2 + N_z^2}. \quad (\text{A-67})$$

We need to differentiate A_{GP} with respect to changes in position of node i , but this was already done in Eq. A-33 with expressions of the type:

$$\frac{\partial A_{\text{GP}}}{\partial x_i} = \frac{1}{A_{\text{GP}}} [N_y N_{y,x} + N_z N_{z,x}]. \quad (\text{A-68})$$

Putting it together, we have:

$$\nabla_i A_{e4} = \sum_{\text{GP} \in e4} w_{\text{GP}} \nabla A_{\text{GP}} \quad (\text{A-69})$$

and

$$\nabla_i(\gamma A) = \sum_{e4/e6} \gamma_{e4/e6} \nabla_i A_{e4/e6} \quad (\text{A-70})$$

where the summation takes place over all the surface elements $e4/e6$ to which node i belongs and $\gamma_{e4/e6}$ represents the (possibly varying) surface tension for each surface element. Note that if γ is constant the vector $\nabla_i(\gamma A)$ defines a perpendicular plane which is the surface tangent at node i , i.e., the plane along which motion does not change the total surface area (the analog of the volume tangent plane discussed above).

ACKNOWLEDGMENTS

We thank Assad Oberai for assistance with certain fine points of the finite element method, A. Jesse Connell for performing some confirmatory calculations, and F. Ortega for support of his free gmV visualization package. A current version of the Cytopen code is available by request to interested researchers. This work was supported by NIH grant RO1 GM72002.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

- Bakal, C., Aach, J., Church, G., et al. 2007. Quantitative morphological signatures define local signaling networks regulating cell morphology. *Science* 316, 1753–1756.
- Bashforth, F., and Adams, J.C. 1883. *An Attempt to Test the Theories of Capillary Action*. Cambridge University Press, New York.
- Beningo, K.A., Hamao, K., Dembo, M., et al. 2006. Traction forces of fibroblasts are regulated by the rho-dependent kinase but not by the myosin light chain kinase. *Arch. Biochem. Biophys.* 456, 224–231.

