

A COMPUTATIONAL MODEL OF CELL POLARIZATION AND MOTILITY COUPLING MECHANICS AND BIOCHEMISTRY*

BEN VANDERLEIF[†], JAMES J. FENG[‡], AND LEAH EDELSTEIN-KESHET[†]

Abstract. The motion of a eukaryotic cell presents a variety of interesting and challenging problems from both a modeling and a computational perspective. The processes span many spatial scales (from molecular to tissue) as well as disparate time scales, with reaction kinetics on the order of seconds, and the deformation and motion of the cell occurring on the order of minutes. The computational difficulty, even in two dimensions, resides in the fact that the problem is inherently one of deforming, nonstationary domains, bounded by an elastic perimeter, inside of which there is redistribution of biochemical signaling substances. Here we report the results of a computational scheme using the immersed-boundary method to address this problem. We adopt a simple reaction-diffusion (RD) system that represents an internal regulatory mechanism controlling the polarization of a cell and determining the strength of protrusion forces at the front of its elastic perimeter. Using this computational scheme we are able to study the effect of protrusive and elastic forces on cell shapes on their own, the distribution of the RD system in irregular domains on its own, and the coupled mechanical-chemical system. We find that this representation of cell crawling can recover important aspects of the spontaneous polarization and motion of certain types of crawling cells.

Key words. 2D eukaryotic cell motility, simulation, immersed-boundary method, tension, protrusion, reaction-diffusion system, deforming domain, polarization, wave-pinning, elastic perimeter

AMS subject classifications. 92C17, 74F10, 35R37

DOI. 10.1137/100815335

1. Introduction. Eukaryotic cell crawling is a complex process that involves interactions between mechanical forces and dynamics of biochemically active signaling molecules. The deformation and motion of such cells are governed by a dynamic internal structure (the cytoskeleton) that is regulated by numerous kinds of proteins and lipids. Unlike bacteria, whose motility is powered by flagella, eukaryotic cells move by a combination of protrusion, retraction, and contraction. To crawl in a directed way, the cell has to first polarize and form a front, where the cytoskeleton assembles and leads to protrusion, and a rear, where either contraction or passive elastic forces dominate. The determination of front and rear depends on external stimuli (e.g., gradients of chemo-attractant) and has to be dynamic, sensitive, and yet robust. How cells manage this complex task is a question of great interest in current cellular biology.

Mathematical and computational researchers can provide techniques that help to dissect this complex process into simpler, more easily understood modules. Prototypical “in silico” cells that share certain qualitative features with crawling cells allow us to address important questions that are not as accessible in real cells. In this paper, using simulations, we ask the following questions: (Q1) How would the mechanics decoupled

*Received by the editors November 18, 2010; accepted for publication (in revised form) September 1, 2011; published electronically November 17, 2011.

<http://www.siam.org/journals/mms/9-4/81533.html>

[†]Mathematics Department, University of British Columbia, Vancouver V6T1Z4, BC, Canada (bvander@math.ubc.ca, keshet@math.ubc.ca). The first and third author’s research is funded by the Natural Sciences and Engineering Research Council (NSERC Discovery and Accelerator grants) Canada and by a subcontract from the National Institutes of Health (grant R01 GM086882 to Anders Carlsson, Washington University, St. Louis).

[‡]Mathematics Department & Chemical Engineering Department, University of British Columbia, Vancouver V6T1Z4, BC, Canada (jfeng@math.ubc.ca). This author’s research is supported by the Petroleum Research Fund, the Canada Research Chair program, NSERC (Discovery, Accelerator, and Strategic grants), and the Canadian Foundation for Innovation.

from biochemistry dictate the changing shape of a cell as it polarizes and crawls? (Q2) How does the irregular shape of the cell affect the internal distribution of signaling proteins? And finally, (Q3) does the coupling between the two lead to emergent properties not present in each on its own? We explore the feedbacks between cell shape and the internal reaction-diffusion (RD) system that determine the chemical distribution. Such questions are difficult to probe in experimental systems and easier to test computationally. In order to investigate the mechanical-chemical coupling in cell shape and cell motion, we need suitable models and computational algorithms. Simulating the motion of a eukaryotic cell is a difficult undertaking because it leads to solving chemical equations on a moving and deforming domain. Such problems are recognized as numerically challenging. Here we present the results of simulations that aim to meet this challenge at an intermediate level of complexity. We describe a simulation package, developed from a composite of well-established techniques, that allows us to simulate the chemical polarization of the cell (determining front and back), the forces of protrusion at the front (based on implicit growth of the cytoskeleton), and deformation of the cell, culminating in its motility in two spatial dimensions.

We are not the first to simulate the motion of a eukaryotic cell. In [7] a one-dimensional (1D) continuum model is developed describing viscoelastic properties of the cytoplasm in response to internal stresses generated by the contraction. Treating the cell as a two-phase reactive fluid is a well-studied idea in which the cytoplasm is treated locally as being a mixture, one phase of actin network and another of cytosol [9], [10], [1]. The immersed-boundary method (IBM) has been used to model explicitly the actin network and its adhesive links to a substrate [2]. The cell was modeled as a two-dimensional (2D) elastic plate in [26]. Level-set methods have also been used to include chemistry dynamics on a deforming cell domain [29], [35], [36]. In most of these models, the asymmetries that lead to the motion of the cell are imposed rather than self-organized. Other models such as [27] couple chemical distribution on the cell edge with protrusion from a central hub. A previous work [19] using the Potts model approach includes more detailed biochemistry, with a Hamiltonian approach where mechanical forces are implicit rather than explicit.

Our simulations belong to the class of mechanical/fluid-based models with the following features: (1) The perimeter of our cell is elastic. This elasticity is attributable to the cortex, that part of the cytoskeleton directly adjacent to the cell membrane, generally composed of a network of actin filaments. While we do not represent that network explicitly, we assign its elastic properties to the “cell boundary.” (2) We model explicit forces, representing protrusion of the cytoskeleton, to the cell edge. Thus, we can also study the interplay between elastic and protrusive forces. This differs from level-set models (or cellular Potts models) that do not explicitly represent these forces on the edge of the cell. Level-set methods can prescribe a velocity (e.g., see [29], [35], [36]), but the boundary curve has no mechanical properties of its own. Our model is among the first models of cell motility that implements force-bearing elements along the perimeter of the cell and explicitly deals with tension and protrusion forces. (3) Our platform is one of fluid-based computations. We can simulate both the diffusion and advection of substances inside the cell. In Potts models and many other simulations, net flows of substances inside the cell are not tracked. (4) Our simulation currently has a simple but effective module that represents the internal self-organization of the cell. Other models have included simple or more detailed internal biochemistry. For example, Zajac et al. [35] studied the effect of the balanced inactivation model of [15] in their 2D level-set cell, and Marée et al. included three interacting regulatory proteins of the small GTPase

family and later added lipids such as PIP_2 , PIP_3 , etc. [19]. Keeping the regulatory system simple but biologically faithful (to small GTPases) allows us to establish overall qualitative properties of the system, while pointing to the emergent aspects of the coupling between chemistry and mechanics.

In this paper we describe how the simulation package was assembled and present some of its first results. The mathematical model at the core of the simulation comprises two coupled systems of equations. The first system is the mechanical model which describes the fluid flow and the motion of the elastic cell edge. The second system is a set of RD equations that describe a reduced model for cell polarization analyzed in one dimension in [21]. These equations will be solved on the 2D deforming domain that represents the model cell. The solution of the RD system will be directly coupled to the forces that the cell generates to crawl and deform. The shape of the cell domain will, in turn, influence the solution of the RD system.

We use our platform to probe three distinct but interrelated regimes of behavior. In the first, we consider the effect of mechanical forces that are artificially prescribed, and we investigate the shape of the cell for various force and elastic regimes. We show that even such simple cases lead to shapes and motility relevant to some cell types. Next, we consider the regulatory RD system on a static but irregular domain. We show how the shape of the domain influences the distribution of peaks of activity and in particular the effect of curvature on the ability of multiple peaks to persist. Finally, we couple the mechanical and regulatory systems, assigning forces to the cell boundary in direct relation to the local level of the active molecules. We show that such cells have very well-defined self-organized polarization and reasonable shapes and that they move in a realistic manner.

2. Model equations.

2.1. Mechanical model. Our model is a 2D representation of a cell, viewed from the top-down perspective. (See Figures 2.1(a) and 2.1(e).) We follow the common convention of ignoring the cell body and nucleus as a passive load. This is reasonable in view of the fact that cell fragments (e.g., of keratocytes) devoid of nuclei can still migrate [34].

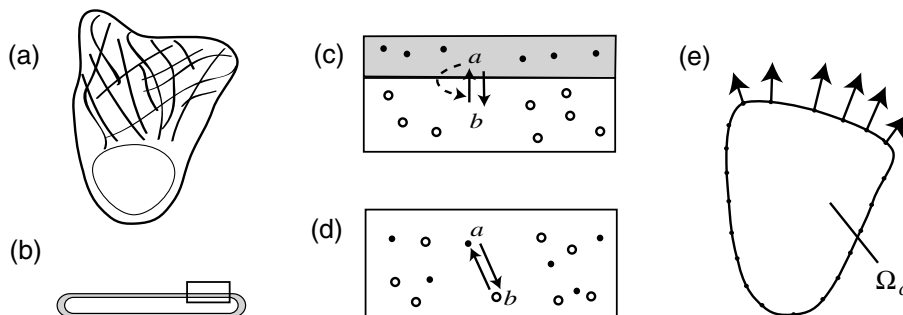


FIG. 2.1. (a) Schematic diagram of a crawling cell in top-down view (with nucleus and cytoskeleton). (b) Side view of a cell fragment with diameter $\approx 10 \mu\text{m}$ and thickness $\approx 0.1 - 0.2 \mu\text{m}$ showing membrane (grey) and cytosol (white). The rectangular region from (b) is enlarged in (c) to show the interconversion of two proteins: a is the active membrane-bound form, and b is the inactive cytosolic form. The regulatory system includes (in)activation (with positive feedback, dashed arrow) and diffusion of a, b everywhere inside the cell domain Ω_c . (d) In the simulation, we do not distinguish membrane from cytosol in the interior of Ω_c , so a, b occupy the same “compartment,” with $D_b \gg D_a$. (e) We model the 2D top-down projection of the cell Ω_c , devoid of nucleus and other structures, with elastic boundary representing the cortex, assuming a uniform thickness. The cell moves in the direction of the outward normal forces.

