AIP | The Journal of Chemical Physics

# A framework for discrete stochastic simulation on 3D moving boundary domains

Brian Drawert, Stefan Hellander, Michael Trogdon, Tau-Mu Yi, and Linda Petzold

Citation: The Journal of Chemical Physics **145**, 184113 (2016); doi: 10.1063/1.4967338

View online: http://dx.doi.org/10.1063/1.4967338

View Table of Contents: http://scitation.aip.org/content/aip/journal/jcp/145/18?ver=pdfcov

Published by the AIP Publishing

# A framework for discrete stochastic simulation on 3D moving boundary domains

Brian Drawert,[1,a),b)] Stefan Hellander,[1,a)] Michael Trogdon,[2,a)] Tau-Mu Yi,[3] and Linda Petzold[1,2]

[1)] *Department of Computer Science, University of California-Santa Barbara, Santa Barbara, California 93106, USA*
[2)] *Department of Mechanical Engineering, University of California-Santa Barbara, Santa Barbara, California 93106, USA*
[3)] *Department of Molecular, Cellular, and Developmental Biology, University of California-Santa Barbara, Santa Barbara, California 93106, USA*

(Received 24 August 2016; accepted 26 October 2016; published online 14 November 2016)

We have developed a method for modeling spatial stochastic biochemical reactions in complex, three-dimensional, and time-dependent domains using the reaction-diffusion master equation formalism. In particular, we look to address the fully coupled problems that arise in systems biology where the shape and mechanical properties of a cell are determined by the state of the biochemistry and vice versa. To validate our method and characterize the error involved, we compare our results for a carefully constructed test problem to those of a microscale implementation. We demonstrate the effectiveness of our method by simulating a model of polarization and shmoo formation during the mating of yeast. The method is generally applicable to problems in systems biology where biochemistry and mechanics are coupled, and spatial stochastic effects are critical. [http://dx.doi.org/10.1063/1.4967338]

## I. INTRODUCTION

Stochastic simulation of biochemical reactions has become an essential part of systems biology. Many examples exist where mean-field or deterministic analysis is insufficient to capture the relevant dynamics of real biological systems.[1–3] In particular, systems in which the copy number of any relevant species is small will often be more accurately modeled with stochastic simulation. There exist several methods to model biochemical reactions stochastically, the most popular of which is the Stochastic Simulation Algorithm (SSA) or Gillespie algorithm.[4] This algorithm assumes that the system is spatially homogeneous, or well-mixed, which is not the case in many interesting biological problems.

Polarization in yeast[5] and neutrophils,[6] Min oscillations during cell division of *E. coli*,[7,8] and development[9] are a few examples where the well-mixed assumption does not apply. There are several different methods for modeling spatial stochastic biochemical reactions, which can be broadly grouped into two categories. First are the particle-tracking, or free-space based on the Brownian dynamics formalism,[10] methods that resolve the system on a microscopic scale.[11–13] These methods are more accurate but can be quite difficult to simulate in an efficient manner. The other group of methods works on the mesoscopic scale and is based on the reaction-diffusion master equation (RDME) formalism.[7,14–16] These methods discretize the domain into spatially homogeneous subvolumes (or voxels). Reactions within a voxel are modeled with the SSA algorithm, while diffusion between voxels is modeled as events occurring at intensities chosen to be consistent with the diffusion equation. The RDME is significantly faster to simulate than microscopic methods, with some sacrifice in accuracy. A more detailed background on these methods will be given in Section II A. One assumption that underlies all of the methods mentioned here is that the physical domain is static in time. This is often not the case in biology.

Shmoo growth during the mating of yeast,[17] tip growth in fungal hyphae,[18] chemotaxis in neutrophils,[6] cell migration,[19] and cell division[7,8] are some examples where the physical domain of the cell is changing in time. There has also been recent work dealing with the critical role that geometry can play in fundamental biological processes, such as polarization.[20] In this paper, we present a method to efficiently model stochastic reaction-diffusion systems in complicated, three dimensional (3D), time-dependent geometries using the RDME framework. Previous work to model stochastic reaction-diffusion systems in time-dependent domains has focused on particle-based approaches.[21] While these methods are viable in some settings, it is generally accepted that the accurate particle-tracking methods become prohibitively expensive as the system size grows large. An efficient implementation of the RDME on time-dependent domains can effectively handle complex geometries as well as large reaction networks, while still being practical in terms of simulation time. There has also been previous work that has dealt with particle migration on 1D growing domains using the RDME formalism, where results have been compared to those of partial differential equation (PDE) models.[22–24] Additionally, we note

[a)] B. Drawert, S. Hellander, and M. Trogdon contributed equally to this work.
[b)] Author to whom correspondence should be addressed. Electronic mail: briandrawert@gmail.com

that the particle-based software Smoldyn[11] has capabilities to change the size of domain during simulation.

A key problem that our method addresses arises in systems where the biochemical reaction network is fully coupled to the mechanical properties of the physical domain. Our motivating example is the growth of the mating projection in yeast. In this system, enzymes modify the material properties of the cell wall, softening it, and as a result the force of the internal turgor pressure deforms the cell. At the same time, cell wall construction proteins strengthen the cell wall and slow the movement. A diagram of this process is illustrated in Figure 1. Our method does not propose to solve arbitrary mechanics of the model system under consideration. Instead, we integrate an external function or software that models and solves the equations governing the mechanics of the system, which will take as input the state of our biochemical system and provide as output the instantaneous velocity of the boundary of the domain. It is important to note, however, that this is simply one possible problem that can be handled by our proposed algorithm. In general, the function that moves the boundary does not need to be defined by mechanics (e.g., a constantly expanding sphere). The only requirement is that there is a velocity field provided to move the boundary, which can be empirically or theoretically derived.

It is important to note that in our method it is necessary to have a separation of time scales between the diffusion of the biochemical species and the movement of the boundary. It is critical that diffusion is faster than the boundary velocity to accurately simulate the system. In Sections IV A and IV B we characterize the error our method incurs and how it relates to the difference in time scales of diffusion and the velocity of the boundary. Essentially, it is possible to find a time step small enough to satisfy a user specified error tolerance if there is in fact a separation of time scales.

In this paper, we present a method to efficiently simulate stochastic reaction-diffusion models coupled to time-dependent domains using the RDME formalism. An outline for the rest of this paper is as follows: in Section II, we give a brief review of the RDME and how it can be simulated efficiently in complex domains using the Next Subvolume Method (NSM)[15] on unstructured meshes.[25] In Section III, we present our method and discuss implementation and theoretical considerations. Next, in Section IV, we present three examples that serve to show the convergence properties and accuracy of the method along with demonstrating the applicability of the method to biologically relevant problems. Specifically, in Section IV A, we compare the results of our method to a microscale implementation of a single species diffusing within an expanding 1D line that can react with the boundary. This example serves to demonstrate the accuracy and convergence (in the spatial distribution of molecules) of our method compared with a microscale implementation over a range of expansion velocities and time steps. To accurately compare these different scales of simulation, we also derived the relationship between mesoscale and microscale reaction rates (see Appendix C for details). Next, in Section IV B, we present a more biologically relevant, yet still theoretically tractable, model of polarization in yeast, introduced in Ref. 26. This model contains a density dependent switch for polarization which we explore through an expanding and contracting sphere and compare to theoretical steady state results from Ref. 26. Lastly, in Section IV C we present a model in which the state of the biochemical system dictates the movement of the boundary. In particular, we present a new model for the polarization of Cdc42 in mating yeast (see Appendix A for details) and qualitatively compare to experimental data. Finally, we end the paper in Section V with a discussion of the method and our results.

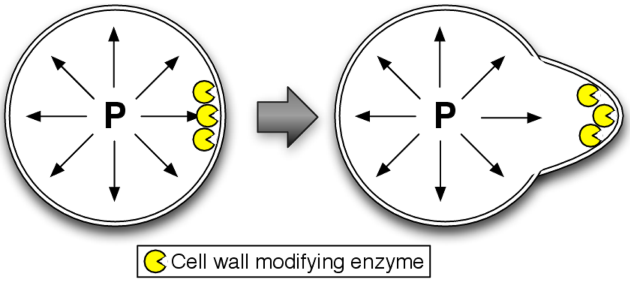

FIG. 1. Diagram illustrating the how the yeast mating projection grows, our motivating example. Cell wall modifying enzymes (yellow) are localized to the polarized region of the cell membrane. These enzymes soften the cell wall. The internal turgor pressure pushes on the cell wall, deforming it, and creating the mating projection.

