SIAM J. SCI. COMPUT.
Vol. 42, No. 6, pp. B1490–B1516

AN EFFECTIVE FINITE ELEMENT ITERATIVE SOLVER FOR A POISSON–NERNST–PLANCK ION CHANNEL MODEL WITH PERIODIC BOUNDARY CONDITIONS*

DEXUAN XIE† AND BENZHUO LU‡

Abstract. A system of Poisson–Nernst–Planck equations (PNP) is an important dielectric continuum model for simulating ion transport across biological membrane. In this paper, a PNP ion channel model with periodic boundary value conditions, denoted by PNPic, is presented and solved numerically with an effective finite element iterative method. In particular, the periodic boundary value conditions are used to mimic an infinitely large ion channel membrane, and the PNPic finite element solver includes (1) a PNPic solution decomposition scheme for overcoming the singularity difficulty caused by atomic charges, (2) Slotboom variables for transforming each related Nernst–Planck equation to avoid gradient calculation for any electrostatic potential function, (3) an efficient modified Newton iterative algorithm for solving each related nonlinear finite element equation, and (4) communication operators for carrying out functions operations between different finite element function spaces. This effective PNPic solver is implemented as a software package based on the state-of-the-art finite element library from the FEniCS project and an ion channel mesh generation package developed in Lu's group. Numerical results demonstrate the convergence of the PNPic finite element iterative solver and the performance of the PNPic software package. Moreover, the PNPic model is validated by the cation selectivity property and electric current experimental data of an ion channel protein.

Key words. Poisson–Nernst–Planck model, finite element method, ion channel protein, periodic boundary conditions

AMS subject classifications. 92-08, 65N30, 35J66, 65K10

DOI. 10.1137/19M1297099

1. Introduction.

Electrodiffusion describes a diffusion process of charged particles in a self-induced electric field (sometimes together with an external electric field), which widely exists in electrochemistry, biology, nanofluidics, and semiconductor physics, etc. A dielectric continuum implicit solvent model defined by Poisson–Nernst–Planck (PNP) equations has been recognized to have significant advantages in computational efficiency and in the calculation of macroscopic properties (e.g., electric current) for a diffusion process at the mean field level compared to the corresponding explicit solvent model [45, 13, 8, 26]. In the last two decades, many PNP ion channel models were developed through considering volume-exclusion entropy effects [37, 28, 44], hard sphere interactions [4, 17, 18, 32, 44, 43], van der Waals interactions [22], ionic solvation effects [33], electric charge correlations [29], variable dielectric properties [34], and surface energies [51], etc. They were solved numerically by using finite difference methods [14, 15, 26, 27, 54], finite element

*Submitted to the journal's Computational Methods in Science and Engineering section November 1, 2019; accepted for publication (in revised form) September 2, 2020; published electronically December 14, 2020.

https://doi.org/10.1137/19M1297099

Funding: The work of the second author was supported by the National Key Research and Development Program of China through grant 2016YFB0201304, by the Science Challenge Program through grant TZ2016003, and by the National Natural Science Foundation of China through grants NSFC 22073110 and 11771435.

†Corresponding author. Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201-0413 USA (dxie@uwm.edu, http://www.uwm.edu/~dxie/).

‡LSEC, National Center for Mathematics and Interdisciplinary Sciences, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, People's Republic of China (bzlu@lsec.cc.ac.cn).

methods [16, 30, 36, 38, 41, 49], finite volume methods [40], and spectral element methods [21] in either a simplified one-dimensional or a complex three-dimensional setting. Special numerical techniques and implementation strategies were developed to improve the performance of PNP numerical solvers, including a second-order finite difference method [54], a parallel finite element method [49], a potential decomposition technique [36], stabilized techniques [7, 50], energy and mass preservation schemes [14, 15, 20, 27, 41], and mixed finite element methods [16]. Slotboom variable transformation [47] and Gummel's iteration technique [19], developed in the early semiconductor device system simulations, were also used to solve PNP ion channel models [26, 36, 49].

