

A Two-dimensional Cell Motility Model

Ravi K Vuta, Worcester Polytechnic Institute, Massachusetts, ravivuta@wpi.edu
Roger Lui, Worcester Polytechnic Institute, Massachusetts, rlui@wpi.edu¹
Gretar Tryggvason, Worcester Polytechnic Institute, Massachusetts, gretar@wpi.edu

Abstract

A simple two-dimensional continuum model of a cell crawling on a surface is introduced. The model resulted in a moving boundary problem (MBP) involving b the length density of certain protein filaments inside the cell. This MBP may be solved numerically by a level set method and we show by examples that under certain assumptions on the protrusion rate, solutions of the MBP converge to the traveling domain solution as time goes to infinity. Biologically, this means that under the assumptions of the model, all cells, regardless of their initial shapes and sizes, will eventually become identical and move in a particular direction at the same speed. Extensions of this model are also discussed.

1 Introduction

The movement of cells along surfaces is fundamental to many important biological processes and has been the subject of intense investigations for several decades. At the level of an entire cell, movement involves protrusion at the front of the cell, graded adhesion of the cell to a surface, followed by detachment and retraction of the cell's rear. This four-step process is generally accepted by biologists as how eukaryotic cells move [9]. Many detailed studies, including sophisticated mathematical and computational models, have examined each of these steps, especially protrusion [12]. However, as of this writing, a complete satisfactory model of the movement of an entire cell is still lacking. The introduction section of the paper [7] contains a brief summary of current mathematical models of cell motility.

The complexity of cell movement gives rise to mathematical models that are very difficult to analyze. Consequently, researchers have either looked at special cases of these models, such as by reducing the model to one-dimension by focusing on a narrow strip of the cell's lamellipodium in the direction of movement, or by solving the model equations numerically. The purpose of this paper is two-fold. First, we propose a simple two-dimensional continuum model of cell motility that can be applied to a wide class of eukaryotic cells. Second, we show that the resulting MBP can be solved efficiently by a level set method. We demonstrate this by showing that under certain assumptions on the protrusion rate, solutions of the MBP converge to a traveling domain solution as $t \rightarrow \infty$. The existence of traveling domain solution has recently been established by Choi and Lui [6].

The functioning of a cell at the molecular level is extremely complex and it is not possible to integrate all the relevant biochemical processes to develop a mathematical model of cell movement. Models of cell motility are rapidly evolving and most of these models have not been rigorously analyzed. We hope to develop a mathematical model that is longer lasting and that can be improved and build upon. Our model is motivated by the one-dimensional model of Mogilner and Verzi [10]. Substantial difficulties arise in two dimensions because now the cell has infinitely many directions to move. As of this writing, the only two-dimensional models we are aware of are those by Bottino and Fauci [1], Bottino, Mogilner, Roberts, Stewart and Oster [2], and Rubinstein, Jacobson and Mogilner [14]. Bottino and Fauci's model uses immersed boundary method to simulate amoeboid movement while Rubinstein et al.'s model uses finite element method to simulate the movement of a fish keratocyte.

¹Research partially supported by NSF grant no. DMS-0456570.

The organization of this paper is as follows. In section 2, we propose a simple two-dimensional model of cell motility. In section 3, we use a level set method to compute solutions of the MBP under certain assumptions on the protrusion rate. In section 4, we compute the traveling domain solution under the same assumptions on the protrusion rate and compare the results to the MBP. Section 5 summarizes our results and suggest possible extensions to our model.

2 A Simple Mathematical Model

Many crawling cells have length between 10 to 50 microns (μm) and an individual cell consists of a flat lamellipodium at the front which is fraction of a micron in thickness. The lamellipodium is composed of polarized actin filaments that are bundled and networked together and they are responsible for maintaining the cell's shape as well as its movement [3]. The actin filaments are polymers that undergo constant polymerization and depolymerization. Polymerization at the front pushes the cell forward. For most eukaryotic cells there are also motor proteins found mostly near the rear of the cell. These motor proteins walk on the actin filaments in one direction, causing them to contract and thus pulling the rear end of the cell forward. The relatively thick cell body which contains the nucleus and other organelles sits near the rear of the cell but it is not responsible for the cell's locomotion. The actin meshwork in the lamellipodium is called the cytoskeleton of the cell. In our model we ignore the cell body and model the lamellipodium as a closed region in \mathbf{R}^2 which we call the domain of the cell. We now propose our two-dimensional model.

The boundary of the cell is described by the zero level curve of a distance function $\psi(x, y, t)$ and we let $\psi < 0$ in the domain of the cell denoted by Ω . Inside Ω , the density of actin filaments b satisfies the mass balance equation

$$b_t + \nabla \cdot (b \mathbf{v}) = -\gamma b. \quad (2.1)$$

Here γ is the rate of depolymerization and \mathbf{v} is the cytoskeleton velocity. As the cell crawls forward, it breaks the adhesion to the surface and there is also resistance to the motion of the actin filaments by the cytoplasm of the cell. This force manifests itself as a drag which we model by

$$\xi(x, y)\mathbf{v} = \text{drag force on the cell}$$

where ξ is called the drag coefficient [2], [10], [7]. Using force balance [8], we have

$$\xi(x, y)\mathbf{v} = \nabla \cdot \sigma \quad (2.2)$$

where σ is the Cauchy stress tensor and is a 2×2 matrix. As a first model, we shall make the assumption that σ is isotropic meaning that it remains the same under rotation. Hence $U\sigma U^T = \sigma$ for any orthogonal matrix U . This implies that $\sigma = \theta I$ where I is the identity matrix. In other words, stress is only due to contraction ($\theta < 0$) or expansion ($\theta > 0$) and is the same in all directions. We further assume that $\theta = C_1(b_0 - b)$ where C_1, b_0 are given positive constants. This is to model the fact that actin filaments form bundles and these bundled filaments act like springs which cause the cell to contract [10]. We assume that the amount of contractile force resulting from the bundled filaments is proportional to $b - b_0$.