## II. BACKGROUND

### A. The reaction-diffusion master equation

As mentioned above, the RDME[27] is a mesoscopic model for spatial stochastic chemical reactions. It gives the time evolution of the probability distribution for the state of the system. First, the physical domain is partitioned into $K$ nonoverlapping subvolumes or voxels, similar to numerical methods for PDEs. Molecules are taken to be point particles and the state of the system is the discrete number of molecules of each species for each of the voxels in the mesh. The computational mesh can either be a structured Cartesian grid or an unstructured triangular or tetrahedral mesh. Here, we focus on the unstructured case which will be discussed further in Section II C. Modeling the reaction-diffusion dynamics as a Markov process gives the following forward Kolmogorov equation for the time evolution of $p(\mathrm{x},t) = p(x,t|\mathrm{x}_0,t_0)$ (the probability that the system can be found in state x at time $t$, conditioned on the initial condition $\mathrm{x}_0$ at time $t_0$)

$$\frac{\partial p(\mathbf{x},t)}{\partial t} = \mathcal{R}p(\mathbf{x},t) + \mathcal{D}p(\mathbf{x},t), \tag{1}$$

$$\mathcal{R}p(\mathbf{x},t) = \sum_{i=1}^{K}\sum_{r=1}^{M} a_{ir}(\mathbf{x}-\nu_{ir})p(\mathbf{x}-\nu_{ir},t) - a_{ir}(\mathbf{x})p(\mathbf{x},t), \tag{2}$$

$$\mathcal{D}p(\mathbf{x},t) = \sum_{s=1}^{N}\sum_{i=1}^{K}\sum_{j=1}^{K} d_{sij}(\mathbf{x}-\mu_{sij})p(\mathbf{x}-\mu_{sij},t) - d_{sij}(\mathbf{x})p(\mathbf{x},t), \tag{3}$$

where $\mathbf{x}_{i\cdot}$ denotes the $i$th row and $\mathbf{x}_{\cdot j}$ denotes the $j$th column of the $K \times S$ state matrix $x$, where $S$ is the number of chemical species. The functions $a_{ir}(x_i)$ define the propensity functions of the $M$ chemical reactions, and $\nu_{ir}$ are stoichiometry vectors associated with the reactions. The propensity functions are defined such that $a_{ir}(\mathbf{x})\Delta t$ gives the probability that reaction $r$ occurs in a small time interval of length $\Delta t$. The stoichiometry vector $\nu_{ir}$ defines the rules for how the state changes when reaction $r$ is executed. $d_{ijk}(x_i)$ are propensities for the diffusion jump events, and $\mu_{ijk}$ are stoichiometry vectors for diffusion events. $\mu_{ijk}$ has only two non-zero entries, corresponding to the removal of one molecule of species $x_k$ in voxel $i$ and the addition of a molecule in voxel $j$. The propensity functions for the diffusion jumps, $d_{ijk}$, are selected to provide a consistent and local discretization of the diffusion equation, or equivalently the Fokker-Planck equation for Brownian motion. It is important to note, as mentioned in Section I, that this formalism is defined for a physical domain that is static in time. We will relax this assumption with our method to accommodate time-dependent domains.

### B. The next subvolume method

In most cases, the RDME is too high-dimensional to solve directly. Thus, algorithms have been developed that generate exact realizations of the Markov process described by the RDME, in a Monte Carlo fashion. One particularly efficient algorithm that we focus on for our implementation is the Next Subvolume Method (NSM).[15] In this algorithm, the time to the next event in each voxel (either a chemical reaction or diffusion event) calculated by the Direct Method formulation of the SSA.[28] To identify in which voxel the event occurs, the algorithm uses the Next Reaction Method formulation of the SSA.[29] If it was a chemical reaction event that occurred, then only the voxel in which the event occurred needs to be updated, while if a diffusion event occurs both the voxel where the molecule started and the voxel where the molecule ended up need to be updated. The key to the efficiency of the NSM is the use of an event priority queue which gives a scaling of $\mathcal{O}(\log_2(K))$, where $K$ is the number of voxels in the mesh.

### C. RDME on unstructured meshes

The use of unstructured meshes allows for complicated geometries in 3D to be more easily accommodated, such as the curved surfaces of cell membranes. PyURDME[30] (based on URDME[25]) is a software framework for simulation of the RDME on unstructured meshes that we extend to time-dependent domains. For the theoretical details of how to obtain mesoscopic diffusion constants on unstructured meshes, see Ref. 31. Using the finite element package DOLFIN[32] we obtain the diffusion matrix for the system, from which we get the jump coefficients for individual voxels. The flexibility of simulating on unstructured meshes allows our method to handle complex time-dependent domains in 3D.

## III. COMPUTATIONAL METHOD FOR SPATIAL STOCHASTIC SIMULATION WITH A MOVING BOUNDARY

In this section we develop a computational method for simulation of spatial stochastic systems defined by the RDME formalism on domains with moving boundaries. Our method utilizes the time scale separation between the diffusion of the biochemical species in the RDME system and the movement of the boundary of the domain. The method is formulated for systems where diffusion is faster than the boundary movement. In this context we use operator splitting to decouple the reaction-diffusion operator from the domain movement operator, solving each operator sequentially over the same time step.

In the formulation of the RDME, the spatial domain is discretized into volume units known as voxels. For example, in our PyURDME software package[30] the 3D domain is discretized using tetrahedral elements. Since the RDME is formulated to be solved on a static domain, we cannot directly adapt it to a moving domain. Instead, we employ an operator splitting style simulation, by first solving the RDME for a small time $\tau$ (the splitting time step) on a static mesh. Then we use the state of the biochemistry to find the velocity of the boundary of the domain through a function specified by the user, and evolve the mesh over the same time step $\tau$. Finally, the state of the biochemical system is transformed to the newly evolved mesh. A sequence of these steps are taken until the simulation reaches the final time.

The algorithm for moving the mesh has four components. Figure 2 illustrates these components and the process flow between them. The first component is the simulation of the biochemical system for a time $\tau$ on a specific mesh (denoted as $\Omega_a$) starting from an initial state $x_a(t)$ to a final state $\mathbf{x}_a(t+\tau)$,

$$\mathbf{x}_a(t+\tau) = \mathrm{RDME}(\mathbf{x}_a(t), \Omega_a, \tau). \tag{4}$$

The second component is the computation of the velocity field $v$ at the boundary of the domain, as a function the state of the biochemical system. In our motivating example of the growth of the yeast mating projection, the internal turgor pressure pushes uniformly on the cell wall, but the wall expands preferentially where it has been softened by the enzymes. Thus, the instantaneous velocity field of the growing mating projection is a function of the spatial distribution of the cell wall modifying enzymes within cell. See Section IV C for more details, and Figure 7 for an illustration.

This is the component that couples the biochemical simulation with the moving domain. This function is a part of the

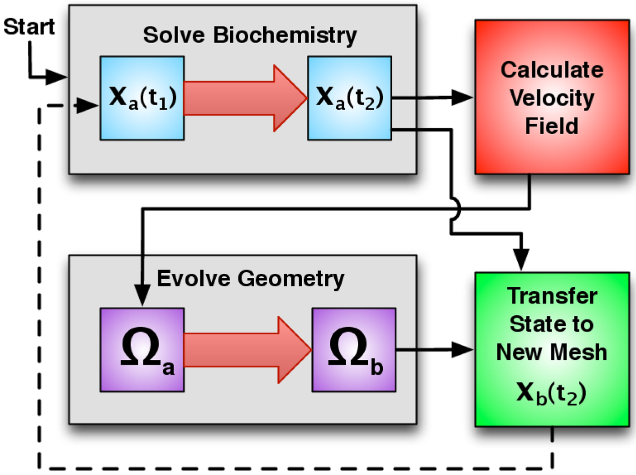

FIG. 2. Diagram illustrating the process flow of the moving mesh algorithm.

model being simulated, and is thus provided by the user as input to the method. The function can be of only the final state of the biochemical system, $\mathbf{x}_a(t+\tau)$, or of some aggregate of all previous states, $\mathbf{x}_a(0:t+\tau)$,

$$v = \left.\frac{d\Omega}{dt}\right|_{t+\tau} = f_{\Delta\Omega}(\mathbf{x}_a(0:t+\tau)). \quad (5)$$

The third component is the evolution of the mesh over the simulation time step $\tau$. Using the domain boundary velocity calculated previously, a new mesh is created by evolving the mesh linearly over the time step. This is done by moving each mesh point according to the velocity field at that point, via the forward Euler method. In our software implementation, this is done via the FEniCS/Dolfin package,[32]

$$\Omega_b = \Omega_a + \tau\, v. \quad (6)$$

The final component of the algorithm is the method for transferring the state of the biochemical system from the current mesh, $\Omega_a$, to the newly created mesh, $\Omega_b$. On each step of the moving mesh simulation the $x$, $y$, $z$ position of each particle is sampled on $\Omega_a$. An assumption of the RDME is that particles are uniformly distributed within each voxel. Consequently, the position of the particle is sampled uniformly from its containing voxel's volume. Since the boundaries of a voxel are often difficult to compute on an unstructured mesh, we make an approximation and sample the position from a sphere with a volume equivalent to that of the containing voxel. It is important to note that this assumption induces a spatial error that is proportional to the mesh resolution and the quality of the mesh. That is, the more elongated the tetrahedrons are, the more error is induced in the sampled particle's spatial position. Implementations of this algorithm must ensure that the mesh is of sufficient quality throughout the simulation. Next, the particle is assigned to the closest voxel new mesh, $\Omega_b$ (minimizing Euclidian distance), to the sampled $x$, $y$, $z$ position. Often in systems biology models, biochemical species are required to remain in specific subdomains of the system. For example, membrane bound proteins must remain on the membrane, which is modeled as the voxels on the boundary of the mesh. If the species of a particle is restricted to a subdomain in this way, then it is moved to the closest voxel that is within that subdomain. If the sampled position of a particle falls outside the domain $\Omega_b$, then it is placed at the closest voxel (that is of an appropriate subdomain) within $\Omega_b$. See Figure 3 for an illustration. This procedure is repeated for each particle within the system, thus the biochemical state of the system is transferred from $\Omega_a$ to $\Omega_b$, which we denote as

$$\mathbf{x}_b(t) = \text{ParticleRedistribution}(\mathbf{x}_a(t), \Omega_a, \Omega_b). \quad (7)$$

The iterative algorithm is described in Algorithm 1.