- Bottino, D., Mogilner, A., Roberts, T., et al. 2002. How nematode sperm crawl. *J. Cell. Sci.* 115, 367–384.
- Charras, G.T., Mitchison, T.J., and Mahadevan, L. 2009. Animal cell hydraulics. *J. Cell. Sci.* 122, 3233–3241.
- Chen, W.T. 1981a. Mechanism of retraction of the trailing edge during fibroblast movement. *J. Cell. Biol.* 90, 187–200.
- Chen, W.T. 1981b. Surface changes during retraction-induced spreading of fibroblasts. *J. Cell. Sci.* 49, 1–13.
- Danuser, G., and Waterman-Storer, C.M. 2006. Quantitative fluorescent speckle microscopy of cytoskeleton dynamics. *Annu. Rev. Biophys. Biomol. Struct.* 35, 361–387.
- Dembo, M., and Harlow, F. 1986. Cell motion, contractile networks, and the physics of interpenetrating reactive flow. *Biophys. J.* 50, 109–121.
- Dembo, M., Oliver, T., Ishihara, A., et al. 1996. Imaging the traction stresses exerted by locomoting cells with the elastic substratum method. *Biophys. J.* 70, 2008–2022.
- Dembo, M., and Wang, Y.L. 1999. Stresses at the cell-to-substrate interface during locomotion of fibroblasts. *Biophys. J.* 76, 2307–2316.
- Dembo, M. 1994. *On Free Boundary and Amoeboid Motion. Nato Advanced Study Institute Series.* Springer-Verlag, New York.
- Dickinson, R.B., and Purich, D.L. 2002. Clamped-filament elongation model for actin-based motors. *Biophys. J.* 82, 605–617.
- Drury, J.L., and Dembo, M. 1999. Hydrodynamics of micropipette aspiration. *Biophys. J.* 76, 110–128.
- Fukui, Y., Lynch, T.J., Brzeska, H., et al. 1989. Myosin I is located at the leading edges of locomoting dictyostelium amoebae. *Nature* 341, 328–331.
- Galbraith, C.G., Yamada, K.M., and Galbraith, J.A. 2007. Polymerizing actin fibers position integrins primed to probe for adhesion sites. *Science* 315, 992–995.
- Gilman, A.G., Simon, M.I., Bourne, H.R., et al. 2002. Overview of the alliance for cellular signaling. *Nature* 420, 703–706.
- Gupton, S.L., Anderson, K.L., Kole, T.P., et al. 2005. Cell migration without a lamellipodium: translation of actin dynamics into cell movement mediated by tropomyosin. *J. Cell. Biol.* 168, 619–631.
- Harris, A.K., Wild, P., and Stopak, D. 1980. Silicone rubber substrata: a new wrinkle in the study of cell locomotion. *Science* 208, 177–179.
- He, X., and Dembo, M. 1997. On the mechanics of the first cleavage division of the sea urchin egg. *Exp. Cell. Res.* 233, 252–273.
- Herant, M., and Dembo, M. 2006. *Cytoskeletal Mechanics.* Cambridge University Press, New York.
- Herant, M., and Dembo, M. 2010. Form and function in cell motility: from fibroblasts to keratocytes. *Biophys. J.* 98, 1408–1417.
- Herant, M., Marganski, W.A., and Dembo, M. 2003. The mechanics of neutrophils: synthetic modeling of three experiments. *Biophys. J.* 84, 3389–3413.
- Herant, M., Heinrich, V., and Dembo, M. 2005. Mechanics of neutrophil phagocytosis: behavior of the cortical tension. *J. Cell. Sci.* 118, 1789–1797.
- Herant, M., Heinrich, V., and Dembo, M. 2006. Mechanics of neutrophil phagocytosis: experiments and quantitative models. *J. Cell. Sci.* 119, 1903–1913.
- Hill, T.L., and Kirschner, M.W. 1982. Subunit treadmill of microtubules or actin in the presence of cellular barriers: possible conversion of chemical free energy into mechanical work. *Proc. Natl. Acad. Sci. USA* 79, 490–494.
- Hocking, L.M. 1983. The spreading of a thin drop by gravity and capillarity. *Q. J. Mech. Appl. Math.* 36, 55–69.
- Hughes, T.J.R. 2000. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis.* Dover, New York.
- Jaffe, A.B., and Hall, A. 2005. Rho gtpases: biochemistry and biology. *Annu. Rev. Cell. Dev. Biol.* 21, 247–269.
- Knupp, P., and Steinberg, S. 1993. *Fundamentals of Grid Generation.* CRC, Boca Raton, FL.
- Kovar, D.R., and Pollard, T.D. 2004. Insertional assembly of actin filament barbed ends in association with formins produces piconewton forces. *Proc. Natl. Acad. Sci. USA* 101, 14725–14730.
- Leibler, S., and Huse, D.A. 1993. Porters versus rowers: a unified stochastic model of motor proteins. *J. Cell. Biol.* 121, 1357–1368.
- Mitchison, T.J., Charras, G.T., and Mahadevan, L. 2008. Implications of a poroelastic cytoplasm for the dynamics of animal cell shape. *Semin. Cell. Dev. Biol.* 19, 215–223.
- Nishimura, S.I., Ueda, M., and Sasai, M. 2009. Cortical factor feedback model for cellular locomotion and cytofission. *PLoS Comput. Biol.* 5, e1000310.
- Oliver, J.M., King, J.R., McKinlay, K.J., et al. 2005. Thin-film theories for two-phase reactive flow models of active cell motion. *Math. Med. Biol.* 22, 53–98.
- Padday, J.F., and Pitt, A.R. 1972. Surface and interfacial tensions from profile of a sessile drop. *Proc. R. Soc. Lond. Ser. A Math. Phys. Sci.* 329, 421.
- Parent, C.A., and Devreotes, P.N. 1999. A cell's sense of direction. *Science* 284, 765–770.