The cell is assumed to be resting on and adhering uniformly to a flat substrate. Typical dimensions of cell fragments are 10–30 μm diameter and 0.1–0.2 μm thickness.

The regulatory module represents proteins (Rho GTPases) that have an active form, a , bound to the cell membrane (gray surface in Figures 2.1(b) and 2.1(c)), and an inactive form b , in the fluid cytosol (white portion of the same panels). The interconversion $a \leftrightarrow b$ is regulated by other proteins (guanine nucleotide exchange factors, GTPase activating proteins, not explicitly modeled) with positive feedback from a assumed in $b \rightarrow a$ (dashed arrow in Figure 2.1(c)). Because a is membrane bound, whereas b is cytosolic, the rates of diffusion of these two proteins are very different ($D_b \gg D_a$), even though they are similar in size and shape. This is an important feature of Rho GTPases that plays a vital role in our model. Note that we do not explicitly track a membrane and a cytosol compartment. Rather, we abstract the view in Figure 2.1(c) by the simpler representation in Figure 2.1(d).

The cell is represented as a 2D domain $\Omega_c(t)$ (Figure 2.1(c)). We associate a concentration of the signaling chemicals a and b to every element of area in $\Omega_c(t)$, since every such element corresponds to a small “sandwich” of cytosol and membrane. This description is in contrast with many current models [3], [20], [17], [14] that model chemicals distributed only along the 1D cell boundary $\partial\Omega_c$. The thinness of the “cell” is taken to mean that no significant gradients form in a direction orthogonal to $\Omega_c(t)$.

Mechanically, the cell domain has an elastic perimeter enclosing a viscous incompressible fluid. The elastic perimeter represents the cell cortex as previously defined. For the purposes of the mechanical description of the cell, it is only the perimeter of Ω_c that is endowed with elasticity and bears forces due to the actin network. So long as it is not stimulated, the model cell is symmetric, assuming a circular shape with homogeneous internal chemical distribution; its perimeter is then under no elastic tension. At this point in model development, the actin network and the cell adhesion to the substrate are assumed implicitly rather than modeled explicitly. That is, we connect the internal actin-regulating biochemistry directly to forces of protrusion at the cell perimeter rather than model the regulated growth of the actin network that actually produces the force (but see [19], [25], [26]).

The mechanical model represents the interaction of the elastic membrane of the cell with a viscous incompressible fluid. To model this physical system, we use the well-known formulation of the immersed-boundary method (IBM) [24]. The key idea of this method is to replace the physical boundary conditions at the cell edge with a suitable contribution to a force density term (2.3) in the fluid equations (2.1)–(2.2) that are then solved by more conventional means. To compute fluid flows, we use Stokes’s equations. This utilizes the well-known fact that at the cellular scale, the flow is at a very low Reynolds number, and inertial effects are negligible. The other distinguishing feature of the IBM is the use of Lagrangian marker points to track the boundary of $\Omega_c(t)$, as shown in Figure 2.2. The evolution of these material points are then used to compute the elastic stresses in the cortex.

The position of the membrane is given by a vector function $\mathbf{X}(s, t)$, where s is the arc length with respect to some reference configuration and t is time. The boundary condition at $\mathbf{X}(s, t)$ is replaced by a singular force density term in the fluid momentum equation. In order to satisfy the no-slip boundary condition along the membrane, the boundary moves with the local fluid velocity \mathbf{u} (2.4). Then the immersed-boundary equations are

$$(2.1) \quad 0 = -\nabla p + \mu \Delta \mathbf{u} + \mathbf{f}(x, t),$$

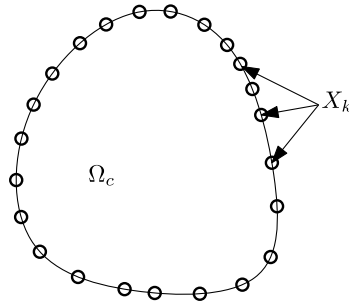


FIG. 2.2. The cell domain Ω_c is the interior of a closed loop defined by X_k , a discrete collection of Lagrangian marker points. The RD polarization model is defined only on Ω_c , while the fluid equations are defined both inside and outside Ω_c .

$$(2.2) \quad 0 = \nabla \cdot \mathbf{u},$$

$$(2.3) \quad \mathbf{f}(x, t) = \int_{\Gamma} \mathbf{F}(s, t) \delta(x - \mathbf{X}(s, t)) ds,$$

$$(2.4) \quad \frac{\partial \mathbf{X}}{\partial t} = \mathbf{u}(\mathbf{X}(s, t), t),$$

where p is pressure and μ is viscosity. The function $\mathbf{F}(s, t)$ is the magnitude of the singular force density (units: force per unit arc length s) defined along the boundary, Γ , of the domain $\Omega_c(t)$. It is composed of a force due to the elasticity of the membrane and a protrusive force due to the (implicit) polymerization of network, represented by $h(a)$. We take

$$(2.5) \quad \mathbf{F}(s, t) = \mathbf{F}_{\text{el}} + \mathbf{F}_{\text{net}}, \quad F_{\text{el}} = \frac{\partial}{\partial s} [T(s, t) \boldsymbol{\tau}(s, t)], \quad \mathbf{F}_{\text{net}} = h(a) \mathbf{n}(s, t),$$

where \mathbf{F}_{el} is the elastic force and \mathbf{F}_{net} is the protrusive network force. In \mathbf{F}_{el} , the quantity $\boldsymbol{\tau}(s, t)$ is a unit vector in the tangential direction, T_0 is the elastic modulus, and T is the tension, assumed to take the form

$$(2.6) \quad T(s, t) = T_0 \left(\left\| \frac{\partial \mathbf{X}}{\partial s} \right\| - 1 \right).$$

In \mathbf{F}_{net} , $\mathbf{n}(s, t)$ is the outward unit normal vector, and $h(a)$ is a constitutive relationship between the local concentration of the activated signaling system and the force generated by the network. At this point, the mechanics equations are coupled to the biochemical RD equations that depict the cell polarization and signaling.

Before describing the biochemical component of the model, we elaborate on our approach to the implicit representation of adhesion forces. From a modeling standpoint, we may view cell adhesion on the substrate as performing two separate functions: (a) providing a footing to the active force-generating structure inside the cell, and (b) producing a friction against the cell's movement. As for (a), we imagine that the actin network pushes the leading edge of the cell at the plus ("barbed") ends while being tethered to the substrate via adhesion proteins. Thus, polymerization and lengthening of the actin filaments can provide a pushing force \mathbf{F}_{net} that drives the cell's motion. In computing the motion of the cell and the membrane, we have treated the actin network as an external object providing the forcing \mathbf{F}_{net} . In this sense, the adhesion was accounted for

implicitly. This treatment is a simplification; it gives us a means to introduce the driving force without having to describe how the actin network evolves or how it is anchored onto the substrate, but it is dynamically consistent so far as \mathbf{F}_{net} is treated as an external force. Note, however, that because we omit the force transmitted to the substrate, in this formulation we can make no statements regarding the spatial distribution or variation of this force. Further, for function (b), the friction between the substrate and the cell can be included relatively easily by adding a drag term in (2.1). In the literature, this type of description has been considered by some authors (e.g., [24]). Its effect on the cell motion is primarily to slow it down, and we have neglected it in this preliminary model.

2.2. Biochemical model. In [19] we considered a multilayer signaling biochemistry that regulates the cell polarization and controls the growth and decay, protrusion, and contraction of the actin cytoskeleton. Those models include such proteins as Rho GTPases (Cdc42, Rac, Rho), phosphoinositides (PIP, PIP₂, PIP₃), actin, and Arp2/3. However, for the purposes of establishing polarization and determining the front and back of the cell, we later showed that a far simpler model, consisting of a single Rho GTPase in active/inactive forms, suffices. That model, described in [21] in one dimension, suits our purposes here. It is sufficiently simple for the preliminary tests of our simulation platform, while producing results in two dimensions that have inherent features of interest.

Recall that the computational cell is a 2D projection of a thin three-dimensional cell. The active form of the signaling protein, $a(x, t)$, and the inactive form, $b(x, t)$, diffuse in the domain Ω_c with disparate diffusion coefficients D_a and D_b . The two forms exchange at rate $g(a, b)$. The dynamics of our signaling model are thus governed by a pair of RD advection equations

$$(2.7) \quad a_t + \mathbf{u} \cdot \nabla a = D_a \Delta a + g(a, b),$$

$$(2.8) \quad b_t + \mathbf{u} \cdot \nabla b = D_b \Delta b - g(a, b).$$

The advection terms in (2.7)–(2.8) account for the fact that the domain $\Omega_c(t)$ moves with respect to lab coordinates and carries the biochemistry along. For $g(a, b)$, we assume positive feedback enhancing the conversion of the inactive form b to the active form a , but a constant rate δ for converting a to b :

$$(2.9) \quad g(a, b) = \left(k_0 + \frac{\gamma a^2}{K^2 + a^2} \right) b - \delta a.$$

In this term k_0 is a basal rate of activation and γ is the magnitude of the feedback activation rate. The parameters k_0 , γ , and δ all have dimensions s^{-1} . The parameter K has units of concentration of a , and we normalize concentrations so that $K = 1$.

