Journal of Computational Chemistry
(in the press, 2013)

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

Bin Tu,[†] Minxin Chen,[‡] Yan Xie,[†] Linbo Zhang,[†] Bob Eisenberg,[¶] and Benzhuo 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 , Center for System Biology, Department of Mathematics, Soochow University, Suzhou 215006, China , and Department of Molecular Biophysics and Physiology, Rush University Medical Center, 1653 West Congress Parkway, Chicago IL 60612, USA

E-mail: bzlu@lsec.cc.ac.cn

Abstract

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 tools 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

[*] To whom correspondence should be addressed
[†] Chinese Academy of Sciences
[‡] Soochow University
[¶] Rush University Medical Center

tools 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 PHG developed by one of the author,[1] 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 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.

1 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.[2] Ion channels regulate the flow of ions across the membrane in all cells. Ion channels 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 sustaining 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.[3] There are over 300 types of ion channels in a living cell.[4] Ion channels may be classified by gating or by selectivity.[5–8] 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.

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.[9–11] 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),[12] classical molecular dynamics (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 to describe molecular motions and is able to handle an entire ion channel, including ions, counterions, solvent, lipids and proteins. Unfortunately, it is still quite difficult for commonly used MD methods to run up to the time scale of ion permeation across most real channel membranes and to determine ion conductance.[13] As a result, approaches that reduce the dimensionality of the ion channel systems are playing important roles in ion channel dynamics and transport. One of these approaches is Monte Carlo (MC) methods[14] which rely on repeated random sampling to compute the probability of movement of a selected set of particles. Monte Carlo 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),[15–17] 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 treat 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,[18,19] potassium channels,[20] α-hemolysin[21] and the VDAC[22] have been investigated utilizing BD simulations.

A further simplification in the ion representation, i.e., the so called mean-field approximation of ionic solution, leads to a fully continuous model, Poisson-Nernst-Planck (PNP) equations,[23,24] 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[25,26] and Poisson equation (PE). PNP can be derived by explicit averaging of correlated Brownian trajectories.[27] 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 the Poisson equation. PNP theory has previously been applied to the study of ion transport in electrochemical liquid junction systems[28] and electron transport in semiconductor devices,[29] as well as ion permeation through biological membrane channels.[30,31]

Mathematical analyses of the Poisson-Nernst-Planck equations have been developed in the last few decades, but most are limited to 1-D case.[32–35] The reduction of the dimensionality greatly simplifies the mathematical analysis of the electrodiffusion systems, and the results provide useful guide lines for the analysis of the corresponding fully 3-D systems at some limit 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,[36,37] finite element,[38–40] spectral element[41] and finite volume methods,[42] 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 gramicidin A 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.[41] 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 to $I-V$ curves. Mathur and Murthy[42] developed a multigrid algorithm based on an unstructured cell-centered finite volume method for solving the

PNP equations.

The object of this paper is to develop a parallel finite element simulator for ion transport through ion channel systems. To our best knowledge, neither finite element algorithms, nor parallel numerical simulations for solving the 3D PNP equations for ion channels 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 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 tools chain to generate high-quality biomolecule meshes by combining a few mesh generation tools. A parallel adaptive finite element method is implemented and high parallel efficiency is shown in our numerical studies.

This paper is organized as follows. The method for ion transport simulations is introduced in Section 2. First, we briefly review the 3D ion channel model and the PNP equations system. Then, we present our finite element algorithms for solving the PNP equations in which an iterative scheme is used for solving the coupled nonlinear discrete equations. Finally, we introduce our tools chain for getting the surface and volume meshes for ion channel systems. In Section 3, we present some mumerical results and assess the performance of our ion channel simulator in ion transport simulations. The electrostatic potential, ion concentrations profiles and $I-V$ curves are computed with certain range of ion concentrations and applied voltages. The paper ends with a summary in Section 4. It is important to note that finite size effects and resulting correlations can now be included in PNP model.[43–47] This makes future biological computations quite relevant to the extensive experimental literature on channels.

2 Method

2.1 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. Figure 1 gives an illustration of a simple cation-selective ion channel, the gramicidin A channel.

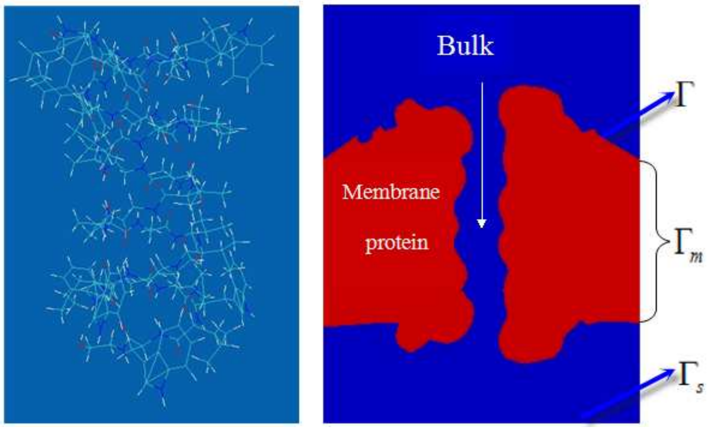

Figure 1: Gramicidin A dimer (left colume). A 2D cut through the center of the simulation box along the z axis illustrates the mesh representation of the protein and the membrane. The membrane and the protein region are shown in red, solvent reservoirs and the channel region are shown in blue (right colume).

