Received: 31 March 2021 | Revised: 30 June 2021 | Accepted: 9 July 2021

DOI: 10.1002/jcc.26723

FULL PAPER

Journal of COMPUTATIONAL CHEMISTRY WILEY

An improved Poisson-Nernst-Planck ion channel model and numerical studies on effects of boundary conditions, membrane charges, and bulk concentrations

Zhen Chao[1] | Dexuan Xie[2]

[1]Department of Mathematics, University of Michigan, Ann Arbor, Michigan, USA

[2]Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, Wisconsin, USA

Correspondence
Dexuan Xie, Department of Mathematics, University of Michigan, Ann Arbor, MI 48109, USA.
Email: dxie@uwm.edu

Funding information
Simons Foundation

Abstract

In this paper, an improved Poisson-Nernst-Planck ion channel (PNPic) model is presented, along with its effective finite element solver and software package for an ion channel protein in a solution of multiple ionic species. Numerical studies are then done on the effects of boundary value conditions, membrane charges, and bulk concentrations on electrostatics and ionic concentrations for an ion channel protein, a gramicidin A (gA), and five different ionic solvents with up to four species. Numerical results indicate that the cation selectivity property of gA occurs within a central portion of ion channel pore, insensitively to any change of boundary value condition, membrane charge, or bulk concentration. Moreover, a numerical scheme for computing the electric currents induced by ion transports across membrane via an ion channel pore is presented and implemented as a part of the PNPic finite element package. It is then applied to the calculation of current–voltage curves, well validating the PNPic model and finite element package by electric current experimental data.

KEYWORDS
cation selectivity, electric current calculation, finite element method, ion channel model, Poisson-Nernst-Planck equations

1 | INTRODUCTION

Transport of ions through an ion channel pore is a fundamental process in cell biology. However, its simulation is very complicated and challenging in atomistic level. A system of Poisson-Nernst-Planck (PNP) equations is one popular dielectric continuum model for calculating the electrostatics and ionic concentrations induced by charges from an ion channel protein, a membrane, and an ionic solution. It has been widely applied to the simulation of ionic transport and the studies of various membrane kinetics such as membrane potentials, conductances, transport fluxes, and electric current, and so forth, in the one-dimensional (1D) case. For instance, the Teorell-Meyer-Sievers multi-layered model[1,2] was successfully used in the numerical simulation of nonequilibrium diffuse double layer[3] and the research on ionic transport in ion-exchange membranes.[4] Mathematically, several 1D steady-state PNP systems are studied for their solution existence and uniqueness.[5–7] Analytic PNP solutions for some 1D cases are also derived in References 8,9. However, in a realistic bimolecular context, it is almost impossible to find a three-dimensional (3D) PNP analytic solution. Hence, developing numerical methods becomes essential to search for PNP approximate solutions. So far, different PNP numerical algorithms were developed in the past two decades by using finite difference methods,[10,11] finite element methods,[12,13] a matched interface and boundary method,[14] a finite volume method,[15] a hybrid finite-difference/finite-volume method,[16] and a spectral element method,[17] and so forth. Of these numerical methods, the finite element method is the most suitable to handle an irregular molecular surface of an ion channel protein and complicated interface conditions to generate a PNP approximate solution in high accuracy. Hence, we will use the finite element method to develop numerical algorithms for solving an improved PNP ion channel (PNPic) model to be presented in this paper.

A PNPic model is a system of partial differential equations for one electrostatic potential function u and n ionic concentration functions for a solvent of n ionic species. Here u is defined in a box domain Ω, which is partitioned into a protein region, D_p, a membrane region, D_m, and a solvent region, D_s. It is known that one major difficulty of solving a PNP model comes from the singularity caused by the atomic charges of an ion channel protein. This difficulty has been overcome by adapting solution decomposition techniques developed in the numerical solution of the Poisson-Boltzmann equation (PBE).[18,19] Recall that in the solution decomposition from Reference 18, the electrostatic potential function u is split into three component functions within the protein region D_p only, resulting in a Laplace boundary value problem in D_p, which has serious regularity problems since D_p is strongly non-convex. In contrast, in the solution decomposition from Reference 19, u is split into three component functions, denoted by G, Ψ, and $\widetilde{\Phi}$, over the whole box domain Ω such that $u = G + \Psi + \widetilde{\Phi}$ with G, Ψ, and $\widetilde{\Phi}$ being the potentials induced by atomic charges, the potentials and charges on interfaces and boundaries, and ionic charges, respectively. Since G contains all the singularity points of u, both Ψ and $\widetilde{\Phi}$ become much easier to calculate numerically than the original u. Hence, we adapt the splitting technique from Reference 19 for the development of our PNPic finite element solver in this work.

Actually, a box domain Ω is a truncation of an infinite ion channel domain such that Ω contains only one ion channel protein embedded in the central position of a membrane. To this end, a PNPic model is defined as a boundary value problem, where Dirichlet boundary value conditions are commonly used due to their simplicity in implementation and flexibility in application. Hence, in this work, we develop an improved PNPic model using Dirichlet boundary value conditions and its finite element solver based on our recent work,[20] in which we constructed a PNPic model using Neumann boundary value conditions on the four side surfaces of Ω to avoid the difficulties of selecting boundary value functions. With our new PNPic solver, in this paper, we will carry out numerical experiments to explore the PNPic solution differences caused by Dirichlet and Neumann boundary value conditions. So far, none of such comparison tests were done due to lacking of required software packages. Together with our recent PNPic package, our new PNPic package makes such numerical studies possible.

From biochemistry and physiology it has been known that charges from membrane can have significant effects on electrostatic and ionic concentration. To reflect such effects, we recently added a membrane surface charge density function, σ, to the both sides of a membrane in a PNPic model[20] and a size modified PBE ion channel model.[21] But, due to the interactions of a charged membrane with cations and anions surrounding the membrane, charges on each side surface of a membrane may have different values.[22] This motivated us to define σ as a piecewise function with two expressions—one for the bottom surface and the other one for the top surface of the membrane. This improvement will give us more options to construct numerical tests for the study of membrane charge effects.

Naturally, a solvent domain is separated by a membrane into three portions—an extracellular portion, an intracellular portion, and a channel pore portion. Different portions may have different bulk concentrations and different diffusion constants for different ionic species. To reflect this fact, in this work, we will define bulk concentrations and diffusion constants as piecewise constant functions so that we can simulate different experimental environments through selecting either different bulk concentrations or different diffusion constants within one or more than one portion of the solvent domain. For example, we can add a mixture of 0.2 mol NaCl and 0.1 mol KNO_3 to an extracellular portion only similarly to what is done in a chemical laboratory.

Our PNPic finite element package will be a valuable tool for ion channel studies and simulations. In this paper, we will use it to carry out numerical tests to study the effects of boundary value conditions, membrane charges, simulation domain sizes, and bulk concentrations on electrostatic potentials and ionic concentrations for an ion channel protein, a gramicidin A (gA). In these tests, we construct five different ionic solvents with up to four ionic species and three box meshes with different mesh sizes. These numerical tests demonstrate that the cation selectivity property of gA occurs within a central portion of ion channel pore, insensitively to any change of boundary value conditions, membrane charges, bulk concentrations, or mesh sizes.

One important application of a PNPic model is to calculate the electric currents induced by ion transport across a membrane via an ion channel pore. In this paper, we will present a numerical scheme for computing electric current in terms of a PNPic finite element solution. In fact, it is nontrivial to do so due to difficulties caused by an irregular cross section S of an ion channel pore and a nonuniform flux field, which involves the gradient vectors of potential u and ionic concentration functions c_i. From the finite element theory it is known that a direct calculation of these two gradient vectors produces a discontinuous flux field when u and c_i are approximated as linear finite element functions. This may cause numerical errors since from the definition of a PNPic model it is known that the flux field is continuous. Hence, in order to derive an effective numerical scheme for computing current, we not only need a triangular mesh of an irregular cross section but also need to find a way to retain the continuity property of a flux field. In addition, a proper quadrature is needed to calculate each related surface integral. We will do so and implement the new scheme as a tool for computing current in this work. With this new tool, we will produce I-V (current–voltage) curves for gA and validate our PNPic model and package by experimental data reported in Reference 23.

The rest of the paper is organized into three parts as follows. In the part of Section 2, we present our PNPic model, our PNPic finite element method, and our numerical scheme for computing electric current. In the part of Section 3, we present numerical studies, numerical results, and validation tests. The paper is closed with the part of Section 4.

2 | METHODS

2.1 | An improved Possion-Nernst-Planck ion channel model

Let a rectangular box open domain, Ω, be defined by

$$\Omega = \{(x,y,z) \mid L_{x_1} < x < L_{x_2}, L_{y_1} < y < L_{y_2}, L_{z_1} < x < L_{z_2}\},$$

where L_{x_1}, L_{x_2}, L_{y_1}, L_{y_2}, L_{z_1}, and L_{z_2} are real numbers. The domain Ω is divided as

$$\Omega = D_p \cup D_m \cup D_s \cup \Gamma_m \cup \Gamma_p \cup \Gamma_{pm}, \tag{1}$$

where D_p denotes a protein region, which contains an ion channel protein molecular structure with n_p atoms; D_s is a solvent region, which contains a solution of n ionic species; D_m is a membrane region; and Γ_m, Γ_p, and Γ_{pm} denote the interfaces between D_m and D_s, D_p and D_s, and D_p and D_m, respectively. The interface Γ_m and the boundary $\partial\Omega$ of Ω are further split by

$$\Gamma_m = \Gamma_{mb} \cup \Gamma_{mt}, \quad \partial\Omega = \Gamma_D \cup \Gamma_{Ns} \cup \Gamma_{Nm}, \tag{2}$$

where Γ_{mb} and Γ_{mt} denote the bottom and top surfaces of membrane, respectively, Γ_D consists of the bottom and top surfaces of $\partial\Omega$, $\Gamma_{Ns} = \Gamma_N \cap \partial D_s$, and $\Gamma_{Nm} = \Gamma_N \cap \partial D_m$. Here Γ_N consists of the four side surfaces of Ω, ∂D_s denote the boundary of D_s, and ∂D_m the boundary of D_m. Moreover, we set the z-axis direction as the membrane normal direction and determine the membrane location by two real numbers $Z1$ and $Z2$. An illustration of these partitions is given in Figure 1, where different interfaces and boundary surfaces are drawn in different line styles and different colors. A view of an interface fitted irregular tetrahedral mesh of a box domain, which is used in numerical calculations, is also displayed in Figure 2 to illustrate the complicated shapes of D_p (in green color), D_m (in yellow color), and D_s (in blue-violet color). Here the channel pore part of D_s is hidden by D_p. See Figure 5 for two views of a mesh of D_s. These meshes were generated by our finite element mesh generation tool,[24] which is an improvement of the mesh package reported in Reference 25.

