Journal of Computational Physics 545 (2026) 114500

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An unfitted finite element method for the Poisson-Nernst-Planck equations avoiding mesh generation for membrane-channel protein systems

Ziyang Liu [a,b], Sheng Gui [a,b], Benzhuo Lu [a,b,*], Linbo Zhang [a,b,*]

[a] SKLMS, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190, Beijing, China
[b] School of Mathematical Sciences, University of Chinese Academy of Sciences, 100049, Beijing, China

ARTICLE INFO

Keywords:
Poisson-Nernst-Planck equations
Ion channel
Interface-penalty finite element method
Unfitted mesh
Flood fill algorithm

ABSTRACT

The Poisson-Nernst-Planck (PNP) model, which characterizes the electro-diffusion process, is widely used in ion channel simulations. However, generating qualified body-fitted meshes for membrane-channel protein systems and applying them to numerical methods remain a significant challenge. The existing mesh generation methods, such as those used to construct tetrahedral meshes for the ion channel protein, membrane, and ionic solvent regions, as well as interface-fitted irregular meshes for the simulation box, are highly complex and technically demanding. This difficulty in mesh generation has hindered the further development and application of numerical methods in this field. Currently, the only available mesh generation software for ion channels has been developed by our group. In this work, we propose a significant breakthrough in computational methods by introducing an interface-penalty finite element method (IPFEM), a typical unfitted finite element method, to solve the PNP equations avoiding mesh generation for membrane-channel protein systems. Our approach significantly mitigates the challenges and laborious tasks associated with mesh generation for membrane-channel protein systems. The Flood Fill Algorithm (FFA) is employed to effectively identify the ion channel protein, membrane, and ionic solvent regions. In this work we show for the first time some examples to handle multipore systems using FFA, such as a membrane with an embedded ion channel protein tilted with respect to the z-axis, a membrane containing an ion channel protein with a two-branch pore, and a membrane with two embedded ion channel proteins. Numerical examples demonstrate the accuracy of our method on benchmark problems and illustrate its good performance in solving the PNP equations for both the Gramicidin A (gA) system and the voltage-dependent anion channel (VDAC) system.

1. Introduction

Electro-diffusion describes the diffusion process of charged particles in a self-induced electric field (sometimes along with an external electric field), a phenomenon that is prevalent in fields such as electrochemistry, biology, nanofluidics, and semiconductor physics. The dielectric continuum implicit solvent model, defined by the Poisson-Nernst-Planck (PNP) equations, is widely used to characterize the electro-diffusion process of ions in electrolyte solutions [1–4]. Compared to discrete models, such as the molecular

* Corresponding authors.
E-mail addresses: bzlu@lsec.cc.ac.cn (B. Lu), zlb@lsec.cc.ac.cn (L. Zhang).

https://doi.org/10.1016/j.jcp.2025.114500
Received 10 January 2025; Received in revised form 26 August 2025; Accepted 29 October 2025
Available online 4 November 2025

dynamics model [5] and the Brownian dynamics model [6], the PNP model offers higher computational efficiency and is more convenient for extracting macroscopic properties of certain biological systems, such as current-voltage characteristics and conductance rectification.

The molecular surface geometry of an ion channel, which is usually highly irregular, significantly affects the transport behavior of ions in the channel. In the last two decades, PNP ion channel models were solved numerically by using finite difference method (FDM) [7,8] and finite volume method (FVM) [9,10] in either a simplified one-dimensional or a complex three-dimensional framework. The finite element method (FEM) [11–16] is also a choice for discretization of the PNP model in ion channel simulations. However, the performance of the above methods in simulating three-dimensional ion channels is always hindered by poor mesh quality. The FDM is most suitable for regular, structured meshes, the FVM requires body-fitted Delauney meshes, while the traditional FEM usually works with body fitted tetrahedral meshes.

In ion channel simulations, the construction of high-quality meshes continues to pose a significant challenge for the implementation of finite element methods (FEM), primarily due to the highly irregular shapes of membrane-channel protein systems. This task becomes even more complex as it involves accurately capturing the irregular molecular surfaces of the protein and membrane regions, as well as the solvent region, which has a complicated geometric structure influenced by the membrane. These mesh generation methods are highly complex and technically demanding, often requiring significant manual intervention. Currently, the only available mesh generation software [17] for membrane-channel protein systems is developed by our group, which consists of surface meshing, quality improvement, volume mesh generation, and membrane-protein mesh construction. The process begins by generating a manifold triangular mesh of the Gaussian surface of the channel protein molecule using our program TMSmesh [18–20]. In the subsequent step, the SMOPT [21] software package is used to optimize the protein surface triangular mesh while maintaining manifoldness. A tetrahedral volume mesh is then generated using TetGen [22] based on the surface mesh. One of the key challenges for meshing membrane-channel protein system is to correctly distinguish the pore and membrane regions, as they are interconnected after being generated by TetGen. A walk-and-detect algorithm [17] is employed to numerically detect the inner surface of the ion channel pore, enabling the creation of a body-fitted mesh for membrane-channel protein systems, which can be used for 3D finite element simulations. More details can be found in the literature [17]. To improve mesh quality and reduce CPU time, Xie's group has updated two versions of the mesh generation software package [23,24] based on TMSmesh and SMOPT. Although this toolchain has successfully generated meshes for some systems, the overall process is labor-intensive and involves substantial manual intervention [21,25,26].

Thus, we aim to find a new strategy that does not rely on high-quality body-fitted mesh for membrane-channel protein system while ensuring the accuracy of numerical results. The singularities of PNP, like those in Poisson-Boltzmann equation (PBE), can be addressed using a solution decomposition scheme to handle Dirac delta distributions and ensure numerical accuracy. That is, the electrostatic potential ϕ can be split into two components: $\phi = G + \phi^r$. G collects all singularities from the Dirac delta distributions with available analytical formula, ϕ^r is the solution of a typical elliptic interface problem. Over the past decades, various unfitted mesh methods, in which the interface is allowed to cross mesh elements, have been proposed for solving elliptic interface problems. Examples of such methods are the multiscale finite element methods [27], the immersed interface methods [28], the immersed finite element methods [29], the extended finite element methods [30] and the penalty finite element methods [31]. Unfitted mesh methods involving penalty terms can be traced back to the penalty finite element methods proposed by Babuška [32]. A. Hansbo and P. Hansbo [33] presented an unfitted finite element method, in which through the introduction of a geometry-dependent average of flux at the interface, they established a stable discretization and proved that the linear finite element scheme is nearly optimal in two dimensions. This approach has motivated many follow-up works, e.g., the cut finite element methods [34], the unfitted finite element methods [35], the unfitted discontinuous Galerkin methods [36]. Wu et al. [31] proposed an hp-unfitted discontinuous Galerkin method for solving the linear elliptic interface problem and Liu et al.[37] proved that, for the general interface problem in H^1, $\boldsymbol{H}(\mathbf{curl})$ and $\boldsymbol{H}(\mathrm{div})$, the method converges in broken H^1 norm at an optimal rate with respect to h and at a suboptimal rate with respect to p by a factor of p.

In our previous work, we applied interface-penalty finite element methods (IPFEM) to solve Poisson-Boltzmann equation without requiring molecular mesh generation [38]. Motivated by that, we propose IPFEM for the PNP equations in ion channel system. In this paper, the PNP equations are decoupled using the Gummel iterative method [39], and we present the IPFEM scheme for the Poisson equation, along with the corresponding numerical scheme for the Nernst-Planck equations. A significant contribution of our approach is the introduction of the Flood Fill Algorithm (FFA), a classic graph-based region-growing (filling) technique [40], to identify and mark the ion channel protein, membrane, and solvent regions. Using FFA we can effectively deal with complex protein systems which can not be handled with existing approaches, such as a membrane with an embedded ion channel protein tilted with respect to the z-axis, a membrane containing an ion channel protein with a two-branch pore, and a membrane with two embedded ion channel proteins. Our method significantly reduces the complexity associated with numerical simulation and mesh handling, thereby enhancing its applicability to more complex ion channel models. The accuracy of our method is validated through numerical examples on benchmark problems, and it exhibits good performance in solving the PNP equations in the examples, gramicidin A (gA) system and the voltage-dependent anion channel (VDAC) system.

The rest of the paper is organized as follows. In Section 2, we introduce the PNP model in ion channel system and its regularization form. Our method is presented in Section 3, including the strategy of handling the membrane-channel protein system, and the IPFEM for PNP equations in ion channel simulation. Benchmark tests are provided to show the accuracy of our method in Section 4, as well as the performance of our method on gA system and VDAC system, including comparisons with traditional FEM. In Section 5, we conclude our paper.

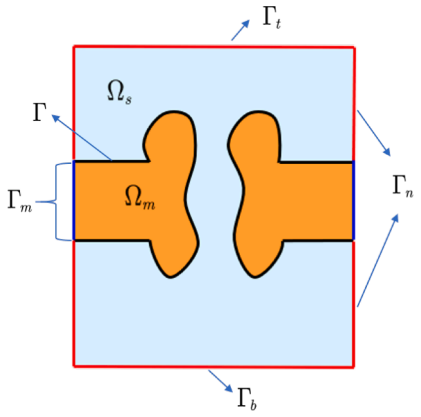

Fig. 1. A cut plane through the center of the simulation box along the z axis.

Table 1
Parameters of the PNP model in SI units.

Parameter	Name	Value	Unit
ϵ_0	Vacuum permittivity	$8.854187817\times 10^{-12}$	F/m
e_c	Elementary charge	$1.602176565\times 10^{-19}$	C
k_B	Boltzmann constant	$1.380648813\times 10^{-23}$	J/K
T	Absolute temperature	298.15	K

2. The Poisson-Nernst-Planck equations

In this work, the PNP model is used to describe the electrostatics, density distributions, and electro-diffusion processes of mobile ions in solvated biomolecular systems under non-equilibrium conditions. The computational region is an ion channel embedded in membrane proteins (see Fig. 1). We consider an open domain $\Omega \subset \mathbb{R}^3$, with $\overline{\Omega} = \overline{\Omega}_s \cup \overline{\Omega}_m$, $\Omega_s \cap \Omega_m = \emptyset$, where $\Omega_m \subset \Omega$ represents the solute region, i.e., the protein and membrane region, and $\Omega_s \subset \Omega$ represents solvent region, including the solvent reservoirs and the channel region. The Ω_m and Ω_s are separated by the molecular surface Γ. Γ_t and Γ_b are the top and bottom boundaries, respectively. We also let $\Gamma_m = \partial\Omega_m \backslash \Gamma$ and $\Gamma_n = \partial\Omega \backslash (\Gamma_t \cup \Gamma_b \cup \Gamma_m)$.

