A Parallel Finite Element Simulator for Ion Transport ...
Tu, Chen, Xie, Zhang, Eisenberg, and Benzhuo Lu
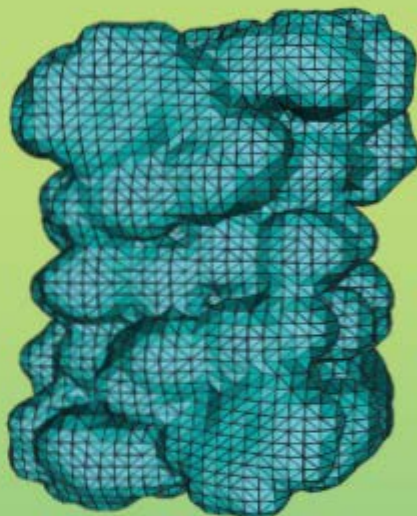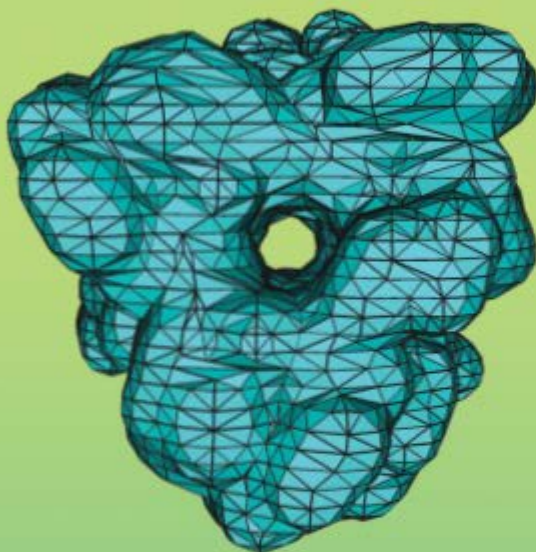
Journal of
COMPUTATIONAL
CHEMISTRY

Organic • Inorganic • Physical
Biological • Materials

Volume 34, p.2065-2078

WILEY

A Parallel Finite Element Simulator for Ion Transport through Three-Dimensional Ion Channel Systems

Bin Tu,[a] Minxin Chen,[b] Yan Xie,[a] Linbo Zhang,[a] Bob Eisenberg,[c] and Benzhuo Lu*[a]

A parallel finite element simulator, ichannel, is developed for ion transport through three-dimensional ion channel systems that consist of protein and membrane. The coordinates of heavy atoms of the protein are taken from the Protein Data Bank and the membrane is represented as a slab. The simulator contains two components: a parallel adaptive finite element solver for a set of Poisson–Nernst–Planck (PNP) equations that describe the electrodiffusion process of ion transport, and a mesh generation tool chain for ion channel systems, which is an essential component for the finite element computations. The finite element method has advantages in modeling irregular geometries and complex boundary conditions. We have built a tool chain to get the surface and volume mesh for ion channel systems, which consists of a set of mesh generation tools. The adaptive finite element solver in our simulator is implemented using the parallel adaptive finite element package Parallel Hierarchical Grid (PHG) developed by one of the authors, which provides the capability of doing large scale parallel computations with high parallel efficiency and the flexibility of choosing high order elements to achieve high order accuracy. The simulator is applied to a real transmembrane protein, the gramicidin A (gA) channel protein, to calculate the electrostatic potential, ion concentrations and $I-V$ curve, with which both primitive and transformed PNP equations are studied and their numerical performances are compared. To further validate the method, we also apply the simulator to two other ion channel systems, the voltage dependent anion channel (VDAC) and α-Hemolysin (α-HL). The simulation results agree well with Brownian dynamics (BD) simulation results and experimental results. Moreover, because ionic finite size effects can be included in PNP model now, we also perform simulations using a size-modified PNP (SMPNP) model on VDAC and α-HL. It is shown that the size effects in SMPNP can effectively lead to reduced current in the channel, and the results are closer to BD simulation results. © 2013 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23329

Introduction

Ion channels are pore-forming proteins that help establish and control the small voltage gradient across the plasma membrane of cells by allowing the flow of ions down their electrochemical gradient.[1] Ion channels regulate the flow of ions across the membrane in all cells. They are integral membrane proteins; or, more typically, an assembly of several proteins. They are present on all membranes of cell (plasma membrane) and intracellular organelles (nucleus, mitochondria, endoplasmic reticulum, and so on). Ion channels are essential to cell life and control a wide variety of important physiological processes, ranging from nerve and muscle excitation, muscle contraction, action potential generation and resting, sensory transduction, cell volume and blood pressure regulation, cell proliferation, hormone secretion, fertilization, maintenance of salt and water balance, learning and memory, to programmed cell death.[2] There are over 300 types of ion channels in a living cell.[3] They may be classified by gating or by selectivity.[4–7] Voltage-gated ion channels open or close depending on the voltage gradient across the plasma membrane, while ligand-gated ion channels open or close depending on binding of ligands to the channel. Different channels are selective for different ions and the flows and resulting concentration changes of different ions carry different biological signals.

[a] B. Tu, Y. Xie, L. Zhang, B. Lu
State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China
E-mail: bzlu@lsec.cc.ac.cn

[b] M. Chen
Center for System Biology, Department of Mathematics, Soochow University, Suzhou 215006, China

[c] B. Eisenberg
Department of Molecular Biophysics and Physiology, Rush University Medical Center, Chicago, Illinois 60612

Over the past decade, methodological developments in nuclear magnetic resonance (NMR), X-ray crystallography, and electron spectroscopy have led to significant progress in determining structures of integral membrane proteins that form ion channels.[8–10] This accumulation of high-resolution structural information has enabled better understanding of channel conductance, gating, and selectivity. Various theoretical and computational approaches, from fundamental to phenomenological, also help understand the biological mechanism of ion channels. The most commonly used theoretical techniques in the field are stochastic models, *ab initio* molecular dynamics (MD),[11] classical MD, and continuum descriptions. Among these techniques, *ab initio* MD has played a crucial role in revealing the complex mechanism of molecular systems, such as chemical reactions. Due to its detailed description of electronic structure and the extremely demanding nature of coupling electronic structure to molecular motion, *ab initio* MD is limited to small systems at present. Classical MD utilizes empirical interaction potentials or force fields calibrated by macroscopic data to describe molecular motions and is able to handle an entire ion channel, including ions, counterions, solvent, lipids, and proteins. Unfortunately, there are two issues for commonly used MD methods: one issue is to develop appropriate force fields for the ionic mixtures and concentrated solutions in and near channels; another issue is a scale problem. It is quite difficult to run up to the time scale of ion permeation across most real channel membranes and to determine ion conductance. As a result, approaches that reduce the dimensionality of the ion channel systems play important roles in ion channel dynamics and transport. One of these approaches is Monte Carlo (MC) methods[12] which rely on repeated random sampling to compute the probability of movement of a selected set of particles. MC approaches simulate the ion permeation across the membrane over long time-scales without having to treat all the solvent molecules explicitly. Another class of important reduced models is Brownian dynamics (BD),[13–15] which is based on the stochastic equation of motions of ion particles driven by some effective potential functions. Both MC and BD approaches provide an explicit representation of ions while treating solvent and lipids as featureless dielectrics. These reduced models are simpler and computationally less expensive than all-atom MD and have been some of the main workhorses in ion channel transport modeling and prediction for many years. There are many successful applications of BD in ion channel systems. Several biological pores such as OmpF,[16,17] potassium channels,[18] α-HL,[19] and the voltage dependent anion channel (VDAC)[20] have been investigated utilizing BD simulations.

A further simplification in the ion representation, that is, the so called mean-field approximation of ionic solution, leads to a fully continuous model, Poisson–Nernst–Planck (PNP) equations,[21,22] in which ions are not treated as microscopic discrete entities but as continuous charge densities. The PNP system is a combination of Nernst–Planck (NP) equations introduced by Nernst and Planck[23,24] and Poisson equation (PE). PNP can be derived by explicit averaging of correlated Brownian trajectories.[25] In the context of ion flow through a membrane channel, the flow of ions is driven by their concentration gradients and by the electric field modeled together by the NP equations, and the electric field is in turn determined by the concentrations through PE. PNP theory has previously been applied to the study of ion transport in electrochemical liquid junction systems[26] and electron transport in semiconductor devices,[27] as well as ion permeation through biological membrane channels.[28,29] It is well known that the PNP theory neglects discrete particle effect. The size-modified PNP (SMPNP) model was applied to simulate biomolecular diffusion-reaction processes.[30–32] In this article, we tested a SMPNP model to simulate ion transport through 3D ion channel system.

Mathematical analyses of the PNP equations have been developed in the last few decades, but most are limited to 1D cases.[33–36] The reduction of the dimensionality greatly simplifies the mathematical analysis of the electrodiffusion systems, and the results provide useful guidelines for the analysis of the corresponding fully 3D systems in some limiting cases. As a trade-off, they are generally unable to reproduce the diffusion and reaction processes that critically depend on the geometry of the system and complicated boundary conditions. In contrast to the limited amount of work on the mathematical analysis of the PNP equations for biophysical applications, a number of numerical algorithms, including finite difference,[37,38] finite element,[39–41], spectral element,[42] and finite volume methods,[43] have been utilized in the past two decades for solving the PNP equations. A lattice relaxation algorithm in conjugation with the finite difference method was developed by Kurnikova and coauthors to solve the PNP equations for ion transport with the three-dimensional (3D) realistic geometry of the gA dimer. The accuracy of their method was calibrated with simple parallel plate and cylindrical pore geometries but convergence was not easy or automatic. Hollerbach et al.[42] applied a highly accurate and quite convergent spectral element method for solving the 3D PNP equations with various sensitivity analysis to determine the impact of the PNP model parameters on *I–V* curves. Mathur and Murthy[43] developed a multigrid algorithm based on an unstructured cell-centered finite volume method for solving the PNP equations. However, it is a challenging numerical task to solve PNP and SMPNP equations for simulating ion transport through large ion channels.

