2D FORCE CONSTRAINTS IN THE METHOD OF REGULARIZED STOKESLETS

ONDREJ MAXIAN AND WANDA STRYCHALSKI

msp

2D FORCE CONSTRAINTS IN THE METHOD OF REGULARIZED STOKESLETS

ONDREJ MAXIAN AND WANDA STRYCHALSKI

For many biological systems that involve elastic structures immersed in fluid, small length scales mean that inertial effects are also small, and the fluid obeys the Stokes equations. One way to solve the model equations representing such systems is through the Stokeslet, the fundamental solution to the Stokes equations, and its regularized counterpart, which treats the singularity of the velocity at points where force is applied. In two dimensions, an additional complication arises from Stokes' paradox, whereby the velocity from the Stokeslet is unbounded at infinity when the net hydrodynamic force within the domain is nonzero, invalidating any solutions that use the free space Stokeslet. A straightforward computationally inexpensive method is presented for obtaining valid solutions to the Stokes equations for net nonzero forcing. The approach is based on modifying the boundary conditions of the Stokes equations to impose a mean zero velocity condition on a large curve that surrounds the domain of interest. The corresponding Green's function is derived and used as a fundamental solution in the case of net nonzero forcing. The numerical method is applied to models of cellular motility and blebbing, both of which involve tether forces that are not required to integrate to zero.

1. Introduction

Stokes flow refers to the regime of viscous flow where inertial effects are small, and the Navier–Stokes equations simplify to the Stokes equations. For fluid-structure interaction problems in cell biology, such as an elastic red blood cell membrane deforming in capillary flow [28; 29], the small length scales of the cell diameter ($\sim 10\,\mu$m) lead to a small Reynolds number. Other important phenomena in cell biology that involve zero Reynolds number flow are cell motility [20; 38] and microorganism swimming [7; 15; 17].

Because the Stokes equations are linear, boundary integral and boundary element methods can be used to determine the velocity and pressure fields that come from a collection of forces [26; 27]. The velocity field generated from a point force is known as a *Stokeslet*. One problem that arises when using the Stokeslet in practice

is the singularity at the point where the force arises. For closed interfaces, this singularity is integrable, but careful numerical quadratures are necessary to correctly calculate the velocity and pressure [26; 27]. In [9], Cortez introduced the method of regularized Stokeslets to overcome the singularities in both the pressure and velocity expressions for forces located at scattered points. Instead of the force being applied at a point, the force is applied over a small ball of radius ϵ. The regularized Stokeslet and pressure expressions are then obtained analytically from the particular function used to represent the small ball. The method of regularized Stokeslets can also be used for closed surfaces, bypassing the associated issues with numerical quadrature [26].

It is convenient to model and simulate fluid-structure interaction problems in two dimensional domains where model parameter studies can be conducted in a computationally inexpensive manner. Data visualization is also easier in 2D than in 3D. The free space Stokes equations in 2D are actually ill-posed because the velocity obtained from the free space Stokeslet is *unbounded* at infinity when there is a nonzero net hydrodynamic force acting within the domain of flow. This contradicts the assumption in the derivation of the Stokeslet that the velocity is zero at infinity and renders the problem ill-posed [9; 21; 26]. Numerical simulations of such systems can therefore lead to unphysical spurious velocities. The phenomenon of unbounded velocities in systems with nonzero net force, especially as $\|x\| \to \infty$, is usually referred to as Stokes' paradox [38]. We emphasize that this is a unique feature of Stokes flow in 2D. In 3D, the problem is well-posed; the Stokeslet decays to zero at infinity regardless of the net forcing, so the boundary condition is satisfied and the solution is valid for any collection of forces with bounded magnitude.

One way to ensure that the 2D velocity is valid is to add conditions to the original system of equations. The most straightforward way to do this is to impose additional boundary conditions within the region of interest. The method of regularized Stokeslets was employed in [9] to simulate the flow due to a cylinder moving at an imposed velocity. In [1; 4; 12], the method of images was used to add additional Stokeslets outside of the flow domain that enforce a zero boundary condition near a plane wall. In these approaches, there is an additional constraint on the *velocity* that leads to valid solutions near the immersed objects of interest. However, if no boundary conditions within the flow domain are specified by the model of the physical system (e.g., when modeling flexible fibers in Stokes flow [5; 11; 35]), a different approach must be used.

One such approach is to enforce a constraint on the *force* rather than the velocity, in particular that the net hydrodynamic force over the entire domain be zero. Sometimes, this constraint comes naturally, such as in models of flagellar swimming [40] or fibers immersed in a background flow [5]. However, the only a priori requirement of fluid-structure interaction in Stokes flow is that the hydrodynamic force at a point

is exactly balanced by the internal and external forces on the immersed structures [22]. In fact, there are many systems with zero Reynolds number that contain force imbalances, including any system that contains tether forces or objects tied to boundaries. For example, Cortez's model of a moving cylinder [9] had a nonzero net hydrodynamic force within the domain of flow. In this case, one potential solution is to subtract the mean force from the force at each point, which automatically gives a zero-sum total force. Here we show this approach can result in nonphysical, displaced equilibrium states.

Teran and Peskin [33] treated the problem of unbalanced forces within the immersed boundary (IB) method [24; 25] by adding a unique, constant velocity throughout the periodic domain to ensure that the net force is zero for all time. In the formulation from [33], an additional constant velocity is permitted because the equations are simulated on a periodic domain, where the solution is unique up to a constant. In this case, it is required that the net force be zero [3]. The net zero force requirement in a periodic IB method presents some challenges. For example, tether forces *must* be introduced in the domain in order to simulate body forces (as the authors did in [33] when modeling peristaltic pumping). In order for the immersed structures to remain stationary, the tether spring stiffness must be large, which in turn increases the overall stiffness of the numerical scheme and the cost of the IB method formulation as a whole.

We present a method to simulate models in 2D Stokes flow with net nonzero forcing using the method of regularized Stokeslets. We accomplish this by surrounding the domain by a large circle and constraining the *mean velocity* on the circle to be zero. Given this boundary condition, we derive the corresponding Green's function and show that a mean zero velocity at the large circle can be achieved simply by adding a constant velocity to the free space Stokeslet solution throughout a large domain of flow. In this way, we avoid having to solve a linear system on the large circle (as in [38]). This observation results in an algorithm that is very straightforward to implement. After presenting our method in Sections 2 and 3, we show in Section 4 how it can be applied to 2D models of cells immersed in viscous fluid. In the process, we compare our formulation to both the explicit zero velocity condition on the large circle, e.g., from [38], and the force-free formulation obtained from subtracting the mean force at each point.

2. Mathematical framework

The steady Stokes equations in two dimensions are

$$\mu \Delta \boldsymbol{u} - \nabla p = -\boldsymbol{f}, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2}$$

where μ is the fluid viscosity, p is the pressure, $\boldsymbol{u}$ is the fluid velocity, and $\boldsymbol{f}$ is the hydrodynamic force, exactly equal to the external applied force that comes from fibers or other structures immersed in the fluid [22]. We begin by summarizing the method of regularized Stokeslets [9] for computing $\boldsymbol{u}$ and p from (1) and (2). Then we present the modification for addressing Stokes' paradox.

2.1. *Method of regularized Stokeslets.* In the method of regularized Stokeslets, a force of strength $\boldsymbol{f_0}$ is distributed primarily (but not entirely) over a small ball centered on a point $\boldsymbol{x_0}$, so that

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f_0}\phi_\epsilon(\boldsymbol{x} - \boldsymbol{x_0}). \tag{3}$$

The Stokes equations can be solved with the force in (3) to derive the resulting velocity and pressure from a given "blob" or "cutoff" function ϕ_ϵ. For example, if

$$\phi_\epsilon(\boldsymbol{x}) = \frac{3\epsilon^3}{2\pi(\|\boldsymbol{x}\|^2 + \epsilon^2)^{5/2}}, \tag{4}$$

then

$$p^\epsilon(\boldsymbol{x}, \boldsymbol{x_0}) = \frac{1}{2\pi}(\boldsymbol{f_0} \cdot (\boldsymbol{x} - \boldsymbol{x_0}))\left(\frac{r_0^2 + 2\epsilon^2 + \epsilon\sqrt{r_0^2 + \epsilon^2}}{(\sqrt{r_0^2 + \epsilon^2} + \epsilon)(r_0^2 + \epsilon^2)^{3/2}}\right) \tag{5}$$

and

$$\boldsymbol{u}^\epsilon(\boldsymbol{x}, \boldsymbol{x_0}) = -\frac{\boldsymbol{f_0}}{4\pi\mu}\left(\ln(\sqrt{r_0^2 + \epsilon^2} + \epsilon) - \frac{\epsilon(\sqrt{r_0^2 + \epsilon^2} + 2\epsilon)}{(\sqrt{r_0^2 + \epsilon^2} + \epsilon)\sqrt{r_0^2 + \epsilon^2}}\right)$$
$$+ \frac{1}{4\pi\mu}(\boldsymbol{f_0} \cdot (\boldsymbol{x} - \boldsymbol{x_0}))(\boldsymbol{x} - \boldsymbol{x_0})\frac{\sqrt{r_0^2 + \epsilon^2} + 2\epsilon}{(\sqrt{r_0^2 + \epsilon^2} + \epsilon)^2\sqrt{r_0^2 + \epsilon^2}} \tag{6}$$

are the pressure and velocity that result from the force in (3), where $r_0 = \|\boldsymbol{x} - \boldsymbol{x_0}\|$. The derivation of these expressions can be found in [9]. Notice that for $r_0 \gg \epsilon$, the standard Stokeslet expressions [26] are recovered,

$$p(\boldsymbol{x}, \boldsymbol{x_0}) = \frac{\boldsymbol{f_0} \cdot (\boldsymbol{x} - \boldsymbol{x_0})}{2\pi r_0^2}, \tag{7}$$

$$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{x_0}) = -\frac{\boldsymbol{f_0}}{4\pi\mu}\ln r_0 + (\boldsymbol{f_0} \cdot (\boldsymbol{x} - \boldsymbol{x_0}))\frac{(\boldsymbol{x} - \boldsymbol{x_0})}{4\pi\mu r_0^2}. \tag{8}$$

