

A numerical method for osmotic water flow and solute diffusion with deformable membrane boundaries in two spatial dimension

Lingxing Yao^a, Yoichiro Mori^b

^a*Department of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University*

^b*School of Mathematics, University of Minnesota*

Abstract

Osmotic forces and solute diffusion are increasingly seen as playing a fundamental role in cell movement. Here, we present a numerical method that allows for studying the interplay between diffusive, osmotic and mechanical effects. An osmotically active solute obeys an advection-diffusion equation in a region demarcated by a deformable membrane. The interfacial membrane allows transmembrane water flow which is determined by osmotic and mechanical pressure differences across the membrane. The numerical method is based on an immersed boundary method for fluid-structure interaction and a Cartesian grid embedded boundary method for the solute. We demonstrate our numerical algorithm with the test case of an osmotic engine, a recently proposed mechanism for cell propulsion.

Keywords: Fluid structure interaction, Osmosis, Immersed boundary method, Cartesian grid embedded boundary method, Advection diffusion

1. Introduction

Differences in solute concentration across a semipermeable membrane or interface generates transmembrane osmotic water flow, which is of central importance in dialysis and desalination [1, 2], water absorption in epithelial systems [3, 4], and the swelling/deswelling of polyelectrolyte gels [5, 6]. The interaction of such flows with membrane and flow mechanics is a little explored area despite its potential significance in science and engineering [7, 8]. Here, we consider a model problem of such an interaction.

Our interest in this problem stems primarily from the problem of cell movement. Much recent evidence suggests that membrane ion channels and aquaporins (water channels), and thus, solute diffusion and osmosis, play an important role in cell movement [9, 10, 11]. To clarify the role of osmosis in cell movement, one needs to understand the interplay between solute diffusion, osmosis and mechanical forces. Our contribution in this paper is a step toward building a computational tool to study this interplay.

Email addresses: `lyx195@case.edu` (Lingxing Yao), `ymori@umn.edu` (Yoichiro Mori)

In this paper, we present a numerical method for the following model problem in two spatial dimension. A cell (or a number of cells) is immersed in a surrounding fluid separated by the cell membrane. The cell membrane is elastic and the fluid flow satisfies the Stokes equation. Un-charged solutes diffuse and are advected with the fluid flow, and may pass through the membrane through active and passive mechanisms. To allow for transmembrane water flow, the membrane has a slip velocity with respect to the underlying fluid flow. This slip velocity is determined by the jump in the mechanical as well as the osmotic pressure across the membrane. The problem we consider in this paper is close to the problems considered in [12, 13, 14, 15, 16]. In [12, 15], the author considers the problem in which the membrane is permeable to water but not to solute (semipermeable membrane). In papers [13, 14] the authors extend the numerical method in [12] to the case when the membrane may be permeable to both solute and water. In [16], the authors incorporate osmotic forces as a part of a fluid-solute-structure interaction term. We also mention the recent work in [17], where the authors consider the well-posedness of a related problem.

We now briefly discuss our numerical method. At each time step, we alternate between solving the fluid-structure interaction problem and the solute diffusion problem. For the fluid-structure interaction problem, we use the immersed boundary (IB) method [18]. Once the fluid velocity is found, the position of the membrane (immersed boundary points) must be updated. As discussed above, there is a slip between the underlying fluid velocity and the velocity with which the membrane moves. Problems in which an interface or structure has a slip velocity have been simulated using the IB method in numerous papers including [19, 20, 21, 22]. The update of the immersed boundary points are treated in an explicit fashion in these papers. In this paper, we employ a partially implicit treatment of this update, which confers better stability properties to our method.

For the solute diffusion problem, we use a Cartesian grid embedded boundary method [23, 24], which allows us to capture the sharp interfacial discontinuity of the solution concentration (and its gradient) across the membrane interface. This is similar in spirit [12, 13, 14, 15] where the immersed interface method [25] is used to deal with the discontinuity. The major difference and novelty of our method with respect to previous work is that we treat the solute interface conditions implicitly. The (passive) transmembrane solute flux is proportional to the solute difference and is thus a “diffusive” term. An implicit discretization of the solute boundary condition is crucial for stability especially when the solute permeability coefficient is large; indeed, we have found that, without this implicit treatment, stable computations require prohibitively small time steps. For the solute update step, we thus solve a linear system whose unknowns include the concentration at Cartesian grid points in the bulk and on both faces of the membrane. This is in contrast to previous work in which the unknown values are located only in the bulk grid points. Our method in this sense is analogous to the augmented methods in the immersed interface literature [25, 26, 27]. This results in a linear system whose structure is more complicated than in [12, 13, 14, 15] where the boundary conditions are treated explicitly. This linear system is solved using preconditioned GMRES.

When the boundary moves, some grid points that were on the intracellular (or extracellular) side of the membrane will be on the other side of the membrane at the next time step. To solve the (advection) diffusion equation using the method of lines, we must modify the time derivative at these freshly cleared locations; simply using the grid value

at the previous time step will not work since the solute concentration is discontinuous across the interface. We employ a new discretization method for this modification which avoids a complicated extrapolation procedure.

We test our numerical method using a model problem inspired by the recent experimental results in [28]. In [28], the authors present experimental results in which a cancer cell confined to a one-dimensional geometry can move even when its cytoskeletal and motor machinery is disrupted by pharmacological means. The authors suggest that this cellular movement is due to osmotic pressure differences, which the cell generates by pumping solute in at one end of the cell and pumping solute out of the other end. We also point out that this osmotic engine mechanism can be thought of as a version of osmophoresis [29, 30], in which a vesicle with a semi-permeable membrane moves when subjected to an externally imposed concentration gradient. The osmotic engine mechanism suggested in [28], then, can be seen as osmophoresis with a concentration gradient generated by the cell itself. In this paper, we shall test whether such an osmotic engine mechanism can propel the cell forward in a two-dimensional setting using biophysically realistic parameter values, and study the interplay between shape changes, solute diffusivity and osmotic forces. Our simulations show that the osmotic engine mechanism is a potentially feasible mechanism of cell propulsion in the 2D setting. We also mention related work in [31, 32, 33], where the authors consider a colloidal particle being propelled by an osmotic mechanism. There, in contrast to our model simulations, a concentration gradient generated by a chemical reaction provides the driving force for osmotic fluid flow. A variant of this mechanism is also considered in [34, 35].

The outline of our paper is as follows. In Section 2, we define our model problem. In Section 3, we give a detailed description of the numerical algorithm. In Section 4, we present simulations using our numerical method. We first present an example in Section 4.1, in which the solute diffusion component of our method is tested. In Section 4.2, we test our numerical method using the osmotic engine model discussed above, which couples solute diffusion with membrane mechanics and fluid flow. In all of these we perform convergence studies to confirm the expected convergence rate. Finally, in Section 4.3, we perform further parametric studies on the osmotic engine model.

2. Setup

Consider a rectangular domain $\Omega \subset \mathbb{R}^2$ and one or multiple smooth closed surfaces denoted by $\Gamma \subset \Omega$. This closed surface divides Ω into two domains. Let $\Omega_i \subset \Omega$ be the region bounded by Γ , and let $\Omega_e = \Omega \setminus (\Omega_i \cup \Gamma)$. The region Ω_i the intracellular region and Ω_e is the extracellular region. The equations to follow are written in suitable dimensionless form.

We begin by writing down the equations of chemical concentration c . At any point in Ω_i or Ω_e

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) = \nabla \cdot (D\nabla c) \quad (1)$$

where D is the diffusion coefficient and \mathbf{u} is the fluid velocity field.

Let us now consider the interfacial boundary conditions on the membrane Γ . Since we want to account for osmotic water flow, the membrane Γ will deform in time. Sometimes, we shall use the notation Γ_t to make this time dependence explicit. Let Γ_{ref} be the resting

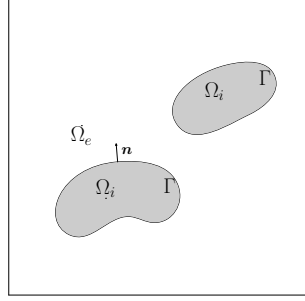


Figure 1: Diagram of the computational domain and flexible membrane.

or reference configuration of Γ . The membrane will then be a smooth deformation of this reference surface. We may take a coordinate system s on Γ_{ref} , which would serve as a material coordinate for Γ_t . The trajectory of a point that corresponds to $s = s_0$ is given by $\mathbf{X}(s_0, t) \in \Omega$. For fixed t , $\mathbf{X}(\cdot, t)$ gives us the shape of the membrane Γ_t .

Consider a point $\mathbf{x} = \mathbf{X}(s, t)$ on the membrane. Let \mathbf{n} be the outward unit normal on Γ at this point. The boundary conditions satisfied on the intracellular and extracellular faces of the membrane are given by:

$$(\mathbf{u}c - D\nabla c) \cdot \mathbf{n} = c \frac{\partial \mathbf{X}}{\partial t} \cdot \mathbf{n} + j_c + j_p \text{ on } \Gamma_i \text{ or } \Gamma_e. \quad (2)$$

The expression “on $\Gamma_{i,e}$ ” indicates that the quantities are to be evaluated on the intracellular and extracellular faces of Γ respectively. Equation (2) is just a statement of conservation of ions at the moving membrane. The sum $j_c + j_p$ is the transmembrane chemical flux. Flux going from Ω_i to Ω_e is taken to be positive. The flux is divided into the passive channel flux j_c and the active pump flux j_p . For j_c , we let:

$$j_c = \hat{k}_c [c], \quad \hat{k}_c = k_c \left| \frac{\partial \mathbf{X}}{\partial s} \right|^{-1}, \quad (3)$$

where k_c is a positive constant. The Jacobian factor accounts reflects the view that the chemicals pass through the membrane through channels, whose density transforms inversely with extension of the membrane. The solute thus flows from where the concentration is high to low. For the active flux j_p we set:

$$j_p = \hat{k}_p H(s, c_i, c_e), \quad \hat{k}_p = k_p(s) \left| \frac{\partial \mathbf{X}}{\partial s} \right|^{-1}, \quad (4)$$

$$H(s, c_i, c_e) = \begin{cases} c_i & \text{if } k_p(s) \geq 0, \\ c_e & \text{if } k_p(s) < 0, \end{cases}$$

where $c_{i,e}$ are the values of the concentration at $\Gamma_{i,e}$ respectively. When $k_p \geq 0$, the membrane pump actively pumps solute out of the cell and the flux is dependent on c_i , and vice versa when $k_p < 0$. The Jacobian factor is present for the same reason as for (3).

We now discuss force balance. Consider the equations of fluid flow. The flow field \mathbf{u} satisfies the Stokes equation in $\Omega_{i,e}$:

$$0 = \nabla \cdot \Sigma_m(\mathbf{u}, p), \quad \nabla \cdot \mathbf{u} = 0, \quad \Sigma_m(\mathbf{u}, p) = \nu(\nabla \mathbf{u} + (\nabla \mathbf{u})^T) - pI = 2\nu \nabla_S \mathbf{u} - pI \quad (5)$$

where $\nu > 0$ is the dynamic viscosity, I is the 2×2 identity matrix and $(\nabla \mathbf{u})^T$ is the transpose of $\nabla \mathbf{u}$, and p is the pressure. We have introduced the notation ∇_S to denote the symmetric part of the velocity gradient. Σ_m is just the Stokes stress tensor.