We now consider adhesion to the surface. From above, suppose drag is due only to adhesion and the cytoplasm is viscous. Then $\xi(x, y)$ should be an increasing function of the distance between (x, y) and the rear of the cell. This is because adhesion is known to be stronger at the front and weaker at the rear to facilitate the cell's movement forward. Since we are in two dimensions, it is not clear how to identify the front and rear of the cell. One way to do this is to identify the rear as those boundary points where \mathbf{v} is pointing into the cell; that is, the cell is retracting. However, doing so will complicate the model significantly because ξ will then be a function of \mathbf{v} which will make our model highly nonlinear. Since this is a first model, we assume that ξ is a constant. From (2.2), $\mathbf{v} = -C_1 \nabla b / \xi$. The amount of actin filaments on the boundary should be less at the rear than at the front but to keep our model simple, we assume that $b = b_0$ on the boundary. Under these assumptions, from (2.1), b satisfies the following equations

$$\begin{cases} b_t = (C_1/\xi)\Delta(b^2/2) - \gamma b & \text{in } \Omega \\ b = b_0 & \text{on } \partial\Omega \end{cases} \quad (2.3)$$

where $\partial\Omega$ denotes the boundary of Ω . In (2.3), the domain Ω moves with time so we need to write down the equation that governs its movement. It is well known [11] that ψ satisfies the Hamilton-Jacobi equation

$$\psi_t + V_n |\nabla\psi| = 0 \quad (2.4)$$

where V_n is the normal speed; i.e. the speed at the boundary of Ω in the direction of the outward normal \mathbf{n} . We should add the effect of polymerization at the cell boundary and hence $V_n = V_p + v$ where V_p is the rate of polymerization. From (2.3) and the maximum principle, $b \leq b_0$ in Ω . Since b is constant on the boundary, $\mathbf{n} = \nabla b / |\nabla b|$. Since $\mathbf{v} = -C_1 \nabla b / \xi$, we have $v = \mathbf{v} \cdot \mathbf{n} = -C_1 |\nabla b| / \xi$. Thus we arrive at the formula

$$V_n = V_p - C_1 |\nabla b| / \xi. \quad (2.5)$$

Our model, which is a MBP, consists of equations (2.3), (2.4), and (2.5).

Returning to our model, different cells will have different V_p and it controls the shape of the cell. Since cells more or less maintain their area as they move, V_p should depend inversely on the cell's area so that protrusion is slower for larger cells and faster for smaller cells. V_p may also include a stochastic term to account for the random changes in directions of some cells often observed in the laboratory. The MBP may be solved numerically by a variety of methods. In the next section, we illustrate how to do this using a level set method. This method is especially suitable if V_p depends on the local curvature of the boundary.

We now propose an example for V_p . For this example, mathematical analysis is possible which, as mentioned in the introduction, is one of the goals of this project. It has recently been proved that a traveling domain solution exists for this example. The meaning of this will be made clear in section 4. Global existence and uniqueness of solutions of this MBP is still an open problem.

The rate of polymerization V_p should be positive near the front of the cell and negative near the rear. We want to prescribe V_p so that the cell moves in the positive x direction. We introduce the normal angle ϕ which is defined as the angle between the vector $\mathbf{e}_1 = (1, 0)$ and \mathbf{n} at a boundary point. Let $V_p(\phi) = A + B \cos \phi$ where B is a positive constant. Then the maximum value of V_p occurs at $\phi = 0$ where $V_p(0) = A + B$ and the minimum value occurs at $\phi = \pi$ where $V_p(\pi) = A - B$. Therefore, $\phi = 0$ is the front and $\phi = \pi$ is the back. As mentioned above, V_p should be inversely proportional to the area of the cell. Hence we assume that $A + B = 2\beta/s_0^2$ and $A - B = -V_d$ where β and V_d are positive constants and $2s_0$ is the perimeter of the cell. We also include a negative curvature term in V_p to minimize pseudopods at the cell boundary. The formula for V_n is

$$V_n = \left(\frac{\beta}{s_0^2} - \frac{V_d}{2} \right) + \left(\frac{\beta}{s_0^2} + \frac{V_d}{2} \right) \cos \phi - C_2 \kappa - \frac{C_1 |\nabla b|}{\xi}. \quad (2.6)$$

Let $\bar{x} = \delta x, \bar{y} = \delta y, \tau = \gamma t$ and let $\bar{b}(\bar{x}, \tau) = b_1 b(x, t) / b_0$. If we choose $\delta = \sqrt{\xi \gamma b_1 / C_1 b_0}$, then \bar{b} satisfies (2.3) with $C_1 / \xi = 1, \gamma = 1$ and $\bar{b} = b_1$ on the scaled domain Ω' . In other words, we may assume that the coefficients in (2.3a) are all equal to 1 and b_0 can be any value we like. The same change of variables will result in (2.4) and (2.6) having the same form with different constants and $C_1 / \xi = 1$ in (2.6).

3 Time-dependent Problem

In this section, we shall present our numerical methods and results on computation of the MBP. From above, we assume that $C/\xi = 1$ and $\gamma = 1$ in (2.3).

Numerical Methods

A sufficiently large grid with grid spacing h is constructed which contains the initial domain Ω^0 and its anticipated position for the time period we want to solve the MBP. In the following, superscript and subscripts will denote discretization of time and space, respectively. Let the initial data $b_{i,j}^0$ and the initial curve Γ^0 , which represents the

boundary of the domain at $t = 0$, be given. We initialize $\psi_{i,j}^0$ by computing the distance of grid point (i, j) from Γ^0 at $t = 0$. We define $\psi_{i,j}^0$ to be negative inside the domain and positive outside. This is accomplished by computing the winding number of (i, j) with respect to Γ^0 to determine if it is inside or outside the domain. After that, we enter into a time loop $t_n, n = 1, \dots, N$.

