COMSOL CONFERENCE 2016 BOSTON

Using COMSOL Multiphysics® for Benchmarking Problems in Cell Migration

Masoud Nickaeen*, Igor L. Novak, Alexander Mogilner, Boris M. Slepchenko

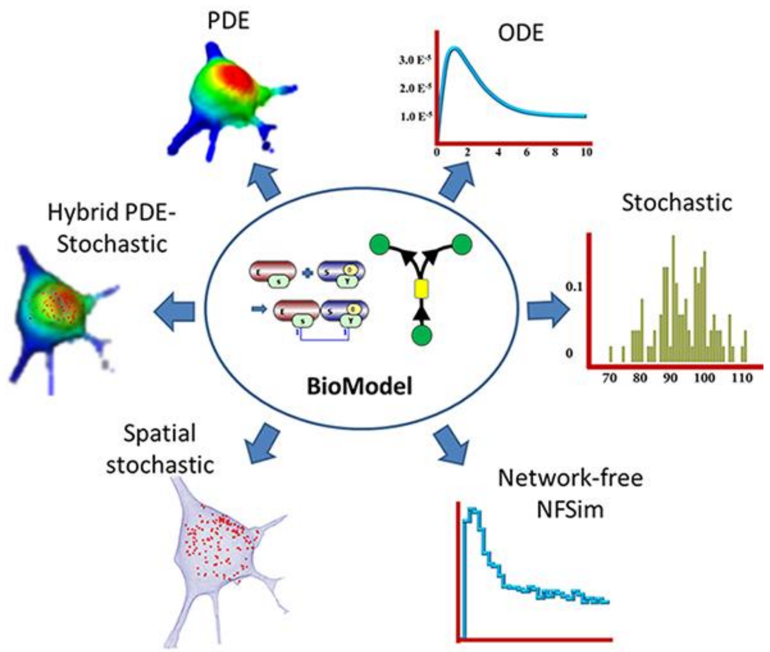
*Richard D. Berlin Center for Cell Analysis and Modeling
University of Connecticut Health Center

Virtual Cell (VCell) moving boundaries project

- development of numerical tools for modeling cell shape dynamics and motility

- developing and implementing such tools within a general-purpose computational framework, (VCell)

- making simulation of processes in migrating cells accessible to cell biologists and bio-physicists



current VCell capabilities (vcell.org)

Moving boundary algorithm

Using an Eulerian approach to solving a parabolic system in the domain with moving boundaries, assuming the sharp boundary is accurately tracked

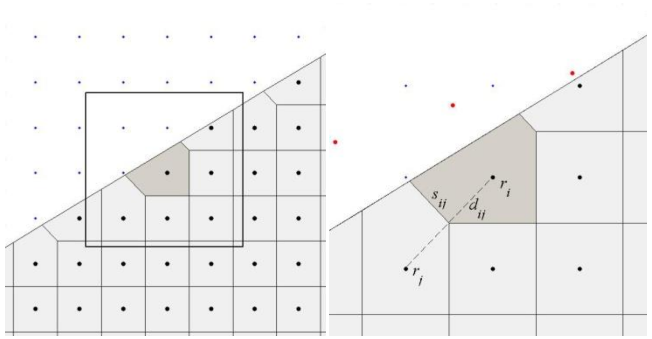
Ideas behind the algorithm

• Domain discretization is done by applying Voronoi decomposition to a fixed orthogonal grid

• Local mass conservation is ensured by finite volume spatial discretization and natural-neighbor interpolation

• Front tracking by integrating with FronTier (robust front-tracking technique in 2D and 3D)

- ✓ extending the algorithm by coupling cell kinematics and intracellular dynamics
- ✓ using COMSOL to evaluate the accuracy of the extended algorithm

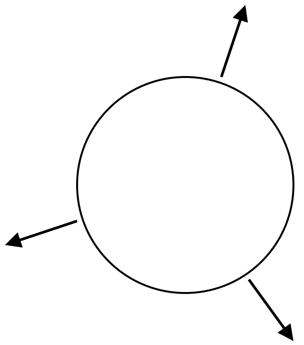


Generating control volumes by Voronoi decomposition.