Compared with finite difference and finite volume methods, one major advantage of a finite element method is to be able to approximate a complex geometrical shape of an ion channel protein in a high degree of accuracy due to using an irregular tetrahedral mesh. Indeed, well retaining the geometry of a three-dimensional X-ray crystallographic ion channel molecular structure can significantly raise the quality of a PNP ion channel model. But the generation of an irregular tetrahedral mesh that can fit well a complex ion channel molecular surface is highly technical. In the last ten years, Lu's research team developed an ion channel mesh software package based on the molecular surface triangular mesh package TMSmesh [9, 30, 31]. This mesh package has been released to the public through the cloud computing website https://www.xyzgate.com. As a unique ion channel tetrahedral mesh package, it will be applied to the development of a new PNP ion channel finite element solver in this paper.

Typically, a PNP ion channel model is based on a box domain that is separated into two solvent compartments by a membrane. A single ion channel protein is then embedded centrally in the membrane and acts as the conduct for transporting ions from one solvent compartment to the other. The membrane normal direction and the ion channel pore are set to coincide with the z-axis direction for the simplicity of implementation. To account for the influence of other ion channel proteins on this single ion channel model, it is natural to set periodic boundary value conditions on the four side surfaces of the box. In fact, periodic boundary techniques have been routinely applied to molecular dynamics for a protein simulation in a box of water molecules. They were also applied to the construction of Poisson–Boltzmann (PB) ion channel models [5, 24] and a finite difference PNP solver [23]. Even so, they have not been considered in any PNP finite element solver yet since it is very difficult to develop a PNP ion channel finite element solver even in the case that does not consider any periodic boundary. In this paper, we attempt to develop an improved PNP ion channel model using the periodic boundary value conditions that are different from those used in [5, 24]. In fact, the periodic boundary conditions in [5] are set on the boundary of a box domain as if one side surface is adjacent to the opposing side surface, while the periodic boundary value conditions in [24] are constructed by setting the mesh nodes of two opposite side surfaces to have the same labeling numbers on the four side surfaces of the box. In our periodic boundary value conditions, each PNP unknown function is set to have the same values on the two opposite side surfaces as done commonly in a periodic boundary value problem.

Another major difficulty in solving a PNP ion channel model comes from the solution singularity caused by atomic charges. As shown in [53, Figure 3], such a difficulty cannot be overcome unless all the singularity points can be isolated by a solution decomposition scheme. Two different solution decomposition schemes were reported in [11, 52], respectively, to overcome this difficulty in the numerical solution of a PB model for a protein surrounded by an ionic solvent. We recall that in [11], a PB

unknown function, u, which gives an electrostatic potential density of the electric field, is split into three component functions, u^s, u^h, and u^r, within a protein region D_p only, resulting in a Laplace boundary value problem of u_h in D_p and a nonlinear interface boundary value problem of u^r in the box domain Ω. Since D_p is a strongly nonconvex domain with a complicated nonsmooth boundary (i.e., a molecular surface), especially for an ion channel protein, solving such a Laplace boundary value problem may cause problems in solution accuracy and solution regularity. The equation of u^r is also difficult to solve due to involving a jumpily discontinuous flux interface condition on the interface between D_p and a solvent region D_s. In contrast, in [52], u is split into three component functions, G, Ψ, and $\tilde{\Phi}$, over the box domain Ω such that G, Ψ, and $\tilde{\Phi}$ represent the electrostatic potentials induced by the atomic charges, the potentials from the interfaces and boundary, and the ionic charges from a solvent region, D_s, respectively. Since G contains all the singularity points of u, both Ψ and $\tilde{\Phi}$ become smooth within the solvent and solute regions. Note that $u^r = u$ within D_s, and $u = G + \Psi + \tilde{\Phi}$. Hence, $\tilde{\Phi} = u^r - G - \Psi$. This shows that $\tilde{\Phi}$ does not involve any tough part of u^r from G and Ψ so that it is much smoother than u^r. As a result, the interface boundary value problem of $\tilde{\Phi}$ does not involve any jumpily discontinuous flux interface condition and can be much easier to solve numerically than that of u^r. It is this splitting scheme that leads to an efficient PB finite element solver in [52]. The splitting scheme from [11] has been adapted to construct a PNP finite difference solver in [54] and a PNP finite element solver in [49]. In this paper, we will adapt the splitting scheme from [52] to construct a new finite element PNP ion channel solver subject to periodic boundary constraints.