The object of this article is to develop a parallel finite element simulator for ion transport through ion channel systems. To our best knowledge, no finite element algorithms for solving the 3D PNP equations for ion channel systems have been reported in the literature. The finite element method (FEM) has advantages in modeling irregular geometries with complex boundary conditions. In our finite element approach, the ion channel surface (boundary) is identified and discretized; this discretization is used as the boundary of the volume mesh generation. Such meshes are said to be boundary or surface "conforming" because they are aligned with the "real" ion channel surface, whereas in the finite difference method, the

mesh is nonconforming because it is allowed to "cut through" the ion channel surface. The advantage of adopting conforming mesh in our study is that it is convenient to use and/or accurately treat proper specification of boundary conditions on the surface for the modeling of reaction-diffusion processes. However, due to the complexity of ion channel structures, the surface meshes are often of poor quality, and even have defects. This makes it difficult to get high-quality tetrahedral meshes. In addition, the embedding of a membrane slab representation in a tetrahedral mesh is also a tricky task. We have built a tool chain to generate high-quality meshes for ion channel systems by combining a few mesh generation tools. The tool chain has essentially four components: surface meshing,[44,45] quality improving, volume mesh generation,[45] and membrane mesh construction. A parallel adaptive finite element method (AFEM) is implemented and high parallel efficiency is shown in our numerical studies.

This article is organized as follows. The method for ion transport simulations is introduced in the section entitled Methods. First, we briefly review the 3D ion channel model and the PNP equations system. Then, we present our finite element algorithms for solving a set of PNP equations in which an iterative scheme is used for solving the coupled nonlinear discrete equations. Finally, we introduce our tool chain for getting the surface and volume meshes for ion channel systems. In the section Numerical Test, first, we present some numerical results and assess the performance of our ion channel simulator in ion transport simulations. The electrostatic potential, ion concentrations, ion conduction profiles, and $I-V$ curves are computed with certain range of ion concentrations and applied voltages. Moreover, the simulator is applied to VDAC and α-HL ion channels, and the simulation results are compared with BD simulation results. Finally, we perform a simulation with our SMPNP model on the above two ion channels to show improved results by including the finite size effects in the model. The article ends with a summary section.

Methods

Mathematical models of ion channel system

The model system consists of a protein, a membrane surrounding it, and a simulation box. The coordinates of heavy atoms of the protein were taken from the Protein Data Bank. Partial charges for the protein atoms were taken from the AMBER force field. Here, the membrane is represented as a slab and no charge is assigned to the membrane in the present work.

The PNP model combines the Nernst–Planck theory describing electrodiffusion of ions in the transmembrane channel with the Poisson theory describing the electrostatic potential whose gradient serves as a driving force of the ion motion. Consider an open domain $\Omega \in \mathbb{R}^3$, $\bar{\Omega}=\bar{\Omega}_m \cup \bar{\Omega}_s$, $\Omega_m \cap \Omega_s = \varnothing$ where Ω_m represents the protein and membrane region and Ω_s represents the solvent reservoirs and the channel region. We use Γ to denote the interface between the two regions, such that $\bar{\Gamma} = \bar{\Omega}_m \cap \bar{\Omega}_s$, and Γ_m to denote the membrane

boundary on the simulation box. We obtain the PNP equations by coupling the Nernst–Planck equation

$$\frac{\partial c_i}{\partial t} = -\nabla \cdot J_i, \qquad x \in \Omega_s, \qquad 1 \leq i \leq N, \qquad (1)$$

$$J_i = -D_i(\nabla c_i + \beta q_i c_i \nabla \phi), \qquad (2)$$

and the electrostatic PE with internal interface Γ:

$$-\nabla \cdot (\in \nabla \phi) = \lambda \sum_i q_i c_i + \rho^f, \qquad x \in \Omega, \qquad (3)$$

$$\phi_m = \phi_s, \qquad x \in \Gamma,$$

$$\in_m \frac{\partial \phi_m}{\partial n} = \in_s \frac{\partial \phi_s}{\partial n}, \qquad x \in \Gamma,$$

where $c_i(x,t)$ is the concentration of the ith ion species carrying charge q_i. D_i is the spatial-dependent diffusion coefficient, and φ is the electrostatic potential. N is the number of diffusive ion species in solution that are considered in the system. The constant $\beta = 1/(k_B T)$ is the inverse Boltzmann energy where k_B is the Boltzmann constant and T is the absolute temperature. We assume that the dielectric permittivity is piecewise constant with $\epsilon = \epsilon_m \epsilon_0$ in Ω_m and $\epsilon = \epsilon_s \epsilon_0$ in Ω_s, where ε_0 is the dielectric constant of vacuum. Typical values of ε_m and ε_s are 2 and 80, respectively. The permanent (fixed) charge distribution

$$\rho^f(x) = \sum_j q_j \delta(x - x_j)$$

is an ensemble of singular atomic charges q_j located at x_j inside biomolecules. The characteristic function λ is equal to 1 in Ω_s and 0 in Ω_m, implying that mobile ions are present only in the solvent region.

Regularization and transformation of PNP system

In this article, we only consider the steady-state PNP equations. An effective strategy for solving eq. (3) is to decompose the solution of the PE into a singular component, a harmonic component and a regular component,[39,46] that is, $\phi = \phi^s + \phi^h + \phi^r$. The singular component φ_s is the restriction on Γ_m of the solution of

$$-\in_m \Delta \phi^s(x) = \rho^f(x), \qquad x \in \mathbb{R}^3, \qquad (4)$$

φ^h and the harmonic component ϕ^h is the solution of a Laplace equation:

$$-\Delta \phi^h(x) = 0, \qquad x \in \Omega_m, \qquad (5)$$

$$\phi^h(x) = -\phi^s(x), \qquad x \in \Gamma \cup \Gamma_m.$$

It is seen that $\phi^s(x)$ can be given analytically by the sum of Coulomb potentials. This $\phi^s(x)$ is then used to compute the boundary condition for $\phi^h(x)$, the latter is to be solved numerically from eq. (5), for which we use FEMs in this study.

Subtracting these two components from eq. (3), we get the governing equation for the regular component $\phi^r(x)$:

$$-\nabla\cdot(\in \nabla\phi^r(x,t))=\lambda\sum_i q_i c_i(x,t), \qquad x \in \Omega, \qquad (6)$$

and the interface conditions

$$\phi_s^r-\phi_m^r=0,$$

$$\in_s \frac{\partial\phi_s^r}{\partial n}-\in_m \frac{\partial\phi_m^r}{\partial n}=\in_m \frac{\partial(\phi^s+\phi^h)}{\partial n}, \qquad x \in \Gamma.$$

It is worth noting that there is no decomposition of the potential in the solvent region, thus $\phi(x)=\phi^r(x)$ in Ω_s. For the steady-state of the system, the final regularized PNP equations consist of the regularized Poisson eq. (6) and the steady-state Nernst–Planck equations

$$\nabla\cdot D_i(x)(\nabla c_i(x)+\beta q_i c_i(x)\nabla\phi^r(x))=0, \qquad x \in \Omega_s, \qquad (7)$$

Physically there is no ion penetration through the interface Γ, that is, a zero macroscopic normal flux

$$D_i(\nabla c_i+\beta q_i c_i\nabla\phi^r)\cdot n=0, \qquad \text{on} \qquad \Gamma. \qquad (8)$$

To get a symmetric weak form for Nernst–Planck equations, the Slotboom variables, which are widely used in the study of semiconductor devices, can be employed. It is seen that by introducing the Slotboom variables

$$\tilde{c}_i=c_i e^{q_i\beta\phi^r}, \qquad \hat{D}_i=D_i e^{-q_i\beta\phi^r}, \qquad (9)$$

the PNP equations can be transformed as

$$-\nabla\cdot(\in \nabla\phi^r)=\lambda\sum_i q_i\tilde{c}_i e^{-q_i\beta\phi^r}, \qquad (10)$$

$$\nabla\cdot(\hat{D}_i\nabla\tilde{c}_i)=0$$

We will refer to numerical algorithms based on the transformed equations as the transformation method, while those based on the original equations as the primitive method. In our numerical computations, both primitive and transformation methods were used for solving the PNP equations. Numerical results indicate that the transformation method converges at a much higher rate than the primitive method.

Finite element discretization

The PNP equations are solved using the FEM and the algorithms are implemented with the parallel adaptive finite element package PHG. We now describe the numerical algorithms employed for the static PNP equations. For the boundary condition, fixed electric potential and ion concentrations are set on the upper and lower faces of the computational box. The channel is normal to these two faces (along the z-axis). On the side faces, the potential is a linear function of the vertical coordinate. The concentrations of the positively and negatively charged ions are equal to each other on both top and bottom faces to ensure charge neutrality in the reservoirs. Moreover, there is a no-flux boundary along the interface Γ. Let $u=\phi^r$, then consider the transformed PE

$$-\nabla\cdot(\in \nabla u)=\lambda\sum_i q_i\tilde{c}_i e^{-q_i\beta u}, \qquad (11)$$

$$u=\varphi, \qquad \text{on} \qquad \partial\Omega, \qquad (12)$$

$$[u]=0, \quad \left[\in \frac{\partial u}{\partial n}\right]=\in_m \frac{\partial(\phi^s+\phi^h)}{\partial n}, \qquad \text{on} \qquad \Gamma. \qquad (13)$$

Let $H_b^1(\Omega)=\{u \in H^1(\Omega)|u=\varphi$ on $\partial\Omega\}$, here φ denotes the boundary function, and $H^1(\Omega)$ is a Sobolev space of weakly differentiable functions. First, multiply both sides of eq. (11) by v that is a vector in $H_0^1(\Omega)$ and integrate them on the global domain Ω:

$$\int_\Omega -\nabla\cdot(\in \nabla u)v d\Omega=\lambda\int_\Omega \sum_i q_i\tilde{c}_i e^{-q_i\beta u}v d\Omega, \quad \forall v \in H_0^1(\Omega), \quad (14)$$

where $H_0^1(\Omega)$ is a Sobolev space of weakly differentiable functions which vanish on the boundary of the domain Ω. Then, integrating by parts, the weak form of eq. (11) is obtained as follows:

Find $u \in H_b^1(\Omega)$ which satisfies:

$$\int_\Omega (\in \nabla u\nabla v)d\Omega=\int_{\Omega_s} \sum_i (q_i\tilde{c}_i e^{-q_i\beta u})v d\Omega_s$$

$$-\in_m\int_\Gamma \frac{\partial(\phi^s+\phi^h)}{\partial n}v dS, \qquad \forall v \in H_0^1(\Omega) \qquad (15)$$