Novak and Slepchenko, JCP - 2014

Diffusion inside an expanding circle

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot (\mathrm{D}\, \nabla u) & in \quad \Omega(t) \\ (\mathrm{D}\nabla u + \boldsymbol{v}_b u)\boldsymbol{n} = 0 & on \quad \partial\Omega(t) \end{cases}$$



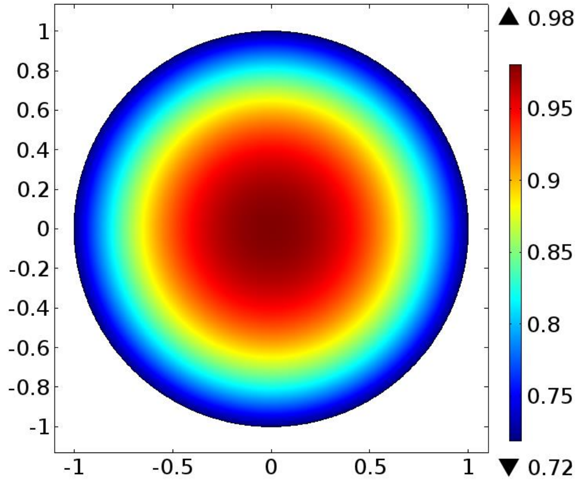
$$v_b \boldsymbol{n} = u|_{\partial\Omega}$$

$$\mathrm{u}(\mathbf{x}, 0) = 1$$

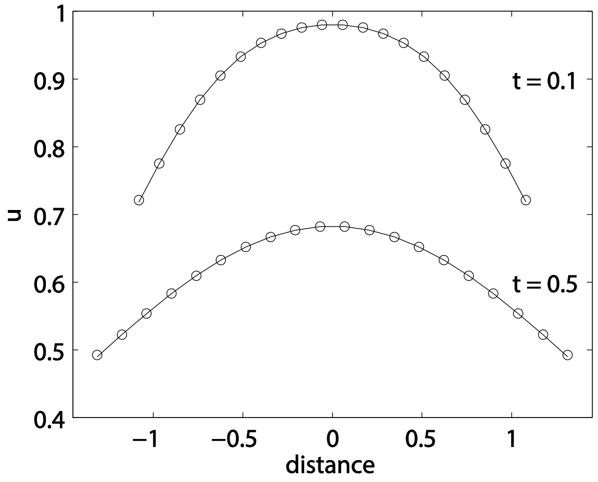
change of variables $\quad X = \frac{x}{R(t)}$ and $Y = \frac{y}{R(t)}$

$$\begin{cases} \dfrac{\partial u}{\partial t} = \widetilde{D}\Delta u - \widetilde{\boldsymbol{v}}\nabla u & in \quad \Omega_0 \\ \left(\widetilde{D}\nabla u - \widetilde{\boldsymbol{v}}u\right)\boldsymbol{n} = 0 & on \quad \partial\Omega_0 \qquad (1) \\ \widetilde{D} = \frac{D}{R(t)^2} \quad , \quad \widetilde{\boldsymbol{v}} = -\frac{\dot{R}(t)}{R(t)}(X, Y) \\ \dot{R}(t) = u|_{\partial\Omega} \end{cases}$$

✓ system of equations (1) is solved using COMSOL ‘Transport of Diluted Species’ module