We now turn to boundary conditions at the cell membrane Γ . Take a point $\mathbf{x} = \mathbf{X}(s, t)$ on the boundary Γ , and let \mathbf{n} be the unit outward normal on Γ at this point. First, by force balance, we have:

$$[\Sigma_m(\mathbf{u}, p)\mathbf{n}] = \hat{\mathbf{F}}_{\text{mem}}. \quad (6)$$

Here, \mathbf{F}_{mem} is the elastic force per unit area of membrane and $[\cdot]$ denotes the jump in the enclosed quantity across the membrane (evaluation on Γ_i minus evaluation on Γ_e). For $\hat{\mathbf{F}}_{\text{mem}}$, we take the constitutive law:

$$\begin{aligned} \hat{\mathbf{F}}_{\text{mem}} &= \mathbf{F}_{\text{mem}} \left| \frac{\partial \mathbf{X}}{\partial s} \right|^{-1}, \quad \mathbf{F}_{\text{mem}} = \mathbf{F}_{\text{elas}} + \mathbf{F}_{\text{bend}}, \\ \mathbf{F}_{\text{elas}} &= k_{\text{elas}} \frac{\partial}{\partial s} \left(\left(\left| \frac{\partial \mathbf{X}}{\partial s} \right| - \ell \right) \boldsymbol{\tau} \right), \quad \boldsymbol{\tau} = \frac{\partial \mathbf{X}}{\partial s} \left| \frac{\partial \mathbf{X}}{\partial s} \right|^{-1}, \\ \mathbf{F}_{\text{bend}} &= -k_{\text{bend}} \frac{\partial^4 \mathbf{X}}{\partial s^4}, \end{aligned} \quad (7)$$

where $k_{\text{elas}} > 0$ is the elasticity constant, ℓ is the resting length and $k_{\text{bend}} > 0$ is the bending stiffness.

In addition to the force balance condition (6), we need a continuity condition on the interface Γ . The velocity field is assumed continuous across Γ :

$$[\mathbf{u}] = 0. \quad (8)$$

Since we are allowing for osmotic water flow, we have a slip between the movement of the membrane and the flow field. At a point $\mathbf{x} = \mathbf{X}(\boldsymbol{\theta}, t)$ on the boundary Γ we have:

$$\mathbf{u} - \frac{\partial \mathbf{X}}{\partial t} = j_w \mathbf{n} \quad (9)$$

where j_w is water flux through the membrane. We are thus assuming that water flow is always normal to the membrane and that there is no slip between the fluid and the membrane in the direction tangent to the membrane. Given that \mathbf{n} is the outward normal, j_w is positive when water is flowing out of the cell. We let:

$$j_w = k_w[\psi], \quad \psi = -RTc - \mathbf{n} \cdot ((\Sigma_m(\mathbf{u}, p))\mathbf{n}), \quad (10)$$

where R is gas constant and T is absolute temperature. Transmembrane water flow is thus driven by the difference in mechanical force ($[\mathbf{n} \cdot ((\Sigma_m(\mathbf{u}, p))\mathbf{n})]$) as well as the difference in osmotic pressure ($[c]$). Given (6), we may also set :

$$j_w = -k_w \left(RT[c] + \hat{\mathbf{F}}_{\text{mem}} \cdot \mathbf{n} \right). \quad (11)$$

On the outer boundary of the rectangular region Ω , we set periodic boundary conditions in the horizontal x direction for both the concentration c and the velocity field \mathbf{u} . In the vertical y direction, we set $\mathbf{u} = 0$ and no-flux boundary conditions for the concentration c .

We point out that the above equations are thermodynamically consistent in the sense that it satisfies the following free energy identity [36]. Now, suppose c, \mathbf{u}, p and \mathbf{X} are smooth functions that satisfy the equations and boundary conditions just described. Then, the following free energy identity holds:

$$\begin{aligned} \frac{d}{dt}(E_{\text{bulk}} + E_{\text{mem}}) &= -I - J, \\ E_{\text{bulk}} &= \int_{\Omega_i \cup \Omega_e} \omega d\mathbf{x}, \quad \omega = RT(c \ln c - c), \\ E_{\text{mem}} &= \int_{\Gamma_{\text{ref}}} \left(k_{\text{elas}} \left(\left| \frac{\partial \mathbf{X}}{\partial s} \right| - \ell \right)^2 + k_{\text{bend}} \left| \frac{\partial^2 \mathbf{X}}{\partial s^2} \right|^2 \right) ds, \\ I &= \int_{\Omega_i \cup \Omega_e} \left(2\nu |\nabla_S \mathbf{u}|^2 + \frac{D}{RT} c |\nabla \mu|^2 \right) d\mathbf{x}, \quad \mu = RT \ln c, \\ J &= \int_{\Gamma} ([\psi] j_w + [\mu](j_c + j_p)) dm_{\Gamma}, \quad dm_{\Gamma} = \left| \frac{\partial \mathbf{X}}{\partial s} \right| ds. \end{aligned} \tag{12}$$

Here, $|\nabla_S \mathbf{u}|$ is the Frobenius norm of the 2×2 symmetric rate of deformation matrix $\nabla_S \mathbf{u}$. The derivation of this identity follows from a standard integration by parts argument [36], and is given in the Appendix A for convenience of the reader. Note that $I \geq 0$ and J is non-negative if j_p is 0. Thus, in the absence of an active pump flux, the free energy is monotone decreasing, in concordance with the second law of thermodynamics. The above energy relation, in particular, shows that the only external free energy input to the system is through active ionic pumps. An interesting aspect of the osmotic engine mechanism we examine in this paper is that this purely chemical free energy input is turned into directional movement.

It will be convenient to rewrite the fluid equations in the following form. Equations (5), (6) and (8) can be written together as:

$$0 = \nu \Delta \mathbf{u} - \nabla p + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0, \tag{13}$$

$$\mathbf{f}(\mathbf{x}, t) = \int_{\Gamma_{\text{ref}}} \mathbf{F}_{\text{mem}}(\mathbf{X}) \delta(\mathbf{x} - \mathbf{X}(s, t)) ds, \tag{14}$$

where δ is the two-dimensional Dirac delta function, and the above equations are to be understood in the sense of distributions. The immersed boundary (IB) method, which we will use for the discretization of the fluid equations, makes use of the above reformulation.

3. Numerical algorithm

We consider a square computational domain $\Omega = [0, L] \times [0, L]$ and lay a fixed Cartesian grid with vertices at (x_i, y_j) , $x_i = i\Delta x$, $i = 0, \dots, N$, $y_j = j\Delta y$, $j = 0, \dots, N$, and cell centers $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$. In each computational cell, chemical concentration c and fluid pressure p are defined at the cell center, and the velocity components $(u, v) = \mathbf{u}$ are

arranged at vertical and horizontal edges of the cells respectively, following the MAC (marker and cell) grid arrangement.

The position $\mathbf{X}(s, t)$ of the internal boundary Γ , which we shall also call the immersed boundary, is discretized using a uniform discretization on the reference grid Γ_{ref} . We thus have immersed boundary (IB) points $\mathbf{X}(s_i, t)$, $i = 1, \dots, N_{\text{ring}}$ that represent the immersed boundary Γ . In order to compute geometric and grid quantities, at each time step we construct a cubic spline parametric representation of the immersed boundary using the IB point locations.

The internal interface Γ divides the computational domain Ω into intra- and extra-cellular regions Ω_i and Ω_e . Each cell center point is thus either on the extracellular side, or intracellular side. According to their relative positions to the immersed boundary, the cell centers can be grouped into: 1) *regular cell centers* on either side, which do not have any neighboring cell center on the other side of Γ ; and 2) *irregular cell centers* which have at least one neighboring cell center that is located on the other side of the membrane. The chemical concentration equation (1) is discretized differently depending on whether the cell center is regular or irregular.

We introduce auxiliary concentration variables along the membrane as follows. At each time step, we compute *grid crossings*, the points at which the immersed boundary Γ and grid lines connecting cell centers intersect. At each of these grid crossings, two auxiliary variables, c_i^b and c_e^b are defined, which correspond to the value of the concentration evaluated at the Ω_i face and the Ω_e face of the membrane respectively. Note that the concentration c will in general have a jump across the membrane and the values c_i^b and c_e^b will be different. The variables $c_{i,e}^b$ are used to discretize the boundary condition (2) and to discretize the concentration equation (1) at irregular cell centers. An illustration of the grid crossing and its relation to Γ are shown in Figure 2.

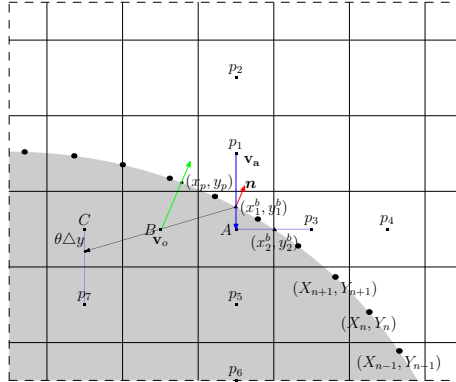


Figure 2: Grid crossing and the immersed boundary . The grid lines connecting cell center A with cell centers p_1 and p_3 intersect Γ at grid crossings (x_1^b, y_1^b) and (x_2^b, y_2^b) respectively. At each of these points, the auxiliary concentration variables $c_{i,e}^b$ are defined. The points (X_k, Y_k) are IB points along Γ .

The outline of the overall algorithm is as follows. For each time step, we alternate between the fluid-structure interaction substep and the concentration substep. Suppose we are given $u^n, v^n, p^n, \mathbf{X}^n, c^n, c_{i,e}^{b,n}$ where the superscript n denotes the value of the grid variables at time $t = n\Delta t$, $n = 0, 1, \dots$ and Δt is the time step.

Substep 1 Given u^n, v^n, p^n and \mathbf{X}^n , use the IB method to compute u^{n+1}, v^{n+1} and p^{n+1} with a discretization of (13), (14) and (7). Computations of (14) and (7) is performed using \mathbf{X}^n . Once this is found, use $\mathbf{X}^n, c_{i,e}^{b,n}$ as well as the newly found u^{n+1}, v^{n+1} to discretize (9) and (11) to obtain the new IB locations \mathbf{X}^{n+1} . Equation (11) requires concentration values at IB locations, which we denote by $c_{i,e}^{IB,n}$. This is obtained by interpolating the concentration values $c_{i,e}^{b,n}$ defined at grid crossings.

Substep 2 Given our new IB point locations \mathbf{X}^{n+1} and the fluid velocity (u^{n+1}, v^{n+1}) , solve moving boundary advection diffusion problem for the concentration c^{n+1} and $c_{i,e}^{b,n+1}$. At irregular cell centers discretization of equations (1) requires care. This is especially the case for *freshly cleared points*, which are computational cell centers that were located in Ω_i in the previous time step but are now in Ω_e or vice versa. Boundary conditions (2) are enforced at grid crossing with the help of the auxiliary concentration variables $c_{i,e}^b$. A linear system for c^{n+1} and $c_{i,e}^{b,n+1}$ is obtained and solved using an iterative method.

We now discuss each computational substep in detail.