The steady-state PNP model couples the Nernst-Planck (NP) equations

$$\begin{cases} -\nabla\cdot\mathbf{J}_i = 0, & \text{in } \Omega_s,\ i = 1, 2, \dots, K, \\ \mathbf{J}_i = -D_i\left(\nabla c_i + \beta q_i \nabla\phi c_i\right), \end{cases} \tag{1}$$

and the electrostatic Poisson equation with the interface Γ

$$-\nabla\cdot\epsilon\epsilon_0\nabla\phi = \rho^f + \lambda\sum_{i=1}^{K} q_i c_i, \quad \text{in } \Omega. \tag{2}$$

where ϕ is the electrostatic potential, c_i is the concentration of the i-th ion species carrying charge $q_i = z_i e_c$, z_i is the valence of the i-th ion species and e_c is the elementary charge. K is the number of diffusive ion species, D_i denotes the spatial-dependent diffusion coefficient. The constant $\beta = 1/\left(k_B T\right)$ is the inverse Boltzmann energy with the Boltzmann constant k_B and the absolute temperature T. The characteristic function λ is equal to 0 in Ω_m and 1 in Ω_s, which implies that the solute region is impenetrable to mobile ions. The relative permittivity ϵ is defined as $\epsilon = \epsilon_m$ in Ω_m and $\epsilon = \epsilon_s$ in Ω_s, where ϵ_m (or ϵ_s) usually takes 2 (or 80) and ϵ_0 is the vacuum dielectric constant. In the permanent charge distribution

$$\rho^f = \sum_{j=1}^{N} q_j \delta\left(\mathbf{x} - \mathbf{x}_j\right), \tag{3}$$

$\delta\left(\mathbf{x} - \mathbf{x}_j\right)$ is the Dirac delta distribution at point $\mathbf{x}_j$, N is the number of particles with singular atomic charges q_j located at $\mathbf{x}_j$ inside biological membrane proteins. The values and units of physical parameters ϵ_0, e_c, k_B and T are listed in Table 1.

This PNP system is augmented with the following boundary conditions:

$$\begin{cases} \phi = V_m & \text{on } \Gamma_t, \quad \phi = 0, & \text{on } \Gamma_b, \\ c_i = c_{it}^{\text{bulk}} & \text{on } \Gamma_t, \quad c_i = c_{ib}^{\text{bulk}}, & \text{on } \Gamma_b, \\ \dfrac{\partial\phi}{\partial\mathbf{n}} = 0 & \text{on } \Gamma_n \cup \Gamma_m, \quad \mathbf{J}_i\cdot\mathbf{n} = 0, & \text{on } \Gamma \cup \Gamma_n, \end{cases} \tag{4}$$

where $\mathbf{n}$ is the unit normal vector at Γ (pointing from Ω_m to Ω_s) or Γ_N. There are two conditions on Γ needing to be satisfied from the dielectric theory:

$$[\![\phi]\!] = 0, \quad \left[\!\left[\epsilon \frac{\partial \phi}{\partial \mathbf{n}}\right]\!\right] = 0, \quad \text{on } \Gamma,$$

where the jump $[\![v]\!]$ of v on Γ is defined as $[\![v]\!] = v|_{\Omega_m} - v|_{\Omega_s}$.

2.1. Regularization

The permanent charge distribution ρ^f in (3) implies that the electrostatic potential ϕ is not with full regularity. A regularization scheme [11,41] can remove the singular component of the potential such that the remaining component can be solved. The potential is decomposed into a singular component G and a regular component ϕ_r,

$$\phi = \phi_r + G, \quad G(\mathbf{x}) = \frac{1}{4\pi\epsilon_m\epsilon_0} \sum_{j=1}^{N} \frac{q_j}{||\mathbf{x} - \mathbf{x}_j||} \quad \text{in } \Omega_m, \tag{5}$$

with $-\nabla \cdot (\epsilon_m \epsilon_0 \nabla G) = \sum_{j=1}^{N} q_j \delta(\mathbf{x} - \mathbf{x}_j)$.

For convenience, we let $u = e_c \beta \phi_r$, which is dimensionless. Then, we can get the regular component u from

$$\begin{cases} -\nabla \cdot (\epsilon \nabla u) = \lambda \dfrac{e_c^2 \beta}{\epsilon_0} \displaystyle\sum_{i=1}^{K} z_i c_i, & \text{in } \Omega, \\ [\![u]\!] = -e_c \beta G := g_{\mathrm{D}}, & \text{on } \Gamma, \\ \left[\!\left[\epsilon \dfrac{\partial u}{\partial \mathbf{n}}\right]\!\right] = e_c \beta \epsilon_m \dfrac{\partial G}{\partial \mathbf{n}} := g_{\mathrm{N}}, & \text{on } \Gamma. \end{cases} \tag{6}$$

Accordingly, the dimensionless NP equations are as follows,

$$\begin{cases} -\nabla \cdot \mathbf{J}_i = 0, & \text{in } \Omega_s, \ i = 1, 2, \ldots, K, \\ \mathbf{J}_i = -D_i \left(\nabla c_i + z_i \nabla u c_i\right). \end{cases} \tag{7}$$

3. Unfitted finite element method

In this section, we present the interface-penalty finite element method for solving the PNP equations in membrane-channel system. The interface-penalty finite element method belongs to the class of unfitted finite element methods, in which the finite element mesh is not required to be fitted with the interface (Γ in Fig. 1). In unfitted finite element methods, an element which intersects with the interface is called an interface element or cut element. In the case of a molecular surface, an interface element is divided into two parts by the molecular surface, one in Ω_m (the molecular and membrane part) and the other one in Ω_s (the solvent part). The basic idea of the IPFEM consists of using two sets of degrees of freedom in each interface element to define two finite element functions which approximate the solution in the solvent part and the molecular part respectively, and handling the interface conditions on the molecular surface by penalizing them in the finite element discretization. A formal and more detailed description of the IPFEM will be given in Section 3.2.

3.1. The computational mesh

Fig. 2 shows a cross-section of a membrane-channel protein system. The system consists of the channel protein, the membrane and the solvent region. The channel protein region and the membrane region constitute the solute region Ω_m. The membrane-protein is located in a solvent box which is separated into two parts. The up and down parts are connected by the open channel pore. The part of the solvent region between the two white dotted lines is the pore region. The presence of the membrane renders the geometries of the entire solute and solvent regions more complex.

In this subsection, we describe our algorithms and implementations for generating these meshes, which are based on the Flood Fill Algorithm (FFA). Fig. 4 illustrates the flow chart of FFA. First, we divide the computational domain into three parts, $\Omega = D_t \cup D_m \cup D_b$:

$$D_t = \{(x, y, z) \in \Omega \mid z > z_{\mathrm{m}}^t\}, \ D_m = \{(x, y, z) \in \Omega \mid z_{\mathrm{m}}^b \le z \le z_{\mathrm{m}}^t\}, \ D_b = \{(x, y, z) \in \Omega \mid z < z_{\mathrm{m}}^b\},$$

with the membrane of bottom surface at $z = z_{\mathrm{m}}^b$ and top surface at $z = z_{\mathrm{m}}^t$. Then, we use the unfitted mesh generation algorithm with interface (Gaussian molecular surface) resolving, which is proposed in our previous work [38]. After this step, we assign each element a mark: the elements entirely in the ion channel are marked as `Protein_Mark`, those intersecting with the molecular surface are marked as `Surface_Mark` and those not intersecting with the channel protein region are marked as `Solvent_Mark`. The computational mesh for the unfitted finite element method typically contains three types of elements: solution elements (entirely in Ω_m), solvent elements (entirely in Ω_s), and interface elements (intersecting the interface Γ). The task for exactly marking these elements is non trivial due to the irregular shape of the solvent region caused by the membrane-channel protein system.

One of the key challenges in traditional methods for meshing membrane-channel protein systems is correctly distinguishing between the pore and membrane regions. To this end, we introduce the Flood Fill Algorithm (FFA) [40], a region-filling technique

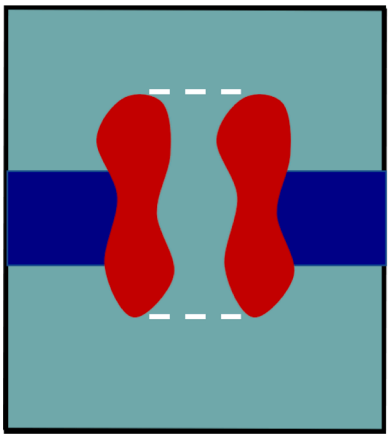

Fig. 2. 2D schematic picture for the cross section of a membrane-channel protein system. The solvent region is colored cyan, the channel protein is colored red and the membrane is colored blue. The solvent part between the white dotted lines is the pore region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

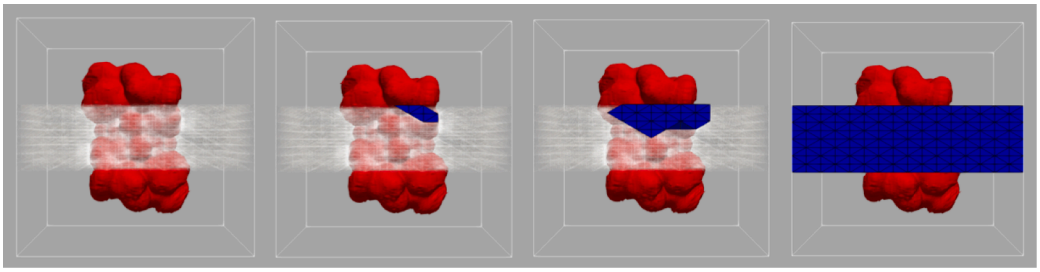

Fig. 3. An iterative search process for the elements that belong to the membrane region (in blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

based on the principle of connectivity, widely used in image processing and computer vision. The core idea of the FFA is to start from a specified initial point and iteratively expand, via breadth-first search or depth-first search, to fill adjacent regions satisfying certain conditions until encountering a boundary or obstacle. In this work, we adapt this concept by starting from a "seed" element and progressively identifying all elements in D_m that either are entirely contained within or intersect the membrane region, subsequently re-marking them as `Membrane_Mark`. Since the membrane region is a connected domain, we arbitrarily select one element e_s entirely contained in the membrane region (e.g., satisfying $e_s \in D_m$ and $\partial e_s \cap \partial\Omega \neq \emptyset$) as the seed element, and then recursively traverse the entire membrane region from e_s according to the following rules:

- do nothing if the element is not in D_m, has been traversed, or is marked as `Protein_Mark`;
- mark the element as `Membrane_Mark` and traverse all its neighborhood elements.

Algorithm 1 outlines the recursive neighbor traversal algorithm MARK_MEMBRANE, while Fig. 3 depicts the corresponding recursive traversal process.