Compared with the original PE, these transformations lead to a nonlinear part of the potential field and a Newton or inexact-Newton method is used here to solve eq. (15). Denote by $\{\Phi_1,\cdots,\Phi_M\}$ the finite element basis, where M denotes the number of bases (or degrees of freedom, DOF). Let u_n be the finite element approximation of u at the nth Newton iteration, which can be expressed by its expansion in the finite element space with respect to the given bases and regarded as a vector in $\mathbb{R}^M$, that is, $u_n=\sum_{k=1}^M u_k^n\Phi^j$, where u_k^n is the kth degree of freedom (DOF) in the nth Newton iteration. We define a nonlinear function $F(u_n)$ ($\mathbb{R}^M \to \mathbb{R}^M$) whose jth component is given by:

$$F_j(u_n)=\int_\Omega (\in \nabla u_n\nabla\Phi_j)d\Omega-\int_{\Omega_s} \sum_i (q_i\tilde{c}_i e^{-q_i\beta u_n})\Phi_j d\Omega_s$$

$$+\in_m\int_\Gamma \frac{\partial(\phi^s+\phi^h)}{\partial n}\Phi_j dS, \qquad j=1\cdots M. \qquad (16)$$

Then the Newton iteration of the PE reads

$$F'(u_n)(u_n-u_{n+1})=F(u_n),$$

where $F'(u_n)$ is the ($M \times M$) Jacobian matrix whose j,k-th element is given by:

$$F_{j,k'}(u_n) = \int_{\Omega} (\in \nabla \Phi_k \nabla \Phi_j) d\Omega + \int_{\Omega_s} \sum_i (q_i^2 \beta \tilde{c}_i e^{-q_i \beta u_n}) \Phi_k \Phi_j d\Omega_s. \tag{17}$$

In each Newton iteration, we need to solve a linear system of equations of the form

$$\mathbf{Au} = \mathbf{f}, \tag{18}$$

where the stiffness matrix $\mathbf{A} = [A_{j,k}]_{M \times M} = [F_{j,k'}(u_n)]_{M \times M}$, the vector $\mathbf{f} = [f_j]_M = [F_j(u_n)]_M$, and the solution vector $\mathbf{u} = [u_n^k - u_{n+1}^k]_M$. We start from an initial guess u_0, which should satisfy the boundary condition $u_0 = \varphi$ on Ω, and carry out a certain number of Newton iterations until a given criterion for convergence is met, to obtain the final solution u_n.

The NP equations are linear equations, whose weak form is obtained as follows:

For each i, $1 \leq i \leq N$, find $\tilde{c}_i \in H_a^1(\Omega_s)$ which satisfies

$$\int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \tilde{c}_i \nabla v) d\Omega_s = 0, \qquad \forall v \in H_c^1(\Omega_s), \tag{19}$$

where $H_a^1(\Omega) = \{\tilde{c}_i \in H^1(\Omega) | \tilde{c}_i = \eta_i$ on $\Gamma_s\}$, here η_i denotes the dirichlet boundary function, and $H_c^1(\Omega) = \{v \in H^1(\Omega) | v = 0$ on $\Gamma_s\}$.

For a discrete solution to eq. (19), we denote the discretized approximation of $\tilde{c}_i$ by $\tilde{c}_i^h$. We employ a finite element space $V^h = \text{span}\{\psi^1, \cdots, \psi^L\}$, with L denoting the number of DOF in the finite element space, and denoting a subspace of $H_a^1(\Omega_s)$ by $\tilde{V}^h = \text{span}\{\psi^1, \cdots, \psi^L, \psi^{L+1}, \cdots, \psi^{L+T}\}$, with $\psi^{L+1}, \cdots, \psi^{L+T}$ referring to the finite element bases on the vertex $A_{L+1}, \cdots, A_{L+T}$ of the dirichlet boundary.

Denoting the approximate solution $\tilde{c}_i^h$ by its expansion with respect to the finite element bases as follow:

$$\tilde{c}_i^h = \sum_{j=1}^{L} \tilde{c}_i^j \psi^j + \sum_{s=1}^{T} \eta_i(A_{L+s}) \psi^{L+s} \in \tilde{V}^h, \tag{20}$$

where $\tilde{c}_i^j$ is the jth DOF of the ion concentration, and the discrete weak form is given by

$$\int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \tilde{c}_i^h \nabla \psi^j) d\Omega_s = 0, \qquad \forall \psi^j \in \{\psi^1, \cdots, \psi^L\} \tag{21}$$

To formulate eq. (21) into a matrix equation, we write its left hand side as

$$\int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \tilde{c}_i^h \nabla \psi^j) d\Omega_s = \sum_{k}^{L} \left[\tilde{c}_i^k \int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^k) d\Omega_s \right] \tag{22}$$

$$+ \sum_{s}^{T} \left[\eta_i(A_{L+s}) \int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^{s+L}) d\Omega_s \right]. \tag{23}$$

Then we get a linear system of equations in the following form

$$\mathbf{Bx} = \mathbf{y}, \tag{24}$$

where the stiffness matrix $\mathbf{B} = [\mathbf{B}_{j,k}]_{\mathbf{L} \times \mathbf{L}} = \left[\iint_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^k) d\Omega_s \right]_{\mathbf{L} \times \mathbf{L}}$, the vector $\mathbf{y} = [\mathbf{y_j}]_{\mathbf{L}} = \left[-\sum_{s}^{T} \left[\eta_i(A_{L+s}) \int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^{s+L}) d\Omega_s \right] \right]_{\mathbf{L}}$ and the solution vectors $\mathbf{x} = [\tilde{c}_i^k]_L$.

Iteration procedure between the coupled NP equations and PE

For the steady-state case, in order to make the iterations between the diffusion and electrostatic equations to converge, it was found necessary to employ under-relaxation, especially when macromolecules exist. In other words, variables are updated with a linear combination of old values and calculated new values, rather than just using the new values. The under-relaxation scheme[41,47] is described by

$$u^{\text{new}} = \alpha u^{\text{old}} + (1-\alpha) u^{\text{new}}$$

$$\tilde{c}_i^{\text{new}} = \alpha \tilde{c}_i^{\text{old}} + (1-\alpha) \tilde{c}_i^{\text{new}}, \qquad i = 1, \ldots, N$$

where the relaxation parameter $0 < \alpha < 1$ is a predefined constant. We have noted that without under-relaxation, the iterations may not converge.

Mesh generation for ion channel system

Our finite element algorithms use tetrahedral meshes. Mesh generation is a prerequisite for FEMs. However, it has been a long-existing and challenging task for meshing biomolecular systems due to their highly irregular shapes, which, historically, was actually a great impediment to using the FEM in continuum molecular modeling. A reasonable strategy to generate biomolecular meshes follows two steps: first generate a molecular surface conforming mesh, then generate a volume mesh based on the surface mesh.[41] Among the two steps, surface meshing is the more difficult one. Recently, we have developed a tool called TMSmesh that is potentially capable of generating manifold surface meshes for arbitrarily large molecular systems,[44,45] which, we hope, will facilitate the finite element simulations of biomolecular systems.

In this work, we have built a tool chain for high-quality biomolecule volume mesh generation by using TMSmesh and a few other meshing tools. The tool chain has essentially four components: surface meshing, quality improving, volume mesh generation, and membrane mesh construction. First, a triangulation of the Gaussian surface is generated using our recently developed program TMSmesh,[44] which is a robust tool for meshing molecular Gaussian surfaces and has been shown to be capable of handling molecules consisting of more than one million atoms. It reads a PQR file as an input file and exports a molecular surface mesh in OFF file format. The surface meshes generated by the old version of TMSmesh for large molecules sometimes have geometric defects such as containing intersecting, overlapping, and other nonmanifold surface triangles. Recently, we have improved TMSmesh by

developing a method that avoids intersections, ensuring mesh manifoldness and preserving the topology of the molecular Gaussian surface.[45] The surface meshes produced by the new version of TMSmesh are manifold meshes without intersections, but their quality still needs to be improved. Here, a manifold mesh means that the surface formed by all the elements of the mesh is a manifold. Therefore, in the second step, we first used the program ISO2Mesh[48] to simplify the surface mesh by reducing the number of faces or adding some nodes while preserving its manifoldness, volume, and boundary shape. ISO2mesh is a free matlab/octave-based mesh generation and processing toolbox, which can read the OFF file format exported from TMSmesh and export the filtered molecular surface in OFF file format. Subsequently, if self-intersecting faces exist, then the program TransforMesh,[49] which can robustly handle topology changes and remove self-intersections, is used to find and remove self-intersecting faces. Finally, in the third step, a tetrahedral volume mesh is generated using the program TetGen,[50] which consists of four-node tetrahedral elements and is ready for 3D finite element simulations. TetGen provides a set of switches to control its behavior. We generally use the "-pq" switch to get a high-quality tetrahedral mesh, where the "-p" switch reads a piecewise linear complex (PLC) stored in a ".poly" file and generates a constrained Delaunay tetrahedralization (CDT) of the PLC, and the "-q" switch performs quality mesh generation by using the Shewchuk's Delaunay refinement algorithm.[51]

Additionally, the membrane meshing also contains tricky tasks, such as how to find the set of tetrahedra belonging to the membrane region and get the membrane boundary. The membrane mesh is obtained by three steps. In the first step, two planes $z = z_1$ and $z = z_2$ are used to mark the position of the membrane region, and tetrahedra with all their four vertices located between $z = z_1$ and $z = z_2$ are marked as belonging to the membrane region. In the second step, tetrahedra which intersect with the planes $z = z_1$ or $z = z_2$ are first marked as the "interface tetrahedra" between the membrane region and the bulk region, then the faces of these "interface tetrahedra" are picked up and connected together to form the membrane boundary. Finally in the third step, the membrane boundary is submitted to a careful topology check to ensure its continuity, closedness, etc.

In order to facilitate the simulation of ion transport through ion channel systems, in the generated tetrahedral mesh, tetrahedra belonging to different regions are properly marked with different numbers. The triangles on the faces of the simulation box and the membrane boundary are also marked with different numbers. The final mesh is exported to a file in the Medit[52] file format which can be read by PHG.

Adaptive finite element method

The AFEM was originally proposed by Ivo Babuska et al.[53] It provides an efficient and systematic way of drastically improving the accuracy of finite element simulations by repeatedly adjusting the finite element mesh using a mesh adaptation strategy and an *a posteriori* error estimate, which would eventually lead to a quasi-optimal mesh for the given problem. For steady-state problems, the AFEM consists of starting from an initial mesh and performing the following loop to get a final adaptive mesh and a solution on it:

- Step 1: compute an approximate solution on the current mesh.
- Step 2: compute an error indicator using the *a posteriori* error estimate on each element. If the error estimate meets the convergence criterion then stop.
- Step 3: mark (select) the elements to be refined (adjusted) using the error indicators with a prescribed marking strategy.
- Step 4: refine the marked elements, plus possibly a few more to maintain mesh conformity, and goto Step 1.