### A. Rejection-based step size selection

We extend the method presented in Algorithm 1 to include an adaptive method for error control. We will see in Example 2 that the error our method incurs will depend on variables of the velocity of the domain and the operator splitting time step. To implement our adaptive error control time stepping scheme, we define a new input $d_{max}$ as the maximum distance any given point on the boundary is allowed to move in any single time step. In each step, if the magnitude of any component of the velocity multiplied by the time step $\tau$ is greater than $d_{max}$, then that step is rejected, the time step is set to half its previous value, and the state is recomputed over the new time step. The accepted time step will then be used for the next step of the algorithm. Finally, if the step is accepted on the first pass (no rejection) and the time step had been previously reduced ($\tau$ is less than $\tau_{split}$), then time step for the subsequent step, $\tau_{next}$, is increased to double the current time step size. See Algorithm 2 for details. Note that, as the simulation of the RDME by the NSM algorithm is a continuous operator, we are able to sample at any specified time within the interval $[t, t+\tau]$. This allows us to avoid recomputing the biochemical state when a step is rejected, leading to a more efficient implementation. In our simulations, for each invocation of the NSM operator, we sample the state of the RDME at $[t+\frac{\tau}{8}, t+\frac{\tau}{4}, t+\frac{\tau}{2}, t+\tau]$. This allows us to half the time step three times without

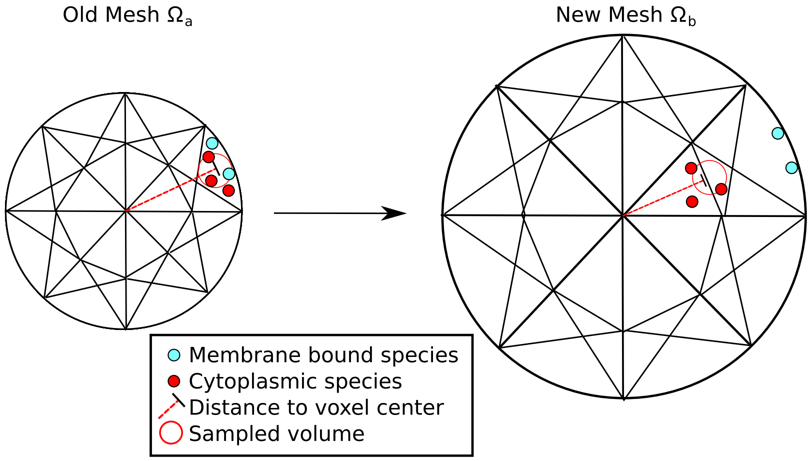

FIG. 3. Diagram illustrating the particle redistribution process. On each step of the moving mesh simulation, the state of the biochemical system is transferred from the old mesh (left) to the new mesh (right). The $x$, $y$, $z$ position of each particle is sampled on the old mesh (uniformly from within the volume of the containing voxel); the particle is assigned to the voxel in the new mesh that is closest to that sampled position. If the species of a particle is restricted to a subdomain (e.g., membrane-bound proteins), then it is moved to the closest voxel in that subdomain.

ALGORITHM I. Spatial stochastic simulation for a domain with a moving boundary.

```
Input: $\Omega_0, f_{\Delta\Omega}, \tau_{split}, x_0(0), t_{final}$, and the Biochemical Reaction Network
Output: $[\Omega_0 \cdots \Omega_n]$, $[\mathbf{x}_0 \cdots \mathbf{x}_n]$
$i = 0, \quad t = 0, \quad \tau = \tau_{split}$
while t $< t_{final}$ do
    $\mathbf{x}_i(t : t+\tau) = \mathrm{RDME}(\mathbf{x}_i(t), \Omega_i, \tau)$
    $v = f_{\Delta\Omega}(\mathbf{x}_i(t : t+\tau))$
    $\Omega_{i+1} = v\,\tau + \Omega_i$
    $\mathbf{x}_{i+1}(t+\tau) = \mathrm{ParticleRedistribution}(\mathbf{x}_i(t+\tau), \Omega_i, \Omega_{i+1})$
    $i = i$+1, $\quad t = t + \tau$
end while
```

ALGORITHM II. Adaptive spatial stochastic simulation for a moving boundary domain.

```
Input: $\Omega_0, f_{\Delta\Omega}, d_{max}, \tau_{split}, X_0(0), t_{final}$, and the Biochemical Reaction Network
Output: $[\Omega_0 \cdots \Omega_n]$, $[\mathbf{x}_0 \cdots \mathbf{x}_n]$
$i = 0, \quad t = 0, \quad \tau = \tau_{split}$
while t $< t_{final}$ do
    $\mathbf{x}_i(t : t+\tau) = \mathrm{RDME}(\mathbf{x}_i(t), \Omega_i, \tau)$
    $v = f_{\Delta\Omega}(\mathbf{x}_i(t : t+\tau))$
    $d = ||v||_\infty \tau$
    If $d > d_{max}$ then
        repeat
            $\tau = \tau/2$
            $v = f_{\Delta\Omega}(\mathbf{x}_i(t : t+\tau))$
            $d = ||v||_\infty \tau$
            $\tau_{next} = \tau$
        Until $d \le d_{max}$
    else if $\tau < \tau_{split}$ then
        $\tau_{next} = 2\tau$
    end if
    $\Omega_{i+1} = v\,\tau + \Omega_i$
    $\mathbf{x}_{i+1}(t+\tau) = \mathrm{ParticleRedistribution}(\mathbf{x}_i(t+\tau), \Omega_i, \Omega_{i+1})$
    $i = i$+1, $\quad t = t + \tau, \quad \tau = \tau_{next}$
end while
```

recomputation of the biochemical system. It should be noted that to avoid biased simulations, the state of the random number generator must be preserved when the step is rejected and the step restarted with the same state.

## IV. RESULTS

Here we present three examples to verify and show the utility of our method.

### A. Example 1

In our first example, we demonstrate numerically that our method converges in distribution as the time step decreases. In a general problem the error depends on multiple factors, such as the time step, the mesh resolution, and the quality of the mesh. To isolate the error induced by the time step selection, we consider a 1D domain $\Omega$ of width $R - L$, where $R$ is the right endpoint and $L$ the left endpoint. A single species $S$ diffuses (with $D = 1$), associates with, and dissociates from the left boundary. We let $\Omega$ expand to the left; thus $L$ is a function of time $t$. Specifically we let

$$L(t) = -vt,$$

where $v$ is the constant speed of the expansion.

The domain is discretized into $N_{\text{vox}}$ voxels, each of width $h$. We denote the microscopic association rate by $k_r$ and the microscopic dissociation rate by $k_d$. The mesoscopic rates, $k_a^{\text{meso}}$ and $k_d^{\text{meso}}$, are then given by $k_a^{\text{meso}} = k_a/h$ and $k_d^{\text{meso}} = k_d$, as shown in Appendix C. We let $L = 0$ and $R = 1$ initially.

To show that our method is accurate we simulate the system until the final time $T = 1$, and compare the spatial distribution of particles to the spatial distribution obtained with a more detailed Brownian dynamics method.[10] The error will be a function of the speed of expansion $v$, the time step $\Delta t_{\text{split}}$, the number of voxels $N_{\text{vox}}$, and the reaction rates $k_r$ and $k_d$. We expect the error to be larger for a large $v$, as the boundary moves more during each time step. For small enough $N_{\text{vox}}$ the spatial resolution will be insufficient, and the error will consequently be large. To demonstrate these effects, we ran simulations with $\Delta t_{\text{split}}$ varying from 0.01 to 0.2 with $N_{\text{vox}} \in \{5, 20, 50\}$.

We simulated $10^6$ molecules and computed the Kolmogorov-Smirnov distance between the spatial distributions of unbound particles at the final time point $T$ (with the domain expanding at constant velocity, convergence at the final time point implies convergence throughout). In Figure 4 we show that, as expected, the error decreases with decreasing time step $\Delta t_{\text{split}}$.

