Published in final edited form as:
Comm App Math Comp Sci. 2010 ; 5(1): 31–53.

A Cut Cell Method for Simulating Spatial Models of Biochemical Reaction Networks in Arbitrary Geometries

Wanda Strychalski[a], **David Adalsteinsson**[a], and **Timothy C. Elston**[b,*]

[a]Carolina Center for Interdisciplinary Applied Mathematics, Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC, 27599, USA

[b]Department of Pharmacology, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA

Abstract

Cells use signaling networks consisting of multiple interacting proteins to respond to changes in their environment. In many situations, such as chemotaxis, spatial and temporal information must be transmitted through the network. Recent computational studies have emphasized the importance of cellular geometry in signal transduction, but have been limited in their ability to accurately represent complex cell morphologies. We present a finite volume method that addresses this problem. Our method uses Cartesian cut cells and is second order in space and time. We use our method to simulate several models of signaling systems in realistic cell morphologies obtained from live cell images and examine the effects of geometry on signal transduction.

Keywords

Systems biology; Numerical methods; Reaction-diffusion equation

1 Introduction

Cells must be able to sense and respond to external environmental cues. Information about external signals, such as hormones or growth factors, is transmitted by signaling pathways to the cellular machinery required to generate the appropriate response. Defects in these pathways can lead to diseases, such as cancer, diabetes, and heart disease. Therefore, understanding how intracellular signaling pathways function is not only a fundamental problem in cell biology, but also important for developing therapeutic strategies for treating disease.

In many pathways, proper signal transduction requires that both the spatial and temporal dynamics of the system are tightly regulated [1]. For example, recent experiments have revealed spatial gradients of protein activation in migrating cells [2]. Mathematical models can be used to elucidate the control mechanisms used to regulate the spatiotemporal dynamics of signaling pathways, and recent computational studies emphasize the importance of cellular geometry in signaling networks [3,4]. For computational simplicity, many of these investigations assume idealized cell geometries [4,5], whereas others approximate irregularly shaped cells using a “staircase” representation of the cell membrane [6]. Both finite element and finite volume methods have been used to simulate spatial models of biochemical reaction networks [4,7]. For finite element methods, which typically require a triangulation of the computational domain, grid generation can be a challenge. This becomes especially true if the

[*]Corresponding author. elston@med.unc.edu (Timothy C. Elston).

boundaries of the computational domain are moving. To overcome this issue, we developed a finite volume method that utilizes a Cartesian grid. Our numerical scheme is based on a cut cell method that accurately represents the cell boundary using a piecewise-linear approximation. This approach allows us to achieve second order accuracy in space and time. Our method also provides an appropriate framework for addressing moving boundary problems using level set methods [8,9].

2 Mathematical Formulation

Spatial models of biochemical reaction networks are typically represented using partial differential equations consisting of reaction and diffusion terms. For simplicity we restrict ourselves to two spatial dimensions x and y. For a given chemical species, the reaction terms encompass processes such as activation, degradation, protein modifications and the formation of molecular complexes. These reactions typically include nonlinear terms, such as those arising from Michaelis-Menten kinetics. In a system consisting of n chemical species, the concentration of the i^{th} species c_i evolves in space and time according to the following equation:

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot \mathbf{J} + f_i(\boldsymbol{c}). \quad (1)$$

where $\mathbf{J} = -D_i \nabla c_i$ is the flux density, D_i is the diffusion coefficient, and the function $f_i(\boldsymbol{c})$ models the reactions within the cell that affect c_i. The elements of the vector $\boldsymbol{c}$ are the concentrations of the n chemical species. Reactions also may occur on the cell membrane yielding mixed-type, nonlinear conditions on the boundary $\partial\Omega$,

$$-D\overrightarrow{n} \cdot \nabla c_i|_{\partial\Omega} + g(\boldsymbol{c})|_{\partial\Omega} = 0. \quad (2)$$

Eqs. (1) and (2) are solved subject to appropriate initial conditions $c_i(x, y, 0)$ for each species in the system.

3 Numerical Methods

Our goal is to develop a simulation tool that can accurately and efficiently solve spatial models of signaling and regulatory pathways in realistic cellular geometries. We obtain the computational domain from live-cell images. The model equations are solved on a Cartesian grid by discretizing the Laplacian operator, which models molecular diffusion, using a finite volume method.

3.1 Computational domain

Fig. 1 shows a grayscale image of a mouse fibroblast [2]. Because the original image is noisy, the image was smoothed by convolving it twice with the standard five point Gaussian smoothing filter. After smoothing, a suitable thresholding value was picked, and the front was computed by an iso-contour finder. A signed distance function is constructed with the smoothened boundary using Fast Marching Methods [10]. The zero level set of the signed distance function yields piecewise linear segments used to define cut cells (Fig. 2).

3.2 Discretization of the Spatial Operator

We utilize a Cartesian grid-based, finite volume algorithm originally presented in [11] to discretize the diffusion operator arising from Eq. (1). Finite volume methods store the average value of the concentration over a computational grid cell at the location (i, j). That is,

$$\bar{c}_{i,j}=\frac{1}{V_{i,j}}\int\int_{V_{i,j}} c(x,y)dV, \quad (3)$$

where $V_{i,j}$ is the volume of the (i, j) grid cell. Inserting Eq. (3) into Eq. (1) produces

$$\frac{\partial \bar{c}_{i,j}}{\partial t}-\overline{f(c)}_{i,j}=-\frac{1}{V_{i,j}}\int\int_{V_{i,j}} \nabla\cdot\mathbf{J}dV. \quad (4)$$

The divergence theorem allows us to convert the above volume integral into a surface integral,

$$\frac{\partial \bar{c}_{i,j}}{\partial t}-\overline{f(c)}_{i,j}=-\frac{1}{V_{i,j}}\int_{\partial V_{i,j}} (\mathbf{J}\cdot\overrightarrow{n})dS. \quad (5)$$

For interior grid cells, we have

$$\frac{\partial \bar{c}_{i,j}}{\partial t}-\overline{f(c)}_{i,j}=-\frac{1}{V_{i,j}}\left[\int_{y_{j-1/2}}^{y_{j+1/2}}\left(J_x(x_{i+1/2},y)-J_x(x_{i-1/2},y)\right)dy+\int_{x_{i-1/2}}^{x_{i+1/2}}\left(J_y(x,y_{j+1/2})-J_y(x,y_{j-1/2})\right)dx\right], \quad (6)$$

where $J_x=-D\frac{\partial c}{\partial x}$ and $J_y=-D\frac{\partial c}{\partial y}$. Approximation of the integrals in Eq. (6) with the midpoint rule yields

$$\frac{\partial c_{i,j}}{\partial t}-f(c_{i,j})\approx-\frac{1}{V_{i,j}}\left[\Delta y\left(J_x(x_{i+1/2},y_j)-J_x(x_{i-1/2},y_j)\right)+\Delta x\left(J_y(x_i,y_{j+1/2})-J_y(x_i,y_{j-1/2})\right)\right]. \quad (7)$$

By approximating the gradient terms with centered differences, we arrive at the standard five point Laplacian.

The cut cell method generalizes as follows. The boundary of the computational domain is approximated as a piecewise linear segments (Fig. 2, dashed line), and grid cells that the boundary passes through are referred to as cut cells. The volume of a cut cell is computed by recasting the volume integral as a boundary integral:

$$V_{i,j}=\int\int_{V_{i,j}} dV=\int\int_{V_{i,j}} \nabla\cdot\left[\frac{x}{2},\frac{y}{2}\right]^T dV=\int_{\partial V_{i,j}}\left(\left[\frac{x}{2},\frac{y}{2}\right]\overrightarrow{n}\right)dS, \quad (8)$$

where the integral on the right can be computed exactly for the polygon. Each segment is evaluated, then summed. The center of mass can also be computed using a boundary integral, for example:

$$\int\int_{V_{ij}} xdV=\int\int_{V_{ij}} \nabla\cdot\left[\frac{x^2}{2},0\right]^T dV=\int_{\partial V_{i,j}}\left(\left[\frac{x^2}{2},0\right]\overrightarrow{n}\right)dS. \quad (9)$$

Next, we construct the integral on the right side of Eq. (5) for a cut cell. In general, there are up to five surface integrals to approximate. Let $a_{l,m} \in [0, 1]$ represent the portion of each of the four cell edges covered by the cut cell and a_f be the length of the line segment representing the boundary. Then Eq. (7) becomes

$$\frac{\partial c_{i,j}}{\partial t} - f(c_{i,j}) \approx -\frac{1}{V_{i,j}}\Big[\Delta y\Big(a_{i+1/2,j}J_x(x_{i+1/2},j) - a_{i-1/2,j}J_x(x_{i-1/2},y_j)\Big) + \Delta x(a_{i,j+1/2}J_y(x_i,y_{j+1/2}) - a_{i,j-1/2}J_y(x_i,y_{j-1/2})\Big) + a^f J_f. \quad (10)$$