Let c_i denote a concentration function of the i-th ionic species in moles per liter (mol/L) and u a dimensionless electrostatic potential. The three regions D_m, D_p, and D_s are treated as dielectric media with permittivity constants ϵ_m, ϵ_p, and ϵ_s, respectively. We define a PNPic model (in steady state) by coupling the Nernst-Planck equations

$$\nabla \cdot D_i(\mathbf{r})[\nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r})] = 0, \quad \mathbf{r} \in D_s, \quad i = 1,2,\cdots,n, \tag{3}$$

with the Poisson dielectric equations

$$-\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, \mathbf{r} \in D_p, \tag{4a}$$

$$-\epsilon_m \Delta u(\mathbf{r}) = 0, \mathbf{r} \in D_m, \tag{4b}$$

$$-\epsilon_s \Delta u(\mathbf{r}) = \beta \sum_{i=1}^{n} Z_i c_i(\mathbf{r}), \mathbf{r} \in D_s, \tag{4c}$$

together with the interface conditions

$$u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, \quad \mathbf{s} \in \Gamma_p, \tag{5a}$$

$$u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_m \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} + \tau\sigma(\mathbf{s}), \quad \mathbf{s} \in \Gamma_m, \tag{5b}$$

$$u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, \quad \mathbf{s} \in \Gamma_{pm}, \tag{5c}$$

the Robin boundary value conditions

$$-D_i(\mathbf{s})[\nabla c_i(\mathbf{s}) + Z_i c_i(\mathbf{s}) \nabla u(\mathbf{s})] \cdot \mathbf{n}_s(\mathbf{s}) = 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m, \quad i = 1,2,...,n,$$

which reflect the fact that the channel walls are insulating (i.e., charged particles cannot penetrate them), and the Dirichlet boundary value conditions

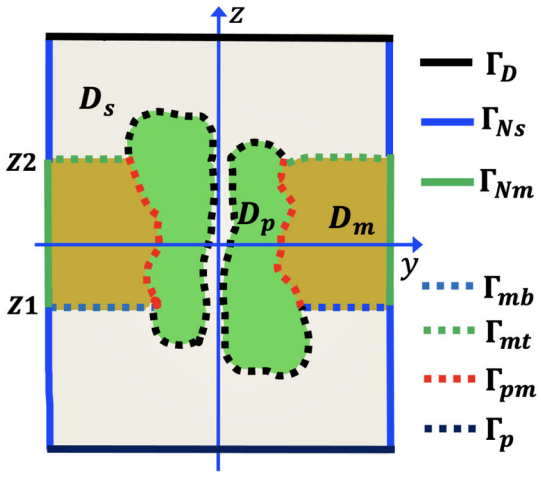


FIGURE 1 An illustration of box domain partition (1), interface and boundary partition (2), and membrane location numbers $Z1$ and $Z2$

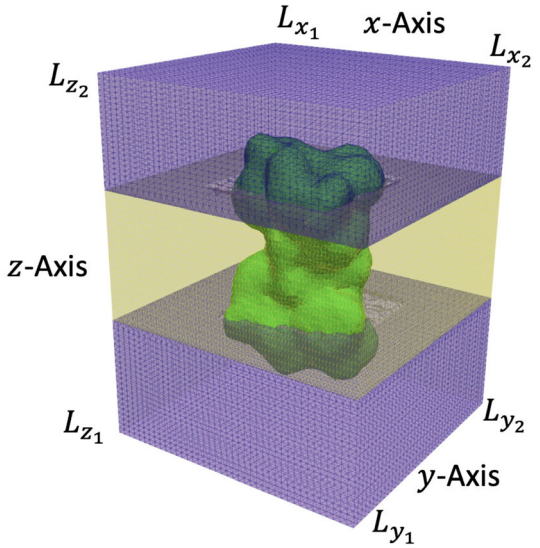


FIGURE 2 An interface fitted tetrahedral mesh of box domain Ω. here D_p, D_m, and D_s are highlighted in green, yellow, and blue-violet

$$u(\mathbf{s}) = g(\mathbf{s}), \quad \mathbf{s} \in \partial\Omega, \tag{6a}$$

$$c_i(\mathbf{s}) = g_i(\mathbf{s}), \quad \mathbf{s} \in \Gamma_D \cup \Gamma_{Ns}, \quad i = 1,2,...,n. \tag{6b}$$

Here D_i denotes a diffusion coefficient function of the i-th ionic species; g and g_i are given boundary value functions; σ denotes a membrane surface charge density function; Z_i is the charge number of the i-th ionic species; z_j and $\mathbf{r}_j$ are the charge number and atomic position of the j-th atom, respectively; $\delta_{\mathbf{r}_j}$ denotes the Dirac delta distribution at $\mathbf{r}_j$; $\mathbf{n}_m$, $\mathbf{n}_p$, and $\mathbf{n}_s$ denote the unit outward normal directions of D_m, D_p, and D_s, respectively; and the constants α,β, and τ are given by

$$\alpha = \frac{10^{10} e_c^2}{\epsilon_0 k_B T}, \quad \beta = \frac{N_A e_c^2}{10^{17} \epsilon_0 k_B T}, \quad \tau = \frac{10^{-12} e_c}{\epsilon_0 k_B T},$$

with N_A being the Avogadro constant (an estimate of the number of ions per mol), e_c the elementary charge, k_B the Boltzmann constant, and T the absolute temperature.

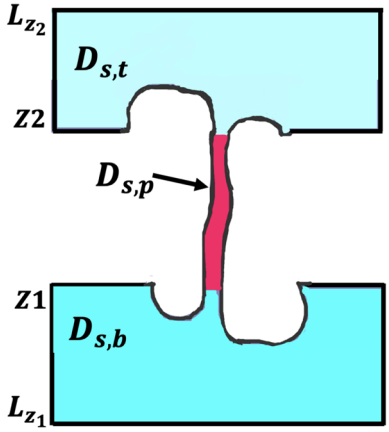


FIGURE 3 An illustration of the partition (8) of a solvent domain D_s. Here $D_{s,b}$ and $D_{s,t}$ can be regarded as the extracellular and intracellular compartments, respectively

In this work, we define σ as a piecewise function as follows:

$$\sigma(\mathbf{s}) = \begin{cases} \sigma_t, & \mathbf{s} \in \Gamma_{mt}, \\ \sigma_b, & \mathbf{s} \in \Gamma_{mb}, \end{cases} \tag{7}$$

where σ_b and σ_t denote the surface charge density functions defined on the bottom surface Γ_{mb} and top surface Γ_{mt} of membrane, respectively. This membrane surface charge density function is an improvement of the one used in Reference 20. It enables us to deal with more membrane charge cases.

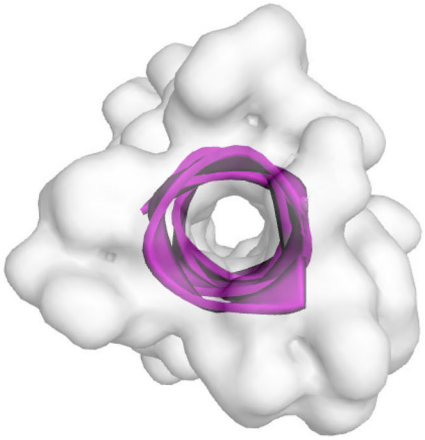
Because of membrane, the solvent domain D_s can be naturally divided into the bottom portion $D_{s,b}$, top portion $D_{s,t}$, and pore portion $D_{s,p}$, as illustrated in Figure 3, such that

$$D_s = D_{s,b} \cup D_{s,t} \cup D_{s,p}, \tag{8}$$

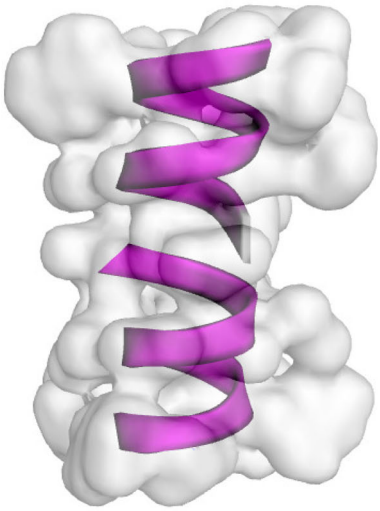
where $D_{s,b} = \{\mathbf{r} \in D_s \mid \mathbf{r} = (x,y,z)$ with $z < Z1\}$, $D_{s,t} = \{\mathbf{r} \in D_s \mid \mathbf{r} = (x,y,z)$ with $z > Z2\}$, and $D_{s,p} = \{\mathbf{r} \in D_s \mid \mathbf{r} = (x,y,z)$ with $Z1 \le z \le Z2\}$. Noting that the bulk concentration c_i^b and diffusion coefficient D_i of ionic species i within these three portions may be different, we define them by piecewise constant functions as follows:

$$c_i^b(\mathbf{r}) = \begin{cases} c_{i,p}^b, & \mathbf{r} \in D_{s,p}, \\ c_{i,b}^b, & \mathbf{r} \in D_{s,b}, \\ c_{i,t}^b, & \mathbf{r} \in D_{s,t}, \end{cases} \quad D_i(\mathbf{r}) = \begin{cases} D_{i,p}, & \mathbf{r} \in D_{s,p}, \\ D_{i,b}, & \mathbf{r} \in D_{s,b}, \\ D_{i,t}, & \mathbf{r} \in D_{s,t}, \end{cases} \quad i = 1,2,...,n, \tag{9}$$

where $c_{i,p}^b$ and $D_{i,p}$, $c_{i,b}^b$ and $D_{i,b}$, and $c_{i,t}^b$ and $D_{i,t}$ denote the bulk concentration and diffusion constant of species i within the three portions $D_{s,p}$, $D_{s,b}$ and $D_{s,t}$, respectively. The above diffusion coefficient functions work for a finite element approximation of Nernst-Planck equation (see Reference 17). To ensure that the Nernst-Planck equations of (3) are well defined, we modify D_i as a smooth function in the expression