solution at $t = 0.1$



comparison with COMSOL

Error analysis using COMSOL solution

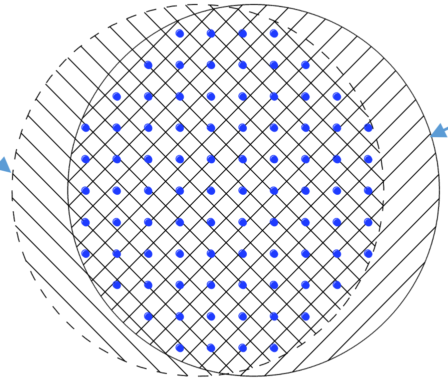
Interface position error

$$e_{Pos} = \frac{A_U - A_I}{A_e}$$

$$A_e = |\Omega_e|$$

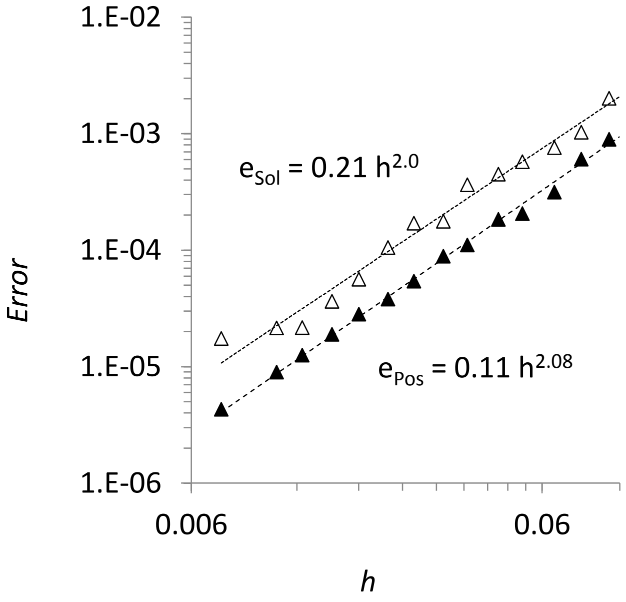
$$A_I = |\cap(\Omega_e, \Omega_h)|$$

$$A_U = |\cup(\Omega_e, \Omega_h)|$$



Solution error

$$e_{Sol} = \frac{\sqrt{\sum\left(u^i - u_h^i\right)^2}}{\sqrt{\sum(u^i)^2}} \quad , \quad \left(x^i, y^i\right) \in \cap(\Omega_e, \Omega_h)$$



Minimal models of actin-driven cell motility

- approximation with few variables!
- elements of cell mechanics are included

Mechanisms

advection-diffusion of myosin: $\partial_T M = D\Delta M - \kappa \nabla \cdot (\mathbf{U} M)$

cell mechanics and adhesion: $\partial_T \mathbf{U} = \mu \Delta \mathbf{U} + \nabla \cdot (M\mathbf{I}) - \xi \mathbf{U}$

membrane kinematics: $\mathbf{V}_\mathrm{f} = V_\mathrm{p} \mathbf{n} + \mathbf{U}|_{\partial\Omega}$

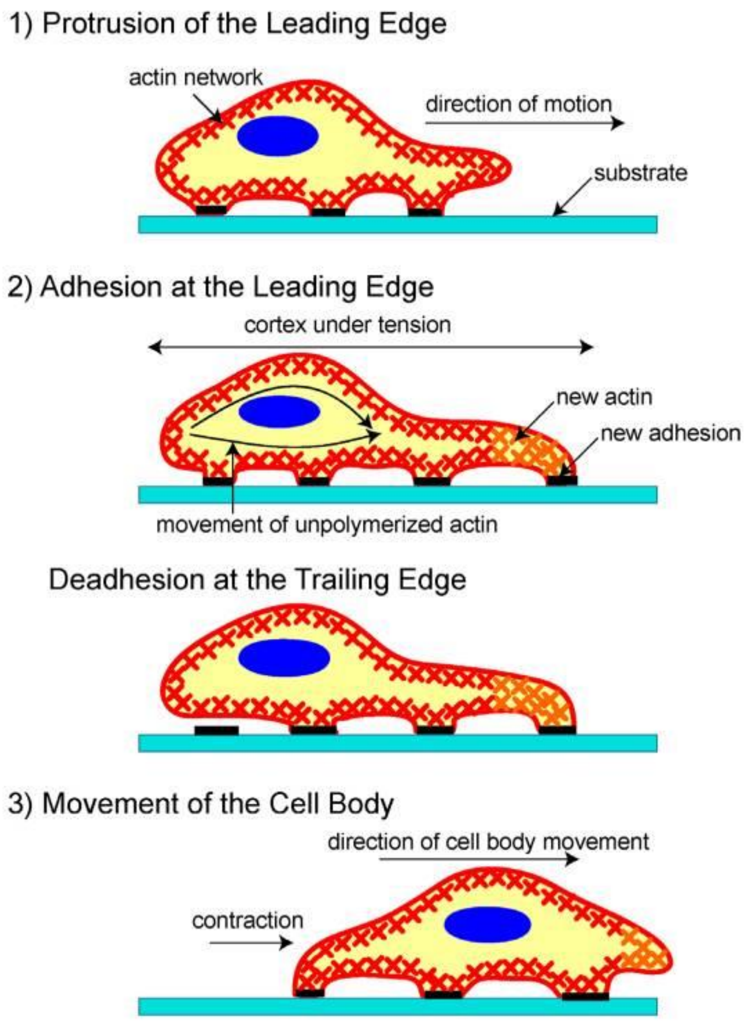
two formulations of protrusion give rise to two models:

‘Zero-velocity’ model:

$$\mathbf{n} \cdot \mathbf{U}|_{\partial\Omega} = 0 \qquad V_\mathrm{p} = \frac{V_0}{M_0 + M|_{\partial\Omega}} - K|\Omega|$$

‘Zero-stress’ model:

$$\mathbf{n} \cdot (\mu \nabla \mathbf{U} + M \cdot \mathbf{I})|_{\partial\Omega} = 0 \qquad V_\mathrm{p} = \frac{V_0}{M_0} - K|\Omega|$$



Alternative solution using COMSOL

- The equation for myosin and actin velocities were implemented using ‘Coefficient Form’ PDE framework

  - ✓ Ranking-Hugoniot BC for myosin $\mathbf{n}\cdot(-D\nabla M + (\kappa\mathbf{U} - \mathbf{V}_{\mathrm{f}})M)|_{\partial\Omega} = 0$
  - ✓ biquadratic FEs

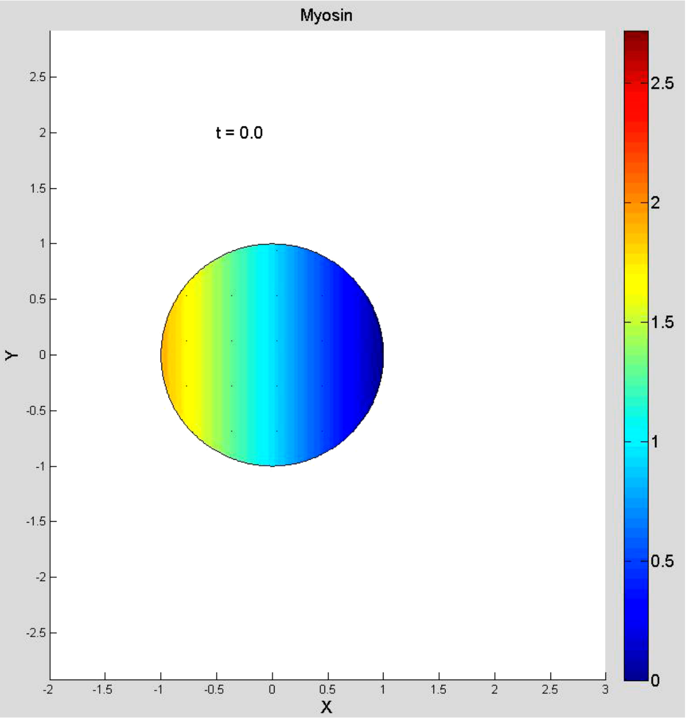
- Moving domain problem was implemented using ‘Moving Mesh’ framework, which is based on ALE finite element methods

  - ✓ Laplace mesh smoothing was used for the interior mesh deformations
  - ✓ geometry shape order: 1

