

A computational kinetic model of diffusion for molecular systems

Ivan Teo and Klaus Schulten

Citation: *J. Chem. Phys.* **139**, 121929 (2013); doi: 10.1063/1.4820876

View online: <http://dx.doi.org/10.1063/1.4820876>

View Table of Contents: <http://jcp.aip.org/resource/1/JCPSA6/v139/i12>

Published by the AIP Publishing LLC.

Additional information on J. Chem. Phys.

Journal Homepage: <http://jcp.aip.org/>

Journal Information: http://jcp.aip.org/about/about_the_journal

Top downloads: http://jcp.aip.org/features/most_downloaded

Information for Authors: <http://jcp.aip.org/authors>

ADVERTISEMENT



**SHARPEN YOUR
COMPUTATIONAL
SKILLS.**

Subscribe for
\$49 | year

Computing
in **SCIENCE & ENGINEERING**
Scientific
Computing
with GPUs

A computational kinetic model of diffusion for molecular systems

Ivan Teo^{a)} and Klaus Schulten^{a)}

Beckman Institute for Advanced Science and Technology, University of Illinois, Urbana, Illinois 61801, USA

(Received 31 May 2013; accepted 15 August 2013; published online 23 September 2013)

Regulation of biomolecular transport in cells involves intra-protein steps like gating and passage through channels, but these steps are preceded by extra-protein steps, namely, diffusive approach and admittance of solutes. The extra-protein steps develop over a 10–100 nm length scale typically in a highly particular environment, characterized through the protein's geometry, surrounding electrostatic field, and location. In order to account for solute energetics and mobility of solutes in this environment at a relevant resolution, we propose a particle-based kinetic model of diffusion based on a Markov State Model framework. Prerequisite input data consist of diffusion coefficient and potential of mean force maps generated from extensive molecular dynamics simulations of proteins and their environment that sample multi-nanosecond durations. The suggested diffusion model can describe transport processes beyond microsecond duration, relevant for biological function and beyond the realm of molecular dynamics simulation. For this purpose the systems are represented by a discrete set of states specified by the positions, volumes, and surface elements of Voronoi grid cells distributed according to a density function resolving the often intricate relevant diffusion space. Validation tests carried out for generic diffusion spaces show that the model and the associated Brownian motion algorithm are viable over a large range of parameter values such as time step, diffusion coefficient, and grid density. A concrete application of the method is demonstrated for ion diffusion around and through the *Escherichia coli* mechanosensitive channel of small conductance ecMscS. © 2013 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4820876>]

I. INTRODUCTION

Diffusion is a mainstay of biological systems across many time and length scales. On the biological cell level, many phenomena have been framed as diffusion-controlled processes, from transport processes,^{1–3} ligand binding,^{4–8} and signal transduction within the cell,^{9–12} to cell-to-cell signaling.^{12–14} These processes can depend on molecular-level detail in regard to the geometry of the diffusion space, energetics, and local variation of diffusivity. Experimental investigations of molecular-scale transport are often unfeasible. Fortunately, observations can be complemented by computer simulations. In fact, diffusion theory^{2,6,15} is well-established, making diffusion-controlled processes amenable to computer simulation. However, applications of diffusion theory in the past have glossed over the molecular-scale variation of geometries, energetics, and mobilities of transported solutes.

To appreciate the need for the use of diffusion theory we also note that biophysics has made great progress in understanding the regulation of transport at the intra-protein level, in particular in case of membrane channels. However, the intra-protein steps are preceded by diffusive approach and control of access to the relevant surface openings of channel proteins, in particular, since the relevant overall diffusion space is often highly intricate in regard to local geometry as well as solute energetics and mobility. Spatial and

time scales for the diffusive approach are typically 10–100 nm and ms, respectively. In the present study, we suggest and test a flexible computational scheme to describe the initial diffusive approach step of biological transport. This scheme is based on extensive prior sampling through nanosecond molecular dynamics (MD) simulations and a subsequent application of diffusion theory that furnishes extremely realistic microsecond to millisecond descriptions at molecular resolution.

A diverse set of simulation techniques are already routinely employed to model diffusion on the spatial and temporal scales relevant for cellular transport mechanisms. Programs such as Smoldyn,¹⁶ MCell,¹⁷ and VCell¹⁸ have been used successfully for reaction-diffusion simulations but on a larger scale than considered in the present study, namely, on the scale of whole cells. Another approach commonly used to describe biological diffusion processes is that of Green's function reaction dynamics,^{19–21} which solves the diffusion equation for one particle or two particles and uses the resulting Green's function solution to propagate particle positions in time. Until now, the aforementioned computational tools assume free diffusion or the presence of a simple potential, typically arising from a few inter-particle interactions, and in this case are able to describe large systems well. On the much smaller molecular scale, however, inter-molecular interactions with the environment need to be accounted for through detailed, complex potentials that require descriptions based on advanced numerical techniques. Of these techniques, MD remains the most detailed, but also the computationally most expensive technique, the expense placing limits on spatial and

^{a)}Also at Department of Physics, University of Illinois, Urbana, Illinois 61801, USA. Electronic addresses: teo7@ks.uiuc.edu and kschulte@ks.uiuc.edu

temporal scales that can actually be covered.^{22–24} To overcome such limitations, MD must often be supplemented by sampling techniques and parallelization schemes.²⁵ Brownian dynamics,²⁶ which sacrifices some level of detail by treating solvents as implicit and large molecules as reflective barriers without internal degrees of freedom, has been successfully used to simulate larger systems.^{27–29} Efforts to extend the reach of molecular-level simulations to greater length and time scales include diffusion Monte Carlo algorithms, such as that implemented in BioMOCA,³⁰ and mean field descriptions of diffusion, commonly implemented through a finite element approach.^{10,11,14} However, in the mean field description, one loses information about individual particle trajectories, which may be important to trace in certain cases.

Heterogeneous hybrid methods have also been developed, largely for cases where flexible resolution allows a small subset of the system to be modeled at a higher level of detail than the rest of the system, which is coarsely modeled to avoid computational expense. In particular, the low computational cost of the finite element approach makes it a good candidate for combination with a higher resolution approach, such as Brownian dynamics.³¹ However, correctly describing particle transport across the interface between the two distinctly modeled regions is not a straightforward task.

In any computational investigation of a diffusive process, the limitations of existing simulation methods described above necessitates a careful choice of method, guided by the scale of the system, knowledge of the process to be studied, and the availability of computational resources. The latter two factors, together with the need to be familiar with multiple simulation methods, present a hurdle to cross during early-stage investigations. Obtaining enough sampling of diffusive processes on relevant time scales presents a challenge to atomistic methods such as MD and Brownian dynamics. In the particular case of MD, one may not observe an expected phenomenon even after extensive sampling. On the other hand, coarse-grained simulations may not be adequately detailed to reproduce diffusive motions influenced by intricate environmental effects. Thus, there is a motivation to interface MD calculations with coarse-grained diffusion algorithms to take advantage of the atomic detail obtained by the former and the long time scales accessible to the latter.

In order to realize such interface, we propose to capitalize on the finite element methodology that has the ability to describe arbitrary potential fields and local mobilities. The method offers the flexibility of a multi-resolution grid and permits descriptions afforded by the Markov State Model (MSM) protocol³² to develop a versatile particle-based method that is valid and computationally viable over a wider range of length and time scales as compared to other diffusion methods, without compromising the level of detail in describing the potential field. Through the MSM scheme, one can divide the computational effort in an extremely useful manner between a sampling step that gathers the physical characteristics of the diffusion model and a diffusion execution step that describes the actual transport between cell environment and protein channel. This division allows the description of

diffusion within an arbitrary potential field, extending earlier methods applicable in case of large systems only to free diffusion.

The need for detail in large systems is motivated by the aim of describing the diffusive approach of solutes in simulations of membrane channels, as discussed further in Sec. V. Our algorithm models the system at varying levels of detail and optimizes computational efficiency through adjustment to the level of detail required for the system's description. Our method allows also the use of large time steps, extending thereby the reach of simulations to time scales longer than those of other molecular-scale methods.

In our algorithm, diffusion is implemented through a kinetic model of particles transitioning between pre-defined states. The rates of transition between states are specified by a rate matrix. Given the size of a time step, one can obtain the respective probabilities of the particle transitioning from its current state to each of the other states within the span of a time step by solving for the eigenvalues and eigenvectors of the rate matrix.

The set of states is characterized by positions in the system, namely, the centers of cells in an irregular grid of varying resolution overlaid on the system. The rate matrix is calculated from the discretized Smoluchowski equation, using pre-obtained input parameters, namely, the diffusion coefficient and a potential of mean force (PMF) map of the system for the diffusing species. Solving the eigenproblem of the rate matrix then gives the transition matrix, which propagates a particle's position through time. Key to the method is the pairing of a prior MD sampling and a following diffusion calculation; here pairing implies that the two computations are linked through straightforward data exchange.

Of the contemporary diffusion descriptions discussed above, the Monte Carlo algorithm used in BioMOCA³⁰ is the one most similar to the present method in terms of functionality. The key distinction lies in how simulation parameters are chosen. BioMOCA obtains the electrostatic potential map from an *a priori* solution of the Poisson equation, treats ion-protein interactions by imposing suitable boundary conditions, and assumes diffusion coefficients from literature. Instead, the present method uses extensive MD simulation to calculate the PMF map and local diffusion coefficient map. The advantage of using an MD calculation is that it bypasses difficulties in determining local dielectric effects and diffusion coefficients at every position in the system. MD calculations require significant computational effort in order to reach sufficient convergence, but modern software and hardware readily permit the needed calculations as demonstrated below.

The implementation of the finite difference scheme for diffusion calculations has been carried out at this point in Matlab.³³ The subroutines used and specified below were taken from existing Matlab libraries.

For the diffusion model a scheme of space-filling Voronoi cells is employed, each cell geometrically defined through position, volume, as well as number and sizes of surface elements shared with neighboring cells. The physical characteristics of the cells are gathered through extensive MD simulation.

For the purpose of validation, our computational scheme has been used to reproduce the expected statistical behavior in simple systems for which analytic descriptions are known. It was found that, to the extent of the values of parameters tested (namely, time step and diffusion coefficient), the accuracy of the scheme is limited by the resolution of the grid used to discretize the system. The model is subsequently applied to the realistic case of ion diffusion through the *Escherichia coli* mechanosensitive channel of small conductance (ecMscS). Results agree with previous simulations using established methods.

II. ALGORITHMS EMPLOYED

Actual use of the methodology suggested in the present study begins with MD simulations sampling solute energetics and diffusion coefficient maps. These maps serve as input into a diffusion algorithm. We present in this section first the diffusion algorithm and apply it in Sec. III to test cases that do not require MD simulation input. In Sec. V, we employ our methodology in a realistic test case that requires MD simulation input. We introduce the concepts and algorithms used for pairing MD data and the diffusion algorithm at this late point of the present paper, even though in an actual application the MD calculations need to always be carried out first.

Upon the premise that the MD simulations have been run and the relevant data obtained, we describe the derivation of the algorithms involved in the diffusion model. The algorithms employed in setting up the geometrical and physical details of the diffusion space are described first. We then formulate the discrete form of the Smoluchowski equation governing transport processes on time scales of picoseconds or longer, i.e., in the strong friction regime. Finally, we introduce the numerical solution employed which actually corresponds to a Brownian dynamics scheme, though one with a novel numerical integrator adapted to the highly heterogeneous nature of actual diffusion spaces.

A. Setting up a discrete representation of the system

The efficiency of our diffusion algorithm is dependent on the choice of the grid used to discretize the system. In this regard, one should choose a grid density profile suited to the local level of detail in the description of the system, while minimizing the error arising naturally from approximating a continuous space with a discrete one. Our approach employs for this purpose a topology-conforming self-organizing map^{34–36} that distributes cell centers and determines then the respective Voronoi tessellation.