### B. Example 2: Density dependent switch for polarization in an expanding and contracting sphere

In our second example, we verify the accuracy of our method by comparing against the analytical solution of a biochemical model found in the literature. To demonstrate the applicability of our method to biologically relevant problems, we have implemented a simple model of polarization in yeast on a moving domain. In particular, we focus on a model of polarization presented in Ref. 26 that relies on a minimal positive feedback circuit. This model is particularly interesting as it has been shown to polarize only when modeled stochastically, opposed to deterministically. The yeast cell is modeled as a sphere with a membrane on the surface of the sphere. The model has three reactions between two species: cytosolic Cdc42 spontaneously attaches to the membrane with rate $k_{on}$ (Eq. (8)), membrane-bound Cdc42 likewise spontaneously detaches with rate $k_{off}$ (Eq. (9)), and finally membrane-bound Cdc42 recruits cytosolic Cdc42 to the membrane at rate $k_{fb}$, to close the positive feedback loop (Eq. (10)),

$$Cdc42_c \xrightarrow{k_{on}} Cdc42_m, \tag{8}$$

$$Cdc42_m \xrightarrow{k_{off}} Cdc42_c, \tag{9}$$

$$Cdc42_c + Cdc42_m \xrightarrow{k_{fb}} 2Cdc42_m. \tag{10}$$

The cytosolic and membrane-bound species can diffuse at rates $D_{cyt}$ and $D_{mem}$, respectively, (the diffusion of the membrane-bound Cdc42 being restricted to the membrane). This model was shown in Ref. 26 to have a density dependent switch. That

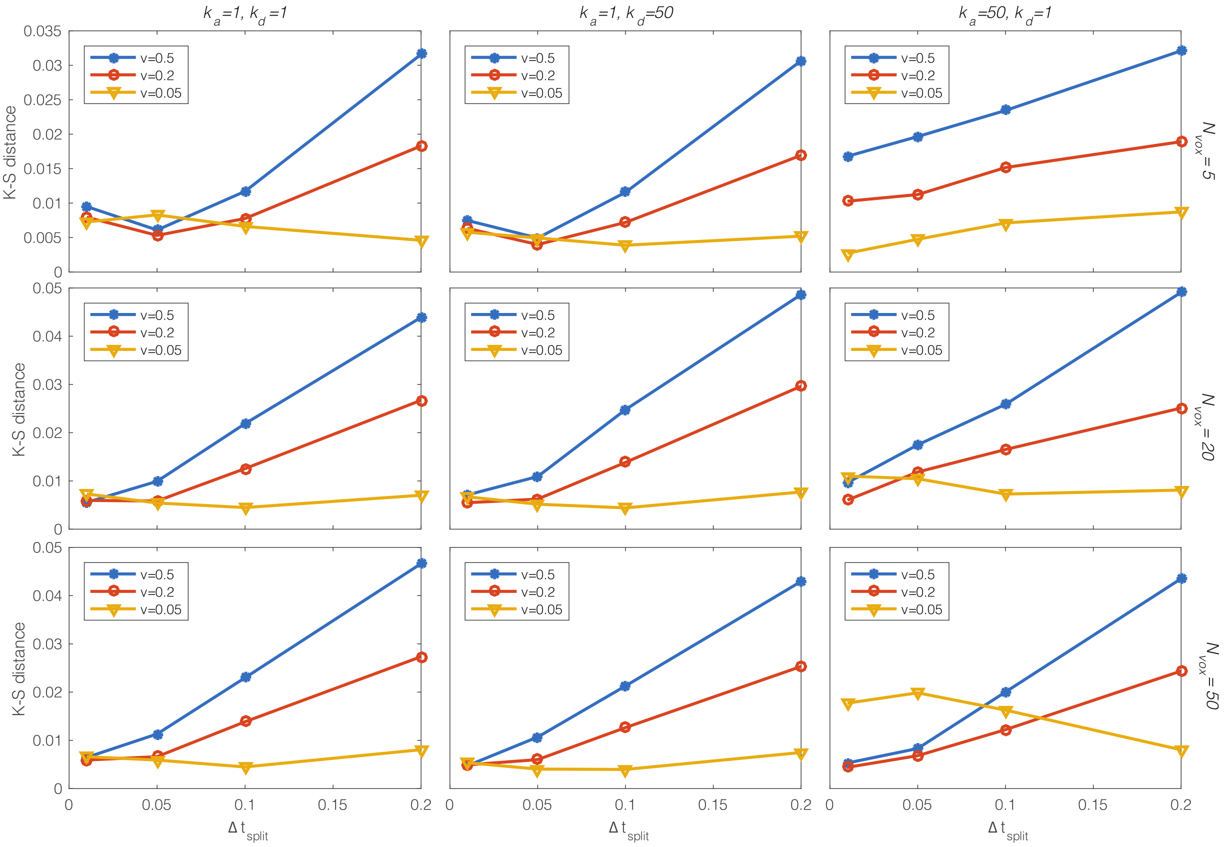

FIG. 4. Error in distribution as a function of the size of the time step for a 1D moving domain with an absorbing/desorbing boundary for three different reactive rates (columns) and three different spatial discretizations (rows). Our mesoscopic method was compared with a Brownian dynamics microscopic simulation of $10^6$ molecules. We define the error as the Kolmogorov-Smirnov distance between the spatial distributions of unbound particles at the end time (1s). As we can see by comparing the middle three panels to the bottom three panels, the error is similar to $N_{vox}$ = 20 and $N_{vox}$ = 50, meaning that the problem is spatially well-resolved already with $N_{vox}$ = 20, while for $N_{vox}$ = 5 we can see that the system is not fully resolved for the case of $k_a$ = 50 and $k_d$ = 1.0. As expected, the larger the speed $v$ of the expansion, the larger the error, but as the splitting time step $\Delta t_{\text{split}}$ decreases, so does the error. For $v$ = 0.05, the domain is expanding so slowly that the stochastic error dominates, and no difference is seen as $\Delta t_{\text{split}}$ varies between 0.01 and 0.2.

is, there is a critical range for polarization of molecules on the membrane. This range is from a lower critical density necessary to facilitate polarization to an upper density above which molecules become essentially homogeneous on the membrane (i.e., not polarized). From Ref. 26 it is also possible to calculate theoretically the steady state ratio of molecules in the cytoplasm for any given density, which we will use as a comparison for our simulations.

To test our moving mesh algorithm, we implemented the model described above in an expanding and contracting sphere for a fixed number of total molecules. For the expanding sphere case, the initial radius was set below the theoretical switch value calculated from Ref. 26. Specifically, the critical radius can be calculated as follows:

$$\frac{4}{3}\pi r_{crit}^3 = N\frac{k_{fb}}{k_{off}}, \qquad (11)$$

where $N$ is the total number of molecules (cytosolic and membrane-bound) and $r_{crit}$ is the critical radius. Here we set $N$ = 1000 and $\frac{k_{fb}}{k_{off}}$ = 0.9, thus from Eq. 11 we have $r_{crit}$ = 6.425 $\mu m$. From the initial radius (below the critical radius), the radius of the sphere expands at a constant velocity to a final value which is greater than the critical radius. The error incurred by our operator split method is dependent on both the speed at which the sphere expands and the operator split time step that is chosen. Again, the error here is defined as the relative error between the number of cytosolic molecules in the simulation and the theoretical steady state value calculated from Ref. 26. We calculated this error over a range of expansion velocities and operator split time steps to investigate the convergence behavior for our method. A similar test was performed for a contracting sphere which starts at a radius just below the critical value and decreases at a constant velocity to some final radius.The results of these convergence studies are shown in Fig. 5. As expected, the larger the velocity of radial expansion, the more error the method will generate for both the expanding (Fig. 5(a)) and contracting (Fig. 5(b)) sphere cases. The same trend can be seen for large time steps (Figs. 5(a) and 5(b)). Note that for each parameter value multiple realizations were run and averaged before being compared to the theoretical value. The 95% confidence interval for three realizations of the number of cytosolic molecules at each parameter value is shown for the expanding sphere in Fig. 5(c) and for the contracting sphere in Fig. 5(d). An example time series of heatmaps showing the number of membrane bound molecules on the surface of an expanding sphere can be seen in Fig. 6(a), with red being a higher number of molecules and blue a lower value. Also, a sample trajectory of the expanding sphere simulation can be seen in comparison to the theoretical steady