3.1. Fluid-structure interaction substep

3.1.1. IB method for fluid velocity

Consider the Stokes equation (5) for fluid velocity \mathbf{u} and pressure p :

$$0 = \nu \Delta \mathbf{u} - \nabla p + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0. \quad (15)$$

We use immersed boundary method to discretize it. We arrange all variables in a MAC grid, in which the pressure p is defined at cell centers and the velocity field \mathbf{u} is defined at the horizontal (u) and vertical (v) cell faces. Define the following differencing operators for any grid function w , where $w_{\alpha,\beta}^n$ denotes the value of w at $(x, y) = (\alpha h, \beta h)$ at time $t = n\Delta t$:

$$\begin{aligned} \mathcal{D}_x^\pm w_{\alpha,\beta} &= \pm \frac{w_{\alpha\pm 1,\beta} - w_{\alpha,\beta}}{h}, \quad \mathcal{D}_y^\pm w_{\alpha,\beta} = \pm \frac{w_{\alpha,\beta\pm 1} - w_{\alpha,\beta}}{h}, \\ \mathcal{L}w_{\alpha,\beta} &= \mathcal{D}_x^+ \mathcal{D}_x^- w_{\alpha,\beta} + \mathcal{D}_y^+ \mathcal{D}_y^- w_{\alpha,\beta} \\ &= \frac{w_{\alpha+1,\beta} + w_{\alpha,\beta+1} + w_{\alpha-1,\beta} + w_{\alpha,\beta-1} - 4w_{\alpha,\beta}}{h^2}, \end{aligned} \quad (16)$$

Let $\mathbf{u} = (u, v)$, and $\mathbf{f} = (f, g)$, equation (15) is then

$$\begin{aligned} \mathcal{D}_x^- p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} &= \nu \mathcal{L}u_{i,j+\frac{1}{2}}^{n+1} + f_{i,j+\frac{1}{2}}^n, \\ \mathcal{D}_y^- p_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} &= \nu \mathcal{L}v_{i+\frac{1}{2},j}^{n+1} + g_{i+\frac{1}{2},j}^n, \\ 0 &= \mathcal{D}_x^- u_{i+1,j+\frac{1}{2}}^{n+1} + \mathcal{D}_y^- v_{i+\frac{1}{2},j+1}^{n+1}. \end{aligned} \quad (17)$$

As we assume periodic boundary condition for \mathbf{u} and p on the left and right edge of the computational domain, and homogeneous Dirichlet boundary condition for \mathbf{u} on the top and bottom edges, we can solve (u, v) and p from linear system (17) by using FFT along

x direction, which will result in block diagonal linear system to solve at each x_i , and can be solved efficiently using a direct solver.

We turn to the determination of the body forces $\mathbf{f} = (f, g)$. Let $\mathbf{F}_{\text{mem}} = (F_x, F_y)$ and the IB point positions $\mathbf{X} = (X, Y)$. For a quantity W defined on the immersed boundary grid parametrized by s , we let W_k denote the value of W at point $s = s_k = k\Delta s$, where Δs is the grid spacing in s . Equation (14) for $\mathbf{f} = (f, g)$ are discretized as follows

$$\begin{aligned} f_{i,j+\frac{1}{2}}^n &= \sum_{k=1}^{N_{\text{ring}}} F_{x,k}^n \delta_h(x_i - X_k^n) \delta_h(y_{j+\frac{1}{2}} - Y_k^n) \Delta s \\ g_{i+\frac{1}{2},j}^n &= \sum_{k=1}^{N_{\text{ring}}} F_{y,k}^n \delta_h(x_{i+\frac{1}{2}} - X_k^n) \delta_h(y_j - Y_k^n) \Delta s \end{aligned} \quad (18)$$

where $\delta_h(r)$ is a regularized discrete delta function. In this paper, we use the following discrete delta function:

$$\begin{aligned} \delta_h(r) &= \frac{1}{h} \phi\left(\frac{r}{h}\right), \\ \phi(r) &= \begin{cases} \frac{1}{8}(3 - 2|r| + \sqrt{1 + 4|r| - 4r^2}) & |r| \leq 1, \\ \frac{1}{8}(5 - 2|r| - \sqrt{-7 + 12|r| - 4r^2}) & 1 < |r| \leq 2, \\ 0 & 2 < |r|. \end{cases} \end{aligned} \quad (19)$$

The rationale for this particular choice of regularization is discussed in [18]. To compute the membrane force \mathbf{F}_{mem} let us first introduce the following differencing operators acting on functions W defined on the IB grid:

$$\mathcal{D}_s^\pm W_k = \pm \frac{W_{k\pm 1} - W_k}{\Delta s}, \quad \mathcal{L}_s W_k = \mathcal{D}_s^+ \mathcal{D}_s^- W_k. \quad (20)$$

Using these operators, we discretize (14) as follows:

$$\mathbf{F}_{\text{mem},k}^n = k_{\text{elas}} \mathcal{D}_s^+ \left(\left(1 - \ell |\mathcal{D}_s^- \mathbf{X}_k^n|^{-1} \right) \mathcal{D}_s^- \mathbf{X}_k^n \right) - k_{\text{bend}} \mathcal{L}_s \mathcal{L}_s \mathbf{X}_k^n, \quad (21)$$

where the differencing operators above act component-wise.

Thus, given the IB point locations \mathbf{X}^n , the membrane forces \mathbf{F}_{mem} is computed using (21), which is then used to compute \mathbf{f}^n with the regularized discrete delta functions as in (18). This body force \mathbf{f}^n is then fed into equation (17), which is solved to produce \mathbf{u}^{n+1} and p^{n+1} .

3.1.2. Semi-implicit update of IB locations

We turn to the update of the IB point locations. Combining (9) and (11), we employ the following discretization (for any grid function w , we have $\mathcal{D}_t^- w^n = \frac{w^n - w^{n-1}}{\Delta t}$):

$$\begin{aligned}\mathcal{D}_t^- \mathbf{X}_k^{n+1} &= \mathbf{U}_k^{n+1} - j_{\mathbf{w}}^n, \quad j_{\mathbf{w}}^n = -k_{\mathbf{w}} \left([c_k^{\text{IB},n}] + \hat{\mathbf{F}}_{\text{mem},k}^{n+1} \cdot \mathbf{n}^n \right), \quad \mathbf{U}_k = (U_k, V_k), \\ \hat{\mathbf{F}}_{\text{mem},k}^{n+1} &= \mathbf{F}_{\text{mem},k}^{n+1} \frac{1}{2} \left(|\mathcal{D}_s^+ \mathbf{X}^n|^{-1} + |\mathcal{D}_s^- \mathbf{X}^n|^{-1} \right), \\ U_k^n &= \sum_{i,j} u_{i,j+\frac{1}{2}}^n \delta_h(x_i - X_k^n) \delta_h(y_{j+\frac{1}{2}} - Y_k^n) h^2, \\ V_k^n &= \sum_{i,j} v_{i+\frac{1}{2},j}^n \delta_h(x_{i+\frac{1}{2}} - X_k^n) \delta_h(y_j - Y_k^n) h^2,\end{aligned}\tag{22}$$

where $\mathbf{F}_{\text{mem},k}^{n+1}$ is specified as in (21). The values $c_{i,e,k}^{\text{IB},n}$ are the intracellular and extra-cellular concentrations at the IB point k at time $n\Delta t$, and is evaluated by interpolating the membrane concentration values at grid crossings $c_{i,e}^{\text{b},n}$. The jump $[c_k^{\text{IB},n}]$ is equal to $c_{i,k}^{\text{IB},n} - c_{e,k}^{\text{IB},n}$. In the right hand side of the first equation, all terms except for $\mathbf{F}_{\text{mem},k}$ are known quantities. This is thus a nonlinear equation for \mathbf{X}^{n+1} , which is solved using a Newton iteration. We have found that the implicit treatment of $\mathbf{F}_{\text{mem},k}$ lead to better stability properties.

3.2. Concentration Substep

We turn to the update of the chemicals. The strategy used here in updating chemical variables is similar to many Cartesian grid embedded boundary methods (see like [24, 23, 37] and many others), which use interpolations to set up locally smooth functions for constructing stencils on the Cartesian grids. Our construction of stencils at regular and irregular cell centers and the inclusion of auxiliary variables at grid crossings is inspired by the work in [24]. The most important feature of the concentration substep is that the concentration boundary conditions (2) are treated implicitly, so that the unknown concentrations are located at Cartesian grid points and the grid crossings. As discussed earlier, this has proved crucial for stable computations. Another new feature of our discretization strategy is the treatment of the time derivative in freshly cleared computational cell centers, i.e., the cell centers that change sides between time steps.

3.2.1. Regular cell centers

Recall that cell centers can be classified into regular cell centers and irregular cell centers. The regular cell centers have no neighboring grid cell centers on the other side of the membrane Γ_{n+1} , the immersed boundary defined by the IB point locations \mathbf{X}^{n+1} .

At any regular Cartesian cell center, we use a standard implicit Euler discretization of the (1):

$$\begin{aligned}\mathcal{D}_t^- c_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} &+ \mathcal{D}_x^- \left(u_{i+1,j+\frac{1}{2}}^{n+1} \mathcal{A}_x^+ c_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} \right) + \mathcal{D}_y^- \left(v_{i+\frac{1}{2},j+1}^{n+1} \mathcal{A}_y^+ c_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1} \right) \\ &= D\mathcal{L}c_{i+\frac{1}{2},j+\frac{1}{2}}^{n+1},\end{aligned}\tag{23}$$

where, for any quantity w on the Cartesian grid,

$$\mathcal{A}_x^+ w_{\alpha,\beta} = \frac{1}{2} (w_{\alpha+1,\beta} + w_{\alpha,\beta}), \quad \mathcal{A}_y^+ w_{\alpha,\beta} = \frac{1}{2} (w_{\alpha,\beta+1} + w_{\alpha,\beta}), \quad (24)$$

where α, β are integer or half integer. Note here that the velocity field (u^{n+1}, v^{n+1}) have been determined at the immersed boundary substep.

At irregular cell centers, the above spatial discretization cannot be performed as is, since some of the concentration variables at neighboring cell centers represent concentrations at the other side of the membrane. It may also be the case that the cell center of interest was *freshly cleared*. That is to say, the cell center may have been on the Ω_i side of Γ_n (the immersed boundary position at the n -th time step) whereas it is on the Ω_e side of Γ_{n+1} , or vice versa. In this case, the discretization of the time derivative must also be modified. We discuss these two modifications in turn.

3.2.2. Stencil at irregular cell centers

For chemicals at irregular cell centers, we use (23) but with the following modifications. Suppose we try to update the chemical c_A^{n+1} defined at cell center A (contained in Ω_i) in Figure 2 using (23). The difference operators require chemical concentration variables at p_1 and p_3 . We obtain an expression for the concentration at these two points (called *ghost cells*) using an extrapolation procedure using concentration variables at the cell centers and grid crossings in Ω_i (where point A is located).

The extrapolation scheme for ghost cell chemicals is adopted from [24]. At point p_1 , in Figure 2, we use

$$c_{p_1} = \frac{2(1-\theta)}{2+\theta} c_{p_6} - \frac{3(1-\theta)}{1+\theta} c_{p_5} + \frac{6}{(1+\theta)(2+\theta)} c_i^b, \quad (25)$$

where θ is the ratio of distance from the grid crossing (x_1^b, y_1^b) to A and distance from p_1 to A (grid spacing in the y direction), c_{p_6} and c_{p_5} are chemicals at p_6 and p_5 respectively, and c_i^b is the auxiliary intracellular chemical concentration defined at grid crossing (x_1^b, y_1^b) . A similar procedure is performed in the x direction to obtain an extrapolation formula at point p_3 .