(A) Top view



(B) Side view

FIGURE 4 Two views of a gA ion channel molecular structure (in cartoon) and its Van deer Waals volume (in gray) occupied by the atomic balls of gA

$$D_i(\mathbf{r}) = \begin{cases} D_{i,t}, & \mathbf{r} \in D_{s,t} \text{ (top portion of } D_s), \\ D_{i,p} + (D_{i,t} - D_{i,p}) I_2(\mathbf{r}), & \mathbf{r} \in D_{b,2} \text{ (buffer region 2)}, \\ D_{i,p}, & \mathbf{r} \in D_{cp} \text{ (central pore region)}, \\ D_{i,p} + (D_{i,b} - D_{i,p}) I_1(\mathbf{r}), & \mathbf{r} \in D_{b,1} \text{ (buffer region 1)}, \\ D_{i,b}, & \mathbf{r} \in D_{s,b} \text{ (bottom portion of } D_s), \end{cases} \tag{10}$$

where the two buffer regions $D_{b,1}$ and $D_{b,2}$ are defined by

$$D_{b,1} = \left\{ \mathbf{r} \in D_s \mid \mathbf{r} = (x,y,z) \text{ with } Z1 \leqslant z \leqslant Z1 + \hat{h} \right\},$$
$$D_{b,2} = \left\{ \mathbf{r} \in D_s \mid Z2 - \hat{h} \leqslant z \leqslant Z2 \right\},$$

with $\hat{h}$ being a parameter for adjusting buffer region size, the central pore region D_{cp} is defined by

$$D_{cp} = \left\{ \mathbf{r} \in D_s \mid \mathbf{r} = (x,y,z) \text{ with } Z1 + \hat{h} \leqslant z \leqslant Z2 - \hat{h} \right\}, \tag{11}$$

and I_1 and I_2 denote the two interpolation functions satisfying the interpolation conditions

$$I_1(x,y,Z1) = 1, \quad I_1\left(x,y,Z1+\hat{h}\right) = 0, \quad I_2(x,y,Z2) = 1, \quad I_2\left(x,y,Z2-\hat{h}\right) = 0.$$

Since none of $D_{i,p}$ were known in experiments, we estimate $D_{i,p}$ by

$$D_{i,p} = \theta D_{i,b} \quad \text{for} \quad 0 < \theta < 1. \tag{12}$$

Consequently, with (5a), (7), (9), and (10), we have obtained an improved PNPic model in the case of using the Dirichlet boundary value conditions. For clarity, we will call this model either our PNPic model or the PNPic model in the remaining part of this paper.

2.2 | Our PNPic finite element solver

To overcome the singularity difficulty caused by atomic charges, we define a potential function, w, independent of any ionic concentration c_i, by

$$w(\mathbf{r}) = G(\mathbf{r}) + \Psi(\mathbf{r}), \quad \mathbf{r} \in \Omega, \tag{13}$$

where G is given by

$$G(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|},$$

and $\Psi(\mathbf{r})$ is a solution of a linear interface boundary value problem as follows:

$$\begin{cases} \Delta\Psi(\mathbf{r}) = 0, & \mathbf{r} \in D_p \cup D_s \cup D_m, \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_s \dfrac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})} + (\epsilon_s - \epsilon_p) \dfrac{\partial G(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_p, \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_m \dfrac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_s \dfrac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} + (\epsilon_s - \epsilon_m) \dfrac{\partial G(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} + \tau\sigma(\mathbf{s}), & \mathbf{s} \in \Gamma_m, \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \dfrac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})} + (\epsilon_m - \epsilon_p) \dfrac{\partial G(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_{pm}, \\ \Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s}), & \mathbf{s} \in \partial\Omega. \end{cases} \tag{14}$$

Using (13), we can simplify the PNPic model as a nonlinear system for finding another potential function, $\widetilde{\Phi}$, and the n ionic concentrations c_i for $i = 1, 2, \ldots, n$ as follows:

$$\begin{cases} \nabla \cdot D_i(\mathbf{r}) \left[\nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla \left(w(\mathbf{r}) + \widetilde{\Phi}(\mathbf{r}) \right) \right] = 0, & \mathbf{r} \in D_s, \quad i = 1,2,\ldots,n, \\ \dfrac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(\mathbf{s}) \dfrac{\partial \left(w(\mathbf{s}) + \widetilde{\Phi}(\mathbf{s}) \right)}{\partial \mathbf{n}_s(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\ c_i(\mathbf{s}) = g_i(\mathbf{s}), & \mathbf{s} \in \Gamma_{Ns} \cup \Gamma_D, \\ \Delta\widetilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r} \in D_p \cup D_m, \\ -\epsilon_s \Delta\widetilde{\Phi}(\mathbf{r}) = \beta \sum_{j=1}^{n} Z_j c_j(\mathbf{r}), & \mathbf{r} \in D_s, \\ \widetilde{\Phi}(\mathbf{s}^-) = \widetilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_s \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_p, \\ \widetilde{\Phi}(\mathbf{s}^-) = \widetilde{\Phi}(\mathbf{s}^+), \quad \epsilon_m \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_s \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})}, & \mathbf{s} \in \Gamma_m, \\ \widetilde{\Phi}(\mathbf{s}^-) = \widetilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \dfrac{\partial \widetilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_{pm}, \\ \widetilde{\Phi}(\mathbf{s}) = 0, & \mathbf{s} \in \partial\Omega. \end{cases} \tag{15}$$

Note that both (14) and (15) are well defined without involving any singularity since G has collected all the singular points $\mathbf{r}_j$ of the PNPic model. Hence, they can be solved more easily than the PNPic model and their solutions give the PNPic solution (u,c) with $u = G + \Psi + \widetilde{\Phi}$ and $c = (c_1, c_2, \ldots, c_n)$.

We now construct the finite element approximations of (14) and (15).

To do so, we first generate an interface fitted irregular tetrahedral mesh, Ω_h, of Ω, an irregular tetrahedral mesh, $D_{s,h}$, of D_s. We then use them to construct two linear finite element spaces, U and V, and their subspaces, U_0 and V_0, respectively, as follows:

$$U_0 = \{\nu \in U \mid \nu = 0 \text{ on } \partial\Omega\}, \quad V_0 = \{\mu \in V \mid \mu = 0 \text{ on } \Gamma_{Ns} \cup \Gamma_D\}.$$

Using the above finite element spaces, we can get a finite element approximation of (14) as follows:

Find $\Psi \in U$ satisfying $\Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s})$ on $\partial\Omega$ such that

$$a(\Psi, \nu) = L(\nu) \qquad \forall \nu \in U_0, \tag{16}$$

where $a(\cdot, \cdot)$ is a bilinear functional defined by

$$a(\Psi,\nu)=\epsilon_p\int_{D_p}\nabla\Psi(\mathbf{r})\cdot\nabla\nu(\mathbf{r})d\mathbf{r}+\epsilon_m\int_{D_m}\nabla\Psi(\mathbf{r})\cdot\nabla\nu(\mathbf{r})d\mathbf{r}$$
$$+\epsilon_s\int_{D_s}\nabla\Psi(\mathbf{r})\cdot\nabla\nu(\mathbf{r})d\mathbf{r},$$

and $L(\cdot)$ is a linear functional defined by

$$L(\nu)=(\epsilon_s-\epsilon_p)\int_{\Gamma_p}\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_p(\mathbf{s})}\nu(\mathbf{s})d\mathbf{s}+(\epsilon_s-\epsilon_m)\int_{\Gamma_m}\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_m(\mathbf{s})}\nu(\mathbf{s})d\mathbf{s}$$
$$+(\epsilon_m-\epsilon_p)\int_{\Gamma_{pm}}\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_p(\mathbf{s})}\nu(\mathbf{s})d\mathbf{s}+\tau\int_{\Gamma_m}\sigma(\mathbf{s})\nu(\mathbf{s})d\mathbf{s}.$$

Once a solution Ψ of (16) is found, we can obtain a finite element approximation of (15) as follows: Find $\widetilde{\Phi}\in U_0$ and $c_i\in V$ satisfying $c_i=g_i$ on $\Gamma_{Ns}\cup\Gamma_D$ for $i=1,2,...,n$ such that

$$\begin{cases}\int_{D_s}D_i\left[\nabla c_i+Z_ic_i\nabla\left(w+\widetilde{\Phi}\right)\right]\nabla\mu_i d\mathbf{r}=0 & \forall\mu_i\in V_0,\quad i=1,2,...,n,\\ a\left(\widetilde{\Phi},\nu\right)-\beta\sum_{j=1}^{n}Z_j\int_{D_s}c_j\nu d\mathbf{r}=0 & \forall\nu\in U_0,\end{cases}\tag{17}$$

where w has been defined in (13).

We developed effective iterative schemes for solving the linear finite element Equation (16) and the nonlinear finite element system (17), respectively, and implemented them in Python and Fortran as a software package based on our previous work.[20] We omit the description of these iterative schemes here since these schemes are very similar to those given in Reference 20.

2.3 | Electric current calculation

As one important application of our PNPic model, we present a numerical scheme for computing electric currents induced by ion transports across the membrane through an ion channel pore. These currents include the electric currents $I_{i,S}$ of species i and total current I_S as follows:

$$I_{i,S}=-\frac{e_cN_A}{10^3}Z_iD_{i,p}\int_S\left[\frac{\partial c_i(\mathbf{s})}{\partial z}+Z_ic_i(\mathbf{s})\frac{\partial u(\mathbf{s})}{\partial z}\right]d\mathbf{s}\ \text{ for } i=1,2,...,n,\quad I_S=\sum_{i=1}^{n}I_{i,S},\tag{18}$$

where S denotes a cross section of an ion channel pore with the normal direction being the z-axis direction, $D_{i,p}$ is a diffusion constant of species i within the channel pore in Å/ps (pico-second), and the currents have been measured in pA (pico-ampere).

However, the surface integral $I_{i,S}$ is difficult to calculate since an ion channel pore may have a very complicated shape, causing cross section S to have an irregular boundary curve. Another difficulty arises from the calculation of partial derivatives $\frac{\partial c_i(\mathbf{s})}{\partial z}$ and $\frac{\partial u(\mathbf{s})}{\partial z}$ as required in the calculation of terms, J_i, by the formulas

$$J_i=\frac{\partial c_i(\mathbf{s})}{\partial z}+Z_ic_i(\mathbf{s})\frac{\partial u(\mathbf{s})}{\partial z},\quad i=1,2,...,n.$$

In fact, u and c_i are produced by our PNPic finite element package as linear finite element functions. Their partial derivatives become piecewise constant functions, which are discontinuous across the four faces of each tetrahedron. This may cause large errors in the calculation of currents since from the PNPic definition it can be seen that the partial derivatives (or gradient vectors) of u and c_i are continuous. In this work, we overcame the cross section mesh generation difficulty by using the visualization tool ParaView.[26] We then deal with the discontinuous problem in two steps.