Let X , a subset of \mathbb{R}^n , represent the diffusion space. Typically, X is a subset of the three-dimensional space \mathbb{R}^3 . $U(\mathbf{x})$ and $p(\mathbf{x}, t)$ are, respectively, the underlying potential felt by a particle at $\mathbf{x} \in X$ and the probability distribution of diffusing particles at \mathbf{x} and time t . Our aim is to divide the otherwise continuous system up into a discrete collection of regions. For this purpose, a set of N points $\{\mathbf{w}_i \in X | i = 1, 2, \dots, N\}$ are selected at random. These points represent the centers of cells in the Voronoi tessellation grid \mathcal{I} of X . Formally, \mathcal{I} is defined

through

$$\mathcal{I} = \left\{ I_i \subset X \mid \bigcup_i I_i = X \text{ and } I_i \cap I_j = \emptyset \forall i \neq j, \right. \\ \left. \mathbf{x} \in I_i \Leftrightarrow \|\mathbf{x} - \mathbf{w}_i\| < \|\mathbf{x} - \mathbf{w}_j\| \forall i \neq j, \quad i = 1, 2, \dots, N \right\}. \quad (1)$$

Before constructing \mathcal{I} , we adapt the positions $\{\mathbf{w}_i\}$ to conform topologically to the pre-defined distribution $\rho(\mathbf{x})$. The resulting grid will have two desirable properties.

First, the density of cells is locally homogeneous, i.e., for any point \mathbf{x} in X , the density of cells within a neighborhood of \mathbf{x} approaches $\rho(\mathbf{x})$ as the size of the neighborhood approaches the average distance between cells. This notion of local homogeneity is formalized as

$$r \rightarrow r_\rho(\mathbf{x}) \Rightarrow \rho_w(S_r(\mathbf{x})) \rightarrow \rho(\mathbf{x}), \quad (2)$$

where $\rho_w(S_r(\mathbf{x}))$ is the density of cell centers \mathbf{w}_i , and thus of cells, within a sphere $S_r(\mathbf{x})$ of radius r centered on \mathbf{x} , and $r_\rho(\mathbf{x})$ is the length scale associated with $\rho(\mathbf{x})$, so that $r_\rho(\mathbf{x}) \sim \rho(\mathbf{x})^{-1/3}$. Thus, the local diffusive behavior of particles will be subject to minimal error arising from local grid density deviations from $\rho(\mathbf{x})$.

Second, \mathcal{I} is a centroidal Voronoi tessellation (CVT).³⁷ In other words, each \mathbf{w}_i coincides with the centroid of cell I_i . A measure of how well a cell center represents the entire cell is given by the quadrature sum of distances of every point within the cell from the center. The CVT property minimizes this quadrature sum, called the mean square deviation (MSD) of points within I_i , given by

$$\text{MSD}_x^{(i)} = \left(\int_{I_i} d\mathbf{x} \|\mathbf{x} - \mathbf{w}_i\|^2 \right) / \int_{I_i} d\mathbf{x}. \quad (3)$$

One would also like to obtain a good discrete representation of a continuous variable p defined on X . Under the assumption that $p(\mathbf{x})|_{I_i}$ varies linearly with $\|\mathbf{x} - \mathbf{w}_i\|$ (a good approximation for small grid cells), the chosen discretization scheme also minimizes the mean square deviation in p , given by

$$\text{MSD}_p^{(i)} = \left(\int_{I_i} d\mathbf{x} |p(\mathbf{x}) - p(\mathbf{w}_i)|^2 \right) / \int_{I_i} d\mathbf{x}. \quad (4)$$

A topology-conforming distribution of $\{\mathbf{w}_i\}$ can be constructed using an iterative procedure, due to Martinez, Berkovich, and Schulten,³⁵ composed of the following four steps:

1. Begin with any random distribution of $\{\mathbf{w}_i\}$ over X .
2. From the reference distribution $\rho(\mathbf{x})$, draw a test point $\mathbf{v} \in X$.
3. Rank $\{\mathbf{w}_i\}$ according to each respective element's distance from \mathbf{v} . According to this order, assign a rank integer $k(\mathbf{v}, \mathbf{w}_i) = 0, 1, 2, \dots, N - 1$ to each \mathbf{w}_i , increasing from the nearest to the farthest.
4. Update each \mathbf{w}_i as follows:

$$\mathbf{w}_i(s+1) = \mathbf{w}_i(s) + \epsilon(s)(\mathbf{v}(s) - \mathbf{w}_i(s))e^{[-k(\mathbf{v}(s), \mathbf{w}_i(s))/\lambda(s)]}, \quad (5)$$

where s labels the adaptation step, ϵ and λ control, respectively, the magnitude of the change and the extent of the area of influence around \mathbf{v} . Each adaptation step s comprises steps 2–4. The prescription³⁵ calls for a gradual decrease of ϵ and λ , such that for each s ,

$$\epsilon(s) = \epsilon_{\text{initial}} \left(\frac{\epsilon_{\text{final}}}{\epsilon_{\text{initial}}} \right)^{s/S}, \quad (6)$$

$$\lambda(s) = \lambda_{\text{initial}} \left(\frac{\lambda_{\text{final}}}{\lambda_{\text{initial}}} \right)^{s/S}, \quad (7)$$

where S is the total number of adaptation steps chosen by the user. Thence, set $\mathbf{w}_i \equiv \mathbf{w}_i(S)$. The parameters $\epsilon_{\text{initial}}$, ϵ_{final} , λ_{initial} , λ_{final} , and S must be tuned through trial-and-error to achieve convergence to the desired distribution of \mathbf{w}_i . Convergence may be measured by the mean square deviation of the grid or of some subset of cells J in the grid, namely, $\sum_{i \in J} \text{MSD}_x^{(i)}$.

Having determined \mathbf{w}_i , the Matlab subroutine `DelaunayTri` is employed to extract the Delaunay triangulation. The latter is obtained by connecting lines between pairs of nearest neighbor \mathbf{w}_i 's. The Voronoi tessellation is calculated from the Delaunay triangulation by means of the Matlab subroutine `voronoiDiagram`. The tessellation is specified by the vertices of each cell I_i , from which the cell volumes and interfacial cell surface areas needed for the discretization of the diffusion equation can be calculated.

B. Discretization of the Smoluchowski equation

In the presence of a potential $U(\mathbf{x})$, diffusion is described by the Smoluchowski equation in continuous space:

$$\dot{p}(\mathbf{x}, t) = \nabla \cdot D(\mathbf{x}) e^{-\beta U(\mathbf{x})} \nabla e^{\beta U(\mathbf{x})} p(\mathbf{x}, t), \quad (8)$$

where $D(\mathbf{x})$ is the local diffusion coefficient and $\beta = (k_B T)^{-1}$. This equation had been used extensively to mimic diffusion outside and inside of the ecMscS cytoplasmic domain.³

In order to choose $D(\mathbf{x})$ and $U(\mathbf{x})$, a MD simulation under equilibrium conditions can be used to calculate the diffusion coefficient and generate a PMF map, which is subsequently interpolated to provide an associated potential value for every grid cell. Such analysis is demonstrated for ecMscS in Sec. V. In case that simplified models for $D(\mathbf{x})$ and $U(\mathbf{x})$ suffice, the two quantities can be obtained by some other means, e.g., as in a recent study of SecY,³⁸ where the potential map within a protein channel was obtained by calculating the potential along a specific radial direction and generalizing the result by assuming radial symmetry.

With the required data in hand, we discretize Eq. (8) in space by integrating over a generic cell I_i resulting in

$$\int_{I_i} d\mathbf{x} \dot{p}(\mathbf{x}, t) = \int_{I_i} d\mathbf{x} \nabla \cdot D(\mathbf{x}) e^{-\beta U(\mathbf{x})} \nabla e^{\beta U(\mathbf{x})} p(\mathbf{x}, t) \quad (9a)$$

$$= \int_{\partial I_i} d\sigma \hat{\mathbf{n}}(\mathbf{x}) \cdot D(\mathbf{x}) e^{-\beta U(\mathbf{x})} \nabla e^{\beta U(\mathbf{x})} p(\mathbf{x}, t) \quad (9b)$$

$$= \int_{\partial I_i} d\sigma D(\mathbf{x}) e^{-\beta U(\mathbf{x})} \frac{\partial}{\partial \hat{\mathbf{n}}} e^{\beta U(\mathbf{x})} p(\mathbf{x}, t). \quad (9c)$$

Here Gauss' theorem has been applied in the second line to obtain an integral over the surface ∂I_i of cell I_i with $\hat{\mathbf{n}}(\mathbf{x})$ representing the unit surface normal. The dot product in the second line is then converted to a directional derivative in the third line.

Next, we make the approximation that the quantities $p(\mathbf{x}, t)$, $U(\mathbf{x})$, and $D(\mathbf{x})$ are uniformly valued in the interior of each cell i with center \mathbf{w}_i resulting in

$$p(\mathbf{x}, t) \approx p(\mathbf{w}_i, t) \quad \forall \mathbf{x} \in I_i \setminus \partial I_i, \quad (10)$$

$$U(\mathbf{x}) \approx U(\mathbf{w}_i) \quad \forall \mathbf{x} \in I_i \setminus \partial I_i, \quad (11)$$

$$D(\mathbf{x}) \approx D(\mathbf{w}_i) \quad \forall \mathbf{x} \in I_i \setminus \partial I_i. \quad (12)$$

Furthermore, we set the values of variables at each cell interface to be the average of the values in the two cells, namely,

$$\begin{aligned} D e^{-\beta U(\mathbf{x})} \frac{\partial}{\partial \hat{\mathbf{n}}} e^{\beta U(\mathbf{x})} p(\mathbf{x}, t) \Big|_{\partial I_{ij}} \\ \approx \frac{D(\mathbf{w}_i) + D(\mathbf{w}_j)}{2} e^{-\beta[U(\mathbf{w}_i) + U(\mathbf{w}_j)]/2} \\ \times \frac{e^{\beta U(\mathbf{w}_j)} p(\mathbf{w}_j, t) - e^{\beta U(\mathbf{w}_i)} p(\mathbf{w}_i, t)}{\|\mathbf{w}_j - \mathbf{w}_i\|}, \end{aligned} \quad (13)$$

where ∂I_{ij} is the interface between I_i and I_j . Putting Eqs. (10)–(13) into Eq. (9c) gives

$$\begin{aligned} V_i \dot{p}(\mathbf{w}_i, t) = \sum'_{j \neq i} A_{ij} \frac{D(\mathbf{w}_i) + D(\mathbf{w}_j)}{2} e^{-\beta[U(\mathbf{w}_i) + U(\mathbf{w}_j)]/2} \\ \times \frac{e^{\beta U(\mathbf{w}_j)} p(\mathbf{w}_j, t) - e^{\beta U(\mathbf{w}_i)} p(\mathbf{w}_i, t)}{\|\mathbf{w}_j - \mathbf{w}_i\|}, \end{aligned} \quad (14)$$

where V_i is the volume of cell I_i and A_{ij} is the interfacial area between I_i and I_j . The sum over j involves solely Voronoi cells sharing an interface with cell i , namely cells that are direct neighbors of cell i ; the restriction is denoted by the prime attached to the summation symbol. In order to be consistent in the use of extensive quantities in the model, we rewrite the probability density p in terms of total probability in a cell P , such that

$$p(\mathbf{w}_i, t) = P_i(t)/V_i. \quad (15)$$

Substituting the above into Eq. (14) gives

$$\begin{aligned} \dot{P}_i(t) = \sum'_{j \neq i} A_{ij} \frac{D(\mathbf{w}_i) + D(\mathbf{w}_j)}{2} e^{-\beta[U(\mathbf{w}_i) + U(\mathbf{w}_j)]/2} \\ \times \frac{e^{\beta U(\mathbf{w}_j)} P_j(t)/V_j - e^{\beta U(\mathbf{w}_i)} P_i(t)/V_i}{\|\mathbf{w}_j - \mathbf{w}_i\|} \end{aligned} \quad (16a)$$

$$\begin{aligned} = \sum'_{j \neq i} \left\{ A_{ij} \frac{D(\mathbf{w}_i) + D(\mathbf{w}_j)}{2} \right. \\ \cdot \frac{\exp\left[-\frac{\beta}{2}(U(\mathbf{w}_i) - U(\mathbf{w}_j))\right]}{V_j \|\mathbf{w}_j - \mathbf{w}_i\|} P_j(t) \left. \right\} \\ - \left\{ \sum'_{j \neq i} A_{ij} \frac{D(\mathbf{w}_i) + D(\mathbf{w}_j)}{2} \right. \\ \cdot \frac{\exp\left[-\frac{\beta}{2}(U(\mathbf{w}_j) - U(\mathbf{w}_i))\right]}{V_i \|\mathbf{w}_j - \mathbf{w}_i\|} \left. \right\} P_i(t). \end{aligned} \quad (16b)$$