In order to reduce numerical complexity and computer memory requirement sharply, a PNP iterative scheme is often constructed by classic successive relaxation iterative techniques [42] (or related Gummel's iterative technique [19]). In such a scheme, however, each Nernst–Planck equation of a PNP system is modified as an equation that requires calculating the gradient of a given potential function. From the finite element theory, it is known that a gradient calculation may decay one degree of a finite element solution accuracy [6]. To avoid such a potential numerical problem, the Slotboom variables, introduced in [47], can be used to transform each related Nernst–Planck equation as the one that does not involve any gradient of a potential function, but on the other hand, the related linear Poisson dielectric equation is transformed as a strongly nonlinear equation. Consequently, how to solve such a nonlinear equation becomes a key step in the development of an effective PNP numerical solver. Hence, one important task of this paper is to develop new numerical techniques for solving each related nonlinear equation efficiently.

A system of PNP finite element equations involves ionic concentration functions c_i and an electrostatic potential function u that belong to two different finite element function spaces, respectively. A communication operator is thus required to carry out function operations between these two spaces. Currently, such a function operation issue was simply addressed by extending each c_i from D_s to Ω through setting the values of c_i to be zero at the mesh nodes outside the solvent region D_s so that both c_i and u are defined on the same finite element function space based on a mesh of Ω. But this simple treatment may decay the accuracy of a PNP finite element system significantly since it actually causes c_i to be nonzero outside D_s on a layer of tetrahedra along the interface between D_s and a protein-membrane region. Under periodic boundary constraints, each of these two spaces is modified as a space with a reduced dimensionality, further increasing the difficulty of dealing with this issue.

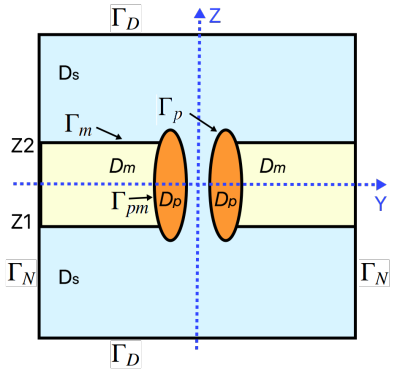


FIG. 1. *An illustration of the region partition* (2.2) *of a rectangular box domain* Ω.

In this paper, we will directly construct a finite element function space for each ionic concentration function c_i based on an irregular tetrahedral mesh of D_s. We then derive all the required communication operators so that we can well retain the accuracy of a PNP finite element system in the implementation of function operations between different function spaces.

The rest of the paper is organized as follows. In section 2, we present a PNP ion channel model using periodic boundary value conditions (denoted by PNPic). In section 3, we present a PNPic solution decomposition. In section 4, we reformulate each equation of the PNPic solution decomposition into a variational problem. In section 5, we describe the construction of our PNPic finite element solver. In section 6, we report our PNPic software package and numerical results to demonstrate the convergence and performance of our PNPic finite element iterative solver and to validate our PNPic software package, along with two new formulas for estimating the distribution of ions and electric current within an ion channel pore. Finally, conclusions are made in section 7.

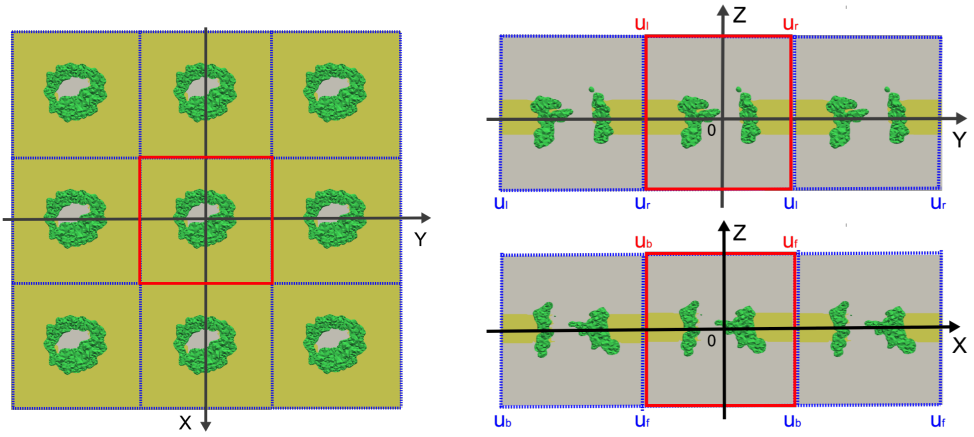
2. A PNP ion channel model with periodic boundary value conditions. We construct a sufficiently large open box domain, Ω, by