Let $F_{i\pm1/2,j} = -a_{i\pm1/2,j}\Delta y J_x(x_{i\pm1/2}, y_j)$ and $F_{i,j\pm1/2} = -a_{i,j\pm1/2}\Delta x J_y(x_i, y_{j\pm1/2})$. With this notation, we rewrite the previous equation as

$$\frac{\partial c_{i,j}}{\partial t} - f(c_{i,j}) \approx \frac{1}{V_{i,j}}(F_{i+1/2,j} - F_{i-1/2,j} + F_{i,j+1/2} - F_{i,j-1/2} - F^f_{i,j}). \quad (11)$$

We refer to the F's as the surface fluxes (Fig. 3). On a full edge with $a_{l,m} = 1$ the surface flux is calculated with centered differences. For example, in Fig. 3, we have

$$F_{i+1/2,j} = D\Delta y\frac{c_{i+1,j} - c_{i,j}}{\Delta x}. \quad (12)$$

The flux gradient across a cut edge, e.g. $(x_{i-1/2}, y_j)$, is approximated by a linear interpolation of two gradients, which are computed by centered differences. A linear interpolation formula between two points y_1 and y_2 as a function of a parameter $\mu \in [0, 1]$ is

$$y^I = (1-\mu)y_1 + \mu y_2. \quad (13)$$

In the case of a cut cell edge, $\mu = (1 + a_{l,m})/2$. For example, to construct $F_{i-1/2,j}$ in Fig. 3, the gradient at $(x_{i-1/2}, y_j)$ and $(x_{i-1/2}, y_{j-1})$ is used,

$$F_{i-1/2,j} = Da_{i-1/2,j}\Delta y\left[\frac{(1+a_{i-1/2,j})}{2}\frac{(c_{i-1,j} - c_{i,j})}{\Delta x} + \frac{(1-a_{i-1/2,j})}{2}\frac{(c_{i-1,j-1} - c_{i-1,j-1})}{\Delta x}\right]. \quad (14)$$

To calculate the flux through a boundary, e.g. $F^f_{i,j}$, we compute the gradient along a line normal to the boundary and centered at the boundary midpoint. To find function values on the normal line, we interpolate using three equally spaced cell centered points (Fig. 4). If the normal line is oriented with an angle of $\pi/4 < |\theta| < 3\pi/4$ relative to the horizontal grid lines, horizontal grid points are used to compute the values on the line. Otherwise vertical points are used. The two points computed along the normal line and the value on the boundary are then used to construct a quadratic polynomial. The concentration gradient is calculated by differentiating the quadratic polynomial and evaluating the result at the boundary point c_f,

$$G^f = \frac{1}{d_2 - d_1}\left[\frac{d_2}{d_1}(c^f - c^I_1) - \frac{d_1}{d_2}(c^f - c^I_2)\right], \quad (15)$$

where c^I_1 and c^I_2 are the interpolated values along the normal line and d_1 and d_2, respectively, are the distances of these two points from the boundary. The flux F^f_{ij} in Eq. (11) is calculated by multiplying G^f by the area of the cut-cell edge a_f and the diffusion coefficient D. Then the boundary condition Eq. (2) becomes the algebraic equation

$$DG^{f}+g(\boldsymbol{c}^{f})=0. \quad (16)$$

Because all gradients are constructed with second order methods, the overall discretization scheme is second order in space. Further discussion on the accuracy of the spatial discretization scheme can be found in [11].

3.3 Time Discretization

Because of the boundary conditions in Eq. (2), operator splitting methods are limited to first order accuracy. Therefore we treat Eqs. (1) and (2) as a differential-algebraic system of non-linear equations. The general form of differential-algebraic system is

$$F(t,\boldsymbol{C},\boldsymbol{C}')=0, \quad (17)$$

where ***C*** is a 2*n* × 1 vector. The first *n* entries in the differential-algebraic system of equations are formed by the discretization of Eq. (1) for the chemical species concentrations, and the (*n* +1)th to 2*n*th entries arise from the boundary conditions given in Eq. (2).

This formulation in conjunction with the discretization of the spatial operator presented in the previous sections allows us to use the DASPK solver described in [12]. Backward differentiation formulas are used to solve Eq. (17) with methods up to fifth order accurate in time. The resulting nonlinear system of equations is solved with a modified Newton's method, given by

$$\boldsymbol{C}^{m+1}=\boldsymbol{C}^{m}-\gamma\left(\alpha\frac{\partial F}{\partial\boldsymbol{C}'}+\frac{\partial F}{\partial\boldsymbol{C}}\right)^{-1}F(t,\boldsymbol{C}^{m},\alpha\boldsymbol{C}^{m}+\beta), \quad (18)$$

where α is a constant that changes when the order or step size changes, β is a vector that depends on the solution at previous time values, and γ is a constant chosen to speed up convergence. Note that each step of the Newton iteration requires inverting the matrix

$$A=\alpha\frac{\partial F}{\partial\boldsymbol{C}'}+\frac{\partial F}{\partial\boldsymbol{C}}. \quad (19)$$

We store this matrix in sparse triple format, and use routines from SPARSKIT [13] to solve the linear system iteratively. The Generalized Minimal Residual (GMRES) method [14] with an incomplete LU (ILU) preconditioner is used to solve the linear system. The combination of the second order accurate cut cell discretization of the Laplacian operator with the fifth order accurate DASPK time integrator yields an overall second order method with no splitting error.

4 Results

4.1 Convergence Tests

To demonstrate the accuracy of our method on a domain containing all types of cut cells, the convergence of our method is compared against an exact solution on a circle. The exact solution to the diffusion equation with a zero Dirichlet boundary condition can be found in terms of Bessel functions. Let λ denote the first root of the Bessel function $J_0(x)$, and *r* be the radius of the circle. Then the expression:

$$f(x,y,t)=\exp\left[-D\left(\frac{\lambda}{r}\right)^{2}t\right]J_{0}\left(\lambda\frac{\sqrt{(x-0.5)^{2}+(y-0.5)^{2}}}{r}\right)\quad(20)$$

is an exact solution to the diffusion equation. Fig. 5 shows the initial condition and the computed solution at $t = 0.2$ using the constants $D = 0.05$ and $r = 0.3$. For visualization purposes, the computational domain and boundary points are triangulated using Triangle [15]. The error in the infinity norm over time is shown in Fig. 6. These results demonstrate the second order accuracy of the numerical method.

Next we tested a nonlinear system in which a protein C can exist in two distinct chemical states: active and inactive. The reactions that convert the protein between the two states are assumed to follow Michaelis-Menten kinetics, which describes the kinetics of many enzymatic reactions including phosphorylation and dephosphorylation events [16]. The protein C is deactivated in the interior of the computational domain according to the following equations:

$$\frac{\partial C_i}{\partial t}=D\Delta C_i+\frac{k_2C_a}{K_{m2}+C_a}\ \frac{\partial C_a}{\partial t}=D\Delta C_a-\frac{k_2C_a}{k_{m2}+C_a}\quad(21)$$

where C_i and C_a are the concentrations of inactive and active protein, respectively, k_2 is the maximum deactivation rate, and K_{m2} is the Michaelis constant. Activation occurs on the boundary, $\partial\Omega$, according to the following boundary conditions:

$$-D\overrightarrow{n}\cdot\nabla C_i|_{\partial\Omega}=\frac{k_1SC_i}{K_{m1}+C_i}\ -D\overrightarrow{n}\cdot\nabla C_a|_{\partial\Omega}=-\frac{k_1SC_i}{K_{m1}+C_i}\quad(22)$$

where k_1 is the maximum activation rate and K_{m1} is the Michaelis constant. The equations are solved in the domain

$$(r,\theta)=r\leq 0.3-0.09\sin(4\theta).\quad(23)$$

The initial concentration of inactive protein is assumed to be constant and equal to 1. There is initially no active protein. Fig. 7 shows a plot of the concentrations at $t = 0.25$.

We compute the error as the difference between successive refinements of the grid. That is, $E(x, y, t) = c_{\Delta x}(x, y, t) - c_{\Delta x/2}(x, y, t)$. Since the grids do not align, the values on the finer grid are averaged before subtracting the value on the courser grid. $E(x, y, t)$ is defined on interior values of the course grid. By computing the error with this method, Fig. 8 shows second order convergence in the infinity norm for the chemical species C_i. The convergence plot of C_a was identical. The time step was set to $\Delta x/4$.