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 $\Omega \in \mathbb{R}^3$, $\overline{\Omega} = \overline{\Omega}_m \cup \overline{\Omega}_s$, $\Omega_m \cap \Omega_s = \emptyset$, 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 $\overline{\Gamma} = \overline{\Omega}_m \cap \overline{\Omega}_s$, and Γ_m to denote the membrane boundary on the simulation box. We obtain the PNP equations by coupling the Nernst-Planck equation

[redacted] $x \in \Omega_s, \quad 1 \leq i \leq N,$ (1)

[redacted] (2)

and the electrostatic Poisson equation with internal interface Γ:

$$\begin{aligned} -\nabla \cdot (\varepsilon \nabla \phi) = \lambda \sum_i q_i c_i + \rho^f, &\quad x \in \Omega, \qquad (3) \\ \phi_m = \phi_s, &\quad x \in \Gamma, \\ \varepsilon_m \frac{\partial \phi_m}{\partial n} = \varepsilon_s \frac{\partial \phi_s}{\partial n}, &\quad x \in \Gamma, \end{aligned}$$

where $c_i(x,t)$ is the concentration of the i-th 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 piecewisely constant with $\varepsilon = \varepsilon_m \varepsilon_0$ in Ω_m and $\varepsilon = \varepsilon_s \varepsilon_0$ 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. In later work, three dielectric regions will be included. 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, suggesting that mobile ions are present only in the solvent region.

2.2 Regularization and transformation of PNP system

In this paper, we only consider the steady state PNP equations. An effective strategy for solving eq 3 is to decompose the solution of the Poisson equation into a singular component, a harmonic component and a regular component,[38,48] i.e., $\phi = \phi^s + \phi^h + \phi^r$. The singular component ϕ_s is the restriction on Ω_m of the solution of

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

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

$$-\Delta \phi^h(x) = 0, \quad x \in \Omega_m, \tag{5}$$

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

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

$$-\nabla \cdot (\varepsilon \nabla \phi^r(x,t)) = \lambda \sum_i q_i c_i(x,t), \quad x \in \Omega, \tag{6}$$

and the interface conditions

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

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

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 Poisson-Nernst-Planck equations consist of the regularized Poisson eq 6 and the steady-state Nernst-Planck equations

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

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

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

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[36]

$$\text{[redacted]}, \quad \hat{D}_i = D_i e^{-q_i \beta \phi^r}, \tag{8}$$

the Poisson-Nernst-Planck equations can be transformed as

$$-\nabla \cdot (\varepsilon \nabla \phi^r) = \lambda \sum_i q_i \tilde{c}_i e^{-q_i \beta \phi^r}, \tag{9}$$

$$\nabla \cdot (\hat{D}_i \nabla \tilde{c}_i) = 0. \tag{10}$$

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.

2.3 Finite element discretization

The PNP equations are solved using finite element method 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 Poisson equation

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

$$u = \varphi, \text{ on } \partial\Omega, \tag{12}$$

$$[u] = 0, [\varepsilon \frac{\partial u}{\partial n}] = \varepsilon_m \frac{\partial(\phi^s + \phi^h)}{\partial n}, \text{ on } \Gamma. \tag{13}$$

Let $H_b^1(\Omega) = \{u \in H^1(\Omega) \mid u = \varphi \text{ on } \partial\Omega\}$, 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:

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

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

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

$$\begin{aligned} \int_\Omega (\varepsilon \nabla u \nabla v) d\Omega &= \int_{\Omega_s} \sum_i (q_i \tilde{c}_i e^{-q_i \beta u}) v d\Omega_s \\ &- \varepsilon_m \int_\Gamma \frac{\partial(\phi^s + \phi^h)}{\partial n} v dS, \quad \forall v \in H_0^1(\Omega). \end{aligned} \tag{15}$$

Compared with the original PE, these transformations lead to a nonlinear part of 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 n-th 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 $\mathbb{R}^M$, i.e., $u_n = \sum_{k=1}^{M} u_k^n \Phi^i$, where u_k^n is the k-th degree of freedom (DOF) in the n-th Newton iteration. We define a nonlinear function $F(u_n)$ ($\mathbb{R}^M \to \mathbb{R}^M$) whose j-th component is given by:

$$\begin{aligned} F_j(u_n) &= \int_\Omega (\varepsilon \nabla u_n \nabla \Phi_j) d\Omega - \int_{\Omega_s} \sum_i (q_i \tilde{c}_i e^{-q_i \beta u_n}) \Phi_j d\Omega_s \\ &+ \varepsilon_m \int_\Gamma \frac{\partial(\phi^s + \phi^h)}{\partial n} \Phi_j dS, \quad j = 1 \cdots M. \end{aligned} \tag{16}$$

Then the Newton iteration of the Poisson equation 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:

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

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

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{18}$$

where the stiffness matrix $\mathbf{A} = [\mathbf{A}_{\mathbf{j},\mathbf{k}}]_{\mathbf{M}\times\mathbf{M}} = [\mathbf{F}'_{\mathbf{j},\mathbf{k}}(\mathbf{u_n})]_{\mathbf{M}\times\mathbf{M}}$, the vector $\mathbf{f} = [\mathbf{f_j}]_{\mathbf{M}} = [\mathbf{F_j}(\mathbf{u_n})]_{\mathbf{M}}$, and the solution vector $\mathbf{u} = [\mathbf{u_n^k} - \mathbf{u_{n+1}^k}]_{\mathbf{M}}$. We start from an initial guess u_0, which should satisfy the boundary condition $u_0 = \varphi$ on $\partial\Omega$, 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, $1 \le i \le N$, find $\tilde{c}_i \in H_a^1(\Omega_s)$ which satisfies

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

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

For a discrete solution to eq 19, 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 denote a subspace of $H_a^1(\Omega_s)$ by $\widetilde{V}^h = \text{span}\{\psi^1, \cdots, \psi^L, \psi^{L+1}, \cdots, \psi^{L+T}\}$, with $\psi^{L+1}, \cdots, \psi^{L+T}$ denoting 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:

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

where $\tilde{c_i}^j$ is the j-th DOF of the ion concentration, and the discrete weak form is given by

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

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

$$\begin{aligned}
\int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \tilde{c_i}^h \nabla \psi^j) d\Omega_s &= \sum_{k}^{L} [\tilde{c_i}^k \int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^k] d\Omega_s && (22) \\
&+ \sum_{s}^{T} [\eta_i(A_{L+s}) \int_{\Omega_s} (D_i e^{-q_i \beta u} \nabla \psi^j \nabla \psi^{s+L}] d\Omega_s. && (23)
\end{aligned}$$

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

$$\mathbf{B}\mathbf{x} = \mathbf{y}, \tag{24}$$

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

2.4 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 macromolecule exists. 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[40] is described by

$$\begin{aligned} u^{\text{new}} &= \alpha u^{\text{old}} + (1-\alpha)u^{\text{new}} \\ \tilde{c}_i^{\text{new}} &= \alpha \tilde{c}_i^{\text{old}} + (1-\alpha)\tilde{c}_i^{\text{new}}, \quad i = 1,\ldots,N \end{aligned}$$

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

2.5 Mesh generation for ion channel system

Our finite element algorithms use tetrahedral meshes. Mesh generation is a prerequisite for finite element method. 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 finite element method 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 the surface mesh.[40] 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,[49,50] which, we hope, will facilitate the finite element simulations of biomolecular systems.

In this work, we have built a tools chain for high-quality biomolecule volume mesh generation by using TMSmesh and a few other meshing tools. The tools chain has essentially three components: surface meshing, quality improving, and volume mesh generation. First, a triangulation of the Gaussian surface is generated using our recently developed program TMSmesh,[49] 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.[50] The surface meshes produced by the new version of TMSmesh are manifold mesh 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 firstly use the program ISO2Mesh[51] 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,[52] 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,[53] which consists of 4-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.[54]

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[55] file format which can be read by PHG.

2.6 Adaptive finite element method

The adaptive finite element method (AFEM) was originally proposed by Ivo Babuska et al.[56] 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), i.e., 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:[57]

$$\eta_s = \left(h_s^2 \parallel \sum_i q_i c_i + \rho^f + \nabla \cdot (\varepsilon \phi_h) \parallel_{L^2(s)}^2 + \frac{1}{2} \sum_{f \in F(s)} h_f \parallel [(\varepsilon \phi_h) \cdot n_f] \parallel_{L^2(f)}^2 \right)^{1/2} \tag{25}$$

where s represents an arbitrary element in the mesh and η_s is the error indicator on s, $F(s)$ denotes the set of (non-boundary) 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 M. Holst[58] if we consider steady-state diffusion process.

Our adaptive finite element solver for the PNP system is implemented using the toolbox PHG.[1] PHG is a parallel toolbox for writing adaptive finite element programs. It is developed at the State Key Laboratory of Scientific and Engineering Computing of Chinese Academy of Sciences and is featured by supporting bisection based conforming parallel adaptive tetrahedral meshes and the ability to scale to thousands of 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 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.

3 Numerical test with gramicidin A ion channel

In this section, we apply our parallel finite element simulator, ichannel, to gramicidin A ion channel to compute the electrostatic potential, ion concentrations and $I-V$ curve under various combinations of inputs. 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

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

where T_x denotes the execution time (wall-clock time) when using x processes in the computation, and p_0, $1 \leq p_0 \leq p$, 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.

3.1 Ion channel simulation

One of the most widely studied ion channels is gramicidin A (gA) channel (PDB code: 1MAG) which forms aqueous pores in lipid bilayers that selectively pass monovalent cations.[59,60] GA is a small 15 amino acid β helical peptide with a narrow pore. Because it is relatively small and well characterized experimentally, a wide variety of theoretical models has 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.[61] The gA channel pore region is along the z direction. The box size is 30 Å $\times$ 30 Å $\times$ 45 Å. The membrane layer is represented as a slab with a length of 21 Å along z direction (from $z = 11$ Å to $z = 32$ Å).

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 8 shows an example of the unstructured tetrahedral volume mesh and triangular surface mesh of gramicidin A ion channel. The mesh over the whole domain has a total of 22753 vertices and 142954 tetrahedra.