$$\Omega = \{(x, y, z) \mid L_{x1} < x < L_{x2},\ L_{y1} < y < L_{y2},\ L_{z1} < z < L_{z2}\ \}, \tag{2.1}$$

and partition it and its boundary $\partial\Omega$, as illustrated in Figure 1, as follows:

$$\Omega = D_p \cup D_m \cup D_s \cup \Gamma_m \cup \Gamma_p \cup \Gamma_{pm}, \quad \partial\Omega = \Gamma_D \cup \Gamma_N, \tag{2.2}$$

where $L_{x1}, L_{x2}, L_{y1}, L_{y2}, L_{z1}$, and L_{z2} are real numbers; D_p, D_m, and D_s denote an ion channel protein region, a membrane region, and a solvent region, respectively; Γ_m denotes the interface between D_m and D_s, Γ_p the interface between D_p and D_s, and Γ_{pm} the interface between D_p and D_m; and Γ_D consists of the bottom and top surfaces of the box domain Ω and Γ_N the four side surfaces of Ω. In Figure 1, $Z1$ and $Z2$ set the location of the membrane, D_s contains an ionic solvent with n ionic species, and D_p hosts an ion channel protein with n_p atoms. We have set the normal direction of the membrane in the z-axis direction and the z-axis to pass the channel pore. Moreover, the position vector $\mathbf{r}_j$ and charge number z_j of atom j are given from a three-dimensional X-ray crystallographic molecular structure of the ion channel protein. The bulk concentration c_i^b and charge number Z_i of species i are also given for the ionic solvent.



(a) Top view of membrane (in z-direction) (b) Two side views of membrane

FIG. 2. (a) *A membrane embedded with many ion channel proteins of the same type.* (b) *An illustration of the periodic boundary value conditions of a function u. Here the box domain for simulation is colored in red; u_l, u_r, u_f, and u_b denote the boundary values of u on the left, right, front, and back surfaces of each box domain, respectively; ion channel proteins are colored in green; and the membrane is colored in yellow.*

Based on the dielectric continuum approach, the three regions D_p, D_m, and D_s are treated as dielectric media with permittivity constants ϵ_p, ϵ_m, and ϵ_s, respectively. Since D_m consists of a double layer of phospholipid, cholesterol, and glycolipid molecules whereas D_p is composed of amino acids, ϵ_m may be greater than ϵ_p [48, 24].

We can duplicate the box domain Ω in the four side surface directions, as illustrated in Figure 2(a), to produce an infinitely large membrane that is embedded with ion channel proteins of the same type. Since a dimensionless electrostatic potential function, u, on each box is identical to each other, it satisfies the periodic boundary value conditions, $u_l = u_r$ and $u_b = u_f$, as illustrated in Figure 2(b). Here u_l, u_r, u_b, and u_f, respectively, denote the values of u on the left, right, back, and front side surfaces of the simulation box Ω, which is marked in red to differ from its neighboring boxes (in blue color). Hence, for a function, $u(t, \mathbf{r})$, of time t and spatial variable $\mathbf{r}$ with $\mathbf{r} = (x, y, z) \in \Omega$, we obtain periodic boundary value conditions as follows:

$$\begin{aligned} u(t, L_{x1}, y, z) &= u(t, L_{x2}, y, z), \qquad (y, z) \in D_1, \\ u(t, x, L_{y1}, z) &= u(t, x, L_{y2}, z), \qquad (x, z) \in D_2, \end{aligned} \tag{2.3}$$

where $D_1 = \{(y, z) \mid L_{y1} < y < L_{y2},\ L_{z1} < z < L_{z2}\}$, $D_2 = \{(x, z) \mid L_{x1} < x < L_{x2},\ L_{z1} < z < L_{z2}\}$. Similarly, we can obtain the periodic boundary value conditions for an ionic concentration function, $c_i(t, \mathbf{r})$ for $\mathbf{r} \in D_s$ and $t \geq 0$, of species i on the four side surface $\Gamma_N \cap \partial D_s$ of D_s. Here ∂D_s denotes the boundary of D_s.