The pressure and velocity resulting from a collection of forces $\boldsymbol{f_k}$ spread around a collection of points $\boldsymbol{x_k}$ is simply a superposition of the results from (5) and (6). It is easy to see that if $\sum_k \boldsymbol{f_k} \neq \boldsymbol{0}$, the velocity in (6) or (8) is unbounded as $\|\boldsymbol{x}\| \to \infty$, and the boundary conditions $\boldsymbol{u} \to \boldsymbol{0}$ are not satisfied as $\|\boldsymbol{x}\| \to \infty$.
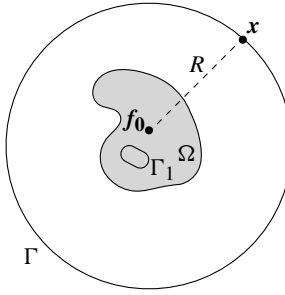
Figure 1. Ω denotes the region bounding immersed interfaces and/or point forces. Γ_1 indicates an immersed interface and $\boldsymbol{f_0}$ denotes a point force at $\boldsymbol{x_0}$ enclosed by Ω. A circle of radius R is denoted by Γ.

2.2. *Modification for nonzero net force.* Suppose that all of the forces $\boldsymbol{f_k}$ and immersed interfaces in a model system are located within a domain Ω (see Figure 1). We note that Ω is not necessarily an immersed interface, but rather a sort of "bounding box" in which all of the forces are contained. If there are no other boundary conditions within Ω, such as a specified velocity on a curve Γ_1 contained within the domain, the problem is ill-posed and the free space solution for the velocity in (6) is not valid, even near Ω. To construct a mathematically valid solution inside some space containing Ω, we surround Ω with a large circle, denoted by Γ with radius R (illustrated in Figure 1). One approach from [38] is to enforce a zero velocity boundary condition at every point on the discretized circle. The resulting linear system obtained from (6) is well-conditioned when ϵ is of magnitude less than or equal to the discrete point spacing on the large circle. The system can be solved for the forces required to obtain a zero velocity on the large circle. These forces can then be used in (6) to compute the velocity at locations enclosed by the smaller domain Ω. We note that this requires a linear system to be solved at each time value when simulating a model of a dynamic process.

We take a slightly different approach. Instead of requiring $\boldsymbol{u}|_\Gamma = \boldsymbol{0}$ at every point, we solve Stokes equations with a slightly weaker boundary condition, that the average value of $\boldsymbol{u}$ on Γ be zero: $\langle \boldsymbol{u} \rangle |_\Gamma = \boldsymbol{0}$. Notice that as $R \to \infty$, the regularized Stokeslet solution in (6) converges to the free space Stokeslet in (8) and the velocity on the large circle is approximately *constant* because of the dominance of the radially symmetric first term in (8). If the velocity on the large circle is constant, the two conditions are equivalent. Imposing the mean velocity condition allows us to add an extra velocity $\boldsymbol{u}^R$ throughout the domain so that the average velocity on the large circle is zero.

We begin by deriving the velocity $\boldsymbol{u}^R$ in the case of a single point force followed by the generalization to multiple forces by superposition. Let $\boldsymbol{f_0}$ be the force at a point $\boldsymbol{x_0}$ in Ω. Let $\boldsymbol{x}$ be a point on the large circle Γ (see Figure 1), and let

$r_0 = \|x - x_0\|$. Since x is on a large circle with arbitrarily large radius R, $r_0(x) \gg \epsilon$ for all $x \in \Gamma$, and we can represent the velocity at the large circle using the standard Stokeslet. Thus, the velocity at x due to the force f_0 applied at x_0 is given by (8).

Using s as the arclength parameter and treating $r_0 = R$ as constant, the average value of $u(x, x_0)$ is

$$\langle u(x, x_0) \rangle = \frac{1}{2\pi R} \int_\Gamma \left(-\frac{f_0}{4\pi\mu} \ln r_0 + (f_0 \cdot (x - x_0)) \frac{x - x_0}{4\pi\mu r_0^2} \right) ds$$

$$= \frac{1}{2\pi R} \left(-\frac{f_0}{4\pi\mu} \ln R \int_\Gamma ds + \frac{1}{4\pi\mu R^2} \int_\Gamma (f_0 \cdot (x - x_0))(x - x_0) \, ds \right)$$

$$= -\frac{f_0}{4\pi\mu} \ln R + \left(\frac{1}{2\pi R} \right) \frac{1}{4\pi\mu R^2} \int_\Gamma (f_0 \cdot (x - x_0))(x - x_0) \, ds. \quad (9)$$

The last equality used the fact that $\int_\Gamma ds = 2\pi R$. Computing the second integral, we begin by changing to an angle parametrization of Γ via $s = R\theta$,

$$\frac{1}{2\pi R} \int_\Gamma (f_0 \cdot (x - x_0))(x - x_0) \, ds = \frac{1}{2\pi R} \int_0^{2\pi} (f_0 \cdot (x - x_0))(x - x_0) R \, d\theta. \quad (10)$$

Because the circle is rotationally invariant, we can place the x axis on the same direction as f_0 without loss of generality. Therefore, let $f_0 = f \binom{1}{0}$. Furthermore, in the limit $R \to \infty$, $x - x_0 = x = \binom{R\cos\theta}{R\sin\theta}$. Then (10) simplifies to

$$\frac{1}{2\pi R} \int_0^{2\pi} (f_0 \cdot (x - x_0))(x - x_0) R \, d\theta = \frac{1}{2\pi} \int_0^{2\pi} fR\cos\theta \binom{R\cos\theta}{R\sin\theta} \, d\theta \quad (11)$$

$$= \frac{R^2 f}{2} \binom{1}{0} = \frac{R^2 f_0}{2}. \quad (12)$$

Substituting (12) into (9), we have the average velocity over the large circle given the force f_0 at x_0 as

$$\langle u(x, f_0) \rangle = \frac{f_0}{4\pi\mu} \left(\tfrac{1}{2} - \ln R \right). \quad (13)$$

Our goal is to impose a boundary condition on the large circle. Rather than impose a boundary condition pointwise, we impose a weaker condition on the *mean velocity* on the large circle, namely that the mean velocity is zero. It follows immediately that this can be done by subtracting the constant velocity in (13) throughout the domain of flow. The additional velocity due to a point force f_0 is therefore

$$u^R(f_0) = -\frac{f_0}{4\pi\mu} \left(\tfrac{1}{2} - \ln R \right). \quad (14)$$

In the case of multiple time-dependent forces $f_k(t)$ (for example, forces that come from an interface such as Γ_1 in Figure 1), the constant velocity is simply the

superposition of velocities from (14):

$$\boldsymbol{u}^R(t) = \sum_{k=1}^{N} \boldsymbol{u}^R(\boldsymbol{f_k}(t)) = \sum_{k=1}^{N} -\frac{\boldsymbol{f_k}(t)}{4\pi\mu}\left(\tfrac{1}{2} - \ln R\right). \qquad (15)$$

This velocity is added throughout the domain of flow to ensure that $\langle \boldsymbol{u}(t)\rangle|_\Gamma = 0$. In effect, the solution from (6) and (15) together form the *Green's function for Stokes equations with a mean zero boundary condition on the circle of radius R*. We note that the Green's function would change if the domain was surrounded by a large square with edge length $2R$ as opposed to a circle of radius R. However, any geometry, such as a square, can be thought of as bounded by two concentric large circles (for a square centered at 0 with edge length $2R$, the points on the square are between concentric circles of radius R and $R\sqrt{2}$). Because the velocity dependence on R is weak for large R (the derivative of $\boldsymbol{u}^R$ scales with $1/R$), altering the geometry of the boundary results in small changes in the velocity on the bounding region.

It is no coincidence that (15) expresses the average value of the velocity due to forces of strength $-\boldsymbol{f_k}$. By adding the constant velocity in (15), we are effectively adding a force on the large circle that *has the effect of adding an equal and opposite force within the region of interest* Ω. Meanwhile, the addition of a constant velocity throughout the domain results in a relative velocity profile that is unchanged from that computed by (6). Our approach contrasts with adding more Stokeslets at arbitrary locations in the domain Ω, which could be problematic because the relative local profile (and subsequent physical conclusions) are dependent on the locations of the additional Stokeslets. However, our approach of imposing forces on the large circle does result in the introduction of a much larger length scale in the problem; the length scale becomes R, the radius of the large circle, instead of the length scale of the immersed objects.

The addition of this constant velocity has no effect on the pressure profile calculated from (5). Because a constant velocity is added, no pressure gradient is generated within the domain. Equivalently, our addition of a constant velocity is a shortcut around explicitly adding forces on the large circle that enforce the zero boundary condition exactly (e.g., [38]). Due to the nature of the pressure solution in (5) (i.e., that it decays as $1/\text{length}$), the additional forces from the boundary condition on the large circle have no effect on the local pressure profile for large R.