Suppose $\psi_{i,j}^n$ and $b_{i,j}^n$ are known. We solve (2.3) inside Ω^n using the FTCS numerical scheme:

$$b_{i,j}^{n+1} = b_{i,j}^n + (\Delta t/2h^2) \{ (b_{i+1,j}^n)^2 + (b_{i-1,j}^n)^2 + (b_{i,j+1}^n)^2 + (b_{i,j-1}^n)^2 - 4(b_{i,j}^n)^2 \} - \gamma \Delta t b_{i,j}^n. \quad (3.1)$$

For some grid points (i, j) that are close to the boundary, the points $(i \pm 1, j \pm 1)$ may lie outside of Ω^n . The values of b at these ghostpoints are found by linear extrapolation using the boundary condition in (2.3) and $\psi_{i,j}^n$ and $\psi_{i \pm 1, j \pm 1}^n$. We then compute $|\nabla b|_{i,j}^n$ inside the domain using the following formula and the values of b at the ghostpoints if necessary

$$|\nabla b|_{i,j}^n = \sqrt{[(b_{i+1,j}^n - b_{i,j}^n)/h]^2 + [(b_{i,j+1}^n - b_{i,j}^n)/h]^2}.$$

The next step is to compute the normal vector $\mathbf{n} = (n_x, n_y)$ at time $t = n$ for each grid point using the formula $\mathbf{n} = \nabla \psi / |\nabla \psi|$. After that, we compute the normal angle $\phi_{i,j}^n \in [-\pi, \pi]$ by the formula $\phi = \arctan(n_y/n_x)$. The curvature $\kappa_{i,j}^n$ can then be found using the formula $\kappa = \nabla \cdot \mathbf{n}$. To compute the perimeter and area of the domain at time $t = n$, we use the divergence theorem $\int_{\Omega} \nabla \cdot \mathbf{F} dV = \int_{\partial\Omega} \mathbf{F} \cdot \mathbf{n} dS$. Letting $\mathbf{F} = \mathbf{n}$, the perimeter of Ω^n , P^n , equals $\int_{\partial\Omega} dS = \int_{\Omega} \nabla \cdot \mathbf{n} dV$. Area A^n is approximated in the following way: For each cell we let $m_{i,j} \in \{0, 1, 2, 3, 4\}$ be the number of vertices of this cell that lie inside the domain. Then $A^n = \sum_{i,j} h^2 m_{i,j} / 4$. We also compute the centroid of the domain (x_c, y_c) by $x_c = \int_{\Omega} x \psi dA / \int_{\Omega} \psi dA$ and $y_c = \int_{\Omega} y \psi dA / \int_{\Omega} \psi dA$. The normal speed (2.6) is computed by

$$V_{i,j}^n = (\beta/s_0^2 - V_d/2) + (\beta/s_0^2 + V_d/2) \cos \phi_{i,j}^n - \kappa_{i,j}^n - |\nabla b|_{i,j}^n$$

where $s_0 = P^n/2$. From this we find $\mathbf{U} = (U_1, U_2)$ where $(U_1)_{i,j}^n = V_{i,j}^n \cos \phi_{i,j}^n$ and $(U_2)_{i,j}^n = V_{i,j}^n \sin \phi_{i,j}^n$. The domain Ω^n is advected to the next time step by solving equation (2.4). Since $\nabla \psi / |\nabla \psi| = \mathbf{n} = (\cos \phi, \sin \phi)$, we can rewrite (2.4) as $\psi_t + (U_1, U_2) \cdot \nabla \psi = 0$ where (U_1, U_2) is a known function at $t = n$. This equation is solved by a second order accurate in space ENO method described in [15], [13]. For completeness, we summarize the method below.

The above equation for ψ is $\psi_t = -(U_1, U_2) \cdot \nabla \psi \equiv L\psi$. Let

$$\psi_{i,j}^* = \psi_{i,j}^n + \Delta t L\psi^n.$$

Then find ψ^{n+1} by

$$\psi_{i,j}^{n+1} = \psi_{i,j}^n + \frac{\Delta t}{2} (L\psi^n + L\psi^*)$$

where

$$L\psi = -(U_1)_{i,j} \frac{\psi_{i+1/2,j} - \psi_{i-1/2,j}}{h} - (U_2)_{i,j} \frac{\psi_{i,j+1/2} - \psi_{i,j-1/2}}{h}.$$

Here

$$\psi_{i+1/2,j} = \begin{cases} \psi_{i,j} + \frac{1}{2} M(D_x^+ \psi_{i,j}, D_x^- \psi_{i,j}) & \text{if } \frac{1}{2}(u_{i+1,j}, u_{i,j}) > 0 \\ \psi_{i+1,j} - \frac{1}{2} M(D_x^+ \psi_{i+1,j}, D_x^- \psi_{i+1,j}) & \text{if } \frac{1}{2}(u_{i+1,j}, u_{i,j}) < 0, \end{cases}$$

$$M(a, b) = \begin{cases} a & \text{if } |a| < |b| \\ b & \text{if } |a| \geq |b| \end{cases} \quad (3.2)$$

and

$$D_x^+ \psi_{i,j} = \psi_{i+1,j} - \psi_{i,j}, \quad D_x^- \psi_{i,j} = \psi_{i,j} - \psi_{i-1,j}. \quad (3.3)$$

The equation for $\psi_{i,j+1/2}$ is defined similarly. Using this method and (3.1), we obtain $\psi_{i,j}^{n+1}$ and $b_{i,j}^{n+1}$, completing the computation for one time step. The velocity of the domain is approximated by $(x_c^{n+100} - x_c^n)/(100 \Delta t)$ and

$(y_c^{n+100} - y_c^n)/(100 \Delta t)$. This, as well as the perimeter and area of the domain, will be compared to that of the traveling domain solution described in the next section.