Our PNP ion channel model using the above periodic boundary value conditions, which is denoted as PNPic, consists of the Poisson equations

$$-\epsilon_p \Delta u(t, \mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, \quad \mathbf{r} \in D_p, \tag{2.4}$$

$$-\epsilon_m \Delta u(t, \mathbf{r}) = 0,\ \mathbf{r} \in D_m, \quad -\epsilon_s \Delta u(t, \mathbf{r}) = \beta \sum_{i=1}^{n} Z_i c_i(t, \mathbf{r}),\ \mathbf{r} \in D_s,$$

and the Nernst–Planck equations

$$\frac{\partial c_i(t,\mathbf{r})}{\partial t} = \nabla\cdot\mathcal{D}_i\left[\nabla c_i(t,\mathbf{r}) + Z_i c_i(t,\mathbf{r})\nabla u(t,\mathbf{r})\right], \quad \mathbf{r}\in D_s,\ t>0, \tag{2.5}$$

for $i = 1, 2, \ldots, n$, subject to the following interface conditions, initial value conditions, and boundary value conditions:

- Interface conditions:

$$\begin{aligned} &u(t,\mathbf{s}^-) = u(t,\mathbf{s}^+), \quad \epsilon_p \tfrac{\partial u(t,\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_s \tfrac{\partial u(t,\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, \quad \mathbf{s}\in\Gamma_p,\\ &u(t,\mathbf{s}^-) = u(t,\mathbf{s}^+), \quad \epsilon_m \tfrac{\partial u(t,\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_s \tfrac{\partial u(t,\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})}, \quad \mathbf{s}\in\Gamma_m,\\ &u(t,\mathbf{s}^-) = u(t,\mathbf{s}^+), \quad \epsilon_p \tfrac{\partial u(t,\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \tfrac{\partial u(t,\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, \quad \mathbf{s}\in\Gamma_{pm}. \end{aligned} \tag{2.6}$$

- Initial value conditions:

$$c_i(0,\mathbf{r}) = c_i^0(\mathbf{r}), \quad \mathbf{r}\in D_s, \quad i = 1, 2, \ldots, n. \tag{2.7}$$

- Dirichlet boundary value conditions on the bottom and top surfaces:

$$u(t,\mathbf{s}) = g(\mathbf{s}), \quad \mathbf{s}\in\Gamma_D, \quad c_i(t,\mathbf{s}) = g_i(\mathbf{s}), \quad \mathbf{s}\in\Gamma_D. \tag{2.8}$$

- Periodic boundary value conditions on the four side surfaces:

$$u(t,\mathbf{s}) \text{ is periodic for } \mathbf{s}\in\Gamma_N,\ c_i(t,\mathbf{s}) \text{ is periodic for } \mathbf{s}\in\Gamma_N\cap\partial D_s. \tag{2.9}$$

- Robin boundary value conditions on the interface $\Gamma_p\cup\Gamma_m$:

$$\frac{\partial c_i(t,\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(t,\mathbf{s})\frac{\partial u(t,\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = 0, \quad \mathbf{s}\in\Gamma_p\cup\Gamma_m. \tag{2.10}$$

Here $\delta_{\mathbf{r}_j}$ is the Dirac delta distribution at $\mathbf{r}_j$; α and β are defined by

$$\alpha = \frac{10^{10}e_c^2}{\epsilon_0 k_B T}, \quad \beta = \frac{N_A e_c^2}{10^{17}\epsilon_0 k_B T}; \tag{2.11}$$

$\mathbf{n}_p$, $\mathbf{n}_m$, and $\mathbf{n}_s$ are the unit outward normal directions of D_p, D_m, and D_s, respectively; g and g_i are boundary value functions; c_i^0 is an initial value function; and $\mathcal{D}_i$ denote a diffusion coefficient function of the ith ionic species. Here ϵ_0 is the permittivity of the vacuum, e_c is the elementary charge, k_B is the Boltzmann constant, T is the absolute temperature, and N_A is the Avogadro number, which estimates the number of ions per mole. Note that we have measured ionic concentration function c_i in moles per liter (mol/L), time t in picoseconds (ps), spatial length in angstroms (Å), and diffusion function $\mathcal{D}_i$ in units Å^2/ps. In physics, the Robin boundary condition (2.10) reflects the fact that none of ionic particles cross the interface $\Gamma_p\cup\Gamma_m$ to enter the protein and membrane regions D_p and D_m; the boundary value functions g and g_i can be properly selected, as shown in (6.1) in section 6, to mimic an external voltage across the membrane.