Equation (25) uses two grid point locations p_5 and p_6 , and in exceptional cases depending on the geometry of Γ relative to the Cartesian grid, two such grid points may not be available. When only one such point is available, we use the formula:

$$c_{p_1} = \frac{-(1-\theta)}{1+\theta} c_{p_5} + \frac{2}{(1+\theta)} c_i^b. \quad (26)$$

In the extreme case when no such grid locations are available, we set $c_{p_1} = c_i^b$. The use of these lower order extrapolation procedures (as opposed to (25)) will in general lead to order 1 consistency errors at these grid points. However, the points at which such errors are committed remains a small fraction of cell centers (the proportion should become smaller with finer grid spacing), and thus does not affect the order of convergence, as is documented in [24] and demonstrated below.

These extrapolation formulae are substituted into the corresponding terms in (23) to produce the spatial stencil at the irregular cell centers. The stencil at the irregular stencils, therefore, depend not only on concentration values at cell centers but also at grid crossings c_i^b or c_e^b .

3.2.3. Time discretization at freshly cleared cell

At a freshly cleared cell we must modify the time discretization. Such modifications are discussed in [23, 38]. Here, we propose new procedure which is simple to implement with good stability properties.

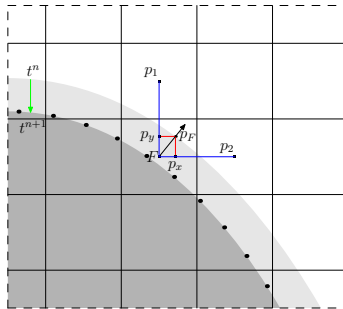


Figure 3: Treatment of freshly cleared grid

Consider the point F in Figure 3. It is a freshly cleared point that was in Ω_i at time $t = n\Delta t$ but is in Ω_e at $t = (n + 1)\Delta t$. In evaluating the time differencing term in (23) at this point, c_F^n has to be available. Since point F was not in Ω_e at time level n , c_F^n is not available. A standard way to obtain this value is to extrapolate the $t = n\Delta t$ level chemicals at neighbor cell centers, at p_1 and p_2 , say, on the intracellular side, to F . This extrapolation could lead to large numerical errors especially in extreme geometric situations. Here we propose a new scheme for the discretization of the time derivative.

First we find the point $p_F = (x_F^*, y_F^*) \in \Gamma_n$ that is closest to the point $F = (x_F, y_F)$. The chemical concentration at point p_F at time $t = n\Delta t$ (on the extracellular side, which we call c_{e,p_F}^n) may be obtained by interpolating the chemical concentration values at grid crossings $c_e^{b,n}$ at time $t = n\Delta t$. A (seemingly) reasonable approximation to the partial time derivative at point F would be:

$$\frac{c_F^{n+1} - c_{e,p_F}^n}{\Delta t}. \quad (27)$$

The above expression, however, will not be a consistent discretization because this differencing corresponds to an “advective” derivative. The velocity of this advection is given by:

$$(\tilde{u}_F, \tilde{v}_F) = \left(\frac{x_F - x_F^*}{\Delta t}, \frac{y_F - y_F^*}{\Delta t} \right). \quad (28)$$

Therefore, (27) must be corrected to remove the advective component resulting from the above velocity. The following is thus a consistent discretization of the time derivative of concentration c_F at point F :

$$\left. \frac{\partial c_F}{\partial t} \right|_{t=n\Delta t} \approx \frac{c_F^{n+1} - c_{e,p_F}^n}{\Delta t} - \tilde{u}_F \mathcal{D}_x^0 c_F^{n+1} - \tilde{v}_F \mathcal{D}_y^0 c_F^{n+1}. \quad (29)$$

where

$$\mathcal{D}_{x,y}^0 w = \frac{1}{2} (\mathcal{D}_{x,y}^+ w + \mathcal{D}_{x,y}^- w). \quad (30)$$

At freshly cleared points, expression (29) is used in place of the $\mathcal{D}_t^- c$ term in (23). The above spatial differencing in many cases involves ghost cell locations. In such cases, extrapolation formulae discussed in Section 3.2.2 are used.

3.2.4. Enforcing chemical boundary condition with auxiliary variables

The chemical boundary conditions (2) are enforced along the membrane Γ , with the help of the auxiliary chemicals defined at grid crossings on both sides of the interface. We first rewrite boundary condition (2) using (11), (3) and (4):

$$j_w c - D \nabla c \cdot \mathbf{n} = \hat{k}_c [c] + \hat{k}_p H(s, c_i, c_e). \quad (31)$$

Let us consider grid crossing (x_1^b, y_1^b) as in Figure 2. The above boundary condition is satisfied on both sides of the membrane. We consider the Ω_i side of the membrane. Our discretization of (31) is:

$$j_w^n c_i^n - D \mathcal{N}(c_i^{b,n+1}, c^{n+1}) = \hat{k}_c [c^{b,n+1}] + \hat{k}_p H(s, c_i^n, c_e^n). \quad (32)$$

The diffusive flux term and the passive membrane flux term are treated implicitly at time $t = (n+1)\Delta t$. This implicit treatment of the boundary condition is the key to stable computations. Our computational experience indicates that an explicit treatment of these terms leads to persistent spurious oscillations of the chemical concentrations near the interfacial boundary. The other terms are evaluated at explicitly, but we point out that these values are not available at the grid crossings; the above equations are defined on the grid crossings at $t = (n+1)\Delta t$ but grid crossings change with every time step. We compute these terms in the following fashion. We take j_w^n and c^n defined at IB points (see equation (22), Section 3.1.2) assign these values to the corresponding IB points at time $t = (n+1)\Delta t$. Then, we interpolate these values to the grid crossing locations.

We now discuss the discretization of the normal derivative (the \mathcal{N} term in (32)). Our procedure follows [24]. Taking intracellular side of the grid crossing at (x_1^b, y_1^b) in Figure 2, the treatment is illustrated as follows.

The unit normal direction \mathbf{n} along Γ can be decomposed into two directions: along grid line \mathbf{v}_a (from point p_1 to A); and off grid line \mathbf{v}_o (from boundary point (x_1^b, y_1^b) through grid point B and stop on grid line between C and p_7). We thus have $\mathbf{n} = a_o \mathbf{v}_o + a_a \mathbf{v}_a$. With these two directions, the normal derivative $\nabla c_i^b \cdot \mathbf{n}$ can be decomposed as a linear combination of directional derivatives along the \mathbf{v}_o and \mathbf{v}_a directions:

$$\begin{aligned} \nabla c_i^b \cdot \mathbf{n} &= a_o \nabla c_i^b \cdot \mathbf{v}_o + a_a \nabla c_i^b \cdot \mathbf{v}_a \\ &= a_o \|\mathbf{v}_o\| \frac{\partial c_i^b}{\partial(\mathbf{v}_o/\|\mathbf{v}_o\|)} + a_a \|\mathbf{v}_a\| \frac{\partial c_i^b}{\partial(\mathbf{v}_a/\|\mathbf{v}_a\|)}. \end{aligned} \quad (33)$$

So along \mathbf{v}_o and \mathbf{v}_a , we need to approximate the partial derivatives, using chemicals at cell centers and the auxiliary variables at grid crossings. If a first order approximation is used, we will have

$$\nabla c_i^b \cdot \mathbf{n} \approx a_o (c_B - c_i^b) + a_a (c_{p_5} - c_i^b), \quad (34)$$

where c_B is the chemical at point B , and c_i^b is auxiliary chemical at the grid crossing on intracellular side. The direction vectors are given by

$$\mathbf{v}_o = (x_B - x_1^b, y_B - y_1^b), \quad \mathbf{v}_a = (x_{p_5} - x_1^b, y_{p_5} - y_1^b). \quad (35)$$

In choosing the cell centers used for this process, we avoid using cell centers that are directly adjacent to the grid crossing. Otherwise, \mathbf{v}_o or \mathbf{v}_a can be arbitrarily small in length, and the coefficient a_o and a_a can become arbitrarily large thus leading to possible numerical instabilities.

We use higher order approximations if more cell centers are available. For example if instead of just using point B for the off grid line direction, we may use point B , C , and p_7

$$\frac{\partial c_i^b}{\partial(\mathbf{v}_o/\|\mathbf{v}_o\|)} \approx -\frac{3}{2\|\mathbf{v}_o\|}c_i^b + \frac{2}{\|\mathbf{v}_o\|}c_B - \frac{1}{2\|\mathbf{v}_o\|}[(1-\theta)c_C + \theta c_{p_7}], \quad (36)$$

and if we use the grid crossing, point p_5 and p_6 , we have

$$\frac{\partial c_i^b}{\partial(\mathbf{v}_a/\|\mathbf{v}_a\|)} \approx -\frac{1+\theta}{(2+\theta)\Delta y}c_{p_6} + \frac{2+\theta}{(1+\theta)\Delta y}c_{p_5} - \frac{3+2\theta}{(1+\theta)(2+\theta)\Delta y}c_i^b. \quad (37)$$

The θ here in (37) and (36) is the same as in (25).

3.2.5. Linear Solver

The resulting linear equations have, as unknowns, the concentrations at the cell centers as well as the intracellular and extracellular concentrations at grid-crossings. The linear equation is non-symmetric, and we use GMRES with Jacobi preconditioning to solve this system.

4. Test cases and numerical convergence study

To validate our numerical scheme for simulating chemical advection-diffusion in a moving domain, and the coupling of chemical osmosis effects and fluid structure interactions, we design several numerical test cases as follows. We will first test the chemical module only, and then the coupling with fluid flow. In all the examples discussed, parameters are dimensionless.

4.1. Chemical evolution with prescribed cell motion and background flow

We assume that the computational domain, the $[0, 1] \times [0, 1]$ unit square, is filled with an incompressible viscous fluid, and there is a cell with initial configuration $\mathbf{r}(s)$ immersed in the domain,

$$\mathbf{r}(s) = \langle 0.5 + 0.25 \cos s, 0.5 + 0.25 \sin s \rangle. \quad (38)$$

To verify the accuracy of chemical update module, we consider the situation in which the flow field and the immersed boundary locations are prescribed. The update of chemicals in the whole domain is done with prescribed analytical background fluid velocity \mathbf{u} and cell motion $d\mathbf{X}/dt$, as follows:

$$\mathbf{u} = \langle u(x, y), v(x, y) \rangle = \langle \frac{1}{4} - (y - \frac{1}{2})^2, 0 \rangle; \quad (39)$$

$$\frac{d\mathbf{X}}{dt} = \langle (\frac{1}{4} - (y - \frac{1}{2})^2) - \frac{1}{2} \cos t, 0 \rangle. \quad (40)$$

The prescribed background velocity (39) and cell motion (40) will be used in the chemical boundary condition (2), and (39) in evaluating advection term in the chemical diffusion advection equation (1). In the chemical boundary condition (2), we allow passive chemical flux j_c through the membrane with constant rate $k_c = 4$ in (3). We set the chemical pump strength to be $k_p(s) = 0.1$ in (4), so chemicals will be actively moved from the intracellular space to the extracellular space.

To run the chemical module, the diffusion coefficient is set to $D = 0.2$, and initial chemical field is given by

$$c(x, y, 0) = 0.5(1.25 + \sin(2\pi(x - 0.5)))(1 + \sin(2\pi(y - 0.25))). \quad (41)$$

Boundary conditions for the chemicals at the top and bottom of the domain are Dirichlet and set to 0, while the left and right edges will have periodic boundary conditions.