As shown in one dimension in [21], under appropriate conditions, the system (2.7)–(2.9) supports solutions in the form of a traveling wave that stalls inside the domain (the wave is then said to be “pinned”). In a pinned wave, the domain is roughly subdivided into one region that supports a high plateau of a , while the remaining region has a low a plateau; a sharp interface separates these zones, while the level of b is relatively uniform throughout. We refer to the high (low) levels of a as a^+ (respectively, a^-). Such a solution will be our description of a polarized cell, with a^+ the front portion and a^- the rear portion of the cell. Two features of the model essential for this kind of polarized outcome

are $D_b \gg D_a$ and conservation of total a and b . We therefore impose no-flux conditions for both a and b at the boundary of Ω_c . A third necessary condition is that for a fixed $b = b_0$ within some range of values, the function $g(a, b_0)$ has three steady-states (a^-, a_h, a^+) , the outer two of which are stable, i.e., that the well-mixed system is bistable in the variable a . We choose initial conditions that place b within the appropriate range and a close to the value $a^-(b)$.

2.3. Coupling biochemistry and mechanics. We approximate the complex mechanical-biochemical coupling by assuming a direct link between the concentration of the activated signal protein a and the local force normal to the cell membrane. This assumption shortcuts the dynamics of growth of the actin cytoskeleton and replaces the actin polymerization force by an “effective force of protrusion” due to local activation of the Rho GTPase. That is, when a is above some threshold value a_0 , we assume that there is a local force directed outward. For our study we take the force magnitude $h(a)$ to be a piecewise quadratic function, with adjustable parameters H and a_0 . A convenient form for the relationship of force to the level a of the activated protein is

$$(2.10) \quad h(a) = \begin{cases} H \left(1 - \frac{(a-a^+)^2}{(a_0-a^+)^2} \right) & \text{if } a > a_0, \\ 0 & \text{otherwise.} \end{cases}$$

This type of force distribution parallels an experimentally observed distribution of actin filament ends that push on the leading edge in cells such as keratocytes [13]. However, we stress that for now our representation of the force distribution is meant to be qualitative.

3. Numerical methods. The primary difficulty in solving the model equations numerically is that the immersed-boundary equations are coupled to the RD system through the specification of the cell domain. The solution of the immersed-boundary equations determines the locations of marker points, X_k , which comprise the discretization of the immersed boundary. The location of these marker points define Ω_c , on which the RD system is to be solved. The solution of the RD system is then coupled to the IBM since it appears in the forcing term $f(x, t)$. To further add to the difficulty, the reaction kinetics happen on the time scale of seconds, much faster than the motion of the cell (typically on the order of minutes or longer).

We couple together several known and tested numerical methods and tools in the solution of our model equations. We use the IBM for the formulation of the mechanical boundary condition and discretization of the boundary [24], the method of regularized Stokeslets for the flow computation [4], and the immersed interface method for the solution of the RD system on the cell domain [6], [16]. In order to represent the interface and track its location, we compute a level-set function. In addition, we use adaptivity locally in time in order to make the computation robust. In terms of computational cost, this combination of methods is comparable to using finite element or level-set methods but is more costly than simulations based on a Potts model.

The solution of the model equations is carried out in the following steps:

1. Compute the force distribution along the cell boundary due to membrane elasticity and protrusion.
2. Compute the flow field at the boundary marker points and on an internal Cartesian grid that holds the signal concentrations.
3. Advect the membrane using the computed velocity.
4. Advect the solution of a and b according to the current fluid velocity.
5. Evolve the solution of a and b according to the RD system.

TABLE 3.1
Typical numerical parameters used for simulations.

dx	$1 \mu m$
ds	$0.75 \mu m$
ϵ	$0.6 \mu m$
Δt_{fl}	0.001 sec
Δt_{rd}	0.001 sec
Δt_{ad}	0.1 sec
m	100
n	100

We define the three time steps Δt_{fl} , Δt_{ad} , Δt_{rd} for the coupled system. These represent the discretization of time in the fluid equations, the coupling advection term, and the RD system, respectively. The motivation for this method of splitting is that there is a wide separation among the time scales of the different processes and a time step restriction that is a result of the stiff immersed-boundary equations. We allow for the possibility of taking multiple time steps in the fluid equations before updating the RD system. A further reason for this split is that there is further computational overhead in the advection of a and b . This will be discussed shortly. The time-marching scheme is thus comprised of the following steps:

1. Carry out m time steps of the immersed-boundary system.
2. Carry out a single time step of the advection of the signal concentration.
3. Carry out n time steps of the RD system.

We will take $m\Delta t_{fl} = n\Delta t_{rd} = \Delta t_{ad}$ and set each of the time steps small enough to accurately capture the associated physical phenomena and maintain stability of the numerical solution. We further wish to restrict the movement of Ω_c to less than Δx in the time interval Δt_{ad} . We provide typical numerical parameters in Table 3.1.

3.1. Fluid velocity. The simulation of the immersed elastic membrane is a well-documented canonical problem that is treated by the IBM [22], [23], [28], [32]. We discuss here only the computation of the terms in (2.5) that comprise $F(s, t)$ in our model. The restoring elastic force of the membrane, F_{el} , is treated identically to other IBMs. We take a centered difference discretization of derivatives comprising F_{el} , $T(s, t)$, and $\tau(s, t)$:

$$(3.1) \quad F_{el}(s_k, t) = \frac{T(X_{k+1/2}, t)\tau(X_{k+1/2}, t) - T(X_{k-1/2}, t)\tau(X_{k-1/2}, t)}{\Delta s},$$

$$(3.2) \quad T(X_{k+1/2}, t) = T_0 \left(\frac{|X_{k+1} - X_k|}{\Delta s} - 1 \right),$$

$$(3.3) \quad \tau(X_{k+1/2}, t) = \frac{X_{k+1} - X_k}{|X_{k+1} - X_k|}.$$

Substituting (3.2) and (3.3) into (3.1), one can rewrite the formula for F_{el} as follows:

$$(3.4) \quad F_{\text{el}}(s_k, t) = \sum_{i=k-1}^{k+1} \frac{T_0}{\Delta s} (|X_i - X_k| - \Delta s) \left(\frac{X_i - X_k}{|X_i - X_k|} \right) \frac{1}{\Delta s}.$$

From (3.4) we see that the elastic force is equivalent to a force produced by discrete springs connecting the point X_k with its neighbors. The resting length of the springs is Δs , taken in our calculation to be the spacing of the initial discretization of the immersed-boundary points. We define the spring constant $\sigma = \frac{T_0}{\Delta s}$. Note that while T_0 has units of force, σ has units of force/length. To compute F_{net} , we determine the local value of a using a piecewise constant reconstruction evaluated at X_k .

The analytic representation of the Stokes flow that results from a single point force in the absence of external boundaries is known as the Stokeslet. This velocity field is singular at the location of the point source. For the purpose of computing the Stokes flow that results from a discrete collection of point forces F_k located at X_k , we use the method of regularized Stokeslets [4]. The method replaces the singular force with a regularized force and computes an approximation of the true flow, which does not have the singularity. We use the following regularization found in [31]:

$$(3.5) \quad u(x) = -\frac{f_0}{4\pi\mu} \left[\frac{1}{2} \ln(r^2 + \epsilon^2) - \frac{\epsilon^2}{r^2 + \epsilon^2} \right] + \frac{1}{4\pi\mu} [f_0 \cdot (x - x_0)](x - x_0) \left[\frac{1}{r^2 + \epsilon^2} \right].$$

Here, f_0 is the regularized force positioned at x_0 , r is the distance between x and x_0 , and ϵ is the regularization parameter. Based on linearity of the Stokes equations, we obtain the velocity due to a collection of point forces by simple superposition. (See [5] for an example of the application of the method to a three-dimensional model of a swimming microorganism.)

In typical IBMs, the solution of the fluid equations is found on a periodic rectangular domain with standard Stokes solvers. In some models, fixed boundaries are included by means of another discretized boundary, along which forces are distributed in order to produce zero flow at the boundaries. In this formulation, the problem is ill-posed when the integral of the forces does not vanish. Furthermore, the solution is only unique up to an arbitrary constant. The work in [30] discusses this problem and presents a solution method.

A similar issue arises with the use of regularized Stokeslets due to the logarithmic growth of the Stokeslet as $r \rightarrow \infty$. If the integral of the forces is nonzero, the velocity will be nonzero at infinity due to Stokes's paradox. The inclusion of protrusive forces in our model means that we must take additional measures to ensure that the flow will decrease to zero away from the cell domain $\Omega_c(t)$. For this purpose, we include fixed walls in our computation, at which we enforce the condition that the flow velocity vanishes. This is achieved by discretizing the wall and distributing forces that cancel the flow due to the logarithmic term.

The computation of the flow velocity due to a collection of N point forces is the superposition of their individual contributions due to the linearity of the equations. The velocity at a point X_j due to the point forces F_k positioned at X_k is thus a sum:

$$(3.6) \quad u(X_j) = \sum_k S_{jk} \cdot F_k.$$

In this sum, the S_{ij} are the terms in (3.5) that multiply the forces F_k . Alternatively, this may be written as a matrix multiplication where f is a vector of the forces, u is a vector of the unknown velocities, and S is a $2N \times 2N$ matrix:

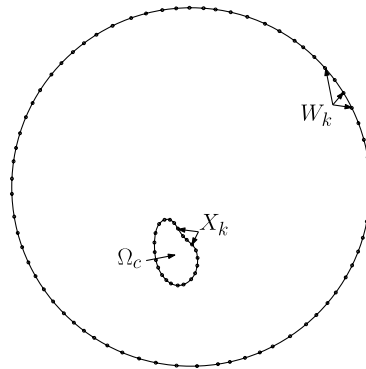


FIG. 3.1. In order to enforce the condition of zero flow far away from the cell, additional immersed-boundary points, W_k , are included in the computation as the representation of a fixed wall.

$$(3.7) \quad Sf = u.$$

The inverse of this problem can also be considered, where the velocity at a collection of points is specified and the forces that produce the velocity are unknown. The flow around a cylinder is demonstrated in [4] using this method.

In our model, we include a collection of wall points W_k , as shown in Figure 3.1, at which the velocity will be zero. To compute the flow at a given point in the fluid domain due to the point forces F_k at X_k , we take the following steps:

1. Compute the velocity at the wall points W_k . Call this velocity u_w .
2. Solve a linear system for unknown forces G_k positioned at W_k that will produce $-u_w$ at W_k .
3. Sum over contributions to the velocity from F_k and G_k .

