

A software platform for continuum modeling of ion channels based on unstructured mesh

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2014 Comput. Sci. Disc. 7 014002

(<http://iopscience.iop.org/1749-4699/7/1/014002>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 124.16.148.3

This content was downloaded on 24/10/2014 at 05:11

Please note that [terms and conditions apply](#).

A software platform for continuum modeling of ion channels based on unstructured mesh

B Tu¹, S Y Bai¹, M X Chen², Y Xie¹, L B Zhang¹ and B Z Lu¹

¹ 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, People's Republic of China

² Center for System Biology, Department of Mathematics, Soochow University, Suzhou 215006, People's Republic of China

E-mail: bzlu@lsectioncc.ac.cn

Received 14 October 2013, revised 16 April 2014

Accepted for publication 26 June 2014

Published 29 July 2014

Computational Science & Discovery 7 (2014) 014002

doi:[10.1088/1749-4699/7/1/014002](https://doi.org/10.1088/1749-4699/7/1/014002)

Abstract

Most traditional continuum molecular modeling adopted finite difference or finite volume methods which were based on a structured mesh (grid). Unstructured meshes were only occasionally used, but an increased number of applications emerge in molecular simulations. To facilitate the continuum modeling of biomolecular systems based on unstructured meshes, we are developing a software platform with tools which are particularly beneficial to those approaches. This work describes the software system specifically for the simulation of a typical, complex molecular procedure: ion transport through a three-dimensional channel system that consists of a protein and a membrane. The platform contains three parts: a meshing tool chain for ion channel systems, a parallel finite element solver for the Poisson–Nernst–Planck equations describing the electrodiffusion process of ion transport, and a visualization program for continuum molecular modeling. The meshing tool chain in the platform, which consists of a set of mesh generation tools, is able to generate high-quality surface and volume meshes for ion channel systems. The parallel finite element solver in our platform is based on the parallel adaptive finite element package PHG which was developed by one of the authors [1]. As a featured component of the platform, a new visualization program, VCMM, has specifically been developed for continuum molecular modeling with an emphasis on providing useful facilities for unstructured mesh-based methods and for their output analysis and visualization. VCMM provides a graphic user interface and consists of three modules: a molecular module, a meshing module

and a numerical module. A demonstration of the platform is provided with a study of two real proteins, the connexin 26 and hemolysin ion channels.

Keywords: ion channels, software platform, finite element method, visualization, Poisson–Nernst–Planck

1. Introduction

Various theoretical and computational approaches have been developed to help understand the biological mechanism of ion channels. The most commonly used theoretical techniques in the field are molecular dynamics simulations [2–4], Brownian dynamics simulations [5–8] and continuum modelings [9–12]. Since the explicit ion methods provide the most accurate description of the systems behavior, in both the spatial and temporal domains, they require computationally expensive simulations to obtain the average properties. Furthermore, the application of an explicit ion method usually requires the system to be described with the same resolution over the entire simulation domain. This often leads to a situation where a majority of the computational effort is applied to simulate a nearly uniform solution where the quantities of interest exhibit little variation. In contrast, the continuum methods allow different regions of the same system to be described with varying levels of detail, and thus allow us to focus the computational effort on regions that require a more precise description. In addition to being more computationally efficient, continuum models can be more convenient when incorporating certain types of boundary conditions that arise in physical systems, such as boundaries of fixed concentration or electrostatic potential.

A widely used continuum model for simulating ionic transport is based on the Poisson–Nernst–Planck (PNP) [10, 13], in which ions are not treated as microscopic discrete entities but instead as continuous charge densities. 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 Nernst–Planck (NP) equations, and the electric field is in turn determined by the concentrations using the Poisson equation. PNP theory has successfully been applied to the study of ion transport in electrochemical liquid junction systems [14] and electron transport in semiconductor devices [15], as well as ion permeation through biological membrane channels [11, 12]. A number of numerical algorithms, including the finite difference (FD) [16, 17], finite element (FE) [18–21], spectral element [22] and finite volume methods [23], have been utilized in the past two decades to solve the PNP equations. There are various packages to solve PNP equations for ion channel simulations, such as HARLEM [24] and PROPHET [25]. HARLEM is a multipurpose interactive molecular modeling package in which the PNP equations are discretized using FD methods. This package is designed to combine modern electronic structure and statistical mechanics techniques controlled by a graphic interface to provide an effective theoretical tool to study large molecules. PROPHET was originally developed for semiconductor simulations. This program is used to obtain solutions of partial differential equations (PDEs), in which the PDEs are mostly discretized using finite volume methods. Although the FD and finite volume methods for structured meshes are straightforward to implement, it is challenging to apply them to systems with curved boundaries and complex geometries. If the surface and volume meshes of proteins are available, then the finite element method (FEM) has the advantage of naturally handling complex geometries, such

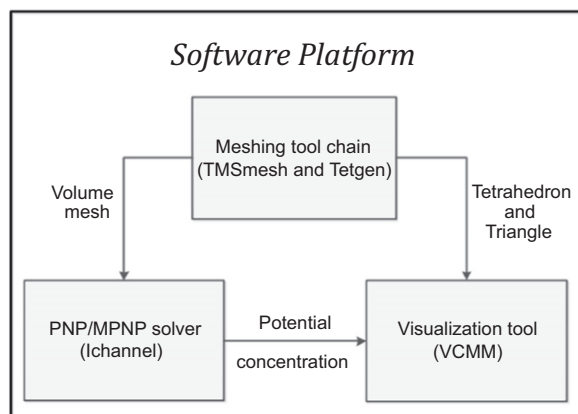


Figure 1. Schematic architecture of the software platform.

as the molecular surfaces of DNA molecules and ion channels. We have recently reported our work on using FEM to solve the three-dimensional (3D) PNP equations for ion channel systems [26]. However, many nontrivial techniques are required for users to conduct an FE simulation of ion permeation in a channel, such as a PDB preprocessed technique to assign a force field and proper orientation, surface and volume mesh generation for a channel protein and membrane system. In addition, the existing molecular visualization software lacks the ability to manage unstructured mesh and analyze specific numerical results, which indicates that the visual aids for FE modeling approaches are currently very limited. In this work, we have greatly enhanced the capability of the visualization tools to cope with unstructured mesh, which is the main addition to our previous work.