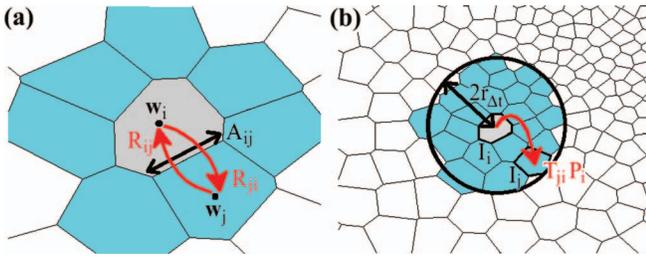


FIG. 1. Discretization scheme for the case of two-dimensional diffusion. (a) Rate matrix components R_{ij} and R_{ji} give the rates of probability flow between cells I_i and its nearest neighbors, indexed by j , and are dependent on geometric properties such as the inter-cell distance $\|\mathbf{w}_i - \mathbf{w}_j\|$, the interfacial area A_{ij} , and the cell volumes V_i and V_j (not shown in diagram). (b) Illustration of local probability flow from cell I_i to other cells (indexed by j) in its neighborhood (encircled area). In the deterministic approach, $T_{ji}P_i(t)$ gives the amount of probability that flows from cell I_i to cell I_j between times t and $t + \Delta t$. In the stochastic approach, T_{ji} gives the probability that a particle in cell I_i at time t hops to cell I_j between times t and $t + \Delta t$.

This equation is key to our discretization scheme as it expresses the Smoluchowski equation through the values of D and U at the centers of the Voronoi cells as well as through the cell volumes, areas of the connecting faces and center-center distances between cells. The equation obeys detailed balance (see Eq. (20)) such that it ensures the existence of an equilibrium state given by the Boltzmann distribution. The equation is of great value as it provides the simplest possible account for the geometry of the Voronoi cells in the context of a discretized diffusion model reproducing the Smoluchowski equation in the limit of vanishingly small cells.

Finally, we define the coefficients of $P_j(t)$ and $P_i(t)$ to be R_{ij} and R_{ii} , respectively, such that Eq. (16b) reads

$$\dot{P}_i(t) = \sum_{j \neq i} R_{ij} P_j(t) + R_{ii} P_i(t), \quad (17)$$

which can be written as a linear kinetic equation

$$\dot{\mathbf{P}}(t) = \mathbf{R} \cdot \mathbf{P}(t). \quad (18)$$

In the following we will eliminate the prime labeling the summation symbol, assuming from then on that R_{ij} is non-zero only for nearest neighbor Voronoi cells. The rate matrix \mathbf{R} arising in Eq. (18) has four important properties.

1. $R_{ij}P_j(t)$ is the rate of probability inflow from I_j to I_i and $R_{ii}P_i(t)$ is the rate of outflow from I_i to its nearest neighbors (see Fig. 1(a)). By observing the terms in Eq. (16b), one will find that the total flow rate to other cells from I_i , given by $\sum_{i \neq j} R_{ji}P_i(t)$, is equal to $R_{ii}P_i(t)$, the outflow rate from I_i , thus ensuring particle conservation.
2. The solution of (18) relaxes to a stationary, i.e., equilibrium, distribution \mathbf{P}^0 , which is given by the Boltzmann distribution

$$P_i^0 \equiv Z^{-1} V_i e^{-\beta U(\mathbf{w}_i)}, \quad (19)$$

where Z is the partition function. By construction, detailed balance is obeyed, namely

$$\begin{aligned} R_{ij}P_j^0 &= \frac{A_{ij}}{2Z} [D(\mathbf{w}_i) + D(\mathbf{w}_j)] \\ &\times \frac{\exp[-\frac{\beta}{2}(U(\mathbf{w}_i) + U(\mathbf{w}_j))]}{\|\mathbf{w}_j - \mathbf{w}_i\|} \\ &= R_{ji}P_i^0. \end{aligned} \quad (20)$$

3. To solve the Smoluchowski Eq. (8), a complete boundary of reflective or absorptive faces of Voronoi cells needs to be defined; completeness implies that the boundary faces define a complete enclosure. To develop such a boundary, one needs to specify which Voronoi cells are considered beyond the boundary and therefore eliminated from the system. The boundary faces are then defined as the interfacial intersection between kept and eliminated Voronoi cells.
4. Reflective or absorptive boundary conditions for a cell I_i are imposed as follows. In case of a reflective boundary to an adjacent, but eliminated cell I_j , i.e., for A_{ij} being a reflective face on the system boundary, one sets

$$R_{ij} = R_{ji} = 0. \quad (21)$$

If instead the face A_{ij} on the system boundary is absorptive, probability may flow out, but not in, such that one sets

$$R_{ji} \neq 0, R_{ij} = 0. \quad (22)$$

C. Solution of the rate equation

The next step is to solve Eq. (18), given the initial distribution $\mathbf{P}(0)$. The approach adopted here is to solve for the eigenvalues and eigenvectors of \mathbf{R} , which yield, together with the initial condition, an exact solution of Eq. (18). Thus, the only source of error due to time discretization is the assumption that a particle begins each time step being completely equilibrated within its current cell.³²

Depending on the number of cells N in \mathcal{I} , solving for the entire matrix \mathbf{R} at once can be computationally expensive. We briefly discuss the complexity involved in Sec. IV. A better alternative is to solve for the diffusive behavior locally, as done in the framework of the Brownian dynamics algorithm,^{2,26,39} as well as in the MSM.³² For this purpose, we make use of the fact that for a particle initially in I_i , the extent of diffusion is effectively limited to a region characterized by the radius $r_{\Delta t} = \sqrt{2nD\Delta t}$ about \mathbf{w}_i in time step Δt , where n is the number of spatial dimensions, ignoring presently a possible drift of probability due to the non-zero local force $-\nabla U(\mathbf{x})$.

For each I_i , we consider the cell centers contained within some radius r_{restrict} of \mathbf{w}_i and construct a restriction $\mathbf{R}^{(i)}$ of \mathbf{R} to these cells. For our purposes, we set r_{restrict} to $2r_{\Delta t}$. $\mathbf{R}^{(i)}$ is composed of only elements of \mathbf{R} associated with probability transfers between cells within the $2r_{\Delta t}$ radius. For the purpose of the local computation, we index the matrix elements of $\mathbf{R}^{(i)}$ differently from those of \mathbf{R} such that in general holds $R_{jk}^{(i)} \neq R_{jk}$. For the sake of bookkeeping, we henceforth use

the local index notation lmn for local elements of $\mathbf{R}^{(i)}$, while we continue to employ ijk for the global indices. We also introduce the permutation $\sigma^{(i)}$ that maps the global index k of a cell to the corresponding local index n specific to the restricted region centered on cell I_i , namely,

$$\sigma^{(i)}(k) = n, \quad (23)$$

so that by construction holds

$$R_{\sigma^{(i)}(j), \sigma^{(i)}(k)}^{(i)} = R_{jk}. \quad (24)$$

In order to preserve particle conservation, reflective boundary conditions are applied to the bordering cells of each restricted neighborhood, and $R_{ll}^{(i)}$'s are re-calculated so that the sum over elements in any column of $\mathbf{R}^{(i)}$ equals zero.

The problem to be solved then is

$$\dot{\mathbf{P}}^{(i)}(t) = \mathbf{R}^{(i)} \cdot \mathbf{P}^{(i)}(t), \quad (25)$$

where $\mathbf{P}^{(i)}(t)$ is the vector with components representing the probability in each locally indexed cell. Henceforth, the superscript (i) will denote quantities that have been re-indexed locally. The eig Matlab subroutine is used to calculate the eigenvalues $\lambda_n^{(i)}$ and corresponding eigenvectors $v_n^{(i)}$ of $\mathbf{R}^{(i)}$, where $n = 1, 2, \dots, \dim(\mathbf{R}^{(i)})$. The solution of Eq. (25) is then

$$\mathbf{P}^{(i)}(t) = \sum_n \alpha_n^{(i)} \exp(\lambda_n^{(i)} t) v_n^{(i)}, \quad (26)$$

where $\{\alpha_n^{(i)}\}$ are scalar constants to be determined. The initial distribution defines the coefficients $\alpha_n^{(i)}$ through

$$\mathbf{P}^{(i)}(0) = \sum_n \alpha_n^{(i)} v_n^{(i)}. \quad (27)$$

In case that the particle is initially in cell i , holds

$$P_n^{(i)}(0) = \delta_{\sigma^{(i)}(i), n}. \quad (28)$$

Since $\mathbf{R}^{(i)}$ is not symmetric in general, $\{v_n^{(i)}\}$ is not expected to be an orthogonal set of vectors. However, $\mathbf{R}^{(i)}$ is similar to a symmetric matrix $\tilde{\mathbf{R}}^{(i)}$ under the transformation

$$\tilde{\mathbf{R}}^{(i)} = (\mathbf{M}^{(i)})^{-1} \mathbf{R}^{(i)} \mathbf{M}^{(i)}, \quad (29)$$

where $\mathbf{M}^{(i)}$ is specified by $M_{lm}^{(i)} = \delta_{lm} [V_l^{(i)} \exp(-\beta U(\mathbf{w}_l^{(i)}))]^{1/2}$. Thence,

$$\tilde{\mathbf{R}}_{lm}^{(i)} = (M^{(i)})_{ll}^{-1} R_{lm}^{(i)} M_{mm}^{(i)} \quad (30a)$$

$$= A_{lm}^{(i)} \frac{D(\mathbf{w}_l^{(i)}) + D(\mathbf{w}_m^{(i)})}{2\sqrt{V_l^{(i)} V_m^{(i)}} \|\mathbf{w}_l^{(i)} - \mathbf{w}_m^{(i)}\|} \quad (30b)$$

$$= \tilde{\mathbf{R}}_{ml}^{(i)}. \quad (30c)$$

Thus, $\tilde{\mathbf{R}}^{(i)}$ is symmetric, so that its eigenvectors $\{\tilde{v}_n^{(i)}\}$ are orthogonal. To find $\tilde{v}_n^{(i)}$, one observes that from the definition $\mathbf{R}^{(i)} v_n^{(i)} = \lambda_n^{(i)} v_n^{(i)}$ follows:

$$\tilde{\mathbf{R}}^{(i)} (\mathbf{M}^{(i)})^{-1} v_n^{(i)} = ((\mathbf{M}^{(i)})^{-1} \mathbf{R}^{(i)} \mathbf{M}^{(i)}) (\mathbf{M}^{(i)})^{-1} v_n^{(i)} \quad (31a)$$

$$= (\mathbf{M}^{(i)})^{-1} \mathbf{R}^{(i)} v_n^{(i)} \quad (31b)$$

$$= \lambda_n^{(i)} ((\mathbf{M}^{(i)})^{-1} v_n^{(i)}). \quad (31c)$$

Hence, $\tilde{v}_n^{(i)} = (\mathbf{M}^{(i)})^{-1} v_n^{(i)}$ is an eigenvector of $\tilde{\mathbf{R}}^{(i)}$ with eigenvalue $\lambda_n^{(i)}$. The orthogonality condition of the eigenvectors $\{\tilde{v}_n^{(i)}\}$ reads

$$\frac{\tilde{v}_l^{(i)} \cdot \tilde{v}_m^{(i)}}{|\tilde{v}_l^{(i)}| \cdot |\tilde{v}_m^{(i)}|} = \delta_{lm}. \quad (32)$$