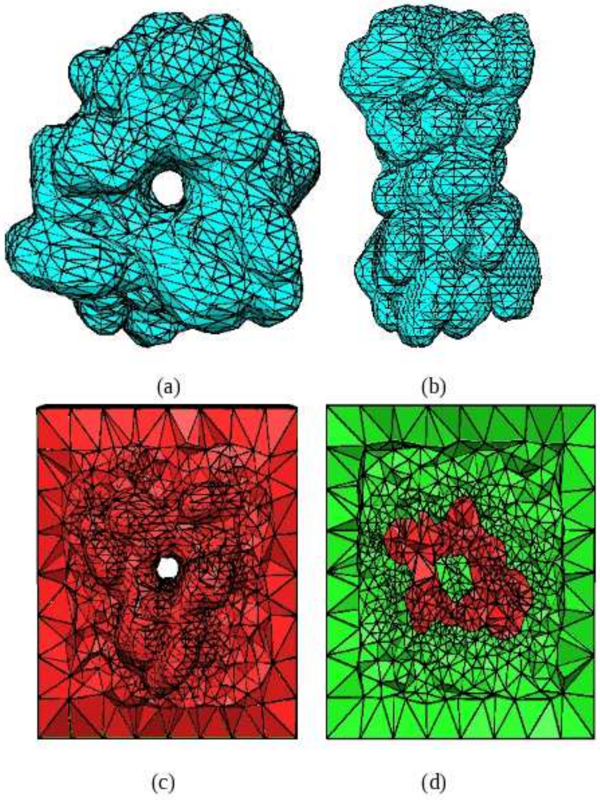

Figure 2: Triangular boundary mesh conforming to the gramicidin A ion channel surface: (a) top view. (b) lateral view. (c) Boundary surface mesh of ion channel with the membrane which is represented as a slab. (d) A view of cross section of the whole tetrahedral volume mesh.

In the following computations, the membrane and protein regions (red area in Figure 1) are described by low dielectric constant $\varepsilon_m = 2$. The high dielectric constant $\varepsilon_s = 80$ is assigned to the aqueous region, i.e., the volume outside of the protein-membrane region (blue region in Figure 1).

The diffusion coefficients for cation and anion, e.g., K^+ and Cl^-, in the bulk region are set to their experimental values: $D_{\mathrm{Cl}} = 0.203$ 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. 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:[62]

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

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

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

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, $z_{\mathrm{chan}} = 11$ and $z_{\mathrm{bulk}} = 9$. For the top boundary, $z_{\mathrm{chan}} = 32$ and $z_{\mathrm{bulk}} = 34$. 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 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:

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

$$\phi(x=0,y,z)=\phi(x=L_x,y,z)=\frac{V_{\text{applied}}}{L_z}\cdot z, \tag{30}$$

$$\phi(x,y=0,z)=\phi(x,y=L_y,z)=\frac{V_{\text{applied}}}{L_z}\cdot z, \tag{31}$$

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

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

$$c_i(x,y=0,z\in\text{bulk region})=c_i(x,y=L_y,z\in\text{bulk region})=c_{i,\text{bulk}}, \tag{34}$$

$$J_i\cdot n=0 \text{ on } \Gamma. \tag{35}$$

We solve the coupled eqs 9 and 10 to obtain the steady-state ion concentrations and electrostatic potential. For a given boundary condition ($V_{\text{applied}}=100\ mV$ and $c_{i,\text{bulk}}=1.0\ M$), 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 (non-equilibrium state) for the same channel system, Figure 4 shows a comparison between these two potential profiles.

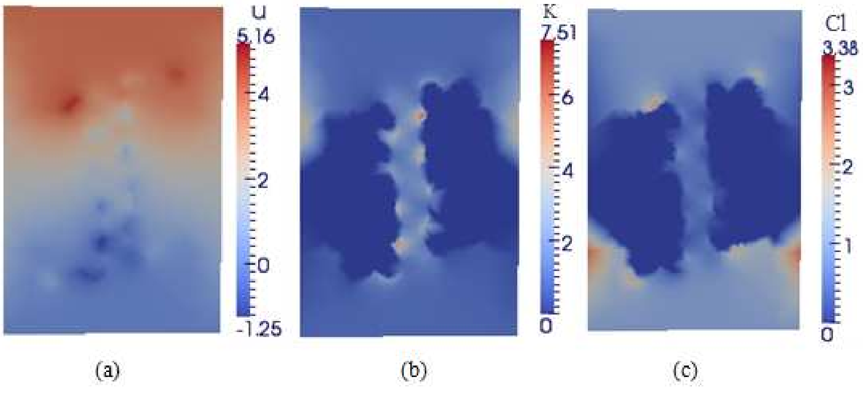

Figure 3: Electrostatic potential (k_BT/e_c) and ion concentration (M). (a) is a cross section view of the electrostatic potential of the whole domain. (b) is a cross section view of K^+ ion concentration of the whole domain. (c) is a cross section view of Cl^- ion concentration of the whole domain.