Therefore, to integrate the tools and facilities that have been developed or are currently in development, and to make it convenient to conduct FE simulations, we present a user-friendly software platform for the simulation of ion channels. The platform contains three parts: a meshing tool chain, a parallel FE solver and a visualization program VCMM. The structure of the platform is shown in figure 1.

Due to highly irregular shapes, mesh generation for biomolecular systems is widely considered to be a challenging task. We have built a tool chain to obtain high-quality surface and volume meshes for large biomolecules [27, 28]. In addition, for ion channel systems, which contain a protein and a membrane, embedding a membrane slab representation in a tetrahedral mesh is also tricky. A parallel finite element solver has been developed for PNP equations, the effectiveness of which was validated by several numerical tests [26]. To enhance the robustness of our previous FE solver, stabilized FEM methods [29] are being researched to solve PNP equations for ion channel simulations. The VCMM program is a featured component of the software platform which is designed for continuum molecular modeling, and specifically for unstructured mesh-based approaches. Many existing molecular visualization programs, such as Pymol [30], VMD [31] and GRASP [32], only support the visualization of molecular structure and the analysis of structured mesh-based results, such as an FD Poisson–Boltzmann electric field. For FE simulations, users have to combine some third-party tools, such as ParaView [33] and TetView [34], to achieve separately specific visualization features. Furthermore, current molecular visualization software does not provide interfaces for unstructured mesh-based methods such as the finite element/boundary element (BE) methods. On this platform, our

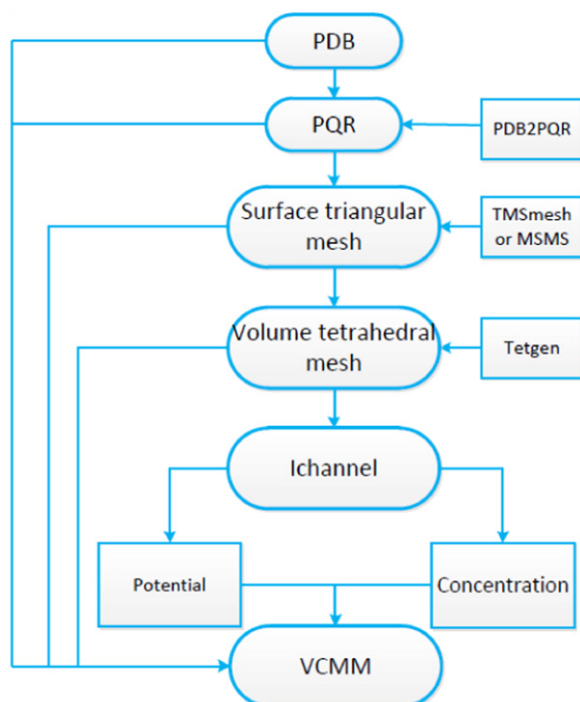


Figure 2. A work-flow chart of the platform to simulate an ion channel system.

recently developed program, VCMM, extends the visualization strengths by adding the management of unstructured meshes. By further adding the meshing tool to the chain and numerical solvers into VCMM as plug-ins, the interface can be used to conduct the whole procedure of the ion transport simulation.

The rest of the paper is organized as follows. Section 2 introduces in detail the platform for the simulation of ion transport through the ion channel systems. First, we briefly review the 3D ion channel model and the PNP equations system. Then, we introduce our finite element solver of PNP equations and a tool chain for obtaining the surface and volume meshes for ion channel systems. Finally, we introduce a new visualization program for continuum molecular modeling. In section 3, we present examples to apply the platform to ion channel simulations. The paper ends with a summary in section 4.

2. The software platform description

In this section, we introduce each part of the platform in detail following a work-flow chart as described in figure 2. The core parts include a tool chain for meshing ion channel systems, a parallel finite element solver for PNP equations and a visualization program.

2.1. System setup

An ion channel system typically consists of a pore-forming protein and membrane. To numerically model the system, a simulation box containing the protein and membrane is usually required. The coordinates of the heavy atoms of the protein can be taken from the Protein Data Bank (PDB). The partial charges and radii for protein atoms can be consistently taken from a

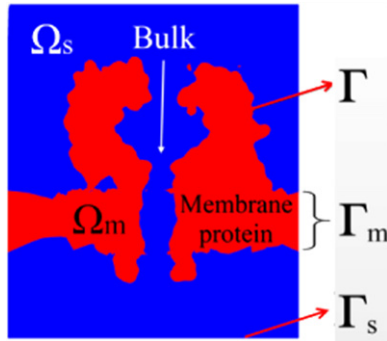


Figure 3. Two-dimensional cut through the center of the simulation box along the z axis illustrating the mesh representation of the protein and the membrane. The membrane and the protein region are shown in red, and the solvent reservoirs and the channel region are shown in blue.

force field, such as AMBER, CHARMM or OPLS. Today, such information is often stored in a PQR format file. The PQR file of the protein can be obtained by using the open-source software PDB2PQR [35], which is also integrated into our platform. The PQR file is used as an input file for meshing the protein and solving the PNP equations. The protein pore is aligned along the z axis. The size of the simulation box can be adaptively changed with the size of the protein. There are two strategies used to model the membrane for the ion channel simulations: one method is to adopt the real lipid which can be obtained either from PDB or by using certain membrane construction tools, and the other method is to adopt a slab to represent the membrane. Both strategies may be implemented in this platform. In this work, the membrane is represented as a slab and no charge is assigned to it.

2.2. The governing equations

The PNP model couples 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. We consider an open domain $\Omega \in \mathbb{R}^3$, $\bar{\Omega} = \bar{\Omega}_m \cup \bar{\Omega}_s$, $\Omega_m \cap \Omega_s = \emptyset$, where Ω_m represents the protein and membrane region and Ω_s represents the solvent reservoirs and the channel region, which is shown in figure 3.