In this test case, the cell initially given in (38) is marked by a series Lagrangian marker points $\mathbf{X}_i = \mathbf{r}(s_i)$, and $s_i = (i - 1)2\pi/N_{\text{ring}}$, $i = 1, \dots, N_{\text{ring}}$. The velocity of these marker points is prescribed in (40). We should note that here the chemical boundary condition (2) is implemented without using the water flux term j_w , since it is not available due to the problem set up. The difference in background velocity \mathbf{u} and prescribed cell motion is used instead.

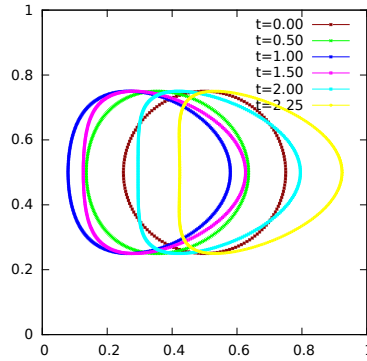


Figure 4: Test case 1. Cell snapshots with its movement prescribed.

Snapshots of the moving cell are plotted in Figure 4. We can see that the cell moves against the background flow to the left first and then to the right, due to the prescribed cell motion. In Figure 5 and 6, snapshots of chemical field with cell locations at those time stamps shown in Figure 4 are plotted using results on a 128×128 grid with $N_{\text{ring}} = 320$.

In Table 1, we list the convergence rates. To obtain these rates, we use ratios of differences of computed solutions. Set:

$$\epsilon_{\{\Delta x, \Delta t\}} = c_{\{\Delta x, \Delta t\}} - \mathcal{I}c_{\{\Delta x/2, \Delta t/2\}}, \quad (42)$$

where subscript $\{\Delta x, \Delta t\}$ indicates levels of space and time spacing, and \mathcal{I} is the interpolation operator to the coarser grid. We refine Δs proportionally to Δx , i.e., the number of immersed boundary marker points is doubled as the Eulerian grid spacing is halved. The error ratio is computed by:

$$r_p(\Delta x, \Delta t) = \frac{\|\epsilon_{\{\Delta x, \Delta t\}}\|_p}{\|\epsilon_{\{\Delta x/2, \Delta t/2\}}\|_p} \quad (43)$$

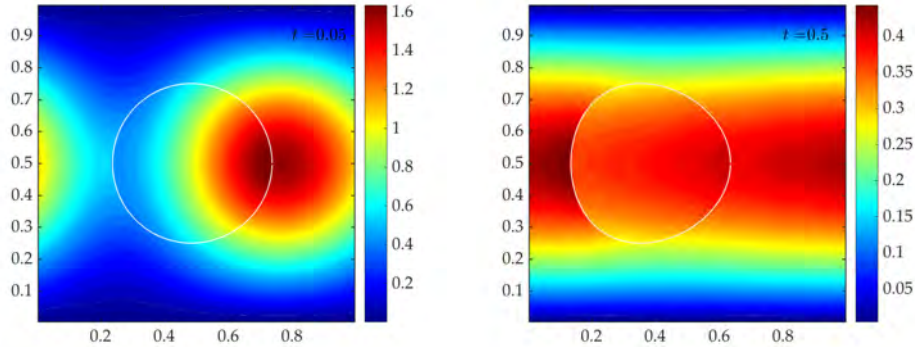


Figure 5: Test case 1. Chemical field at $t = 0.05$ and $t = 0.5$, with cell locations plotted. Figures are generated using numerical solution on 128×128 grid.

where the subscript $p = 2, \infty$ denotes the following norms defined for grid functions ϕ :

$$\|\phi\|_2 = \left(\sum_{(i,j) \in \mathcal{S}} \phi_{i,j}^2 (\Delta x)^2 \right)^{1/2}, \quad \|\phi\|_\infty = \max_{(i,j) \in \mathcal{S}} |\phi_{i,j}|. \quad (44)$$

The set \mathcal{S} in the definition of the above norms is a subset of the grid point indices. We take \mathcal{S} to be either the intracellular or extracellular grid points (at the spatial refinement level Δx).

To assess the convergence of concentration near the immersed boundary, we consider the intracellular and extracellular concentration at the IB points c_i^{IB} and c_e^{IB} (see Section 3.1.2 and (22)). The error ratios are computed in a way that is analogous to the ratios discussed above for the concentrations on the MAC grid.

From the results shown in Table 1, we see the convergence rates are first order while the cell moves against or along the direction of the background fluid flow. In fact, convergence for the chemical at cell centers exhibit convergence rates that are greater than first order. We now build upon this chemical module to test the coupling of chemical osmosis and fluid structure interactions as in the following section.

4.2. Osmotic Engine with Cell Deformation

Now we discuss the situation in which osmotic water flow and fluid structure interaction are fully coupled. This simulation is inspired by the recent experiments in [28]. In [28], the authors reported that a certain cancer cell, confined to a one-dimensional geometry, can propel itself forward even when its cytoskeletal machinery is pharmacologically disrupted. The authors suggest that this is due to an osmotic engine mechanism. The cell pumps in solute at the front and pumps out solute at the back. This induces an osmotic pressure difference at the front and back cell membranes, inducing water flow. The authors argue that this can propel the cell forward.

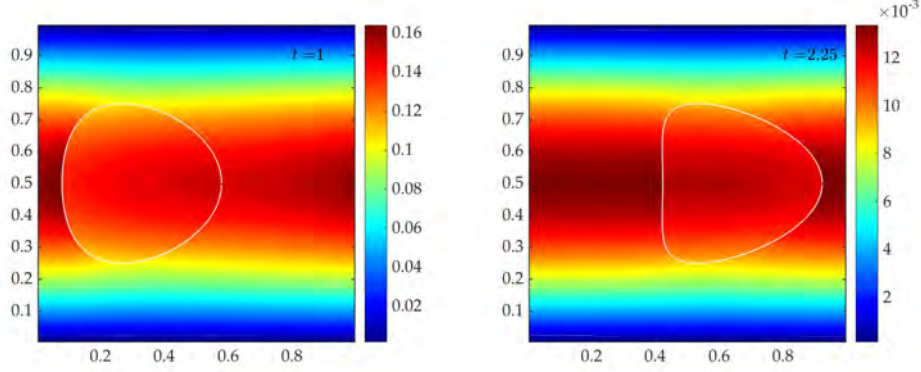


Figure 6: Test case 1. Chemical field developed at $t = 1$ and $t = 2.25$, with cell locations plotted. Figures are generated using numerical solution on 128×128 grid.

In what follows, we present a computational simulation inspired by this result. We place active pumps at the front and back of the cell, and demonstrate that the cell can move forward. For simulations in this and next subsections, we choose typical biophysical parameters in the literature for membrane elasticity, permeability and chemical concentration [39, 40, 41]. Some of these parameters are listed in Table 2. In Section 4.3 we vary certain parameter values and study its effect on the cell speed. These studies will give us insight into the feasibility of the osmotic engine mechanism in two (or more) spatial dimension, beyond the one dimensional setting of the experiments.

In our test case here, a flexible elastic membrane in the shape of ellipse is set initially at the center of a square $[0, 100\mu m] \times [0, 100\mu m]$:

$$\mathbf{r}(s) = \langle x_c + r_x \cos s, y_c + r_y \sin s \rangle, \quad (45)$$

with $x_c = 50\mu m$, $y_c = 50\mu m$ and $r_x = r_y = 19\mu m$. There are N_{ring} immersed boundary points located at \mathbf{X}_i , to discretize the membrane $\Gamma(t)$, with $\mathbf{X}_i = \mathbf{r}(s_i)$, $s_i = (i - 1)2\pi/N_{\text{ring}}$, $i = 1, \dots, N_{\text{ring}}$, as used in last test case. Elastic properties of the cell membrane are determined by the three parameters in (7). We set elastic stiffness of the link to $k_{\text{elas}} = 0.001Pa$, and rest length $\ell = 3.125\mu m$.

The permeability of the elastic membrane is controlled by three factors: the passive chemical flux j_c and the active chemical pump j_p in the chemical boundary condition (2), and the water flux j_w in the velocity slip boundary condition (11). Here, it is assumed the active pump j_p in (4) is distributed along the membrane $\Gamma(t)$ to move chemicals from one side of the cell to the other. The strength of the pump in (4) is now coordinate-dependent and asymmetric in the x direction, prescribed by

$$k_p(s) = -k_h \left(e^{-\frac{s^2}{h_l^2}} + e^{-\frac{(s-2\pi)^2}{h_l^2}} \right) + k_t e^{-\frac{(s-\pi)^2}{t_l^2}} \quad (46)$$

where $k_h = 0.06\mu m/s$, $k_t = 0.12\mu m/s$, $h_l = 2t_l = \pi/8$ and $s \in [0, 2\pi]$ is the material coordinates winding counter clock-wise. The value of k_p is chosen to be comparable to k_c ,

| | $t = 0.25$ | $t = 0.5$ | $t = 1$ | $t = 1.5$ | $t = 2$ | $t = 1$ | $t = 1$ |
|-------------------------------|------------|-----------|---------|-----------|---------|----------|----------|
| $c, \text{ ext. } (L^2)$ | 0.9937 | 0.8001 | 0.8280 | 1.4013 | 1.6716 | 3.041E-4 | 1.721E-5 |
| $c, \text{ int. } (L^2)$ | 1.1085 | 1.2015 | 0.9422 | 1.4024 | 1.6363 | 5.760E-4 | 3.011E-4 |
| $c, \text{ ext. } (L^\infty)$ | 0.6376 | 0.7954 | 1.1509 | 1.6642 | 1.5434 | 1.957E-3 | 8.811E-4 |
| $c, \text{ int. } (L^\infty)$ | 1.2252 | 1.7286 | 0.9772 | 1.6260 | 1.3828 | 5.112E-3 | 2.596E-3 |
| $c_e^{\text{IB}} (L^2)$ | 0.9788 | 0.9605 | 0.9909 | 1.0154 | 1.0727 | 4.663E-3 | 2.346E-3 |
| $c_i^{\text{IB}} (L^2)$ | 1.0036 | 0.9183 | 1.0020 | 1.0687 | 1.0901 | 3.540E-3 | 1.768E-3 |
| $c_e^{\text{IB}} (L^\infty)$ | 0.8823 | 0.8740 | 0.7502 | 1.0526 | 1.0008 | 9.855E-3 | 5.850E-3 |
| $c_i^{\text{IB}} (L^\infty)$ | 1.0090 | 0.7308 | 0.7412 | 1.2616 | 0.9478 | 9.006E-3 | 5.394E-3 |

Table 1: Test case 1. Convergence rates of chemical c at cell centers and IB points, at different time points for the simulation with prescribed velocity field and IB locations. Listed are convergence rates calculated in the L^2 and L^∞ norm, where the rates are computed as described in the text with $\Delta x = 1/64$, and $\Delta t = 1/200$, and $\Delta s = 2\pi/160$. The labels int. and ext. denote intracellular and extracellular cell centers. The last two column shows relative error at $t = 1$, which are calculated using $\|\epsilon_{\{\Delta t, \Delta x\}}\|_p / \|w\|_p$, with w being the variables considered, and p being 2 or ∞ . We note w are from data on $\Delta x/2$ and $\Delta t/2$, and $\Delta x/4$ and $\Delta t/4$ for the last two columns, respectively.

| | | | | | |
|-------------------------|---------|-------------------------|--------------------------|-------------------------------------|-----------------------|
| $D : \frac{\mu m^2}{s}$ | $T : K$ | $k_c : \frac{\mu m}{s}$ | $k_w : \frac{m^2 s}{kg}$ | $k_{\text{elas}} : \frac{kg}{ms^2}$ | $\nu : \frac{kg}{ms}$ |
| 10^2 | 300 | 1 | 1.11×10^{-12} | 0.001 | 0.05 |

Table 2: Parameters used in this simulation. Here k_w is from [39], k_{elas} is from [40, 41], and D , ν and k_c are chosen to be within the range people usually used in biological applications.

assigned below. The membrane location at $s = 0$ corresponds to the right most point of the membrane initially. We note the active pump prescribed in (46) will move chemicals from the outside to the inside at the front (right) of the cell, and pump chemicals from the inside to the outside at the back (left) of the cell. The spatial distribution of pumps at the front (right) is twice as wide as the pump at the back (left) of the cell). Passive flux of chemicals across membrane is also considered using j_c of (3), with $k_c = 1\mu m/s$. The osmosis effect is included by setting $k_w = 1.11\mu m^2 s/kg$ in the water flux j_w in the velocity boundary condition along cell membrane (11).