Two issues remain. One is the choice of Δt and the second is that as time progresses, ψ will lose its accuracy as a distance function so we need to re-initialize it back to a distance function. The time step Δt is chosen to meet the stability criteria for equations (2.3) and (2.4). Hence Δt must satisfy $\Delta t \leq \min\{h^2/(4b_0), h/U_m\}$ where $U_m = \max\{|U_1|, |U_2|\}$. Since the first term in the minimum is $O(h^2)$, we set $\Delta t = h^2/(4b_0)$. For the second issue of restoring ψ back to a distance function, we follow the method in [16] and chapter 3.5 of [13] by solving the initial value problem

$$\bar{\psi}_\tau = L\bar{\psi} \equiv \text{sgn}(\bar{\psi})(1 - |\nabla\bar{\psi}|), \quad \bar{\psi}(x, 0) = \psi(x) \quad (3.4)$$

in a band with width 2α containing boundary of the domain at its center. This is done every J number of time steps. In (3.4), τ is an artificial time and $\text{sgn}(\psi)$ is a smooth function that increases from -1 inside the domain to 1 outside the domain across the band. Equation (3.4) is solved iteratively using the same method we used to solve (2.4). Details may be found in [16] and [13].

Numerical Results

We apply the above method to compute the MBP (2.3), (2.4) and (2.6) with $C_1/\xi = 1, \gamma = 1, b_0 = 20, \beta = 5, V_d = 2\beta/\pi^2 = 1.01, C_2 = 1$ and $b = b_0$ initially. A 180×90 grid was used over an area of 6×3 square units with $h = 0.03333$. Also, $\Delta t = h^2/(5b_0) = 1.1109 \times 10^{-5}$ and we ran our program 50000 iterations, reinitializing the distance function every $J = 100$ iterations with $\alpha = 50$. The four initial domains we tested are:

1. Circle centered at the origin with radius 0.5.
2. Ellipse centered at the origin with $a = 0.4$ and $b = 0.7$.
3. Peanut shaped domain whose parametric equation is given by

$$(x(u), y(u)) = (1.5 + \sqrt{0.3 + 0.25 \cos(2u)} \cos(u), 1.5 + \sqrt{0.3 + 0.25 \sin(2u)} \sin(u)).$$

4. Petal whose parametric equation is given by

$$(x(u), y(u)) = (1.5 + \sqrt{0.13 \times 3.5 + 0.15 \cos(8u)} \cos(u), 1.5 + \sqrt{0.13 \times 3.5 + 0.15 \sin(8u)} \sin(u)).$$

Figure 1 shows the areas and perimeters of the moving domain for 50000 iterations. Figure 2 shows the velocity of the domain in the x -direction. Figure 3 shows the positions of the four domains at various iterations. Table 1 summarizes the values of these properties after 50000 iterations.

Initial Domain	Area	Perimeter	Velocity in x -direction	Velocity in y -direction
Circle	0.5058	2.5194	3.6637	-0.0021
Ellipse	0.5056	2.5190	3.6650	-0.000915
Peanut	0.5105	2.5310	3.6438	0.0078
Petal	0.5059	2.5198	3.6662	-0.0026

Table 1: Domain properties after 50000 iterations.

Interpretation of Results

Using the numbers in (1), we estimate that $\gamma = 0.175/s, b_0 = 500 \mu m/\mu m^2$ and the function ξ has a mean value of $70 pN \cdot s/\mu m^2$. Note that C_1 has unit of $pN/\mu m$ and σ has unit of $pN/\mu m^2$. Simulation of the one-dimensional model reveals that b_0 and b are close to each other so if we estimate that $\sigma = C_1(b_0 - b)$ has the same order of magnitude as ξ , we can set $C_1 = 100 pN/\mu m$. From the last paragraph of section 2, we have $\delta = \sqrt{70 \times 0.175 \times 20/100 \times 500} = 0.07$. From Table 1, area is approximately 0.5 square unit. Since $\bar{x} = 0.07x, \bar{y} = 0.07y$, assuming that the domain in our computation is a circle, the diameter of the domain is estimated to be $2\sqrt{0.5/\pi} \approx 0.8$. Therefore, the estimated size of the traveling cell is about $0.8/0.07 = 11.4 \mu m$. The size of an actual cell varies greatly but many crawling cells are between 10 to 50 microns in length. For speed, recall that $\tau = \gamma t = 0.175t$. In our calculation, velocity in the x -direction was found to be around $\Delta\bar{x}/\Delta\tau = 3.66$ so we estimate that the cell is moving in x -direction at the velocity $3.6 \times (0.175/0.07) = 9.15 \mu m/s$ which is about 80% of its body size per second.

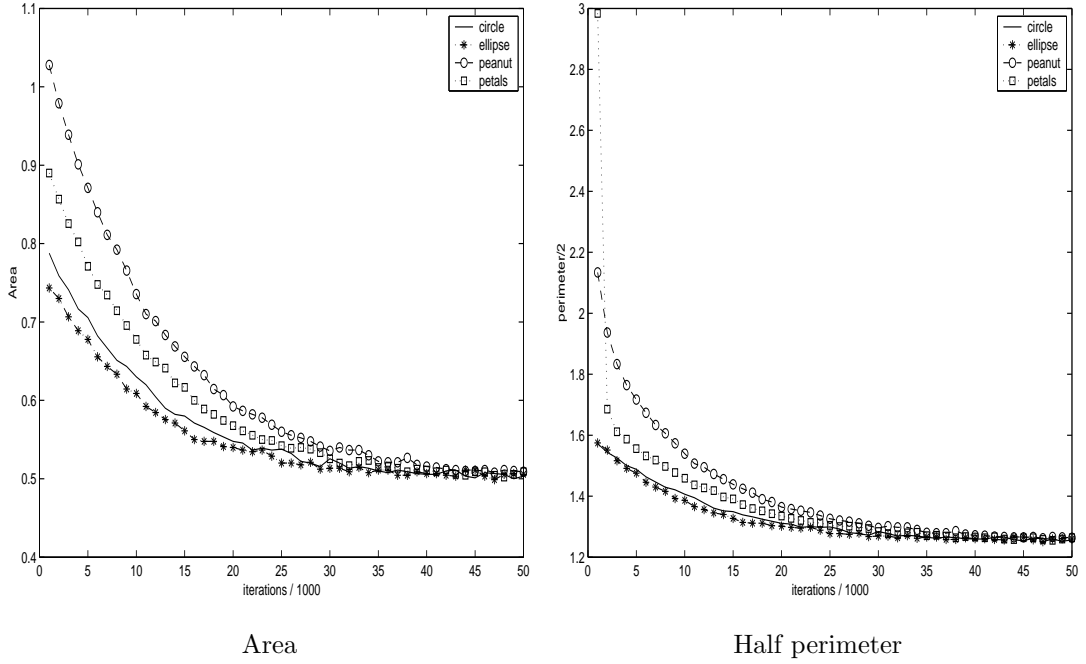


Figure 1: Area and half perimeter of domain during 50000 time iterations.