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. The PNP equations are obtained by coupling the NP equation

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

$$J_i = -D_i(x) \left(\nabla c_i(x, t) + \beta q_i c_i(x, t) \nabla \phi(x) \right), \quad (2)$$

and the electrostatic Poisson equation

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

where $c_i(x, t)$ is the concentration of the i th ion species carrying a charge q_i . $D_i(x)$ is the spatially dependent diffusion coefficient, and $\phi(x)$ is the electrostatic potential. N is the number of diffusive ion species in the 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 a piecewise constant with $\epsilon(x) = \epsilon_m(x)\epsilon_0$ in Ω_m and $\epsilon(x) = \epsilon_s(x)\epsilon_0$ in Ω_s , where ϵ_0 is the dielectric constant of the vacuum. Typical values of $\epsilon_m(x)$ and $\epsilon_s(x)$ are chosen as 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 only present in the solvent region.

2.3. Finite element discretization

FEM is an efficient method used to solve a general PDE system, which uses variational methods to minimize the error function and produce a stable solution. Usually, FEM solves the weak form of a PDE, and the weak form is discretized by representing the solution in a finite element space expanded by the given bases. Then, the solution of the PDE can be obtained by solving linear or nonlinear algebraic system(s).

In our platform, we use FEM to solve the PNP equations. We now describe the numerical algorithms employed for the steady-state PNP equations. For the boundary condition, a 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 the top and bottom faces to ensure charge neutrality in the reservoirs. Additionally, there is a no-flux boundary surrounding the protein and membrane that prevents ions from penetrating through the region occupied by the protein and lipids.

If we let $u = \phi^r$ for the NP equations, then the weak form is obtained by integrating against a test function $v \in H^1(\Omega_s)$. Here, $H^1(\Omega)$ denotes a Sobolev space of weakly differentiable functions defined in the domain Ω .

For each i , $1 \leq i \leq N$, we find $c_i \in H_a^1(\Omega_s)$, which satisfies

$$\int_{\Omega_s} D_i (\nabla c_i \nabla v + \beta q_i c_i \nabla u \nabla v) d\Omega_s = 0, \quad \forall v \in H_c^1(\Omega_s), \quad (4)$$

where $H_a^1(\Omega) = \{c_i \in H^1(\Omega) | c_i = \eta_i \text{ on } \Gamma_s\}$; here, η_i denotes the dirichlet boundary function; and $H_c^1(\Omega) = \{c_i \in H^1(\Omega) | c_i = 0 \text{ on } \Gamma_s\}$. $H_a^1(\Omega)$ and $H_c^1(\Omega)$ are Sobolev spaces of weakly differentiable functions which satisfy certain conditions on the boundary of the domain $\partial\Omega$.

To find a discrete solution to equation (4), we denote the discretized approximation of c_i by c_i^h . We employ a finite element space of $V^h = \text{span}\{\psi^1, \dots, \psi^L\}$, with L denoting the number of degrees of freedom (DOF) in the finite element space. We also denote a subspace of $H_a^1(\Omega_s)$ by $\tilde{V}^h = \text{span}\{\psi^1, \dots, \psi^L, \psi^{L+1}, \dots, \psi^{L+T}\}$ with $\psi^{L+1}, \dots, \psi^{L+T}$ denoting the finite element bases on the vertices A_{L+1}, \dots, A_{L+T} of the dirichlet boundary.

We denote the approximate solution c_i^h by its expansion with respect to the finite element bases as follows:

$$c_i^h = \sum_{j=1}^L c_i^j \psi^j + \sum_{s=1}^T \eta_i(A_{L+s}) \psi^{L+s} \in \tilde{V}^h, \quad (5)$$

where c_i^j is the j th DOF of the ion concentration and the discrete weak form is given by

$$\int_{\Omega_s} D_i (\nabla c_i^h \nabla \psi^j + \beta q_i c_i^h \nabla u \nabla \psi^j) d\Omega_s = 0, \quad \forall \psi^j \in \{\psi^1, \dots, \psi^L\}. \quad (6)$$

Then, we obtain a linear system of equations in the following form:

$$\mathbf{B}\mathbf{x} = \mathbf{y}, \quad (7)$$

where the stiffness matrix $\mathbf{B} = [\mathbf{B}_{j,k}]_{\mathbf{L} \times \mathbf{L}} = [\int_{\Omega_s} D_i (\nabla \psi^j \nabla \psi^k + \beta q_i \psi^j \nabla u \nabla \psi^k) d\Omega_s]_{\mathbf{L} \times \mathbf{L}}$, the vector $\mathbf{y} = [\mathbf{y}_j]_{\mathbf{L}} = [-\sum_s^T [\eta_i(A_{L+s}) \int_{\Omega_s} D_i (\nabla \psi^{s+L} \nabla \psi^j + \beta q_i \psi^{s+L} \nabla u \nabla \psi^j) d\Omega_s]_{\mathbf{L}}$ and the solution vector $\mathbf{x} = [\mathbf{c}_i^k]_{\mathbf{L}}$. Similarly, we also use a finite element discretization for solving the Poissons equation. More details and discussion can be found in [26].

Because our PNP solver is based on the finite element toolbox PHG, it can deal with the volume integral, but for other integral techniques, additional functions that are not available in PHG need to be added. If the user wants to solve some modifications of PNP, such as a size-modified PNP model [9], then van der Waals interaction-included PNP models [19, 36, 37], the weak form and the corresponding stiffness need to be modified. This requires further modification of the FEM codes for solving the PNP equation.

2.4. Mesh generation for the ion channel system

Mesh generation is a prerequisite for finite element methods. Our finite element algorithms use tetrahedral meshes. A reasonable strategy to generate biomolecular meshes follows two steps: first, generate a molecular surface conforming mesh, and then generate a volume mesh based on the surface mesh [20]. Of the two steps, surface meshing is the more difficult task. Recently, we have developed a tool called TSMesh that is capable of generating manifold surface meshes for arbitrarily large molecular systems [27, 28], which will facilitate the finite element simulations of ion channel systems. A very recent development in this direction using a different approach can be found in [38].