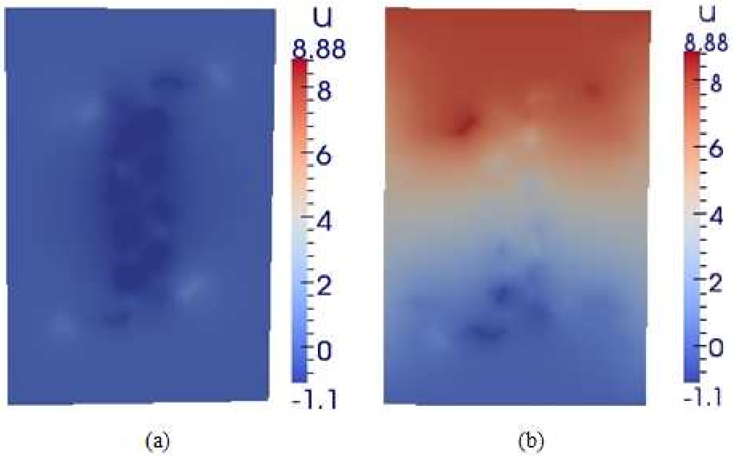

Figure 4: Comparison of electrostatic potential (k_BT/e_c) between PB and PNP calculations. (a) is a cross section view of the electrostatic potential obtained from PB calculation with the Debey-H$\ddot{u}$ckel boundary condition and with the same bulk ion concentrations as in the compared PNP calculation. (b) is a cross section view of the electrostatic potential obtained from PNP calculation with $V_{\text{applied}} = 200\ mV$ and $c_{i,\text{bulk}} = 0.5\ M$.

The electrical current across the pore can be calculated as:

$$I_z = -\sum_i q_i \int_{L_x,L_y} D_i(\frac{\partial c_i}{\partial z} + \frac{q_i}{k_BT} c_i \frac{\partial \phi}{\partial z})dxdy. \tag{36}$$

Eq 36 can be applied at any z-position along the pore axis, and shows only minor differences in the current values 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 examples, $V_{\text{applied}} = 0\ mV$, $50\ mV$, $100\ mV$, $150\ mV$ and $200\ mV$, and $c_{i,\text{bulk}} = 0.1\ M$, $0.2\ M$, $0.5\ M$, $1.0\ M$ and $2.0\ M$. The potential profile for five different bulk concentrations with the same voltage ($V_{\text{applied}} = 100\ mV$) 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 is the same ($c_{i,\text{bulk}} = 0.5\ M$). It is seen that although different voltage values as boundary condition are applied, the changes of concentrations have almost the same

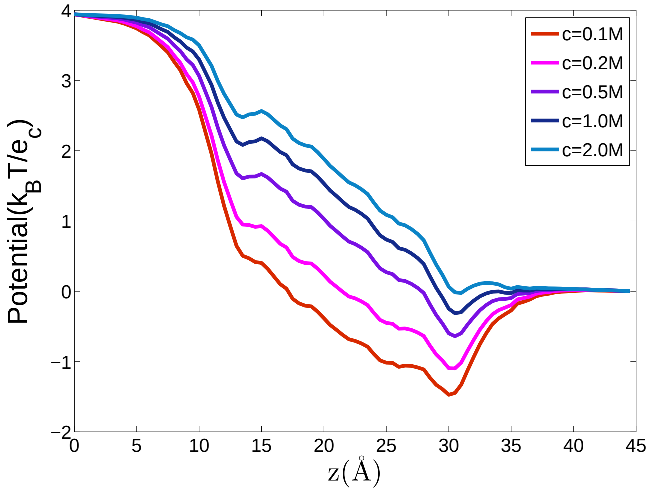

Figure 5: Electrostatic potential at the center of the gA channel plotted along z-axis obtained from the PNP calculation with $V_{\text{applied}} = 100\ mV$.

tendency. It is seen that the Cl^- concentration inside the gA is not exactly zero in our simulation, though it should be zero experimentally. But we think this would be a common problem of tranditional mean-field continuum model, such as PNP model.

The experimental $I-V$ data are obtained from Cole et al.,[63] which are used as the reference data for the 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 channel pore through matching the experiment data, i.e., 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, i.e, $D_{\text{Cl}} = 1.13\times10^{-2}$ Å^2/ps, $D_K = 1.09\times10^{-2}$ 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 the agreement is better at low concentration than at high concentration. It is a defect of the standard PNP model. In future, the size modified PNP model is to be used for the ion channel simulation.