Right multiplying Eq. (27) with the inverse similarity transform matrix and applying to it the orthogonality condition Eq. (32) gives

$$\frac{((\mathbf{M}^{(i)})^{-1} \mathbf{P}^{(i)}(0)) \cdot \tilde{v}_l^{(i)}}{|\tilde{v}_l^{(i)}|^2} = \sum_m \alpha_m^{(i)} \delta_{lm} = \alpha_l^{(i)}. \quad (33)$$

On condition that the particle is initially in cell I_i , we calculate the coefficients $\alpha_l^{(i)}$ by putting Eq. (28) into Eq. (33) and obtaining

$$\alpha_l^{(i)} = \frac{(M^{(i)})_{\sigma^{(i)}(i), \sigma^{(i)}(i)}^{-1} (\tilde{v}_l^{(i)})_{\sigma^{(i)}(i)}}{|\tilde{v}_l^{(i)}|^2}. \quad (34)$$

Setting $t = \Delta t$, we thus obtain for the probability distribution in the neighborhood of the initial position after one time step

$$\mathbf{P}^{(i)}(\Delta t) = \sum_n \alpha_n^{(i)} \exp(\lambda_n^{(i)} \Delta t) v_n^{(i)}. \quad (35)$$

Using the local transition probabilities given by the elements of $\mathbf{P}^{(i)}$ for every $i \in \{1, 2, \dots, N\}$, we construct the transition matrix \mathbf{T} , where each element T_{ji} is the transition probability of a particle moving from cell I_i to cell I_j (see Fig. 1(b)), given by

$$T_{ji} = P_{\sigma^{(i)}(j)}^{(i)}(\Delta t). \quad (36)$$

The framework developed thus far can be used for both deterministic and stochastic simulations. In the deterministic case, one uses the transition matrix to propagate a probability distribution in time

$$\mathbf{P}(t + \Delta t) = \mathbf{T} \cdot \mathbf{P}(t). \quad (37)$$

The deterministic approach has been employed in a recent study on the two-dimensional diffusion of a nascent peptide chain within the SecY channel.³⁸

In the stochastic case, a random number generator iterates the position of each diffusing particle based on the relevant probabilities given by \mathbf{T} . At a given time t , suppose a given particle is in cell I_i . The position of the particle at the next time step $t + \Delta t$ is chosen, through the use of the Matlab random number generator `rand`, to be I_j with probability T_{ji} . Subsequent iterations of these algorithmic steps propagate the particle along its trajectory during the simulation. The rest of this paper will be concerned with the stochastic method. This method is closely related to the Brownian dynamics method.^{2,39}

III. VALIDATION

In series 1 of the validation trials, the model was tested for the behavior of bulk diffusion of particles. Particles were initialized at the center of a spherically symmetric system of large enough radius that no particle reached the

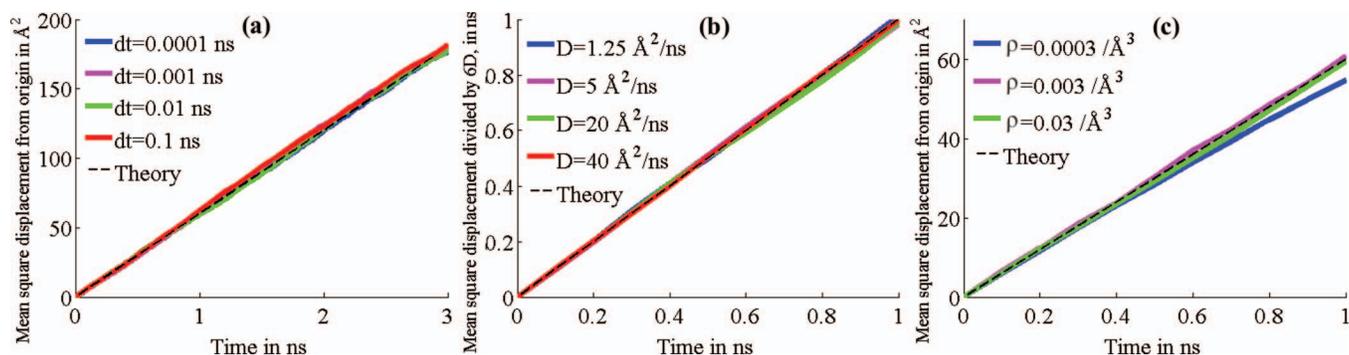


FIG. 2. Results for series 1 tests. Except for $dt = 0.0003/\text{\AA}^3$ in (c), graphs in each test set are almost identical and obscure each other as a result. (a) Mean square displacement for varying time steps. (b) Mean square displacement for varying diffusion coefficients. (c) Mean square displacement for varying grid densities.

boundary throughout the duration of each trial. Each particle was allowed to freely diffuse (corresponding to a potential $U(\mathbf{x}) = 0$) as its displacement was recorded over time. The mean square displacement of all the particles was then calculated as a function of time. The mean square displacement of a 3D diffusing particle from its initial position ($t = 0$) to its position at time t is $\langle \Delta x^2(t) \rangle = 6Dt$.

In set 1A, each trial had a sample size of 10^4 particles. The diffusion coefficient was set to $D = 10 \text{\AA}^2/\text{ns}$, the grid resolution was $\rho_1 = 0.03/\text{\AA}^3$ within radius 25\AA of the center, scaled linearly from 25\AA to 35\AA down to $\rho_2 = 0.003/\text{\AA}^3$ at which the density remains fixed up to the reflective system boundary at 60\AA . The time step was varied across values $dt = 0.0001, 0.001, 0.01, 0.1 \text{ ns}$. As seen in Fig. 2(a), the simulation agrees well with theory regardless of the time step used. A 10^4 -particle sample size was also used for set 1B, in which case the grid resolution was as in set 1A, $\rho_1 = 0.03/\text{\AA}^3$ and $\rho_2 = 0.003/\text{\AA}^3$, the time step was set to $dt = 0.1 \text{ ns}$, while the diffusion coefficient was varied across values $D = 1.25, 5, 10, 40 \text{\AA}^2/\text{ns}$. The results presented in Fig. 2(b) show that the simulation is also accurate across different values of the diffusion coefficient. In set 1C, the parameters used were sample size of 10^4 , $D = 10 \text{\AA}^2/\text{ns}$, $dt = 0.1 \text{ ns}$, $\rho_2 = 0.0003/\text{\AA}^3$, while the variable parameter was $\rho_1 = 0.0003, 0.003, 0.03/\text{\AA}^3$. The results shown in Fig. 2(c) show that the simulated behavior of bulk diffusion was robust over a wide range of grid resolutions, breaking down only at low grid densities on the order of $10^{-4}/\text{\AA}^3$ or less.

In series 2, absorptive and reflective boundary conditions were imposed: cells on the spherical shell of radius $r_1 = 10 \text{\AA}$ were assigned absorptive faces, while cells on the shell of radius $r_2 = 30 \text{\AA}$ were assigned reflective faces. Figure 3(c) shows the geometry of the system. Particles were initialized in cells with centers within 0.5\AA of the shell of radius $r_i = 20 \text{\AA}$, and then allowed to diffuse until all particles had been absorbed. The particle count was tracked as a function of time and compared against the theoretical behavior, given by Eq. (A26c) derived in Appendix A, and found in Carslaw and Jaeger⁴⁰ (2nd ed., Eqs. (12)–(15) on p. 367).

In set 2A, the sample size was 10^4 for each trial, the diffusion coefficient $D = 150 \text{\AA}^2/\text{ns}$, the grid density $\rho_1 = 0.5/\text{\AA}^3$ within radius 15\AA of the center and $\rho_2 = 0.25/\text{\AA}^3$ from 20\AA to the system boundary at 35\AA , with the value scaling down linearly between 15\AA and 20\AA ; the time step was varied across values $dt = 0.0002, 0.002, 0.02 \text{ ns}$. The results shown in Fig. 3 closely approximate the results from the analytical solutions with differences in time step size resulting in negligible differences in results on the time evolution of particle number. In set 2B, the sample size was 10^4 , $D = 150 \text{\AA}^2/\text{ns}$, $dt = 0.02 \text{ ns}$, and grid density was varied across values $\{\rho_1, \rho_2\} = \{0.1/\text{\AA}^3, 0.05/\text{\AA}^3\}$, $\{0.1/\text{\AA}^3, 0.25/\text{\AA}^3\}$, $\{0.5/\text{\AA}^3, 0.05/\text{\AA}^3\}$, $\{0.5/\text{\AA}^3, 0.25/\text{\AA}^3\}$.

In a final series of trials, the system and parameters used in sets 3A and 3B were the same as those in sets 2A and 2B. However, a linear potential $U(\mathbf{x}) = \alpha||\mathbf{x}||$ was imposed, with $\alpha = 0.2 k_B T/\text{\AA}$. The results for the respective sets are shown

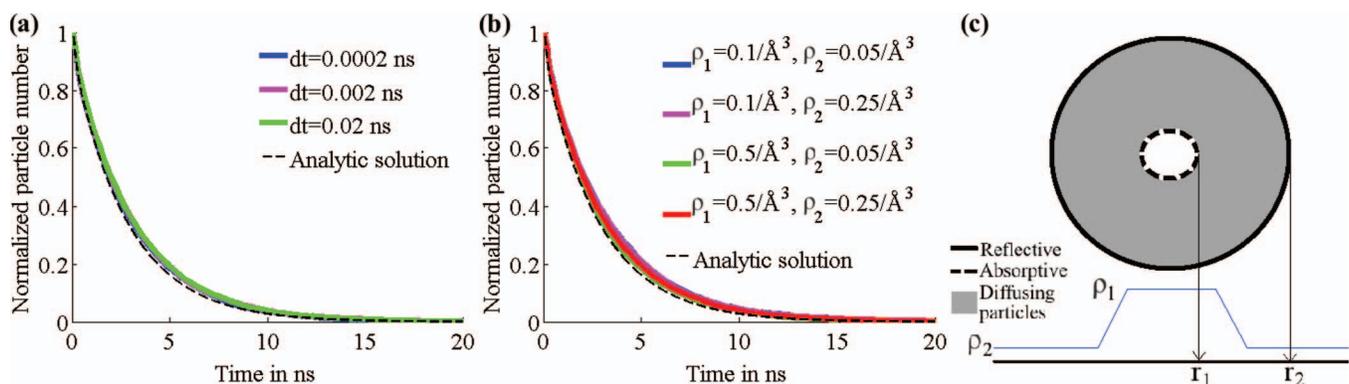


FIG. 3. Results for series 2 tests. Graphs in each test set are almost identical and obscure each other as a result. (a) Particle number for varying time steps. (b) Particle number for varying grid densities. (c) (Top) Schematic of system used for series 2; (bottom) radial profile of grid density assumed.

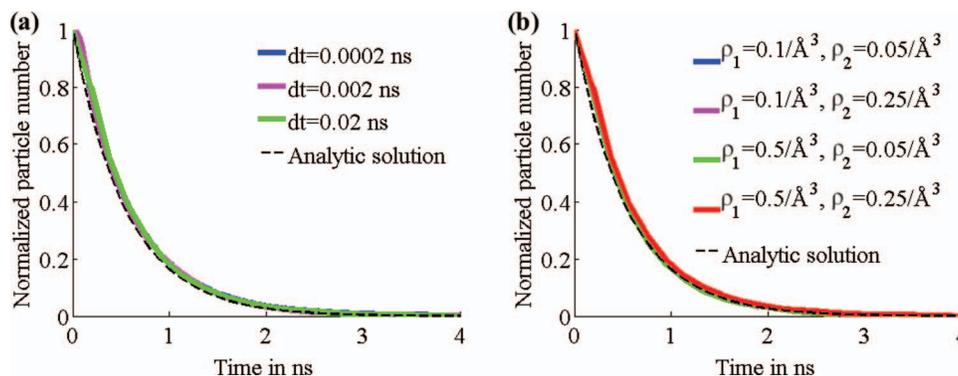


FIG. 4. Results for series 3 tests. Graphs in each test set are almost identical and obscure each other as a result. (a) Particle number for varying time steps. (b) Particle number for varying grid densities.

in Fig. 4 and compared with the results from the analytic solution, given by Eq. (A27) in Appendix A. Again, the numerical results are accurate over the range of values of parameters tested.