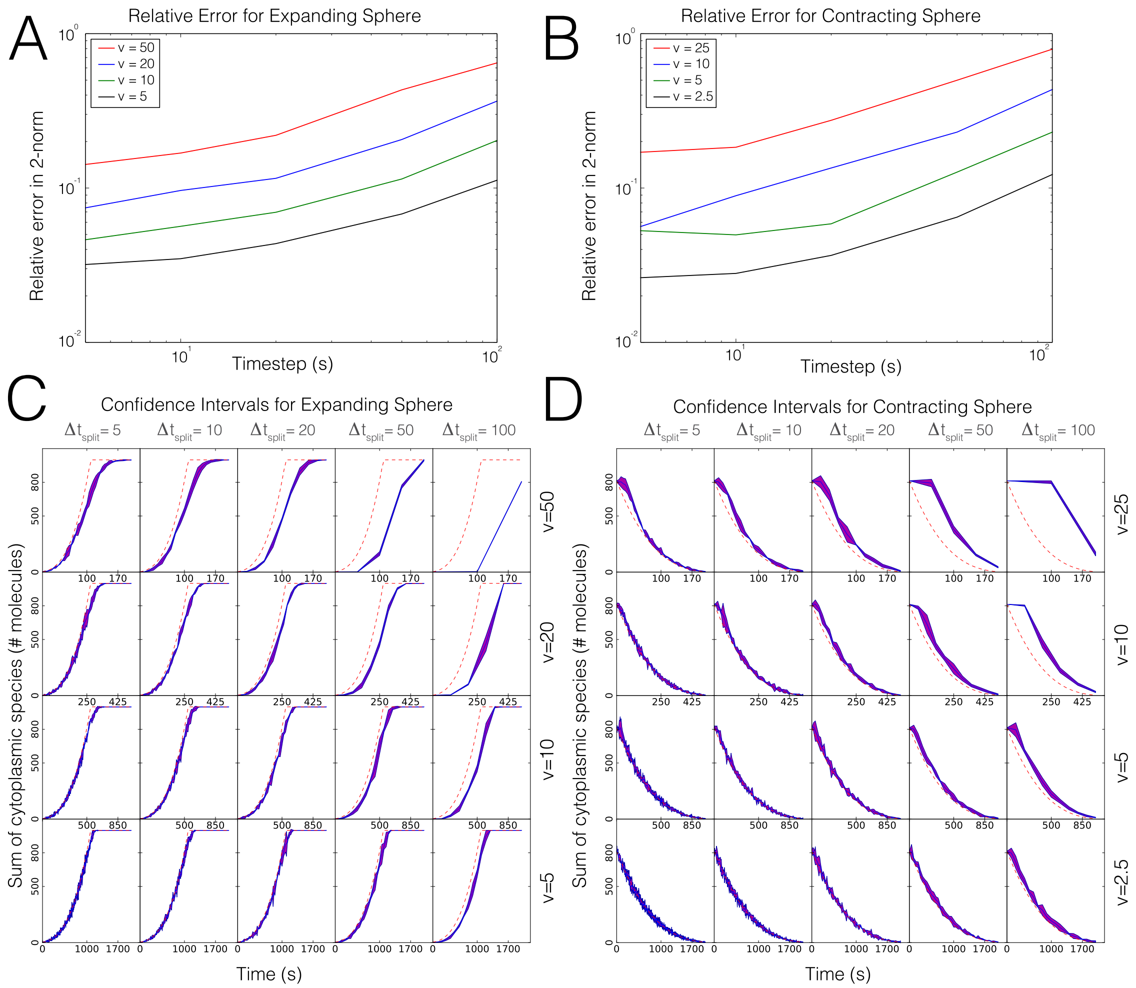

FIG. 5. Comparison of our method to theoretical results from the literature. As expected, the error for our method decreases with the velocity of the boundary and the time step. (a) Relative error in 2-norm for a variety of expansion velocities (v) and operator-split time steps for an expanding sphere. (b) Relative error in 2-norm for a variety of contraction velocities (v) and operator-split time steps for a contracting sphere. (c) 95% confidence intervals for three trajectories at a variety of expansion velocities (v) and operator-split time steps ($\Delta t_{split}$) for an expanding sphere, along with the theoretical value (dashed red line). (d) 95% confidence intervals for three trajectories at a variety of contraction velocities (v) and operator-split time steps ($\Delta t_{split}$) for a contracting sphere, along with the theoretical value (dashed red line).

state solution in Fig. 6(b) (here with parameter values of $\frac{dr}{dt}$ = 10 nm/s and $t_{split}$ = 20 s). Again, the purpose of this example problem is to characterize the error incurred by our method as compared to theoretical results in the literature and specifically to elucidate how this error depends on the velocity of the boundary and the time step. This example also demonstrates the utility of our method for studying problems of biological relevance.

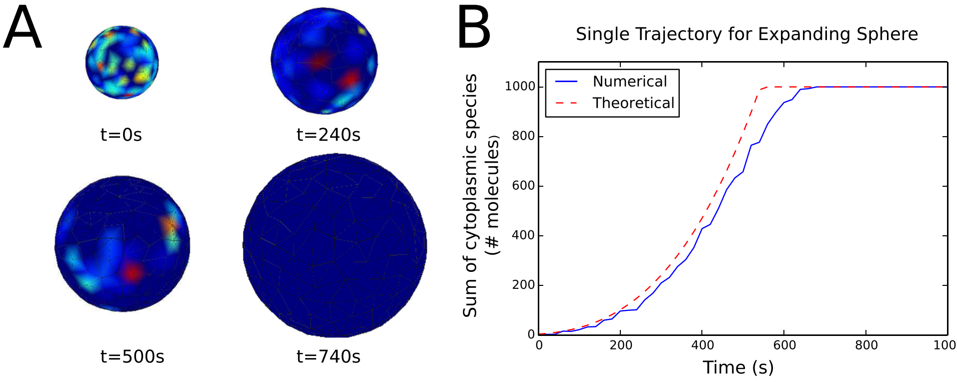

FIG. 6. (a) Time series of heatmaps showing the number of membrane-bound molecules on the surface of an expanding sphere (note that the radii are not to scale and by 740 s there are no molecules on the membrane). (b) An example of a single trajectory showing the number of cytoplasmic molecules versus time for an expanding sphere along with the theoretical value as calculated from Ref. 26. Note that for each parameter value multiple realizations were run and averaged before being compared to the theoretical value.

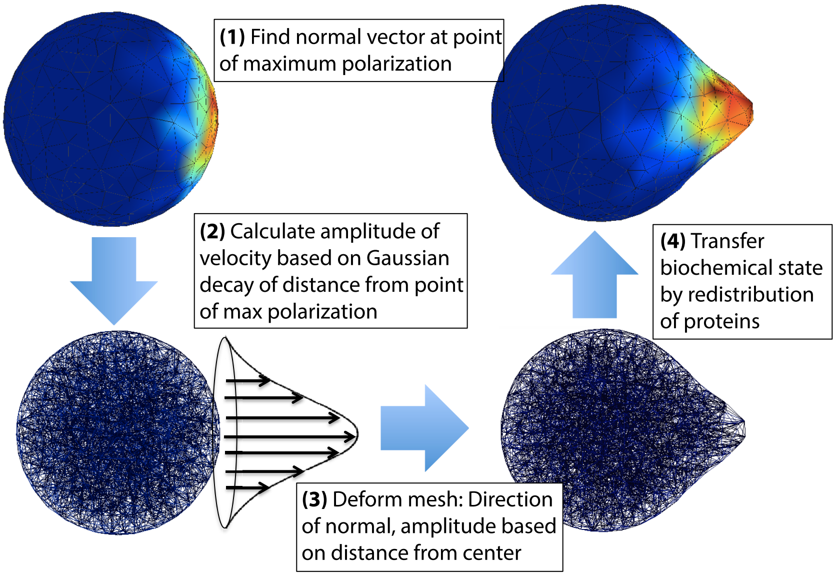
**(1)** Find normal vector at point of maximum polarization
**(2)** Calculate amplitude of velocity based on Gaussian decay of distance from point of max polarization
**(3)** Deform mesh: Direction of normal, amplitude based on distance from center
**(4)** Transfer biochemical state by redistribution of proteins

FIG. 7. Diagram describing the process where polarization of the yeast model (spatial psrofile of protein concentration) is used to calculate the deformation of the domain. At each time step of the algorithm the following process is repeated. First the biochemical system is simulated using the spatial stochastic solvers in PyURDME for a length of Δt. Next the point of maximum polarization is found and the normal vector is calculated at that point. Then, the velocity of the surface is calculated using the normal vector as the direction, with the amplitude calculated from a Gaussian function centered at the maximum polarization point (empirically parameterized). The mesh is then deformed by the application of the velocity field. Finally, the biochemical state of the system is transferred to the new mesh.

## C. Example 3: Formation of the yeast mating projection

In our third example, we show the use of our method on a biologically relevant system that couples the biochemical reactions with the moving boundary. Our motivating example is the polarization of proteins during mating of *Saccharomyces cerevisiae*, and the resulting growth of the mating projection.

Yeast cells sense mating pheromone in their extra-cellular environment, and determine the direction of their mating partner by sensing the chemical gradient. The chemical gradient of pheromone induces polarization of the yeast cell, localizing proteins, and actin cables to the region of the yeast cell that is closest to a nearby mating partner. The mating projection starts to form when the polarisome organelle is formed at the site of polarization. The polarisome acts to coordinate the formation of the mating projection via the transport of cell wall cutting enzymes as well as cell wall material and synthase proteins. As these processes work together, the yeast cell changes shape from a spheroid to grow a projection.