4.2 A two compartment model

In this model, we have two compartments: the cytoplasm and nucleus. The cellular geometry was taken from a yeast cell undergoing chemotrophic growth in the direction of a pheromone gradient [17]. The nucleus is modeled as a circle located toward the front of the cell. The model consists of two species, A and C, with inactive and active forms. Protein C is allowed to enter and exit the nucleus, whereas protein A is restricted to the cytoplasm. Initially both A and C are in their inactive forms. At $t = 0$, the reaction that activates A is initiated. This is meant to model the cell receiving an external signal. Once A is activated it is assumed to interact with the cell membrane, causing a reduction in the protein's diffusion coefficient. Active A can then activate C. The active form of C is only deactivated within the nucleus. This simple model captures some of the signaling events that occur during the pheromone response of yeast. If we denote the concentration of a chemical species with brackets, the equations for the cytoplasmic species are:

$$\frac{\partial[A_c]}{\partial t}=D_1\Delta[A_c]-k_0[A_c]\ \frac{\partial[A_c^*]}{\partial t}=D_2\Delta[A_c^*]+k_0[A_c]\ \frac{\partial[C_c]}{\partial t}=D_1\Delta[C_c]-k_1[A_c^*][C_c]\ \frac{\partial[C_c^*]}{\partial t}=D_1\Delta[C_c^*]+k_1[A_c^*][C_c].$$

where the asterisks denote the active form of the protein, D_1 is the diffusion coefficient in the cytoplasm, D_2 is diffusion coefficient in the membrane, and the k's represent the reaction rates. Subscripts indicate cytosolic and nuclear species. The boundary conditions at the cell membrane are no flux for all chemical species. The nuclear boundary conditions for A species are also no flux, whereas C species are allowed to move through the nuclear membrane and satisfy the conditions

$$\begin{aligned}
-D_1(\overrightarrow{n}\cdot\nabla[C_c])|_{\partial\Omega_2}&=-k_2([C_n]-[C_c])|_{\partial\Omega_2}\\
-D_1(\overrightarrow{n}\cdot\nabla[C_c^*])|_{\partial\Omega_2}&=-k_2([C_n^*]-[C_c^*])|_{\partial\Omega_2}\\
-D_1(\overrightarrow{n}\cdot\nabla[C_n])|_{\partial\Omega_2}&=k_2([C_n]-[C_c])|_{\partial\Omega_2}\\
-D_1(\overrightarrow{n}\cdot\nabla[C_n^*]|_{\partial\Omega_2}&=k_2([C_n^*]-[C_c^*])|_{\partial\Omega_2},
\end{aligned}$$

Nuclear C^* is deactivated according to the equations

$$\frac{\partial[C_n]}{\partial t}=D_1\Delta[C_n]+k_3[C_n^*]\ \frac{\partial[C_n^*]}{\partial t}=D_1\Delta[C_n^*]-k_3[C_n^*].$$

Simulation results are shown in Figs. 9 and 10. All reaction constants were set to one, D_1 = 0.1, D_2 = 0.01, and Δx = 1/100. The initial values were zero except for $[A_c]$ $(x, y, 0)$ = $[C_c]$ $(x, y, 0)$ = 1. The system reached steady state at t = 10. The execution time of the simulation was 22 seconds on Mac Pro desktop computer with dual-core 2.66 GHz Intel Xeon processors. The model simulation suggests a spatial activation gradient can be generated by the position of the nucleus. The inactivation of C in the nucleus leads to a higher concentration of active protein in the rear of the cell in spite of a uniform spatial signal from active A.

4.3 Rho family GTPase Model

The Rho family of GTPases regulates many cellular functions, including polarization and motility. We created a model with three key members of this family, *Cdc*42, *Rac*, and *Rho* (Fig. 11). In our model, an extracellular signal triggers the activation of *Cdc*42 protein on the cell membrane,

$$\left.-D\overrightarrow{n}\cdot\nabla[Cdc42_i]\right|_{\partial\Omega}=\frac{k_1S[Cdc42_i]}{K_{m1}+[Cdc42_i]}\ \left.-D\overrightarrow{n}\cdot\nabla[Cdc42_a]\right|_{\partial\Omega}=-\frac{k_1S[Cdc42_i]}{k_{m2}+[Cdc42_i]}. \quad (24)$$

In the cell interior, active *Cdc*42 is inactivated. A positive feedback loop increases the activation of *Cdc*42,

$$\begin{aligned}\frac{\partial[Cdc42_i]}{\partial t}&=D\Delta[Cdc42_i]+\frac{k_2[Cdc42_a]}{K_{m3}+[Cdc42_a]}-\frac{k_3[Cdc42_a][Cdc42_i]}{K_{m4}+[Cdc42_i]}\\ \frac{[\partial Cdc42_a]}{\partial t}&=D\Delta[Cdc42_a]-\frac{k_2[Cdc42_a]}{k_{m5}+[Cdc42_a]}+\frac{k_3[Cdc42_a][Cdc42_i]}{k_{m6}+[Cdc42_i]}.\end{aligned} \tag{25}$$

Rac is activated by *Cdc*42, and a positive feedback loop increases the concentration of active *Rac*. Active *Rho* increases the deactivation of *Rac* in the cytosol,

$$\begin{aligned}\frac{\partial[Rac_i]}{\partial t}&=D\Delta[Rac_i]+\frac{(k_4[Rho_a]+k_5)[Rac_a]}{K_{m7}+[Rac_a]}-\frac{(k_6[Cdc42_a]+k_7[Rac_a])[Rac_i]}{K_{m8}+[Rac_i]}\\ \frac{\partial[Rac_a]}{\partial t}&=D\Delta[Rac_a]-\frac{(k_4[Rho_a]+k_5)[Rac_a]}{k_{m9}+[Rac_a]}+\frac{(k_6[Cdc42_a]+k_7[Rac_a])[Rac_i]}{k_{m10}+[Rac_i]}\end{aligned} \tag{26}$$

Rho is activated by the active form of *Rac* and deactivated in the interior,

$$\frac{\partial[Rho_i]}{\partial t}=D\Delta[Rho_i]+\frac{k_8[Rho_a]}{K_{m11}+[Rho_a]}-\frac{k_9[Rac_a][Rho_i]}{K_{m12}+[Rho_i]}\quad\frac{\partial[Rho_a]}{\partial t}=D\Delta[Rho_a]-\frac{k_8[Rho_a]}{k_{m13}+[Rho_a]}+\frac{k_9[Rac_a][Rho_i]}{k_{m14}+[Rho_i]} \tag{27}$$

The boundary conditions for *Rac* and *Rho* species are no flux. The steady state distribution at $t = 2.0$ is displayed in Fig. 12. To achieve these results a step size $\Delta x = 1/100$ and a diffusion coefficient $D = 0.1$ were used. The reaction constants from the simulation are listed in Table 1. The initial concentration of inactive chemical species was set to one and zero for active species. The execution time was 39 seconds on a Mac Pro desktop computer with dual-core 2.66 GHz Intel Xeon processors.

In this model, a gradient is formed by protein activation on the membrane. The gradient is propagated to the downstream signaling components *Rac* and *Rho*. Fig. 12 shows that filopodia and thin protrusions have higher activation levels due the increased ratio of cell membrane to cell volume in these regions [4].

5 Discussion

We have developed an accurate and efficient cut-cell method for simulating spatial models of signaling pathways in realistic cellular geometries. Our algorithm builds upon previous work on embedded boundary methods [11,18–20].

These methods have been implemented in two and three dimension for Poisson’s equation, the heat equation, and hyperbolic conservation laws. In [18], a second order implicit method was used to update the heat equation in time [21]. In our method, we use a implicit nonlinear solver to avoid operator splitting error. The capability to solve systems of nonlinear reaction-diffusion equations with nonlinear, mixed-type boundary conditions also distinguishes our method from previous work on embedded boundary methods. The underlying Cartesian-grid based finite volume discretization allows us to use advection schemes originally developed for hyperbolic conservation laws to simulate active transport or motility. In future reports, we will show how level set methods [8,9] can be combined with biochemical reaction networks to investigate the effect of moving boundaries on cell signaling. Future work also includes a three dimensional implementation of our fixed boundary algorithm.

Acknowledgments

We thank Meng Jin and Yi Wu for insightful discussions. This work was supported by NIH grant R01-GM079271.