We also note that the choice of blob in (4) yields the resulting analytical expressions for the regularized pressure and velocity in (5) and (6), respectively. We present these expressions derived from ϕ_ϵ because we use them in our numerical simulations. The derivation of $\boldsymbol{u}^R$ is independent of the regularized Stokeslet because it is derived from the true free space Stokeslet. The large circle Γ is assumed to be far enough from the domain of interest that the two are equivalent.

The method we present here is therefore compatible with any regularization kernel, including compactly supported immersed boundary kernels [2]. However, IB kernels are generally used over a periodic fluid grid, not over free space, and so the method of [33] is more appropriate in that context.

3. Discretization

In general, we begin with a collection of N points $\boldsymbol{x}_k$ with forces $\boldsymbol{f}_k$ in some domain Ω. At each time step, we compute the velocity of each point $\boldsymbol{x}_i$ as

$$\boldsymbol{u}(\boldsymbol{x}_i) = \boldsymbol{u}^R + \sum_{k=1}^{N} \boldsymbol{u}^\epsilon(\boldsymbol{x}_i, \boldsymbol{x}_k), \tag{16}$$

where $\boldsymbol{u}^R$ is given by (15) and $\boldsymbol{u}^\epsilon(\boldsymbol{x}_i, \boldsymbol{x}_k)$ is given by (6). Because $\boldsymbol{u}^R$ is constant throughout the domain, the calculation in (16) is $\mathbb{O}(2N + (2N)^2)$ operations. The first $\mathbb{O}(2N)$ operations arise from the computation of the constant additional velocity in (15). The second $\mathbb{O}((2N)^2)$ operations come from computing the regularized velocities in (6) for all of the points.

We compare this operation count to alternative formulations. Suppose that the large circle was discretized with M points and the forces on the large circle were solved for explicitly, as in [38]. This calculation is a $2M \times 2M$ linear solve and requires $\mathbb{O}((2M)^2)$ operations using GMRES or $\mathbb{O}((2M)^3)$ operations if done directly. In addition, the calculation of the added velocities at each point in the domain from the forces on the large circle requires another $\mathbb{O}(2NM)$ flops, and it is unclear how to choose the number and location of the M points. Alternatively, the addition of more Stokeslets in a method similar to the method of images [1; 4; 12] would require $\mathbb{O}(2SN)$ operations to compute the added velocity, where S is the number of added Stokeslets. Our added velocity is computed in $2N$ flops, making it much more efficient than any of these alternatives.

3.1. *Choosing the radius*. Central to our method is the assumption that the velocity computed from (8) on the large circle Γ is relatively constant. The validity of this assumption dictates a lower bound on R. In order to test the variation of the velocity on the large circle Γ, we first impose a force of $\boldsymbol{f_0} = \binom{1}{0}$ at the origin. Next, we use (8) to compute and measure the velocity from the Stokeslet (i.e., the velocity without the addition of $\boldsymbol{u}^R$) on the large circle.

Specifically, we discretize the circle Γ with $N = 100$ points and quantify velocity variability by defining

$$\sigma_u(R) = \max_i \left| \frac{u_x^i - \bar{u}_x}{\bar{u}_x} \right|. \tag{17}$$
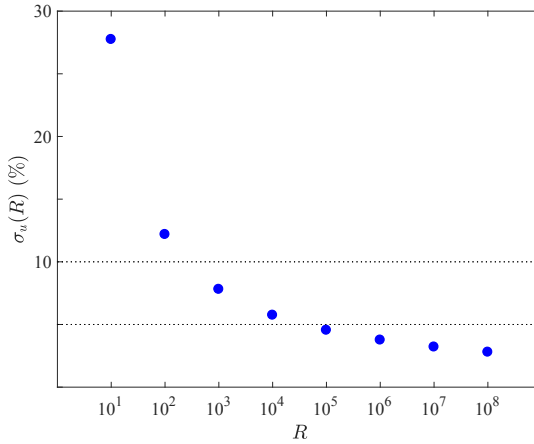
Figure 2. Variation of the velocity on the large circle Γ for different values of R. Dotted lines indicate 10% and 5% variation, $\sigma_u(R)$.

Here the index i runs from 1 to $N = 100$, u_x^i refers to the velocity in the x direction (the direction of the force) at point i, and $\bar{u}_x$ refers to the arithmetic mean of u_x taken over Γ.

Figure 2 shows the values of $\sigma_u(R)$ (as a percentage) for different values of R. We observe a variation in the velocity of less than 10% for $R \geq 10^3$ and a variation of less than 5% for $R \geq 10^5$.

With this in mind, we specify the lower limit on R, $R \geq 10^3$. We can find an upper limit on R based on the Reynolds number. Let v and L be the relevant velocity and length scales. Our systems have values of viscosity μ and density ρ of the same magnitude as water, so that $\mathrm{Re} = \rho v L / \mu = 10^6 v L$ is the relevant Reynolds number. Our model systems are from applications in cell biology, so the relevant velocity scale is in μm/s. Therefore, we take $v = 10^{-6}$ m/s and $\mathrm{Re} = L$, where L is the relevant length scale. In order for Stokes flow to be valid, we need ($\mathrm{Re} \ll 1$), which we define to be $\mathrm{Re} \leq 0.1$. Then the relevant length scale cannot exceed $0.1\,\mathrm{m} = 10^5\,\mu$m. Because we have confined the domain and effectively introduced forces on the large circle, the radius of the large circle is now the largest relevant length scale, and we have determined an upper bound on R, $R \leq 10^5\,\mu$m. We note that this upper bound may change depending on the characteristic time and length scales used to compute the Reynolds number, but it is straightforward to derive it as we have here. Thus, we have determined in general that $10^3 \leq R \leq 10^5$. For the examples in Section 4, we choose $R = 10^3\,\mu$m to ensure the validity of the Stokes equations for these systems. This value of R results in a velocity variation from (17) that is less than 10%. In addition, we did not find any additional stiffness when using R in this range (for larger R, e.g., $R = 10^{16}$, the velocities in (15) would increase, thereby increasing the overall problem stiffness).
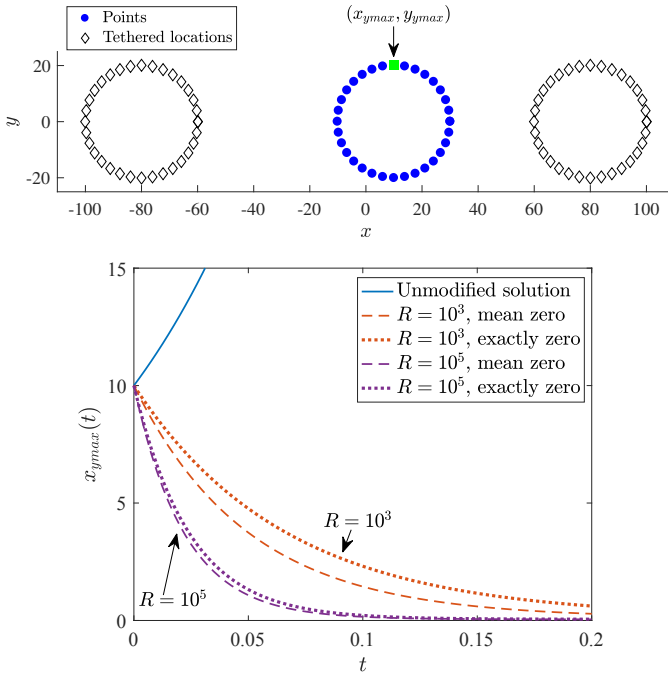
4. Examples

The motivation for this work is *tether forces* that arise in the modeling of some biological systems. These are forces that penalize displacement from an initial or resting configuration; points on an immersed object are physically *tethered* to other points in 2D space. We begin by considering a simplified system of tethered particles. This motivating example establishes the need for the additional velocity in (15). We then present a model of a cell motility problem where tether forces are useful for modeling the cell's external environment. We conclude by analyzing a boundary integral model of cellular blebbing with nonzero net forcing that has already been used for modeling bleb initiation and amoeboid cell motility [14; 19; 20]. In all cases, all of the objects are flexible so that no boundary conditions are provided from the physics of each model system.

4.1. *A motivating example.* We are interested in modeling the motion of a cell through a viscoelastic structure called the extracellular matrix (ECM). For example, our model of the ECM represents a lattice of collagen fibrils immersed in a viscous fluid. Elasticity of the ECM can be modeled in 2D by a lattice of points that are tied to specified reference points by springs. In order to demonstrate why our methodology is key for modeling this process, we introduce a set of $N = 32$ points distributed on a circle of radius r, shown in Figure 3, top. The points (solid blue dots) are centered at $(10, 0)$ and are tethered to two sets of fixed points (hollow black diamonds) centered at $(80, 0)$ and $(-80, 0)$. For simplicity, we set

$$F_i(t) = -k_{\text{teth}}(X_i(t) - X_i^R + X_i(t) - X_i^L), \qquad (18)$$

where X_i^R and X_i^L stand for the position of point i on the right and left fixed circles, respectively. The parameters for the system are $\mu = 1\,\text{Pa s}$, $k_{\text{teth}} = 1\,\text{pN}/\mu\text{m}^2$, $\epsilon = 2\pi r/N =$ the point spacing (although the dynamics are independent of the parameters μ, k_{teth}, and ϵ). The initial configuration results in a force imbalance, with a net force to the left (negative horizontal direction) on the set of moving points. Physically, we expect the points to move in the negative horizontal direction and approach their equilibrium position exactly between the two fixed fibers, i.e., centered at $x = 0$. However, this is not always what occurs when using (6) to update the velocities.