4 Traveling Domain Solution And Its Computation

Mathematical Theory

Traveling domain solution of our MVP is defined as a pair of functions \tilde{b} and $\tilde{\psi}$ such that $b(x, y, t) = \tilde{b}(x - kt, y)$ and $\psi(x, y, t) = \tilde{\psi}(x - kt, y)$ satisfy equations (2.3), (2.4) and (2.6). Here, k is the domain speed and is an unknown constant that needs to be determined together with \tilde{b} and $\tilde{\psi}$. Carrying out the differentiation, \tilde{b} satisfies the equation $\Delta \tilde{b}^2 + k \tilde{b}_x - \gamma \tilde{b} = 0$ in Ω , $\tilde{b} = b_0$ on the boundary of Ω which is given by the zero level curve of the distance function $\tilde{\psi}$. From (2.4), $\tilde{\psi}$ satisfies the equation $-k \tilde{\psi}_x + V_n |\nabla \tilde{\psi}| = 0$. The unit outward normal $\mathbf{n} = (\cos \phi, \sin \phi) = \nabla \tilde{\psi} / |\nabla \tilde{\psi}|$. Hence finding $\tilde{\psi}$ is equivalent to finding a domain Ω that satisfies the relation $k \cos \phi = V_n$ on $\partial\Omega$. The existence of traveling domain solution for this particular choice of V_p has recently been established by Choi and Lui [6] and is summarized in the following theorem. For the existence of traveling cell solutions for the Mogilner-Verzi model, see [4].

Theorem 4.1 *Let V_d, β, b_0 and γ be positive constants. Then there exists a smooth convex domain $\Omega \subset \mathbf{R}^2$ that is symmetric about the x -axis and centered at the origin, a smooth positive function \tilde{b} and a constant $k > 0$ such that*

$$\begin{cases} \Delta \tilde{b}^2 + k \tilde{b}_x - \gamma \tilde{b} = 0 & \text{in } \Omega \\ \tilde{b} = b_0 & \text{on } \partial\Omega \\ k \cos \phi = V_p(\phi) - |\nabla \tilde{b}|^*(\phi) & \text{on } \partial\Omega. \end{cases} \quad (4.1)$$

In the above statement, smooth means twice continuously differentiable,

$$V_p(\phi) = (\beta/s_0^2 - V_d/2) + (\beta/s_0^2 + V_d/2) \cos \phi - \kappa, \quad (4.2)$$

$2s_0$ is the length of $\partial\Omega$, and $|\nabla \tilde{b}|^*(\phi)$ is a continuously differentiable function defined in $[0, \pi]$ such that $0 \leq |\nabla \tilde{b}|^*(\phi) \leq |\nabla \tilde{b}|(\phi)$ in $[0, \pi]$.

Note that even though \tilde{b} is smooth in the x, y variables, $|\nabla \tilde{b}|$ is not smooth in the ϕ variable.

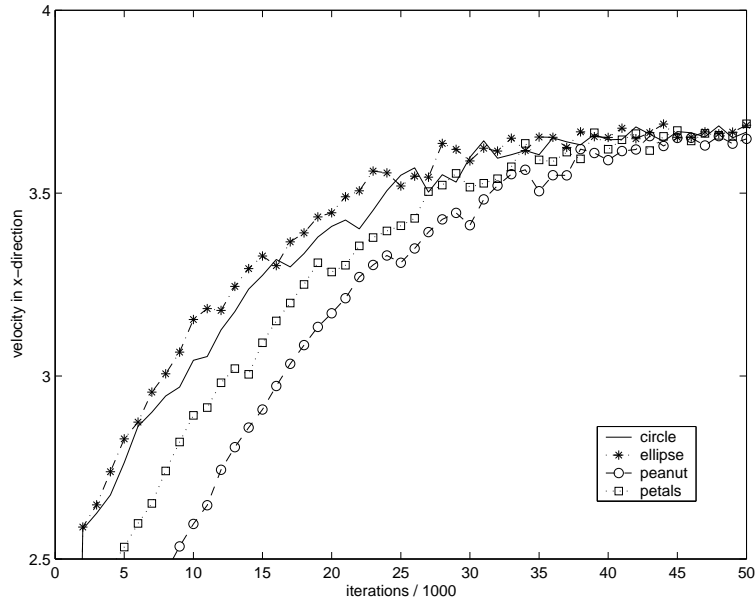


Figure 2: Velocity in the x-direction.

Theorem 4.1 says that the cell motility model we proposed in section 2 with this specific choice of protrusion rate has a special solution which is a cell that maintains its shape and moves in the positive x -direction at a constant speed k . The only catch is that we have to replace $|\nabla b|(\phi)$ by $|\nabla \tilde{b}|^*(\phi)$ in the definition of V_n (compare last equation in (4.1) with (2.6)). This is due to a technical difficulty involved in the proof of the theorem; the mapping from arc-length s to normal angle ϕ is not always one-to-one. For example, if one part of the cell boundary is a line, then s increases but ϕ remains constant along the line.