When u is known, an electrostatic potential function, Φ, is found by

$$\Phi(t,\mathbf{r}) = \frac{k_B T}{e_c}u(t,\mathbf{r}), \qquad \mathbf{r}\in\Omega,\ t>0,$$

in volts. Due to the above relation, the dimensionless potential u can be viewed as an electrostatic potential with the constant k_BT/e_c as its physical unit.

At $T = 298.15$, the values of α, β, and $\frac{k_BT}{e_c}$ can be estimated as

$$\alpha \approx 7042.9399, \quad \beta \approx 4.2413, \quad k_BT/e_c \approx 0.0257 \text{ volts}.$$

Thus, $u = 1$ is about 0.0257 volts or 25.7 millivolts (mV).

3. PNPic solution decomposition. To overcome the singularity difficulty caused by atomic charges, we split the electrostatic potential function u into three component functions, G, Ψ, and $\tilde{\Phi}$, such that

$$u(t,\mathbf{r}) = G(\mathbf{r}) + \Psi(\mathbf{r}) + \tilde{\Phi}(t,\mathbf{r}), \qquad \mathbf{r}\in\Omega, \quad t\geq 0, \tag{3.1}$$

where G is a potential induced by atomic charges from the protein region D_p, Ψ is a potential induced by potentials from interface and boundary, and $\tilde{\Phi}$ is a potential induced by ionic charges from the solvent region D_s.

In particular, G can be found in the analytical expression

$$G(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p}\sum_{j=1}^{n_p}\frac{z_j}{|\mathbf{r}-\mathbf{r}_j|} \tag{3.2}$$

as a solution of the Poisson equation in the whole space $\mathbb{R}^3$:

$$-\epsilon_p\Delta G(\mathbf{r}) = \alpha\sum_{j=1}^{n_p} z_j\delta_{\mathbf{r}_j}, \quad \mathbf{r}\in\mathbb{R}^3. \tag{3.3}$$

Since G and Ψ are independent of ionic concentrations c_i, they can be calculated prior to the calculation of c_i and $\tilde{\Phi}$ so that we can treat them as two given functions during an iterative process of searching for c_i and $\tilde{\Phi}$. With this observation, we construct a linear interface boundary value problem of Ψ such that it collects all the jumpily discontinuous interface conditions produced by the splitting formula (3.1) and the related inhomogeneous boundary value conditions for the purpose of making the equation of $\tilde{\Phi}$ as simple as possible. Clearly, $\tilde{\Phi}$ is periodic on the four side surfaces of the box domain Ω. To get its periodic boundary value conditions, we set Ψ to satisfy the Dirichlet boundary value condition $\Psi + G = 0$ on Γ_N. In this way, we derive a linear interface boundary value problem of Ψ,

$$\begin{cases}
\Delta\Psi(\mathbf{r}) = 0, & \mathbf{r}\in D_m\cup D_p\cup D_s,\\
\Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p\frac{\partial\Psi(\mathbf{s}^-)}{\partial\mathbf{n}_p(\mathbf{s})} = \epsilon_s\frac{\partial\Psi(\mathbf{s}^+)}{\partial\mathbf{n}_p(\mathbf{s})} + (\epsilon_s-\epsilon_p)\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_p(\mathbf{s})}, & \mathbf{s}\in\Gamma_p,\\
\Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_m\frac{\partial\Psi(\mathbf{s}^-)}{\partial\mathbf{n}_m(\mathbf{s})} = \epsilon_s\frac{\partial\Psi(\mathbf{s}^+)}{\partial\mathbf{n}_m(\mathbf{s})} + (\epsilon_s-\epsilon_m)\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_m(\mathbf{s})}, & \mathbf{s}\in\Gamma_m,\\
\Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p\frac{\partial\Psi(\mathbf{s}^-)}{\partial\mathbf{n}_p(\mathbf{s})} = \epsilon_m\frac{\partial\Psi(\mathbf{s}^+)}{\partial\mathbf{n}_p(\mathbf{s})} + (\epsilon_m-\epsilon_p)\frac{\partial G(\mathbf{s})}{\partial\mathbf{n}_p(\mathbf{s})}, & \mathbf{s}\in\Gamma_{pm},\\
\Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s}), & \mathbf{s}\in\Gamma_D,\\
\Psi(\mathbf{s}) = -G(\mathbf{s}), & \mathbf{s}\in\Gamma_N,
\end{cases} \tag{3.4}$$