Although we must solve a dense system for G_k at every time step, the matrix remains the same since the position of the wall points does not vary. We can therefore compute a Cholesky decomposition once and perform back substitution to obtain the solution needed at each time step. We use the standard routines in LAPACK for these calculations.

One advantage of the IBM is the modularity of the solver that is used to solve the flow equations (2.1)–(2.2). More complex solution methods could be used if the model were generalized to give the fluid inside the cell a higher viscosity. Another possibility is the use of viscoelastic constitutive equations for the fluid and solution methods appropriate for such equations. The immersed-boundary formulation of the interaction between the fluid and the elastic cell perimeter remains valid for such extensions.

3.2. RD equations. Next we describe the solution of the RD system on the irregular cell domain. We will address the motion of the domain shortly, but for now, we focus on approximating the solution of this system subject to no-flux boundary conditions on a static domain of arbitrary shape. The approach we take is that of the immersed interface method. The irregular domain is embedded in a rectangular domain on which a regular Cartesian grid is used. We produce a numerical solution at all grid points. For grid points outside the irregular domain, we set the numerical solution to zero. For grid points inside the domain, away from the irregular boundary, we use the standard centered finite difference approximations to the second derivatives in the Laplacian. It remains to deal with grid points inside the domain next to the boundary, where the standard discretization fails. These are denoted irregular points. At these

irregular points we must use a special finite difference formula that incorporates the no-flux condition and the local geometry of the boundary. To reiterate, the solution of the RD system will be nonzero only on grid points determined to be inside the closed loop defined by the current location of the marker points.

We use the discretization scheme presented in [12] and refer the reader to that work for the details of the derivation. We summarize here the main idea and comment on the robustness with respect to geometry. See Figure 3.2 for the typical irregular point geometry. If the boundary conditions were Dirichlet, we would know the solution values Ψ_N and Ψ_E at the Lagrangian marker points. We could then write a discretization of Ψ_{xx} using Ψ_E and the unknowns Ψ_0 and Ψ_1 and a discretization of Ψ_{yy} using Ψ_N and the unknowns Ψ_0 and Ψ_2 . However, for our model, we must employ Neumann (no-flux) boundary conditions. Thus, we do not have the values of Ψ_N and Ψ_E . Instead, these values must be interpolated from Ψ_0, Ψ_1, Ψ_2 , and Ψ_3 using the fact that $\Psi_n = 0$ along the boundary.

A system of two algebraic equations can be written for the unknowns Ψ_N and Ψ_E . We solve this system analytically and write Ψ_N and Ψ_E explicitly in terms of Ψ_0, Ψ_1, Ψ_2 , and Ψ_3 . In order to make the method robust, we enforce a lower bound on the determinant of the system. This precludes cases where the geometry is nearly degenerate, as is the case when the interface is very near a grid point. This is a common problem observed in similar numerical methods.

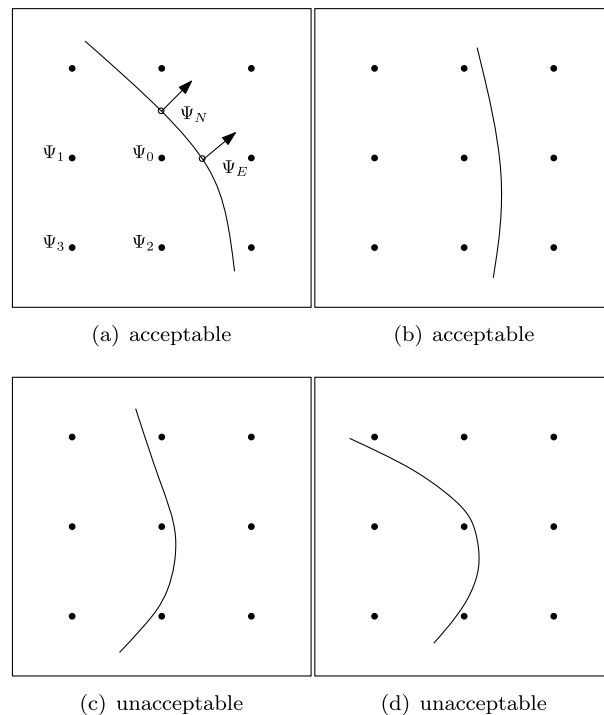


FIG. 3.2. The points included in the finite difference scheme at a typical irregular point are shown in (a). Ψ_N and Ψ_E are the (unknown) values of the solution to the RD system on the “cell perimeter.” Ψ_i for $i = 0 \dots 3$ are the values of the solution at the grid points inside the computational domain Ω_c . The configuration in (b) is treated similarly. At times when the boundary is in unacceptable configurations, we interpolate the solution to a finer mesh in order to proceed with the computation.

The matrix of coefficients depends on the positions of the projections Ψ_N and Ψ_E as well as the unit normal vectors at these points. We must therefore have some continuous representation of the boundary constructed from the immersed-boundary points X_k . The representation we use is a level-set, as discussed in the next section.

A further complication we face in making the computation robust is that we must handle all potential configurations of the boundary with respect to the grid. For example, the discretization scheme just described will fail for configurations such as those indicated in Figures 3.2(c) and 3.2(d). One approach would be to write discretizations for all configurations [6]. We choose instead to use an adaptive grid to avoid all cases except for the two configurations shown in Figures 3.2(a) and 3.2(b) (and their rotations). We perform refinements of the grid and bilinear interpolations of the solution until the unwanted cases are eliminated. The refinements are performed globally in space but locally in time. That is, for any time step we check for unwanted configurations. If any are found, we refine the grid everywhere, interpolate the solution to the fine grid, and repeat if necessary. Once we have a satisfactory grid, we evolve the system and then project the solution back to the original grid. While this simple adaptivity requires additional work, in practice it applies to less than 10% of the time steps, and the number of refinements needed is at most two. The boundary is relatively smooth because it is a level-set of the grid-based function defined in the next section.

With the spatial discretization established, it remains to choose a suitable temporal discretization. We use a fully implicit time method for the diffusion operator in order to alleviate the otherwise prohibitive restriction on the time step size. We use an explicit step for the reaction term. This means that we must solve two sparse linear systems at every time step:

$$A_a a^{n+1} = a^n + \Delta t g(a^n, b^n), \quad A_b b^{n+1} = b^n - \Delta t g(a^n, b^n).$$

We use a GMRES method that is implemented in the library IML++.

3.3. Advection and coupling. In order to couple the mechanical system and the chemical system, we use a method first described in [33] for the computation of multi-fluid flows. The idea is to construct a regularized version of an indicator function for the cell domain, that is, a function that is equal to one in Ω_c and zero outside Ω_c and that varies smoothly within a transition region near the boundary. Although the Lagrangian markers X_k give us an explicit representation of the boundary, having such a function eliminates the need for complicated routines to track which of the fixed Cartesian grid points are inside Ω_c at any given time, knowing only the trajectories of the markers. We can further use this indicator function to compute a level curve to use as a convenient representation of the boundary of Ω_c when finding the discretizations needed for the irregular points in the immersed interface method.

Given only the location of the marker points, we compute $I_\epsilon(x)$, the regularized indicator function which defines the domain on which the RD system will be solved. The gradient of the discontinuous indicator function $I(x)$ is zero everywhere except at the interface, where it has a singularity. For the purpose of approximation, this singularity is represented again with the regularized delta function δ_ϵ , and the gradient is then written as

$$(3.8) \quad \nabla I_\epsilon = \sum_k \delta_\epsilon(x - X_k) n_k \Delta s_k.$$

We then take the numerical divergence of ∇I_ϵ to find ΔI_ϵ . There remains only to solve the following Poisson problem in order to obtain I_ϵ :

$$(3.9) \quad \Delta I_\epsilon = \nabla \cdot \nabla I_\epsilon.$$

This problem is solved on a rectangular domain which contains Ω_ϵ . We use the same δ_ϵ presented in [32], which has compact support of length $4h$ in each dimension. We position the rectangular domain such that its boundary is not within $2h$ of $\partial\Omega_\epsilon$, so that the Dirichlet data for (3.9) is zero. We then use a GMRES routine to find the solution.

Note here that a coarse discretization of the boundary of Ω_ϵ relative to the Cartesian grid will produce an I_ϵ with a level curve that is oscillatory. If a given boundary configuration requires the use of adaptivity as described earlier, (3.9) is solved only on the coarsest grid. This solution is then interpolated to any refined grids in order to give a consistent description of the boundary on these grids.

The final detail that remains is to specify how to determine the solution of a and b at new points that enter Ω_ϵ as it evolves with time. We need a way to map the solution of a and b from the previous domain to the current domain (Figure 3.3). We use a basic semi-Lagrangian discretization of the advection term that couples the RD equations to the flow equations. Our Stokeslet solution to the fluid equations allows us to compute the flow anywhere. Specifically we can compute the velocity on the same grid as we are using for a and b . At each grid point x_{ij} in the current domain we use the fluid velocity to approximate the previous location of the material.

$$(3.10) \quad x_{ij}^{n-1} = x_{ij}^n - \Delta t_{ad} u_{ij}^n.$$

The location x_{ij}^{n-1} will normally not lie on a grid point, but we can now use bilinear interpolation of the solution values of a and b at neighboring grid points to compute the value at x_{ij}^{n-1} . In order to ensure that we have four neighboring grid points for this interpolation, we first extrapolate the solutions at time $n-1$ to the nearest grid points outside the boundary.

The advection scheme should conserve the total mass of the system $M_{\text{tot}} = \int_{\Omega_\epsilon} (a+b) dx$. We approximate this integral with a discrete sum M_h , again making use of the smooth indicator function:

$$(3.11) \quad M_h = \sum_{(x_i, y_j) \in \Omega_\epsilon} I_\epsilon(x_i, y_j) (a(x_i, y_j) + b(x_i, y_j)) h^2.$$

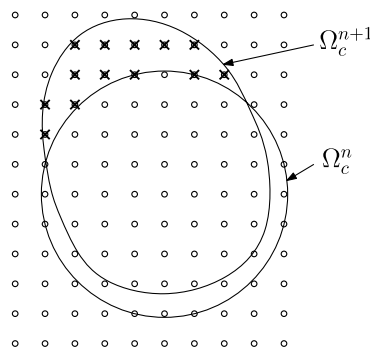


FIG. 3.3. The known concentration at grid points in the domain Ω_ϵ^n get advected with the fluid velocity to the grid points in the domain at the next time step Ω_ϵ^{n+1} . As the cell perimeter moves, new grid points (marked with crosses) enter Ω_ϵ . The advection defines values of the solution at these points in a way that is consistent with the model equations.