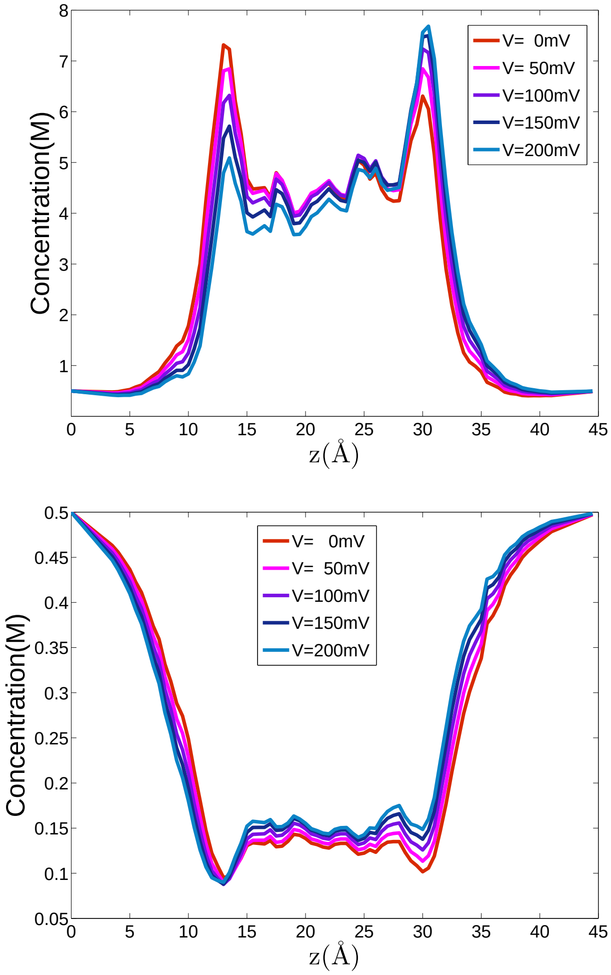

Figure 6: K^+ (top) and Cl^- (bottom) concentrations at the center of the gA channel plotted along z-axis obtained from the PNP calculation with $c_{i,\mathrm{bulk}} = 0.5\ M$.

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

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

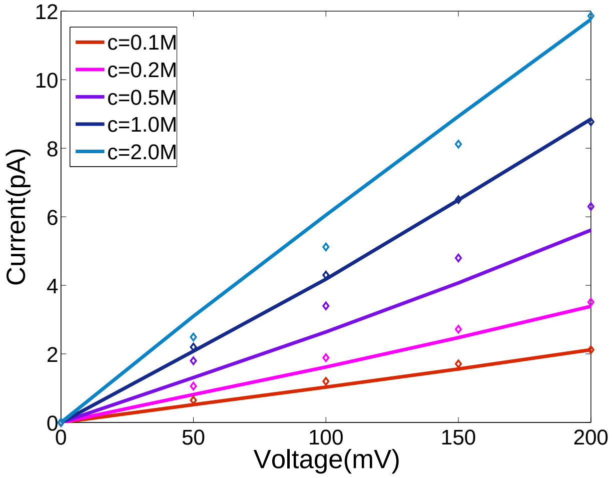

Figure 7: Comparisons of the computed $I-V$ curves with experimental data. $I-V$ curves are plotted for five bulk ion concentrations and the squares represent the experimental data.

3.2 Convergence rate and parallel efficiency

With the transformation method, the Poisson equation 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 the transformation method than with the primitive method. This is true when solving the PNP equations for gA channel, as demonstrated by the numbers of iterations in Table 2, in which a same relaxation parameter $\alpha = 0.8$ is used in the external iterations of all the computations.

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 1523013 vertices and 9149056 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

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	Primitive	Transformed
$0.1\ M, 50\ mV$	156	16
$0.1\ M, 100\ mV$	156	16
$0.1\ M, 200\ mV$	159	16
$0.5\ M, 50\ mV$	154	21
$0.5\ M, 100\ mV$	154	21
$0.5\ M, 200\ mV$	156	21
$2.0\ M, 50\ mV$	158	23
$2.0\ M, 100\ mV$	154	23
$2.0\ M, 200\ mV$	155	23

processes, which we believe is caused the interconnection topology of the underlying InfiniBand network. The code is expected to be able to solve larger systems with more CPUs through using computers with faster interconnection network and/or exploiting the MPI+OpenMP two level parallelism provided by PHG.

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

Num of Procs	Num of Iters	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%

One feature of PHG is that one can easily change the type of element used in the computations without changing the code. By exploiting this feature, we have experimented with a few high order elements, including quadratic and cubic elements. We find that with high order elements one can achieve at a given error tolerance using much fewer DOF than with the linear element.

4 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.

In this paper, we present a parallel adaptive finite element simulator, ichannel, for ion transport through ion channel systems. Numerical applications are carried out with the gramicidin A 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. Moreover, due to 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 built a tools chain for high-quality biomolecule mesh generation by using a few of mesh generation tools including the surface meshing tool developed by us recently.

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 Poisson equation 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. Further examinations and applications of the simulator to complex, large ion channel are under way. However, there are a number of inherent problems in standard-PNP model. Such as, the PNP theory neglects the finite volume effect of ion particles, which can be important for narrow channel pores, and non-electrostatic interactions between ions are not accounted in the PNP model. Therefore, improved continuum models such as the size modified PNP model[44] and coupled PNP/ECP[64] model are also under investigation. We aim to develop an user-friendly software platform for ion transport through three-dimensional 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.[65]

Acknowledgement

B. Tu and B.Z. Lu were supported by the State Key Laboratory of Scientific/Engineering Computing, National Center for Mathematics and Interdisciplinary Sciences, the Chinese Academy of Sciences and the China NSF (NSFC10971218). Y. Xie and L.B. Zhang were supported by National 973 Project of China (2011CB309703), National 863 Project of China (2012AA01A3094), China NSF (11171334 and 11021101), and National Center for Mathematics and Interdisciplinary Sciences of Chinese Academy of Sciences. M.X. Chen was supported by the China NSF (NSFC11001062) and the Collegiate NSF of Jiangsu Province (No.11KJB110010).