Algorithm 1 Recursive neighbor traversal procedure.

```
procedure MARK_MEMBRANE(e_s)
    if e_s ∉ D_m or has been traversed or has Protein_Mark then
        return
    end if
    assign Membrane_Mark to e_s
    for all face neighbor e of e_s do
        MARK_MEMBRANE(e)
    end for
    return
end procedure
```

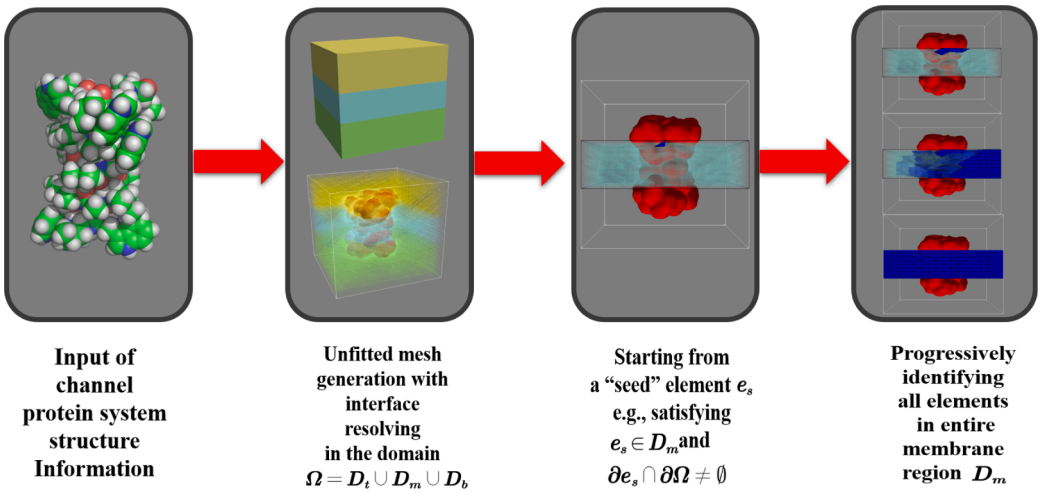

Fig. 4. Pipeline of computational mesh generation for unfitted finite element method of membrane-channel protein system.

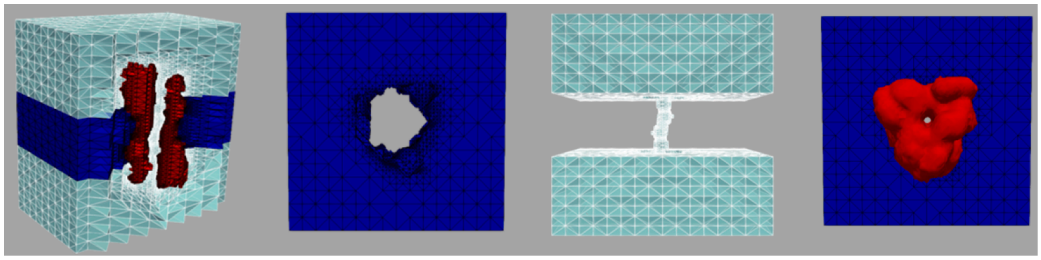

Fig. 5. The final computational mesh generated by FFA on gA system.

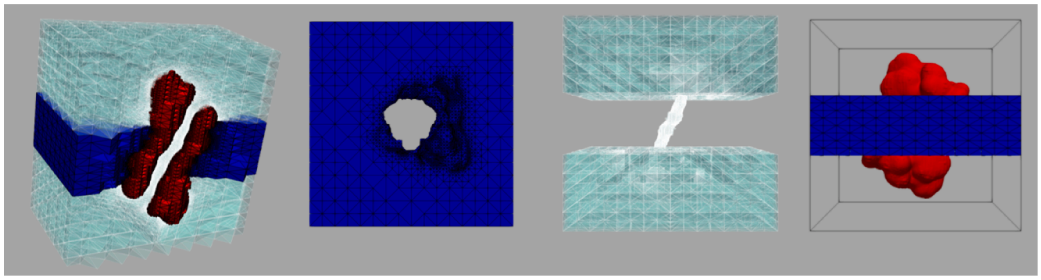

Fig. 6. The unfitted mesh generated by FFA on a membrane with an embedded ion channel protein tilted with respect to the z-axis.

At this stage, the elements marked as `Membrane_Mark` and `Protein_Mark` are designated as solute elements, those marked as `Surface_Mark` are designated as interface elements, and those marked as `Solvent_Mark` are designated as solvent elements, thereby yielding the final unfitted finite element computational mesh for the membrane-ion channel protein system. Fig. 4 illustrates the schematic diagram of the entire generation process.

In Fig. 5 the gA (PDB ID: 1MAG) system is used as an example to demonstrate the results of the proposed method, where the first subfigure is a computational mesh of the membrane-channel protein system, the second is a mesh of the membrane region, the third is a mesh of the solvent region, and the last is a mesh of the solute region consisting of the ion channel and the membrane region. Recently, Xie's group has been dedicated to developing mesh generation package [23,24] that yields smoother membrane region. It is noteworthy that the membrane region in the computational mesh generated by our method is notably smooth, a feature that was missed in our previous work. Notice FFA can also be applied to marking elements in body-fitted meshes for traditional FEM, to overcome some limitations of existing algorithms.

In order to test the robustness of our method, we have carried out further tests with examples which cause difficulties or can not be handled with the algorithms proposed in previous works. The first test is an embedded ion channel protein tilted with respect to the z-axis. Specifically, the gA channel is rotated by 20° around the x-axis to simulate this tilt and Fig. 6 demonstrates the unfitted mesh generated by FFA.

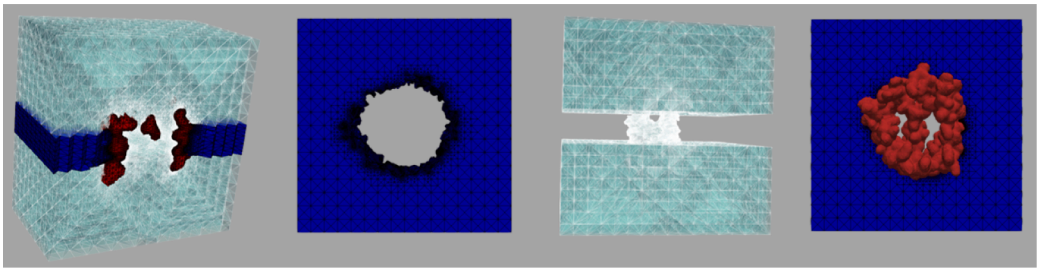

Fig. 7. The unfitted mesh generated by FFA on an ion channel molecular structure featuring a two-branch pore.

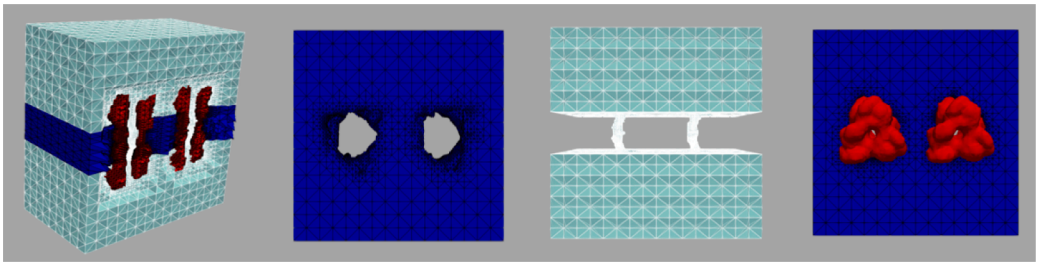

Fig. 8. The unfitted mesh generated by FFA on our two-channel proteins systems.

Next we conduct a test on an ion channel molecular structure featuring a two-branch pore, which is created by rotating the α-helix of VDAC by 80° along the z axis at the hinge region (Gly-21-Tyr-22-Gly-23-Phe-24-Gly-25), the results of the unfitted mesh generated by our method are presented in Fig. 7.

In the final test, we translate the coordinates of the gA channel in the x direction by 25 $\mathring{A}$ to obtain new coordinate data, resulting in a new PQR file. This PQR file is then merged with the initial coordinate information to generate a two-channel system and its corresponding PQR file. The two-channel PQR file is subsequently used as the input of the FFA, and Fig. 8 displays the results of the unfitted mesh generated by our method. To the best of our knowledge, we are the first to generate an unfitted mesh for a membrane with two embedded ion channel proteins, showing the robustness of FFA.

3.2. The interface-penalty finite element method (IPFEM)

This section introduces the interface-penalty finite element method for the PNP equations. We decouple the nonlinear system (6) and (7) by Gummel iteration [39] with relaxation technique. In each iteration, the Poisson equation and each NP equation are solved successively, with the ionic concentrations treated as known functions when solving the electrostatic potential, and vice versa. The process is repeated until the change of the solutions in two adjacent iterations becomes smaller than a given tolerance.

In this paper, we adopt the standard notation for Sobolev spaces $W^{s,p}(\Omega)$. Specifically, when $p = 2$, we use $H^s(\Omega) = W^{s,2}(\Omega)$. To facilitate the derivation of the equations, we simplify some notations with, $\Omega_1 = \Omega_m, \Omega_2 = \Omega_s, \Gamma_1 = \Gamma_t, \Gamma_2 = \Gamma_b, \Gamma_3 = \Gamma_m, \Gamma_4 = \Gamma_n$. Let $\{\mathcal{T}_h\}$ be a family of conforming and shape-regular partitions of the domain Ω into closed tetrahedra. For any $T \in \mathcal{T}_h$, we define $h_T := \mathrm{diam}(T)$, $h := \max_{T\in\mathcal{T}_h} h_T$. $\mathcal{I}_h$ represents the partition induced by $\{\mathcal{T}_h\}$ of the interface (molecular surface) Γ,

$$\mathcal{I}_h = \{e : e = T \cap \Gamma, |e| > 0, T \in \mathcal{T}_h\},$$

where $|e|$ denotes the area of e. Let $T^e \in \mathcal{T}_h$ be one of the element(s) containing e. It is reasonable to assume that each $e \in \mathcal{I}_h$ belongs to one of the two cases:

- the interior of e is entirely included in the interior of T^e, and $T_1^e := T^e \cap \Omega_1, T_2^e := T^e \cap \Omega_2$,
- $e \in \partial T_1^e \cap \partial T_2^e$ for two tetrahedras $T_1^e, T_2^e \in \mathcal{T}_h$.