In AFEM literature the above loop is often called the *Solve–Estimate–Mark–Refine* loop, which represents the standard mesh adaptation procedure in AFEM. There are traditionally three ways to refine an element: (a) divide the element into smaller elements (*h*-refinement); (b) increase the polynomial order of finite element bases in the element (*p*-refinement); (c) a combination of (a) and/or (b), that is, perform one of or both *h*- and *p*-refinement on the element (*hp*-refinement). The *a posteriori* error estimate also plays an important role in AFEM. It should give an accurate estimation of the error of the approximate solution on each element and can be computed using known data such as the numerical solution and other given data.

For the PNP system, we use the following *a posteriori* error estimate: [54]

$$\eta_s = (h_s^2 \| \sum_i q_i c_i + \rho^f + \nabla \cdot (\epsilon \nabla \phi_h) \|_{L^2(s)}^2$$

$$+ \frac{1}{2} \sum_{f \in F(s)} h_f \| [(\epsilon \nabla \phi_h) \cdot n_f] \|_{L^2(f)}^2)^{1/2} \quad (25)$$

where s represents an arbitrary element in the mesh and η_s is the error indicator on s, $F(s)$ denotes the set of (nonboundary) faces of s, h_s denotes the diameter of s, and h_f and n_f denote the diameter and normal vector of the face f, respectively. This error estimate is similar to a well-known *a posteriori* error estimate for the Poisson–Boltzmann equation introduced by Holst et al.[55] if we consider steady-state diffusion process.

Our adaptive finite element solver for the PNP system is implemented using the toolbox PHG.[56] PHG is a parallel toolbox for writing adaptive finite element programs. It was developed at the State Key Laboratory of Scientific and Engineering Computing of Chinese Academy of Sciences and its key features include supporting bisection-based conforming parallel adaptive tetrahedral meshes and the ability to scale to thousands of Message Passing Interface (MPI) processes (or tens of thousands of CPU cores through MPI+OpenMP two level parallelism). For parallel processing, PHG partitions a mesh into submeshes, which are then distributed onto MPI processes. PHG supports fully parallel local mesh refinement and coarsening based on a tetrahedron bisection algorithm. It has an
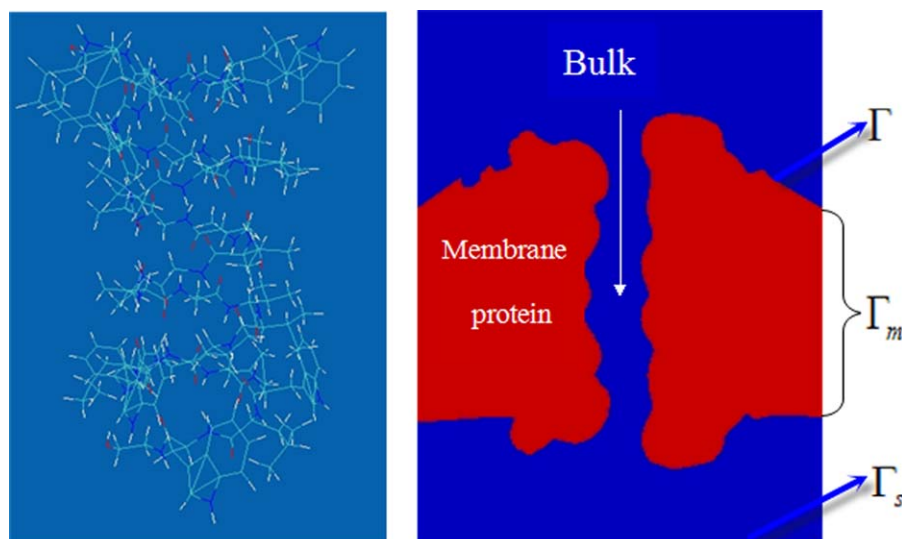
Figure 1. GA dimer (left column). A 2D cut through the center of the simulation box along the z axis illustrates the mesh representation of the protein and the membrane. The membrane and the protein region are shown in red, solvent reservoirs and the channel region are shown in blue (right column). [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

aqueous pores in lipid bilayers that selectively pass monovalent cations.[57,58] GA is a small 15-amino-acid β helical peptide with a narrow pore. As it is relatively small and well characterized experimentally, a wide variety of theoretical models have been applied to the gA channel. In the present work, we utilize the PNP equations to calculate the current as a function of voltage applied across the channel. The whole domain of the gA channel consists of the membrane protein region, bulk region, and the channel region. The layout of the gA channel on the grid is shown in Figure 1. The partial charges and atomic radii for each atom in the protein are obtained by using the PDB2PQR software.[59] The gA channel pore region is along the z direction. The box size is 30 Å ×30 Å ×45 Å.

object-oriented design which hides parallelization details, allowing the users to concentrate on the modeling and numerical algorithms. Although PHG provides full *hp*-refinement support, only *h*-refinement is used in this work.

Numerical Test

In this section, we apply our parallel finite element simulator, ichannel, to gA ion channel to compute the electrostatic potential, ion concentrations, and $I-V$ curve under various combinations of inputs. To further validate the effectiveness of our simulator, the simulator is also applied to other two ion channels, VDAC and α-HL, and the simulation results are compared with BD simulation results and experimental data. The convergence rate of the solver and its parallel efficiency are also investigated. In the computations, only MPI parallelism is used and the parallel efficiency for p processes is defined as

$$E_p = \frac{p_0 T_{p_0}}{p T_p},$$ (26)

where T_x denotes the execution time (wall-clock time) when using x processes in the computation, and p_0, $1 \le p_0 \le p$, denotes the smallest number of processes used in the computations (due to memory requirement p_0 may be greater than 1).

The computations were carried out on the cluster LSSC-III of the State Key Laboratory of Scientific and Engineering Computing of China, which consists of compute nodes with dual Intel Xeon X5550 quad-core CPUs, interconnected via DDR InfiniBand network.

Numerical test with gA ion channel

Gramicidin A system setup. One of the most widely studied ion channels is gA channel (PDB code: 1MAG) which forms
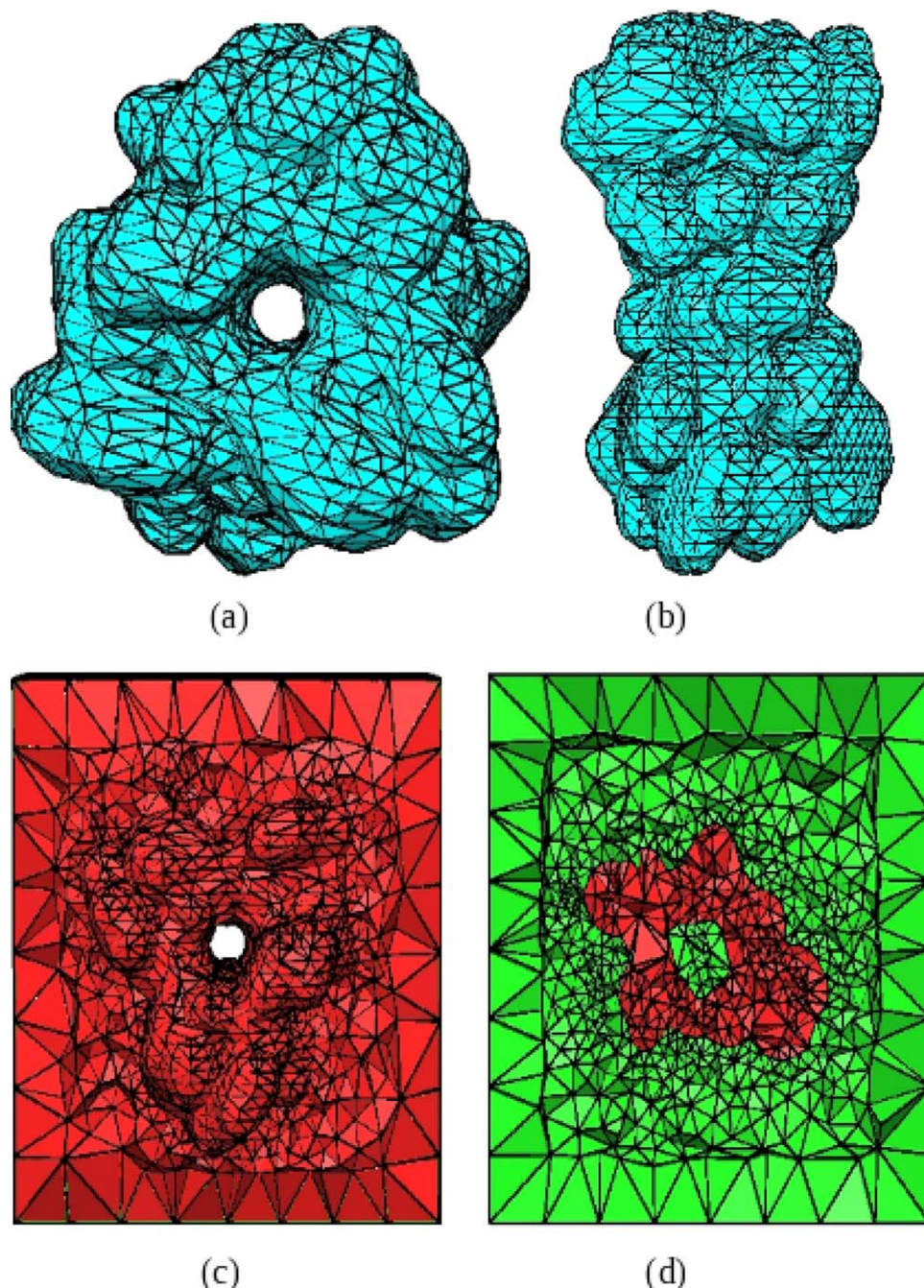
The membrane layer is represented as a slab.

The triangular surface mesh and tetrahedral volume mesh are generated using the methods mentioned above. The molecular surface mesh of the gA channel protein is generated by the TMSmesh program and the mesh quality is improved through topology check and smoothing. Then the volume mesh is generated using TetGen. Finally, the membrane region is extracted and the involved tetrahedra and boundary faces are properly marked, which end the mesh construction for the whole ion channel systems. Figure 2 shows an example of the unstructured tetrahedral volume mesh and triangular surface mesh of the gA ion channel. The mesh over the whole domain has a total of 22,753 vertices and 142,954 tetrahedra.