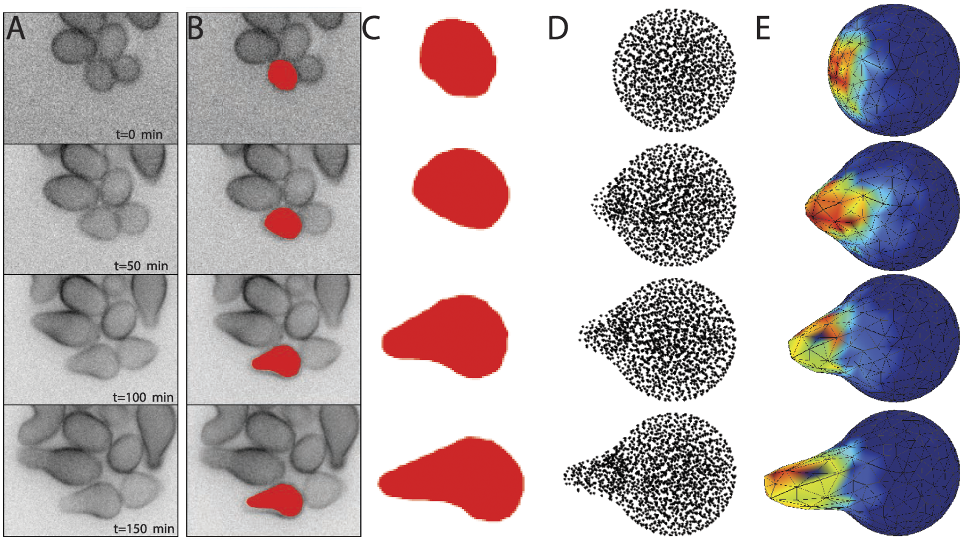

FIG. 8. Comparison between microscopy images of yeast cells during polarized growth and simulations of the growing yeast mating projection. (a) Fluorescent microscopy time-lapse images of yeast cells during exposure to mating pheromone ($\alpha$-factor). Cells are tagged with Mid2-GFP (see Appendix D for experimental details). (b) Manual cell shape extraction overlaid on microscopy images for a single cell. (c) Enlarged cell shape without microscopy image. (d) Scatter plot of the voxel centers of our simulation results projected onto a plane containing the origin and the point of greatest polarization. (e) 3D visualization of the simulated growing yeast cell, where the color map shows the concentration of active Cdc42 on the membrane (red corresponds to the highest concentration, and blue to the least).

To simulate this process we present a new spatial stochastic biochemical model of yeast polarization, centered around the polarization of the protein Cdc42. This model is a novel combination of reactions published in Refs. 33 and 34. As these models were originally presented deterministically, for use with this method we have converted the reactions to a mechanistic and stochastic formulation (see Appendix A for a complete description of the model). To present a simplified model of polarization, we omit the dynamics of the receptor-ligand binding (presented elsewhere[35,36]) and take as input to the model a time-constant spatially varying concentration of the activated G-protein (beta-gamma subunit). The biochemistry determines the moving boundary by expanding the sphere at the point of greatest polarization, in the direction of the normal at that location. The other points on the boundary of the domain are moved in a parallel direction, with the magnitude attenuated by a Gaussian of the distance to the point of maximum polarization. This is illustrated in Figure 7.

Figure 8 shows a comparison between our yeast polarization simulation results (columns (d) and (e)) and microscopy images of a polarizing yeast cell (columns (a)-(c)). Column (b) shows the outline of the yeast cells overlaid on the microscopy images, and column (c) shows just the outline of these cells. Column (d) shows a scatter plot of the voxel centers of our simulation results, projected onto a plane containing the origin and the point of greatest polarization. Column (e) shows the 3D visualization of the simulated growing yeast cell, where the color map shows the concentration of active Cdc42 on the membrane (red corresponds to the highest concentration, and blue to the least). The frames (rows) in column (a)-(c) are separated by 50 min intervals. In this simulation, the state of the biochemical system and the movement of the boundary are fully coupled. This figure shows a qualitative match between a real cell phenotype and our biochemical model simulated via our method.

## V. CONCLUSIONS

We have developed a method for simulating stochastic biochemical reactions on time-dependent domains using the RDME formalism. This method involves the following steps: simulating the RDME on a fixed geometry for a given time step, using the state of the biochemical system as input to a function that moves the boundary in a user-specified manner, over that same time step, redistributing the molecules in the system to the new geometry, and then repeating until a specified end time. Our method simulates the RDME on unstructured meshes, which allow it to easily handle the complex geometries that often show up in biological applications. We have shown, through various example problems, that the error our method will incur depends on a few key factors, including the specified time step, the speed of the moving domain, the diffusion constants of the species, and the reaction rates of the system. We have also demonstrated the potential usefulness of such a method by simulating the biologically relevant problem of shmoo formation during the mating of yeast, a problem where spatial stochastic effects are important and the geometry is changing in time as a result of the state of the biochemical system.

We have implemented this method in our spatial stochastic modeling and simulation software package PyURDME:[30] the Python package for simulation of Unstructured mesh Reaction-Diffusion Master Equation models. The reaction-diffusion biochemical model system has been extended to allow the inclusion of a movement of the mesh, and the inspection of the resulting mesh quality and adaptive mesh refinement are accomplished via integration of the open source finite element package FEniCS/Dolfin.[32] The software package, along with instructive examples, is available from our code repositories on Github.

This method is generally applicable to problems arising in systems biology where spatial and stochastic effects are critical and the physical geometry is changing in time. In particular, this method is applicable to the common case in systems biology where the movement of the boundary of a cell is directly determined by the state of certain biochemical species. The error will be more manageable for systems where there is some separation of time scales between the movement of the boundary and the diffusion rate of the biochemical system. In the future, we hope to extend our analysis of biologically relevant problems with coupled biochemistry and domain movement, such as in shmoo formation in yeast mating. Other specific systems where this method could be useful include the following: tip growth in fungal hyphae,[18] chemotaxis in neutrophils,[6] cell migration,[19] and cell division.[7,8] Another future direction is to develop a method that more closely couples the dynamics of the moving boundary and the biochemistry, thus avoiding the error involved in splitting the two, but this is a considerably more involved problem.

## ACKNOWLEDGMENTS

We would like to thank Professor Otger Campás for his helpful discussions. This work has been funded by National Institutes of Health (NIH) NIGMS Award No. R01-GM113241 and NIBIB Award No. R01-EB014877, Department of Energy (DOE) Award No. DE-SC0008975, Institute for Collaborative Biotechnologies (ICB) Award No. W911NF-09-0001 from the U.S. Army Research Office, and National Science Foundation (NSF) Award No. DMS-1001012. The content of this paper is solely the responsibility of the authors and does not necessarily represent the official views of these agencies.

## APPENDIX A: MODEL OF Cdc42 POLARIZATION IN YEAST

The model for the of *Cdc*42 used in Section IV C is a combination of reactions published in Refs. 33 and 34. The key component from Ref. 33 is what is known as a GDI reaction which preferentially moves the inactive form of *Cdc*42 from the membrane to the cytosol. Diffusion is much faster in the cytosol than on the membrane and this difference in diffusion is what leads to a Turing-type mechanism for pattern formation, as originally discussed in Ref. 37. The essential components from Ref. 34 are a negative feedback mechanism through the protein *Cla*4 and recruitment of *Cdc*24 to the membrane by *Gbg* (the $\beta\gamma$ subunit of the heterotrimeric G-protein which is activated in response to pheremone during yeast mating). Both Refs. 33 and 34 originally modeled their respective reactions using deterministic PDE models. Here we take certain key reactions from both models and model them mechanistically using the RDME formalism, which results in the following reactions:

$$
\begin{aligned}
Cdc24_c + Gbg_m &\xrightarrow{k_{on}} Cdc24_m + Gbg_m,\\
Cdc24_m + Cdc42GDP_m &\xrightarrow{\alpha_1} Cdc24_m + Cdc42GTP_m,\\
Cdc42GTP_m &\xrightarrow{\alpha_3} Cdc42GDP_m,\\
Cdc24_m + Cdc42GDP_c &\xrightarrow{\beta_1} Cdc24_m + Cdc42GTP_m,\\
Cdc42GDP_c &\xrightarrow{\beta_2} Cdc42GDP_m,\\
Cdc42GDP_m &\xrightarrow{\beta_3} Cdc42GDP_c,\\
Bem1_c + Cdc42GTP_m, &\xrightarrow{\gamma_1} Bem1_m + Cdc42GTP_m,\\
Bem1_m &\xrightarrow{\gamma_2} Bem1_c\\
Bem1_m + Cdc24_c &\xrightarrow{\delta_1} Bem1_m + Cdc24_m,\\
Cdc24_m &\xrightarrow{\delta_2} Cdc24_c,\\
Cdc42GTP_m + Cla4_m &\xrightarrow{k_{Cla4a}} Cdc42GTP_m + Cla4_{am},\\
Cdc24_m + Cla4_{am} &\xrightarrow{k_{24d}} Cdc24_c + Cla4_{am},\\
Cla4_{am} &\xrightarrow{k_{Cla4d}} Cla4_m
\end{aligned}
$$

TABLE I. Parameters for the Cdc42 yeast polarization model.