The time step-independence of the results in Figs. 2–4 is not unexpected, since the solution (35) of the restricted matrix is exact. In theory, the accuracy of the MSM description decreases with a decrease in the size of the time step³² because of the implicit assumption that the equilibration time of the particle within the cell at the beginning of each time step is negligible. The implication for our model is that accuracy is independent of time step size above a certain minimum time step size, subject to other parameter values, in particular the radius of the restricted region r_{restrict} , which must be set large enough that the particle's movement is not significantly hindered by the artificial boundaries. The other source of inaccuracy is error due to discretization as one assumes a uniform distribution in the grid cells. This error may be detected by varying either the rate at which particles move from cell to cell or by varying the grid resolution. In this regard, the results of set 1C suggest that boundary interactions are a more significant source of discretization error than is bulk diffusion.

IV. COMPUTATIONAL EFFICIENCY

The simulation process applied in our numerical solutions can be divided into three major phases according to computational expense: discretization of the system, calculation, and solution of the rate matrix, and Brownian motion algorithm for the diffusive displacement of particles. In the discretization phase, the most expensive task to be performed is the adaptation of cell centers, which requires updating N positions in S adaptation steps. Hence, the complexity for the first phase goes as $O(N \cdot S)$.

In the second phase, the bottleneck occurs during the solution of the rate matrix, more specifically during the calculation of eigenvectors and eigenvalues for the neighborhood within radius $r_{\text{restrict}} = 2r_{\Delta t}$ of each cell center \mathbf{w}_i . Given the average density ρ , the approximate number of cells within each neighborhood is $\frac{4}{3}\pi(2r_{\Delta t})^3\rho = \frac{32}{3}\pi\rho(6D\Delta t)^{3/2}$. A modest ballpark estimate for the complexity of eigenvector expansion is $O(n^3)$. For N cells, this estimate gives $O(N\rho^3(D\Delta t)^{9/2})$ complexity for the eigenvector expansion

phase. By this estimate, we justify the algorithmic step of solving restricted matrices as opposed to solving the entire rate matrix: depending on the characteristics of the system and grid chosen, it is often the case that the $O(N^3)$ complexity of solving the entire matrix overshadows that of solving the restricted matrices.

In the final phase, given particle number M and a total simulation time t_{total} , the complexity is $O(M \cdot t_{\text{total}}/\Delta t)$.

The computationally most expensive phase is typically the second one. Fortunately, the calculations for each cell neighborhood can be performed independently of those for other neighborhoods, and, thus, are amenable to parallelization. Furthermore, eigenvector expansion algorithms themselves can sometimes be parallelized. Hence, there is much potential for the reduction of the overall computation time needed.

V. REALISTIC APPLICATION: ION DIFFUSION THROUGH THE MECHANOSENSITIVE CHANNEL OF SMALL CONDUCTANCE

Mechanosensitive channels of small conductance (MscS) are a class of membrane channels that are gated by membrane tension. One such channel in *Escherichia coli* (ecMscS) is part of the bacterium's mechanism of coping with osmotic stress. The increased tension in the membrane during osmotic stress activates ecMscS, allowing the passive efflux of cytoplasmic solutes, thereby mitigating a potentially fatal buildup of osmotic pressure.³

ecMscS, shown in Fig. 5, is a heptamer of identical subunits, which form a trans-membrane domain and a cytoplasmic domain (CD). The CD is shaped like a balloon with seven lateral openings corresponding to its sevenfold symmetry. The lateral openings are just large enough for ions, the key osmolytes in *E. coli* cells, to pass through. The prevalent positive ions in *E. coli* cells are K^+ and Na^+ , while the main negative ion is actually the amino acid glutamate, Glu^- .³ An additional opening exists on the end of the CD distal to the membrane, but this opening is too narrow and hydrophobic for ions to pass through.

The function of the ecMscS CD is not well-understood despite its considerable size and large fraction of protein mass that goes into it. It is worth noting that CDs are a

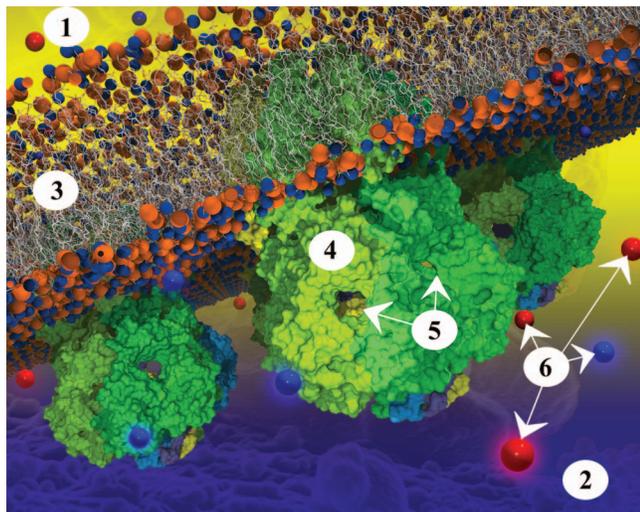


FIG. 5. Three ecMscS channels embedded in a membrane. ecMscS is composed of seven identical subunits (each shown in a different color). ecMscS (shown in shades of green) has a transmembrane domain (faintly visible in the membrane in case of the ecMscS closest to the viewer), forming a channel that opens and closes in the presence or absence of significant osmotic pressure across the cell membrane. ecMscS also has a large extra-membrane domain pointing into the cell interior, the cytoplasm. This domain of ecMscS is called the cytoplasmic domain and is prominently visible in the figure for all three ecMscS proteins. The membrane is a bilayer of lipids; lipids are composed of head groups (shown as orange and blue spheres) and tails (shown as white lines). Ions (positive ions shown as blue spheres, negative ions as red spheres) diffuse through the bulk solvent, the cytoplasm, as seen here below the membrane. Ions enter the ecMscS cytoplasmic domains through the side openings into the domain interior and, in case of osmotic stress having induced an opening of the ecMscS trans-membrane channel, pass through the channel towards the space outside of the cell, the periplasm. The intricate geometry of the cytoplasmic domain, a roughly spherical interior connected to the cytosol through seven narrow openings, plays a determining role in the manner in which ions leave an osmotically challenged cell. The labels shown on the figure correspond to: 1. Periplasm; 2. cytoplasm; 3. membrane; 4. cytoplasmic domain of ecMscS; 5. side openings; 6. ions.

ubiquitous feature of ion channels, present in both Kv^{41–43} and Kir^{44–46} classes of potassium channels, some members of the family of voltage-gated sodium channels,⁴⁷ the sodium-potassium pump,⁴⁸ the CIC chloride channel,⁴⁹ the nicotinic acetylcholine receptor and its homologs,^{50,51} and the family of mechanosensitive channels.^{52,53} In most of these channels, the role of the CD has only in recent years become the subject of many investigative efforts,^{3,54–61} as the importance of the respective CDs' roles in channel function becomes more evident. In particular, CDs are believed to play a role in regulating the diffusion of ions through the pores of certain channels.

In the case of ecMscS, one postulate is that the CD plays a role in gating and in this case would necessarily undergo a large conformational change between the open and closed states of the channel.^{62–64} It has also been suggested that the CD stabilizes the structure of the channel.^{65,66} Another postulate is that the CD acts as a molecular filter that minimizes the loss of Glu[−] solutes.⁵³ Such a filter may also encourage the efflux of cations and anions in pairs such that the efflux is electrically neutral, so as to maintain the cellular membrane potential. More recent studies compared ecMscS with channels exhibiting a similar CD structure, namely, bacterial cyclic nucleotide-gated (bCNG) channels, MscS-Like proteins of *Arabidopsis thaliana* (MSL10) and *Thermoanaer-*

obacter tengcongensis MscS (TtMscS). bCNG channels display slight or no mechanosensitive gating response,⁶⁷ suggesting that the CD plays at most a limited role in channel gating; the fact that MSL10⁶⁸ and TtMscS⁶⁹ are highly anion-selective casts doubt on the view that the CD enforces current neutrality.

Previous attempts to simulate MscS function using MD and Monte Carlo methods showed high selectivity for anions.²⁴ These simulation results run counter to experimental measurements that indicate a much lower selectivity.^{70–72} Barring inaccuracies associated with the simulation method, the results of experiment and simulation may be reconciled if collective inter-ion interactions occur over time scales beyond the reach of the previous simulation methods used ($\sim 10 \mu\text{s}$), that compensate for the channel's bias towards anions. Apart from such interactions, the diffusive approach of the ions to the channel may also play an important role in the description of ion efflux, for example, if the time scale of the diffusive approach is longer than the time scale of passage through the channel. Simulated ion channel systems are typically not large enough to take into account the diffusive approach, which occurs over length scales of 10–100 nm. Addressing the needs for such long time and large length scales requires the simulation of ecMscS in a large box and necessitates the present method. We note that one can use a lower grid density to describe the large regions further away from the channel and a higher grid density for the channel and its vicinity.

However, since the current version of the present method treats particles independently, we cannot yet address the question of ion-ion interactions. Hence, the purpose of this section is not to obtain new results on MscS function. Instead, our simulation of ecMscS serves as a demonstration of the implementation of the method in an actual application, as well as to show that the results obtained are consistent with previous studies of the same system using the BioMOCA software and MD.²⁴ Future modifications of the presented methodology to include inter-ionic interactions will address multi-ion conduction through the ecMscS CD and membrane channel.

A. Setup of molecular dynamics simulation

The system is described through an all-atom MD simulation with ecMscS embedded in the center of a $320 \text{ \AA} \times 320 \text{ \AA}$ POPC membrane patch, immersed in a waterbox of dimension $316 \text{ \AA} \times 317 \text{ \AA} \times 230 \text{ \AA}$. Ions are placed in the solvent in numbers according to physiological ion strengths [K^+], [Glu^-], and [Cl^-], such that the system is electrically neutral. The system is minimized and equilibrated in the presence of an electric field as described before²⁴ so as to widen the ecMscS pore relative to the opening seen in the crystal structure.⁵³ Full details of the all-atom system setup and simulation parameters are furnished in Appendix B.

PMF maps of the system for both K^+ and Glu^- were extracted from a 240-ns equilibration run. For the extraction the applied electric field was removed and the backbone of ecMscS was harmonically restrained to prevent the pore from closing. The distributions of K^+ and Glu^- ions were averaged over the entire course of the run. Regions where the distribution went to zero were assigned a minimal non-zero

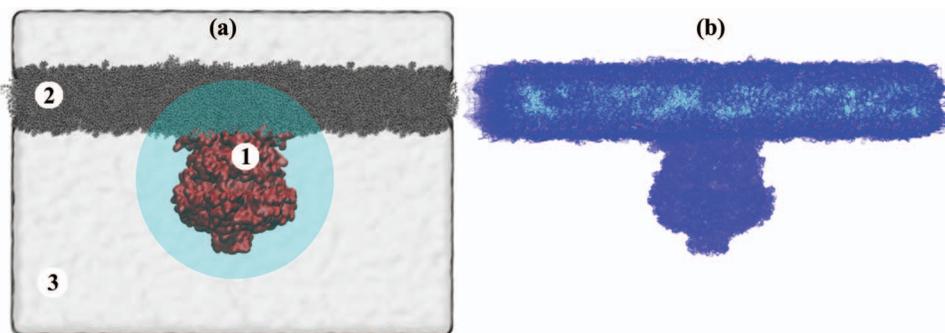


FIG. 6. All-atom system simulated for the present study. (a) ecMscS (1) embedded in a membrane (2) and solvated in a waterbox (3). The highlighted circle represents a locus of 70 Å about the origin, in which a high grid density is used to model ion diffusion in detail near ecMscS. (b) Cells from the resulting grid that are associated with high PMF values and, hence, modeled through reflective barriers.

distribution value, in order to prevent singularities from occurring when taking logarithms in the next step: the logarithm of the averaged distribution map, after normalization, gives the PMF map in units of $k_B T$.