So for both cases, we have

$$e \in \partial T_1^e \cap \partial T_2^e, \quad T_j^e \subset \bar{\Omega}_j, \quad j = 1, 2.$$

In addition, we define the set of boundary surfaces $\mathcal{I}_k$,

$$\mathcal{I}_k = \{e : e = \partial T \cap \partial\Gamma_k, |e| > 0, T \in \mathcal{T}_h, k = 1, 2, 3, 4\}.$$

The energy space is defined as follows,

$$V := \left\{ v : v|_{\Omega_j} = v_j, \text{ where } v_j \in H^1(\Omega), v_j\Big|_T \in H^2(T), \forall T \in \mathcal{T}_h, j = 1, 2 \right\}, \tag{8}$$

and the continuous finite element space U_h^p,

$$U_h^p := \{v_h \in H^1(\Omega) : v_h|_T \in \mathcal{P}_p(T), \forall T \in \mathcal{T}_h\}, \tag{9}$$

where $\mathcal{P}_p(T)$ denote the set of polynomials with total degree less than or equal to p.

We define our interface-penalty finite element approximation space V_h^p as the space of piecewise continuous finite element functions:

$$V_h^p := \left\{v_h : v_h|_{\Omega_j} = v_{jh} \in U_h^p, j = 1, 2\right\}. \tag{10}$$

That is, for any $v_h \in V_h^p$, the restriction of v_h onto Ω_j is a continuous piecewise polynomial for $j = 1, 2$, respectively. Clearly, $V_h^p \subset V \subset L^2(\Omega)$ but $V_h^p \not\subset H^1(\Omega)$. In fact, the functions in V_h^p are allowed to be discontinuous across the interface Γ. We also define the average $\{v\}$ of v on the interface Γ as

$$\{v\} := \frac{v|_{\Omega_1} + v|_{\Omega_2}}{2}.$$

Testing (6) by any $v \in V$, using Green's formula and the identity $[\![uv]\!] = [\![u]\!]\{v\} + [\![v]\!]\{u\}$, we obtain

$$\sum_{j=1}^{2} \int_{\Omega_j} \epsilon \nabla u \cdot \nabla v - \sum_{e \in I_1 \cup I_2} \int_e \epsilon \frac{\partial u}{\partial \mathbf{n}} v - \sum_{e \in I_h} \int_e \left\{\epsilon \frac{\partial u}{\partial \mathbf{n}}\right\} [\![v]\!] = \frac{e_c^2 \beta}{\epsilon_0} \sum_{i=1}^{K} \int_{\Omega_2} z_i c_i v + \int_\Gamma g_N \{v\} - \int_{\Gamma_3} e_c \beta \epsilon_m \frac{\partial G}{\partial \mathbf{n}}. \tag{11}$$

Define

$$a_h^{0,\text{poisson}}(u, v) := \sum_{j=1}^{2} \int_{\Omega_j} \epsilon \nabla u \cdot \nabla v - \sum_{e \in I_1 \cup I_2} \int_e \epsilon \frac{\partial u}{\partial \mathbf{n}} v - \sum_{e \in I_h} \int_e \left\{\epsilon \frac{\partial u}{\partial \mathbf{n}}\right\} [\![v]\!],$$

and

$$f_h^{0,\text{poisson}}(v, \mathbf{c}) := \frac{e_c^2 \beta}{\epsilon_0} \sum_{i=1}^{K} \int_{\Omega_2} z_i c_i v + \int_\Gamma g_N \{v\} - \int_{\Gamma_3} e_c \beta \epsilon_m \frac{\partial G}{\partial \mathbf{n}},$$

where $\mathbf{c} = (c_1, \ldots, c_K)$. We refer to the symmetric interface penalty finite element method in the literature [31] and define the bilinear form $a_h^{\text{poisson}}(\cdot, \cdot)$ on $V \times V$:

$$a_h^{\text{poisson}}(u, v) := a_h^{0,\text{poisson}}(u, v) - S(u, v) + J_0(u, v) + J_1(u, v), \tag{12}$$

where

$$S(u, v) := \sum_{e \in I_1 \cup I_2} \int_e \epsilon \frac{\partial v}{\partial \mathbf{n}} u + \sum_{e \in I_h} \int_e [\![u]\!] \left\{\epsilon \frac{\partial v}{\partial \mathbf{n}}\right\},$$

$$J_0(u, v) := \sum_{e \in I_1 \cup I_2} \frac{\gamma_0 p^2}{h_{T^e}} \int_e uv + \sum_{e \in I_h} \frac{\gamma_0 p^2}{h_{T^e}} \int_e [\![u]\!][\![v]\!],$$

$$J_1(u, v) := \sum_{e \in I_3 \cup I_4} \frac{\gamma_1 h_{T^e}}{p^2} \int_e \left(\epsilon \frac{\partial u}{\partial \mathbf{n}}\right)\left(\epsilon \frac{\partial v}{\partial \mathbf{n}}\right) + \sum_{e \in I_h} \frac{\gamma_1 h_{T^e}}{p^2} \int_e \left[\!\!\left[\epsilon \frac{\partial u}{\partial \mathbf{n}}\right]\!\!\right] \left[\!\!\left[\epsilon \frac{\partial v}{\partial \mathbf{n}}\right]\!\!\right],$$

and the linear form $f_h^{\text{poisson}}(\cdot)$ on V with the ionic concentrations $\mathbf{c}$ regarded as known functions:

$$f_h^{\text{poisson}}(v, \mathbf{c}) := f_h^{0,\text{poisson}}(v, \mathbf{c}) - S^f(v) + J_0^f(v) + J_1^f(v), \tag{13}$$

where

$$S^f(v) := \int_{\Gamma_1} \epsilon \frac{\partial v}{\partial \mathbf{n}}(e_c \beta V_m) + \int_\Gamma g_D \left\{\epsilon \frac{\partial v}{\partial \mathbf{n}}\right\},$$

$$J_0^f(v) := \sum_{e \in I_1} \frac{\gamma_0 p^2}{h_{T^e}} \int_e (e_c \beta V_m) v + \sum_{e \in I_h} \frac{\gamma_0 p^2}{h_{T^e}} \int_e g_D [\![v]\!],$$

$$J_1^f(v) := \sum_{e \in I_3} \frac{\gamma_1 h_{T^e}}{p^2} \int_e \left(-e_c \beta \epsilon_m \frac{\partial G}{\partial \mathbf{n}}\right)\left(\epsilon \frac{\partial v}{\partial \mathbf{n}}\right) + \sum_{e \in I_h} \frac{\gamma_1 h_{T^e}}{p^2} \int_e g_N \left[\!\!\left[\epsilon \frac{\partial v}{\partial \mathbf{n}}\right]\!\!\right].$$

In (12), $S(u, v)$ is to maintain the symmetry of the formulation, that is $a_h^{\text{poisson}}(u, v) = a_h^{\text{poisson}}(v, u)$, J_0 and J_1 are penalty terms, with penalization parameters γ_0 and γ_1 respectively, for enforcing continuity of the numerical solution across Γ. We refer to [31] for more detailed explanations about their roles. The parameters γ_0 and γ_1 are chosen to be large enough in order to ensure the convergence and accuracy of IPFEM, but we also found in numerical experiments that too large values of them can lead to difficulty or even failure in solving PNP. We usually choose $\gamma_0 = 10$ and $\gamma_1 = 1$ which perform well in most cases. These techniques in dealing with the discontinuities are from the interior penalty discontinuous or continuous Galerkin methods [42,43], with $S^f(v)$, $J_0^f(v)$ and $J_1^f(v)$ corresponding to $S(u, v)$, $J_0(u, v)$ and $J_1(u, v)$ respectively.

Notice that in (12) and (13), we need to calculate both volume integrals in complex curved regions (intersection of an element with the subdomains) and surface integrals on complex curved surfaces (patch of the Gaussian molecular surface within an element),

which is done using the open source library for high order numerical integration developed by our group [44], available in our finite element toolbox PHG, see http://lsec.cc.ac.cn/phg/index_en.htm and `doc/quad-XFEM.pdf` in the source code of PHG for related documentations.

Similarly, we define

$$V^{\mathrm{NP}} := \left\{ v : v \in H^1(\Omega_2) \right\}$$

and

$$V_h^{\mathrm{NP}} := \left\{ v_h : v_h \in V^{\mathrm{NP}},\ v_h\big|_{T\cap\Omega_2} \in \mathcal{P}_p(T\cap\Omega_2), \forall T \in \mathcal{T}_h \right\}.$$

Testing (7) by any $v \in V^{\mathrm{NP}}$ and using the same techniques, we can define the bilinear form $a_{i,h}^{\mathrm{NP}}(\cdot,\cdot)$ on $V^{\mathrm{NP}} \times V^{\mathrm{NP}}$ with the electrostatic potential u regarded as known function:

$$\begin{aligned} a_{i,h}^{\mathrm{NP}}(c,v,u) = &\int_{\Omega_2} \left(-D_i(\nabla c + z_i c \nabla u)\right)\cdot\nabla v - \int_{\Gamma_1\cup\Gamma_2} \left(-D_i(\nabla c + z_i c\nabla u)\right)\cdot \mathbf{n} v + \sum_{e\in\mathcal{I}_1\cup\mathcal{I}_2} \frac{\gamma_0 p^2}{h_{T^e}} \int_e cv \\ &+ \sum_{e\in\mathcal{I}_h\cup\mathcal{I}_4} \frac{\gamma_1 h_{T^e}}{p^2} \int_e \left(\left(-D_i(\nabla c + z_i c\nabla u)\right)\cdot\mathbf{n}\right)\left(\left(-D_i(\nabla v + z_i v\nabla u)\right)\cdot\mathbf{n}\right), \quad i = 1,\dots,K, \end{aligned} \tag{14}$$

and the linear form $f_h^{\mathrm{NP}}(\cdot)$ on V^{NP}:

$$f_{i,h}^{\mathrm{NP}}(v) = \sum_{e\in\mathcal{I}_1} \frac{\gamma_0 p^2}{h_{T^e}} \int_e c_{it}^{\mathrm{bulk}} v + \sum_{e\in\mathcal{I}_2} \frac{\gamma_0 p^2}{h_{T^e}} \int_e c_{ib}^{\mathrm{bulk}} v, \quad i = 1,\dots,K. \tag{15}$$

The interface-penalty finite element approximation to the Poisson Eq. (6) reads: find $u_h \in V_h^p$, such that

$$a_h^{\mathrm{poisson}}(u_h, v_h) = f_h^{\mathrm{poisson}}(v_h, \mathbf{c}) \quad \forall v_h \in V_h^p, \tag{16}$$

with the ionic concentrations $\mathbf{c}$ are treated as given functions, and the corresponding approximation to the NP Eq. (7) reads: find $c_{i,h} \in V_h^{\mathrm{NP}}$, such that

$$a_{i,h}^{\mathrm{NP}}(c_{i,h}, v_h, u) = f_{i,h}^{\mathrm{NP}}(v_h) \quad \forall v_h \in V_h^{\mathrm{NP}}, \quad i = 1,\dots,K. \tag{17}$$

with the electrostatic potential u is treated as given.