In Step 1, we use the Slotboom variable transformation $c_i=e^{-Z_iu}\bar{c}_i$ to transform J_i as

$$J_i=e^{-Z_iu}\frac{\partial\bar{c}_i(\mathbf{s})}{\partial z},\quad i=1,2,...,n,\tag{19}$$

where $\bar{c}_i$ denotes the i-th Slotboom variable, which has been calculated and saved during a search for a PNPic finite element solution.[20,24] Because of the transformation (19), we now only need to calculate the partial derivative $\frac{\partial\bar{c}_i(\mathbf{s})}{\partial z}$. Since $\bar{c}_i$ is much smoother than u, by (19), we can significantly reduce the errors of computing J_i.

In Step 2, we calculate J_i, indirectly, as a solution, φ_i, of a variational problem as follows: Find $\varphi_i\in V_0$ such that

$$\int_{D_{s,h}}\varphi_iv d\mathbf{r}=\int_{D_{s,h}}e^{-Z_iu}\frac{\partial\bar{c}_i(\mathbf{s})}{\partial z}v d\mathbf{r}\quad\forall v\in V_0,\quad i=,2,...,n.$$

Clearly, φ_i is continuous over D_s since it belongs to the finite element function space V_0. Hence, the continuity of J_i is retained by setting $J_i=\varphi_i$ for $i=1, 2, ..., n$.

After obtaining φ_i, we calculate the surface integral $I_{i,S}$ by

$$I_{i,S}=-\frac{e_cN_A}{10^3}Z_iD_{i,p}\int_S\varphi_i(\mathbf{s})d\mathbf{s}\approx-\frac{e_cN_A}{10^3}Z_iD_{i,p}\sum_{\mathcal{T}\in S_h}\int_{\mathcal{T}}\varphi_i(x,y,\widehat{z})dxdy,\tag{20}$$

where S_h denotes a triangular mesh of the cross section S, $\mathcal{T}$ is a triangle element, and $\widehat{z}$ is the location number of S on the z-axis. There exist many quadratures for computing triangular integrals (see Reference 27 for example). We can properly select one according to the accuracy requirement.

To further raise the numerical accuracy of currents, we select m cross sessions S_j of an ion channel pore to produce m current values I_{i,S_j} for $j=1, 2, ..., m$. We then use their averages to re-define the currents I_i of species i and total current I as follows:

$$I_i=\frac{1}{m}\sum_{j=1}^{m}I_{i,S_j}\ \text{ for } i=1,2,...,n,\quad I=\sum_{i=1}^{n}I_i,\quad i=1,2,...,n.\tag{21}$$

We implemented our current calculation scheme in Python as a part of our PNPic finite element package based on the state-of-the-

art finite element library from the FEniCS project.[28] To this end, we derived a tool for computing the electric currents I_i and I of (21). This tool enables us to validate our PNPic model and package by using experimental data available in the literature.

3 | NUMERICAL RESULTS AND DISCUSSIONS

We did numerical tests on an ion channel protein, Gramicidin A (gA),[29] in a salt solution of up to four ionic species by our PNPic software package and our recent software package reported in Reference.20 In these numerical tests, we used the bulk diffusion constants $D_i^b = 0.203, 0.196, 0.133, 0.0793$, and 0.19 for Cl^-, K^+, Na^+, Ca^{2+}, and NO_3^- from the website https://www.aqion.de/site/194. Within the channel pore, we set the diffusion constant $D_i^p = \theta D_i^b$ with $\theta = 0.055$ and the buffer region parameter $\widehat{h} = 3$. These diffusion constants were used to construct the diffusion coefficient function $D_i(\mathbf{r})$ of (10). In addition, we used a boundary value function g of (6b) from Reference 30 as follows:

$$g(\mathbf{s}) = \begin{cases} g_b(\mathbf{s}), & \mathbf{s} \in \Gamma_D \text{ with } z = L_{z_1} \text{ (bottom surface of } \Omega), \\ g_t(\mathbf{s}), & \mathbf{s} \in \Gamma_D \text{ with } z = L_{z_2} \text{ (top surface of } \Omega), \\ g_b(\mathbf{s}) + k(\mathbf{s})[g_t(\mathbf{s}) - g_b(\mathbf{s})], & \mathbf{s} \in \Gamma_N \text{ (the four side surfaces of } \Omega), \end{cases}$$

where $\mathbf{s} = (x,y,z)$, g_b and g_t are two potential surface functions defined on the bottom and top surfaces of the box domain Ω, respectively, and $k(\mathbf{s})$ is a linear interpolation function defined by

$$k(\mathbf{s}) = \frac{z - L_{z_1}}{L_{z_2} - L_{z_1}}.$$

In numerical tests, we set $g_t = -2$ and $g_b = 2$. We also fixed the boundary value functions $g_i = c_i^b$ for $i = 1, 2, \ldots, n$ and the permittivity constants $\epsilon_p = 2$, $\epsilon_m = 2$, and $\epsilon_s = 80$.

An ion channel protein, a gramicidin A (gA), is a natural antibiotic peptide with 15 amino acids (552 atoms) forming a cation-permeable channel pore as shown in Figure 4. The molecular structure and cation selectivity property of gA have been known,[31,32] making gA a valuable test case for our studies.

Using the PQR file of gA from Reference 30, we constructed three box domain meshes Ω_h and three related solvent domain meshes $D_{s,h}$. The membrane location numbers $Z1$ and $Z2$ were set as

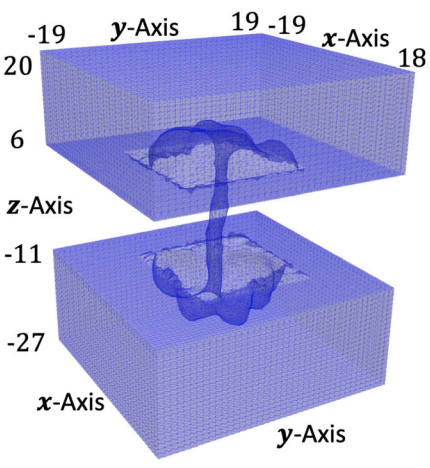
$$Z1 = -11, \quad Z2 = 6.$$

The dimensions and mesh data of these meshes are listed in Table 1. Two views of a solvent domain mesh $D_{s,h}$ extracted from Mesh 1 are displayed in Figure 5, showing a complicated geometrical shape of a solvent domain D_s. One view of Mesh 1 (an irregular interface fitted tetrahedral mesh) has been displayed in Figure 2. These meshes were used in our numerical tests.

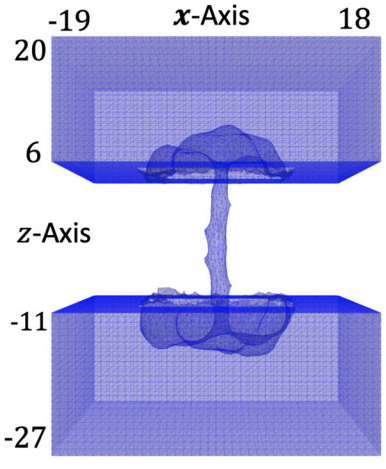
To help understand the influence of electrostatics on ionic concentrations, we split the electrostatic potential u into a positive potential, u^+, and a negative potential, u^-, as follows:

TABLE 1 Box domain dimensions and mesh data for the meshes used in numerical tests

Box mesh Ω_h	Dimensions of box domain $\Omega[L_{x_1}, L_{x_2}; L_{y_1}, L_{y_2}; L_{z_1}, L_{z_2}]$	Number of vertices		Number of tetrahedra	
		Ω_h	$D_{s,h}$	Ω_h	$D_{s,h}$
Mesh 1	[−19, 18; −19, 19; −27, 20]	68,614	40,438	385,740	182,761
Mesh 2	[−39, 38; −39, 39; −27, 20]	83,272	51,803	479,678	252,236
Mesh 3	[−59, 58; −59, 59; −27, 20]	114,001	65,149	671,009	332,121



(A) A solvent region mesh $D_{s,h}$



(B) Side view of $D_{s,h}$

FIGURE 5 Two views of an irregular tetrahedral mesh of solvent domain D_s extracted from mesh 1 for the gA ion channel protein. Here one view of mesh 1 has been displayed in Figure 2

$$u(\mathbf{r}) = u^+(\mathbf{r}) + u^-(\mathbf{r}) \quad \text{with } u^+(\mathbf{r}) = \frac{u(\mathbf{r}) + |u(\mathbf{r})|}{2} \text{ and } u^-(\mathbf{r}) = \frac{u(\mathbf{r}) - |u(\mathbf{r})|}{2}, \quad \mathbf{r} \in \Omega. \tag{22}$$

This makes us easy to show how u^+ and u^- affect the concentration distribution profiles of anions and cations, respectively.

We used the 2D curve scheme from our previous work[20] to draw the potentials $u^\pm$ and ionic concentrations c_i as two dimensional (2D) curves along the z-axis direction (i.e., a normal direction of the membrane). These 2D curves were plotted in terms of points $\left(z^j, u_j^\pm\right)$ and $\left(z^j, c_i^j\right)$ for $j = 1, 2, \ldots, m$, respectively, with $u_j^\pm$ and c_i^j denoting the average values of $u^\pm$ and c_i over the j-th block mesh B_j of a block partition of a solvent domain mesh $D_{s,h}$ along the z-axis direction and z^j being the j-th partition number of interval $L_{z1} \le z \le L_{z2}$. That is, $u_j^\pm$ and c_i^j are defined by

$$u_j^\pm = \frac{1}{V_{B_j}} \int_{B_j} u^\pm(\mathbf{r}) d\mathbf{r}, \quad c_i^j = \frac{1}{V_{B_j}} \int_{B_j} c_i^j(\mathbf{r}) d\mathbf{r}, \quad j = 1,2,\ldots,m \text{ for } i = 1,2,\ldots,n,$$

where V_{B_j} denotes the volume of B_j. We used the same block partition with $m = 50$ and $\overline{h} = 4$ for all our 2D curves. Here $\overline{h}$ is a parameter for controlling the volume of B_j.

Using the same block partition, we also drew a difference function between two functions, say f_1 and f_2, as a 2D curve by using the m points (z^j, d^j) with d^j being defined by

$$d^j = \int_{B_j} |f_1(\mathbf{r}) - f_2(\mathbf{r})| d\mathbf{r}, \quad j = 1,2,\ldots,m. \tag{23}$$

For example, f_1 and f_2 can be the two potential functions generated by our PNPic finite element package using two different boundary value conditions, respectively.

Since our 2D curve is easy to view and can be a good approximation to the distribution profile of a 3D function, it is particularly valuable for us to carry out numerical studies.