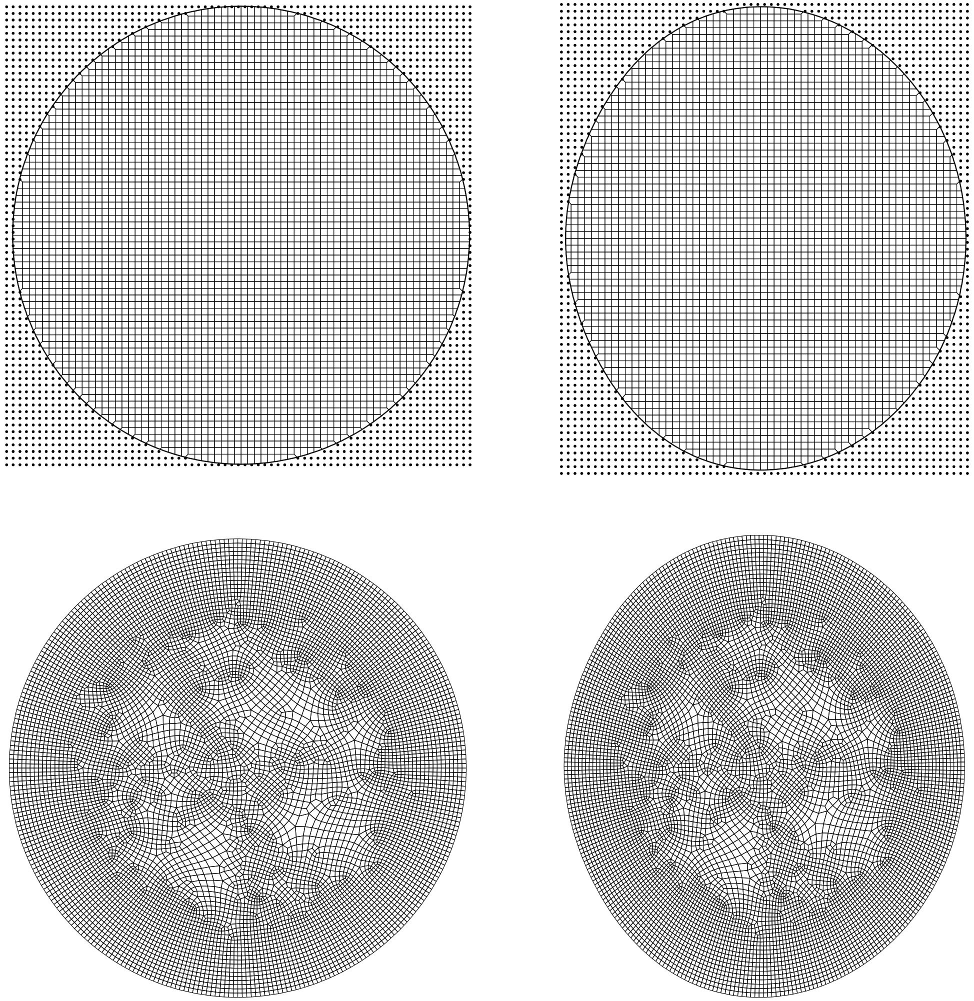
- The system was solved monolithically

  - ✓ linearization was performed using Newton’s method with a constant damping factor
  - ✓ direct (MUMPS) linear solver with the default parameters
  - ✓ backward differentiation time stepping scheme with the default parameters

Zero-velocity model: slightly deforming and translating



solution up to t = 8, pseudo-color is myosin and vectors represent actin velocity