4. Numerical experiments

In this section, we carry out numerical experiments to test the accuracy and robustness of our IPFEM. For simplicity, we set $\epsilon_m = 2.0$, $\epsilon_s = 80.0$, $T = 298.15$ K, and unless otherwise specified, $c_{it}^{\mathrm{bulk}} = c_{ib}^{\mathrm{bulk}} = c_{i,\mathrm{bulk}} = 0.1$ M, $\gamma_0 = 10.0$, $\gamma_1 = 1.0$.

Our method is implemented using the finite element toolbox Parallel Hierarchical Grid (PHG) [45], in which the integrals in the cut elements are computed using the corresponding numerical quadrature functions [44] provided in PHG. All numerical experiments were done on the high performance computer of State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences, which consists of dual Intel Gold 6140 CPU nodes (2×18 cores, 2.30 GHz) and 100 Gbps EDR Infiniband network.

4.1. Benchmark problem

A simple test (a sphere in a cube) is carried out in this subsection to check the accuracy of the IPFEM for PNP model. We assume that the solute region is an atom of radius R Å. Taking the center of the sphere as the coordinate origin, a test analytical solution for the electrostatic potential is defined as follows.

$$\begin{cases} u_m(x,y,z) = x^2 + y^2 + z, & \text{in } \Omega_m, \\ u_s(x,y,z) = \tau_2 \cos(\pi x)\cos(\pi y)\cos(\pi z) + e_c\beta\tau_3 z, & \text{in } \Omega_s, \\ c_1(x,y,z) = \tau_1 + \frac{\tau_1}{2}\cos(\pi x)\cos(\pi y)\cos(\pi z), & \text{in } \Omega_s, \\ c_2(x,y,z) = \tau_1 - \frac{\tau_1}{2}\cos(\pi x)\cos(\pi y)\cos(\pi z), & \text{in } \Omega_s, \end{cases} \tag{18}$$

where $\tau_1 = 0.1$M, $\tau_3 = 0.1$V and τ_2 can be obtained by (6). Substituting (18) into (7), the source term and the corresponding boundary conditions can be calculated accordingly.

Here, we select the calculation region as a cube $\Omega = [-2\ \text{Å}, 2\ \text{Å}]^3$ and the solute region as a sphere $\{x^2 + y^2 + z^2 \le 1\}$.

The results are displayed in Table 2 with Ω uniformly divided into several tetrahedrons as the initial mesh for the calculation (see Fig. 9). The results are displayed in Table 2, which shows that that the relative errors decrease as times of refinement increase, verifying the accuracy of IPFEM to the three-dimensional PNP equations. The first column represents the mesh size relative to the initial mesh, and u, c_1, c_2 represent the analytical solutions while u_h, $c_{1,h}$, $c_{2,h}$ represent the numerical solutions of IPFEM. The numerical errors in L^2 norm are second-order accurate, respectively.

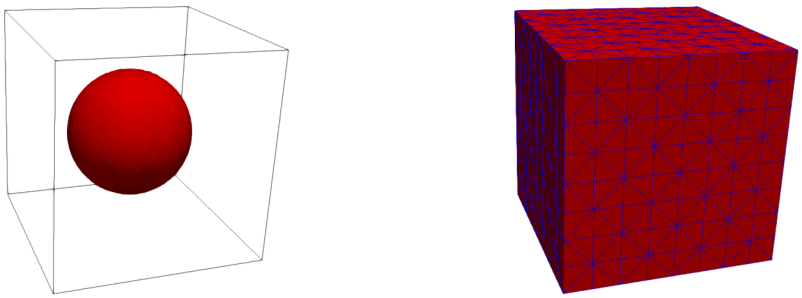

Fig. 9. Calculation area for benchmark problem.

Table 2
The L^2 norms of the relative errors of the IPFEM.

h	$\|\|u_h - u\|\|_{L^2}/\|\|u\|\|_{L^2}$	order	$\|\|c_{1,h} - c_1\|\|_{L^2}/\|\|c_1\|\|_{L^2}$	order	$\|\|c_{2,h} - c_2\|\|_{L^2}/\|\|c_2\|\|_{L^2}$	order
1	3.28E-3	-	9.73E-2	-	9.73E-2	-
1/2	9.59E-4	1.77	5.33E-2	0.86	5.36E-2	0.86
1/4	2.40E-4	1.99	2.29E-2	1.21	2.28E-2	1.21
1/8	5.76E-5	2.05	7.41E-3	1.62	7.40E-3	1.62
1/16	1.58E-5	1.86	2.00E-3	1.88	2.01E-3	1.88
1/32	3.94E-6	2.00	5.10E-4	1.97	5.09E-4	1.97

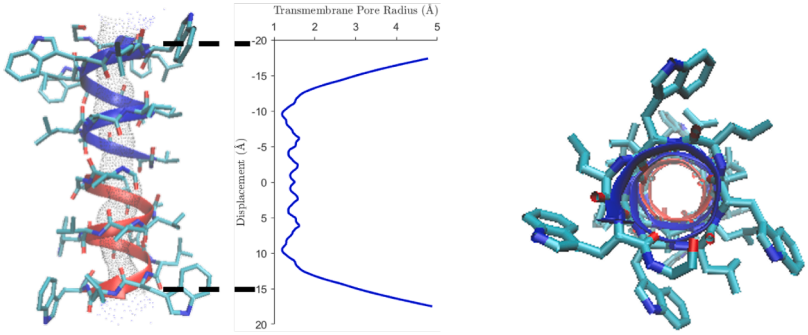

Fig. 10. Molecular structure of 1MAG (sticks for the molecular structure and cartoons for the two helical subunits).

4.2. Simulations of membrane-channel protein systems

4.2.1. Gramicidin A system

The gA channel (PDBID: 1MAG) is widely recognized as one of the most thoroughly studied ion channels, forming aqueous pores in lipid bilayers that allow selective passage of monovalent cations like Na^+ and K^+ [46]. The gA peptide consists of a small, 15-amino acid β-helical structure with a narrow pore. Its compact size and extensive experimental characterization have made it a popular subject for numerous theoretical models. Fig. 10 presents two views of its molecular structure. In this study, we use the PNP equations to compute the current as a function of the voltage applied across the channel. Initial coordinates for 1MAG were downloaded from the PDB website (https://www.rcsb.org/) and then converted to PQR files by the tool PDB2PQR [47].

We suppose that there are two kinds of ions in the solvent region, K^+ and Cl^-, and the diffusion coefficients for them in the bulk region are set to their experimental values: $D_{\mathrm{Cl}} = 0.203\mathring{A}^2/\mathrm{ps}$, $D_{\mathrm{K}} = 0.196\mathring{A}^2/\mathrm{ps}$. Although experimental measurements for the diffusion coefficients inside the channel are unavailable, it is established that the diffusion coefficients differ between the bulk region and the channel pore region, especially for narrow pores [48]. With a diameter of approximately 4 $\mathring{A}$, gA is a narrow ion channel. In this study, we present a case where the diffusion coefficients of ions vary continuously inside the channel. The diffusion coefficient function is as follows [49]:

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

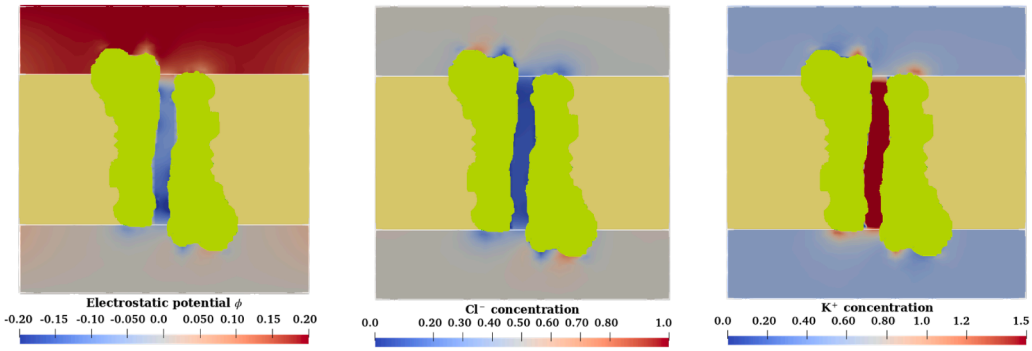

Fig. 11. The electrostatic potential ϕ (V) and the concentrations of K^+ and Cl^- ions (M) in color mapping on a cross section ($x = 0$) of the solvent region for 1MAG. Here the protein and membrane regions are colored in green and yellow, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

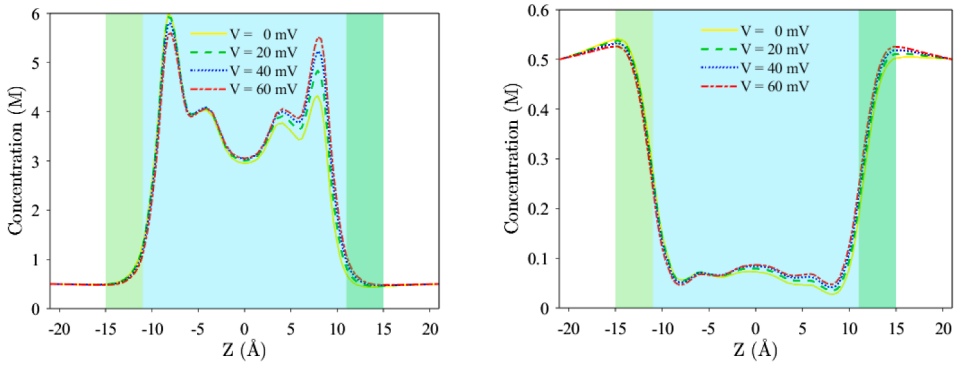

Fig. 12. K^+ (left) and Cl^- (right) concentrations at the center of 1MAG plotted along z-axis obtained from the PNP calculation with $c_{i,\text{bulk}} = 0.5$ M.