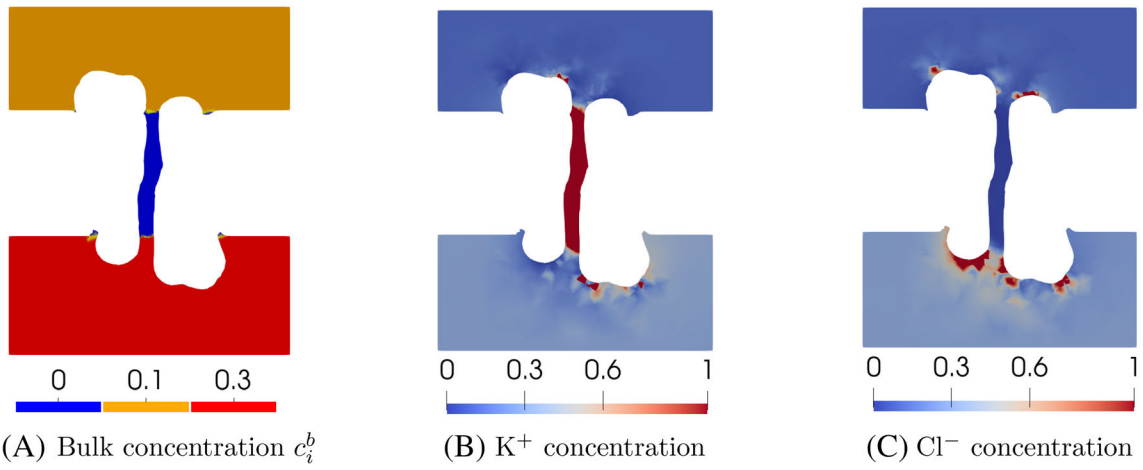
We used five different ionic solvents for our numerical tests as listed below:

1. **Solvent 1**: A solvent of 0.5 mol KCl in $D_{s,b}$ and $D_{s,t}$.
2. **Solvent 2**: A solvent of 0.3 mol KCl in $D_{s,b}$ and 0.1 mol KCl in $D_{s,t}$.
3. **Solvent 3**: A mixture of 0.2 mol NaCl and 0.1 mol KNO_3 in $D_{s,b}$ only.
4. **Solvent 4**: A mixture of 0.1 mol NaCl and 0.1 mol $CaCl_2$ in $D_{s,b}$ and $D_{s,t}$.
5. **Solvent 5**: A mixture of 0.2 mol NaCl and 0.1 mol KNO_3 (potassium nitrate) in $D_{s,b}$ and $D_{s,t}$.

Here the solvent domain D_s has been split by (8) into the bottom portion $D_{s,b}$, top portion $D_{s,t}$, and pore portion $D_{s,p}$ as illustrated in Figure 3. Solvent 1 was applied to the test cases in Table 2 and Solvent 2 in Figures 6, 8–10. Solvent 3 (a mixture) was used for producing the test results reported in Figure 7 and Table 3 while Solvents 4 and 5 (other two mixtures) in Figures 11 and 12, respectively. We have set $c_{i,p}^b = 0$ in all the tests.

Mesh	Calculate G & ∇G	Solve (16) for Ψ	Solve (17) for $\tilde{\Phi}$ & c_i	Total CPU time
Mesh 1	0.21 s	1.76 s	3.395 min	4.238 min
Mesh 2	0.26	2.46	5.218	7.052
Mesh 3	0.35	3.88	7.879	10.361

TABLE 2 Computer CPU times spent by our PNPic finite element package for gA in solvent 1 based on the three meshes reported in Table 1. Here $g_b = 2$, $g_t = -2$, and $\sigma_b = \sigma_t = 0$



(A) Bulk concentration c_i^b (B) K^+ concentration (C) Cl^- concentration

FIGURE 6 A comparison of the bulk concentrations c_i^b for $i = 1, 2$ with the concentrations of K^+ and cl^- generated by the PNPic finite element package for gA in solvent 2. Here c_i^b is a piecewise constant functions defined in (9), $c_{i,b}^b = 0.3$, $c_{i,t}^b = 0.1$, $c_{i,p}^b = 0$, $g_b = 2$, $g_t = -2$, and $\sigma_b = \sigma_t = 0$

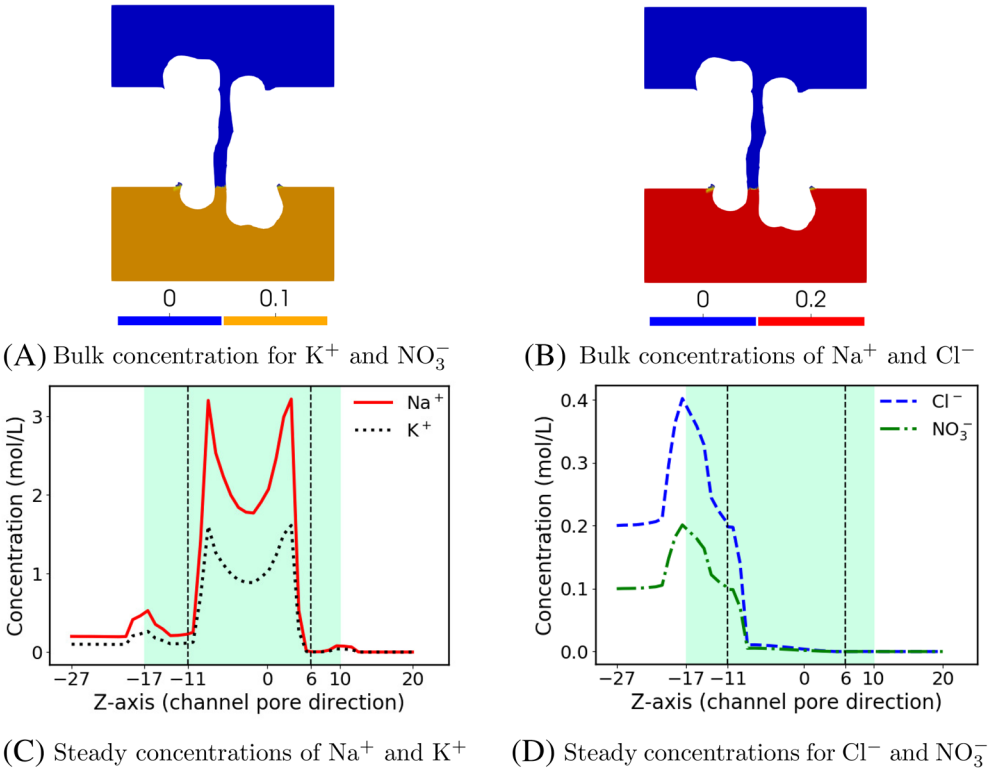
3.1 | Performance of our PNPic finite element package

We did numerical tests by our PNPic finite element package for gA in Solvent 1 based on Meshes 1, 2, and 3 to demonstrate the performance of our PNPic finite element package in computer CPU time. These numerical tests were done on our iMac computer with one 4.2 GHz Intel core i7 processor and 64 GB memory. Here $\sigma_b = \sigma_t = 0$; the nonlinear variational system (17) was solved by an efficient relaxation iterative algorithm with the relaxation parameter $\omega = 1$ and the iteration error tolerance being 10^{-5}; and each related linear system was solved by a generalized minimal residual method using incomplete LU preconditioning with the absolute and relative residual error tolerances being set as 10^{-7}. See References 20,24 for the details of the relaxation iterative algorithm. In each mesh case, the iteration error was quickly reduced from the initial error $O(10^2)$ to $O(10^{-6})$ in only 18 iterations, indicating that our relaxation iterative algorithm for solving (17) has a fast rate of convergence independent of the mesh size.

Table 2 lists the CPU times spent on the three major computing parts as well as the total CPU time for gA in Solvent 1, showing a good performance of our PNPic finite element package.

3.2 | Bulk concentrations versus steady concentrations

Figure 6(A) displays a color mapping of a bulk concentration function, c_i^b, of Solvent 2 on a cross section defined by $y = 0$, showing that the bulk concentrations c_1^b and c_2^b of cations K^+ and anions Cl^-, respectively, are two piecewise constant functions of (9) with $c_{i,b}^b = 0.3$, $c_{i,t}^b = 0.1$, and $c_{i,p}^b = 0$ for $i = 1, 2$ (i.e., $n = 2$). That is, we added 0.3 mol



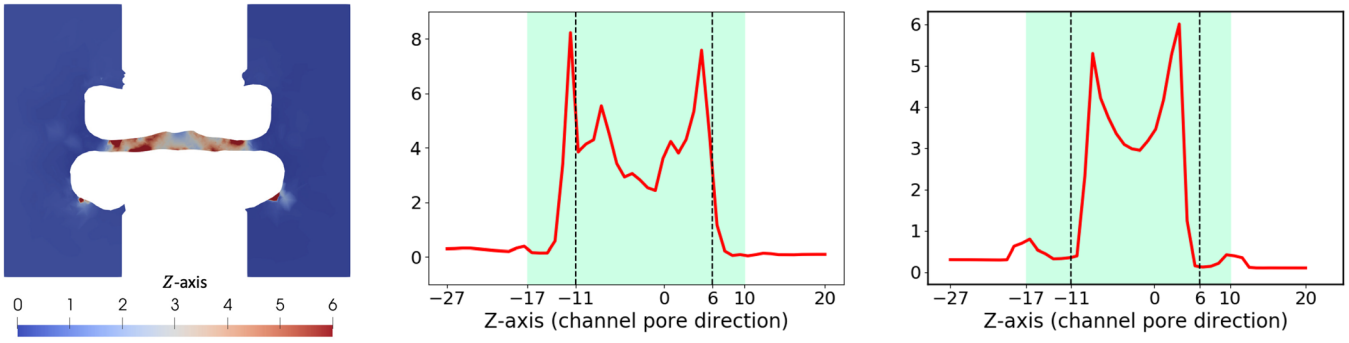
(A) Bulk concentration for K^+ and NO_3^- (B) Bulk concentrations of Na^+ and Cl^-

(C) Steady concentrations of Na^+ and K^+ (D) Steady concentrations for Cl^- and NO_3^-

FIGURE 7 A comparison of the bulk concentrations with the steady concentrations of cations Na^+ and K^+ and anions Cl^- and NO_3^-—The concentrations generated by the PNPic finite element package for gA in solvent 3. Here $g_b = g_t = 0$ and $\sigma_b = \sigma_t = 0$

TABLE 3 A comparison of electric currents I_{i,S_j} and I_{S_j} of (18) on six cross sections S_j with the average currents I_i and I of (21) for gA in solvent 3 (a mixture solution of four ionic species, two anions Cl^- and NO_3^- and two cations Na^+ and K^+) under 100 mV voltage on mesh 1