References

(1) Zhang, L. B. *Numer. Math. Theor. Meth. Appl.* **2009**, *2*, 65–89.

(2) Hille, B. *Ion Channels of Excitable Membranes*, 3rd ed.; Sinauer, 2001.

(3) Hacker, M.; Messer, W.; Bachmann, K. *Pharmacology: Principles and Practice*; Academic Press: New York, 2009.

(4) Gabashvili, I.; Sokolowski, B.; Morton, C.; Giersch, A. *J. Assoc. Res. Otolaryngol.* **2007**, *8*, 305–328.

(5) Conley, E. C. *The Ion Channel Facts Book. I. Extracellular Ligand-gated Channels*; Academic Press: New York, 1996.

(6) Conley, E. C. *The Ion Channel Facts Book. II. Intracellular Ligand-gated Channels*; Academic Press: New York, 1996.

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

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

(9) Doyle, D. A.; Cabral, J. M.; Pfuetzner, R. A.; Kuo, A.; Gulbis, J. M.; Cohen, S. L.; Chait, B. T.; MacKinnon, R. *Science* **1998**, *280*, 69–77.

(10) Song, L.; Hobauch, M. R.; Shustak, C.; Cheley, S.; Bayley, H.; Gouaux, J. E. *Science* **1996**, *274*, 1859–1866.

(11) Ketchem, R. R.; Roux, B.; Cross, T. A. *Structure* **1997**, *5*, 1655–1669.

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

(13) Chung, S.; Kuyucak, S. *Biochim. Biophys. Acta* **2002**, *1565*, 267–286.

(14) Hwang, H.; Schatz, G. C.; Ratner, M. A. *J. Phys. Chem. A* **2007**, *111*, 12506–12512.

(15) Li, S.; Hoyles, M.; Kuyucak, S.; Chung, S. *Biophys. J.* **1998**, *74*, 37–47.

(16) Corry, B.; Kuyucak, S.; Chung, S. H. *Biophys. J.* **2000**, *78*, 2364–2381.

(17) Kuyucak, S.; Andersen, O. S.; Chung, S. H. *Rep. Prog. Phys.* **2001**, *64*, 1427–1472.

(18) Im, W.; Roux, B. *J. Mol. Biol.* **2002**, *322*, 851–869.

(19) Schirmer, T.; Phale, P. S. *J. Mol. Biol.* **1999**, *294*, 1159–1167.

(20) Allen, T.; Hoyles, M.; Kuyucak, S.; Chung, S. *Chem. Phys. Lett.* **1999**, *313*, 358–365.

(21) Noskov, S.; Im, W.; Roux, B. *Biophys. J.* **2004**, *87*, 2299–2309.

(22) K. Lee, R. P., H. Rui; Im, W. *Biophys. J.* **2011**, *100*, 611–619.

(23) Chen, D.; Lear, J.; Eisenberg, R. S. *Biophys. J.* **1997**, *72*, 97–116.

(24) Lu, B. Z. Poisson-Nernst-Planck equations. In *Encyclopedia of Applied and Computational Mathematics;* Engquist, B., Ed.; *Springer*, in press.

(25) Nernst, W. *Z. Physik. Chem.* **1889**, *4*, 129.

(26) Planck, M. *Ann. Phys. Chem.* **1890**, *39*, 161.

(27) Schuss, Z. B. N.; Eisenberg, R. S. *Phys. Rev. E* **2001**, *64*, 1–14.

(28) Riveros, O. J.; Croxton, T. L.; Armstrong, W. M. *J. Theor. Biol.* **1989**, *140*, 221–230.

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

(30) Eisenberg, R. S. *Acc. Chem. Res.* **1998**, *31*, 117–123.

(31) Roux, B.; Allen, T.; Berneche, S.; Im, W. *Q. Rev. Biophys.* **2004**, *7*, 1–103.

(32) Liu, W. S. *SIAM J. Appl. Math.* **2005**, *65*, 754–766.

(33) Barcilon, V.; Chen, D.; Eisenberg, R.; Jerome, J. *SIAM J. Appl. Math.* **1997**, *57*, 63–648.

(34) Abaid, N.; Eisenberg, R. S.; Liu, W. S. *SIAM J. Appl. Dyn. Syst.* **2008**, *7*, 1507–1526.

(35) Eisenberg, R. S.; Liu, W. S. *SIAM J. Math. Anal.* **2008**, *38*, 1932–1966.

(36) Kurnikova, M. G.; Coalson, R. D.; Graf, P.; Nitzan, A. *Biophys. J.* **1999**, *76*, 642–656.

(37) Zheng, Q.; Chen, D.; Wei, G. W. *J. Chem. Phys.* **2011**, *230*, 5239–5262.

(38) Lu, B. Z.; Holst, M. J.; McCammon, J. A.; Zhou, Y. C. *J. Comput. Phys.* **2010**, *229*, 6979–6994.

(39) Lu, B. Z.; McCammon, J. A. *Chem. Phys. Lett.* **2008**, *451*, 282–286.

(40) Lu, B. Z.; Zhou, Y. C.; Huber, G. A.; Bond, S. D.; Holst, M. J.; McCammon, J. A. *J. Chem. Phys.* **2007**, *127*, 135102.