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

$$f(\mathbf{r}) = f(z) = \begin{cases} n\left(\frac{z - z^t_{\text{chan}}}{z^t_{\text{bulk}} - z^t_{\text{chan}}}\right)^{n+1} - (n+1)\left(\frac{z - z^t_{\text{chan}}}{z^t_{\text{bulk}} - z^t_{\text{chan}}}\right)^{n}, & z \in [z^t_{\text{chan}}, z^t_{\text{bulk}}] \\ n\left(\frac{z - z^b_{\text{chan}}}{z^b_{\text{bulk}} - z^b_{\text{chan}}}\right)^{n+1} - (n+1)\left(\frac{z - z^b_{\text{chan}}}{z^b_{\text{bulk}} - z^b_{\text{chan}}}\right)^{n}, & z \in [z^b_{\text{chan}}, z^b_{\text{bulk}}] \end{cases}$$

where n is an integer and we set $n = 9$ in our computations. $z^b_{\text{chan}} = -11$, $z^t_{\text{chan}} = 11$ are the boundary values of channel region on z axis and $z^b_{\text{bulk}} = -13$, $z^t_{\text{bulk}} = 13$ are the boundary values of bulk region. We set $D_{\text{chan}} = \frac{1}{18} D_{\text{bulk}}$ and this profile for the diffusion coefficients ensures that $D(r)$ is differentiable in the Nernst-Planck equation.

We solve the Eqs. (6) and (7) to obtain the steady-state ion concentrations and electrostatic potential. Fig. 11 displays a cross section view of the potential and ion concentration of the whole domain region, under the given boundary condition ($V_m = 0.2$ V and $c_{i,\text{bulk}} = 0.2$ M). It can be seen that the concentration of K^+ is higher than that of Cl^- in the pore.

In order to clearly demonstrate that gA system allows selective passage of monovalent cations, Fig. 12 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 tendency.

The PNP equations are computed for a variety of voltages and concentrations to get $I - V$ curves which are compared with the experimental data. The current across the pore can be calculated as

$$I_z = -\sum_i q_i \int_S D_i \left(\frac{\partial c_i}{\partial z} + \frac{q_i}{k_B T} c_i \frac{\partial \phi}{\partial z}\right) \mathrm{d}x\mathrm{d}y, \tag{19}$$

where S is a cut plane inside the ion channel. Due to the conservation of the total flux, S can be any cut plane located either inside or outside of the ion channel. For convenience, we take S as the top surface Γ_t of our computational region [50].

The experimental $I - V$ data are obtained from Cole et al. [51], which are used as the reference data. Comparisons between the simulation results and experimental data are shown in Fig. 13.

Table 3 shows the relative difference of the current between the simulation results and experimental data. It is seen that while some discrepancies are evident, the overall agreement between the two datasets remains satisfactory.

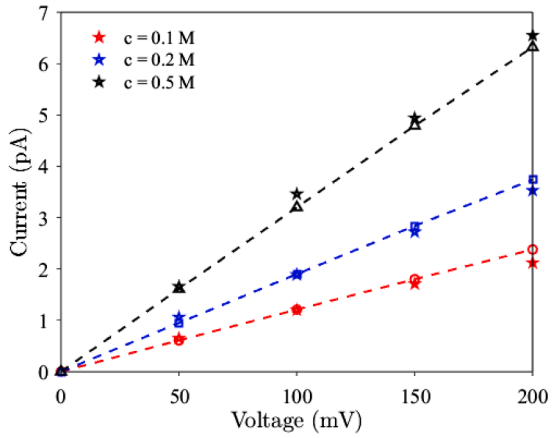

Fig. 13. Comparisons of the computed $I - V$ curves of 1MAG with experimental data (the dashed lines represent the simulation data and the stars represent the experimental data).

Table 3
The relative difference between the simulation results and experimental data.

Ion concentration (M)	Voltage (mV)	Experimental data (pA)	Simulation results (pA)	Relative difference
0.1	50	0.65	0.61	0.06
	100	1.2	1.21	0.01
	150	1.71	1.80	0.05
	200	2.12	2.38	0.12
0.2	50	1.06	0.95	0.10
	100	1.89	1.90	0.01
	150	2.72	2.84	0.04
	200	3.53	3.74	0.06
0.5	50	1.66	1.61	0.03
	100	3.46	3.20	0.08
	150	4.94	4.79	0.03
	200	6.55	6.32	0.04

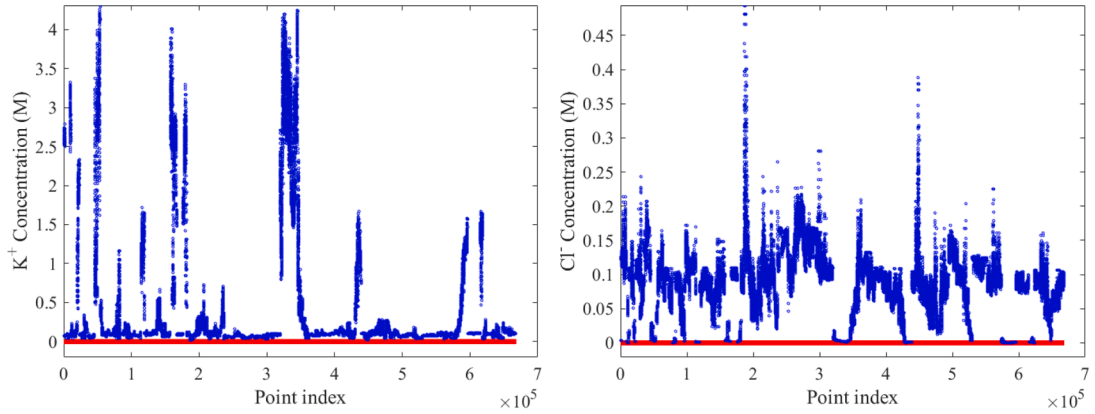

Fig. 14. Calculated distributions of ion concentrations in KCl electrolyte solution with $c_{i,\text{bulk}} = 0.1$ M. Left: K^+ concentration, the percentage of positive numerical solutions is 100 %; Right: Cl^- concentration, the percentage of positive numerical solutions is 100 %.

To check the positivity of ion concentrations in our gA simulation, we design a test case in which the applied voltage V_m is set to its maximum value, namely $V_m = 0.2$ V. The bulk concentration of the KCl electrolyte solution is chosen to be 0.1 M, 0.2 M, 0.3 M, respectively. The corresponding computed ion concentration distributions are shown in Figs. 14–16. The numerical results obtained in this test case indicate that the proposed scheme preserves the positivity of the numerical solution for this example under consideration. Although several recent studies have focused on the development of positivity-preserving algorithms [12,52–54], our work does not specifically address the construction of positivity-preserving schemes; this aspect is not the focus of our study. This example is intended to illustrate the reasonableness of the IPFEM results for solving the PNP equations.

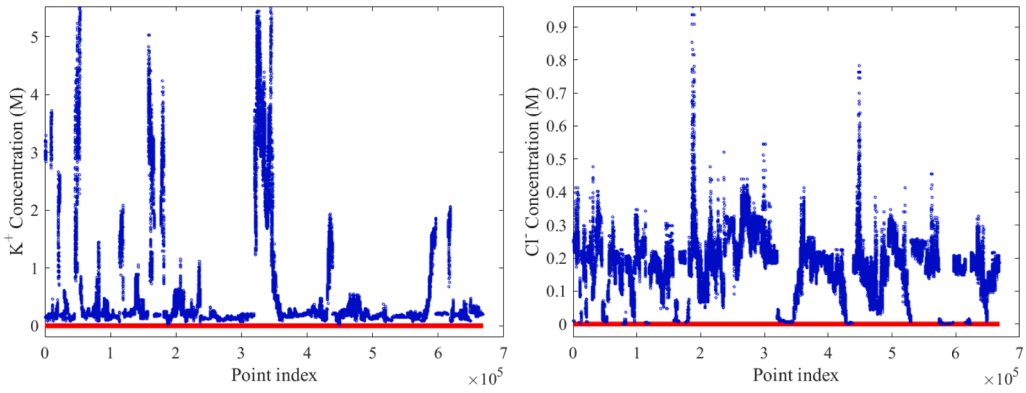

Fig. 15. Calculated distributions of ion concentrations in KCl electrolyte solution with $c_{i,\text{bulk}} = 0.2$ M. Left: K^+ concentration, the percentage of positive numerical solutions is 100 %; Right: Cl^- concentration, the percentage of positive numerical solutions is 100 %.

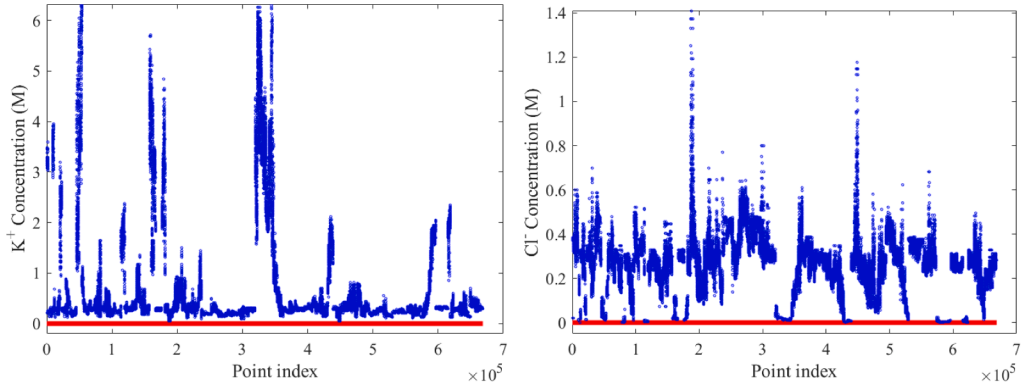

Fig. 16. Calculated distributions of ion concentrations in KCl electrolyte solution with $c_{i,\text{bulk}} = 0.3$ M. Left: K^+ concentration, the percentage of positive numerical solutions is 100 %; Right: Cl^- concentration, the percentage of positive numerical solutions is 100 %.

Table 4
The number of Gummel iterations for the gA system.

Voltage(mV)	0	20	40	60	80	100
Gummel Iteration	25	19	19	19	19	18
Voltage(mV)	120	140	160	180	200	-
Gummel Iteration	18	18	18	18	17	-

Table 4 shows the numbers of Gummel iterations for the gA system under $c_{i,\text{bulk}} = 0.1$ M at different applied voltages. It is observed that the Gummel iteration typically converges in approximately 17-25 iterations for the gA system, to a specified tolerance of 10^{-6} in relative change.

In order to compare performance of the IPFEM solver with traditional FEM solvers, we have simulated the case $V_m = 0$ V and $c_{i,\text{bulk}} = 0.1$ M using both our IPFEM solver and the conventional FEM solver from [13]. For comparable accuracy, the unfitted mesh for the IPFEM solver consisted of 9,504,338 tetrahedra and 1,963,848 vertices, whereas the body-fitted mesh for the FEM solver contained 9,149,056 tetrahedra and 1,523,013 vertices. When run on 32 MPI processes, the wall-clock times were respectively 7710 s for IPFEM and 837 s for FEM. Although the IPFEM solver's computational time is approximately an order of magnitude longer, we consider this entirely acceptable given its fully automated workflow, which requires no human intervention. In contrast, the FEM solver's reported time excludes mesh generation, which cannot be automated.