Consider (6). The dominant part of the first term is $-\ln r = -\ln\|x - x_0\|$, which results in a velocity that goes in the direction *opposite* the force. The second term in (6) is $\mathbb{O}(1, \epsilon^{-2})$ and contributes to the velocity in the same direction as the force. In order for the dynamics to match our physical intuition, the cumulative contribution of the second term at each point must be greater than that of the first term, so either $\ln r \approx 1$ or $\epsilon \ll 1$. Thus, as r becomes large, we expect unphysical behavior. For an r value as small as $r \approx 20$ (determined empirically), no value of ϵ (larger than

Figure 3. Motivating example of a system of points tethered in place, initially out of equilibrium. Top: initial configuration of the system. The solid blue dots show the points, while the unfilled black diamonds show the corresponding tethering locations. Bottom: x coordinate of the point with largest y coordinate, x_{ymax} (filled green square in top). Without any corrections, the motion is in the positive horizontal ($+x$) direction independent of number of points and ϵ (blue line). This nonphysical behavior can be corrected by adding the velocity u^R in (15). We show data for the values $R = 10^3$ (orange) and $R = 10^5$ (purple). Dashed lines give the dynamics for the mean velocity subtraction in (16). Dotted lines show dynamics when the boundary condition $u = 0$ is exactly enforced on a large circle of radius R discretized with $N = 100$ points.

machine epsilon) yields physical results. Figure 3, bottom, shows the horizontal (x) position of the point with the largest y coordinate over time (marked with a green square as (x_{ymax}, y_{ymax}) in Figure 3, top), where the velocity is computed by (6). We observe unphysical motion in the positive horizontal direction (blue line), moving the points to the right and resulting in an increased force imbalance as time increases. While this behavior *dominates* for large values of r, it is also present for smaller r and needs to be corrected to give proper, physically correct simulation results.

Our solution with the additional velocity given in (15) gives the expected behavior. For the initial configuration in Figure 3, top, the motion of the points computed with the velocity in (16) shifts the points in the negative horizontal direction (left), allowing them to approach their steady state positions at $x = 0$. Specifically, the horizontal component of the velocity at the origin can be decomposed into $u^\epsilon \approx 125$

and $\boldsymbol{u}^R \approx -325$ when $R = 10^3$, where the contribution of $\boldsymbol{u}^R$ is necessarily greater to obtain the correct physical motion. Figure 3, bottom, shows the horizontal position of the point with the largest y coordinate for values of R that satisfy our derived bounds, $R = 10^3$ (orange lines) and $R = 10^5$ (purple lines). Dashed lines show the solution obtained from (16), and dotted lines show the solution obtained from discretizing the circle of radius R with $N = 100$ points and explicitly enforcing $\boldsymbol{u} = \boldsymbol{0}$ at those points by determining the additional forces $\boldsymbol{f}$ on the large circle via solving a linear system of equations. The linear system of equations in this case has the form $U = \boldsymbol{M}\boldsymbol{F}$, where $\boldsymbol{M}$ is a dense $2N \times 2N$ matrix. To get an exact solution, we solve this directly with LU factorization, although it could also be done with GMRES.

We observe that the mean velocity solution gives faster dynamics than the discretized large circle solution. As R increases, the velocity on the large circle approaches a constant value and the solutions approach the same curve (as shown previously in Figure 2). Further, the maximum difference of $1.0\,\mu$m in the x-coordinate between the two curves for $R = 10^3$ is only 5.2% of the smallest system length scale r, and for $R = 10^5$ this difference decreases to 1.8%.

The choice for the value of R in (15) affects the velocity and the dynamics of the system. However, we note that additional forces solved for by enforcing additional boundary conditions on any geometry C, i.e., $\boldsymbol{u}|_C = \boldsymbol{0}$, would also affect dynamics of the system. For the values of R within the range $10^3 \leq R \leq 10^5$, the *steady-state* behavior of our model and relevant timescales are shown in Figure 3, bottom, to be nearly identical, with the timescales differing by about a factor of 2. If accurate transient results are desired, the value of R can be tuned to give dynamics that fit within the relevant timescales, with the caveat that increasing R leads to a larger length scale.

We note another feature of this example: the initial force on the configuration shown in Figure 3, top, is *uniform across all of the points*. Suppose one wanted to treat the force imbalance in this example by subtracting the mean force from each point, so the total force sums to zero. Because each point has the same force on it, subtracting the mean force gives zero force and zero velocity at every point. We have therefore shown that subtracting the mean force can create artificial equilibrium configurations. This phenomenon occurs not just in this simple example, but also in a more complicated model of cell motility as discussed in Section 4.2. Thus, while subtracting the mean of the forces maintains the relevant system length scales, doing so can introduce errors in the resting position of the system. Which avenue to choose in this trade-off is application dependent.

4.2. *Model of cell motility.*

Cell motility is an essential process for wound healing, cancer metastasis, and immune responses [30]. In three dimensions, a cell can utilize multiple mechanisms to migrate through the surrounding extracellular matrix

(ECM) [36; 37; 42]. The ECM is a dense network of collagen fibers [13, Figure 1]. Previous studies have used 2D agent based/finite element models [36; 37] to study the effectiveness of bleb-based and protrusion-based mechanisms in different ECM environments. In [42], the authors simulated a variety of mechanisms on a cell with a rigid nucleus via force balance equations. Our goal is to extend this model to a flexible nucleus, where the fluid-structure coupling is treated explicitly via the method of regularized Stokeslets.

We focus here on the movement of the cell via a finger-like protrusive mechanism [18; 42]. In this mechanism, the elastic cell cortex generates random actin-based protrusions. The cortex is the thin layer of the actin cytoskeleton that is attached to the cell membrane. For the purposes of our model, we consider the cortex to represent the combined membrane and cortex. Actin protrusions from the cortex are allowed to bind to ECM fibers. Upon binding, the cortex stiffens, which allows the cell to "pull" on the ECM by generating traction forces on the tip of the protrusion [18; 42]. Here we develop a model of this mechanism in 2D to gain insight into how ECM stiffness affects the ability of the cell to migrate before developing a computationally expensive 3D model.

We consider a 2D cross-section of a cell migrating through an ECM consisting of fibers immersed in fluid. The cell and nucleus are modeled as thin 1D elastic boundaries. The ECM consists of long thin fibers in 3D, and we model the cross section of one fiber as a regularized point force in our 2D model. For the cortex and nucleus, fiber elasticity gives the force density (in $pN/\mu m^2$) on a given configuration by

$$\boldsymbol{F}_{n/c}^{\text{el}} = \frac{\partial}{\partial s}(T_{n/c}\boldsymbol{\tau}), \tag{19}$$

where n/c stands for the nucleus or cortex, s is the reference arclength variable, $\boldsymbol{\tau} = \boldsymbol{X}_s/\|\boldsymbol{X}_s\|$ is the unit tangent vector, and

$$T_{n/c} = k_{n/c}(\|\boldsymbol{X}_s\| - 1) \tag{20}$$

is the fiber tension. Here $k_{n/c}$ ($pN/\mu m$) represents the stiffness of the nucleus/cortex. At the beginning of our simulations, we choose the cortex to be relatively soft with $k_c = 1\,pN/\mu m$ and the nuclear boundary to be much stiffer, $k_n = 50\,pN/\mu m$ [16]. We take the diameter of the cortex to be 1 μm and the diameter of the "nucleus" to be 0.9 μm, with the latter taken to be large to model effective elasticity of the cytoplasm.

We also discretize the cortex with $N_c = 80$ points and the nuclear boundary with $N_m = 40$ points. Using this discretization, one can numerically approximate derivatives in (19) and (20) via centered differences to obtain a force density at each point in $pN/\mu m^2$, then multiply by the reference point spacing to obtain a force, $\widehat{\boldsymbol{F}}_{n/c}^{\text{el}}$ in $pN/\mu m$ at each point on the nucleus/cortex.

The ECM can be represented in a cross-sectional sense as an array of points in space with some characteristic length spacing. In this section, we keep the spacing constant and fix it to be on average the same as the diameter of the cell, so that the cell is not sterically hindered from passing through the ECM. Future work will focus on the effect of ECM density in a more rigorous context; our goal here is instead to show the effect of matrix stiffness at constant fiber density.

We therefore generate 20 ECM nodes that are approximately spaced by the cell diameter on a 4×4 box, shown as blue points in Figure 4a. We triangulate this set of points, with each edge representing a spring that connects two ECM nodes (dashed black lines in Figure 4a). Let k_{teth} (pN/μm^2) denote the stiffness of these springs. Then the time-dependent force (in pN/μm) on each ECM node is given by

$$\widehat{\boldsymbol{F}}_{\text{ECM}}^{j}(t) = -k_{\text{teth}}\left(\boldsymbol{X}^{j}(t) - \boldsymbol{Z}^{j} + \sum_{i \in \mathcal{N}(j)} (\boldsymbol{X}^{j}(t) - \boldsymbol{X}^{i}(t)) \right). \qquad (21)$$

Here $i \in \mathcal{N}(j)$ denotes a point i which is a neighbor of point j, in the sense that the nodes are connected by an edge in the Delaunay triangulation (black dotted lines in Figure 4a). We note also the presence of an anchoring (tether) node, $\boldsymbol{Z}^{j}$, whose purpose is to make sure the network stays in place dynamically. Without linking the nodes to reference nodes $\boldsymbol{Z}^{j}$, the force function in (21) would be translation-invariant, and the entire network of nodes would be free to slide away from the cell without penalty. We can compute $\boldsymbol{Z}^{j}$ for each node by setting the force at $t = 0$ in (21) to zero and solving for $\boldsymbol{Z}^{j}$. This is desired physically for the cell to migrate relative to the ECM. We have therefore determined the forces on the nucleus, cortex, and ECM that need to be computed at each time point and passed to (16).