In the sum, points that are near the boundary of Ω_c are weighted with a value less than h^2 since the area inside Ω_c that is associated with boundary points is less than h^2 . When performing the advection, we want to preserve the quantity M_{tot} as well as the profile of the solution. That is, we do not want to destroy internal gradients through the advection step. It is also important to keep the zero outward normal derivative that is consistent with the discretization of the diffusion operator for boundary points. For our application, we focus on preserving the solution profile and modify the advection step so as to preserve M_{tot} by adding or subtracting a small constant amount that may be lost in the advection scheme. We found that by so doing, we could retain the correct total mass to within 1% of its fixed value at any given advection time step. The mass change due to the diffusion discretization is orders of magnitude smaller.

3.4. Verification of numerical accuracy. When solving coupled systems of PDEs numerically it is difficult to verify the accuracy of approximate solutions to the full system. In order to test our solutions, we examine the three main components of our algorithm dealing with fluid flow, diffusion, and advection. The verification of the method of regularized Stokeslets has been presented in [4], and we do not repeat these results here. The discretization of the diffusion operator was tested in [12] for a Poisson equation. We test the accuracy here on a time-dependent problem, in a domain defined by the contour of the indicator function as it is computed in our simulations. In order to verify our solution of a diffusion problem on a moving domain, we conduct the test used in [28] for the same purpose.

In order to test the accuracy of the discretization of the diffusion operator, we solved the equation

$$(3.12) \quad u_t = \Delta u + f(x, y)$$

with a no-flux boundary condition, on a disk of radius $\frac{1}{4}$ centered at the origin. The initial conditions and the function $f(x, y)$ were chosen such that the analytic solution to the problem is

$$u(r, t) = e^{-4\pi^2 t} \cos(4\pi r),$$

where $r = \sqrt{x^2 + y^2}$. We computed two numerical solutions of this problem, one using an analytical representation of the boundary, and one using the level-set of the indicator function to define the boundary. We compared these two solutions to an analytic solution at dimensionless time $T = 0.1$, chosen to be within the transient period of the solution. Define the error at time T in the numerical solution \hat{u} as

TABLE 3.2

Errors computed for test problems. In the stationary problem we refined the time step Δt_{rd} to be proportional to Δx^2 . For the level-set boundary test we used $2N$ points to discretize the boundary curve. This gives the boundary a discretization $\Delta s \approx 0.8\Delta x$. In the moving boundary problem we refined the diffusion time step Δt_{rd} proportionally to Δx^2 and the advection time step Δt_{ad} proportionally to Δx .

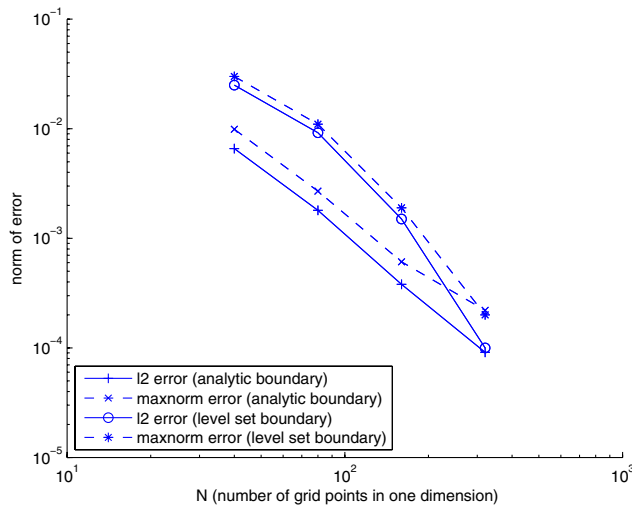
	Stationary domain problem				Moving domain problem	
	Analytic boundary		Level-set boundary		Level-set boundary	
N	$\ e\ _2$	$\ e\ _\infty$	$\ e\ _2$	$\ e\ _\infty$	$\ e\ _2$	$\ e\ _\infty$
40	6.6×10^{-3}	9.9×10^{-3}	2.5×10^{-2}	3.0×10^{-2}	2.7×10^{-3}	2.3×10^{-2}
80	1.8×10^{-3}	2.7×10^{-3}	9.2×10^{-3}	1.1×10^{-2}	1.4×10^{-3}	1.2×10^{-2}
160	3.8×10^{-4}	6.1×10^{-4}	1.5×10^{-3}	1.9×10^{-3}	6.8×10^{-4}	6.0×10^{-3}
320	9.1×10^{-5}	2.2×10^{-4}	1.0×10^{-4}	2.0×10^{-4}	2.4×10^{-4}	3.1×10^{-3}

$$e = u - \hat{u},$$

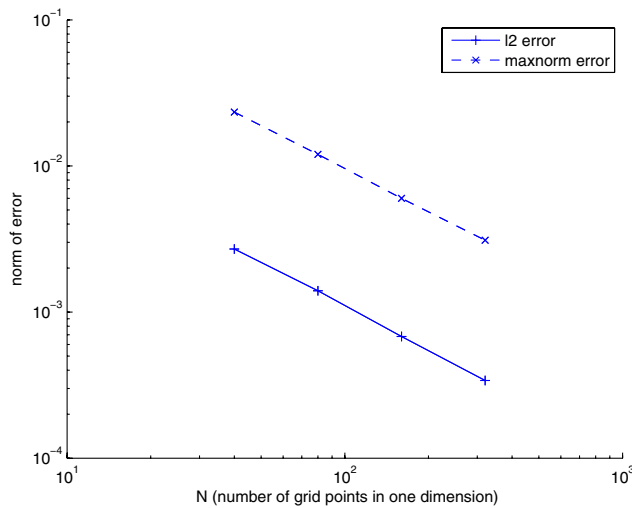
where u is the analytical solution restricted to the computational grid. We report the size of errors in terms of the norms

$$\|e\|_2 = \sqrt{\sum_{i,j} e_{ij}^2} h^2, \quad \|e\|_\infty = \max_{i,j} |e_{ij}|.$$

The computed errors are reported in Table 3.2 and plotted in Figure 3.4(a). This data suggests that we achieve the second order accuracy in space that is reported in [12]. We



(a) stationary domain



(b) moving domain

FIG. 3.4. Plots of the numerical errors reported in Table 3.2. The horizontal axis is the number of grid points in one dimension, and the vertical axis is the norm of the errors. (a) Comparison of the numerical solution of the diffusion problem on a stationary domain with an analytic solution. (b) Comparison of a numerical solution computed on a moving domain with a second numerical solution computed on a stationary domain.

are restricted to first order accuracy in time by the backward Euler scheme used for the reaction terms.

To test the coupling of the advection and diffusion, we solved a diffusion problem on a domain advecting with a constant velocity. We again used a disk of radius $\frac{1}{4}$ with no-flux boundary conditions. We chose the initial condition to be a Gaussian centered on the circle, with variance small enough to give the solution negligible normal derivative along the boundary. We computed a solution to this problem at dimensionless time $T = 1.0$, where again T is chosen within the transient period of the solution. We define this solution on the advecting domain as u_a and the disk's location at time T as Ω_T . We compute a second solution on a stationary Ω_T starting with the same initial condition that produced u_a , translated to Ω_T . We report the differences between these solutions as the errors in Figure 3.4(b). For this problem we observe the expected first order accuracy in space due to the discretization of the advection.

4. Results. We present numerical experiments to illustrate separately the effects of protrusive forces along the domain edge $\partial\Omega_c$, the internal RD solver on a static irregular domain Ω_c , and the coupled system with self-organized forces emanating from the RD system.

4.1. Mechanical model. We first consider a domain with no internal chemistry. Starting in each case with a disk-shaped domain $\Omega(0)$, we prescribe half of the cell boundary to be the “front” and apply the protrusion force along this portion for the duration of the simulation. We assign an outward normal force whose magnitude has a parabolic profile that varies from a maximum of H at the front center to zero at the cell sides.

As our curved domain edge is elastic, we can experiment with a variety of physical edge properties and examine how these affect the evolving shape of the domain. Note that this is one feature that makes the IBM distinct from level-set methods, wherein the domain edge has no mechanical properties in and of itself. The protrusive forces cause the perimeter of the domain to deform and to move in the direction of the force. The front edge that is under forcing tends to stretch as it translocates, while the rear edge (on which no force is prescribed) responds passively via the elastic tension along the edge. Figures 4.1 and 4.2 illustrate solutions with typical shapes that are produced. Figure 4.1 illustrates the transition in time between the initial circularly shaped cell and its steady-state shape. The transition in this example occurs over a time span of 360 s. For cells with a smaller value of H , a steady-state shape is reached in less time.

To determine how mechanical parameters influence the shapes and if such mechanics-only solutions achieve steady-state, we varied the three parameters in our mechanical system: the fluid viscosity μ , the membrane elasticity T_0 , and the protrusion force H . The ranges over which these parameters are varied include estimates for membrane elasticity, and the forces and fluids relevant to them, since we are aiming to tailor our computation to solving problems on the scale of cell size and motion. We use the viscosity of water ($\mu = 0.01 \text{ g} \cdot \text{cm}/\text{s}$), since the fluid is most analogous to the cytosol, and the typical forces (in pN) and edge tension (pN per μm) estimated in [13].