In the platform, a tool chain combines TSMesh and some other meshing tools to generate the surface and volume meshes for the ion channel proteins. The tool chain essentially has four components: surface meshing, quality improvement, volume mesh generation and membrane mesh construction. First, a triangulation of the Gaussian surface is generated using our recently developed program TSMesh [27]. The surface meshes generated by the old version of

TMSmesh for large molecules sometimes have geometric defects such as 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 [28]. In the second step, if necessary, the surface mesh quality is further improved for volume mesh generation. First, a free matlab/octave-based mesh generation and processing tool, ISO2mesh, is used to simplify the surface mesh by reducing the number of faces or adding some nodes while preserving its manifoldness, volume and boundary shape. ISO2mesh can read the OFF format file exported from TMSmesh and export the filtered molecular surface as an OFF format file. Subsequently, if self-intersecting faces exist, then the program TransforMesh [39], which can robustly handle topology changes and remove self-intersections, is used to find and remove self-intersecting faces. Finally, a tetrahedral volume mesh is generated using the program TetGen [34].

Additionally, we have the membrane for a complete ion channel system for the ion channel simulations. In our platform, there are two strategies to model the membrane for the ion channel simulations: one strategy is to adopt the real lipid structure obtained from the PDB, and the other is to adopt a slab representing the membrane. Both strategies can be conveniently implemented in this platform.

Here, we introduce how to add the membrane represented as a slab to the ion channel system in the platform. It is difficult to generate the membrane mesh. Here, a portion of the volume mesh is extracted to represent the membrane region. The membrane mesh is obtained in 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 four of their 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 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.

An example of the mesh for the whole ion channel system is illustrated in figure 4. The meshing tool chain is integrated into our visualization software as described below, VCMM [40], which makes it more convenient and friendly for mesh generation in ion channel systems.

2.5. Visualization

Visualization is an essential aspect of modern computer simulation studies. There have been many molecular visualization tools; today, Pymol [30] and VMD [31] are the most popular. GRASP is another program that places particular emphasis on the display and manipulation of the surfaces of molecules and their electrostatic properties [32]. However, the software often lacks the capabilities of unstructured mesh management and visual analysis for numerical results based on unstructured meshes. Furthermore, the molecular visualization software does not provide any interface for FEM simulations. Therefore, molecular visualization software with these functions is an urgent need.

In our platform, we have developed a visualization program, VCMM [40], for continuum molecular modeling. VCMM focuses on treating the data set based on the unstructured mesh

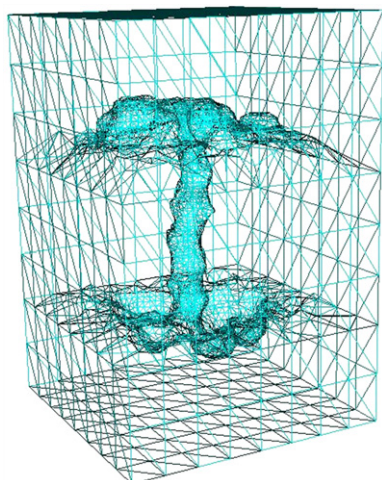


Figure 4. Wire-frame of volume mesh conforming to the boundary of a channel protein and membrane system.

which is used in finite element simulations. VCMM provides a graphic user interface and consists of three modules: the molecular module, the meshing module and the numerical module. A data manager is designed in VCMM to handle various types of data (e.g. PDB [41], PQR, OFF, MESH [42] and visualization toolkit (VTK) [43]) which is used in all modules.

In the molecular module, VCMM supports the most common representations of molecular structures: ball-and-stick, spheres, wire bonds, van der Waals surface, carbon skeleton, etc. Molecules can be loaded from a PDB file as well as several other common formats, such as a PQR file. In some cases, the molecular structure in PDB does not contain hydrogen atoms, and may even miss a fraction of the heavy atom coordinates. Furthermore, continuum modeling requires accurate and complete structural data as well as force field parameters such as atomic charges and radii. PDB2PQR [35] can provide the force field parameters and add the hydrogen atom and some heavy atoms missed in PDB. VCMM integrates the PDB2PQR package and enriches the database with some unusual amino acid types and ion species.

In the meshing module, mesh generation tools can be added into VCMM as plug-ins. Those meshing tools are used to generate surface and volume meshes for our computations and visualization. The octree and KD-tree are two widely adopted methods used for unstructured mesh visualization. However, because a biomolecular object has a highly irregular shape and boundary, the octree method wastes a lot of memory and the ordinary KD-tree method spends a lot of time on pretreatment. We develop a novel model-based KD-tree method by further taking into account the atomic position information, which helps unstructured mesh to be used in the area of molecular visualization. This method utilizes the relationship between the distribution of atoms and the distribution of grid nodes. The node distribution can be estimated before generating the mesh, so that the space can be logically divided into sub-areas with a balance of nodes. This approach can speed up data extraction and rendering. This method needs to be further improved to speed up large molecule modeling. A mesh analysis tool is used to treat region marks and analyze the mesh quality. Meshes can be divided into many regions and interfaces, which can help to distinguish between different domains and boundary conditions. Those region marks are useful for problems with multi-domain, complex interface/boundary conditions and multi-physics. The mesh quality is critical for FE/BE methods to achieve