References

1. Kholodenko B. Cell-signalling dynamics in time and space. Nature Reviews Molecular Computational Biology. 2006; 7(3):165–176.
2. Pertz O, Hodgson L, Klemke R, Hahn K. Spatiotemporal dynamics of RhoA activity in migrating cells. Nature. 2006; 440(7087):1069–1072. [PubMed: 16547516]
3. Neves SR, Tsokas P, Sarkar A, Grace E, Rangamani P, Taubenfeld S, Alberini C, Schaff J, Slizter R, Moraru I, Iyengar R. Cell shape and negative links in regulatory motifs together control spatial information flow in signaling networks. Cell. 2008; 133(4):666–680. [PubMed: 18485874]
4. Meyers J, Craig J, Odde D. Potential for Control of Signaling Pathways via Cell Size and Shape. Current Biology. 2006; 16(17):1685–1693. [PubMed: 16950104]
5. Levine H, Kessler D, Rappel W. Directional sensing in eukaryotic chemotaxis: A balanced inactivation model. PNAS. 2006; 103(26):9761–9766. [PubMed: 16782813]
6. Schaff J, Slepchenko B, Choi Y, Wagner J, Resasco D, Loew L. Analysis of nonlinear dynamics on arbitrary geometries with the Virtual Cell. Chaos. 2001; 11(1):115–131. [PubMed: 12779447]
7. Zienkiewicz, O.; Taylor, R. The Finite Element Method Set. Sixth Edition. Butterworth-Heinemann; 2005.
8. Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation. J. Comput. Phys. 1988; 79(1):12–49.
9. Sethian, J. Level set methods and fast marching methods : evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. 2nd Edition. Cambridge University Press; 1999.
10. Malladi R, Sethian JA, Vemuri BC. A fast level set based algorithm for topology independent shape modeling. Journal of Mathematical Imaging and Vision, special issue on Topology and Geometry in Computer Vision 6. 1996:269–290.
11. Johansen H, Colella P. A Cartesian grid embedded boundary method for Poisson’s equation on irregular Domains. J. Comput. Phys. 1998; 147(1):60–85.
12. Brown PN, Hindmarsh A, Petzold L. Using Krylov methods in the solution of large-scale differential-algebraic systems. SIAM J. Sci. Comput. 1994; 15(6):1467–1488.
13. Saad, Y. SPARSKIT: A basic tool-kit for sparse matrix computations (Version 2). 2005 Mar. http://www-users.cs.umn.edu/~saad/software/SPARSKIT/sparskit.html
14. Saad Y, Schultz M. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear-systems. SIAM J. Sci. Stat. Comp. 1986; 7(1):856–869.
15. Shewchuk, JR. Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In: Lin, MC.; Manocha, D., editors. Applied Computational Geometry: Towards Geometric Engineering, Vol. 1148 of Lecture Notes in Computer Science; from the First ACM Workshop on Applied Computational Geometry; Springer-Verlag; 1996. p. 203-222.
16. Kholodenko B, Brown G, Hoek J. Diffusion control of protein phosphorylation in signal transduction pathways. Biochem. J. 2000; 350(Pt. 3):901–907. [PubMed: 10970807]
17. Nayak S. Bem1 in yeast. URL http://biodynamics.ucsd.edu/sujata/wmather/.
18. McCorquodale P, Colella P, Johansen H. A Cartesian grid embedded boundary method for the heat equation on irregular domains. J. Comput. Phys. 2001; 173(2):620–635.
19. Schwarz P, Barad M, Colella P, Ligocki T. A Cartesian grid embedded boundary method for the heat equation and Poisson’s equation in three dimensions. J. Comput. Phys. 2006; 221(2):531–550.
20. Colella P, Graves DT, Keen BJ, Modiano D. A Cartesian grid embedded boundary method for hyperbolic conservation laws. J. Comput. Phys. 2006; 211(1):347–366.
21. Twizell E, Gumel AB, Arigu MA. Second order L_0-stable methods for the heat equation with time-dependent boundary conditions. Adv. Comp. Math. 1996; 6(1):333–352.

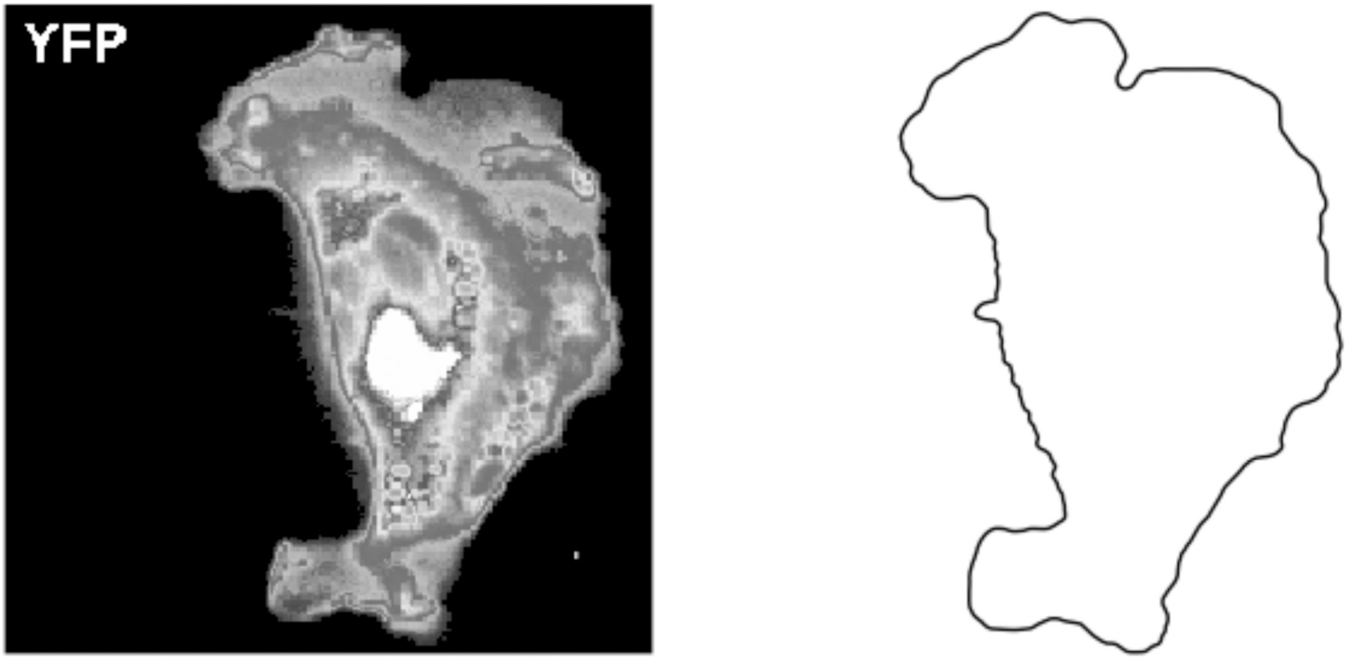

Fig. 1.
Grayscale image of a mouse fibroblast taken from supplemental data in [2] (left) and the smoothened boundary (right).