Figure 4.2 illustrates the variety of shapes obtained by varying the force on the leading edge, and the elasticity of the membrane keeping the viscosity $0.01 \text{ g}/\text{cm} \cdot \text{s}$ constant. For a relatively small value of $T_0 \approx 0.1 \text{ pN}/\mu\text{m}$, we see a variety of cell shapes. At low force magnitude, $H \approx 1 \text{ pN}/\mu\text{m}$, the cell shape resembles a teardrop (top left). At higher protrusive force, $H \approx 5 \text{ pN}/\mu\text{m}$, the teardrop develops a longer

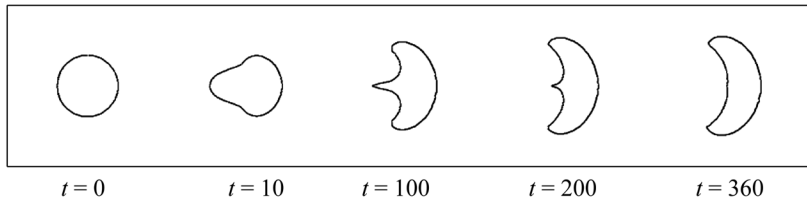


FIG. 4.1. *Mechanical model on its own. A time sequence showing the initial configuration of the model cell and its evolving shape over time. The simulation was carried out using an elastic modulus $T_0 = 0.1$ pN/ μm , protrusion force $H = 12$ pN/ μm , and viscosity $\mu = 0.01$ g/cm \cdot s. By time $t = 360$ s, the cell is in steady-state motion at speed $v = 12$ $\mu\text{m}/\text{s}$ with the shape shown.*

tail, often resulting in a cusp-like endpoint. At yet higher protrusion, the cell flattens out into a “canoe” shape and then a narrow crescent. Our current implementation of the IBM does not produce an accurate result when the magnitude of the protrusive force is much greater than the force due to elasticity. For example, when $T_0 = 0.1$ pN/ μm , values of H greater than 12 pN/ μm lead to large deformations that are not adequately resolved, and steady shapes then fail to form (at the bottom left of Figure 4.2 is a transient shape, beyond the range of applicability of the method).

When the elastic modulus is larger, $T_0 \approx 5 - 100$ pN/ μm , the transition between shapes as H is increased is less dramatic, since protrusion forces are counterbalanced by an increasing tendency to “round up.” Shapes produced for reasonable protrusion forces of 1 – 50 pN/ μm and reasonable membrane elasticity ranges of 0.1 – 100 pN/ μm resemble a transition of shapes of cells such as neutrophils (teardrops) to keratocytes (flattened canoes), cells whose motility is commonly studied experimentally.

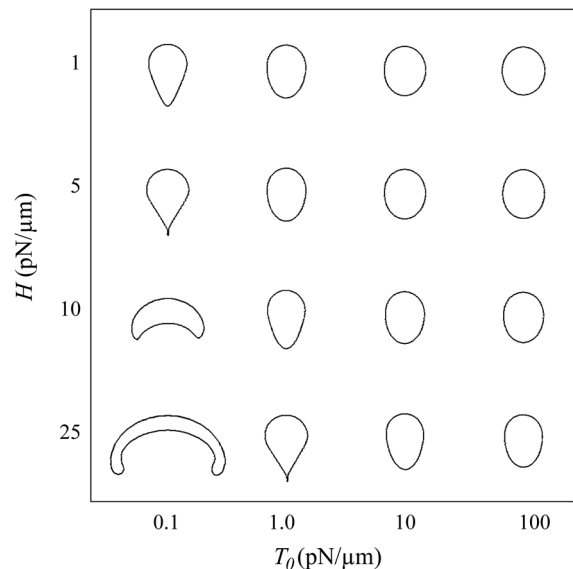


FIG. 4.2. *Mechanical model on its own. A plot of cell shapes obtained as the protrusion force magnitude, H , and the elastic modulus, T_0 , vary for a fixed, prescribed force distributed parabolically along the nodes that form the front half of the domain edge at time zero. The direction of steady-state motion of these shapes is toward the top. Viscosity was 0.01 g/cm \cdot s. Note that the shape at the bottom left is a transient shape included only for completeness. For force in the range of 1 – 25 pN/ μm , shapes obtained resemble a variety of biological cell phenotypes, from neutrophils to keratocytes.*

Varying the viscosity (of the computational fluid inside and outside the domain) has little effect on the shapes that develop. The dominant effect is to stretch the time scale over which those shapes are established. We ran tests for the range $0.01 \leq \mu \leq 10 \text{ g/cm} \cdot \text{s}$ and found similar shapes for corresponding values of the force and elasticity.

4.2. Biochemical model. We next tested the RD solver on its own. To do so, we selected a number of typical static domain shapes with various features and solved a set of pattern-forming PDEs on the given domain. We considered a circular, star-shaped, and ellipsoidal “cell” shape. Shown in Figure 4.3 are solutions to the “wave-pinning” system (2.7)–(2.9) [21]. The initial state of the system is spatially homogeneous with $a = 0.27$ and $b = 2.0$. At $t = 0$ a temporary gradient is introduced via a transient reaction term $s(x, y, t)b(x, y, t)$ that is added to (2.9), with

$$(4.1) \quad s(x, y, t) = \begin{cases} S(1+x) & \text{for } 0 < t_1 \text{ and } -1 \leq x \leq 1, \\ S(1+x) \left(1 - \frac{t-t_1}{t_2-t_1}\right) & \text{for } t_1 \leq t < t_2 \text{ and } -1 \leq x \leq 1, \\ 0 & \text{for } t_2 \leq t, \end{cases}$$

where S is a parameter that controls the magnitude of the stimulus.

This gradient stimulates a portion of the cell to become activated, meaning $a = a^+$ locally, and a traveling wave is initiated. As discussed in [21], as the level of b is reduced

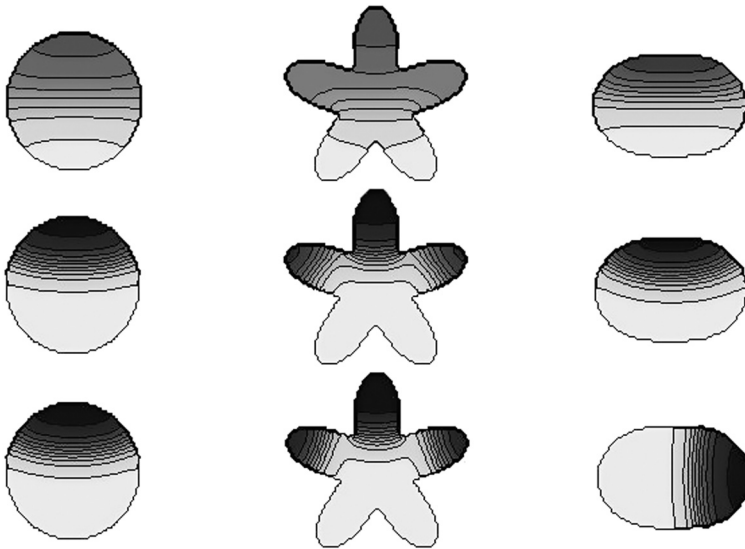


FIG. 4.3. The chemical model on its own. Solutions to the RD wave-pinning system given by (2.7)–(2.9) are shown in a variety of static geometries. The left column shows circular disks, the middle column shows a domain with convex and concave edges, and the right column shows ellipses. Time increases from top to bottom with snapshots shown at $t = 20$ s, 50 s, and 200 s. The first row is the transient phase, during which there is a traveling wave. The middle row shows the solution after the wave has been pinned. The bottom row shows the solutions after a long time. The RD-solution qualitatively captures the expected wave-like and stilled wave positions and shows the effect of concavity of the domain on the eventual location of its maximum. Reaction parameter values used were $\gamma = \delta = 1 \text{ s}^{-1}$ and $k_0 = 0.067 \text{ s}^{-1}$. Initial conditions were $a = 0.27$ and $b = 2.0$. A transient gradient term (4.1) was applied with $S = 0.07$, $t_1 = 5$, and $t_2 = 15$.

by the reaction, the speed of the wave slows and eventually stalls midway through the cell. Based on related studies [19], [21] we anticipate solutions to the RD system with relatively flat (low or high) plateaus, separated by a sharp interface. As shown in Figure 4.3, solutions have this character. In two dimensions with no-flux conditions on the boundary of the cell domain, $\partial\Omega_c$, the level curves (shown in Figure 4.3) also have to meet the curve $\partial\Omega_c$ orthogonally.

The most interesting feature to observe is that the geometry of the domain influences the nature of the solutions. This is related to phenomena studied by Marée et al. [18] showing the tendency of the kind of RD system considered here to minimize the curvature and the total length of the interface. (Marée et al. consider a related but more biochemically detailed wave-pinning system, but our simple caricature has similar properties. See also [11].) In the star-shaped domain, three distinct peaks of activation can be maintained in three “arms” of the star, a phenomenon that would not occur in the simple circular domain. We can understand this heuristically as the energy of the interface being at some local minimum. An application of similar ideas to the dynamics of plant Rho GTPases in the pavement-cells of leaves has been proposed by Grieneisen [8]. On the elliptical domain, in the short time scale, the peak of activation is in the direction of the polarizing stimulus. Once the stimulus is turned off, the polarization persists for a long time in this meta-stable configuration. Eventually however, the peak reorients to occupy the high curvature pole of the ellipse: this allows for the interface length to shrink and ultimately creates a stable configuration.

4.3. Coupled mechanical-chemical model: Self-organized cell shapes and motility. Having tested the mechanical and chemical models separately, we now combine the two and allow the forces on the cell edge to be determined directly by the solutions of the RD system. As described earlier, this leads to a number of new challenges which are associated with solving an RD system on a moving domain. The solution algorithm described earlier overcomes these challenges. We used parameter values as in Figure 4.2 for the mechanical properties and values as in Figure 4.3 for the RD system of equations (2.7)–(2.9).

Initially, the system is in equilibrium both mechanically and chemically. This means the membrane is relaxed to its equilibrium configuration and the cell is disk-shaped. Both b and a are spatially uniform inside Ω_c , and the active form is at a low steady-state level, $a(x, t) = a^-$. At $t = 0$, the same temporary gradient as was used in Figure 4.3 is introduced via the transient reaction term in the signaling system. The inactive protein b diffuses rapidly and is almost spatially uniform (but decreasing in level) as polarization proceeds. As discussed in [21], as the level of b is reduced by the reaction, the speed of the wave slows and eventually stalls midway through the cell. Here the forces are not prescribed; rather, they depend directly on the local magnitude of the active protein $a(x, t)$ through (2.10).