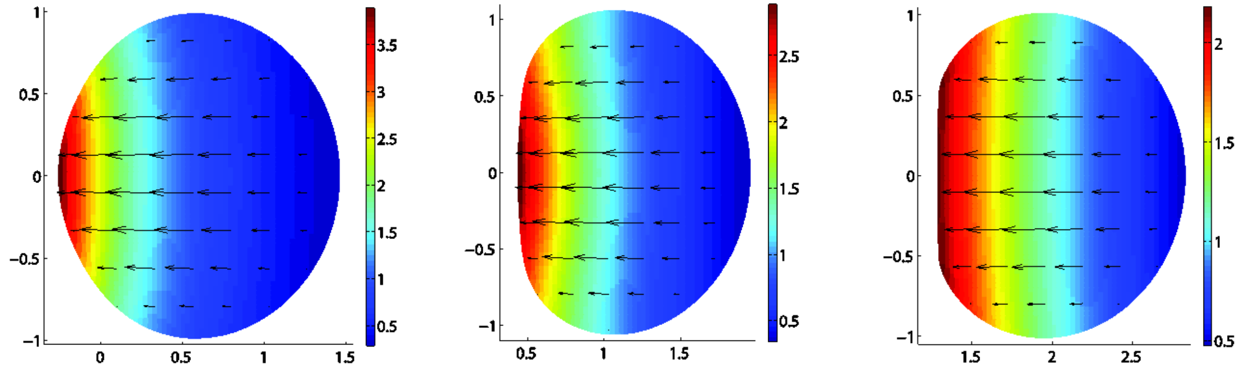
VCell mesh

COMSOL mesh

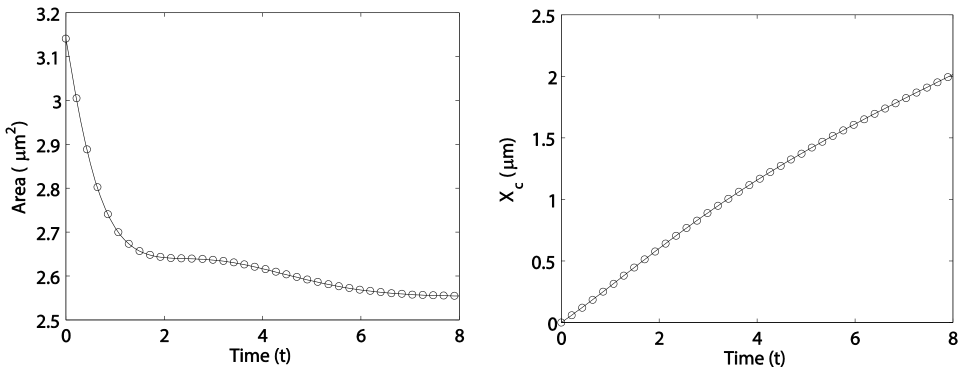
$t = 0$ $t = 2$

comparison of the computational meshes at different times

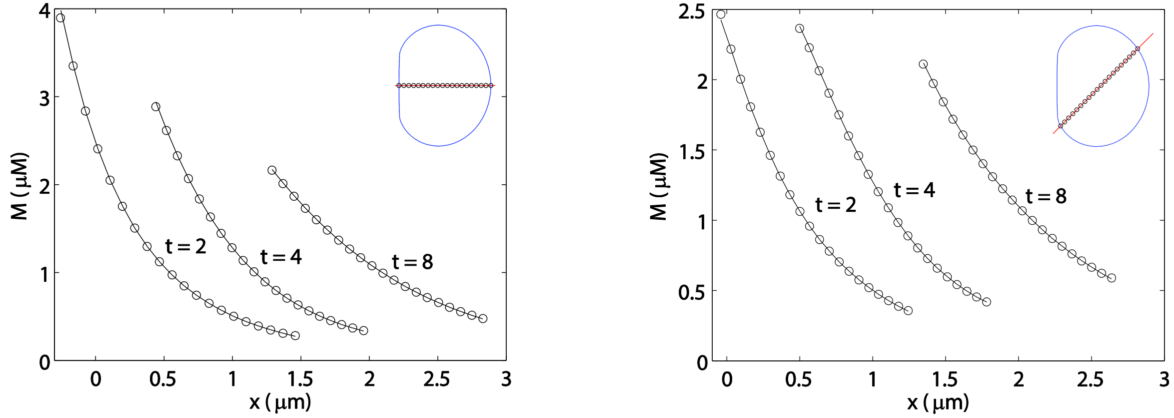
Zero-velocity model: slightly deforming and translating



$t = 2$ $t = 4$ $t = 8$



cell area and centroid comparison against COMSOL



snapshot of the solution at different times (top)
comparison with COMSOL along cut-lines (bottom)

time (s)	2	4	8
diff_{Pos}	0.001016	0.001536	0.002515
diff_{Sol}	0.002709	0.002181	0.002003

relative differences in comparison against COMSOL

Thank you for your attention!