| Parameter | Value | Description | Reference |
|---|---|---|---|
| $D_m$ | 0.0053 $\mu$m$^2$ s$^{-1}$ | Diffusion constant on membrane | 38 |
| $D_c$ | 50 $\mu$m$^2$ s$^{-1}$ | Diffusion constant in cytoplasm | 39 |
| $R$ | 2.00 $\mu$m | Reference radius of cell | 34 and 40 |
| $N_{42}$ | 3000 | Total number of *Cdc*42 molecules | 34 and 40 |
| $N_B$ | 3000 | Total number of *Bem*1 molecules | 34 and 40 |
| $N_{24}$ | 1000 | Total number of *Cdc*24 molecules | 34 and 40 |
| $N_{Cla4}$ | 5000 | Total number of *Cla*4 molecules | 34 and 40 |
| $N_{Gbg}$ | 5000 | Total number of *Gbg* molecules | 34 and 40 |
| $\alpha_1$ | 0.2 $\mu$m$^2$ s$^{-1}$ | *Cdc*24 activating $Cdc42_m$ | 33 |
| $\alpha_3$ | 1 $s^{-1}$ | Deactivation of *Cdc*42 | 33 |
| $\beta_1$ | 0.266 $\mu$m$^3$ s$^{-1}$ | *Cdc*24 activating $Cdc42_c$ | 33 |
| $\beta_2$ | 0.28 $\mu$m s$^{-1}$ | Attachment of $Cdc42_c$ | 33 |
| $\beta_3$ | 1 $s^{-1}$ | Detachment of $Cdc42_m$ | 33 |
| $\gamma_1$ | 0.2667 $\mu$m$^3$ s$^{-1}$ | Active *Cdc*42 recruit $Bem1_c$ | 33 |
| $\gamma_2$ | 0.35 $s^{-1}$ | Detachment of $Bem1_m$ | 33 |
| $\delta_1$ | 0.00297 $\mu$m$^3$ s$^{-1}$ | $Bem1_m$ recruit $Cdc24_c$ | 33 |
| $\delta_2$ | 0.35 $s^{-1}$ | Detachment of $Cdc24_m$ | 33 |
| $k_{on}$ | 0.00297 $\mu$m$^3$ s$^{-1}$ | Gbg recruit $Cdc24_c$ | 33 |
| $k_{Cla4a}$ | 0.006 $s^{-1}$ | Active *Cdc*42 recruit *Cla*4 | 34 and 40 |
| $k_{Cla4d}$ | 0.01 $s^{-1}$ | Negative feedback via *Cla*4 | 34 and 40 |
| $k_{24d}$ | 1.0/30 000 $s^{-1}$ | Deactivation of *Cla*4 | 34 and 40 |

where the subscripts $c$ and $m$ denote cytoplasm and membrane-bound species, respectively. *Cdc*42*GTP* and *Cdc*42*GDP* are the active and inactive forms of *Cdc*42 on the membrane respectively. $Cla4_{am}$ and $Cla4_m$ are the active and inactive forms of *Cla*4 on the membrane respectively. The physical domain starts as a sphere with a radius of 2.0 $\mu$m with a membrane on the surface of the sphere and evolves according to the state of the biochemical system. Diffusion of membrane bound species is restricted to the membrane. *Gbg* is taken to be a constant input function with a uniform concentration throughout the domain (i.e., it is not an explicit species and thus does not diffuse). The parameter values, descriptions, and references used in Section IV C are given in Table I.

## APPENDIX B: MICROSCALE SIMULATIONS

On the microscopic scale we track the continuous position of individual particles, in contrast to the mesoscopic scale on which we track only the copy number of particles in each voxel. Molecules move by normal diffusion and are modeled by hard spheres with reaction radius $\sigma$. The reaction dynamics as two molecules collide is governed by the Smoluchowski equation[41] with a Robin boundary condition.

Consider two molecules, $S_1$ and $S_2$, with diffusion rates $D_1$ and $D_2$ and reaction radii $\sigma_1$ and $\sigma_2$. Let $D = D_1 + D_2$ and $\sigma = \sigma_1 + \sigma_2$. At time $t_n$, the positions are given by $\mathbf{x}_1^n$ and $\mathbf{x}_2^n$. We let $p(x_1, x_2, t|x_{1n}, x_{2n}, t_n)$ be the probability distribution function representing the probability that the positions of the molecules are given by $x_1$ and $x_2$ at time $t$. The diffusion of the molecules is governed by the Smoluchowski equation

$$\partial_t p = D_1\Delta_{\mathbf{x}_1} p(\mathbf{x}_1, \mathbf{x}_2, t) + D_2\Delta_{\mathbf{x}_2} p(\mathbf{x}_1, \mathbf{x}_2, t), \tag{B1}$$

where $\Delta_{\mathbf{x}_i}$ is the $\Delta$-operator applied to $\mathbf{x}_i$, and where $p(x_1, x_2, t) := p(x_1, x_2, t|x_{1n}, x_{2n}, t_n)$.

Via a change of variables, $Y = X_1 - X_2$ and $\mathbf{Y} = \sqrt{D_2/D_1}\mathbf{x}_1 + \sqrt{D_1/D_2}\mathbf{x}_2$, we obtain two independent equations

$$\partial_t p(\mathbf{Y}, t) = D\Delta_{\mathbf{Y}} p(\mathbf{Y}, t), \tag{B2}$$

$$\partial_t p(\mathbf{y}, t) = D\Delta_{\mathbf{y}} p(\mathbf{y}, t). \tag{B3}$$

The reaction dynamics are now governed by the boundary conditions, given by

$$K\left.\frac{\partial p_{\mathbf{y}}}{\partial n}\right|_{\|\mathbf{y}\|=\sigma} = k_r p_{\mathbf{y}}(\|\mathbf{y}\| = \sigma, t), \tag{B4}$$

where

$$K = \begin{cases} 2\pi\sigma D, & (2D),\\ 4\pi\sigma^2 D, & (3D), \end{cases} \tag{B5}$$

and where $k_r$ is the microscopic reaction rate.

The initial condition is given by $p(\mathbf{y}, 0|\mathbf{y}_0, 0) = \delta(\mathbf{y} - \mathbf{y}_0)$, and we enforce $p(\|\mathbf{y}\| \to \infty) = 0$. New positions for two interacting molecules can be sampled by solving the above equations. An efficient algorithm for simulating systems of molecules is the GFRD algorithm.[13] Another approach, aimed at simplifying the sampling process by solving the above equation by performing operator splitting, is described in Ref. 42. For microscopic simulations on moving meshes, we follow the approach described in Ref. 21.

## APPENDIX C: MESOSCOPIC REACTION RATES IN 1D

In Section IV A we compare mesoscopic simulations in 1D with microscopic Smoluchowski simulations in 1D. To do an accurate comparison, we first need to determine how the mesoscopic reaction rates relate to the microscopic rates. To obtain this relationship we follow an approach similar to that in Refs. 43–45.

We consider a particle of species $A$ diffusing with rate $D$ in a 1D domain, and reacting with the left boundary according to

$$A \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftarrows}} A_{\mathrm{mem}}, \tag{C1}$$

where $A_{mem}$ is a molecule of species $A$ bound to the membrane. The rate $k_{on}$ is the effective rate, by which we mean that, given a uniformly distributed $A$ molecule, the molecule will react with the boundary after an average of $V/k_{on}$ s, where $V$ is the volume of the domain (in this case the length of the interval).

Let $k_r^{\mathrm{meso}}$ ($s^{-1}$)be the intensity with which an $A$ molecule reacts with the membrane, given that it is occupying the voxel adjacent to the membrane. First we find $k_r^{\mathrm{meso}}$ as a function of $k_{on}$ and the width $h$ of the voxel, and then proceed to determine $k_{on}$ as a function of the microscopic reaction rate $k_r$.

We first note that a problem equivalent to the above is that of a molecule $A$ starting uniformly on a 1D domain of width $2V$ reacting with a molecule $B$ fixed at the origin (at rate $k_{on}$). This can be seen by considering the symmetry of the problem. We therefore let $L = 2V$, and derive $k_r^{\mathrm{meso}}(k_{\mathrm{on}}, h)$ to match the average binding time of an $A$ molecule to a molecule fixed at the origin on a 1D interval of width $L$.

We denote the mean effective binding time of the two molecules by $\tau_{\mathrm{eff}}$. Thus $\tau_{\mathrm{eff}} = L/k_{\mathrm{on}}$, and

$$\tau_{\mathrm{eff}} = \tau_{\mathrm{meso}}(k_r^{\mathrm{meso}} \to \infty) + \tau_{\mathrm{meso}}^{\mathrm{rebind}}(k_r^{\mathrm{meso}}), \tag{C2}$$

where $\tau_{\mathrm{meso}}(k_r^{\mathrm{meso}})$ is, given a uniform inital distribution, the average time until the $A$ molecule reacts with the $B$ molecule on the mesoscopic (RDME) scale, and $\tau_{\mathrm{meso}}^{\mathrm{rebind}}(k_r^{\mathrm{meso}})$ is the average time until the $A$ molecule rebinds to the $B$ molecule, given that the molecules have just dissociated. Note that the rebinding time is the same as the time until the $A$ molecule reacts with the $B$ molecule, given that they start in the same voxel. For simplicity we will henceforth denote $\tau_{\mathrm{meso}}(k_r^{\mathrm{meso}} \to \infty)$ by $\tau_{\mathrm{meso}}(\infty)$. Let $d_{\mathrm{tot}}$ denote the mesoscopic jump rate out of a voxel (so that $d_{\mathrm{tot}} = h^2/(2D)$). Then

$$\theta = \frac{k_r^{\mathrm{meso}}}{k_r^{\mathrm{meso}} + d_{\mathrm{tot}}} \tag{C3}$$

is the probability that the next event, given that the $A$ molecule occupies the origin voxel, is a reaction event rather than a diffusion event. Let $\tau_e = (k_r^{\mathrm{meso}} + d_{\mathrm{tot}})^{-1}$ be the average time until this event fires. We obtain

$$\tau_{\mathrm{meso}}^{\mathrm{rebind}}(k_r^{\mathrm{meso}}) = \frac{1}{\theta}\left\{\theta\tau_e + (1-\theta)(\tau_e + t_1)\right\}, \tag{C4}$$

where $t_1$ is the average time until the $A$ molecule binds to the $B$ molecule, given that the molecule occupied the origin voxel and then diffused instead of reacted. Some straightforward algebra yields

$$\tau_{\mathrm{meso}}^{\mathrm{rebind}}(k_r^{\mathrm{meso}}) = \frac{1}{k_r^{\mathrm{meso}}}(1 + d_{\mathrm{tot}}t_1). \tag{C5}$$

We now use (C5) in (C2) and solve for $k_r^{\mathrm{meso}}$ to obtain

$$k_r^{\mathrm{meso}} = \frac{1 + d_{\mathrm{tot}}t_1}{\tau_{\mathrm{eff}} - \tau_{\mathrm{meso}}(\infty)}. \tag{C6}$$

In the expression above we have two unknowns: $t_1$ and $\tau_{\mathrm{meso}}(\infty)$. The effective binding time is a constant that we choose ($\tau_{\mathrm{eff}} = V/k_{\mathrm{on}}$).