Cross section S	$I_{1,S}$ for Cl^-	$I_{2,S}$ for NO_3^-	$I_{3,S}$ for Na^+	$I_{4,S}$ for K^+	I_S
S_1	1.73×10^{-4}	8.08×10^{-5}	1.77	1.31	3.08
S_2	1.74×10^{-4}	8.16×10^{-5}	1.78	1.31	3.09
S_3	1.73×10^{-4}	8.10×10^{-5}	1.82	1.34	3.16
S_4	1.72×10^{-4}	8.04×10^{-5}	1.80	1.33	3.13
S_5	1.79×10^{-4}	8.36×10^{-5}	1.76	1.30	3.06
S_6	1.68×10^{-4}	7.85×10^{-5}	1.86	1.37	3.23
Electric currents in average values over the six cross sections S_j					
	I_1 for Cl^-	I_2 for NO_3^-	I_3 for Na^+	I_4 for K^+	I
	1.73×10^{-4}	8.10×10^{-5}	1.80	1.33	3.13



(A) Color mapping at y=0 (B) A currently-used 2D curve (C) Our refined 2D curve

FIGURE 8 A comparison of our 2D curve with a color mapping and a currently-used 2D curve for a concentration of K^+ reported in Figure 9 (A). Here the color mapping visualizes the concentration values on a cross section ($y = 0$) and the currently-used 2D curve is produced by the concentration values on a segment ($-27 \le z \le 20$) of the z-axis

KCl to the bottom portion $D_{s,b}$ and 0.1 mol KCl to the top portion $D_{s,t}$ while not adding any salt to the channel pore portion $D_{s,p}$, resulting in Solvent 2. Figure 6(B,C) also give the color mappings of the steady concentrations — the concentrations of K^+ and Cl^- generated by the PNPic finite element package for gA in Solvent 2 on the cross section defined by $y = 0$. Here none of membrane charges were considered (i.e., $\sigma_b = \sigma_t = 0$). From a comparison of the bulk concentrations with the steady concentrations it can be seen that the bulk concentrations were changed completely in the steady status — most cations K^+ were attracted to the central channel pore $D_{s,p}$ from the both sides of membrane while none of anions entered the pore portion $D_{s,p}$, confirming the cation selectivity property of gA.

To further explore the cation selectivity property of gA, we made numerical tests for gA in Solvent 3. That is, the bulk concentrations c_i^b were set as piecewise constant functions defined in (9) with $c_{1,b}^b = 0.2$ for Na^+, $c_{2,b}^b = 0.2$ for Cl^-, $c_{3,b}^b = 0.1$ for K^+, and $c_{4,b}^b = 0.1$ for NO_3^- while $c_{i,t}^b = 0$ and $c_{i,p}^b = 0$ for $i = 1, 2, 3, 4$ (i.e., none of salt were added to the top and channel pore portions $D_{s,t}$ and $D_{s,p}$). We also set $g_b = g_t = 0$ and $\sigma_b = \sigma_t = 0$ without considering any external voltage and membrane charge. These bulk concentrations are displayed in Figure 7(A,B) as comparisons to the concentrations of two cations Na^+ and K^+ and two anions Cl^- and NO_3^- generated by the PNPic package for gA in Solvent 3.

Figure 7 shows that most anions Cl^- and NO_3^- remained in the bottom portion $D_{s,b}$ while most cation species Na^+ and K^+ entered the central pore region D_{cp} and some of them entered the top portions $D_{s,t}$ of D_s through the ion channel pore. This numerical test further confirms the cation selectivity property of gA.

3.3 | A comparison of our 2D curve with a color mapping and another 2D curve

As an example, in Figure 8, we compare our 2D curve with a color mapping and a currently-used 2D curve (i.e., a curve produced by the values of a 3D function on a line segment, which is commonly used in the literature (see Reference 30 for example)) for a concentration c_K of cations K^+ reported in Figure 9(A). Here the color mapping visualizes the values of c_K on a cross section defined by $y = 0$ and the currently-used 2D curve displays the values of c_K on a segment ($-27 \le z \le 20$) of the z-axis. Thus, more cross sections and line segments are needed to display other values of c_K. In contrast, our 2D curve reflects a global distribution profile of c_K across the membrane in the z-axis direction—the normal direction of membrane since each point represents an average value of c_K over a block partition of a solvent domain D_s in the z-axis direction. Hence, it is particularly valuable in our numerical studies.

3.4 | Effects of membrane charges

To study the effect of membrane charges, we did numerical tests for gA in Solvent 2 by using three different membrane surface charge density functions defined in (7) with $(\sigma_b, \sigma_t) = (-20, -20)$, (20, 20), and (20, −20), respectively. As a comparison, we also repeated the test using $(\sigma_b, \sigma_t) = (0, 0)$, that is, without considering any membrane charge. The test results are reported in Figure 9.

Figure 9 shows that membrane charges have an impact on electrostatics and ionic concentrations. Especially, adding negative charges on one side surface of membrane attracted more cations K^+ because it caused a larger negative potential in magnitude near this membrane surface. And adding positive charges on one side surface of membrane increased positive potential values and attracted more anions Cl^- near this membrane surface. Even so, it is interesting to note that the cation selectivity property of gA is still well retained in the central part of ion channel pore. This indicates that atomic charge within the central portion of ion channel pore can be so strong that membrane charges can not affect the cation selectivity of gA.

3.5 | Effects of boundary value conditions and mesh domain sizes

We used our PNPic finite element package to produce the test cases of using Dirichlet boundary value conditions. We then repeated the

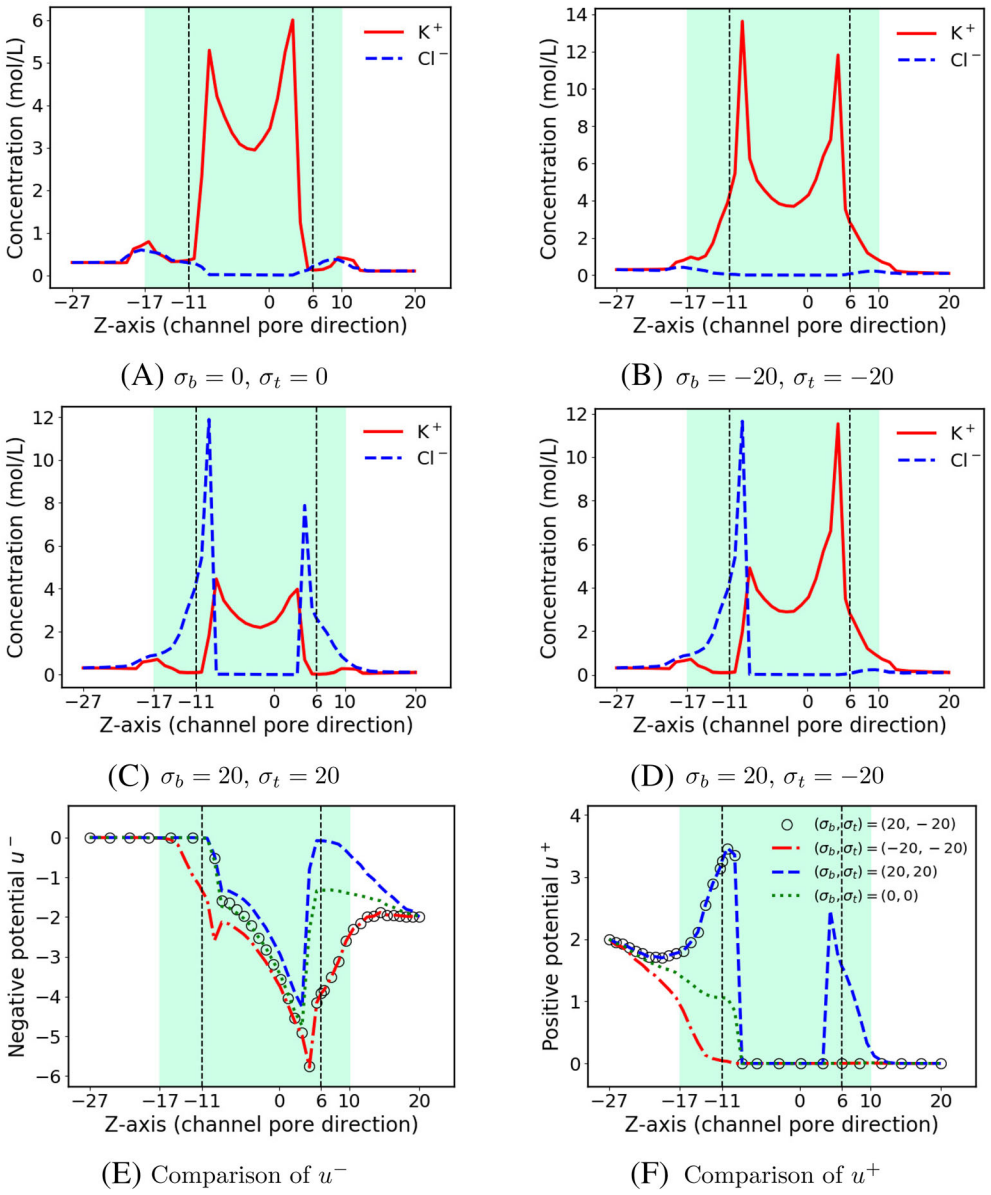


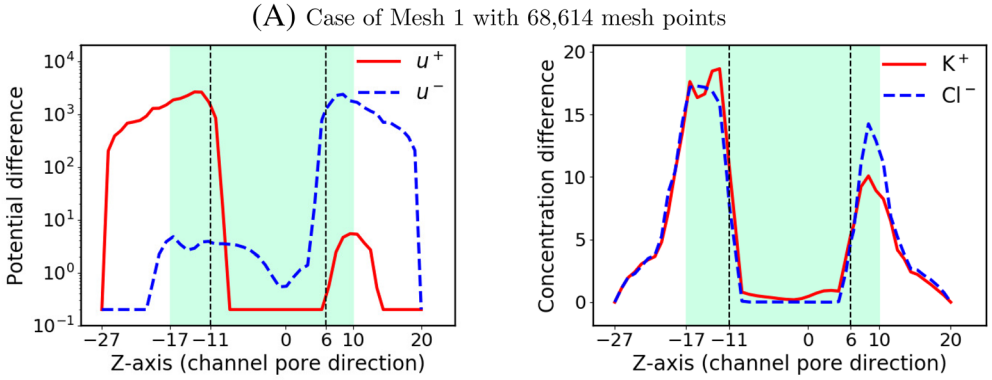
FIGURE 9 Effects of membrane charges on ionic concentrations (a–d) and electrostatic potentials (e, f) for gA in solvent 2. Here σ_b and σ_t are the surface charge densities on the bottom and top surfaces of membrane, respectively; and u^+ and u^- are defined in (22). Here $g_b = 2$ and $g_t = -2$