The construction of the traveling domain solution $(\tilde{b}, \tilde{\psi}, k)$ is by an iterative process which finds the fixed point of a map $T : C^+[0, \pi] \rightarrow C^+[0, \pi]$ where $C^+[0, \pi]$ is the set of all nonnegative continuous functions defined on the interval $[0, \pi]$. Let s denote arc-length measured counterclockwise from the right most point of the domain Ω . Then $\kappa = d\phi/ds$. Let $g \in C^+[0, \pi]$ which represents a guess of $|\nabla \tilde{b}|^*(\phi)$ in (4.1) be given. Consider the boundary value problem

$$\begin{cases} \frac{d\phi}{ds} = (\beta/s_0^2 - V_d/2) + (\beta/s_0^2 + V_d/2 - k) \cos \phi - g(\phi) \\ \frac{dy}{ds} = \cos \phi \\ \phi(0) = 0, \quad \phi(s_0) = \pi, \quad y(0) = 0, \quad y(s_0) = 0. \end{cases} \quad (4.3)$$

The first equation of (4.3) is the last equation of (4.1) with $|\nabla b|^*$ replaced by g . If $(x(s), y(s))$ denotes the parametrization of the upper half boundary of the domain Ω by arc-length, then $dx/ds = -\sin \phi(s)$. The boundary conditions in (4.3) come from the requirement that Ω is symmetric about the x -axis. The first two equations of (4.3) are solved with initial conditions $\phi(0) = 0$, $y(0) = 0$ and s_0 and k are chosen to satisfy the conditions $\phi(s_0) = \pi$ and $y(s_0) = 0$. The domain Ω is then recovered from $(x(s), y(s))$, $0 \leq s \leq s_0$ where x is obtained by integrating $dx/ds = -\sin \phi(s)$ with initial condition $x(0) = \frac{1}{2} \int_0^{s_0} \sin \phi(\xi) d\xi$ to ensure that Ω is centered at the origin. With Ω and k known, the boundary value problem from the first two equations of (4.1) is solved inside this domain and $|\nabla b|(\phi)$ is computed to be our new g for the next iteration. We define $T(g) = |\nabla b| \in C^+[0, \pi]$ whose fixed point is our traveling domain solution.

Numerical Methods

Our numerical algorithm is as follows. The input to the first step of our iteration is a smooth function g defined on $[0, \pi]$. Equation (4.3) is solved as described in the above paragraph to obtain $s_0, k, \mathbf{x}, \mathbf{y}$ and ϕ . Here, (x_i, y_i) , $i = 1, \dots, 2n + 1$ are $2n + 1$ equally spaced points (with respect to arc-length) on the boundary and ϕ_i is the corresponding normal angle which is ϕ in (4.3). Step 2 begins by placing the domain in a square whose side is slightly larger than the maximum diameter of the domain. An $m \times m$ grid is placed on the square. The distance function ψ at each grid point is then found by measuring the distance to the closest boundary point (x_i, y_i) and the sign determined by computing the grid point's winding number. The boundary value problem (4.1) is then solved using finite difference method with k in (4.1) obtained from step 1. Since the equation is nonlinear, iterative method has to be employed to find \tilde{b} inside the domain. The outward normal vector $\mathbf{n} = (n_x, n_y)$ at each grid point is then approximated by $\nabla\psi/|\nabla\psi|$ and the normal angle is calculated by $\phi = \arctan(n_y/n_x)$. The output of the second step are ψ, \tilde{b} and ϕ at the grid points. Step 3 of the iteration involves finding $|\nabla\tilde{b}|$ on the boundary. Since the domain is now approximated by a polygon with vertices at $(x_i, y_i), i = 1, \dots, 2n + 1$, $|\nabla\tilde{b}|$ is singular at the vertices. We therefore find $|\nabla\tilde{b}|$ at the midpoint between two adjacent vertices of the polygon. To do that, let the midpoints be denoted by $P_i, i = 1, \dots, 2n$. At each P_i , we find the point Q_i inside the domain a distance h (grid size) from P_i in the direction opposite to \mathbf{n} . The value of \tilde{b} at Q_i is then approximated by interpolation using the values of \tilde{b} at the four corners of the cell where Q_i lies. Let the value be denoted by \tilde{b}_i . Then the value of $|\nabla\tilde{b}|$ at P_i is approximated by $(b_0 - \tilde{b}_i)/h$. The normal angle at P_i is set to be $(\phi_i + \phi_{i+1})/2$. However, if we plot the approximated values of $|\nabla\tilde{b}|$ at P_i for $i = 1, \dots, 2n$, the resulting graph is highly oscillatory and non-smooth. We therefore approximate this by a smooth function $\tilde{g}(\phi)$ which when properly translated becomes the input to our next iteration.

Numerical Results

For our actual calculations, we let $\beta = 5$, $V_d = 2\beta/\pi^2$, $\gamma = 1$, $b_0 = 20$, $n = 2000$ and $m = 210$. We let $g(\phi) = 1 + \phi/\pi$ be the input to our first step as discussed in the subsection *Numerical Methods* above. In step 3 of our iteration, we approximated $|\nabla\tilde{b}|$ at the boundary by a twelve-degree polynomial $p(\phi)$ using Matlab function *polyfit*. From the paragraph after Theorem 4.1, $\tilde{g}(\phi)$, which is the input to our next iteration, should be a smooth function below $|\nabla\tilde{b}|$. Since we cannot find the minimum of $|\nabla\tilde{b}|$ accurately, we let

$$\tilde{g}(\phi) = p(\phi) - \frac{(adjust - 1)}{10}M, \quad 0 \leq \phi \leq \pi \quad (4.4)$$

where $adjust = 1, \dots, 10$ and M is the minimum value of $|\nabla\tilde{b}|$ at the P_i 's. We ran our program on Matlab for 4 iterations for $adjust = 1, \dots, 10$ and estimated that when $adjust = 5.5$, our results best match the results of the time-dependent problem in terms of s_0, k and $area$. We then ran our program for 15 iterations with $adjust = 5.5$ and obtained $s_0 = 1.2613$, $k = 3.6696$ and $area = 0.5068$. These numbers agree reasonable well with those in Table 1. The contour and surface of \tilde{b} are given in Figures 4 and 5, respectively. In Figure 5, the surface of \tilde{b} is plotted on a coarser 70×70 grid to improve visibility.