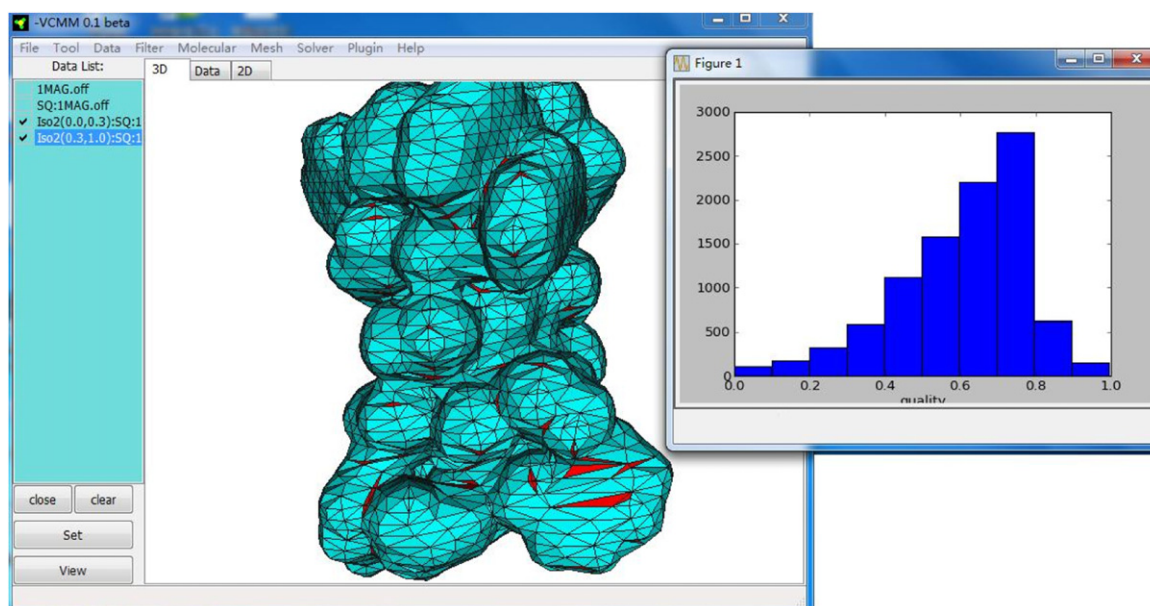


Figure 5. Triangular surface mesh of Gramicidin A protein is shown in the left panel. The poor quality triangles with small edge–edge ratio are marked with the red color. The right histogram shows the distribution of the edge–edge ratio for triangular surface meshes.

converged and reasonable results. VCMM provides a variety of indicators to measure the mesh quality, and marks a mesh with red color to indicate its poor quality. Those indicators are edge–edge ratio, aspect ratio, the maximum angle, etc. The edge–edge ratio is an element's shortest edge length divided by its longest edge length. The aspect ratio is an element's longest edge length divided by the diameter of its inscribed sphere. A triangle or a tetrahedron with a larger edge–edge ratio indicates a higher quality than one with a smaller edge–edge ratio. A triangle or a tetrahedron with a smaller aspect ratio indicates a higher quality than one with a larger aspect ratio. A triangle or a tetrahedron with a smaller maximum angle indicates a higher quality than one with a larger maximum angle. VCMM provides histograms for mesh quality analysis, which is shown in figure 5.

In the numerical module, VCMM has an interface for solvers and provides most of the popular visualization methods for the numerical results. The numerical solutions are usually stored in 3D format. Because it is impossible to observe directly the data inside a 3D model, VCMM visualizes the outside surface of the data set by default. VCMM adopts two methods to observe an inside volume data: sub-region rendering and volume rendering. Rendering a sub-region requires extracting data. VCMM supports two methods for data extraction. The first method is based on the spatial location of data, including line, plane, spherical or user-defined surface in space. The data can be interpolated on the aforementioned regions for users to observe conveniently the internal structure. The second method is based on the range of data values, including contour lines, contour noodles, etc. In the future, VCMM will provide more physically meaningful ways for data extraction for rendering a sub-region, such as extracting data near some special amino acids.

In scientific visualization, the volume ray casting technique is the most popular and it can be derived directly from the rendering equation. Users can control the resulting image by setting

different ranges with different color and transparency. This algorithm takes advantage of all the data, so the rendering image has more precise details. However, when the user changes the perspective of the image, a large amount of computations are required. VCMM uses a volume ray casting algorithm and provides a high quality image. However, volume ray casting is too slow for a usual computer to display an unstructured mesh. Therefore, VCMM develops an adaptive algorithm based on the isosurface to describe an entire 3D data set. This algorithm is used to select a group of isosurfaces, and then set those isosurfaces to different transparencies. The principle of isosurface selection is to retain the special information of the original data, such as extreme points and saddle points, so that the user can obtain the main features of the data distribution. Since this method reduces the amount of data to speed up the rendering, users can easily change the perspective in real time observation. However, as a result of the data reduction, the rendering image ignores some precise details.

The three modules can work with each other to enable a complete visual processing of the molecular simulation. VCMM is written in Python and the visualization core is based on an open-source VTK [43] developed by Kitware. VCMM runs on various systems including Microsoft Windows and Linux.

3. Example of usage

In this section, we will show the complete procedure for using the platform to simulate an ion channel system. The work flow is the same as is shown in figure 2. First, a PQR file is converted into a PQR file using the PDB2PQR package embedded in VCMM. Then, the PQR file is set as an input file to the meshing tool chain for surface and volume mesh generation. After obtaining the mesh, the user can send the mesh file and PQR file to the simulator, *ichannel* [26], to solve the PNP equations. If the user has given certain ion concentrations and applied voltages as the boundary conditions, then the simulator can automatically solve the PNP equations to obtain the electrostatic and concentrations which are stored in a VTK file. Finally, the VTK, PQR, and PDB files can be loaded into VCMM for analysis and visualization.

Two channel proteins, α -hemolysin (α -HL) and connexin 26 (Cx26) channel, are selected to compute the electrostatic potential, ion concentrations and I–V curve. α -HL is a bacterial exotoxin protein involved in many diseases including urinary infection in the 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 heptameric transmembrane channel with a relatively wide pore and $a + 7e$ net charge in the host cell membrane. The main constriction, which is the narrowest part of the pore, is formed by Met113 and a positively charged Lys147. The initial coordinates for α -HL are also obtained from the PDB (code 7AHL) [44]. The PQR file of α -HL contains 32 305 atoms. The protein pore is aligned with the z axis. The membrane layer is represented as a slab. The whole domain of the α -HL channel consists of the membrane protein region, the bulk region and the channel region. The ball-and-stick model of the α -HL channel is illustrated in figure 6.

A triangular surface mesh and a tetrahedral volume mesh of the α -HL system are generated via the tool chain in our platform. The membrane region is extracted and the involved tetrahedron and boundary faces are properly marked. Figure 7 shows an example of the unstructured tetrahedral volume mesh and triangular surface mesh of the α -HL ion channel. The mesh over the whole domain has a total of 467 851 vertices and 2 963 250 tetrahedra.