We note that the use of points for the ECM fibers necessitates the use of a regularized method, as the velocity due to a point force is technically infinite at that point. We set $\epsilon = 0.075 \, \mu$m in the regularized equations, so that each point has an effective radius around it that is much smaller than the radius of the cell. We note that this value for ϵ is also the approximate spacing between the discrete nuclear and cortical points, which is one criterion for choosing ϵ [9; 10].

The overall simulation algorithm is as follows. Draw from a uniform distribution a point j on the front edge of the cortex (representing cell polarization) and suppose that actin polymerizes at that point. We apply a force density of strength $f_0 = 500 \, \text{pN}/\mu$m^2 in the normal direction at point j and a force density of strength $f_0/2$ in the normal direction at points $j - 1$ and $j + 1$. Importantly, the cell physically cannot generate any net force on the fluid, so we spread the equal and opposite force over the other $N_c - 3$ cortex points. The force distribution on the cortex is shown in Figure 4b. We note that the only effect of f_0 is to set the timescale of migration, and we are concerned with the *relative* timescale across different ECM stiffnesses (so that the choice of f_0 is arbitrary). For this reason we also set $\mu = 1$ Pa s for simplicity.
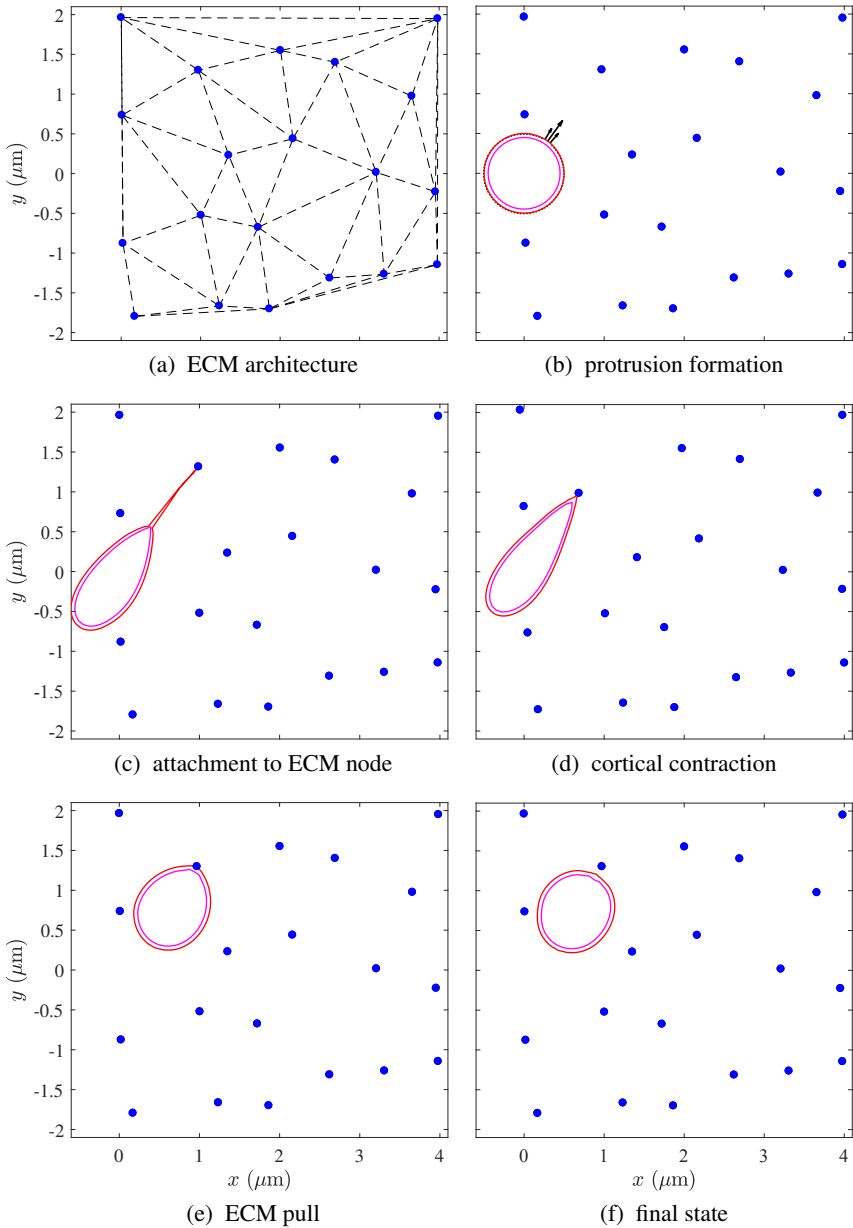
Figure 4. One cycle of a cell migrating via a finger protrusion mechanism through an ECM matrix of elastic nodes with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$. (a) The structure of the ECM, which has 20 nodes (blue points) that are linked together by springs (dashed black lines). (b–f) The dynamic process of cell migration. (b) A protrusion forms on the cell surface. (c) The protrusion binds to a node. (d) The cortical stiffness increases, pulling the node inward. (e) The dynamic balance between elasticity of the cell and ECM elasticity pulls the cell towards the ECM node's resting position. (f) The cell releases the node and is ready to form another protrusion.
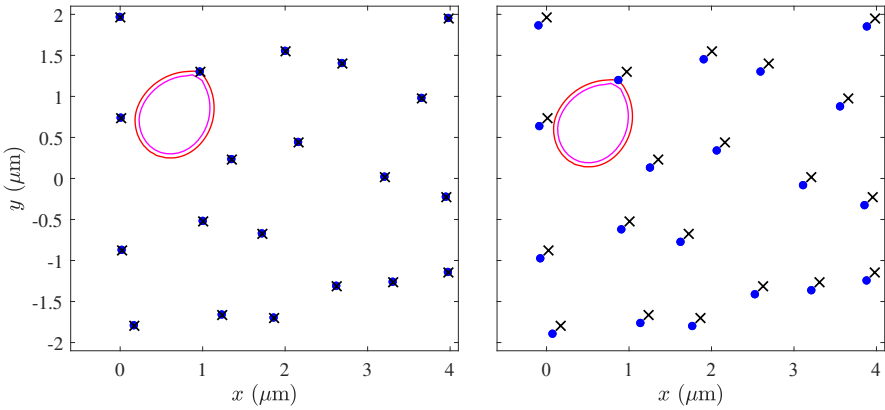
Figure 5. Final states for one cycle of the cell motility problem with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$ and (left) $R = 10^3$ and (right) sum of forces being zero via mean subtraction. Subtracting the mean of the force has created an artificial equilibrium state in the right, where each of the nodes has been displaced from its original position (black x's) by some constant amount. This does not occur in the left.

We then allow the cell protrusion to grow by evolving the system in time until the protrusion tip comes into contact with a node. By contact, we mean that the discrete points are a distance 2ϵ or less from each other, so that their "blob" functions are in contact. Once the discrete points come within a distance 2ϵ of each other, the protrusion tip binds to the node (shown in Figure 4c), and the cortex becomes stiffer by a factor of 100 to model the increased traction at the protrusion tips seen in [18]. The increased stiffness causes the cortex to rapidly become rounder. Since the cortex is attached to the node, it then pulls the node inward as shown in Figure 4d. As the node is pulled in, it generates a force in the opposite direction due to elasticity of the ECM (the node's resting configuration is its initial configuration in the ECM lattice). These forces balance dynamically, so that as the cortex becomes more round, the cortical force due to elasticity decreases, which in turn allows the force on the ECM to decrease, thereby pulling the entire cell and node back towards the initial position of the node. In the final state, Figure 4e, the cell is round and the node returns to a point close to its initial position. At this time, the node detaches from the cell by moving a distance 2ϵ away in the normal direction, as shown in Figure 4f, and the process can then repeat. We define this entire process as one cycle.

In this application, the anchor ECM nodes $\mathbf{Z}^j$ create a net force in the domain. As we observed in Section 4.1, handling this imbalance by subtracting the mean force at each node can create nonphysical translated equilibrium configurations. Figure 5 shows the final configuration when the system velocity has dropped below ϵ for a migrating cell in an ECM with $k_{\text{teth}} = 50\,\text{pN}/\mu\text{m}^2$ for (left) a system simulated using (16) and (right) a system simulated with zero net forcing via subtracting the

mean force at each node. A shift in the entire domain in the negative horizontal and vertical directions is shown in Figure 5, right. Physically, we expect the nodes to return to their initial configuration in Figure 4a (marked with black x's in Figure 5).

The translation of the ECM structure in the case of subtracting the mean forces occurs because the pulling inward of the ECM node (shown in Figure 4d) creates a net force in the positive horizontal and vertical directions. Subtracting the mean force from each node does result in a net applied force of zero, but also results in an equilibrium configuration where the nodes have been shifted in the direction opposite the force imbalance. This situation is analogous to that of Section 4.1, where there was an artificial equilibrium state in the positive horizontal direction (and no relaxation to the equilibrium) resulting from a force imbalance in the negative horizontal direction. Such a shift may not be important for some applications if the *relative* position of the objects is desired. For our application, we are interested in the *absolute distance traveled by the cell*. For this reason, along with the ECM returning to its initial resting configuration for subsequent motility cycles, we conclude that it is better to use (16) to update the velocity rather than subtracting the mean forces.