In the following computations, the membrane and protein regions (red area in Fig. 1) are described by low dielectric constant $\epsilon_m = 2$. The high dielectric constant $\epsilon_s = 80$ is assigned to the aqueous region, that is, the volume outside of the protein-membrane region (blue region in Fig. 1). The diffusion coefficients for cation and anion, for example, K^+ and Cl^-, in the bulk region are set to their experimental values: $D_{Cl} = 0.203$ Å^2/ps, $D_K = 0.196$ Å^2/ps. While there is no experimental measurement of appropriate values for the diffusion coefficients inside the channel, it is known that the diffusion coefficients in the bulk region and the channel pore region should be different, particularly for narrow pores.[60] GA is a narrow ion channel with a diameter of about 4 Å. Here, we present a case where the diffusion coefficients of ions continuously change inside the channel. The diffusion coefficient function is given as follows:[61]

$$D(r) = \begin{cases} D_{bulk}, & r \in \text{bulk region}, \\ D_{chan} + (D_{chan} - D_{bulk})f(r), & r \in \text{buffering region}, \\ D_{chan}, & r \in \text{channel region}, \end{cases}$$ (27)

Figure 2. Triangular boundary mesh conforming to the gA ion channel surface: a) top view. b) lateral view. c) Boundary surface mesh of ion channel with the membrane which is represented as a slab. d) A view of cross section of the whole tetrahedral volume mesh. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]

where the function $f(r)$ is given by

$$f(r)=f(z)=n\left(\frac{z-z_{chan}}{z_{bulk}-z_{chan}}\right)^{n+1}-(n+1)\left(\frac{z-z_{chan}}{z_{bulk}-z_{chan}}\right)^n, \quad (28)$$

where n is an integer and we set $n = 9$ in our computations. z_{chan} is the boundary value of channel region on z axis and z_{bulk} is the boundary value of bulk region on z axis. For the bottom boundary, $z_{chan} = 11$ and $z_{bulk} = 9$. For the top boundary, $z_{chan} = 32$ and $z_{bulk} = 34$. This profile for the

diffusion coefficients ensures that $D(r)$ is differentiable in the Nernst–Planck equation.

For the boundary condition, the voltage applied to the system, $V_{applied}$, is given by the potential difference along the z direction. On the box side boundary faces, the potential is set by interpolating linearly between top and bottom potential values. Ion concentrations on the top and bottom side boundaries are set to their bulk values. Additionally, there is a no-flux boundary surrounding the peptide and membrane that prevents ions from penetrating through the region occupied
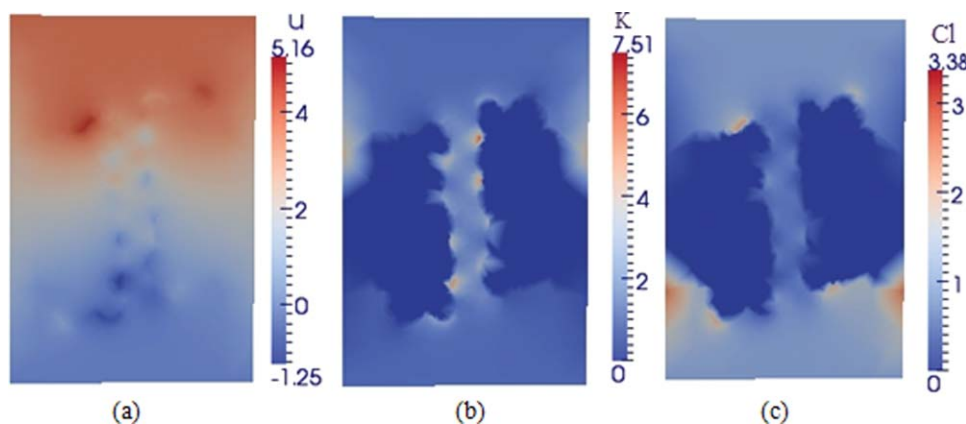
Figure 3. Electrostatic potential (k_BT/e_c) and ion concentration (M). (a) is a cross section view of the electrostatic potential of the whole domain. (b) is a cross section view of K^+ ion concentration of the whole domain. (c) is a cross section view of Cl^- ion concentration of the whole domain.
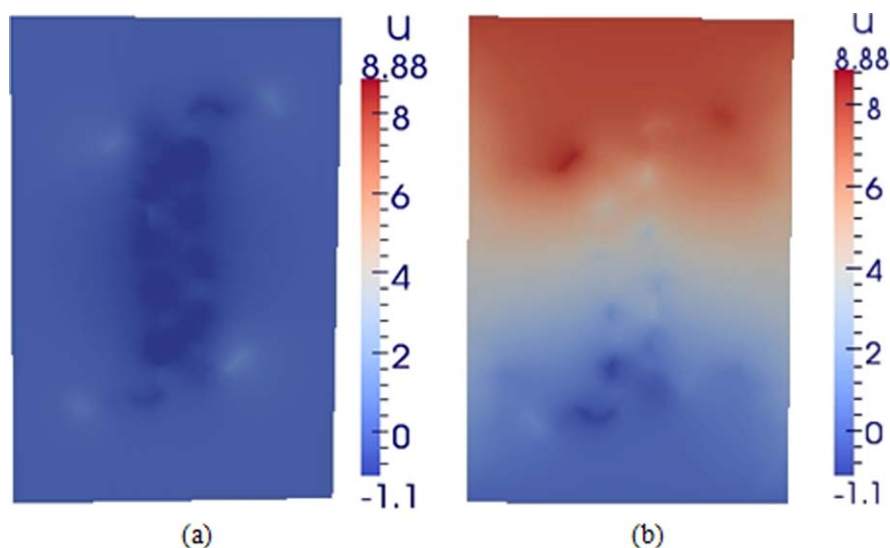
Figure 4. Comparison of electrostatic potential (k_BT/e_c) between PB and PNP calculations. (a) is a cross section view of the electrostatic potential obtained from PB calculation with the Debey–Hückel boundary condition and with the same bulk ion concentrations as in the compared PNP calculation. (b) is a cross section view of the electrostatic potential obtained from PNP calculation with $V_{applied} = 200$ mV and $c_{i,bulk} = 0.5$ M. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

$$c_i(x, y=0, z$$
$$\in \text{bulk region}) = c_i(x, y=L_y, z$$
$$\in \text{bulk region}) = c_{i,\text{bulk}},$$

$$(34)$$

$$J_i \cdot n = 0 \quad \text{on} \quad \Gamma. \quad (35)$$

Comparison with experimental data. We solve the coupled eqs. (9) and (10) to obtain the steady-state ion concentrations and electrostatic potential. For a given boundary condition ($V_{applied} = 100$ mV and $c_{i, bulk} = 1.0$ M), a cross section view of the potential and ion concentration of the whole domain region are shown in Figure 3. It can be seen that the concentration of K^+ is higher than that of Cl^- in the pore. To obtain a rough idea of the difference between the Poisson–Boltzmann (PB) electrical potential (equilibrium state) and the potential resulted from PNP calculation (nonequilibrium state) for the same channel system, Figure 4 shows a comparison between these two potential profiles.

The electrical current across the pore can be calculated as:

$$I_z = -\sum_i q_i \int_S D_i \left(\frac{\partial c_i}{\partial z} + \frac{q_i}{k_B T} c_i \frac{\partial \phi}{\partial z} \right) dx dy$$

$$(36)$$

where S is a cut plane at any cross section inside the pore. Equation (36) can be applied at any z-position along the pore axis, and shows only minor differences in the current value I_z due to numerical inaccuracies. In most cases presented here, these variations are on the order of ~2%.

To get $I{-}V$ curve and compare with the experimental data, the PNP equations are computed for a variety of voltages and concentrations. For example, $V_{applied} = 0$ mV, 50 mV, 100 mV, 150 mV, and 200 mV, and $c_{i, bulk} = 0.1$ M, 0.2 M, 0.5 M, 1.0 M, and 2.0 M. The potential profile for five different bulk concentrations with the same voltage ($V_{applied} = 100$ mV) is presented in Figure 5. It is shown that the potential with a higher concentration boundary condition is larger than that with a lower concentration boundary condition in the channel pore region due to ionic screening effect.

Figure 6 shows K^+ and Cl^- concentration profiles for five different applied voltage values, while the bulk concentration

by the peptides and lipids. Throughout the remainder of this manuscript, the z-direction will refer to the direction along the axis of the channel. Letting L_x, L_y, and L_z represent the length of the computational domain, we can summarize the above boundary conditions as:

$$\phi(x, y, z=L_z) = 0; \quad \phi(x, y, z=0) = V_{applied}, \quad (29)$$

$$\phi(x=0, y, z) = \phi(x=L_x, y, z) = (1 - z/L_z) \cdot V_{applied}, \quad (30)$$

$$\phi(x, y=0, z) = \phi(x, y=L_y, z) = (1 - z/L_z) \cdot V_{applied}, \quad (31)$$

$$c_i(x, y, z=0) = c_i(x, y, z=L_z) = c_{i,\text{bulk}}, \quad (32)$$

$$c_i(x=0, y, z \in \text{bulk region}) = c_i(x=L_x, y, z$$
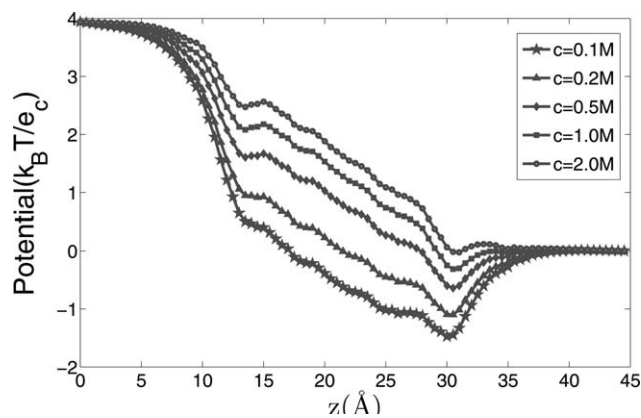$$\in \text{bulk region}) = c_{i,\text{bulk}}, \quad (33)$$

Figure 5. Electrostatic potential at the center of the gA channel plotted along z-axis obtained from the PNP calculation with $V_{applied} = 100$ mV.