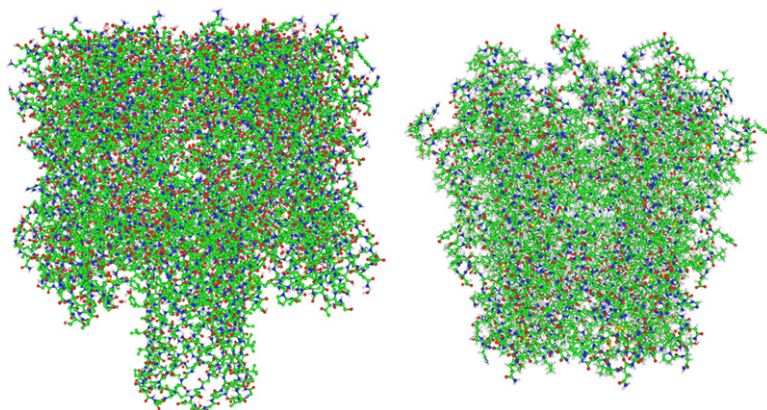


Figure 6. Cx26 hemichannel ball-and-stick model (right column) and α -HL ball-and-stick model (left column).

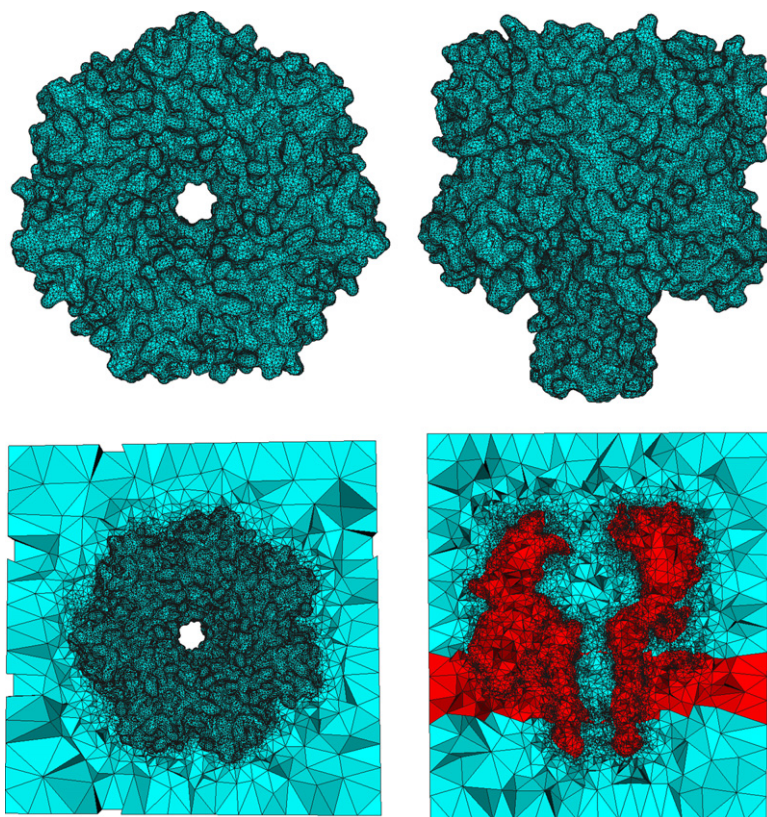


Figure 7. Triangular boundary mesh conforming to the α -HL ion channel surface: (a) top view; (b) lateral view; (c) boundary surface mesh of the ion channel with the membrane, which is represented as a slab; (d) view of the cross section of the whole tetrahedral volume mesh.

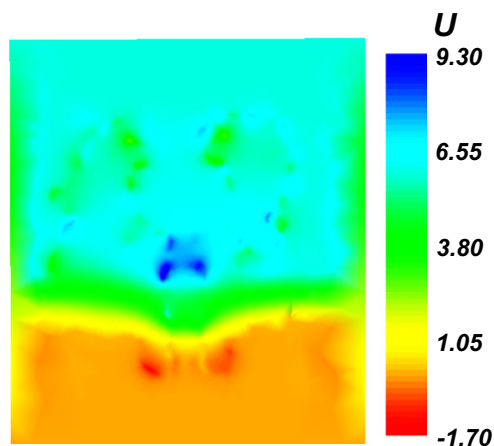


Figure 8. Electrostatic potential ($k_B T/e_c$): the figure is a cross section of the electrostatic potential of the whole domain of α -HL.

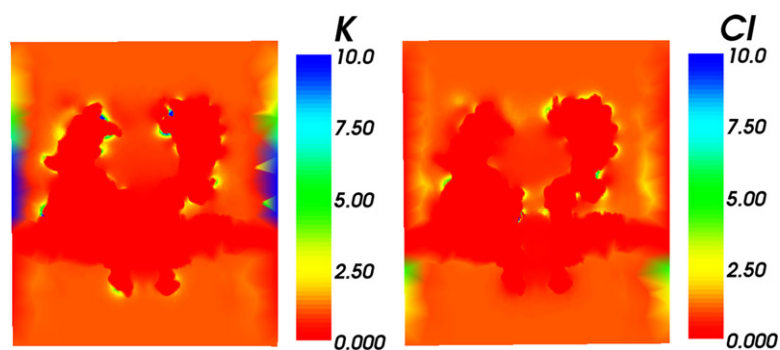


Figure 9. Ionic concentration (M): cross sections of the K^+ (left column) and Cl^- (right column) ion concentrations of the whole domain of α -HL.

The membrane and protein regions (red area in figure 3) are described by a low relative dielectric constant $\epsilon_m = 2$. A high relative dielectric constant $\epsilon_s = 80$ is assigned to the aqueous region, i.e. the volume outside of the protein-membrane region (blue region in figure 3). The diffusion coefficients for the cation and anion, e.g. K^+ and Cl^- , in the bulk region are set to their experimental values: $D_{Cl} = 0.203 \text{ \AA}^2 ps^{-1}$ and $D_K = 0.196 \text{ \AA}^2 ps^{-1}$. The voltage applied to the system, V_{applied} , is given by the potential difference along the z direction. The ion concentrations on the top and bottom side boundaries are set to their bulk values $c_{i,\text{bulk}}$.

We solve the coupled equations (3) and (1) to obtain the steady-state ion concentrations and electrostatic potential. For a given boundary condition ($V_{\text{applied}} = 100 \text{ mV}$ and $c_{i,\text{bulk}} = 1.0 \text{ M}$), a cross section of the potential and ion concentration of the whole domain is shown in figures 8 and 9. From figure 8, it can be seen that there is a high potential near the main constriction (blue region in figure 8), because the main constriction has a positively charged amino acid. From figure 9, it can be found that the concentration of Cl^- is higher than the concentration of K^+ near the main constriction.