Figure 4.4 is a time sequence showing the evolution of the cell shape and internal chemistry. The RD system rapidly becomes polarized on a time scale of 30–60 s, and the cell starts to move and deform as a result. The teardrop-shaped outline and constant speed is then a steady-state moving solution to the combined RD-mechanical system.

To illustrate the distribution of the fluid flow and pushing force, Figure 4.5 shows two snapshots at the early stages of motion and polarization. The fluid velocity is highest at the front of the cell where the forces are largest. Circulations in the flow that are directed from the front of the cell around to the sides give rise to changes in the membrane shape. After this transient regime, the flow becomes approximately constant in

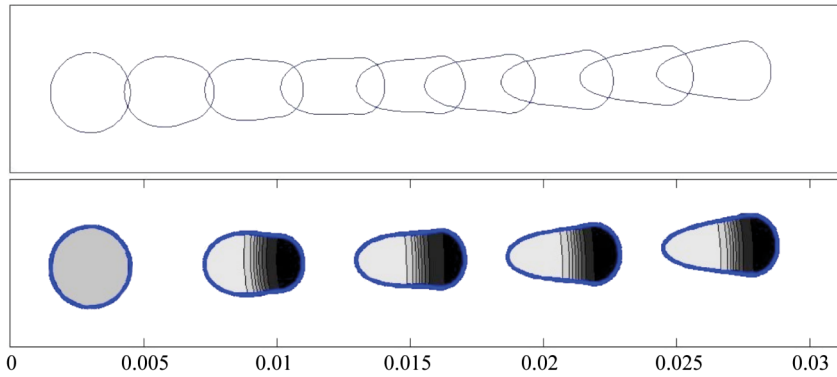


FIG. 4.4. Mechanical and chemical model combined. Forces on the “front” edge are determined by the RD system inside the evolving, moving cell domain. Top: A time sequence of cell shapes and positions, from $t = 0$ s to $t = 800$ s at intervals of 100 s starting from the circle at the left. The distance scale is in cm (10^{-4} cm = 1 μ m). Bottom: Sample shots of the same moving cell at intervals of 200 s showing the internal chemical distribution. The cell is initially disk-shaped, with homogeneous internal concentrations of a , b . At $t = 0$ a stimulus (biased toward the right) leads to chemical polarization. RD equations (2.7)–(2.9) were solved in the domain Ω_c using parameter values as in Figure 4.3. The RD system very rapidly polarizes and maintains the polarization of the cell. There is a slight change in the cell’s direction of travel over long times.

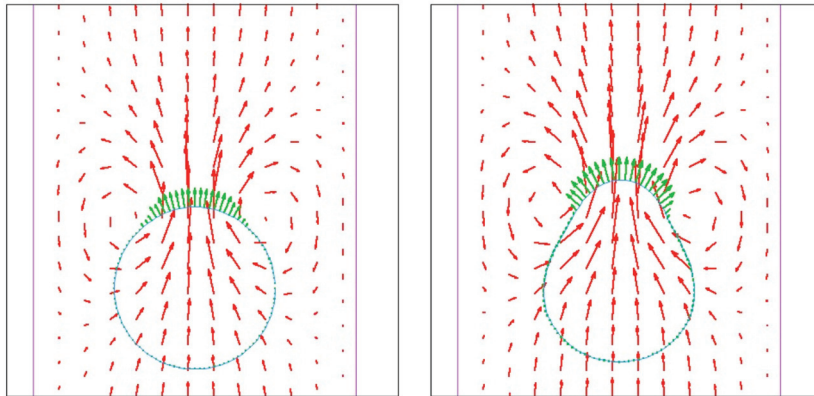


FIG. 4.5. Computational fluid speed (red arrows) and chemically regulated forces on the cell edge (green arrows) are shown at two times, close to the start of cell motion.

the vicinity of the cell. Since the membrane moves with the local fluid velocity, the steady cell shapes are observed.

We asked how varying the mechanical parameters (elastic modulus T_0 , viscosity μ , and maximal force H) affect resulting shapes for the coupled system. Recall that H sets the maximal level of force for $a = a^+ = \text{maximal chemical concentration}$. Figure 4.6 shows the results, analogous to Figure 4.2. The chemical distribution within these “cell” outlines are similar to those of Figure 4.4, with rapid polarization on a time scale of 30–60 s and steady-state shapes by about 400 s.

Figure 4.6 shows that increasing the maximal force magnitude in the chemistry-regulated cells tends to produce a transition from oval to teardrop shape. Figure 4.6 similarly shows how varying the elastic modulus and the maximal force affects the behavior. As expected, with larger elastic modulus T_0 , the steady-state shapes are rounder. A striking change seen from the mechanical to the mechanical-chemical system

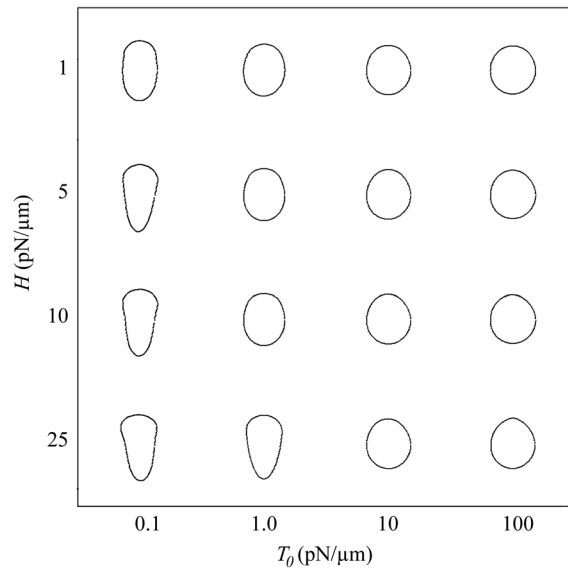


FIG. 4.6. *Steady-state shapes predicted by the full mechanical-chemical system, as in Figure 4.2 but with the leading edge regulated by the signaling system. Cell shapes are much less sensitive to the magnitudes of H in this self-organized system.*

(Figure 4.2 versus Figure 4.6) is that the self-organized chemistry makes for much more stable and persistent cell shapes. We can understand this as follows: with internal chemistry that is a self-organized polarized distribution, the nodes that are assigned protrusion forces are governed by the fraction of nodes at high concentration of the reactant a . If the front stretches, nodes previously experiencing large protrusive force leave the region of high a and no longer carry such forces. This means that the self-regulated system is much more stable and resists forming wide fronts or narrow crescent shapes.

5. Discussion. As described, the implementation of the IBM leads to a convenient way to represent forces along the interface of the cell boundary. Thus, we can capture actual forces and elastic deformations of the cell boundary (depicting the cytoskeletal cortex of the cell). This allows us to track the cell shape deformation and the motion of the cell that results from a coupling between internal RD and boundary forces. Regions of highest activity (the largest values of $a(x, t)$) develop spontaneously, determine the front, and result in distributed forces. The linear elastic cell boundary reacting to the protrusion force deforms to produce steady-state cell shape and motion.

As stated in the introduction, the goal of this study is to answer three key questions. In answer to (Q1), we have shown that the simulations with mechanical forces alone, in the absence of internal regulation, produced a variety of shapes spanning morphologies of cells from neutrophils (teardrop shapes) to keratocytes (canoe or crescent shapes) as protrusion forces increase relative to elastic forces. This suggests that the cell morphology is sensitive to the properties of the cortex and the strength of actin-generated protrusive forces, which may differ among cell types. In answer to (Q2), we showed that irregular (static) shapes can lead to multiple “captive” peaks of concentration in a signaling system that tends to otherwise form plateaus of activity. This was seen in the middle row of Figure 4.2 and echoes similar findings by Grieneisen [8]. Finally, in answer to (Q3), we found, interestingly, that once the simple regulatory system is incorporated, cell shapes are less varied and settle into a round/elliptical shape or

teardrop, with no crescents (compare Figures 4.1 and 4.5). We can explain this fact based on the nature of the RD system, (2.7)–(2.9), as previously discussed. The two key features of this system are that (1) it tends to form a sharp interface whose length tends to be minimized by its dynamics, and (2) the interface meets the boundary orthogonally (by the no-flux condition). When forces tend to push nodes apart at the cell front, the RD interface gets stretched. It reacts by finding a new configuration satisfying (1) and (2). This returns the peak of activity to nodes that are not too far apart and hence avoids the continual broadening of the front edge of the cell. The interaction of the RD with direct outward forces on the boundary thus self-regulates the shape of the front, even as the force or membrane properties are varied. Needless to say, other RD systems with interesting dynamics would, we predict, lead to other cell shape dynamics and steady-state shapes.

Applying the numerical scheme in its mechanical, chemical, and coupled variants, we found a number of intriguing results:

1. Cell shapes generated by protrusion mechanics and passive elasticity, on their own, already lead to morphologies that resemble some cell shapes.
2. Higher protrusion forces (distributed along the front half of the cell) lead to flatter, more crescent-shaped cells, whereas a larger elastic modulus leads to rounder cells (Figure 4.2).
3. The wave-pinning mechanism of Mori, Jikine, and Edelstein-Keshet [21] suffices to polarize the cell once a small bias is introduced transiently. Previously, this system was studied in a 1D setting and in a rectangular domain [11], and we explore it here for the first time in irregular and deforming domains.
4. The shape of the domain feeds back on the chemical distribution. Parts of the domain with higher curvature tend to “attract” the peaks of the active chemical (Figure 4.3.) This phenomenon was discussed by Marée in a related (but more biochemically detailed) cell biology context [18].
5. Coupling the chemical and mechanical systems leads to more stable cell shapes that are less sensitive to variations in such parameters as the magnitude of the protrusion force.