Figure 7. Comparisons of the computed I–V curves with experimental data. I–V curves are plotted for five bulk ion concentrations and the dashed represent the experimental data.
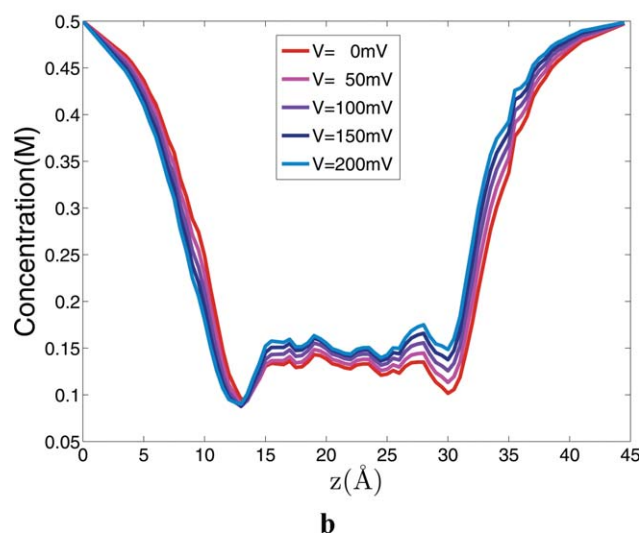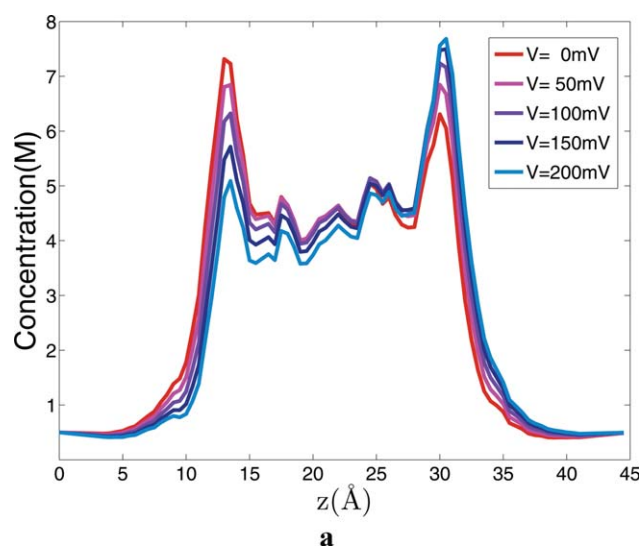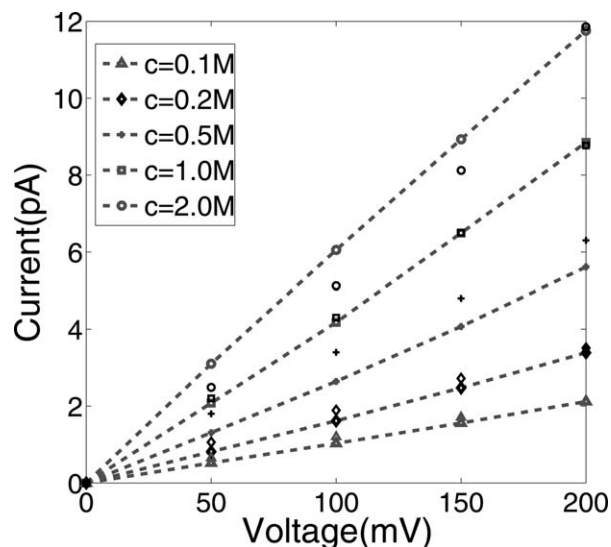
Figure 6. K^+ (top) and Cl^- (bottom) concentrations at the center of the gA channel plotted along z-axis obtained from the PNP calculation with $c_{i,bulk}$ = 0.5 M.

is the same ($c_{i,\,bulk}$ = 0.5 M). It is seen that although different voltage values as boundary condition are applied, the changes of concentrations have almost the same tendency. It is seen that the Cl^- concentration inside the gA is not exactly zero in our simulation, though it is thought to be zero experimentally. But we think this would be a common problem of traditional mean-field continuum model, such as PNP model.

The experimental I−V data are obtained from Cole et al.,[62] which are used as the reference data for comparison. The diffusion coefficient in the bulk region can be got from the experimental data. However, there are no experimental data available for the diffusion coefficient in the channel pore. Here, we obtain the diffusion coefficient in the channel pore by matching the experiment data, that is, the current value at V = 100 mV. We find that a reasonable match will be obtained with a diffusion coefficient which is 18 times smaller than the bulk coefficient, that is, $D_{Cl} = 1.13 \times 10^{-2}$ Å^2/ps, $D_K = 1.09 \times 10^{-2}$ Å^2/ps. We use the same data in all the simulations performed in this work. Comparisons between the simulation results and experimental data are shown in Figure 7. Table 1 shows the standard error of the current between the simulation results and experimental data. It is seen that although there are some deviations between them, overall the agreement is good between these two sets of data.

Convergence rate and parallel efficiency. With the transformation method, the PE is nonlinear and is solved by Newton iterations, thus it costs more internal iterations than with the primitive method. However, the external iterations between NP and the PE can converge much faster with the transformation method than with the primitive method. This is true when solving the PNP equations for gA channel, as demonstrated by the number of iterations in Table 2, in which a same relaxation parameter α = 0.8 is used in the external iterations of all the computations.

Table 1. The standard error between the simulation results and experimental data.

Ion concentration (M)	Voltage (mv)	Experimental data (pA)	The simulation results (pA)	Standard error
0.1	50	0.65	0.52	0.1169
	100	1.2	1.03	
	150	1.71	1.56	
	200	2.12	2.12	
0.2	50	1.06	0.81	0.2049
	100	1.89	1.62	
	150	2.72	2.48	
	200	3.51	3.39	
0.5	50	1.66	1.31	0.6980
	100	3.46	2.64	
	150	4.94	4.07	
	200	6.55	5.61	
1.0	50	2.08	2.21	0.0853
	100	4.18	4.29	
	150	6.49	6.51	
	200	8.86	8.77	
2.0	50	2.49	3.1	0.6169
	100	5.12	6.05	
	150	8.12	8.93	
	200	11.86	11.76	

Table 2. Number of external iterations between the NP and the PE under different combinations of ion concentration (M) and voltage (mV).

Ion concentration and voltage (M, mV)	Primitive	Transformed
0.1, 50	156	16
0.1, 100	156	16
0.1, 200	159	16
0.5, 50	154	21
0.5, 100	154	21
0.5, 200	156	21
2.0, 50	158	23
2.0, 100	154	23
2.0, 200	155	23

Table 3. Parallel efficiency in solving the PNP equations using the transformation method.

Number of processes	Number of iterations	Time (s)	Efficiency (%)
8	11	3755.6	100
16	11	1840.7	102.0
32	11	836.5	112.2
64	11	428.3	109.6
128	11	280.1	83.8
256	11	160.3	73.2
512	11	94.3	62.2
1024	11	76.4	38.4

Further validation of our method

In this subsection, to further validate the effectiveness of our simulator, we apply our method to two proteins as described in the reference,[63] VDAC and α-HL, and compare our PNP results with the BD simulation and experimental results. In addition, we have also performed a simulation with our SMPNP model on above two ion channels to show improved results by including the finite size effects in the model. It is known that it is still a big challenge to solve the PNP/SMPNP equations for large ion channel system due to the algorithm stability and mesh quality. We found there is still some difficulties for us to simulate the biggest ion channel listed in the article.[63] Exploration of more powerful numerical methods and extensive studies on finite size effects are our future tasks and are underway.

Comparison with BD simulation and experimental results. VDAC are a class of porin ion channel located on the outer mitochondrial membrane. VDAC provides a permeation pathway for metabolites and electrolytes between cell cytosol and the mitochondria. Among three isoforms found in many eukaryotic cells, the prototype isoform, VDAC1, shares a sequence identity up to 75% and the characteristic electrophysiological features. Initial coordinates for VDAC1 were taken from the protein data bank (code 2JK4).[64] The partial charges and atomic radii for each atom in the protein are obtained by using the PDB2PQR software. The PQR file of VDAC1 contains 4393 atoms. The diffusion coefficients for K^+ and Cl^- in the bulk region are set to their experimental values: $D_K = 0.196$ Å /ps, $D_{Cl} = 0.203$ Å /ps. In the channel pore region, we set $0.4\ D_K$ and 0.4 as the diffusion coefficients which are used for the simulation. The protein pore is aligned with the z axis. The layout of the VDAC1 channel on the grid is shown in Figure 8. Table 4 summarizes the conduction properties of VDAC1 obtained from BD simulation[63] and PNP simulation. Figure 9 shows $I-V$ curves obtained from BD and PNP simulations of VDAC1 for the asymmetric solution cases. It is seen that the simulation results agree well with BD simulation results and experimental results.

α-HL is a bacterial exotoxin protein involved in many diseases including urinary infection in human body. This toxin causes cell death by binding with the outer membrane, with subsequent oligomerization of the toxin monomer and water-filled channels. It forms a heptametric transmembrane channel with a relatively wide pore and a +7e net charge in the host

In our numerical study of gA ion channel, with the transformation method, the number of internal Newton iterations for solving the PE ranges from 2 to 4, and the total number of iterations is about 50 for each PNP solution, which is fewer than with the primitive method.

To assess the parallel efficiency of our parallel code, we introduce a much larger system with a mesh containing a total of 1,523,013 vertices and 9,149,056 tetrahedra, on which we solve the PNP equations using the transformation method. Table 3 gives the wall-clock time and parallel efficiency for different number of MPI processes. The smallest number of processes used is $p_0 = 8$, whose parallel efficiency is regarded as 100%. The parallel efficiencies obtained are satisfactory. A rapid drop in the parallel efficiency can be noted when going from 512 processes to 1024 processes, which we believe is related to the interconnection topology of the underlying InfiniBand network. The code is expected to be able to solve larger systems with more CPUs by using computers with faster interconnection network and/or exploiting the MPI+OpenMP two level parallelism provided by PHG.

cell membrane. Initial coordinates for α-HL were also obtained from the protein data bank (code 7AHL).[9] The partial charges and atomic radii for each atom in the protein are obtained by using the PDB2PQR software. The PQR file of α-HL contains 32,305 atoms. The protein pore is also aligned with the z axis. The layout of the α-HL channel on the grid is shown in Figure 8. BD and PNP simulation results of α-HL are summarized in Table 5. Figure 10 shows $I-V$ curves obtained from BD and PNP simulations of α-HL for the symmetric solution cases. It is seen that while there are some deviations between the PNP and BD simulation results, overall, the PNP simulation results is close to BD simulation results.