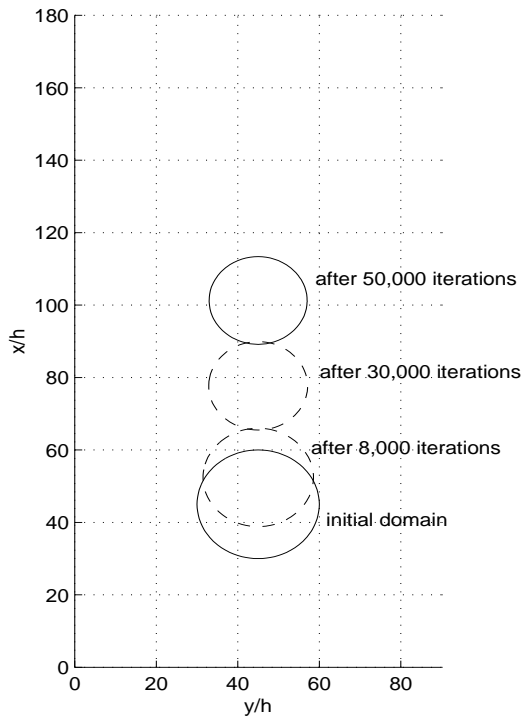
5 Conclusion

In this paper we proposed a two-dimensional cell motility model based on the general principles of how cells move. The model resulted in a MBP which may be solved by a variety of numerical methods depending on the protrusion rate. We then proposed a specific protrusion rate and used a level set method to show how to compute solutions of the MBP with four different types of initial cell shapes. The results appeared to converge to a traveling domain solution which we also computed. The current model may be improved in several ways: (i) include motor proteins in the model to model retraction at the rear, (ii) allow β and γ to depend on x , (iii) model adhesion by considering how cells actually attach to the surface. Some of these issues have already been discussed in section 2. Eventually, one would like to study chemotaxis and go beyond a single cell to study collective behavior.

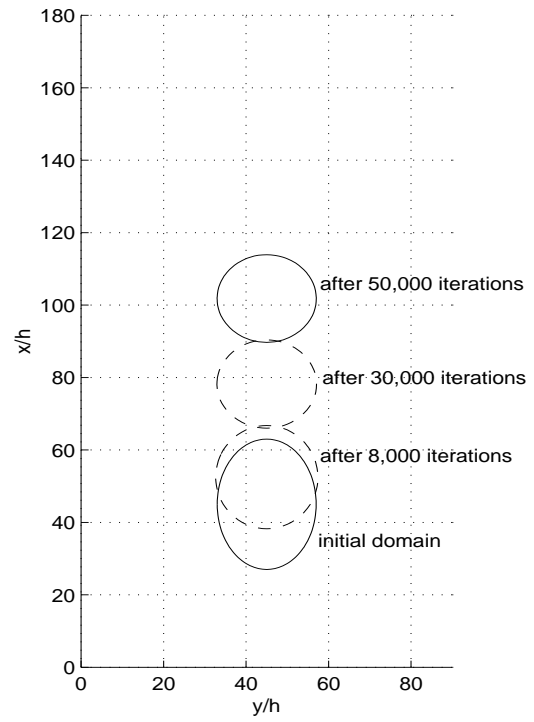
There is a revolution going on in biology because recent advances in laboratory techniques allow many things to be observed and forces to be measured so that biology is now becoming more and more quantitative. Many of the models we propose now may be oversimplified but bringing all the tools in physical and mathematical sciences to help understand cellular and molecular biology is certainly very exciting.

References

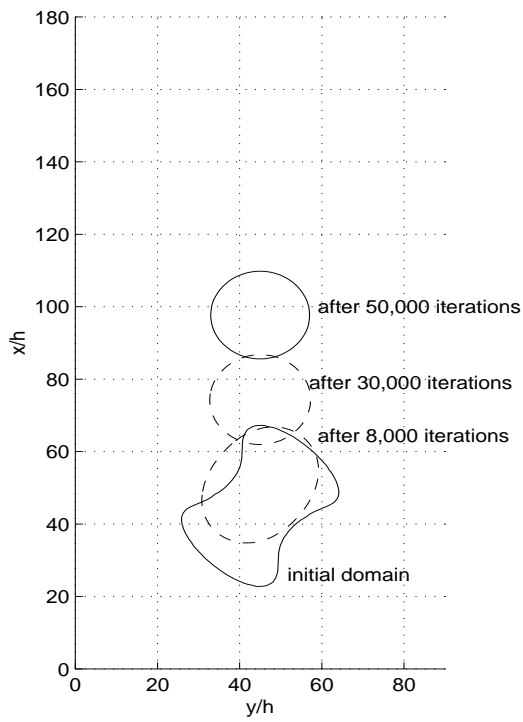
- [1] Bottino, D. and L. J. Fauci. 1998. A computational model of amoeboid deformation and locomotion. *Euro. Biophys. J.* 27:532-539.
- [2] Bottino, D., A. Mogilner, T. Roberts, M. Stewart and G. Oster. 2002. How nematode sperm crawl. *J. Cell Sci.* 115:431-453.
- [3] Bray, Dennis. 2001. *Cell Movements: From Molecules to Motility*, 2nd edition. Garland.
- [4] Choi, Y. S., J. Lee and R. Lui. 2004. Traveling wave solutions for a one-dimensional crawling nematode sperm cell model. *J. Math. Bio.* 49:310-328.
- [5] Choi, Y. S., P. Groulx and R. Lui. 2005. Moving boundary problem for a one-dimensional crawling nematode sperm cell model. *Nonlinear Analysis: Real World Applications.* 6:874-898.
- [6] Choi, Y. S. and R. Lui. 2006. Existence of traveling domain solutions for a two-dimensional moving boundary problem. (submitted)
- [7] Gracheva, M. and H. Othmer. 2004. A continuum model of motility in amoeboid cells. *Bull. Math. Bio.* 66:167-193.
- [8] Landau, L. D. and E. M. Lifshitz. 2002. *Theory of Elasticity*, 3rd edition. Springer.
- [9] Mitchison, T. J. and L. P. Cramer. 1996. Actin-based cell motility and cell locomotion. *Cell.* 84:371-379.
- [10] Mogilner, A. and D. W. Verzi. 2003. A simple 1-d physical model for the crawling nematode sperm cell. *J. Stat. Phys.* 110:1169-1189.
- [11] Osher, S. and R. Fedkiw. 2002. *Level Set Methods and Dynamic Implicit Interfaces*. Springer, New York.
- [12] Pollard D. T. and G. Borisy. 2003. Cell motility driven by assembly and disassembly of actin filaments. *Cell.* 112:453-465.
- [13] Prosperetti, A. and G. Tryggvason (editors). 2006. *Computational Methods for Multiphase Flow*. Cambridge University Press.
- [14] Rubinstein, B., K. Jacsonson and A. Mogilner. 2005. Multiscale two-dimensional modeling of a motile simple-shaped cell. *SIAM J. MMS*, 3:413-439.
- [15] Sussman, M., E. Fatemi, P. Smereka, and S. Osher. 1998. An improved level set method for incompressible two-phase flows. *Computers and Fluids.* 27:663-680.
- [16] Sussman, M., P. Semerka, and S. Osher. 1994. A level set method for computing solutions to incompressible two-phase flow. *J. Comp. Physics*, 114:146-159.