We now use the model to simulate the distance traveled by the cell for different values of ECM stiffness k_{teth}. We calculate the total Euclidean distance traveled by the nucleus' center of mass as a function of time for the same 20-node ECM, but with varying stiffness $k_{\text{teth}} = 10, 25, 50 \,\text{pN}/\mu\text{m}^2$. For comparison, we also simulate a rigid ECM by enforcing a $\boldsymbol{u} = 0$ boundary condition at each of the ECM nodes rather than the mean velocity condition on the large circle. We simulate up to a finite time, which corresponds to the time the cell has finished one cycle of migration (Figure 4e) in the rigid ECM case ($t = 1.36 \approx 1360\Delta t$). The time step is adaptive; generally it is taken to be $\Delta t = 0.001$, but it shrinks to $\Delta t = 2 \times 10^{-4}$ for a small time (0.05 s) beginning when the cortex binds to a node and stiffens to $k_c = 100 \,\text{pN}/\mu\text{m}$. In Figure 6, we plot the Euclidean displacement in the direction of the ECM node over time for different values of stiffness k_{teth}. In all cases, the cell initially moves backwards slightly (as seen in Figure 4c) prior to contacting an ECM node. Once the ECM node is contacted and the cortex contracts, the distance traveled increases with time, with larger velocities for stiffer matrices. However, the data from all simulations appear to be approaching the same steady state value of displacement. For stiffer matrices, the node resists deformation by the cell (shown in Figure 4d), and the cell moves toward the node. In the rigid case, the node does not move, and the cell quickly contracts to form a circular configuration around the node. The conclusion of this preliminary study is therefore that *stiffer matrices* allow for faster cell velocities for finger-like protrusion mechanisms. We plan to study this problem in more detail by varying the matrix density and nuclear and cortical stiffness in future work.
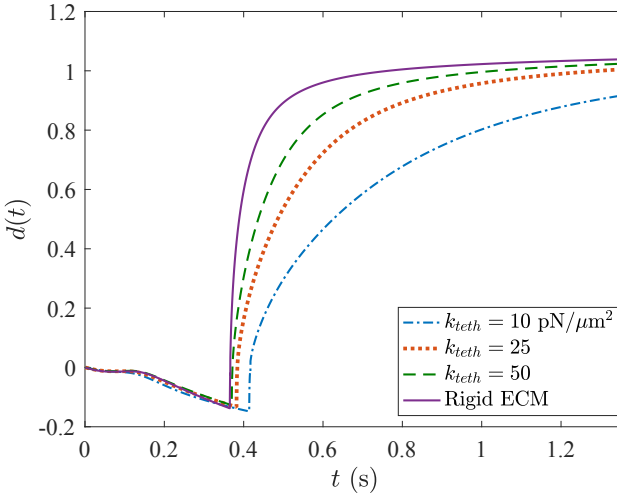
Figure 6. Euclidean distance traveled by the nucleus' center of mass versus time for different values of the ECM stiffness. Stiffer ECMs display faster velocities.

4.3. *Cellular blebbing.* Cellular blebs are spherical membrane protrusions that have been observed during cell migration [6]. A bleb forms when the cell cortex, normally attached to the cell membrane by linker proteins, detaches from the membrane. Cells that bleb are pressurized due to actomyosin contractility within the cortex. Once a bleb is initiated, a pressure driven flow drives the intracellular fluid (cytoplasm) that locally expands the membrane.

Bleb initiation has been modeled using different approaches, including solid mechanics [39], the immersed boundary method [31; 32], and boundary integral methods [14; 19; 20]. Results from several of these models have shown a bleb relieves only a small amount of intracellular pressure when the cytoplasm is modeled as a viscous fluid [31; 34]. Results from other models simulated with boundary integral methods show large pressure relief after bleb expansion [14; 20]. Our goal is to identify the source of this contradiction because maintaining high intracellular pressure is essential for cells to migrate using blebs [23].

Here we present a model of bleb expansion based on [20]. We treat both the membrane and cortex as one dimensional closed curves. The membrane and cortex parametrizations are represented by $X^m(s)$ and $X^c(s)$, respectively, where s is the arclength parameter. The most critical part of the model is the adhesion that connects the membrane and cortex. We model adhesion by an elastic spring connecting the membrane to the cortex with stiffness k_{adh}. The force density on the membrane due to adhesion is given by

$$\boldsymbol{F}_{\text{adh}}^{\text{mem/cor}}(s) = -k_{\text{adh}}(\boldsymbol{X}^m(s) - \boldsymbol{X}^c(s)), \qquad (22)$$
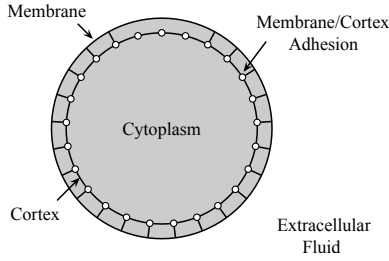
Figure 7. Components of the bleb model. The cytoplasm is modeled as a viscous fluid. A bleb is initiated by removing membrane-cortex adhesive links in a small region at the top of the cell.

with the force density on the cortex, $F_{\text{adh}}^{\text{cor/mem}}(s)$, equal and opposite. Elastic forces on the membrane and cortex are due to surface tension and stretching and are computed by (19) with

$$T = \gamma_m + k_m(\|X_s\| - 1) \tag{23}$$

with constants γ_m and k_m representing membrane surface tension and stiffness, respectively. The corresponding elastic parameters for the cortex are denoted by γ_c and k_c. The membrane satisfies a no-slip boundary condition, and its velocity is computed by (16). The velocity of the cortex is computed via a force balance, similar to [20],

$$\frac{dX^c}{dt} = \frac{1}{\nu_c}(F_{\text{el}}^{\text{cor}} + F_{\text{adh}}^{\text{cor/mem}}), \tag{24}$$

where ν_c is the cortical viscosity. A bleb is initiated by removing the adhesive links in a small region of length approximately $5\,\mu\text{m}$ at the top of the cell (see Figure 7).

This particular blebbing model is a physical example where the net force on the membrane fiber is nonzero. When the links between the top of the membrane and cortex are broken, there is a net vertical force on the membrane because part of the adhesive forces acting in the negative vertical direction are no longer present. Even though the cortex feels the equal and opposite forces, it moves independently of the fluid according to (24). The net hydrodynamic force is therefore equal to the net force on the membrane, and is nonzero. Without including the constant velocity from (15), the membrane would escape from the domain because of a spurious downward velocity resulting from the force imbalance in the positive vertical direction. The addition of u^R results in a well-posed problem so that we may solve for the membrane position in the blebbing model using the method of regularized Stokeslets.

Although previous studies have used the method of regularized Stokeslets to simulate cellular blebbing and migration in 2D [19; 20], the authors did not specify how they addressed the force imbalance in their models. We found our approach

symbol	quantity	value	source
r_{mem}	cell radius	$10\,\mu\text{m}$	[32; 34]
r_{cortex}	cortex radius	9.9 or $9.85\,\mu\text{m}$	[14]
γ_m	membrane surface tension	$40\,\text{pN}/\mu\text{m}$	[32]
k_m	membrane stiffness	$80\,\text{pN}/\mu\text{m}$	
γ_c	cortex surface tension	$250\,\text{pN}/\mu\text{m}$	[31]
k_c	cortical stiffness	$100\,\text{pN}/\mu\text{m}$	[31]
$k_{\text{adh}}^{\text{mem/cortex}}$	membrane/cortex adhesion stiffness coefficient	$247\,\text{pN}/\mu\text{m}^3$	
μ	cytosolic viscosity	$5\,\text{Pa\,s}$	[31]
ν_c	cortical viscosity	$10\,\text{pN\,s}/\mu\text{m}^3$	[31]

Table 1. Parameters for the blebbing model.

to give nearly identical results to a model where the net zero force constraint is enforced by subtracting the mean of the calculated forces at each Stokeslet point.

We simulate the cellular blebbing process with the parameters in Table 1. The two different values of the cortex radius are used to test our hypothesis that a force imbalance on the *cortex* is what drives the pressure relief seen by previous authors [14; 19; 20]. For $k_{\text{adh}} = 247\,\text{pN}/\mu\text{m}^3$, the forces on the cortex (in the absence of a bleb) are exactly in balance when $r_{\text{cortex}} = 9.9\,\mu\text{m}$. When $r_{\text{cortex}} = 9.85\,\mu\text{m}$, there is initially a force imbalance on the cortex independent of bleb initiation.

We first equilibrate the model for ten time steps, then initiate a bleb at $t = 0$ by breaking the adhesion at the 7 (out of $N = 100$) points with largest y coordinate. Figure 8 shows the membrane shape over time for the two different values of the cortex radius, where time units are reported after bleb initiation. The bleb sizes and shapes are exactly the same. Despite this, the pressure dynamics of the two models are quite different. As shown in Figure 9, the pressure drops significantly when $r_{\text{cortex}} = 9.85$ but remains constant in the case $r_{\text{cortex}} = 9.90$. This is because of the force imbalance on the cortex in the former case. When the cortex's initial position is inwards of its resting position, it expands outward dynamically. This decreases the force on the membrane (and on the fluid) due to membrane-cortex adhesion (22), leading to a global pressure decrease inside the cell. Importantly, we observe that at $t \geq 1\,\text{s}$, the cortex has reached its resting position and the two pressure profiles are the same (and are unchanged substantially with bleb expansion).