A test of a SMPNP model for finite size effects. The standard PNP neglects ionic finite size effects and resulting correlations that are expected to be important factors to determine the selectivities of some ion channels. Such effects, to some extent, now can be included in PNP model.[30] In our previous work, a SMPNP model was developed and applied to simulate biomolecular diffusion-reaction processes.[30] The SMPNP can also be adopted here to study the ionic size effects to ion transport in ion channel. Because the finite size effects are not the focus of this article, and they need extensive and

Table 4. Ion conduction properties of VDAC1: Conductance (G), ion selectivity from current ratio (G_{Cl} / G_K).

Method	Voltage (mV)	G	G_{Cl}/G_K
PNP	− 100	3.68	1.79
	+ 100	3.21	1.41
BD	− 100	3.77 ± 0.11	1.92
	+ 100	3.15±0.09	1.94
Exp.		3.9–4.5	

systematic studies in the future, here, as a demonstration, we only perform a few SMPNP simulations on VDAC1 and α-HL to show the performance of a size-included version of PNP. The SMPNP equations add a nonlinear term to each of the Nernst–Planck equations to model the steric repulsion: [30]

$$\frac{\partial c_i}{\partial t} = \nabla \cdot D_i \left(\nabla c_i + \beta q_i c_i \nabla \phi + \frac{k_i c_i}{1 - \sum_l a_l^3 c_l} \sum_l a_l^3 \nabla c_l \right),$$ (37)

$$x \in \Omega_s, \qquad 1 \leq i \leq N,$$

$$-\nabla \cdot (\in \nabla \phi) = \lambda \sum_i q_i c_i + \rho^f, \qquad x \in \Omega,$$ (38)

where $k_i = \frac{a_i^3}{a_0^3}$ and a_i denote the size of ith ion species and a_0 of the water molecule. In SMPNP, all the size values should be obtained through a parameterization procedure and fitted with experimental data or other theoretical results. Here, we simply choose one dataset $a_0 = 6.5$ and $a_1 = a_2 = 5.0$ as an example to test the size effects to ion transport. More details and discussion can be found in Refs. [30,65].

Figure 9 compares $I-V$ curves obtained from BD, PNP, and SMPNP simulations of VDAC1 in asymmetric 0.1:1.0 M and 1.0:0.1 M KCl solutions. Figure 10 compares $I-V$ curves for α-HL in symmetric 1.0:1.0 M KCl solutions. It is seen that, compared with PNP model, SMPNP model can effectively reduce the current due to steric exclusion effects, and the results are closer to BD simulation results.

Summary

The PNP theory is a well-established electrodiffusion model for a wide variety of chemical, physical, and biological applications. It has been extensively used in the ion channel analysis to compute the electrostatic and concentration profiles, as well as $I-V$ curves.
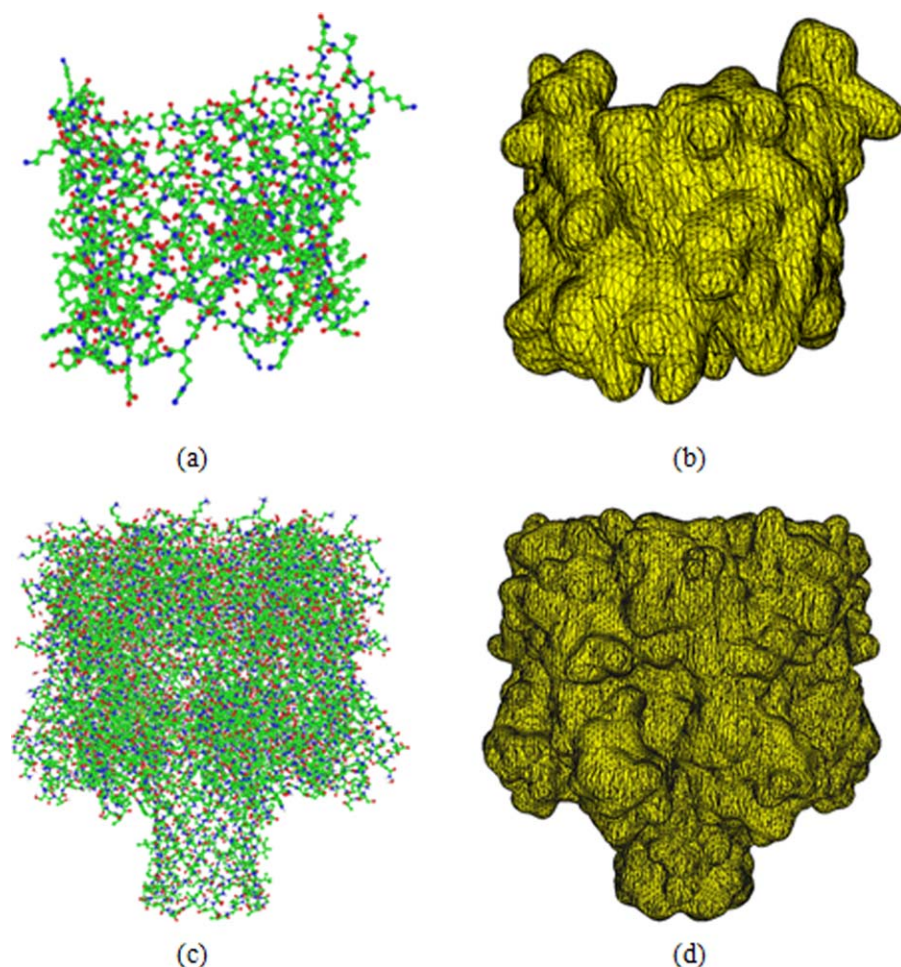
Figure 8. a) VDAC1 dimer. b) Triangular boundary mesh conforming to the VDAC1 surface. c) α-HL dimer. d) Triangular boundary mesh conforming to the α-HL surface. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com]
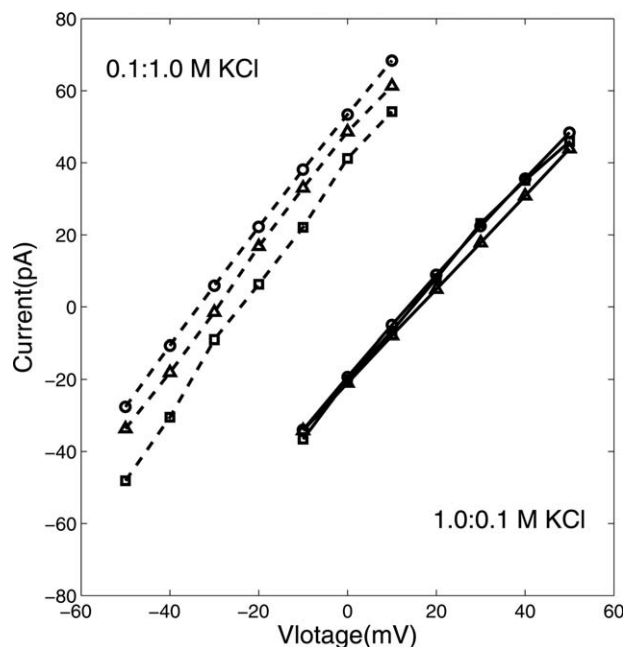
Figure 9. $I-V$ curves for VDAC1 in asymmetric 0.1:1.0 M (dashed) and 1.0:0.1 M (solid) KCl solutions obtained with BD model (square), PNP model (circle) and SMPNP model (triangle).

In this article, we present a parallel adaptive finite element simulator, ichannel, for ion transport through ion channel systems. Due to the complexity of molecular structure, there is no software so far that can solely finish the task of generating high-quality molecular surface mesh and tetrahedral volume mesh for the whole channel system for FE simulation. We have
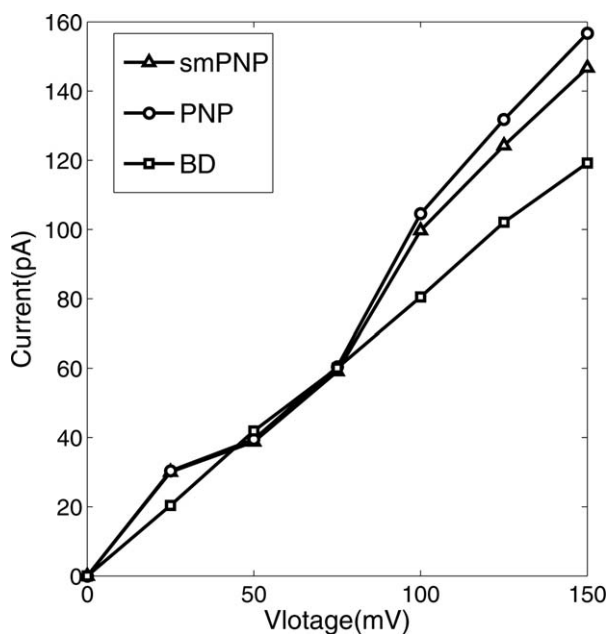

Figure 10. $I-V$ curves for α-HL in symmetric 1.0:1.0 M KCl solutions obtained with BD model (square), PNP model (circle) and SMPNP model (triangle).

Table 5. Ion conduction properties of α-HL: Conductance (G), ion selectivity from current ratio (G_{Cl} / G_K).

Method	Voltage (mV)	G	G_{Cl}/G_K
PNP	<0	0.87 ± 0.11	1.79 ± 0.19
	>0	0.99 ± 0.21	1.65 ± 0.25
BD	<0	0.76 ± 0.01	2.13 ± 0.32
	>0	0.81 ± 0.01	1.96 ± 0.23
Exp.	<0	0.80 ± 0.04	
	>0	1.00 ± 0.05	

built a tool chain for high-quality biomolecule mesh generation by using a few of mesh generation tools including the surface meshing tool developed by us recently. Numerical test is carried out to the gA channel protein. The electrostatic and concentration profiles, as well as $I-V$ curves are obtained under certain range of ion concentrations and applied voltages. A good agreement is achieved between the computed $I-V$ curves and the available experimental data. To further validate the effectiveness of our method, numerical test is also carried out to other two ion channels, VDAC1 and α-HL. The PNP simulation results agree well with BD simulation and experimental results. Moreover, because ionic finite size effects can be included in PNP model now, we also perform simulations using a SMPNP model on VDAC and α-HL. It is shown that the size effects in SMPNP can effectively lead to reduced current in the channel, and the results are closer to BD simulation results.