- Peskin, C.S., Odell, G.M., and Oster, G.F. 1993. Cellular motions and thermal fluctuations: the Brownian ratchet. *Biophys. J.* 65, 316–324.
- Postma, M., and Haastert, P.J.V. 2001. A diffusion-translocation model for gradient sensing by chemotactic cells. *Biophys. J.* 81, 1314–1323.
- Rappel, W.J., Thomas, P.J., Levine, H., et al. 2002. Establishing direction during chemotaxis in eukaryotic cells. *Biophys. J.* 83, 1361–1367.
- Rienstra, S.W. 1990. The shape of a sessile drop for small and large surface-tension. *J. Eng. Math.* 24, 193–202.
- Rubinstein, B., Jacobson, K., and Mogilner, A. 2005. Multiscale two-dimensional modeling of a motile simple-shaped cell. *Multiscale Model. Simul.* 3, 413–439.
- Scheidegger, A.E. 1960. *The Physics of Flow Through Porous Media*. MacMillan, New York.
- Slepchenko, B.M., Schaff, J.C., Macara, I., et al. 2003. Quantitative cell biology with the virtual cell. *Trends Cell. Biol.* 13, 570–576.
- Sousa, A.D., and Cheney, R.E. 2005. Myosin-X: a molecular motor at the cell's fingertips. *Trends Cell. Biol.* 15, 533–539.
- Tokita, M., and Tanaka, T. 1991. Friction coefficient of polymer networks of gels. *J. Chem. Phys.* 95, 4613–4619.
- Uhal, B.D., Ramos, C., Joshi, I., et al. 1998. Cell size, cell cycle, and alpha-smooth muscle actin expression by primary human lung fibroblasts. *Am. J. Physiol.* 275, L998–L1005.
- Vallotton, P., Gupton, S.L., Waterman-Storer, C.M., et al. 2004. Simultaneous mapping of filamentous actin flow and turnover in migrating cells by quantitative fluorescent speckle microscopy. *Proc. Natl. Acad. Sci. USA* 101, 9660–9665.
- Verkhovskiy, A.B., Svitkina, T.M., and Borisy, G.G. 1999. Self-polarization and directional motility of cytoplasm. *Curr. Biol.* 9, 11–20.
- Wagner, M.C., Barylko, B., and Albanesi, J.P. 1992. Tissue distribution and subcellular localization of mammalian myosin I. *J. Cell. Biol.* 119, 163–170.
- Watanabe, N., and Mitchison, T.J. 2002. Single-molecule speckle analysis of actin filament turnover in lamellipodia. *Science* 295, 1083–1086.
- Waterman-Storer, C.M., and Danuser, G. 2002. New directions for fluorescent speckle microscopy. *Curr. Biol.* 12, R633–R640.
- Wedlich-Soldner, R., and Li, R. 2004. Closing the loops: new insights into the role and regulation of actin during cell polarization. *Exp. Cell. Res.* 301, 8–15.
- Yonemura, S., and Pollard, T.D. 1992. The localization of myosin I and myosin II in *Acanthamoeba* by fluorescence microscopy. *J. Cell. Sci.* 102, 629–642.
- Zajac, M., Dacanay, B., Mohler, W.A., et al. 2008. Depolymerization-driven flow in nematode spermatozoa relates crawling speed to size and shape. *Biophys. J.* 94, 3810–3823.
- Zicha, D., Dobbie, I.M., Holt, M.R., et al. 2003. Rapid actin transport during cell protrusion. *Science* 300, 142–145.

Address correspondence to:
Dr. Marc Herant
Biomedical Engineering Department
Boston University
44 Cummington Street
Boston, MA 02215

E-mail: herantm@bu.edu