Models that include dynamic breaking of membrane-cortex adhesive links exhibit drastic changes in intracellular pressure [14; 20]. In such models, the dynamic breaking of adhesive links over time leads to pressure relief because the membrane force is updated suddenly without accounting for the corresponding force imbalance on the cortex. As the cortex slowly responds, it contracts inward in response to the loss of adhesive force, which promotes more link breakage and pressure changes.
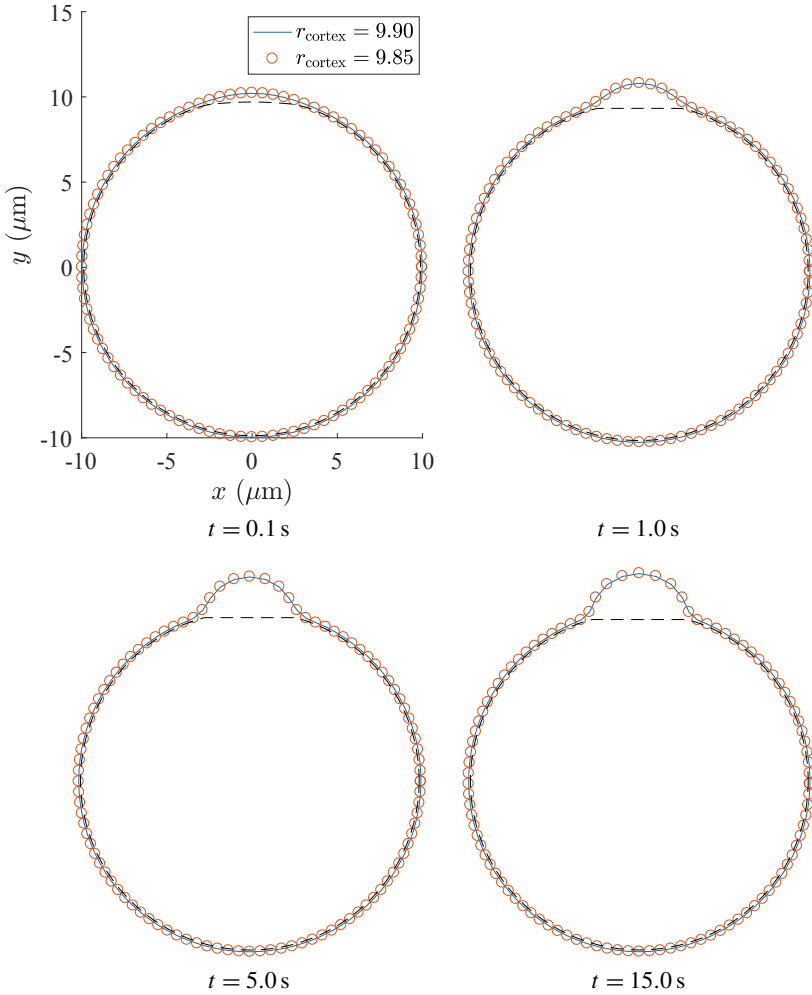
Figure 8. Membrane position for a blebbing cell with initial cortex radius 9.90 μm (blue line) or 9.85 μm (red circles). The position of the cortex is shown as a dashed black line and is in approximately the same position in both simulations. The positions are shown at several time values after bleb initiation.

The cortex itself is therefore never truly in equilibrium, and the force imbalance on it drives pressure changes.

The assumption that forces on the cortex are not equilibrated may be valid during highly dynamic processes such as during cell migration [41]. However, some experiments involve isolating a specific event, such as the expansion of a single bleb in [34]. In this work, experimental data show the cell achieves a quasisteady state behavior after bleb expansion, and the cortex is unlikely to be dynamically relieving pressure.
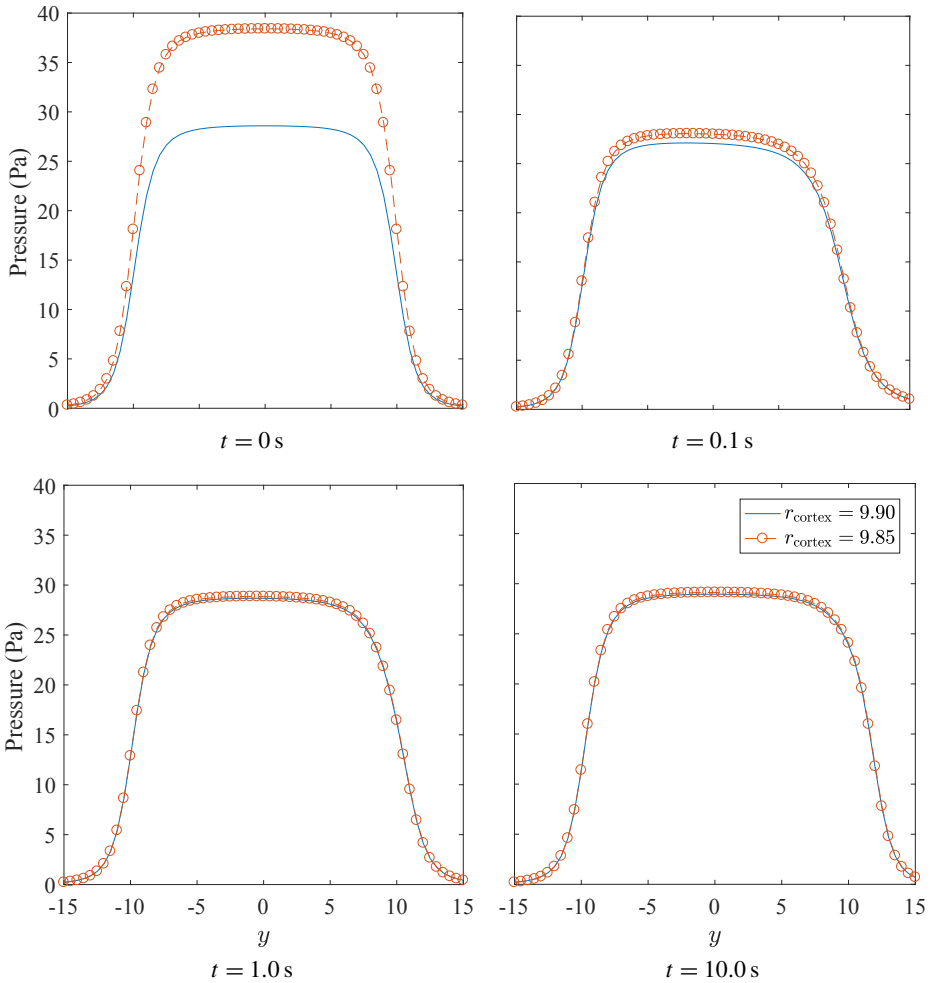
Figure 9. Pressure profile along the line $x = 0$ for the blebbing cell with initial cortex radius $9.90 \, \mu$m (blue line) or $9.85 \, \mu$m (red circles). Profiles are shown at (top left) $t = 0$ s, (top right) $t = 0.1$ s, (bottom left) $t = 1.0$ s, and (bottom right) $t = 10.0$ s. Note the large pressure relief when the forces are initially unbalanced on the cortex ($r_{\text{cortex}} = 9.85$).

5. Conclusion

When developing models for systems from biology, physics, and engineering that involve fluid-structure interaction, the simplification from 3D to 2D allows for model prototyping and fast simulations. In our applications, we seek to simulate cell motility and blebbing under a broad range of parameters, so fast simulations that are easy to visualize are critical for understanding model behavior. In zero Reynolds number flow, boundary integral methods are appealing because the velocity (and position) of immersed structures can be easily computed at the locations of interest

rather than by interpolation after solving for the velocity on an Eulerian grid as in the IB method [25]. The condition of net zero force for 2D boundary integral methods can be a limiting factor during the development and simulation of mathematical models, especially those that include elastic tether-like forcing.

Here we present a numerical method to treat force constraints in 2D Stokes flow in an infinite domain. When the regularized or standard Stokeslet is used with a net nonzero hydrodynamic force in the flow domain, the velocity is unbounded at infinity (Stokes' paradox). For problems where no specific boundary conditions are imposed from the physics of the model system, the standard free space Stokeslet solution without modification fails because it is a Green's function for a system of equations that is not well-posed. The treatment of this problem must therefore involve solving a new, well-posed system with the appropriate Green's function.

One option is to require the net force to be zero within the domain by subtracting the mean force, thereby making the free space problem well-posed. This approach maintains the system time and length scales, but we show here that it can lead to falsely translated equilibrium states and unphysical dynamics. Alternatively, shifting the boundary conditions to create a well-posed system and locally valid solution is appealing (previously treated via solving a linear system [8; 9; 38]). We show here that by using a confined geometry and enforcing the condition of a mean zero velocity on the boundary, we can easily derive a new Green's function for a well-posed system of equations that gives the physically correct behavior. Introducing the new boundary has the effect of introducing a new length scale and a corresponding change in the Reynolds number, and we use this fact to derive an upper bound on the size of the boundary. We combine this with a lower bound that comes from the variation of the velocity on the boundary to obtain a unique choice of R.

We test this method by applying it to several model systems. In Section 4.1 we use a simple example of tethered points to illustrate how the ill-posedness of the free space Stokes equations can lead to nonphysical behavior of the regularized Stokeslet solution. In both this example and the model of a cell migrating through an ECM in Section 4.2, we show that subtracting the mean force can create translations in the structure configurations, which for our applications are problematic because we seek measurements of the cell displacement. Because of this, we find our solution of solving the Stokes equations with a zero mean flow on the boundary to give the most physically relevant results for our applications. Finally, we apply this technique to show how force imbalances on the permeable cell cortex drive pressure relief (independent of bleb formation) in blebbing cells.