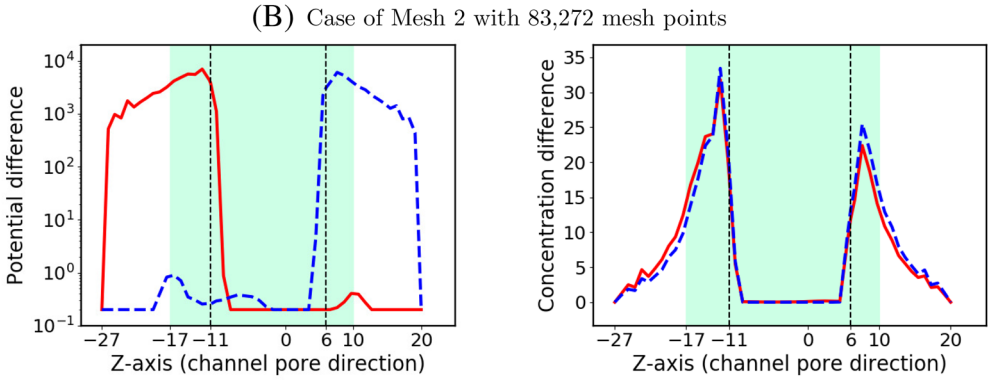
tests using the PNP finite element package reported in Reference 20 to produce the test cases of using Neumann boundary value conditions. The purpose of these tests is to study the effects of boundary value conditions and mesh domain sizes on electrostatics and ionic concentrations. To focus on such a study, we did not consider any membrane charge effect by setting $(\sigma_b,\ \sigma_t) = (0,\ 0)$. We used the meshes given in Table 1 to do these tests to explore the mesh domain size effect. To clearly display the differences, we calculated the difference functions by using the formulas of (23). Here f_1 and f_2 represent the electrostatic potentials $u^{\pm}$ and concentrations c_i generated by our PNPic finite element package and the package from Reference 20, respectively. We then reported these difference functions as 2D curves in Figure 10.

From Figure 10 we can see that the Dirichlet and Neumann boundary value conditions caused different potential and concentration functions mainly outside a central ion channel pore region, D_{cp}, as defined in (11). In other words, the affections of boundary value conditions were little within D_{cp}. It is interesting to note that as the mesh size was increased from 68,614 to 114,001 mesh points, these differences became larger. Even so, the differences remained very small within D_{cp}.

3.6 | Cation selectivity properties of gA under mixture solutions

To explore the cation selectivity properties of gA under mixture solutions, we did numerical tests by the PNPic finite element package for gA in Solvents 3, 4, and 5. The test results of Solvent 3 has been reported and discussed in Figure 7. Hence, we only discuss the test results of Solvents 4 and 5 here, which are reported in Figures 11 and 12, respectively.





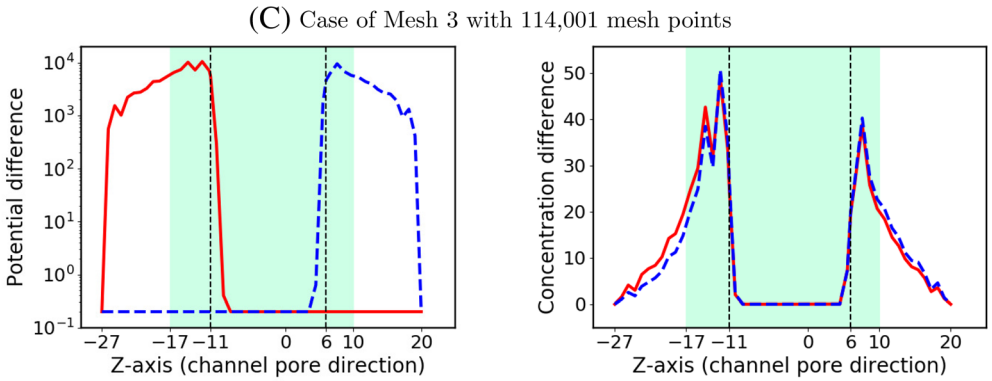


FIGURE 10 Differences between the potentials $u^{\pm}$ and ionic concentrations generated by the Dirichlet boundary conditions and those by the Neumann boundary conditions based on meshes 1, 2, and 3 given in Table 1 for gA in solvent 2. Here $g_b = 2$ and $g_t = -2$

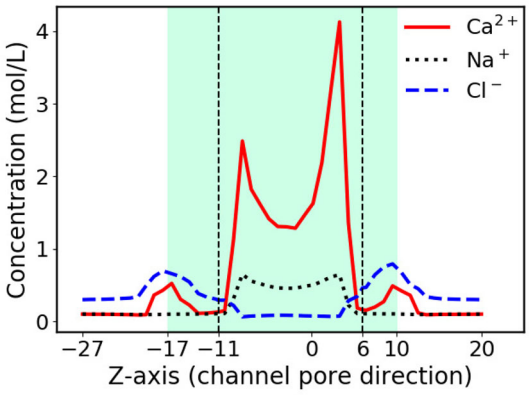


FIGURE 11 Concentrations for gA in solvent 4

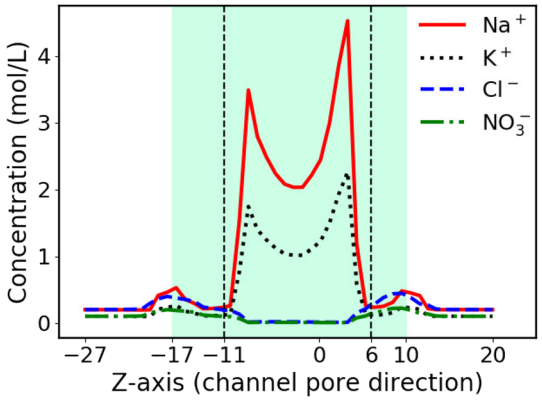


FIGURE 12 Concentrations for gA in solvent 5

Solvent 4 consists of three ionic species: two cations Na^+ and Ca^{2+} and one anion Cl^-. Their bulk concentrations c_i^b were given in mol/L as constants $c_1^b = 0.1$ for cations Na^+, $c_2^b = 0.1$ for cations Ca^{2+}, and $c_2^b = 0.3$ for anions Cl^- over the whole solvent domain D_s. From Figure 11 it can be seen that cations Na^+ and Ca^{2+} are distributed mostly within the central pore region D_{cp} (as defined in (11) with

$\widehat{h} = 3$Å) while anions Cl^- were mostly expelled from D_{cp} in spite of its triple bulk concentration. This test exposes one cation selectivity property of gA under a mixture solution — cations having larger charge numbers can be stronger in competition for space within the central ion channel pore if they have the same bulk concentrations.

In the case of Solvent 5, a finite element solution of our PNPic model gave four ionic concentration functions — two for the cation species Na^+ and K^+ and two for the anion species Cl^- and NO_3^-. The two cation species have the same charge number +1 and the two anion species have the same charge number −1. To distinguish them, the bulk concentrations (0.2 mol/L) of ions Na^+ and Cl^- were set to double those (0.1 mol/L) of ions K^+ and NO_3^- in the whole solvent domain D_s. From Figure 12 it can be seen that more cations Na^+ were attracted into the central ion channel pore region D_{cp} than cations K^+ because of its double bulk concentration. This test exposes another cation selectivity property of gA under a mixture solution—cations having larger bulk concentration can be stronger in competition for space within the ion channel pore if their charge numbers are the same.

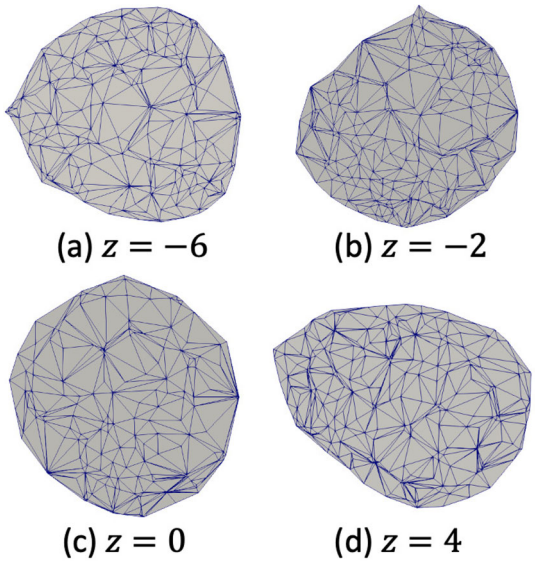


FIGURE 13 Four cross section meshes of gA ion channel pore that were used to calculate the electric current I defined in (21). They were generated by using ParaView[26]

3.7 | Validation by electric current experimental data

We calculated the electric currents using our formulas of (18) and (21) for gA in Solvent 3. In these calculations, we selected the six cross sections S_j of the gA channel pore at $z = -6, -4, -2, 0, 2, 4$ and then generated their meshes by using the visualization tool ParaView.[26] As examples, we display four of them in Figure 13 to demonstrate that different cross sections can have different shapes and meshes. The test results are reported in Table 3. Here we set $g_t = 0$ and $g_b = 3.892$, which yielded a voltage of 100 mV. These currents were calculated by our electric current computational tool from our PNPic finite element package.

From Table 3 it can be seen that the currents of two anions Cl^- and NO_3^- are almost zero, indicating that most anions were blocked by the gA channel pore. While both cations Na^+ and K^+ produced electric currents, the current of Na^+ ions was larger than that of K^+ simply because the bulk concentration of Na^+ doubled that of K^+. The results in Table 2 also show that different cross sections caused different current values. Hence, it is necessary to refine the currents by our average formulas of (21).