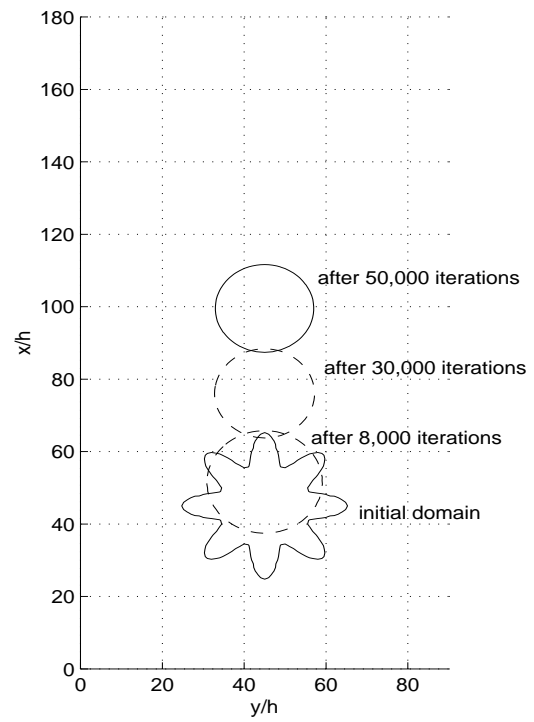
Initial domain a circle.



Initial domain an ellipse.



Initial domain peanut shape.



Initial domain a petal.

Figure 3: Positions of domain during 50000 time iterations.

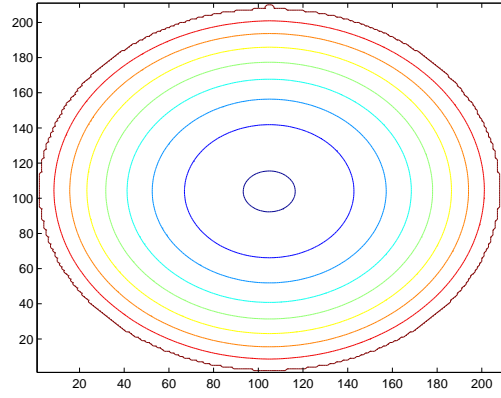


Figure 4: Contour of traveling domain solution \tilde{b} .

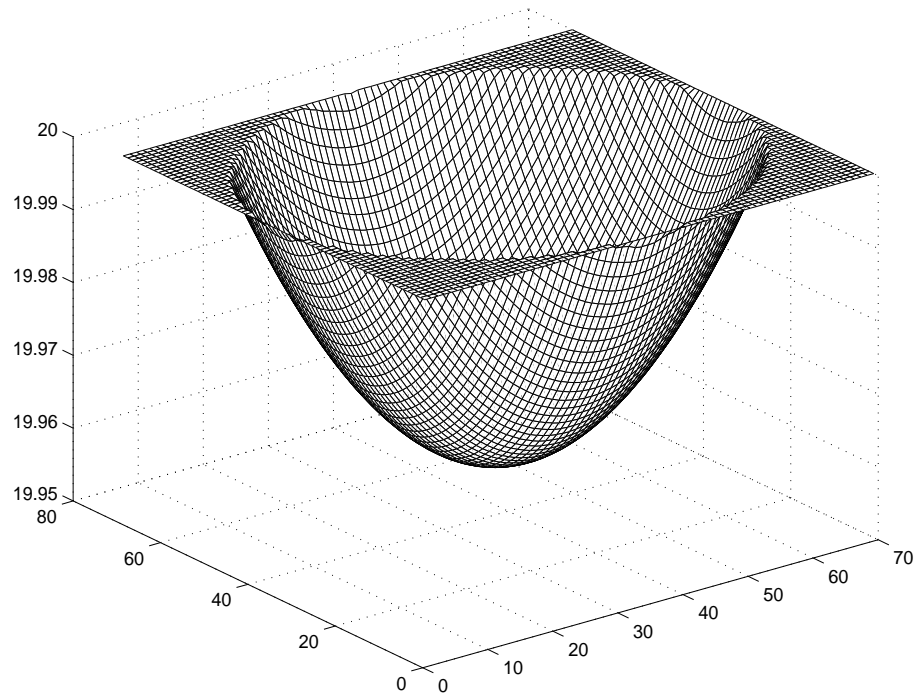


Figure 5: Traveling domain solution \tilde{b} .