We emphasize that this technique is not a solution to generally address Stokes' paradox. The problem remains that no flow is truly 2D, and so representing 3D flows in two dimensions introduces modeling error. However, there are ways to

address models with nonzero net forcing so that insight can be gained from 2D models without having to take on the computational complexity of 3D. Here we describe a method to address Stokes' paradox and show that for our applications, the method gives solutions free of artificial translations. Additionally, the method is straightforward to implement and does not involve solving linear systems.

Future work involves extending the work of Section 4.2 to use our approach to investigate different mechanisms of cell migration (rear contraction in addition to frontal protrusion). We plan to examine the effectiveness of each mechanism for different values of ECM density, ECM stiffness, and cortical tension.

Acknowledgements

References

[1]   J. Ainley, S. Durkin, R. Embid, P. Boindala, and R. Cortez, *The method of images for regularized Stokeslets*, J. Comput. Phys. **227** (2008), no. 9, 4600–4616. MR Zbl

[2]   Y. Bao, J. Kaye, and C. S. Peskin, *A Gaussian-like immersed-boundary kernel with three continuous derivatives and improved translational invariance*, J. Comput. Phys. **316** (2016), 139–144. MR Zbl

[3]   G. K. Batchelor, *An introduction to fluid dynamics*, Cambridge University, 1999. MR Zbl

[4]   J. R. Blake, *A note on the image system for a Stokeslet in a no-slip boundary*, Math. Proc. Combridge **70** (1971), no. 2, 303–310. Zbl

[5]   E. L. Bouzarth, A. T. Layton, and Y.-N. Young, *Modeling a semi-flexible filament in cellular Stokes flow using regularized Stokeslets*, Int. J. Numer. Methods Biomed. Eng. **27** (2011), no. 12, 2021–2034. MR Zbl

[6]   G. Charras and E. Paluch, *Blebs lead the way: how to migrate without lamellipodia*, Nat. Rev. Mol. Cell Bio. **9** (2008), 730–736.

[7]   L. H. Cisneros, R. Cortez, C. Dombrowski, R. E. Goldstein, and J. O. Kessler, *Fluid dynamics of self-propelled microorganisms, from individuals to concentrated populations*, Exp. Fluids **43** (2007), no. 5, 737–753.

[8]   C. A. Copos and R. D. Guy, *A porous viscoelastic model for the cell cytoskeleton*, ANZIAM J. **59** (2018), no. 4, 472–498. MR Zbl

[9]   R. Cortez, *The method of regularized Stokeslets*, SIAM J. Sci. Comput. **23** (2001), no. 4, 1204–1225. MR Zbl

[10] R. Cortez, L. Fauci, and A. Medovikov, *The method of regularized Stokeslets in three dimensions: analysis, validation, and application to helical swimming*, Phys. Fluids **17** (2005), no. 3, 031504. MR Zbl

[11] R. Cortez and M. Nicholas, *Slender body theory for Stokes flows with regularized forces*, Commun. Appl. Math. Comput. Sci. **7** (2012), no. 1, 33–62. MR Zbl

[12] R. Cortez and D. Varela, *A general system of images for regularized Stokeslets and other elements near a plane wall*, J. Comput. Phys. **285** (2015), 41–54. MR Zbl

[13] S. Even-Ram and K. M. Yamada, *Cell migration in 3D matrix*, Curr. Opin. Cell Biol. **17** (2005), no. 5, 524–532.

[14] C. Fang, T. H. Hui, X. Wei, X. Shao, and Y. Lin, *A combined experimental and theoretical investigation on cellular blebbing*, Sci. Rep. UK **7** (2017), 16666.

[15] L. J. Fauci and R. Dillon, *Biofluidmechanics of reproduction*, Annu. Rev. Fluid Mech. **38** (2006), 371–394. MR

[16] P. Friedl, K. Wolf, and J. Lammerding, *Nuclear mechanics during cell migration*, Curr. Opin. Cell Biol. **23** (2011), no. 1, 55–64.

[17] E. Lauga and T. R. Powers, *The hydrodynamics of swimming microorganisms*, Rep. Progr. Phys. **72** (2009), no. 9, 096601, 36. MR

[18] W. R. Legant, J. S. Miller, B. L. Blakely, D. M. Cohen, G. M. Genin, and C. S. Chen, *Measurement of mechanical tractions exerted by cells in three-dimensional matrices*, Nat. Methods **7** (2010), 969–971.

[19] F. Y. Lim, K.-H. Chiam, and L. Mahadevan, *The size, shape, and dynamics of cellular blebs*, Europhys. Lett. **100** (2012), no. 2, 28004.

[20] F. Y. Lim, Y. L. Koon, and K.-H. Chiam, *A computational model of amoeboid cell migration*, Comput. Method. Biomec. **16** (2013), no. 10, 1085–1095.

[21] G. Morra, *Insights on the physics of Stokes flow*, Pythonic geodynamics: implementations for fast computing, Springer, 2018, pp. 93–104. MR

[22] E. Nazockdast, A. Rahimian, D. Zorin, and M. Shelley, *A fast platform for simulating semi-flexible fiber suspensions applied to cell mechanics*, J. Comput. Phys. **329** (2017), 173–209. MR Zbl

[23] E. K. Paluch and E. Raz, *The role and regulation of blebs in cell migration*, Curr. Opin. Cell Biol. **25** (2013), no. 5, 582–590.

[24] C. S. Peskin, *Flow patterns around heart valves: a numerical method*, J. Comput. Phys. **10** (1972), no. 2, 252–271. Zbl

[25] ———, *The immersed boundary method*, Acta Numer. **11** (2002), 479–517. MR Zbl

[26] C. Pozrikidis, *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University, 1992. MR Zbl

[27] ———, *Interfacial dynamics for Stokes flow*, J. Comput. Phys. **169** (2001), no. 2, 250–301. Zbl

[28] ———, *Numerical simulation of the flow-induced deformation of red blood cells*, Ann. Biomed. Eng. **31** (2003), no. 10, 1194–1205.

[29] ———, *Axisymmetric motion of a file of red blood cells through capillaries*, Phys. Fluids **17** (2005), no. 3, 031503. Zbl

[30] B. Rubinstein, K. Jacobson, and A. Mogilner, *Multiscale two-dimensional modeling of a motile simple-shaped cell*, Multiscale Model. Simul. **3** (2005), no. 2, 413–439. MR Zbl

[31] W. Strychalski and R. D. Guy, *A computational model of bleb formation*, Math. Med. Biol. **30** (2013), no. 2, 115–130. MR Zbl

[32] ———, *Intracellular pressure dynamics in blebbing cells*, Biophys. J. **110** (2016), no. 5, 1168–1179.

[33] J. M. Teran and C. S. Peskin, *Tether force constraints in Stokes flow by the immersed boundary method on a periodic domain*, SIAM J. Sci. Comput. **31** (2009), no. 5, 3404–3416. MR Zbl

[34] J.-Y. Tinevez, U. Schulze, G. Salbreux, J. Roensch, J.-F. Joanny, and E. Paluch, *Role of cortical tension in bleb growth*, P. Natl. Acad. Sci. USA **106** (2009), no. 44, 18581–18586.

[35] A.-K. Tornberg and M. J. Shelley, *Simulating the dynamics and interactions of flexible fibers in Stokes flows*, J. Comput. Phys. **196** (2004), no. 1, 8–40. MR Zbl

[36] M. Tozluoğlu, Y. Mao, P. A. Bates, and E. Sahai, *Cost–benefit analysis of the mechanisms that enable migrating cells to sustain motility upon changes in matrix environments*, J. Roy. Soc. Interface **12** (2015), no. 106, 20141355.

[37] M. Tozluoğlu, A. L. Tournier, R. P. Jenkins, S. Hooper, P. A. Bates, and E. Sahai, *Matrix geometry determines optimal cancer cell migration strategy and modulates response to interventions*, Nat. Cell Biol. **15** (2013), 751–762.

[38] B. Vanderlei, J. J. Feng, and L. Edelstein-Keshet, *A computational model of cell polarization and motility coupling mechanics and biochemistry*, Multiscale Model. Simul. **9** (2011), no. 4, 1420–1443. MR

[39] T. E. Woolley, E. A. Gaffney, J. M. Oliver, R. E. Baker, S. L. Waters, and A. Goriely, *Cellular blebs: pressure-driven, axisymmetric, membrane protrusions*, Biomech. Model. Mechan. **13** (2014), no. 2, 463–476.

[40] J. K. Wróbel, S. Lynch, A. Barrett, L. Fauci, and R. Cortez, *Enhanced flagellar swimming through a compliant viscoelastic network in Stokes flow*, J. Fluid Mech. **792** (2016), 775–797. MR Zbl

[41] A. K. Yip, K.-H. Chiam, and P. Matsudaira, *Traction stress analysis and modeling reveal that amoeboid migration in confined spaces is accompanied by expansive forces and requires the structural integrity of the membrane–cortex interactions*, Integr. Biol. **7** (2015), no. 10, 1196–1211.

[42] J. Zhu and A. Mogilner, *Comparison of cell migration mechanical strategies in three-dimensional matrices: a computational study*, Interface Focus **6** (2016), no. 5, 20160040.

ONDREJ MAXIAN: om759@cims.nyu.edu
Department of Mathematics, Courant Institute of Mathematical Sciences, New York University, New York, NY, United States

WANDA STRYCHALSKI: wis6@case.edu
Department of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University, Cleveland, OH, United States

Communications in Applied Mathematics and Computational Science

See inside back cover or msp.org/camcos for submission instructions.

Communications in Applied Mathematics and Computational Science