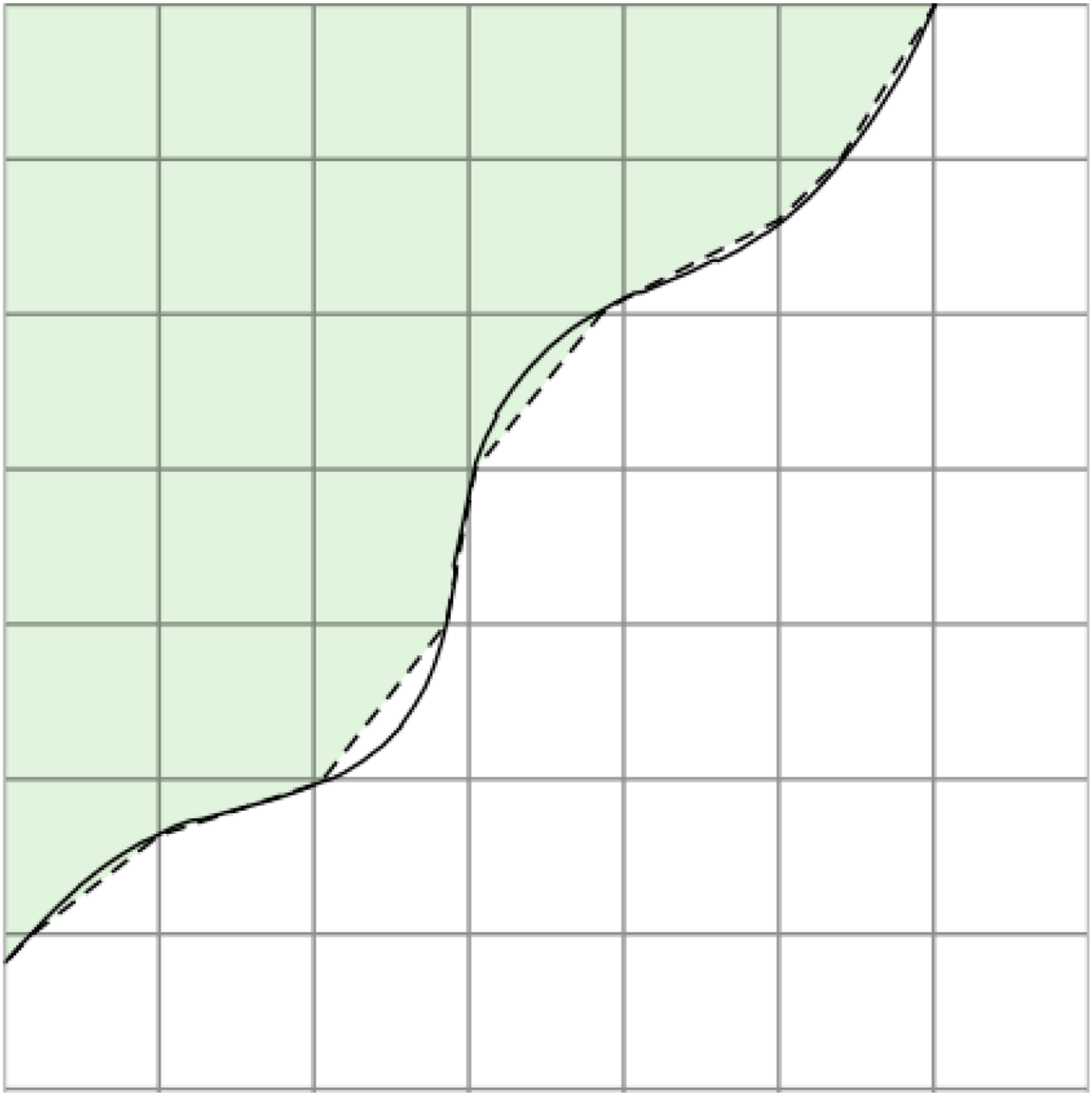

Fig. 2.

Computational boundary (dashed line) with an assumed higher order representation of the cell boundary drawn as a solid line.

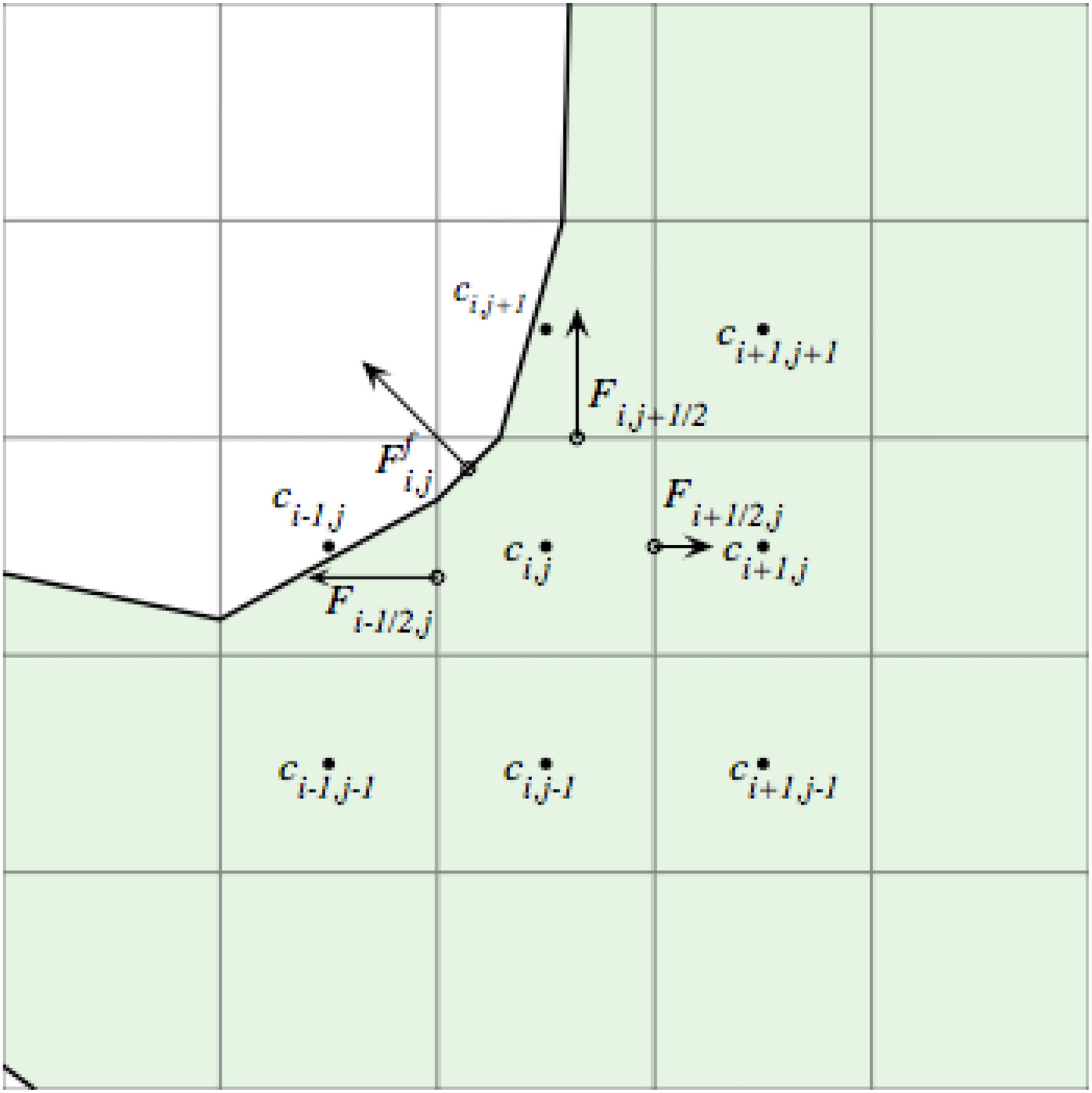

Fig. 3.

Diagram of fluxes for cut cells where shaded boxes indicate cells that are inside the boundary.