From Refs. 46 and 47 we have analytical expressions for $t_1$ and $\tau_{\mathrm{meso}}(\infty)$,

$$1 + d_{\mathrm{tot}}t_1 = N, \tag{C7}$$

$$\frac{D}{2h^2}\tau_{\mathrm{meso}}(\infty) = \frac{N(N+1)}{6}, \tag{C8}$$

where $N$ is the number of voxels. We obtain

$$\tau_{\mathrm{meso}}(\infty) = \frac{2h^2}{D}\frac{N(N+1)}{6} = \frac{2L^2 + Lh}{6D}, \tag{C9}$$

$$k_r^{\mathrm{meso}} = \frac{1}{h}\frac{L}{\tau_{\mathrm{eff}} - \frac{2L^2+Lh}{6D}}. \tag{C10}$$

Note that $\tau_{\mathrm{eff}}$ is often considered a constant that we choose arbitrarily or obtain from experiments. However, in this case we want to compare mesoscopic simulations with microscopic simulations, and we derive $\tau_{\mathrm{eff}}$ as a function of the microscopic rate $k_r$.

The effective binding time $\tau_{\mathrm{eff}}$ can be written in terms of microscale quantities as the sum of the time until the $A$ molecule reaches the boundary for the first time, $L^2/(3D)$, and the time until it reacts with the boundary given that the $A$ molecule starts in contact with the boundary, $L/k_r$. We thus obtain

$$\tau_{\mathrm{eff}} = \frac{L^2}{3D} + \frac{L}{k_r}, \tag{C11}$$

in agreement with Ref. 48. Inserting (C11) into (C10), and assuming $L \gg h$ (so that $2L^2 \gg Lh$), we now obtain

$$k_r^{\mathrm{meso}} \approx \frac{k_r}{h}. \tag{C12}$$

Thus, when comparing a mesoscale simulation to a microscale (Smoluchowski) simulation in 1D, the mesoscopic reaction rate constant should be related to the microscopic rate according to (C12). Note that from (C12) follows that the mesoscopic dissociation rate will be given by $k_d^{\mathrm{meso}} = k_d$ to ensure that the steady states agree.

## APPENDIX D: EXPERIMENTAL DETAILS

Time-lapse imaging was performed on Mid2-GFP cells adhered to glass slides using concanavalin A in standard yeast-peptone-dextrose (YPD) media with the stage heated to 30 °C. Both fluorescent and bright-field images were taken at 10 min intervals over a 3 h period using a Nikon TE-2000 inverted microscope with an ORCA-2 CCD camera controlled by MetaMorph software. The yeast strain was constructed by fusing GFP onto the C-terminal end of MID2 using genomic integration of a W303-1A strain that is $\bar{1}\Delta$ (RJD863).

[1] N. Barkai and S. Leibler, Nature **403**, 267 (2000).
[2] J. Paulsson, O. G. Berg, and M. Ehrenberg, Proc. Natl. Acad. Sci. **97**, 7148 (2000).
[3] M. Thattai and A. van Oudenaarden, Proc. Natl. Acad. Sci. **98**, 8614 (2001).

[4] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).

[5] D. Pruyne and A. Bretscher, J. Cell Sci. **113**, 365 (2000).

[6] G. Servant, O. D. Weiner, P. Herzmark, T. Balla, J. W. Sedat, and H. R. Bourne, Science **287**, 1037 (2000).

[7] D. Fange and J. Elf, PLoS Comput. Biol. **2**, e80 (2006).

[8] K. C. Huang, Y. Meir, and N. S. Wingreen, Proc. Natl. Acad. Sci. **100**, 12724 (2003).

[9] A. Gierer and H. Meinhardt, Kybernetik **12**, 30 (1972).

[10] S. S. Andrews and D. Bray, Phys. Biol. **1**, 137 (2004).

[11] S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, PLoS Comput. Biol. **6**, e1000705 (2010).

[12] R. A. Kerr, T. M. Bartol, B. Kaminsky, M. Dittrich, J.-C. J. Chang, S. B. Baden, T. J. Sejnowski, and J. R. Stiles, SIAM J. Sci. Comput. **30**, 3126 (2008).

[13] J. S. van Zon and P. R. ten Wolde, Phys. Rev. Lett. **94**, 128103 (2005).

[14] D. Bernstein, Phys. Rev. E **71**, 041103 (2005).

[15] J. Elf and M. Ehrenberg, in IEE Proceedings On Systems Biology (IET, 2004) Vol. 1, pp. 230–236.

[16] A. B. Stundzia and C. J. Lumsden, J. Comput. Phys. **127**, 196 (1996).

[17] J. E. Segall, Proc. Natl. Acad. Sci. **90**, 8332 (1993).

[18] A. P. J. Trinci and P. T. Saunders, J. Gen. Microbiol. **103**, 243 (1977).

[19] T. Newman, Math. Biosci. Eng. **2**, 613 (2005).

[20] D. Thalmeier, J. Halatek, and E. Frey, Proc. Natl. Acad. Sci. **113**, 548 (2016).

[21] A. Ghosh and T. T. Marquez-Lago, PLoS One **10**, e0133401 (2015).

[22] R. E. Baker, C. A. Yates, and R. Erban, Bull. Math. Biol. **72**, 719 (2009).

[23] C. A. Yates, J. Theor. Biol. **350**, 37 (2014).

[24] C. A. Yates, R. E. Baker, R. Erban, and P. K. Maini, Phys. Rev. E **86**, 021921 (2012).

[25] B. Drawert, S. Engblom, and A. Hellander, BMC Syst. Biol. **6**, 1 (2012).

[26] A. Jilkine, S. B. Angenent, L. F. Wu, and S. J. Altschuler, PLoS Comput. Biol. **7**, e1002271 (2011).

[27] C. W. Gardiner *et al.*, *Handbook of Stochastic Methods* (Springer Berlin, 1985) Vol. 3.

[28] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).

[29] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).

[30] B. Drawert, M. Trogdon, S. Toor, L. Petzold, and A. Hellander, SIAM J. Sci. Computing **38**, C179 (2016).

[31] S. Engblom, L. Ferm, A. Hellander, and P. Lötstedt, SIAM J. Sci. Computing **31**, 1774 (2009).

[32] A. Logg and G. N. Wells, ACM Trans. Math. Software **37**, 20 (2010).

[33] B. Klünder, T. Freisinger, R. Wedlich-Söldner, and E. Frey, PLoS Comput. Biol. **9**, e1003396 (2013).

[34] T. I. Moore, C.-S. Chou, Q. Nie, N. L. Jeon, and T.-M. Yi, PLoS One **3**, e3865 (2008).

[35] T.-M. Yi, H. Kitano, and M. I. Simon, Proc. Natl. Acad. Sci. **100**, 10764 (2003).

[36] B. Drawert, M. J. Lawson, L. Petzold, and M. Khammash, J. Chem. Phys. **132**, 074101 (2010).

[37] A. B. Goryachev and A. V. Pokhilko, FEBS Lett. **582**, 1437 (2008).

[38] M. J. Lawson, B. Drawert, M. Khammash, L. Petzold, and T.-M. Yi, PLoS Comput. Biol. **9**, e1003139 (2013).

[39] R. Swaminathan, C. P. Hoang, and A. Verkman, Biophys. J. **72**, 1900 (1997).

[40] C.-S. Chou, L. Bardwell, Q. Nie, and T.-M. Yi, BMC Syst. Biol. **5**, 196 (2011).

[41] M. v. Smoluchowski, Z. Phys. Chemie **92**, 129 (1917).

[42] S. Hellander and P. Lötstedt, J. Comput. Phys. **230**, 3948 (2011).

[43] S. Hellander and L. R. Petzold, Phys. Rev. E **93**, 013307 (2015).

[44] S. Hellander, A. Hellander, and L. R. Petzold, Phys. Rev. E **85**, 042901 (2012).

[45] S. Hellander, A. Hellander, and L. R. Petzold, Phys. Rev. E **91**, 023312 (2015).

[46] E. W. Montroll and G. H. Weiss, J. Math. Phys. **6**, 167 (1965).

[47] E. W. Montroll, J. Math. Phys. **10**, 753 (1969).

[48] D. Fange, O. G. Berg, P. Sjöberg, and J. Elf, Proc. Natl. Acad. Sci. U. S. A. **107**, 19820 (2010).