4.2.2. Voltage-dependent anion channel system

To demonstrate the performance of our IPFEM, we perform numerical tests using a crystallographic molecular structure of a voltage-dependent anion channel (VDAC), with two ionic species K^+ and Cl^-. VDAC is the most abundant protein on the outer mitochondrial membraneas the main conduit for the entry and exit of ionic species, playing a crucial role in regulating cell survival and cell death and characterizing health and diseases. Initial coordinates for VDAC (PDBID: 3EMN) were downloaded from the PDB website and then converted to PQR files by the tool PDB2PQR. The molecule 3EMN contains 4313 atoms, and two views of its molecular structure are given in Fig. 17.

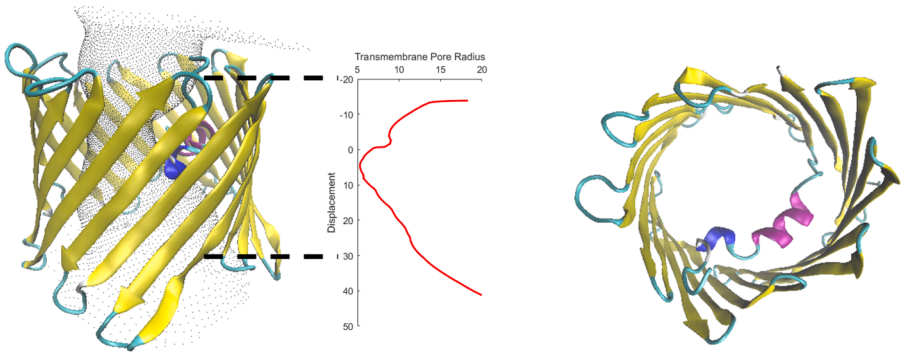

Fig. 17. Molecular structure of 3EMN.

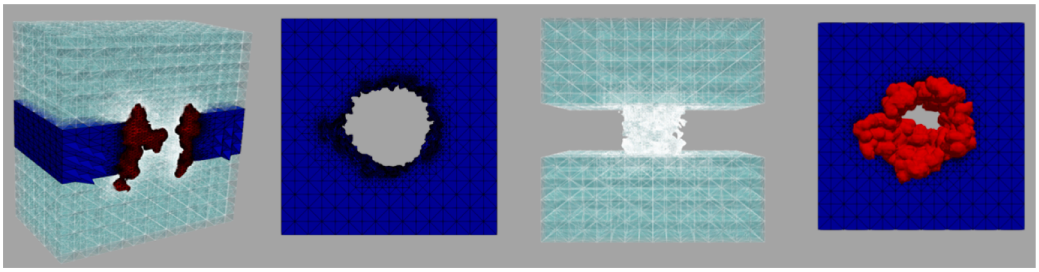

Fig. 18. The final computational mesh generated by FFA on channel protein 3EMN.

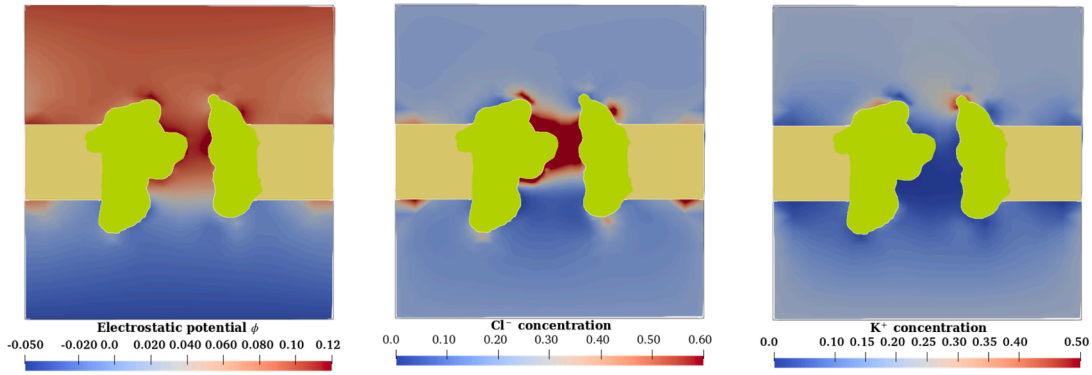

Fig. 19. The electrostatic potential ϕ (V) and the concentrations of K^+ and Cl^- ions (M) in color mapping on a cross section ($x = 0$) of the solvent region for 3EMN. Here the protein and membrane regions are colored in green and yellow, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In our simulation, the diffusion coefficients are the same as those in the gA channel, and the boundaries are set as $z^b_{\text{chan}} = -12$, $z^t_{\text{chan}} = 12$ for the channel region and $z^b_{\text{bulk}} = -15$, $z^t_{\text{bulk}} = 15$ for the bulk region along the z axis.

Fig. 18 shows the final computational mesh generated by our method on channel protein 3EMN, where the first graph is a computational mesh of the membrane-channel protein system, the second is a mesh of the membrane region, the third is a mesh of the solvent region, and the last is a mesh of the solute region consisting of the ion channel and the membrane region.

For the given boundary conditions ($V_m = 100$ mV and $c_{i,\text{bulk}} = 0.2$ M), Fig. 19 illustrates the potential and ion concentration cross sections of the entire domain area.

Fig. 20 shows K^+ and Cl^- concentration profiles for five different applied voltage values, while the bulk concentration is the same ($c_{i,\text{ bulk}} = 0.2\,\text{M}$). It is seen that although different voltage values as boundary condition are applied, the changes of concentrations have almost the same tendency.

The $I - V$ data of simulation are shown in Fig. 21 and it can be seen that the current and voltage are nonlinearly related.

Table 5 shows the numbers of Gummel iterations for the VDAC system under $c_{i,\text{bulk}} = 0.1\text{M}$ at different applied voltages. It is observed that the Gummel iteration typically converges in approximately 38-47 iterations for the VDAC system, to a specified tolerance of 10^{-6} in relative change.

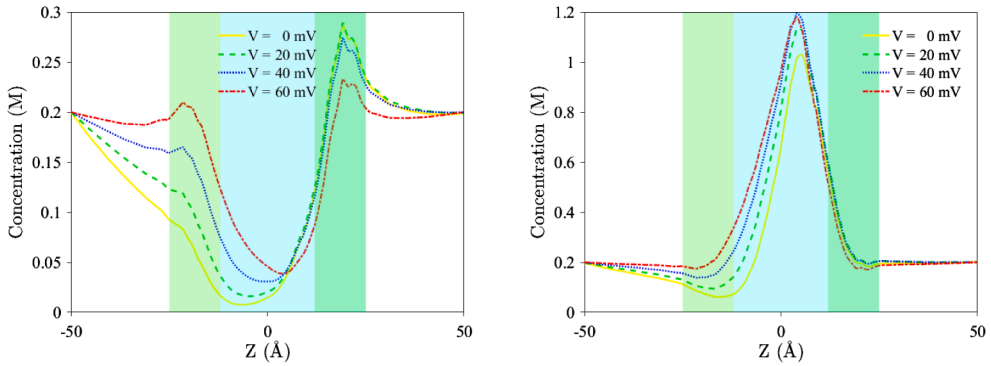

Fig. 20. K^+ (left) and Cl^- (right) concentrations at the center of 3EMN plotted along z-axis obtained from the PNP calculation with $c_{i,\text{bulk}} = 0.2$ M.

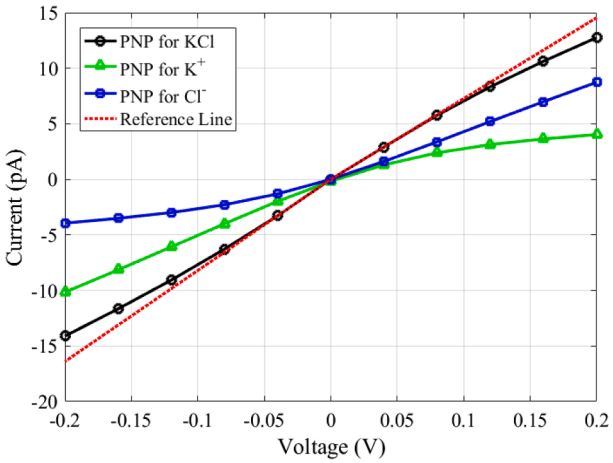

Fig. 21. The computed $I - V$ curves of 3EMN.

Table 5
The number of Gummel iterations for the VDAC system.

Voltage(mV)	0	20	40	60	80	100
Gummel Iteration	47	38	40	42	43	43
Voltage(mV)	120	140	160	180	200	–
Gummel Iteration	43	43	42	42	42	–

5. Conclusion

In this paper, we develop an unfitted-mesh solver for the PNP equations in ion channel simulations based on the IPFEM. A key contribution of our method is eliminating the need for body-fitted mesh generation, which is traditionally required for membrane-channel protein systems. Traditional methods involve constructing complex, irregular meshes for regions such as the ion channel protein, membrane, and ionic solvent, which are computationally demanding and labor-intensive. We propose to use the FFA to efficiently identify and mark the ion channel protein, membrane, and ionic solvent regions, which allows us to successfully generate for the first time an unfitted mesh for more complex membrane-channel protein systems. This includes systems such as membranes with an embedded ion channel protein tilted relative to the z-axis, membranes containing a two-branch pore, and membranes with two embedded ion channel proteins. Our approach significantly reduces the complexity associated with numerical simulations and mesh handling, making it highly applicable to more intricate ion channel models. The accuracy of the method is validated through numerical examples on benchmark problems, demonstrating good performance in solving the PNP equations for both the gA system and the VDAC system. The proposed method can be easily adopted to other types of unfitted meshes, including adaptive Cartesian meshes with hanging nodes. In the future, we will further improve the computational efficiency and stability of unfitted mesh generation and numerical integration over surface elements, and extend our method to more complex models, including transport simulations in complex or multiple channel system, nanopore sequencing, etc.

CRediT authorship contribution statement

Ziyang Liu: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization; **Sheng Gui:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Conceptualization; **Benzhuo Lu:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization; **Linbo Zhang:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Liu and Zhang's work was supported by National Key R&D Program of China (Grant numbers 2020YFA0711900, 2020YFA0711904), and the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant Number XDB0640000). Gui and Lu's work was supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No.XDB0500000) and the National Natural Science Foundation of China (Grant No. 12371413). All authors' work was supported by State Key Laboratory of Mathematical Sciences (SKLMS), and National Center for Mathematics and Interdisciplinary Sciences of Chinese Academy of Sciences (NCMIS).