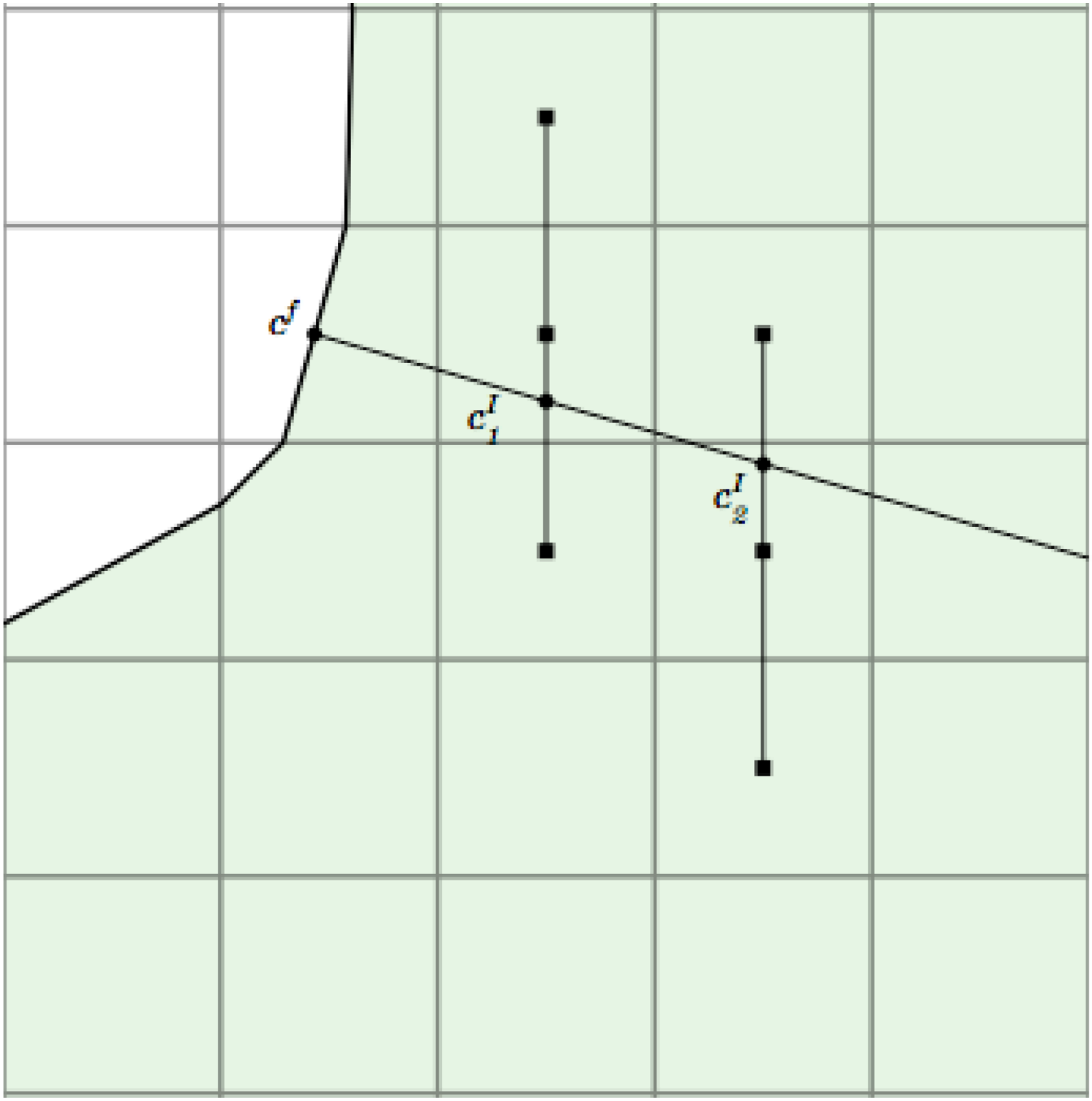

Fig. 4.

Circles indicate interpolated values that depend on the square grid values.

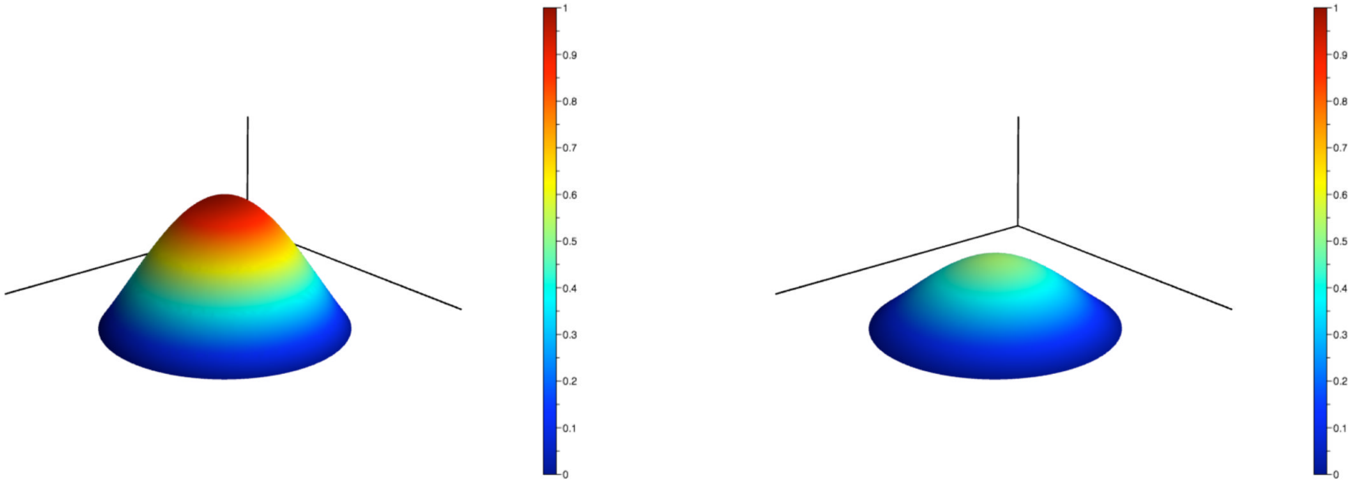

Fig. 5.

Initial condition for the diffusion equation (left) and computed solution at $t = 0.2$ (right).

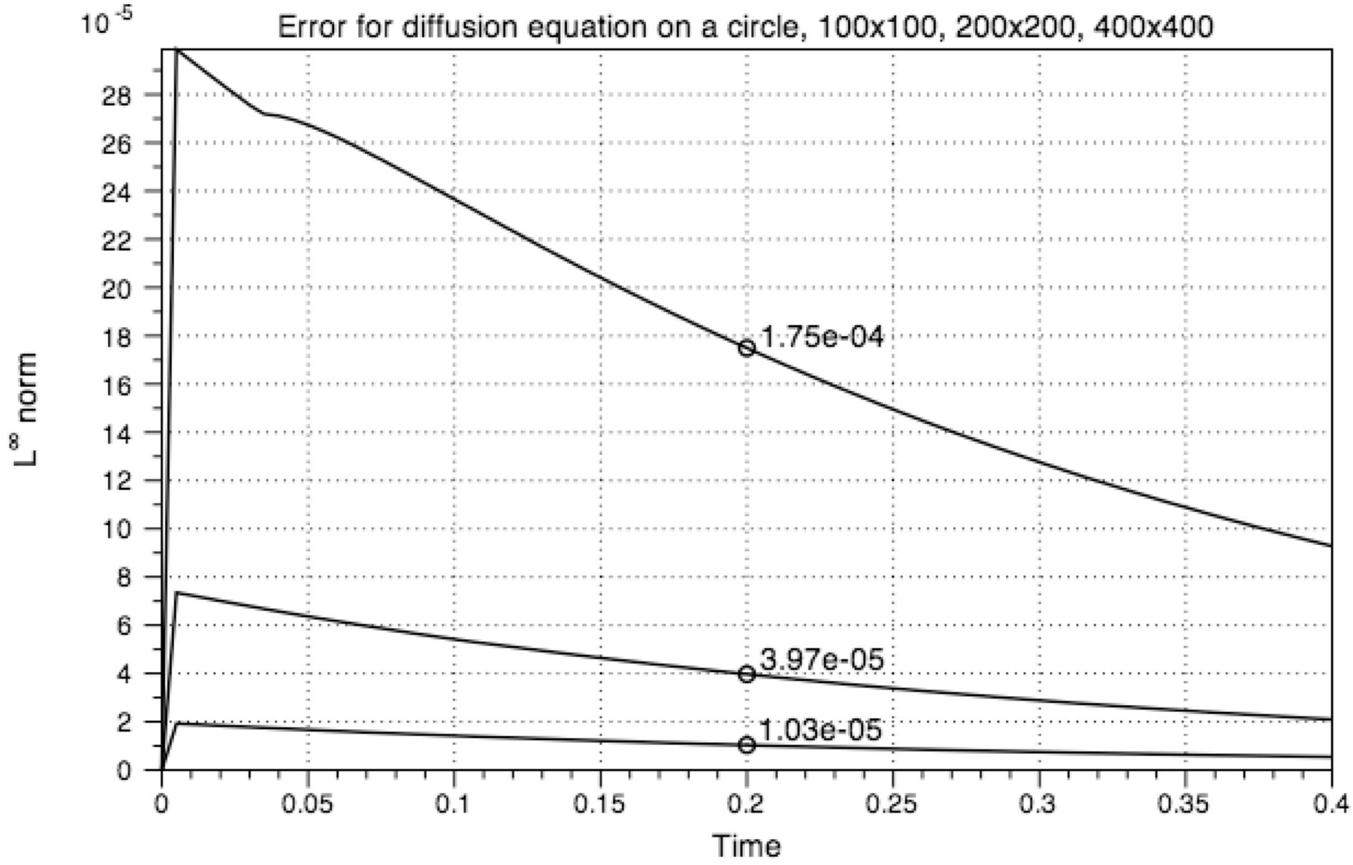

Fig. 6.
Error in the infinity norm for the diffusion equation.