The diffusion coefficients of K^+ and Glu^- were assumed to be constant in space. The average value of each coefficient was obtained from trajectories arising from the MD simulation described above. The trajectories were divided into 0.02 ns intervals. In each interval, the mean square displacement of each ionic species between the beginning and the end of the interval was measured. The mean square displacement values were then averaged over all intervals. Thus, the diffusion coefficient D for each ionic species was calculated from the relation $\langle \Delta x^2(t) \rangle = 6Dt$. The diffusion coefficients were found to assume the values $D_K = 200 \text{ \AA}^2/\text{ns}$ and $D_{Glu} = 75 \text{ \AA}^2/\text{ns}$, which are in close agreement with the experimentally determined values^{73,74} of $D_K^{\text{expt}} = 196 \text{ \AA}^2/\text{ns}$ and $D_{Glu}^{\text{expt}} = 75 \text{ \AA}^2/\text{ns}$ (both measured in bulk solvent), respectively.

A grid representing the discretized system was built, with density $\rho_1 = 0.05/\text{\AA}^3$ within a radius of 70 Å (large enough to encapsulate the ecMscS, as shown in Fig. 6(a), such that the center of the CD coincides with the center of the grid) and $\rho_2 = 0.01/\text{\AA}^3$ outside of a radius of 80 Å with the center of the CD being characterized through zero radius. The density was adapted linearly with the radius between 70 Å and 80 Å. The PMF map, obtained as values on a Cartesian grid, was cubic-interpolated to assign a PMF value to each cell center. Grid cells with the maximum PMF value (shown in Fig. 6(b)), corresponding to regions where the ion distribution is zero, were designated as reflective boundaries. Consequently, these cells were excluded from the rate matrix calculation and solution, thus reducing the total computational cost; correspondingly, cells on the boundary of the system were assigned reflective surfaces.

TABLE I. Number of conduction events for each ion for different biasing voltages.

Ion	0 mV	+100 mV	-100 mV
K^+	222	1992	34
Glu^-	217	54	20

For a time step of $dt = 0.002$ ns, the rate and transition matrices were calculated, as outlined in Sec. II C.

B. Simulation procedure and results

Particles were initialized in randomly chosen cells on the cytoplasmic side. The random selection was performed with weights proportional to the volume of each cell, so that the particles were initially uniformly distributed. The particles were then allowed to diffuse. When a particle crosses via ecMscS from the cytoplasmic side to the periplasmic side of the membrane, it is assumed to diffuse away from the membrane and, accordingly, is removed from the system. For the purpose of assessing the effect of transmembrane potential biases, we conducted the prior MD simulations with positive and negative biases, in addition to a zero bias simulation. We choose the strengths of the voltage bias following the study of Sotomayor *et al.*,⁷⁵ which we will use as a reference for comparison. For each voltage bias, an MD simulation was carried out to obtain the requisite PMF map. The PMF maps were each applied to the present method as described below.

The simulation procedure was repeated for six runs – two ion species, each with bias voltages 0 mV, +100 mV, -100 mV (measured from the cytoplasmic side). For each simulation, the initial particle count was 5000, and the total simulation time was 4 μ s. The number of conduction events for each run is listed in Table I. Table II shows the results of the present study, scaled to match the initial particle concentration and simulation time of the reference study, together with the results for two putative open conformations of ecMscS in the reference study.⁷⁵

TABLE II. Comparison between present and reference study. Values for the present study have been scaled to account for the different initial particle concentration and simulation time in the reference study. The present study employed Glu^- ions, whereas the reference study employed Cl^- ions. The reference study employed two distinct putative open conformations of the channel. For each respective ion and bias, the results for both conformations are presented, separated by a comma.

Ion	Present			Reference		
	0 mV	+100 mV	-100 mV	0 mV	+100 mV	-100 mV
K^+	2	20	0	2, 5	13, 23	3, 0
Glu^-	2	0	0	22, 40	6, 17	53, 72

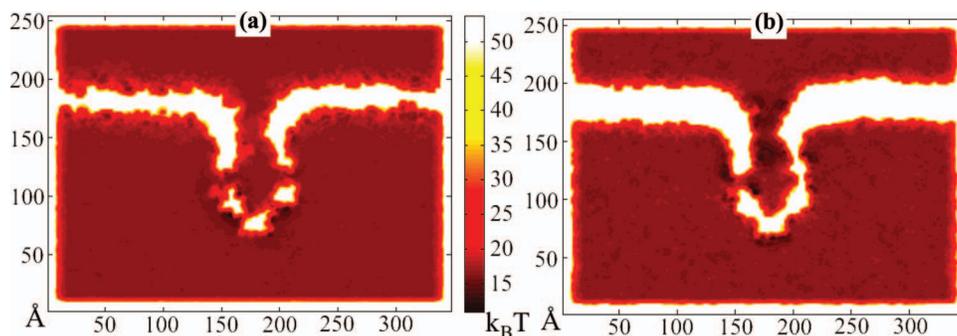


FIG. 7. (a) Cross-section of potential of mean force maps for K^+ ions. (b) Cross-section of potential of mean force maps for Glu^- ions. The energy landscape for ions exhibit higher, more unfavorable values for positive ions inside the cytoplasmic domain and, conversely, lower, more favorable values for negative ions. Indeed, in the molecular dynamics simulations for the three bias conditions tested, anions congregated in the cytoplasmic domain and sterically hindered the passage of cations.

In agreement with the reference study, there were few conduction events for K^+ with bias voltages 0 mV and -100 mV. For Glu^- , the numbers of events were much smaller than in the reference study where actually Cl^- ions were used.

VI. DISCUSSION OF RESULTS AND FUTURE DIRECTIONS

The results for K^+ in the present study agree with those of the reference study, while those of Glu^- differ significantly. The significantly lower event count for the anion is attributed to the fact that the present study employed bulky Glu^- ions as compared to the Cl^- ions employed in the reference study. Glu^- ions are actually the prevalent negative ions in *E. coli*, which is why they have been employed here. During the MD calculations, not only would the bulk diffusion coefficient of the anion be lower in the present study than in the reference one as reported⁷⁶ (the diffusion coefficient of Cl^- is similar to that of K^+), but also the resulting steric exclusion in narrow regions around ecMscS results in higher potential barriers in the Glu^- PMF maps.

Other factors contribute to the discrepancies between the results of the present study and the reference study. An examination of the PMF maps shown in Fig. 7 reveals potential wells in the vicinity of the ecMscS structure. Ions congregating in these wells present an obstacle to other ions that would otherwise also enter the wells. However, in the absence of inter-ionic interactions, the lack of steric exclusion and local electrostatic interactions in the present version of the kinetic model allows the ions to all linger in the wells, substantially increasing the time taken to reach their designated targets. In fact, for the -100 mV bias, a well of -30 kcal/mol trapped Glu^- ions such that few conduction events arose (see Table I). Such an effect had been avoided in the reference study because inter-ionic interactions were included in the respective simulations.

The absence of steric effects in the kinetic simulation illustrates the issues arising from the absence of inter-ionic interactions in simulations, especially in regions where ions come into close proximity of one another, such as in the channel interior. Adding inter-ionic interactions in a manner that is both physically sensible and computationally feasible is diffi-

cult because such interactions are modulated by environmental factors as well as by the presence of more than two particles within interaction range.

In light of the challenging nature of an account of ion-ion interactions, we propose as a first step a naive solution. One starts by identifying local regions in the system with roughly similar environments. One such region might be the interior of the ecMscS structure, namely, the interior of the pore and the CD, and a second might be everywhere outside it. For each region, one determines then the pair correlation function $g(r)$ for the various ion pairs, $K^+ - K^+$, $Glu^- - Glu^-$, and $K^+ - Glu^-$, from the MD trajectories used for the PMF extraction. $g(r)$ can then be used to modulate the transition matrix probabilities of particles that move within a pre-set interaction range of each other.

Another issue of concern is the handling of diffusion coefficients. The assumption in the present study is that the diffusion coefficient for each ion species is constant throughout the system. This assumption was made for the sake of simplicity. However, one expects that the diffusion coefficient of glutamate in the crowded interior of the CD is very different from that in bulk solvent outside of the CD. The proposed remedy is to average over diffusion lengths of ions in local regions of the systems throughout the MD trajectories and from these lengths obtain the local diffusion coefficient in each grid cell.

It would also behoove us to ensure that the constant-value approximation of the diffusion coefficient, and other quantities for that matter, within each grid cell is valid. For that purpose, one could either interpolate the PMF and diffusion coefficient maps within the cell, or use a sufficiently fine grid to describe regions in which the maps vary sharply. The former would be difficult to implement within the present framework due to the complexity of the additional computation required. The latter can conceivably be a future addition to the algorithm that calculates, for each region, the grid density that resolves the local gradient of maps such that the error between the approximation and the actual quantities fall below a pre-set threshold. Since the PMF map describes the geometry of the system, such a scheme would also be a natural means of quantifying the suitability of grid density to the geometric intricacy of the system.

Since the utility of the present method relies greatly on its computational efficiency, it would be useful to consider alternative ways of solving the rate equation (18). In particular, one could consider employing a Chebyshev expansion⁷⁷ to approximate the solution

$$\mathbf{P}(t) = e^{\mathbf{R}t}. \quad (38)$$

Incorporating such a scheme into the solution of either the rate equation directly or of the local rate matrices one can compare the resulting efficiency with the present method.

VII. CONCLUSION

The kinetic model of diffusion described in the present study is a feasible means of simulating diffusion in cellular systems over a wide range of length and time scales. Results from validation tests show robust agreement with analytic descriptions over time step sizes and diffusion coefficients typical of biomolecular systems. Furthermore, the method is adaptable to a wide range of scientific needs and computational capabilities, through the adjustment of simulation parameters. The method can be made more efficient through parallelization of the algorithm and is viable for both deterministic and stochastic calculations.

Algorithmic benefits of the method include accuracy that increases with time step and the restriction of calculations to a local region around each state. These benefits were brought to bear in the ecMscS example, where a simulation on the length scale of hundreds of Ångströms and a time scale of 1 μ s was run serially and completed in two days. The ecMscS example produced agreement with the reference study in the case of K^+ . The examples presented also illustrate the weaknesses of our method, which serve as pointers for future development, namely, the inclusion of dielectric effects, the use of position-dependent diffusion coefficients and the inclusion of inter-ion interactions.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contribution of Fig. 5 by Chris Chipot and Ramya Gamini. This work is supported by Grant No. NIH 9P41GM104601 from the National Institutes of Health and Grant No. NSF PHY0822613 from the National Science Foundation. The authors gladly acknowledge supercomputer time provided via TeraGrid Resource Allocation Committee Grant No. MCA93S028.

APPENDIX A: ANALYTICAL AND NUMERICAL SOLUTIONS TO DESCRIBE SYSTEMS USED IN VALIDATION

1. Analytical solution for validation Sets 2A and 2B

Here, we provide the solution of the diffusion problem for test set 2. The solution is available from Carslaw and Jaeger⁴⁰ (2nd ed., Eqs. (12)–(15) on p. 367), but rather than explaining the terms and constants in the complex solution expression stated by these authors, we derive the solution here as an optimal means of communication to the reader not versed in diffusion theory.

The system is spherically symmetric, with particles diffusing in the space between two concentric spheres of radii r_1 and r_2 , as shown in Fig. 3(c). At time $t = 0$, particles are uniformly distributed on a spherical surface of radius r_i , with $r_1 < r_i < r_2$. The diffusion coefficient D is taken to be constant.

The free diffusion equation is given by

$$(\nabla^2 - D^{-1}\partial_t)p(\mathbf{r}, t) = 0. \quad (A1)$$

Using the ansatz

$$p(\mathbf{r}, t) = e^{\omega t} Y_l^m(\theta, \phi) R(r), \quad (A2)$$

one obtains for the radial dependence

$$\left[\partial_r^2 + \frac{2}{r} \partial_r - \frac{l(l+1)}{r^2} - \frac{\omega}{D} \right] R(r) = 0. \quad (A3)$$

The above equation is known as the spherical Bessel equation, the solutions of which substitute into Eq. (A2) to give the form

$$p(\mathbf{r}, t) = \sum_{k,m,l} e^{\omega t} Y_l^m(\theta, \phi) (A_{kl} j_l(kr) + B_{kl} n_l(kr)), \quad (A4)$$

where j_l and n_l are the spherical Bessel functions, k is defined through $k^2 = \frac{-\omega}{D}$ (ω is less than zero), $Y_l^m(\theta, \phi)$ represents a set of functions called the spherical harmonics, and A_{kl} and B_{kl} are constants.