Finally, we did more current calculations for gA in a salt solution of cations K^+ and anions Cl^- to further validate our PNPic model and software package by electric current experimental data. In these

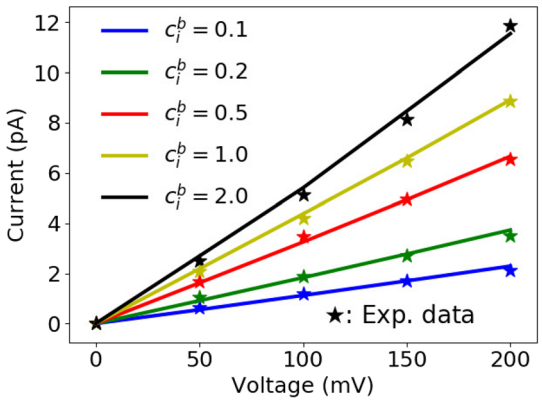


FIGURE 14 Five I-V curves produced by the current data of Table 4. Here the solid lines stand for our predicted curves and the star markers represent the experimental values from Reference 23

TABLE 4 A comparison of the currents I calculated by our formula (21) with the experimental currents I_{exp} extracted from one figure (Figure 8) of the Reference 23 for gA in a salt solution of c_i^b mol KCl on mesh 1. Here c_i^b is the bulk concentration of D_s (with $c_{i,p}^b = 0$); voltage was set as $g_b - g_t$ in mV with $g_t = 0$; I and I_{exp} were in pA; θ is defined in (12)

Voltage (mV)	$c_i^b = 0.1; \theta = 0.023$		$c_i^b = 0.2; \theta = 0.035$		$c_i^b = 0.5; \theta = 0.055$		$c_i^b = 1.0; \theta = 0.065$		$c_i^b = 2.0; \theta = 0.07$	
	I	I_{exp}	I	I_{exp}	I	I_{exp}	I	I_{exp}	I	I_{exp}
50	0.55	0.65	0.91	1.06	1.62	1.66	2.18	2.08	2.89	2.49
100	1.12	1.20	1.83	1.89	3.25	3.46	4.37	4.18	5.70	5.12
150	1.70	1.71	2.77	2.72	4.93	4.94	6.59	6.49	8.46	8.12
200	2.29	2.12	3.73	3.51	6.66	6.55	8.91	8.86	11.54	11.86

validation tests, we generated five I-V curves by using the six cross section meshes of Table 3 and the voltage $V = 50$ mV, 100 mV, 150 mV, 200 mV. By the formula

$$g_b - g_t = \frac{e_c}{10^3 K_B T} V \approx 0.0389216\ V,$$

we set $g_t = 0$ and got $g_b = 1.946, 3.892, 5.838, 7.784$ for the tests. The diffusion coefficient $D_{i,p}$ was set by (12) with a properly-selected value of θ to fit the experimental data. These values of θ are reported in Table 4. Hee the total electric current I was calculated by our formula (21).

Table 4 gives a comparison of the numerical values of current I calculated by our tool from the PNPic finite element package according to our formula (21) with the experimental values extracted from one figure (Figure 8) of the Reference 23. Using these data, we plotted five I-V (current–voltage) curves and displayed them in Figure 14 to more clearly compare them. From Figure 14 it can be seen that the currents computed by our PNPic finite element package match well the experimental data. These tests further validate our PNPic model and finite element package.

4 | CONCLUSIONS

We have presented an improved Poisson-Nernst-Planck ion channel (PNPic) model and an effective finite element method for solving this model. We then implemented this finite element method in Python and Fortran as a software package, which works for an ion channel protein with a three-dimensional molecular structure and an ionic solvent with multiple species. Compared to other PNPic models available in the current literature, the major improvements made in our PNPic model involve boundary value conditions, a membrane surface charge density, bulk ionic concentrations, and diffusion coefficient functions. These improvements provide us with more options in the selection of membrane charges, bulk concentrations, and diffusion coefficient functions, greatly enhancing our capability to carry out channel studies and simulations.

As initial applications, we used our PNPic finite element package to explore the cation selectivity property of an ion channel protein, a gramicidin A (gA), in this work. In these numerical experiments, the two side surfaces of a membrane were set to have different surface charge densities; different ionic bulk concentrations were assigned to the bottom, top, and pore portions of a solvent domain to mimic different chemical test environments. The cation selectivity property of gA was exposed in five different ionic solvents with up to four ionic species. The derived numerical results have shown that varying membrane charges and bulk concentrations has little influence on the cation selectivity property of gA. They also expose that the cation selectivity property of gA happens within the central ion channel pore only, implying that it is mainly determined by the atomic structure and charges of an ion channel protein within the central pore area.

Moreover, we did comparison studies between our PNPic model and our recent PNP ion channel model reported in Reference 20 for gA based on three mesh domains with different mesh sizes. The purpose of this study is to show how Dirichlet and Neumann boundary value conditions affect ion channel selectivity properties. In our recent PNP model, Neumann boundary value conditions are applied to the four side surfaces of both a box domain and a solvent domain to avoid the difficulties of selecting boundary value functions that we face in the case of using Dirichlet boundary value conditions. Since a PNPic model using Dirichlet boundary value conditions is a basic model, it is important for us to further improve it and develop its effective finite element methods in this work. With the new PNPic package, we now can carry out ion channel studies through selecting different boundary value functions. We also can do the comparison studies with other models like what we did with the model of Reference 20 in this paper, from which we observed that either Neumann or Dirichlet boundary value conditions have little influence on the cation selectivity property of gA even though yielding significantly different electrostatic potentials and ionic concentrations near the related boundaries. During these studies, we also found that box domain sizes have little influence on the cation selectivity property of gA.

One important application of a PNP model is to calculate the electric current within an ion channel pore in the steady status. To improve the numerical accuracy of current calculation, we have developed an effective numerical scheme for computing the electric current by using advanced mathematical techniques in this paper. One key part of this scheme is an effective quadrature for computing a surface integral defined on a cross section of an ion channel pore. Each electric current value is then calculated as an average of several surface integral values. We implemented this scheme in Python as a part of our PNPic finite element package. As applications, we used this scheme to calculate the currents of four ionic species for gA in a mixture solution of four species. Furthermore, we produced five I-V (current–voltage) curves for gA in a salt solution of two ionic species, which matched well the experimental data from the literature. Consequently, our PNPic model and finite element package were validated not only by the cation selectivity property of gA but also by the physical and chemical experimental data of gA.

Even though many promising numerical results are reported in this paper, more numerical tests are required to do in the future. We plan to do so for other ion channel proteins and other ionic solvents. We will further explore the effects of boundary value conditions, membrane charges, and bulk concentrations on electrostatics and ionic concentrations by using different boundary value functions, different membrane charges, and different bulk concentrations. Moreover, we will do more validation tests on our PNPic model and finite element package by using more chemical experimental data that we could find in the literature. To this end, our PNPic finite element package will become a powerful tool for ion channel studies.

ACKNOWLEDGMENTS

This work was partially supported by the Simons Foundation, USA, through research award 711776.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

ORCID

Dexuan Xie https://orcid.org/0000-0002-7072-6275

REFERENCES

[1] T. Teorell, *Prog. Biophys. Biophys. Chem.* **1953**, *3*, 305.
[2] K. Meyer, J. Sievers, *Helv. Chim. Acta* **1936**, *19*, 649.
[3] J. Manzanares, W. Murphy, S. Mafe, H. Reiss, *J. Phys. Chem.* **1993**, *97*, 8524.
[4] A. Moya, J. Horno, *J. Phys. Chem. B* **1999**, *103*, 10791.
[5] W. Liu, *J. Diff. Equ.* **2009**, *246*, 428.
[6] J. W. Jerome, *SIAM J. Appl. Math.* **1985**, *45*, 565.
[7] M. Mock, *Commun. Pure Appl. Math.* **1972**, *25*, 781.
[8] A. Golovnev, S. Trimper, *Phys. Lett. A* **2010**, *374*, 2886.
[9] A. Golovnev, S. Trimper, *J. Chem. Phys.* **2011**, *134*, 154902.
[10] D. S. Bolintineanu, A. Sayyed-Ahmad, H. T. Davis, Y. N. Kaznessis, *PLoS Comput. Biol.* **2009**, *5*, e1000277.
[11] A. E. Cardenas, R. D. Coalson, M. G. Kurnikova, *Biophys. J.* **2000**, *79*, 80.
[12] B. Lu, M. J. Holst, J. A. McCammon, Y. Zhou, *J. Comput. Phys.* **2010**, *229*, 6979.
[13] Y. Song, Y. Zhang, T. Shen, C. L. Bajaj, J. A. McCammon, N. A. Baker, *Biophys. J.* **2004**, *86*, 2017.
[14] S. Yu, G. Wei, *J. Comput. Phys.* **2007**, *227*, 602.
[15] S. R. Mathur, J. Y. Murthy, *Int. J. Heat Mass Transfer* **2009**, *52*, 4031.
[16] M. Mirzadeh, F. Gibou, *J. Comput. Phys.* **2014**, *274*, 633.
[17] U. Hollerbach, D.-P. Chen, R. S. Eisenberg, *J. Sci.Comput.* **2001**, *16*, 373.
[18] I.-L. Chern, J.-G. Liu, W.-C. Wang, et al., *Meth. Appl. Anal.* **2003**, *10*, 309.
[19] D. Xie, *J. Comput. Phys.* **2014**, *275*, 294.
[20] D. Xie, Z. Chao, *J. Comput. Phys.* **2020**, *423*, 109915.
[21] D. Xie, S. H. Audi, R. K. Dash, *J. Comput. Chem.* **2020**, *41*, 218.
[22] M. Pekker, M. Shneider, *J. Phys. Chem. Biophys.* **2015**, *5*, 177.
[23] C. D. Cole, A. S. Frost, N. Thompson, M. Cotten, T. A. Cross, D. D. Busath, *Biophysical Journal* **2002**, *83*, 1974.
[24] Z. Chao Ph.D. Thesis, University of Wisconsin-Milwaukee, USA 2020.
[25] T. Liu, S. Bai, B. Tu, M. Chen, B. Lu, *Mol. Based Math. Biol.* **2015**, *3*, 2299.
[26] J. Ahrens, B. Geveci, C. Law, in *The Visualization Handbook*, Elsevier Butterworth–Heinemann, Boston **2005**, Ch. 36.
[27] L. Zhang, T. Cui, H. Liu, *J. Comput. Math.* **2009**, *27*, 89.
[28] *Automated Solution of Differential Equations by the Finite Element Method* (Eds: A. Logg, K.-A. Mardal, G. N. Wells), Vol. 84, Lecture Notes in Computational Science and Engineering, Springer Verlag, Berlin, Heidelberg **2012**.
[29] K.-C. Lee, S. Huo, T. Cross, *Biochemistry* **1995**, *34*, 857.
[30] B. Tu, M. Chen, Y. Xie, L. Zhang, B. Eisenberg, B. Lu, *J. Comput. Chem.* **2013**, *34*, 2065.
[31] D. W. Urry in Proc. of Fifth Jerusalem Symp. on Quantum Chemistry and Biochemistry (Ed: Ernst D), 1973, 723. Israel Academic of Sciences Jerusalem.
[32] M. J. Sampson, R. S. Lovell, W. J. Craigen, *J. Biol. Chem.* **1997**, *272*, 18966.

How to cite this article: Z. Chao, D. Xie, *J. Comput. Chem.* **2021**, *42*(27), 1929. https://doi.org/10.1002/jcc.26723