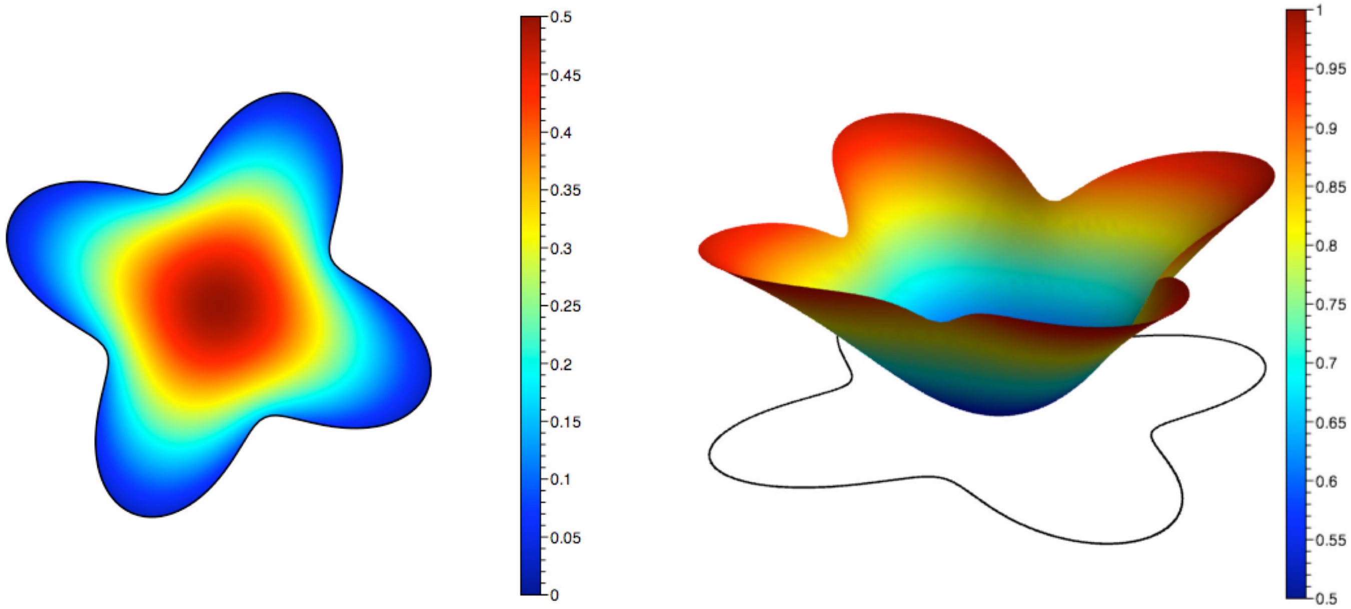

Fig. 7.

Concentrations of inactive (left) and active (right) species at t=0.25 with the coefficients D = 0.1, $k_1 = k_2 = S$ = 1.0, and $K_{m1} = K_{m2}$ = 0.2.

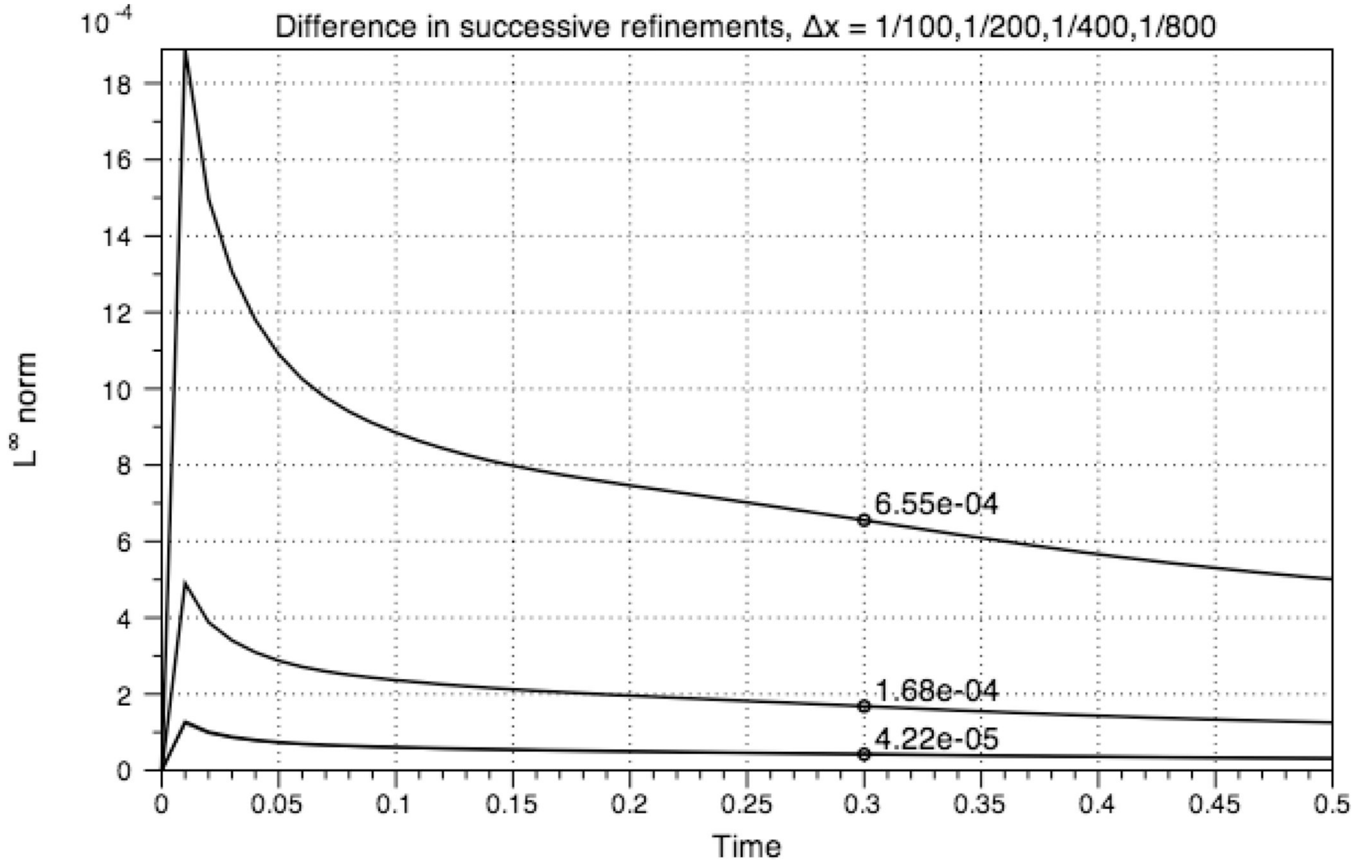

Fig. 8.

C_i error in the infinity norm

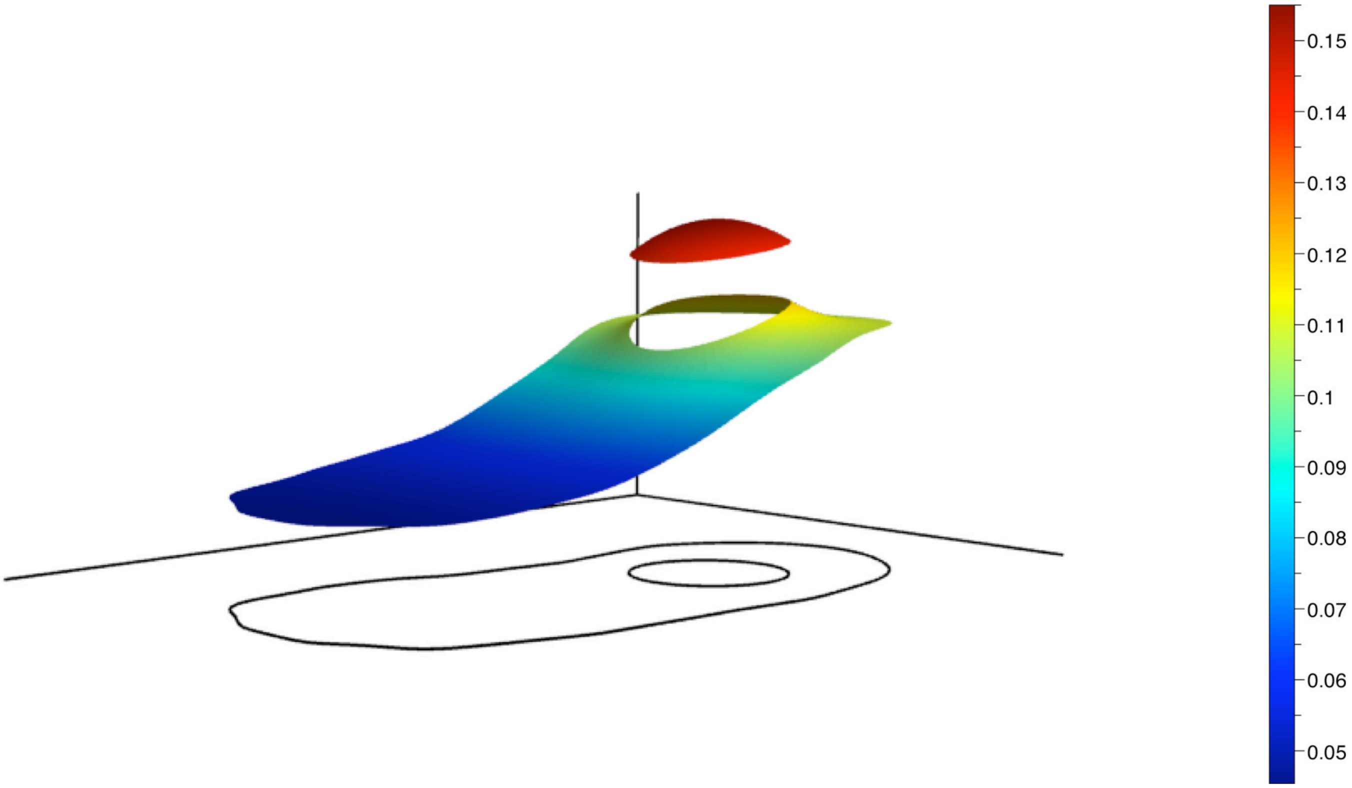

Fig. 9.