By comparing the primitive and the transformed formulations of the PNP equations applied to gA system, it is found that the number of iterations between the PE and the NP equations is significantly reduced using the transformed formulation. Our code is based on the parallel adaptive finite element package PHG, which provides the simulator with the ability of using large scale parallel processing, parallel mesh adaptation, and high order elements. High parallel efficiency of the code is confirmed by the numerical results.

To solve PNP and SMPNP equations for large proteins and/or arbitrary size, it is so far still a numerical challenge. Exploration of more powerful numerical methods and extensive studies on finite size effects are our future tasks and are underway. We aim to develop an user-friendly software platform for studies of ion transport through 3D ion channel systems. Including the correlations produced by the finite size of ions is likely to allow PNP to deal with the selectivity phenomena of calcium, sodium, and Ryr channels.[66]

Acknowledgment

Journal of COMPUTATIONAL CHEMISTRY

[1] B. Hille, Ion Channels of Excitable Membranes, 3rd ed.; Sinauer Associates: Sunderland, **2001**.

[2] M. Hacker, W. Messer, K. Bachmann, Pharmacology: Principles and Practice; Academic Press: New York, **2009**.

[3] I. Gabashvili, B. Sokolowski, C. Morton, A. Giersch, *J. Assoc. Res. Otolaryngol.* **2007**, *8*, 305.

[4] E. C. Conley, The Ion Channel Facts Book. I. Extracellular Ligand-Gated Channels; Academic Press: New York, **1996**.

[5] E. C. Conley, The Ion Channel Facts Book. II. Intracellular Ligand-Gated Channels; Academic Press: New York, **1996**.

[6] E. C. Conley, W. J. Brammar, The Ion Channel Facts Book IV: Voltage Gated Channels; Academic Press: New York, **1996**.

[7] E. C. Conley, W. J. Brammar, The Ion Channel Facts Book III: Inward Rectifier and Intercellular Channels; Academic Press: New York, **1996**.

[8] D. A. Doyle, J. M. Cabral, R. A. Pfuetzner, A. Kuo, J. M. Gulbis, S. L. Cohen, B. T. Chait, R. MacKinnon, *Science*. **1998**, *280*, 69.

[9] L. Song, M. R. Hobauch, C. Shustak, S. Cheley, H. Bayley, J. E. Gouaux, *Science*. **1996**, *274*, 1859.

[10] R. R. Ketchem, B. Roux, T. A. Cross, *Structure*. **1997**, *5*, 1655.

[11] D. Marx, J. Hutter, In Modern Methods and Algorithms of Quantum Chemistry.; J. Grotendorst, Ed.; John von Neumann Institute for Computing.: Jülich, **2000**; pp. 329–477.

[12] H. Hwang, G. C. Schatz, M. A. Ratner, *J. Phys. Chem. A* **2007**, *111*, 12506.

[13] S. Li, M. Hoyles, S. Kuyucak, S. Chung, *Biophys. J.* **1998**, *74*, 37.

[14] B. Corry, S. Kuyucak, S. H. Chung, *Biophys. J.* **2000**, *78*, 2364.

[15] S. Kuyucak, O. S. Andersen, S. H. Chung, *Rep. Prog. Phys.* **2001**, *64*, 1427.

[16] W. Im, B. Roux, *J. Mol. Biol.* **2002**, *322*, 851.

[17] T. Schirmer, P. S. Phale, *J. Mol. Biol.* **1999**, *294*, 1159.

[18] T. Allen, M. Hoyles, S. Kuyucak, S. Chung, *Chem. Phys. Lett.* **1999**, *313*, 358.

[19] S. Noskov, W. Im, B. Roux, *Biophys. J.* **2004**, *87*, 2299.

[20] K. Lee, R. W. Pastor, H. Rui, W. Im, *Biophys. J.* **2011**, *100*, 611.

[21] D. Chen, J. Lear, R. S. Eisenberg, *Biophys. J.* **1997**, *72*, 97.

[22] B. Z. Lu, In Encyclopedia of Applied and Computational Mathematics; B. Engquist, Ed.; Springer-Verlag: Berlin Heidelberg, 2013.

[23] W. Nernst, *Z. Physik. Chem.* **1889**, *4*, 129.

[24] M. Planck, *Ann. Phys. Chem.* **1890**, *39*, 161.

[25] Z. Schuss, B. Nadler, R. S. Eisenberg, *Phys. Rev. E*, **2001**, *64*, 036116.

[26] O. J. Riveros, T. L. Croxton, W. M. Armstrong, *J. Theor. Biol.* **1989**, *140*, 221.

[27] P. A. Markowich, The Stationary Semiconductor Device Equations; Springer-Verlag: Vienna, **1986**; pp. 1–29.

[28] R. S. Eisenberg, *Acc. Chem. Res.* **1998**, *31*, 117.

[29] B. Roux, T. Allen, S. Berneche, W. Im, *Q. Rev. Biophys.* **2004**, *7*, 1.

[30] B. Z. Lu, Y. C. Zhou, *Biophys. J.* **2011**, *100*, 2475.

[31] T. L. Horng, T. C. Lin, C. Liu, B. Eisenberg, *J. Phys. Chem. B* **2012**, *116*, 11422.

[32] S. G. Zhou, Z. M. Wang, B. Li, *Phys. Rev. E.* **2011**, *84*, 021901.

[33] W. S. Liu, *Siam J. Appl. Math.* **2005**, *65*, 754.

[34] V. Barcilon, D. Chen, R. Eisenberg, J. Jerome, *SIAM J. Appl. Math.* **1997**, *57*, 63.

[35] N. Abaid, R. S. Eisenberg, W. S. Liu, *Siam J. Appl. Dyn. Syst.* **2008**, *7*, 1507.

[36] R. S. Eisenberg, W. S. Liu, *Siam J. Math. Anal.* **2008**, *38*, 1932.

[37] M. G. Kurnikova, R. D. Coalson, P. Graf, A. Nitzan, *Biophys. J.* **1999**, *76*, 642.

[38] Q. Zheng, D. Chen, G. W. Wei, *J. Chem. Phys.* **2011**, *230*, 5239.

[39] B. Z. Lu, M. J. Holst, J. A. McCammon, Y. C. Zhou, *J. Comput. Phys.* **2010**, *229*, 6979.

[40] B. Z. Lu, J. A. McCammon, *Chem. Phys. Lett.* **2008**, *451*, 282.

[41] B. Z. Lu, Y. C. Zhou, G. A. Huber, S. D. Bond, M. J. Holst, J. A. McCammon, *J. Chem. Phys.* **2007**, *127*, 135102.

[42] U. Hollerbach, D. P. Chen, R. S. Eisenberg, *J. Sci. Comput.* **2002**, *16*, 373.

[43] S. R. Mathur, J. Y. Murthy, *Siam J. Appl. Math.* **2009**, *52*, 4031.

[44] M. Chen, B. Lu, *J. Chem. Theory Comput.* **2011**, *7*, 203.

[45] M. X. Chen, B. Tu, B. Z. Lu, *J. Mol. Graph. Model.* **2012**, *38*, 411.

[46] U. Hollerbach, R. S. Eisenberg, *Langmuir*. **2002**, *18*, 3262.

[47] W. Nonner, D. Gillespie, D. Henderson, B. Eisenberg, *J. Phys. Chem. B* **2001**, *105*, 6427.

[48] Q. Q. Fang, D. Boas, In Proceedings of IEEE International Symposium on Biomedical Imaging, Boston, June 28–July 1, 2009; Institute of Electrical and Electronics Engineers: New York, pp. 1142–1145.

[49] A. Zaharescu, E. Boyer, R. P. Horaud, In Proceedings of the Eighth Asian Conference on Computer Vision, **2007**, II, pp. 166–175

[50] H. Si, TetGen: A quality tetrahedral mesh generator and a 3D Delaunay triangulator. Available at: http://tetgen.berlios.de/, accessed on May 10, **2010**.

[51] J. R. Shewchuk, In Proceedings of the Fourteenth Annual Symposium on Computational Geometry, June 7-10, 1998, Minneapolis, Minnesota; Assn for Computing Machinery: New York, pp. 86–95.

[52] P. J. Frey, Medit: An interactive mesh visualization software. Available at: http://www.ann.jussieu.fr/frey/logiciels/medit.html, accessed on December 20, **2010**.

[53] B. Guo, I. Babuska, *Adv. Eng. Softw.* **1992**, *15*, 159.

[54] M. J. Berger, P. Colella, *J. Comput. Phys.* **1989**, *81*, 64.

[55] M. Holst, N. Baker, F. Wang, *J. Comput. Phys.* **2000**, *21*, 1319.

[56] L. B. Zhang, *Numer. Math. Theor. Methods. Appl.* **2009**, *2*, 65.

[57] O. S. Andersen, *Annu. Rev. Physiol.* **1984**, *46*, 531.

[58] R. E. Koeppe, O. S. Andersen, *Annu. Rev. Biophys. Biomol. Struct.* **1996**, *25*, 231.

[59] T. J. Dolinsky, J. E. Nielsen, J. A. McCammon, N. A. Baker, *Nucleic Acids Res.* **2004**, *32*, W665.

[60] D. Gillespie, *Biophys. J.* **2008**, *94*, 1169.

[61] H. Hwang, G. C. Schatz, M. A. Ratner, *J. Phys. Chem. A.* **2007**, *111*, 12506.

[62] C. D. Cole, A. S. Frost, M. C. Thompson, A. Cross, D. D. Busath, *Biophys. J.* **2002**, *83*, 1974.

[63] K. I. Lee, S. Jo, H. Rui, B. Egwolf, B. Roux, R. W. Pastor, W. Im, *J. Comput. Chem.* **2012**, *33*, 331.

[64] M. Bayrhuber, T. Meins, M. Habeck, S. Becker, K. Giller, S. Villinger, C. Vonrhein, C. Griesinger, *P. Natl. Acad. Sci. USA* **2008**, *105*, 15370.

[65] V. B. Chu, Y. Bai, J. Lipfert, D. Herschlag, S. Doniach, *Biophys. J.* **2007**, *93*, 3202.

[66] B. Eisenberg, *Adv. Chem. Phys.* **2011**, *148*, 77.