References

[1] D.-P. Chen, J. Lear, R.S. Eisenberg, Permeation through an open channel: Poisson-Nernst-Planck theory of a synthetic ionic channel, Biophys. J. 72 (1) (1997) 97–116.
[2] R.S. Eisenberg, Ionic channels in biological membranes: natural nanotubes, Acc. Chem. Res. 31 (3) (1998) 117–123.
[3] M.G. Kurnikova, R.D. Coalson, P. Graf, A. Nitzan, A lattice relaxation algorithm for three-dimensional Poisson-Nernst-Planck theory with application to ion transport through the gramicidin a channel, Biophys. J. 76 (2) (1999) 642–656.
[4] B. Roux, T. Allen, S. Bernèche, W. Im, Theoretical and computational models of biological ion channels, Q. Rev. Biophys. 37 (1) (2004) 15.
[5] Q. Zhong, Q. Jiang, P.B. Moore, D.M. Newns, M.L. Klein, Molecular dynamics simulation of a synthetic ion channel, Biophys. J. 74 (1) (1998) 3–10.
[6] M.E. Davis, J.D. Madura, B.A. Luty, J.A. McCammon, Electrostatics and diffusion of molecules in solution: simulations with the university of houston brownian dynamics program, Comput. Phys. Commun. 62 (2–3) (1991) 187–197.
[7] R.J. French, Finite difference methods for the numerical solution of the Nernst-Planck-Poisson equations, in: Mathematical Problems in Biology: Victoria Conference, Springer, 1974, pp. 50–61.
[8] H. Liu, Z. Wang, A free energy satisfying finite difference method for Poisson-Nernst-Planck equations, J. Comput. Phys. 268 (2014) 363–376.
[9] J. Fuhrmann, C. Guhlke, A finite volume scheme for Nernst-Planck-Poisson systems with ion size and solvation effects, in: Finite Volumes for Complex Applications VIII-Hyperbolic, Elliptic and Parabolic Problems: FVCA 8, Lille, France, June 2017 8, Springer, 2017, pp. 497–505.
[10] S.R. Mathur, J.Y. Murthy, A multigrid method for the Poisson-Nernst-Planck equations, Int. J. Heat Mass Transf. 52 (17–18) (2009) 4031–4039.
[11] B. Lu, M.J. Holst, J.A. McCammon, Y.C. Zhou, Poisson-Nernst-Planck equations for simulating biomolecular diffusion–reaction processes I: finite element solutions, J. Comput. Phys. 229 (19) (2010) 6979–6994.
[12] M.S. Metti, J. Xu, C. Liu, Energetically stable discretizations for charge transport and electrokinetic models, J. Comput. Phys. 306 (2016) 1–18.
[13] B. Tu, M. Chen, Y. Xie, L. Zhang, R.S. Eisenberg, B. Lu, A parallel finite element simulator for ion transport through three-dimensional ion channel systems, J. Comput. Chem. 34 (24) (2013) 2065–2078.
[14] Q. Wang, H. Li, L. Zhang, B. Lu, A stabilized finite element method for the Poisson–Nernst–Planck equations in three-dimensional ion channel simulations, Appl. Math. Lett. 111 (2021) 106652.
[15] D. Xie, Z. Chao, A finite element iterative solver for a PNP ion channel model with neumann boundary condition and membrane surface charge, J. Comput. Phys. 423 (2020) 109915.
[16] D. Xie, B. Lu, An effective finite element iterative solver for a Poisson–Nernst–Planck ion channel model with periodic boundary conditions, SIAM J. Sci. Comput. 42 (6) (2020) B1490–B1516.
[17] T. Liu, S. Bai, B. Tu, M. Chen, B. Lu, Membrane-channel protein system mesh construction for finite element simulations, Computational and Mathematical Biophysics 3 (1) (2015) 128–139.
[18] M. Chen, B. Lu, TMSmesh: A robust method for molecular surface mesh generation using a trace technique, J. Chem. Theory Comput. 7 (1) (2011) 203–212.
[19] M. Chen, B. Tu, B. Lu, Triangulated manifold meshing method preserving molecular surface topology, J. Mol. Graph. Modell. 38 (2012) 411–418.
[20] T. Liu, M. Chen, B. Lu, Efficient and qualified mesh generation for Gaussian molecular surface using adaptive partition and piecewise polynomial approximation, SIAM J. Sci. Comput. 40 (2) (2018) B507–B527.
[21] T. Liu, M. Chen, Y. Song, H. Li, B. Lu, Quality improvement of surface triangular mesh using a modified Laplacian smoothing approach avoiding intersection, PLoS ONE 12 (9) (2017) e0184206.
[22] S. Hang, Tetgen, a delaunay-based quality tetrahedral mesh generator, ACM Trans. Math. Softw. 41 (2) (2015) 1–36.
[23] Z. Chao, S. Gui, B. Lu, D. Xie, Efficient generation of membrane and solvent tetrahedral meshes for ion channel finite element calculation, Int. J. Numer. Anal. Model. 19 (6) (2022) 887–906.
[24] L. Jemison, M. Stahl, R.K. Dash, D. Xie, VDAC solvation free energy calculation by a nonuniform size modified Poisson–Boltzmann ion channel model, J. Comput. Chem. 46 (1) (2025) e70003.
[25] D. Khan, S. Gui, Z. Cheng, Molecular surface mesh smoothing with subdivision, Lect. Notes Comput. Sci. 14496 (2024) 236–248.
[26] D. Khan, D.-M. Yan, S. Gui, B. Lu, X. Zhang, Molecular surface remeshing with local region refinement, Int. J. Mol. Sci. 19 (5) (2018) 1383.
[27] C.-C. Chu, I. Graham, T.-Y. Hou, A new multiscale finite element method for high-contrast elliptic interface problems, Math. Comput. 79 (272) (2010) 1915–1955.
[28] R.J. LeVeque, Z. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources, SIAM J. Numer. Anal. 31 (4) (1994) 1019–1044.

[29] Z. Li, The immersed interface method using a finite element formulation, Appl. Numer. Math. 27 (3) (1998) 253–267.
[30] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, Int. J. Numer. Methods Eng. 46 (1) (1999) 131–150.
[31] H. Wu, Y. Xiao, An unfitted hp-interface penalty finite element method for elliptic interface problems, J. Comput. Math. 37 (3) (2019) 316–339.
[32] I. Babuška, The finite element method for elliptic equations with discontinuous coefficients, Computing 5 (3) (1970) 207–213.
[33] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche's method, for elliptic interface problems, Comput. Methods Appl. Mech. Eng. 191 (47–48) (2002) 5537–5552.
[34] E. Burman, P. Hansbo, M. Larson, A cut finite element method with boundary value correction, Math. Comput. 87 (310) (2018) 633–657.
[35] Y. Xiao, J. Xu, F. Wang, High-order extended finite element methods for solving interface problems, Comput. Methods Appl. Mech. Eng. 364 (2020) 112964.
[36] R. Massjung, An unfitted discontinuous Galerkin method applied to elliptic interface problems, SIAM J. Numer. Anal. 50 (6) (2012) 3134–3162.
[37] H. Liu, L. Zhang, X. Zhang, W. Zheng, Interface-penalty finite element methods for interface problems in H1, H (curl), and H (div), Comput. Methods Appl. Mech. Eng. 367 (2020) 113137.
[38] Z. Liu, S. Gui, B. Lu, L. Zhang, An unfitted finite element Poisson–Boltzmann solver with automatic resolving of curved molecular surface, J. Phys. Chem. B 128 (27) (2024) 6463–6475.
[39] H.K. Gummel, A self-consistent iterative scheme for one-dimensional steady state transistor calculations, IEEE Trans. Electron Dev. 11 (10) (1964) 455–465.
[40] S.V. Burtsev, Y.P. Kuzmin, An efficient flood-filling algorithm, Comput. Graph. 17 (5) (1993) 549–561.
[41] I.-L. Chern, J.-G. Liu, W.-C. Wang, Accurate evaluation of electrostatics for macromolecules in solution, Methods Appl. Anal. 10 (2) (2003) 309–328.
[42] D. Arnold, An interior penalty finite element method with discontinuous elements, SIAM J. Numer. Anal. 19 (4) (1982) 742–760.
[43] E. Burman, A. Ern, Continuous interior penalty hp-finite element methods for advection and advection-diffusion equations, Math. Comput. 76 (259) (2007) 1119–1140.
[44] T. Cui, W. Leng, H. Liu, L. Zhang, W. Zheng, High-order numerical quadratures in a tetrahedron with an implicitly defined curved interface, ACM Trans. Math. Softw. 46 (1) (2020) 1–18.
[45] L. Zhang, A parallel algorithm for adaptive local refinement of tetrahedral meshes using bisection, Numer. Math. Theory Methods Appl. 2 (1) (2009) 65–89.
[46] R.E. Koeppe, 2nd, O.S. Anderson, Engineering the gramicidin channel, Annu. Rev. Biophys. Biomol. Struct. 25 (1996) 231–258.
[47] T.J. Dolinsky, J.E. Nielsen, J.A. McCammon, N.A. Baker, PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations, Nucl. Acids Res. 32 (2) (2004) 665–667.
[48] D. Gillespie, Energetics of divalent selectivity in a calcium channel: the ryanodine receptor case study, Biophys. J. 94 (4) (2008) 1169–1184.
[49] H. Hwang, G.C. Schatz, M.A. Ratner, Incorporation of inhomogeneous ion diffusion coefficients into kinetic lattice grand canonical Monte Carlo simulations and application to ion current calculations in a simple model ion channel, J. Phys. Chem. A 111 (49) (2007) 12506–12512.
[50] Q. Zhang, S. Gui, H. Li, B. Lu, Model reduction-based initialization methods for solving the Poisson-Nernst-Plank equations in three-dimensional ion channel simulations, J. Comput. Phys. 419 (2020) 109627.
[51] C.D. Cole, A.S. Frost, N. Thompson, M. Cotten, T.A. Cross, D.D. Busath, Noncontact dipole effects on channel permeation. VI. 5F-and 6F-Trp gramicidin channel currents, Biophys. J. 83 (4) (2002) 1974–1986.
[52] J. Ding, S. Zhou, Second-order, positive, and unconditional energy dissipative scheme for modified Poisson-Nernst-Planck equations, J. Comput. Phys. 510 (2024) 113094.
[53] C. Liu, C. Wang, S.M. Wise, X. Yue, S. Zhou, A positivity-preserving, energy stable and convergent numerical scheme for the Poisson-Nernst-Planck system, Math. Comput. 90 (331) (2021) 2071–2106.
[54] J. Shen, J. Xu, Unconditionally positivity preserving and energy dissipative schemes for Poisson–Nernst–Planck equations, Numer. Math. 148 (3) (2021) 671–697.