In the computation, we set the diffusion coefficient of the chemical to be $D = 10^2\mu m^2/s$. Chemical concentration is initially set to $1.45 \times 10^{-4} pmol/\mu m^3$ ($145 mmol/\ell$) uniformly over the entire computational domain, and fixed at the top and bottom edges of the computational domain. Here, we have used the representative concentration of sodium, which is a major osmotic contributor. In general, concentrations of osmolytes are in the $100\mu mol/\ell$ range. At the left and right edges of the computational domain, we have periodic chemical boundary condition. The velocity of the fluid is set to 0 initially and the viscosity ν of the fluid is set to be at temperature $T = 300K$: $0.05 Pa \cdot s$. At the top and bottom edges of the computational domain, we have no slip velocity boundary condition and periodic boundary condition at the left and right edges.

The pump creates a concentration difference across the cell membranes at the front and back, which in turn generates a fluid flow. With this fluid flow, the elastic membrane, which is in the shape of ellipse initially, becomes concave at the back while the front moves forward (to the right) over time, as shown in the Figure 7 and the Figure 8. As the cell is propelled forward, we also see from the right panel of the Figure 7 that fluid flow will

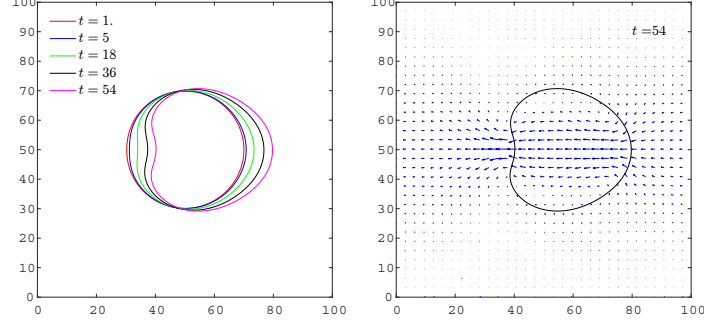


Figure 7: Test case 2. Left panel shows cell translation and deformation over time. Right panel shows velocity profile at $t = 54$ s. Plots are generated using simulation results on 256×256 grid.

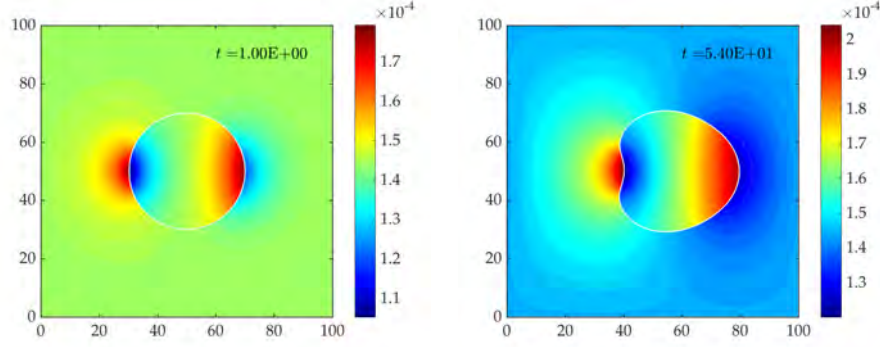


Figure 8: Test case 2. Chemical field developed at $t = 1$ s and $t = 54$ s. Plots are generated using simulations on 256×256 grid, with $N_{\text{ring}} = 640$. Unit of the chemical concentration is $\text{pmol}/\mu\text{m}^3$, and the initial concentration is $1.45 \times 10^{-4} \text{ pmol}/\mu\text{m}^3 (145 \text{ mmol}/\ell)$.

develop vortices around the side and back of the cell. We thus see that osmotic engine mechanism can lead to uni-directional cell movement in a two-dimensional setting. Our simulations produce speeds in the range of $0.01 \sim 0.2 \mu\text{m}/\text{s}$ for different parameter values (see Figure 10, 11 and results in Section 4.3). The lower end of these values compares favorably to the observed speeds on the order of $0.01 \mu\text{m}/\text{s}$ reported in [28], especially given that we have not adjusted our parameters to fit this particular experimental data.

We now study convergence. Convergence rates for solute concentrations c are calculated in the same way as was done in the previous Section. We thus refine Δx and Δt proportionally and test for convergence. Convergence rates for the velocity \mathbf{u} is computed similarly to c except that we do not make the distinction between intracellular and extracellular. For the concentrations at the IB locations \mathbf{X} , the convergence rates are computed using error ratios as was discussed in the previous Section.

The convergence rates in Table 3 show that our numerical scheme gives us first order

accuracy for all the variables in both time and space, and the results are robust over time.

| | $t = 1$ | $t = 5$ | $t = 18$ | $t = 36$ | $t = 54$ | $t = 36$ | $t = 36$ |
|-------------------------------|---------|---------|----------|----------|----------|----------|----------|
| $c, \text{ ext. } (L^2)$ | 1.6345 | 1.0157 | 1.1155 | 0.9351 | 1.1883 | 4.758E-4 | 2.505E-4 |
| $c, \text{ int. } (L^2)$ | 1.1835 | 1.1219 | 1.6057 | 1.1498 | 1.4566 | 1.041E-3 | 4.722E-4 |
| $c, \text{ ext. } (L^\infty)$ | 1.8676 | 1.2409 | 1.3559 | 0.6543 | 1.2756 | 3.102E-3 | 1.960E-3 |
| $c, \text{ int. } (L^\infty)$ | 1.3905 | 1.0501 | 1.5783 | 0.9284 | 1.0145 | 7.487E-3 | 3.913E-3 |
| $c_e^{\text{IB}} (L^2)$ | 1.7171 | 1.8109 | 2.3716 | 1.3956 | 1.0738 | 1.843E-3 | 7.008E-4 |
| $c_i^{\text{IB}} (L^2)$ | 1.2609 | 1.7484 | 2.1079 | 1.4435 | 2.4491 | 6.134E-3 | 2.248E-3 |
| $c_e^{\text{IB}} (L^\infty)$ | 1.9247 | 1.5697 | 2.0211 | 0.8336 | 1.3023 | 8.608E-3 | 4.831E-3 |
| $c_i^{\text{IB}} (L^\infty)$ | 1.1770 | 1.7482 | 1.4565 | 1.9077 | 1.2588 | 1.781E-2 | 4.735E-3 |
| $\mathbf{X} (L^2)$ | 1.1407 | 1.0137 | 1.0385 | 1.0184 | 1.1556 | 1.113E-3 | 5.496E-4 |
| $\mathbf{X} (L^\infty)$ | 1.0478 | 1.0620 | 1.1862 | 1.0291 | 1.0854 | 3.279E-3 | 1.605E-3 |
| $\mathbf{u} (L^2)$ | 1.0363 | 1.0127 | 1.0208 | 1.0005 | 1.0898 | 1.928E-3 | 9.652E-4 |
| $\mathbf{u} (L^\infty)$ | 0.7780 | 1.0466 | 0.9752 | 0.5451 | 0.7845 | 3.715E-2 | 2.524E-2 |

Table 3: Test case 2. Convergence rates for $c, c_{1,e}^{\text{IB}}, \mathbf{X}$ and \mathbf{u} . Convergence rates are calculated in the L^2 and L^∞ norm, at $\Delta x = 100\mu\text{m}/64$, $\Delta t = 1/200\text{s}$, and $\Delta s = 2\pi/160$. The labels int. and ext. denote the concentrations at intracellular and extracellular cell centers. The last two columns show the relative error of all variables at $t = 36$, which are calculated as in Table 1.

4.3. Parameter Study

The parameters in model (1)-(11) include those that control properties of chemicals, elastic membrane, fluid flow and their interactions. In the computational examples in this section, we vary some of the parameters to examine their impact on cell shape and speed. All calculations in this section are carried out in the $100\mu\text{m} \times 100\mu\text{m}$ square on a 128×128 spatial grid with fixed time step $\Delta t = 5 \times 10^{-3}\text{s}$.

First, we study the impact of the two parameter t_l and h_l in (46) on cell shape. We set

$$t_l = k_l h_l, \quad h_l = \pi/8 \quad (47)$$

where k_l is varied from $1/2$ to 1 . Note that we took $k_l = 1/2$ in the convergence study of the previous section. When $k_l = 1$, the distribution of the pumps have the same width at the front and back of the cell (see Figure 9). As a measure of the change in cell shape, we consider the integral of the absolute value of the curvature along the cell:

$$\int_{\Gamma(t)} |\kappa| ds \quad (48)$$

where κ is the curvature and ds denotes integration against the arclength coordinate. If the above quantity is greater than 2π , we see that the cell shape is concave.

We performed simulations using the same parameters as in the test case in the previous section except for the above modification in pump distribution. The initial conditions for the chemical, fluid velocity and cell shape are also the same as in the previous section. The concavity measure versus time t is shown in the right panel of the Figure 9 for different values of k_l . The pump distribution plotted against material coordinate s

is given in the left panel of the same figure. We can see that with narrower pump at the back, the earlier the cell to becomes concave while in translation. With wider pump distribution, the cell starts to translate while remaining convex.

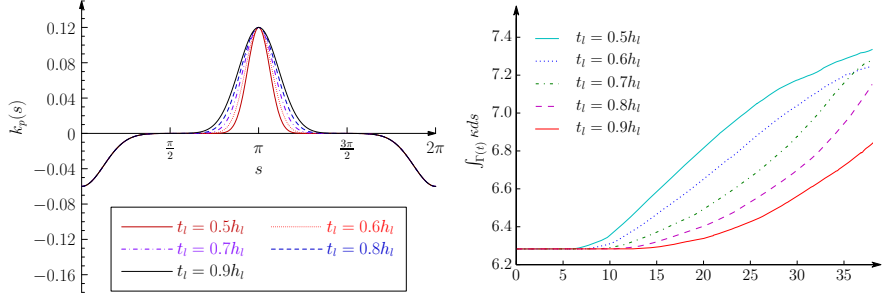


Figure 9: Comparison of the development of cell concavity with different chemical pump distribution. The pump distribution is shown in the left panel. The integral of the absolute value of the curvature over entire cell vs time t is plotted in the right panel.