Due to spherical symmetry, only the $l = 0$ term contributes to the solution, so that

$$p(\mathbf{r}, t) = \sum_k e^{\omega(k)t} R_k(r), \quad (A5)$$

where

$$R_k(r) = A_k j_0(kr) + B_k n_0(kr), \quad (A6)$$

$$j_0(kr) = \frac{\sin(kr)}{kr}, \quad (A7)$$

$$n_0(kr) = -\frac{\cos(kr)}{kr}, \quad (A8)$$

and A_k and B_k are constants to be determined.

In our description, we assume Dirichlet and Neumann boundary conditions at $r = r_1$ and $r = r_2$, respectively. At $r = r_1$ holds

$$R_k(r_1) = 0. \quad (A9)$$

For the convenience of calculation, a constant phase may be introduced without loss of generality

$$\sin(kr) \rightarrow \sin[k(r - r_1)], \quad (A10)$$

$$\cos(kr) \rightarrow \cos[k(r - r_1)]. \quad (A11)$$

Then, Eq. (A9) implies that $B_k = 0 \forall k \in \mathbb{R}$. Thus it holds

$$R_k(r) = A_k \frac{\sin[k(r - r_1)]}{kr}. \quad (A12)$$

At $r = r_2$, the Neumann boundary condition is assumed

$$\partial_r R_k(r)|_{r=r_2} = 0. \quad (A13)$$

This condition reads

$$-\frac{\sin[k(r_2 - r_1)]}{kr_2^2} + \frac{\cos[k(r_2 - r_1)]}{r_2} = 0 \quad (A14)$$

or

$$\tan[k(r_2 - r_1)] = kr_2. \quad (\text{A15})$$

The numerical solution of Eq. (A15) gives a countably infinite set of values of k . For the sake of clear notation, let n index these values. Accordingly, we re-index the following terms:

$$R_k(r) \rightarrow R_n(r), \quad (\text{A16})$$

$$\omega(k) \rightarrow \omega_n, \quad (\text{A17})$$

$$A_k \rightarrow A_n. \quad (\text{A18})$$

Defining the inner product in solution space as

$$\langle R_{n_1} | R_{n_2} \rangle = \int_{r_1}^{r_2} r^2 R_{n_1}(r) R_{n_2}(r) dr, \quad (\text{A19})$$

it can be verified that $\{R_n | n \in \mathbb{Z}\}$ is an orthogonal set. The normalization factor is obtained by evaluating

$$A_n^{-2} \langle R_n | R_n \rangle = (k_n)^{-2} \int_{r_1}^{r_2} \sin^2[k_n(r - r_1)] dr \quad (\text{A20a})$$

$$= \frac{1}{2k_n^2} \left(-r_1 + \frac{k_n^2 r_2^3}{1 + k_n^2 r_2^2} \right). \quad (\text{A20b})$$

Hence, the inner product can be explicitly written

$$\langle R_{n_1} | R_{n_2} \rangle = A_{n_1}^{-2} \frac{\delta_{n_1, n_2}}{2k_{n_1}^2} \left(-r_1 + \frac{k_{n_1}^2 r_2^3}{1 + k_{n_1}^2 r_2^2} \right). \quad (\text{A21})$$

The complete solution is then

$$p(\mathbf{r}, t) = \sum_n e^{\omega_n t} R_n(r), \quad (\text{A22})$$

where $\omega_n = -k_n^2 D$.

In the case of the assumed initial condition $p_0(\mathbf{r}) = (4\pi r_i^2)^{-1} \delta(r - r_i)$ follows:

$$\sum_n R_n(r) = (4\pi r_i^2)^{-1} \delta(r - r_i). \quad (\text{A23})$$

Taking the inner product gives

$$\langle R_m | R_m \rangle = (4\pi r_i^2)^{-1} \int_{r_1}^{r_2} r^2 R_m(r) \delta(r - r_i) dr, \quad (\text{A24})$$

from which follows:

$$A_m = A_m (4\pi \langle R_m | R_m \rangle)^{-1} R_m(r_i) \quad (\text{A25a})$$

$$= \frac{k_m^2}{2\pi} \left(-r_1 + \frac{k_m^2 r_2^3}{1 + k_m^2 r_2^2} \right)^{-1} \frac{\sin[k_m(r_i - r_1)]}{k_m r_i}. \quad (\text{A25b})$$

Finally, the expression of the number of surviving particles as a function of time is

$$\Sigma(t) = \int_{r_1}^{r_2} dr 4\pi r^2 p(r, t) \quad (\text{A26a})$$

$$= 4\pi \int_{r_1}^{r_2} dr \sum_n A_n r e^{-k_n^2 D t} \frac{\sin[k_n(r - r_1)]}{k_n} \quad (\text{A26b})$$

$$= 4\pi \sum_n e^{-k_n^2 D t} \frac{A_n r_1}{k_n^2}. \quad (\text{A26c})$$

2. Numerical mean first passage time description for validation sets 3A and 3B

In the case of a linear potential $U(r) = \alpha r$, a closed-form analytic solution for the time-dependent diffusion probability does not exist. Instead, the time dependence of the probability distribution was captured for the test purpose by the mean first passage time approximation.⁷⁸ In this approximation, the normalized surviving particle count is given by

$$\Sigma(t) = e^{-t/\tau(r_i)}, \quad (\text{A27})$$

where $\tau(r_i)$ is the mean first passage time given by the expression

$$\tau(r_i) = \int_{r_1}^{r_i} dr (Dr^2)^{-1} e^{\alpha r} \int_r^{r_2} dr' r'^2 e^{-\alpha r'}. \quad (\text{A28})$$

Here $\tau(r_i)$ is evaluated through numerical integration.

The mean passage time description through Eqs. (A27) and (A28) is known to be a good approximation to the decay of the total probability of still unreacted particles⁷⁸ and, hence, Eqs. (A27) and (A28) can serve as a test of the numerical scheme suggested in the present study.

APPENDIX B: SPECIFICATIONS FOR MD SIMULATION OF ecMscS SYSTEM

The structure of ecMscS, solved through x-ray crystallography,⁵³ was taken from the protein data bank (PDB 1MXM). The psfgen plugin of VMD 1.9.1⁷⁹ was used to add missing hydrogens to the structure. Asp, Glu, Lys, and Arg were modeled as charged residues. The structure was then aligned with a 320 Å × 320 Å POPC membrane patch such that the transmembrane domain was superimposed on the center region of the patch; the overlapping lipids were deleted. Subsequently, the Solvate plugin in VMD was employed to immerse the protein-embedded membrane in a waterbox of dimension 316 Å × 317 × 230 Å and the cIionize plugin of VMD was employed to place K⁺ and Cl⁻ ions in the solvent, such that the concentration of each ion was 200 mM, the physiological concentration of K⁺ in the *E. coli* cytosol. The physiological concentration of Cl⁻ is on the order of 10 mM,⁸⁰ while that of Glu⁻ has been reported to vary over a wide range of values.⁸¹ Hence, enough Cl⁻ ions were kept in the system to give 10 mM concentration, while the rest were mutated into Glu⁻ zwitterions. Since ecMscS has a net charge of +28e, 28 K⁺ ions were deleted to neutralize the system. Due to Glu⁻ being larger in size than Cl⁻, it was necessary to delete water molecules that overlapped with the Glu⁻ zwitterions. The final system contained 610 961 water molecules, 2997 lipids, 2307 K⁺ ions, 100 Cl⁻ ions, and 2235 Glu⁻ zwitterions. The total atom count of the system was 2 304 943.

MD simulations were run with NAMD 2.9,⁸² using the TIP3P water model,⁸³ and with the CHARMM36⁸⁴ and CMAP-corrected CHARMM22^{85,86} force fields for lipids and non-lipids, respectively. The time step was set to 2 fs. The particle mesh-Ewald method was used to calculate long-range electrostatic forces, with a mesh density of 1/Å³. Van der Waals forces were calculated with a cut-off of 12 Å and a switching function starting at 10 Å. Periodic boundary

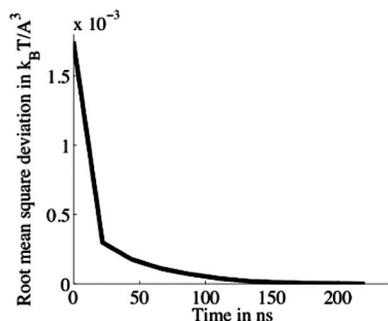


FIG. 8. Root mean square deviation of the sampled PMF from the final map in a $84\text{Å} \times 84 \times 120\text{Å}$ box enclosing the ecMscS channel.

conditions were imposed in all MD runs. Temperature was held at 300 K via Langevin dynamics with a damping coefficient of 1 ps^{-1} . Pressure was held at 1 atm using the Nosé-Hoover Langevin piston method with damping time of 50 fs and period of 200 fs.

The system was minimized over 3.4×10^5 steps. All atoms, except for those of the lipid tails, were fixed during the subsequent 4.6 ns equilibration to allow the lipid tails to melt. Lipid head group constraints were then released and equilibration continued for another 10 ns, to allow the membrane to form a watertight seal around the ecMscS pore. The system was then put through 10 cycles of alternating runs, the first being un-constrained equilibration with voltage 0.6 V across the membrane (measured from the cytoplasmic side), and the other runs being equilibration, with no applied voltage, and with ecMscS backbone atoms being harmonically constrained with spring constant 2 kcal/mol/Å^2 to their last positions in the previous run. The runs with applied voltage induced widening of the ecMscS pore,²⁴ while the runs with constraints on the ecMscS backbone allowed the membrane to relax back to a stable state so that the strong voltage did not break up the membrane before the ecMscS pore widened. The resulting structure was used for the production run, during which protein backbone atoms were again put under harmonic restraint with spring constant 2 kcal/mol/Å^2 , in order to maintain the widened pore width in the absence of an applied voltage.

MD runs for sampling the PMF and other properties were carried out for 240 ns, ensuring adequate sampling as shown in the plot of the convergence metric used in Fig. 8. The convergence metric is defined as the root mean square deviation of the potential map at 20-ns intervals from the final map during the run. Since the interior of ecMscS is the least accessible, and hence the least sampled, region of the system, we confine the convergence metric calculation to a box which just encloses the protein.

¹R. Peters, A. Brünger, and K. Schulten, *Proc. Natl. Acad. Sci. U.S.A.* **78**, 962 (1981).

²G. Lamm and K. Schulten, *J. Chem. Phys.* **75**, 365 (1981).

³R. Gamini, M. Sotomayor, C. Chipot, and K. Schulten, *Biophys. J.* **101**, 80 (2011).

⁴H. C. Berg and E. M. Purcell, *Biophys. J.* **20**, 193 (1977).

⁵D. Shoup and A. Szabo, *Biophys. J.* **40**, 33 (1982).

⁶K. Sharp, R. Fine, K. Schulten, and B. Honig, *Phys. Chem.* **91**, 3624 (1987).

⁷R. Zwanzig and A. Szabo, *Biophys. J.* **60**, 671 (1991).

⁸R. R. Gabdouliline and R. C. Wade, *Curr. Opin. Struct. Biol.* **12**, 204 (2002).

⁹A. Atri, J. Amundson, D. Clapham, and J. Sneyd, *Biophys. J.* **65**, 1727 (1993).

¹⁰J. Elf, A. Dončić, and M. Ehrenberg, *Proc. SPIE* **5110**, 114 (2003).

¹¹K.-H. Chiam, C. M. Tan, V. Bhargava, and G. Rajagopal, *Phys. Rev. E* **74**, 051910 (2006).

¹²A. Camilli and B. L. Bassler, *Science* **311**, 1113 (2006).