(41) Hollerbach, U.; Chen, D. P.; Eisenberg, R. S. *J. SCI. COMPUT.* **2002**, *16*, 373–409.

(42) Mathur, S. R.; Murthy, J. Y. *SIAM J. APPL. MATH.* **2009**, *52*, 4031–4039.

(43) Eisenberg, R. S.; Hyon, Y. Y.; Liu, C. *J. Chem. Phys.* **2010**, *133*, 104104.

(44) Lu, B. Z.; Zhou, Y. C. *Biophys. J.* **2011**, *100*, 2475 – 2485.

(45) Horng, T. L.; ; Lin, T. C.; Liu, C.; Eisenberg, B. *J. Phys. Chem. B* **2012**, *116*, 11422–11441.

(46) Zhou, S. G.; Wang, Z. M.; Li, B. *Phys. Rev. E* **2011**, *84*, 021901.

(47) Eisenberg, B. *SIAM News* **2012**, *45*, 11–12.

(48) Hollerbach, U.; Eisenberg, R. S. *Langmuir* **2002**, *18*, 3262–3631.

(49) Chen, M.; Lu, B. *J. Chem. Theory Comput.* **2011**, *7*, 203–212.

(50) Chen, M. X.; Tu, B.; Lu, B. Z. *J. Mol. Graph. Model.* **2012**, *38*, 411–418.

(51) Fang, Q. Q.; Boas, D. *Proceedings of IEEE International Symposium on Biomedical Imaging 2009* **2009**, 1142–1145.

(52) Zaharescu, A.; Boyer, E.; Horaud, R. P. *In Proceedings of the Eighth Asian Conference on Computer Vision* **2007**, *II*, 166–175.

(53) Si, H. TetGen: A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator. `http://tetgen.berlios.de/ (accessed May 10, 2010)`.

(54) Shewchuk, J. R. *Proceedings of the Fourteenth Annual Symposium on Computational Geometry* **1998**, 86–95.

(55) FREY, P. J. Medit: an interactive mesh visualization software. `http://www.ann.jussieu.fr/~frey/logiciels/medit.html (accessed Dec 20, 2010)`.

(56) Guo, B.; Babuska, I. *Adv. Eng. Softw.* **1992**, *15*.

(57) Berger, M. J.; Colella, P. *J. Comput. Phys.* **1989**, *81*, 64–84.

(58) Holst, M.; Baker, N.; Wang, F. *J. Comput. Phys.* **2000**, *21*, 1319–1342.

(59) Andersen, O. S. *Annu. Rev. Physiol.* **1984**, *46*, 531–548.

(60) Koeppe, R. E.; Andersen, O. S. *Annu. Rev. Biophys. Biomol. Struct.* **1996**, *25*, 231–258.

(61) Dolinsky, T. J.; Nielsen, J. E.; McCammon, J. A.; Baker, N. A. *Nucleic Acids Res.* **2004**, *32*, W665–W667.

(62) Hwang, H.; Schatz, G. C.; Ratner, M. A. *J. Phys. Chem. A* **2007**, *111*, 12506–12512.

(63) Cole, C. D.; Frost, A. S.; Thompson, M. C.; Cross, A.; Busath, D. D. *Biophys. J.* **2002**, *83*, 1974–1986.

(64) Yang, Z.; Straaten, T. A. V. D.; Ravaioli, U.; Liu, Y. *J. Comput. Electron.* **2005**, *4*, 1–2.

(65) Eisenberg, B. *Adv. Chem. Phys.* **2011**, *148*, 77–223.

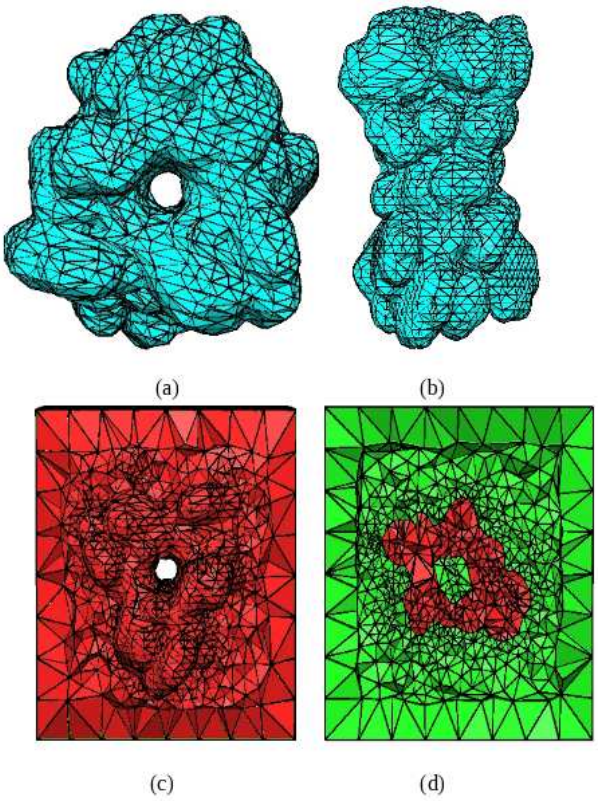

Figure 8: The TOC graphic