Finally, we assess the effect of k_w, k_c and D on the speed of the cell, while keeping $k_h = k_t = 0.06$, and $h_l = t_l = \pi/5$. To maintain roughly a circle shape for cell (without concavity along cell membrane), we use relatively high elasticity for membrane $k_{\text{elas}} = 0.02Pa$. Other constants as well as the initial conditions remain the same as in the test case of the previous section. The speed of the cell is determined by the distance traveled by the x -component of the center of mass between $t = 0s$ and $t = 40s$.

In left panel of Figure 10, we plot cell speed as a function of k_w for different choices of k_c and D . This graph clearly shows that the speed is an increasing function of k_w . This result is quite natural; a cell membrane that is more permeable to water can generate a stronger fluid flow.

In right panel of Figure 10, we plot the speed as a function of k_c for different choices of k_w and D . We note in particular that the study with large values of k_c is only possible thanks to the implicit treatment of the solute boundary conditions. The speed is generally a decreasing function of k_c . A larger chemical permeability tends to dissipate the osmotic gradient across the cell membrane, thus leading to a slower flow. It is interesting that, as k_c becomes large, the effects of D on the speed become less pronounced.

In Figure 11, we plot cell speed as a function of chemical diffusion coefficient D . We see that the speed decreases with D , due to the fact that diffusion will tend to equalize concentrations within the intracellular and extracellular regions, which will again lead to a decrease in the osmotic gradient along the cell membrane.

The above results indicate that small chemical diffusion and/or solute membrane permeability together with a large water permeability will lead to the greatest speed under the osmotic engine mechanism. With small diffusion coefficient and solute permeability, the solute concentration difference across the membrane generated by the pump does not easily dissipate, leading to sustained osmotic driving force. This driving force can be more easily turned into fluid motion if the membrane water permeability is high. Our computational results thus indicate that a small diffusion coefficient will make it easier for the osmotic engine mechanism to work in higher dimension, beyond the one dimensional setting for which the mechanism was proposed in [28].

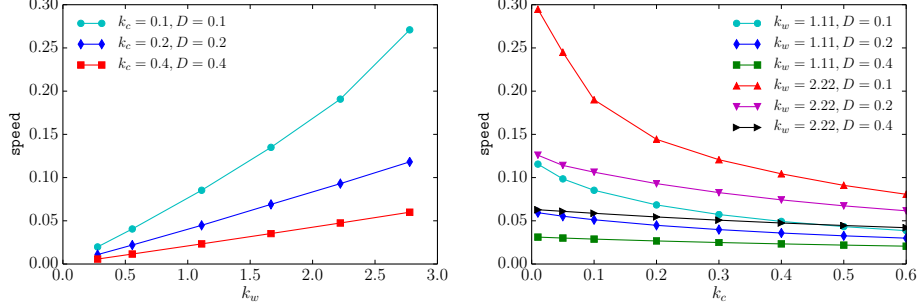


Figure 10: Left panel: the speed (unit $\mu\text{m/s}$) calculated to $t = 40\text{s}$ for the average of the x -coordinate of IB points along the cell (center of mass), vs water flux coefficient k_w (unit $\mu\text{m}^2/\text{kg}$). Right panel: the speed calculated to $t = 40\text{s}$ for the average of the x -coordinate of IB points along the cell (center of mass), vs water passive chemical flux constant k_c (unit $\mu\text{m/s}$), at different diffusion coefficient D (unit $10^3 \mu\text{m}^2/\text{s}$) and k_w .

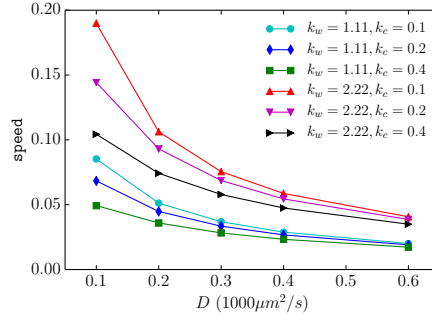


Figure 11: The speed calculated to $t = 40\text{s}$ for the average of the x -coordinate of the IB points along the cell (center of mass), vs chemical diffusion coefficient D , at fixed chemical flux constant k_c and water flux constant k_w .

5. Discussion

In this paper, we developed a numerical method to study the interplay between chemical diffusion and osmosis in the presence of a deformable cell membrane in a Stokes fluid. A salient feature of the problem is that the membrane interface has a slip velocity with respect to the underlying fluid velocity, which is controlled in part by the jump in chemical concentration across the membrane. The numerical framework consists of the chemical module and the fluid-structure interaction module. We use the immersed boundary method for the fluid-structure interaction problem, and a Cartesian grid embedded boundary method for the chemical. In contrast to previous work that deal with similar problems, we treat the membrane chemical interface conditions implicitly, which we have found to be key for stable computations. We also propose a new way of computing time derivative at freshly cleared points, and use a partially implicit treatment of the fluid-structure boundary conditions.

We test the accuracy of our numerical method, first of just the chemical module,

and then of our entire scheme which couples the fluid-structure interaction module and the chemical module. In testing our full scheme, we use a 2D osmotic engine problem as our test case. Our computations demonstrate that the osmotic engine mechanism is biophysically feasible beyond the one spatial dimension in which it was proposed in [28]. In a further parameter study, we examine the effect of different parameters on the osmotic engine mechanism. Membrane elasticity influences the cell shape and speed. Our results also indicate that membrane permeability to water and the diffusion/permeability constants of the solute are crucial control parameters that allow for success of the osmotic engine.

There are numerous improvements that can be made of the current numerical method. Our numerical method is first order accurate in space and time. To obtain higher order accuracy, discretization of interface conditions of the chemical module must be handled with greater care. This may be challenging given that the boundary is moving. We also point out that the immersed boundary method is an inherently first-order accurate scheme. Higher order accuracy may require replacing the IB method with a sharp interface method such as the immersed interface method [25]. Another route to greater accuracy is adaptive mesh refinement. Adaptive mesh refinement, especially around the cell membrane, should lead to efficient computation.

Currently, the matrix problem arising from the chemical module is solved using GMRES with Jacobi preconditioning. This is the simplest conceivable iterative method; an exploration of better preconditioners should result in much faster convergence. A major possible impediment to efficient computations may be the time-step restriction arising from stiff mechanical forces from the membrane. We will be testing implicit IB methods [42, 43, 44, 45] to alleviate this potential difficulty.

To better test the osmotic engine mechanism and to study the interplay between osmotic forces and cell mechanics more broadly, we must construct better models with parameter values that better reflect cell biophysics. Mechanical models in the bulk and the membrane beyond mere (Navier-)Stokes flow and passive elasticity/bending must be considered. Most osmotically active solutes are charged; instead of chemical diffusion, it would be essential to include ionic electrodiffusion, the theoretical framework of which was presented in [36].

Finally, our 2D simulations must be extended to 3D. Extension of our algorithm to 3D in both fluid structure interactions module and chemical reaction advection diffusion module are in principle straightforward. The difficulty in the implementation may be in designing proper data structure and mappings so the construction of the linear system is more efficient and so the process is suitable for parallel computations.

Another challenge in adopting the method to 3D is to quickly detect grid crossings and identify necessary nearby cell centers for stencils. The explicitly tracked IB points will enable local construction of surface representation and then grid crossings will be obtained accordingly. We would like to investigate the capability of available software for the geometry detection and generation of the grid crossings so we can focus on improving the numerical scheme.

The computational costs in 3D simulations could be the most important factor to consider in our implementation. We may need to utilize adaptive mesh refinement and parallel computing to save computing costs. It would be hard to obtain fast solver for the chemical module, which is expected to be more computationally costly compared to the fluid module. By carefully designing the enforcement of boundary conditions in

chemicals on different levels, we may be able to develop multigrid solver to speed up the computation in chemicals.

Acknowledgments We thank Sean Sun, Alex Mogilner and Aaron Fogelson for useful discussion. This work was supported by NSF DMS-1620198 to L.Y. and NSF DMS-1620316 to Y.M.

Appendix A. Derivation of Free Energy Identity

Let us now derive (12). First, multiply (1) with μ in (12) and integrate over Ω_i .

$$\int_{\Omega_i} \mu \left(\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) \right) d\mathbf{x} = \int_{\Omega_i} \mu \nabla \cdot \left(c \frac{D}{RT} \nabla \mu \right) d\mathbf{x}. \quad (\text{A.1})$$

The right hand side becomes:

$$\int_{\Omega_i} \mu \nabla \cdot \left(c \frac{D}{RT} \nabla \mu \right) d\mathbf{x} = \int_{\Gamma_i} \left(\mu c \frac{D}{RT} \nabla \mu \cdot \mathbf{n} \right) dm_\Gamma - \int_{\Omega_i} \left(c \frac{D}{RT} |\nabla \mu|^2 \right) d\mathbf{x}. \quad (\text{A.2})$$

Consider the left hand side of (A.1).

$$\mu \left(\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) \right) = \frac{\partial \omega}{\partial c} \left(\frac{\partial c}{\partial t} + \mathbf{u} \cdot \nabla c \right) = \frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega), \quad (\text{A.3})$$

where ω was defined in (12), and we used the incompressibility condition in (5). Integrating the above over Ω_i , we have:

$$\begin{aligned} \int_{\Omega_i} \left(\frac{\partial \omega}{\partial t} + \nabla \cdot (\mathbf{u}\omega) \right) d\mathbf{x} &= \int_{\Omega_i} \frac{\partial \omega}{\partial t} d\mathbf{x} + \int_{\Gamma_i} \omega \mathbf{u} \cdot \mathbf{n} dm_\Gamma \\ &= \frac{d}{dt} \int_{\Omega_i} \omega d\mathbf{x} + \int_{\Gamma_i} \omega \left(\mathbf{u} - \frac{\partial \mathbf{X}}{\partial t} \right) \cdot \mathbf{n} dm_\Gamma \end{aligned} \quad (\text{A.4})$$

where we used the fact that \mathbf{u} is divergence free in the first equality. The term involving $\frac{\partial \mathbf{X}}{\partial t}$ comes from the fact that the membrane Γ is moving in time. Performing similar calculations on Ω_e , and adding this to the above, we find:

$$\begin{aligned} &\frac{d}{dt} \int_{\Omega_i \cup \Omega_e} \omega d\mathbf{x} + \int_{\Gamma} [\omega] j_w dm_\Gamma \\ &= \int_{\Gamma} \left[\mu c \frac{D}{RT} \nabla \mu \cdot \mathbf{n} \right] dm_\Gamma - \int_{\Omega_i \cup \Omega_e} \left(c \frac{D}{RT} |\nabla \mu|^2 \right) d\mathbf{x}. \end{aligned} \quad (\text{A.5})$$

where we used (9). Using (2) and (9), we may rewrite the second boundary integral as follows:

$$\int_{\Gamma} \left[\mu c \frac{D}{RT} \nabla \mu \cdot \mathbf{n} \right] dm_\Gamma = \int_{\Gamma} ([\mu c] j_w - [\mu](j_c + j_p)) dm_\Gamma. \quad (\text{A.6})$$

We now turn to equation (5). Multiply this by \mathbf{u} and integrate over Ω_i :

$$0 = \int_{\Omega_i} \mathbf{u} \cdot (\nabla \cdot \Sigma_m(\mathbf{u}, p)) d\mathbf{x} = \int_{\Gamma_i} (\Sigma_m(\mathbf{u}, p) \mathbf{n}) \cdot \mathbf{u} dm_\Gamma - \int_{\Omega_i} 2\nu |\nabla_S \mathbf{u}|^2 d\mathbf{x} = 0. \quad (\text{A.7})$$

Performing a similar calculation on Ω_e and adding this to the above, we have:

$$0 = \int_{\Gamma} [(\Sigma_m(\mathbf{u}, p)\mathbf{n})] \cdot \mathbf{u} dm_{\Gamma} - \int_{\Omega_i \cup \Omega_e} 2\nu |\nabla_S \mathbf{u}|^2 d\mathbf{x}. \quad (\text{A.8})$$

Note that:

$$\begin{aligned} & \int_{\Gamma} [(\Sigma_m(\mathbf{u}, p)\mathbf{n})] \cdot \frac{\partial \mathbf{X}}{\partial t} dm_{\Gamma} \\ &= \int_{\Gamma_{\text{ref}}} (\mathbf{F}_{\text{elas}} + \mathbf{F}_{\text{bend}}) \cdot \frac{\partial \mathbf{X}}{\partial t} ds = E_{\text{mem}}(\mathbf{X}), \end{aligned} \quad (\text{A.9})$$

where we used (7) in the first equality and integrated by parts in the second equality.