Another channel is Cx26, which forms a typical four-helix bundle in which any pair of adjacent helices is antiparallel. The pore has an inner diameter of 35 \AA at the cytoplasmic entrance, and the smallest diameter of the pore is 14 \AA . The initial coordinates for the Cx26

hemichannel are obtained from the PDB (code GJB2) [45]. The ball-and-stick model of the Cx26 hemichannel is shown in figure 6.

To compare with the experimental data [46], the PNP equations are computed in the symmetric 120 mM KCl solution to obtain the conductance for the Cx26 hemichannel. The experimental average conductance for the Cx26 gap junction channel is $\sim 130\text{--}150$ pS. Our simulated average conductance for the Cx26 hemichannel is ~ 112 pS. With a single fitted parameter, the diffusion coefficient of which is chosen in the same manner as above, the simulation conductance is comparable with the experimental data. However, because the experimental measurement was on the complete Cx26 gap junction channel, further detailed simulations need to be activated once the structure of the complete channel is given.

4. Summary and discussion

In this paper, we present a software platform for the simulation of ion transport through ion channel systems, which contains three parts: a mesh generation tool chain, a parallel finite element solver and a visualization program. Due to the complexity of the molecular structure, it is a challenging task to generate a high quality molecular surface mesh and a tetrahedral volume mesh for the whole channel system. A tool chain is built in the platform for meshing the ion channel systems by using a few of mesh generation tools, including the surface meshing tool which we recently developed. Moreover, a parallel finite element solver is developed in the platform for PNP equations, the effectiveness of which is validated by some numerical tests [26, 47]. The popular molecular visualization tools often lack the capability of unstructured mesh management and visual analysis for the numerical results of unstructured meshes, and they do not provide functions for the boundary element method and FEM simulations. Finally, in the platform, we develop a visualization program for unstructured mesh visualization and numerical result analysis. The visualization software is the user interface of the platform, which allows for the control of the whole process of ion transport simulations by using the interfaces of the meshing tool chain and parallel finite element solver. The platform is applied to α -HL and the Cx26 channel to compute the electrostatic potential, ion concentrations and I–V curve.

Since it is still challenging to solve PNP and size-modified PNP equations for large ion channel systems, a more robust numerical method, a stabilized FEM, is under research for solving PNP equations for ion transport simulations and is to be integrated with this platform. Additionally, to visualize large ion channel systems, a model-based mesh segmentation method is to be further improved and added to our platform.

Some of the programs and tools of the platform (the latest versions of VCMM, meshing and other tools for continuum modeling) are available to download from our website www.continuummodel.org. We are also making an effort to release the other parts, especially the FEM solver, in the future. Our next step will be to merge them together by adding the meshing tool chain and FEM solver into VCMM for automatic simulations of ion transport through ion channel systems.

Documentation files and up-to-date information on those components can be obtained by accessing the same webpage.

Acknowledgments

B Tu, S Y Bai and B Z Lu are supported by the State Key Laboratory of Scientific/Engineering Computing, National Center for Mathematics and Interdisciplinary Sciences, the Chinese Academy of Sciences, the China NSF (91230106 and 11001257) and 863 program (2012AA020403). Y Xie and L B Zhang are supported by National 973 Project of China (2011CB309703), National 863 Project of China (2012AA01A309), China NSF (11171334 and 11321061), and National Center for Mathematics and Interdisciplinary Sciences of Chinese Academy of Sciences. M X Chen is supported by the China NSF (NSFC11301368) and the NSF of Jiangsu Province (BK20130278).

References

- [1] Zhang L B 2009 A parallel algorithm for adaptive local refinement of tetrahedral meshes using bisection *Numer. Math. Theor. Meth. Appl.* **2** 65–89
- [2] Marx D and Hutter J 2000 *Modern Methods and Algorithms of Quantum Chemistry* (Jülich: John von Neumann Institute for Computing)
- [3] Ostmeier J, Chakrapani S, Pan A C, Perozo E and Roux B 2013 Recovery from slow inactivation in K⁺ channels is controlled by water molecules *Nature* **501** 121–4
- [4] Jensen M, Jogini V, Borhani D W, Leffler A E, Dror R O and Shaw D E 2012 Mechanism of voltage gating in potassium channels *Science* **336** 229–33
- [5] Li S, Hoyles M, Kuyucak S and Chung S 1998 Brownian dynamics study of ion transport in the vestibule of membrane channels *Biophys. J.* **74** 37–47
- [6] Corry B, Kuyucak S and Chung S H 2000 Tests of continuum theories as models of ion channels ii. Poisson–Nernst–Planck theory versus Brownian dynamics *Biophys. J.* **78** 2364–81
- [7] Kuyucak S, Andersen O S and Chung S H 2001 Models of permeation in ion channels *Rep. Prog. Phys.* **64** 1427–72
- [8] Huber G A and McCammon J A 2010 Browndye: a software package for brownian dynamics *Comput. Phys. Commun.* **181** 1896–905
- [9] Lu B Z and Zhou Y C 2011 Poisson–Nernst–Planck equations for simulating biomolecular diffusion-reaction processes II: size effects on ionic distributions and diffusion-reaction rates *Biophys. J.* **100** 2475–2485
- [10] Lu B Z 2013 Poisson–Nernst–Planck equations *Encyclopedia of Applied and Computational Mathematics* ed B Engquist (Berlin: Springer)
- [11] Roux B, Allen T, Berneche S and Im W 2004 Theoretical and computational models of biological ionchannels *Q. Rev. Biophys.* **7** 1–103
- [12] Eisenberg R S 1998 Ionic channels in biological membranes: natural nanotubes *Acc. Chem. Res.* **31** 117–23
- [13] Chen D, Lear J and Eisenberg R S 1997 Permeation through an open channel: Poisson–Nernst–Planck theory of a synthetic ionic channel *Biophys. J.* **72** 97–116
- [14] Riveros O J, Croxton T L and Armstrong W M 1989 Theoretical and computational models of biological ionchannels *J. Theor. Biol.* **140** 221–30
- [15] Markowich P A 1986 *The Stationary Semiconductor Device Equations* (Vienna: Springer)
- [16] Kurnikova M G, Coalson R D, Graf P and Nitzan A 1999 A lattice relaxation algorithm for three-dimensional Poisson–Nernst–Planck theory with application to ion transport through the gramicidin a channel *Biophys. J.* **76** 642–56
- [17] Zheng Q, Chen D and Wei G W 2011 Second-order Poisson–Nernst–Planck solver for ion transport *J. Comput. Phys.* **230** 5239–62
- [18] Lu B Z, Holst M J, McCammon J A and Zhou Y C 2010 Poisson–Nernst–Planck equations for simulating biomolecular diffusion-reaction processes I: finite element solutions *J. Comput. Phys.* **229** 6979–94