Here we concentrated on shapes produced exclusively by the wave-pinning biochemical model. These tend to form robust, broad plateaus of activity at the “cell front.” Changing the internal regulatory module is a future step of interest, as it would reveal how qualitative aspects of the regulatory system would affect qualitative aspects of cell shape.

These results can be compared with the recent work of Wolgemuth and Zajac [35]. They affect cell deformation by modifying the spatial variation in an adhesion coefficient. The primary factor that determines our cell shapes is the strength of the protrusion force relative to the membrane elasticity. We also observed a dramatic difference in shape between cells in which the pushing force on the front was imposed and cells in which the front was regulated by the chemistry. In [35] a velocity with a hyperbolic tangent profile is assumed. Prescribing this velocity is most similar to our mechanical model without any chemistry, in which we prescribed a front and the profile of protrusive forces acting upon it. In our coupled system we have gone one step further and produced a cell that polarizes in response to a transient signal and then maintains a steady shape and direction, without specifying the front of the cell in advance.

In this model we have made several simplifications, each of which can be relaxed within this modeling framework. We have used the same fluid inside and outside of the cell perimeter. The presence of an external fluid should not give the interpretation that our model cell is swimming. Indeed, we imagine that the cell is anchored to a substrate

and is crawling through a surrounding fluid. The viscosity of the external fluid could be made much lower so as to more accurately represent the environment of the cell, but the presence of the external fluid cannot be fully eliminated in this model. We have not tracked the density of the actin network explicitly. Instead we have imagined a simple situation in which the density of the network is proportional to the concentration of the actin-regulating protein and reacts immediately to changing levels of this regulator. We have taken an implicit approach to adhesion, supposing that the whole actin network adheres to the substrate to give the cell traction and the ability to generate a protrusive force through actin polymerization. Details of adhesion could be added in various ways.

Future development of such simulations will aim at addressing a number of questions. First, we will explore how contraction at the rear of the cell affects cell shape (recall that here contraction is absent). We plan to investigate a variety of internal regulatory modules, including those that have a greater diversity of spatio-temporal dynamics. As a later step, we plan to introduce a more explicit representation of the actin cytoskeleton and of the lipids and proteins that regulate it. By gradually including such features one by one, we hope to learn how tuning specific aspects of cell mechanics and biochemistry can lead to the repertoire of responses, deformation, and motility observed in real cells.

Acknowledgments. The computation was done at WestGrid. We are grateful to L. Fauci, R. Cortez, A.F.M. Marée, V. Grieneisen, A. Jilkine, and A. Mogilner for helpful discussions.

REFERENCES

- [1] W. ALT AND M. DEMBO, *Cytoplasm dynamics and cell motion: Two-phase flow models.*, Math. Biosci., 156 (1999), pp. 207–228.
- [2] D. BOTTINO AND L. FAUCI, *A computational model of ameboid deformation and locomotion*, Eur. Biophys. J., 27 (1998), pp. 532–539.
- [3] C. CHOU, Q. NIE, AND T. YI, *Modeling robustness tradeoffs in yeast cell polarization induced by spatial gradients*, PLoS ONE, 3 (2008), p. 3103.
- [4] R. CORTEZ, *The method of regularized Stokeslets*, SIAM J. Sci. Comput., 23 (2001), pp. 1204–1225.
- [5] R. CORTEZ, L. FAUCI, AND A. MEDOVIKOV, *The method of regularized Stokeslets in three dimensions: Analysis, validation, and application to helical swimming.*, Phys. Fluids, (2005), DOI [10.1063/1.1830486](https://doi.org/10.1063/1.1830486).
- [6] A. FOGELSON AND J. KEENER, *Immersed interface methods for Neumann and related problems in two and three dimensions*, SIAM J. Sci. Comput., 22 (2000), pp. 1630–1654.
- [7] M. GRACHEVA AND H. OTHMER, *A continuum model of motility in ameboid cells*, Bull. Math. Biol., 66 (2004), pp. 167–193.
- [8] V. A. GRIENEISEN, *Dynamics of Auxin Patterning in Plant Morphogenesis, A Multilevel Modeling Study*, Ph.D. thesis, Utrecht University, Utrecht, The Netherlands, 2009.
- [9] M. HERANT, V. HEINRICH, AND M. DEMBO, *Mechanics of neutrophil phagocytosis: Experiments and qualitative models*, J. Cell Sci., 119 (2006), pp. 1903–1913.
- [10] M. HERANT, W. MARGANSKI, AND M. DEMBO, *The mechanics of neutrophils: Synthetic modeling of three experiments*, Biophys. J., 84 (2003), pp. 3389–3413.
- [11] A. JILKINE, *A Wave-Pinning Mechanism for Eukaryotic Cell Polarization Based on Rho GTPase Dynamics*, Ph.D. thesis, University of British Columbia, Vancouver, BC, Canada, 2009.
- [12] Z. JOMAA AND C. MACASKILL, *Numerical solution of the 2D Poisson equation on an irregular domain with Robin boundary conditions*, Aust. N. Z. Indust. Appl. Math., 50 (2008), pp. C413–C428.
- [13] K. KEREN, Z. PINCUS, G. M. ALLEND, E. L. BARNHART, G. MARRIOTT, A. MOGILNER, AND J. THERIOT, *Mechanism of shape determination in motile cells*, Nature, 453 (2008), pp. 475–480.
- [14] J. KRISHNAN AND P. IGLESIAS, *Receptor-mediated and intrinsic polarization and their interaction in chemotaxing cells*, Biophys. J., 92 (2007), pp. 816–830.

- [15] H. LEVINE, D. KESSLER, AND W. RAPPEL, *Directional sensing in eukaryotic chemotaxis: A balanced inactivation model*, Proc. Natl. Acad. Sci. U.S.A., (2006), DOI [10.1073/pnas.0601302103](https://doi.org/10.1073/pnas.0601302103)
- [16] Z. LI AND K. ITO, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, Frontiers Appl. Math. 33, SIAM, Philadelphia, 2006.
- [17] L. MA, C. JANETOPOULOS, L. YANG, P. DEVREOTES, AND P. IGLESIAS, *Two complementary, local excitation, global inhibition mechanisms acting in parallel can explain the chemoattractant-induced regulation of PI (3, 4, 5) P3 response in dictyostelium cells*, Biophys. J., 87 (2004), pp. 3764–3774.
- [18] A. F. MARÉE, V. GRIENEISEN, AND L. EDELSTEIN-KESHET, *private communication*, 2011
- [19] A. F. MARÉE, A. JILKINE, A. DAWES, V. A. GRIENEISEN, AND L. EDELSTEIN-KESHET, *Polarization and movement of keratocytes: A multiscale modelling approach*, Bull. Math. Biol., 68 (2006), pp. 1169–1211.
- [20] H. MEINHARDT, *Orientation of chemotactic cells and growth cones: Models and mechanisms*, J. Cell Sci., 112 (1999), pp. 2867–2874.
- [21] Y. MORI, A. JILKINE, AND L. EDELSTEIN-KESHET, *Wave-pinning and cell polarity from a bistable reaction-diffusion system*, Biophys. J., 94 (2008), pp. 3684–3697.
- [22] E. P. NEWREN, A. L. FOGELSON, R. D. GUY, AND R. M. KIRBY, *Unconditionally stable discretizations of the immersed boundary equations*, J. Comput. Phys., 222 (2007), pp. 702–719.
- [23] E. P. NEWREN, A. L. FOGELSON, R. D. GUY, AND R. M. KIRBY, *A comparison of implicit solvers for the immersed boundary equations*, Comput. Methods Appl. Mech. Engrg., 197 (2008), pp. 2290–2304.
- [24] C. PESKIN, *The immersed boundary method*, Acta Numer., 11 (2002), pp. 479–517.
- [25] B. RUBINSTEIN, M. FOURNIER, K. JACOBSON, A. VERKHOVSKY, AND A. MOGLNER, *Actin-myosin viscoelastic flow in the keratocyte lamellipod*, Biophys. J., 97 (2009), pp. 1853–1863.
- [26] B. RUBINSTEIN, K. JACOBSON, AND A. MOGLNER, *Multiscale two-dimensional modeling of a motile simple-shaped cell*, Multiscale Model. Simul., 3 (2005), pp. 413–439.
- [27] J. SATULOVSKY, R. LUI, AND Y. WANG, *Exploring the control circuit of cell migration by mathematical modeling*, Biophys. J., 94 (2008), pp. 3671–3683.
- [28] J. M. STOCKIE AND B. T. R. WETTON, *Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes.*, J. Comput. Phys., 154 (1999), pp. 41–64.
- [29] W. STRYCHALSKI, D. ADALSTEINSSON, AND T. ELSTON, *Simulating biochemical signaling networks in complex moving geometries*, SIAM J. Sci. Comput., 32 (2010), pp. 3039–3070.
- [30] J. TERAN AND C. PESKIN, *Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain*, SIAM J. Sci. Comput., 31 (2009), pp. 3404–3416.
- [31] S. TLUPOVA AND R. CORTEZ, *Boundary integral solutions of coupled Stokes and Darcy flows*, J. Comput. Phys., 228 (2009), pp. 158–179.
- [32] C. TU AND C. PESKIN, *Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 1361–1376.
- [33] S. O. UNVERDI AND G. TRYGGVASON, *A front-tracking method for viscous, incompressible, multi-fluid flows*, J. Comput. Phys., 100 (1992), pp. 25–37.
- [34] A. VERKHOVSKY, T. SVITKINA, AND G. BORISY, *Self-polarization and directional motility of cytoplasm*, Curr. Biol., 9 (1999), pp. 11–20.
- [35] C. WOLGEMUTH AND M. ZAJAC, *The moving boundary node method: A level set-based, finite volume algorithm with applications to cell motility*, J. Comput. Phys., 229 (2010), pp. 7287–7308.
- [36] L. YANG, J. EFFLER, B. KUTSCHER, S. SULLIVAN, D. ROBINSON, AND P. IGLESIAS, *Modeling cellular deformations using the level set formalism.*, BMC Syst. Biol., (2008), DOI [10.1186/1752-0509-2-68](https://doi.org/10.1186/1752-0509-2-68).