We may use (6), (A.9) and (9) to find

$$\begin{aligned} & \frac{d}{dt} E_{\text{mem}}(\mathbf{X}) \\ &= \int_{\Gamma} [(\Sigma_m(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{n}] j_w dm_{\Gamma} - \int_{\Omega_i \cup \Omega_e} 2\nu |\nabla_S \mathbf{u}|^2 d\mathbf{x}. \end{aligned} \quad (\text{A.10})$$

Combining (A.5), (A.6) and (A.10), we have:

$$\begin{aligned} & \frac{d}{dt} (E_{\text{bulk}} + E_{\text{mem}}(\mathbf{X})) \\ &= - \int_{\Omega_i \cup \Omega_e} \left(c \frac{D}{RT} |\nabla \mu|^2 \right) d\mathbf{x} - \int_{\Omega_i \cup \Omega_e} 2\nu |\nabla_S \mathbf{u}|^2 d\mathbf{x} \\ & \quad - \int_{\Gamma} [\mu] (j_c + j_p) dm_{\Gamma} - \int_{\Gamma} [\omega - c\mu - (\Sigma_m(\mathbf{u}, p)\mathbf{n}) \cdot \mathbf{n}] j_w dm_{\Gamma} \end{aligned} \quad (\text{A.11})$$

Since $\omega - c\mu = -RTc$, this completes our derivation of (12).

- [1] J. Waniewski, Mathematical modeling of fluid and solute transport in hemodialysis and peritoneal dialysis, *Journal of Membrane Science* 274 (1) (2006) 24–37.
- [2] S. Sablani, M. Goosen, R. Al-Belushi, M. Wilf, Concentration polarization in ultrafiltration and reverse osmosis: a critical review, *Desalination* 141 (3) (2001) 269–289.
- [3] A. Weinstein, Mathematical models of tubular transport, *Annual review of physiology* 56 (1) (1994) 691–709.
- [4] W. Boron, E. Boulpaep, *Medical physiology*, 2nd Edition, W.B. Saunders, 2008.
- [5] M. Shibayama, T. Tanaka, Volume phase transition and related phenomena of polymer gels, *Responsive gels: volume transitions I* (1993) 1–62.
- [6] T. Tanaka, Collapse of gels and the critical endpoint, *Physical Review Letters* 12 (1978) 820–823.
- [7] J. F. Brady, Brownian motion, hydrodynamics, and the osmotic pressure, *The Journal of chemical physics* 98 (4) (1993) 3335–3341.
- [8] G. Oster, C. S. Peskin, Dynamics of osmotic fluid flow, in: *Mechanics of Swelling*, Springer, 1992, pp. 731–742.
- [9] A. Schwab, A. Fabian, P. J. Hanley, C. Stock, Role of ion channels and transporters in cell migration, *Physiological Reviews* 92 (4) (2012) 1865–1913.
- [10] V. M. Loitto, T. Karlsson, K.-E. Magnusson, Water flux in cell motility: expanding the mechanisms of membrane protrusion, *Cell Motil Cytoskeleton* 66 (5) (2009) 237–247.
- [11] M. Papadopoulos, S. Saadoun, A. Verkman, Aquaporins and cell migration, *Pflügers Archiv-European Journal of Physiology* 456 (4) (2008) 693–700.
- [12] A. Layton, Modeling water transport across elastic boundaries using an explicit jump method, *SIAM Journal on Scientific Computing* 28 (2006) 2189.

- [13] P. Jayathilake, Z. Tan, B. Khoo, N. Wijesundera, Deformation and osmotic swelling of an elastic membrane capsule in stokes flows by the immersed interface method, *Chemical Engineering Science* 65 (3) (2010) 1237–1252.
- [14] P. Jayathilake, B. Khoo, Z. Tan, Effect of membrane permeability on capsule substrate adhesion: Computation using immersed interface method, *Chemical Engineering Science* 65 (11) (2010) 3567–3578.
- [15] C. Vogl, M. Miksis, S. Davis, D. Salac, The effect of glass-forming sugars on vesicle morphology and water distribution during drying, *Journal of The Royal Society Interface* 11 (99) (2014) 20140646.
- [16] P. Lee, B. Griffith, C. Peskin, The immersed boundary method for advection-electrodifffusion with implicit timestepping and local mesh refinement, *Journal of computational physics* 229 (13) (2010) 5208–5227.
- [17] F. Lippoth, M. A. Peletier, G. Prokert, A moving boundary problem for the stokes equations involving osmosis: variational modelling and short-time well-posedness, *European Journal of Applied Mathematics* (2014) 1–20.
- [18] C. Peskin, The immersed boundary method, *Acta Numerica* 11 (2002) 479–517.
- [19] Y. Kim, C. S. Peskin, 2-d parachute simulation by the immersed boundary method, *SIAM Journal on Scientific Computing* 28 (6) (2006) 2294–2312.
- [20] Y. Kim, C. S. Peskin, 3-d parachute simulation by the immersed boundary method, *Computers & Fluids* 38 (6) (2009) 1080–1090.
- [21] J. M. Stockie, Modelling and simulation of porous immersed boundaries, *Computers & Structures* 87 (11) (2009) 701–709.
- [22] W. Strychalski, C. A. Copos, O. L. Lewis, R. D. Guy, A poroelastic immersed boundary method with applications to cell biology, *Journal of Computational Physics* 282 (2015) 77–97.
- [23] P. McCorquodale, P. Colella, H. Johansen, A cartesian grid embedded boundary method for the heat equation on irregular domains, *Journal of Computational Physics* 173 (2001) 620–635.
- [24] P. Macklin, J. S. Lowengrub, A new ghost cell/level set method for moving boundary problems: application to tumor growth, *Journal of scientific computing* 35 (2-3) (2008) 266–299.
- [25] Z. Li, K. Ito, *The Immersed Interface Method: Numerical Solutions of PDEs Involving Interfaces and Irregular Domains*, SIAM, 2006.
- [26] Z. Li, K. Ito, M.-C. Lai, An augmented approach for stokes equations with a discontinuous viscosity and singular forces, *Computers & Fluids* 36 (3) (2007) 622–635.
- [27] W.-F. Hu, M.-C. Lai, Y.-N. Young, A hybrid immersed boundary and immersed interface method for electrohydrodynamic simulations, *Journal of Computational Physics* 282 (2015) 47–61.
- [28] K. M. Stroka, H. Jiang, S.-H. Chen, Z. Tong, D. Wirtz, S. X. Sun, K. Konstantopoulos, Water permeation drives tumor cell migration in confined microenvironments, *Cell* 157 (3) (2014) 611–623.
- [29] J. L. Anderson, Movement of a semipermeable vesicle through an osmotic gradient, *Physics of Fluids* (1958-1988) 26 (10) (1983) 2871–2879.
- [30] D. Zinemanas, A. Nir, Osmophoretic motion of deformable particles, *International journal of multiphase flow* 21 (5) (1995) 787–800.
- [31] U. M. Córdoba-Figueroa, J. F. Brady, Osmotic propulsion: The osmotic motor, *Physical Review Letters* 100 (2008) 158303. doi:10.1103/PhysRevLett.100.158303. URL <https://link.aps.org/doi/10.1103/PhysRevLett.100.158303>
- [32] S. Shklyae, J. F. Brady, U. M. Córdoba-Figueroa, Non-spherical osmotic motor: Chemical sailing, *Journal Of Fluid Mechanics* 748 (2014) 488–520.
- [33] R. Golestanian, T. B. Liverpool, A. Ajdari, Propulsion of a molecular machine by asymmetric distribution of reaction products, *Physical review letters* 94 (22) (2005) 220801.
- [34] P. Atzberger, S. Isaacson, C. Peskin, A microfluidic pumping mechanism driven by non-equilibrium osmotic effects, *Physica D: Nonlinear Phenomena* 238 (14) (2009) 1168–1179.
- [35] C.-H. Wu, T. G. Fai, P. J. Atzberger, C. S. Peskin, Simulation of osmotic swelling by the stochastic immersed boundary method, *SIAM Journal on Scientific Computing* 37 (4) (2015) B660–B688.
- [36] Y. Mori, C. Liu, R. Eisenberg, A Model of Electrodifffusion and Osmotic Water Flow and its Energetic Structure, *Physica D: Nonlinear Phenomena* 240 (2011) 1835–1852.
- [37] L. Yao, A. L. Fogelson, Simulations of chemical transport and reaction in a suspension of cells i: an augmented forcing point method for the stationary case, *International Journal for Numerical Methods in Fluids* 69 (11) (2012) 1736–1752.
- [38] J. Yang, E. Balaras, An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries, *Journal of Computational Physics* 215 (1) (2006) 12–40.
- [39] Y. Li, Y. Mori, S. Sun, Flow-driven cell migration under external electric fields, *Physical Review*

- Letters 115 (2015) 268101.
- [40] B. Vanderlei, J. J. Feng, L. Edelstein-Keshet, A computational model of cell polarization and motility coupling mechanics and biochemistry, *SIAM Multiscale Modeling and Simulation* 9 (2011) 1420–1443.
 - [41] K. Keren, Z. Pincus, G. M. Allen, E. L. Barnhart, G. Marriott, A. Mogilner, J. A. Theriot, Mechanism of shape determination in motile cells, *Nature* 453 (2008) 475–480.
 - [42] Y. Mori, C. S. Peskin, Implicit second-order immersed boundary methods with boundary mass, *Computer methods in applied mechanics and engineering* 197 (25) (2008) 2049–2067.
 - [43] E. P. Newren, A. L. Fogelson, R. D. Guy, R. M. Kirby, Unconditionally stable discretizations of the immersed boundary equations, *Journal of Computational Physics* 222 (2) (2007) 702–719.
 - [44] R. D. Guy, B. Philip, A multigrid method for a model of the implicit immersed boundary equations, *Communications in Computational Physics* 12 (02) (2012) 378–400.
 - [45] R. D. Guy, B. Philip, B. E. Griffith, Geometric multigrid for an implicit-time immersed boundary method, *Advances in Computational Mathematics* 41 (3) (2015) 635–662.