- [19] Lu B Z and McCammon J A 2008 Molecular surface-free continuum model for electrodiffusion processes *Chem. Phys. Lett.* **451** 282–6
- [20] Lu B Z, Zhou Y C, Huber G A, Bond S D, Holst M J and McCammon J A 2007 Electrodiffusion: a continuum modeling framework for biomolecular systems with realistic spatiotemporal resolution *J. Chem. Phys.* **127** 135102
- [21] Kekenus-Huskey P M, Gillette A, Hake J and McCammon J A 2012 Finite element estimation of protein-ligand association rates with post-encounter effects: applications to calcium binding in troponin C and SERCA *Comp. Sci. Disc.* **5** 014015
- [22] Hollerbach U, Chen D P and Eisenberg R S 2002 Two- and three-dimensional Poisson–Nernst–Planck simulations of current flow through Gramicidin A *J. Sci. Comput.* **16** 373–409
- [23] Mathur S R and Murthy J Y 2009 *SIAM J. Appl. Math.* **52** 4031–9
- [24] Kurnikov I and Kurnikova M 2013 Harlem: a multipurpose interactive molecular modeling package (accessed 10 Dec, 2013) (http://crete.chem.cmu.edu/~igor/harlem_main.html)
- [25] Rafferty C S and Smith R K 1996 Solving partial differential equations with the prophet simulator *Lucent Technol. Memo*
- [26] Tu B, Chen M, Xie Y, Zhang L, Eisenberg B and Lu B 2013 A parallel finite element simulator for ion transport through three-dimensional ion channel systems *J. Comput. Chem.* **34** 2065–78
- [27] Chen M X and Lu B Z 2011 TMSmesh: a robust method for molecular surface mesh generation using a trace technique *J. Chem. Theory Comput.* **7** 203–12
- [28] Chen M X, Tu B and Lu B Z 2012 Manifold meshing method preserving molecular surface topology *J. Mol. Graph. Modelling* **38** 411–8
- [29] Tu B, Xie Y, Zhang L and Lu B A stablized finite element method for ion channel simulations (in preparation)
- [30] DeLano W L 2002 The PyMOL molecular graphics system
- [31] Humphrey W, Dalke A and Schulten K 1996 VMD: visual molecular dynamics *J. Mol. Graph.* **14** 33–38
- [32] Nicholls A, Sharp K A and Honig B 1991 Protein folding and association: insights from the interfacial and thermodynamic properties of hydrocarbons *Proteins: Struct. Funct. Bioinf.* **11** 281–96
- [33] Henderson A 2007 *ParaView guide, a parallel visualization application* (New York: Kitware)
- [34] Si H Tetgen: a quality tetrahedral mesh generator and a 3D delaunay triangulator (accessed 10 May, 2010) (<http://tetgen.berlios.de/>)
- [35] Dolinsky T J, Nielsen J E, McCammon J A and Baker N A 2004 PDB2PQR: an automated pipeline for the setup, execution, and analysis of Poisson–Boltzmann electrostatics calculations *Nucleic Acids Res.* **32** 665–7
- [36] Simakov N A and Kurnikova M G 2010 Soft wall ion channel in continuum representation with application to modeling ion currents in α -hemolysin *J. Phys. Chem. B* **114** 15180–90
- [37] Eisenberg R S, Hyon Y Y and Liu C 2010 Energy variational analysis envara of ions in water and channels: field theory for primitive models of complex ionic fluids *J. Chem. Phys.* **133** 104104
- [38] Liao T, Zhang J, Kekenus-Huskey P M, Cheng Y H, Michailova A, McCulloch A D, Holst M and McCammon J A 2013 Multi-core cpu or gpu-accelerated multiscale modeling for biomolecular complexes *Mol. Based Math. Biol.* **1** 164–79
- [39] Zaharescu A, Boyer E and Horaud R P 2007 Transformesh: a topology-adaptive mesh-based approach to surface evolution *Proc. of the Eighth Asian Conf. on Computer Vision II* pp 166–75
- [40] Bai S Y and Lu B VCOMM: a visual tool for continuum molecular modeling submitted
- [41] Berman H M, Westbrook J, Feng Z K, Gilliland G, Bhat T N, Weissig H, Shindyalov I N and Bourne P E 2000 The protein data bank *Nucleic Acids Res.* **28** 235–42
- [42] Frey P J 2010 Medit: an interactive mesh visualization software (accessed 20 Dec, 2010) (<http://www.ann.jussieu.fr/frey/logiciels/medit.html>)
- [43] Schroeder W, Martin K and Lorensen B 2006 The visualization toolkit: an object-oriented approach to 3D graphics (Clifton Park, NY: Kitware)

- [44] Song L, Hobauch M R, Shustak C, Cheley S, Bayley H and Gouaux J E 1996 Structure of staphylococcal α -hemolysin, a heptameric transmembrane pore *Science* **274** 1859–66
- [45] Maeda S, Nakagawa S, Suga M, Yamashita E, Oshima A, Fujiyoshi Y and Tsukihara T 2009 Structure of the connexin 26 gap junction channel at 3.5 Å resolution *Nature* **458** 597–602
- [46] Suchyna T M, Nitsche J M, Chilton M, Harris A L, Veenstra R D and Nicholson B J 1999 Different ionic selectivities for connexins 26 and 32 produce rectifying gap junction channels *Biophys. J.* **77** 2968–87
- [47] Xie Y, Cheng J, Lu B and Zhang L 2013 Parallel adaptive finite element algorithms for solving the coupled electro-diffusion equations *Mol. Based Math. Biol.* **1** 90–108