Steady state concentration of inactive *C* in the cytoplasm and nucleus.

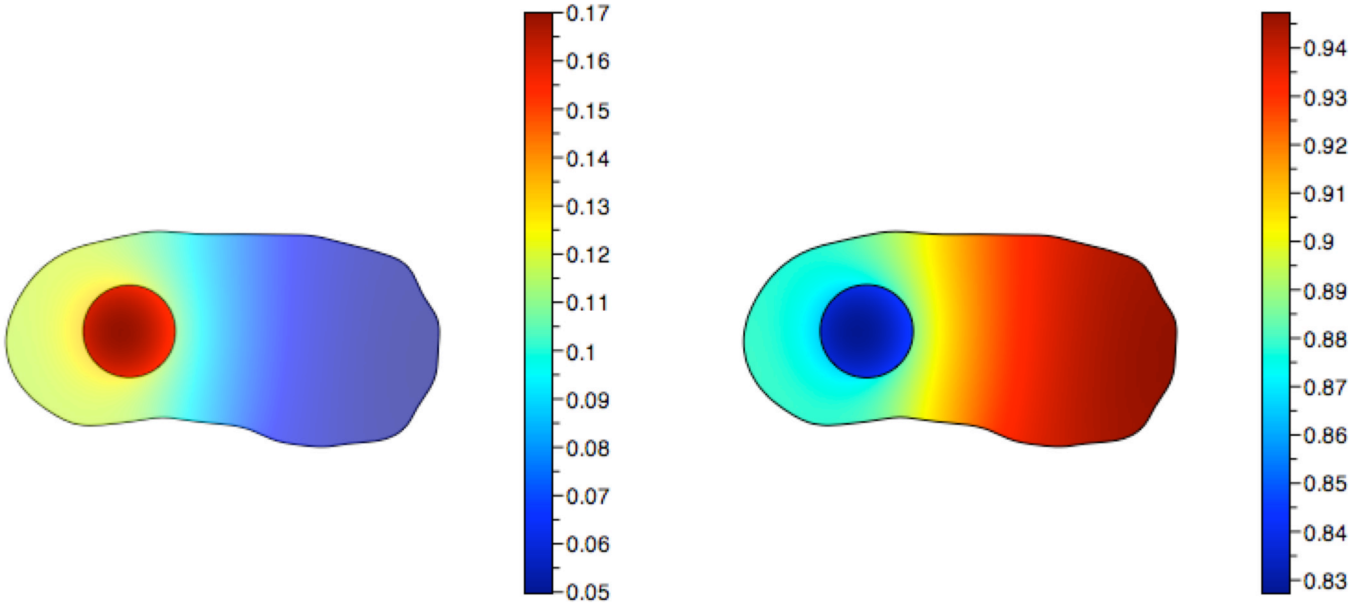

Fig. 10.

Steady state concentration values for inactive *C* species (left) and active *C* species (right).

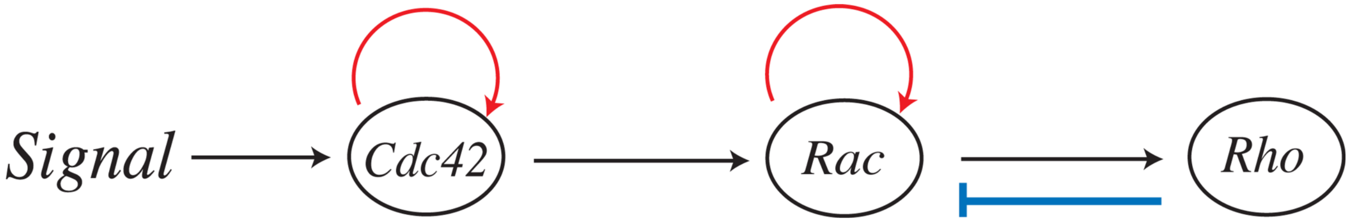

Fig. 11.
Schematic of Interactions

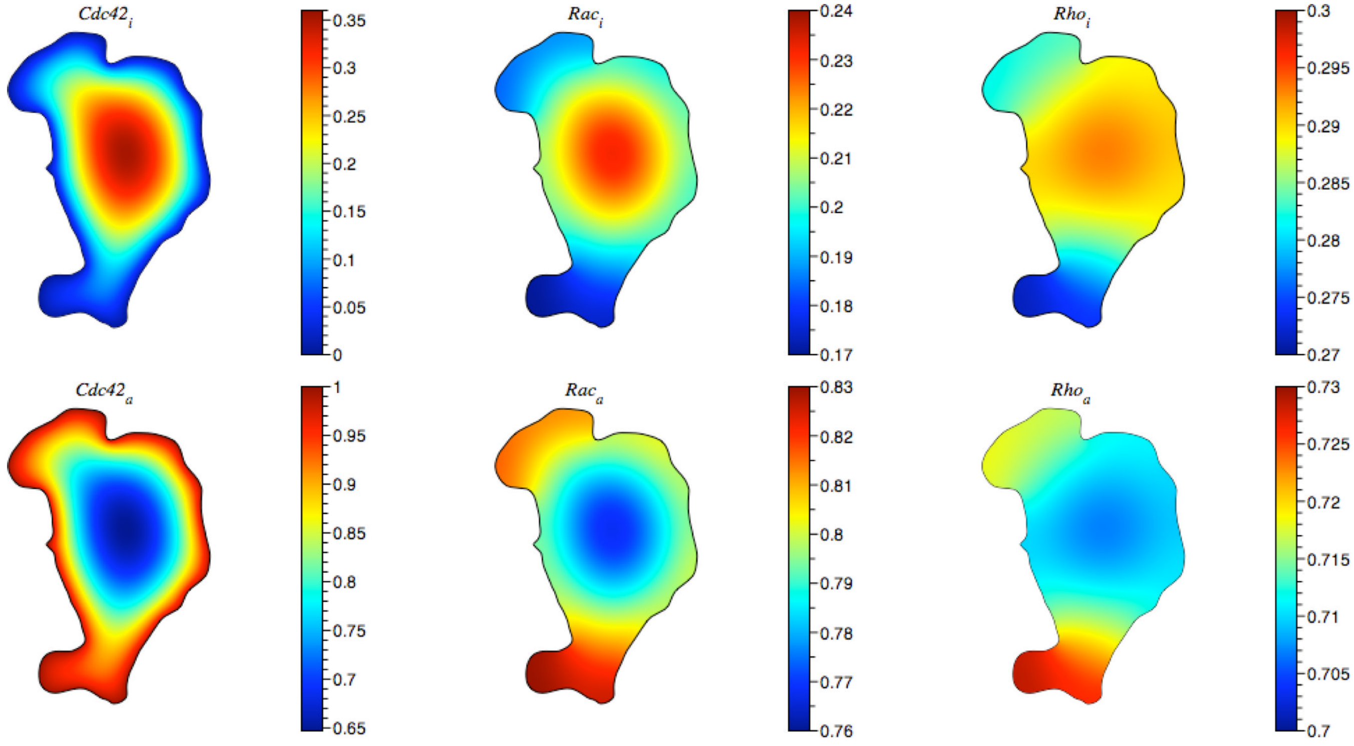

Fig. 12.

Steady state distribution of protein concentration amounts in a fibroblast. The boundary was taken from a live cell image [2].

Table 1

Reaction constants used in simulation of the Rho GTPase model.

$S = 1$	$k_4 = 3$	$k_8 = 3$	$K_{m3} = 0.2$	$K_{m7} = 0.2$	$K_{m11} = 0.2$
$k_1 = 5$	$k_5 = 1$	$k_9 = 5$	$K_{m4} = 0.2$	$K_{m8} = 0.2$	$K_{m12} = 0.2$
$k_2 = 3$	$k_6 = 5$	$K_{m1} = 0.2$	$k_{m5} = 0.2$	$k_{m9} = 0.2$	$k_{m13} = 0.2$
$k_3 = 1$	$k_7 = 1$	$k_{m2} = 0.2$	$k_{m6} = 0.2$	$k_{m10} = 0.2$	$k_{14} = 0.2$