¹³J. Sneyd, B. T. Wetton, A. C. Charles, and M. J. Sanderson, *Am. J. Physiol.: Cell Physiol.* **268**, C1537 (1995).

¹⁴S. Y. Shvartsman, H. S. Wiley, W. M. Deen, and D. A. Lauffenburger, *Biophys. J.* **81**, 1854 (2001).

¹⁵W. Nadler and K. Schulten, *J. Chem. Phys.* **82**, 151 (1985).

¹⁶S. S. Andrews, N. J. Addy, R. Brent, and A. P. Arkin, *PLOS Comput. Biol.* **6**, e1000705 (2010).

¹⁷J. R. Stiles and T. M. Bartol, *Computational Neuroscience: Realistic Modeling for Experimentalists*, edited by E. D. Schutter (CRC Press, 2001), pp. 87–127.

¹⁸A. E. Cowan, I. I. Moraru, J. C. Schaff, B. M. Slepchenko, and L. M. Loew, *Methods Cell Biol.* **110**, 195 (2012).

¹⁹A. V. Popov and N. Agmon, *J. Chem. Phys.* **115**, 8921 (2001).

²⁰J. S. van Zon and P. R. ten Wolde, *J. Chem. Phys.* **123**, 234910 (2005).

²¹J. S. van Zon and P. R. ten Wolde, *Phys. Rev. Lett.* **94**, 128103 (2005).

²²M. Compoint, F. Picaud, C. Ramseyer, and C. Giradet, *J. Chem. Phys.* **122**, 134707 (2005).

²³T. W. Allen, O. S. Andersen, and B. Roux, *Biophys. J.* **90**, 3447 (2006).

²⁴M. Sotomayor, V. Vasquez, E. Perozo, and K. Schulten, *Biophys. J.* **92**, 886 (2007).

²⁵R. Elber, *Curr. Opin. Struct. Biol.* **15**, 151 (2005).

²⁶D. L. Ermak, *J. Chem. Phys.* **62**, 4189 (1975).

²⁷S. R. McGuffee and A. H. Elcock, *J. Am. Chem. Soc.* **128**, 12098 (2006).

²⁸T. Frembgen-Kesner and A. H. Elcock, *J. Chem. Theory Comput.* **5**, 242 (2009).

²⁹T. Ando and J. Skolnick, *Proc. Natl. Acad. Sci. U.S.A.* **107**, 18457 (2010).

³⁰T. van der Straaten, G. Kathawala, and U. Ravaioli, *J. Comput. Electron.* **2**, 231 (2003).

³¹P. Bauler, G. A. Huber, and J. A. McCammon, *J. Chem. Phys.* **136**, 164107 (2012).

³²J.-H. Prinz, H. Wu, M. Sarich, B. Keller, M. Senne, M. Held, J. D. Chodera, C. Schütte, and F. Noé, *J. Chem. Phys.* **134**, 174105 (2011).

³³MATLAB, version 7.10.0 (R2010a), The MathWorks Inc., Natick, MA, 2010.

³⁴H. Ritter, T. Martinetz, and K. Schulten, *Neural Computation and Self-Organizing Maps: An Introduction*, revised English ed. (Addison-Wesley, New York, 1992).

³⁵T. M. Martinetz, S. G. Berkovich, and K. Schulten, *IEEE Trans. Neural Netw.* **4**, 558 (1993).

³⁶T. Martinetz and K. Schulten, *Neural Networks* **7**, 507 (1994).

³⁷Q. Du, V. Faber, and M. Gunzburger, *SIAM Rev.* **41**, 637 (1999).

³⁸J. C. Gumbart, I. Teo, B. Roux, and K. Schulten, *J. Am. Chem. Soc.* **135**, 2291 (2013).

³⁹J. Madura, J. Briggs, R. Wade, M. Davis, B. Luty, A. Illin, J. Antosiewicz, M. Gilson, B. Bagheri, L. Scott, and J. A. McCammon, *Comput. Phys. Commun.* **91**, 57 (1995).

⁴⁰H. S. Carslaw and J. C. Jaeger, *Conduction of Heat in Solids* (Oxford University Press, London, 1959), pp. 367–368.

⁴¹L. W. Tu, V. Santarelli, and C. Deutsch, *Biophys. J.* **68**, 147 (1995).

⁴²Y. Jiang, A. Lee, J. Chen, V. Ruta, M. Cadene, B. T. Chait, and R. MacKinnon, *Nature (London)* **423**, 33 (2003).

⁴³S. B. Long, E. B. Campbell, and R. MacKinnon, *Science* **309**, 897 (2005).

⁴⁴A. Kuo, J. M. Gulbis, J. F. Antcliff, T. Rahman, E. D. Lowe, J. Zimmer, J. Cuthbertson, F. M. Ashcroft, T. Ezaki, and D. A. Doyle, *Science* **300**, 1922 (2003).

⁴⁵X. Tao, J. L. Avalos, J. Chen, and R. MacKinnon, *Science* **326**, 1668 (2009).

⁴⁶S. Uysal, V. Vásquez, V. Tereshko, K. Esaki, F. A. Fellouse, S. D. Sidhu, S. Koide, E. Perozo, and A. Kossiakoff, *Proc. Natl. Acad. Sci. U.S.A.* **106**, 6644 (2009).

⁴⁷F. H. Yu and W. A. Catterall, *Genome Biol.* **4**, 207 (2003).

⁴⁸T. Shinoda, H. Ogawa, F. Cornelius, and C. Toyoshima, *Nat. Lett.* **459**, 446 (2009).

⁴⁹R. Dutzler, E. B. Campbell, M. Cadene, B. T. Chait, and R. MacKinnon, *Nature (London)* **415**, 287 (2002).

⁵⁰A. Karlin, *Nat. Rev. Neurosci.* **3**, 102 (2002).

- ⁵¹P. H. Celie, I. E. Kasheverov, D. Y. Mordvintsev, R. C. Hogg, P. van Nierop, R. van Elk, S. E. van Rossum-Fikkert, M. N. Zhmak, D. Bertrand, V. Tsetlin, T. K. Sixma, and A. B. Smit, *Nat. Struct. Mol. Biol.* **12**, 582 (2005).
- ⁵²G. Chang, R. H. Spencer, A. T. Lee, M. T. Barclay, and D. C. Rees, *Science* **282**, 2220 (1998).
- ⁵³R. B. Bass, P. Strop, M. Barclay, and D. C. Rees, *Science* **298**, 1582 (2002).
- ⁵⁴S. Pegan, C. Arrabit, W. Zhou, W. Kwiatkowski, A. Collins, P. A. Slesinger, and S. Choe, *Nat. Neurosci.* **8**, 279 (2005).
- ⁵⁵W. Han, S. Nattel, T. Noguchi, and A. Shrier, *J. Biol. Chem.* **281**, 27134 (2006).
- ⁵⁶V. R. Schack, J. P. Morth, M. S. Toustrup-Jensen, A. N. Anthonisen, P. Nissen, J. P. Andersen, and B. Vilsen, *J. Biol. Chem.* **283**, 27982 (2008).
- ⁵⁷F. Potet, B. Chagot, M. Anghelescu, P. C. Viswanathan, S. Z. Stepanovic, S. Kupersmidt, W. J. Chazin, and J. R. Balsler, *J. Biol. Chem.* **284**, 8846 (2009).
- ⁵⁸D. Wray, *Eur. Biophys. J.* **38**, 285 (2009).
- ⁵⁹L. M. Sharkey, X. Cheng, V. Drews, D. A. Buchner, J. M. Jones, M. J. Justice, S. G. Waxman, S. D. Dib-Hajj, and M. H. Meisler, *J. Neurosci.* **29**, 2733 (2009).
- ⁶⁰M. Hirano, Y. Takeuchi, T. Aoki, T. Yanagida, and T. Ide, *J. Biol. Chem.* **285**, 3777 (2010).
- ⁶¹M. Jin and P. Lu, *Neural Regen. Res.* **6**, 553 (2011).
- ⁶²M. D. Edwards, I. R. Booth, and S. Miller, *Curr. Opin. Microbiol.* **7**, 163 (2004).
- ⁶³P. Koprowski and A. Kubalski, *J. Biol. Chem.* **278**, 11237 (2003).
- ⁶⁴W. Grajkowski, A. Kubalski, and P. Koprowski, *Biophys. J.* **88**, 3050 (2005).
- ⁶⁵S. Miller, W. Bartlett, S. Chandrasekaran, S. Simpson, M. Edwards, and I. R. Booth, *EMBO J.* **22**, 36 (2003).
- ⁶⁶U. Schumann, M. Edwards, C. Li, and I. R. Booth, *FEBS Lett.* **572**, 233 (2004).
- ⁶⁷H. R. Malcolm, Y. Y. Heo, D. B. Caldwell, J. K. McConnell, J. F. Hawkins, R. C. Guayasamin, D. E. Elmore, and J. A. Maurer, *Eur. Biophys. J.* **41**, 1003 (2012).
- ⁶⁸G. Maksaev and E. S. Haswell, *Proc. Natl. Acad. Sci. U.S.A.* **109**, 19015 (2012).
- ⁶⁹X. Zhang, J. Wang, Y. Feng, J. Ge, W. Li, W. Sun, I. Iscla, J. Yu, P. Blount, Y. Li, and M. Yang, *Proc. Natl. Acad. Sci. U.S.A.* **109**, 18180 (2012).
- ⁷⁰B. Martinac, M. Buechner, A. H. Delcour, J. Adler, and C. Kung, *Proc. Natl. Acad. Sci. U.S.A.* **84**, 2297 (1987).
- ⁷¹S. Sukharev, *Biophys. J.* **83**, 290 (2002).
- ⁷²B. Akitake, A. Anishkin, and S. Sukharev, *J. Gen. Physiol.* **125**, 143 (2005).
- ⁷³E. L. Cussler, *Diffusion: Mass Transfer in Fluid Systems* (Cambridge University Press, 1997).
- ⁷⁴L. G. Longworth, *J. Am. Chem. Soc.* **75**, 5705 (1953).
- ⁷⁵M. Sotomayor, T. A. van der Straaten, U. Ravaioli, and K. Schulten, *Biophys. J.* **90**, 3496 (2006).
- ⁷⁶Y.-H. Li and S. Gregory, *Geochim. Cosmochim. Acta* **38**, 703 (1974).
- ⁷⁷H. Tal-Ezer and R. Kosloff, *J. Chem. Phys.* **81**, 3967 (1984).
- ⁷⁸A. Szabo, K. Schulten, and Z. Schulten, *J. Chem. Phys.* **72**, 4350 (1980).
- ⁷⁹W. Humphrey, A. Dalke, and K. Schulten, *J. Mol. Graphics* **14**, 33 (1996).
- ⁸⁰S. G. Schultz, N. L. Wilson, and W. Epstein, *J. Gen. Physiol.* **46**, 159 (1962).
- ⁸¹T. Ogahara, M. Ohno, M. Takayama, K. Igarashi, and H. Kobayashi, *J. Bacteriol.* **177**, 5987 (1995).
- ⁸²J. C. Phillips, R. Braun, W. Wang, J. Gumbart, E. Tajkhorshid, E. Villa, C. Chipot, R. D. Skeel, L. Kale, and K. Schulten, *J. Comput. Chem.* **26**, 1781 (2005).
- ⁸³W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.* **79**, 926 (1983).
- ⁸⁴J. B. Klauda, R. M. Venable, J. A. Freites, J. W. O'Connor, D. J. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. D. MacKerell, Jr., and R. W. Pastor, *J. Phys. Chem. B* **114**, 7830 (2010).
- ⁸⁵A. D. MacKerell, Jr., D. Bashford, M. Bellott, J. R. L. Dunbrack, J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorcikiewicz-Kuczera, D. Yin, and M. Karplus, *FASEB J.* **6**, A143 (1992).
- ⁸⁶A. D. MacKerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, I. W. E. Reiher, B. Roux, M. Schlenkrich, J. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorcikiewicz-Kuczera, D. Yin, and M. Karplus, *J. Phys. Chem. B* **102**, 3586 (1998).