and a linear interface boundary value problem of $\tilde{\Phi}$, which has continuous interface conditions, a homogeneous Dirichlet boundary condition, and periodic boundary conditions, as follows:

$$\begin{cases}
\Delta\tilde{\Phi}(t,\mathbf{r}) = 0, & \mathbf{r}\in D_m\cup D_p,\\
-\epsilon_s\Delta\tilde{\Phi}(t,\mathbf{r}) = \beta\sum_{i=1}^{n} Z_i c_i(t,\mathbf{r}), & \mathbf{r}\in D_s,\\
\tilde{\Phi}(t,\mathbf{s}^+) = \tilde{\Phi}(t,\mathbf{s}^-), \quad \epsilon_s\frac{\partial\tilde{\Phi}(t,\mathbf{s}^+)}{\partial\mathbf{n}_p(\mathbf{s})} = \epsilon_p\frac{\partial\tilde{\Phi}(t,\mathbf{s}^-)}{\partial\mathbf{n}_p(\mathbf{s})}, & \mathbf{s}\in\Gamma_p,\\
\tilde{\Phi}(t,\mathbf{s}^+) = \tilde{\Phi}(t,\mathbf{s}^-), \quad \epsilon_s\frac{\partial\tilde{\Phi}(t,\mathbf{s}^+)}{\partial\mathbf{n}_m(\mathbf{s})} = \epsilon_m\frac{\partial\tilde{\Phi}(t,\mathbf{s}^-)}{\partial\mathbf{n}_m(\mathbf{s})}, & \mathbf{s}\in\Gamma_m,\\
\tilde{\Phi}(t,\mathbf{s}^-) = \tilde{\Phi}(t,\mathbf{s}^+), \quad \epsilon_p\frac{\partial\tilde{\Phi}(t,\mathbf{s}^-)}{\partial\mathbf{n}_p(\mathbf{s})} = \epsilon_m\frac{\partial\tilde{\Phi}(t,\mathbf{s}^+)}{\partial\mathbf{n}_p(\mathbf{s})}, & \mathbf{s}\in\Gamma_{pm},\\
\tilde{\Phi}(t,\mathbf{s}) = 0, & \mathbf{s}\in\Gamma_D,\\
\tilde{\Phi}(t,\mathbf{s}) \text{ is periodic}, & \mathbf{s}\in\Gamma_N.
\end{cases} \tag{3.5}$$

Here $\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} = \nabla G(\mathbf{s}) \cdot \mathbf{n}(\mathbf{s})$ with ∇G being given by

$$\nabla G(\mathbf{s}) = -\frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} z_j \frac{(\mathbf{s}-\mathbf{r}_j)}{|\mathbf{s}-\mathbf{r}_j|^3}. \tag{3.6}$$

It can be easy to validate that the sum of G with Ψ and $\tilde{\Phi}$ gives the solution of the Poisson ion channel interface boundary value problem (2.4). Clearly, G contains all the singular points of u. Thus, both Ψ and $\tilde{\Phi}$ are smooth within D_p, D_m, or D_s.

Using the given G and Ψ, we can treat each Nernst–Planck equation of (2.5) as an equation of c_i and $\tilde{\Phi}$,

$$\frac{\partial c_i(t,\mathbf{r})}{\partial t} = \nabla\cdot\mathcal{D}_i \left[\nabla c_i + Z_i c_i \mathbf{w} + Z_i c_i \nabla\tilde{\Phi}(t,\mathbf{r})\right], \quad \mathbf{r} \in D_s, \quad t > 0, \tag{3.7}$$

for $i = 1, 2, \ldots, n$. Here $\mathbf{w} = \nabla G(\mathbf{r}) + \nabla\Psi(\mathbf{r})$, which has been calculated.

Consequently, a combination of (3.7) with (3.5) gives a system of equations for solving $\tilde{\Phi}$ and c_i for $i = 1, 2, \ldots, n$, together with the initial and boundary value conditions (2.7)–(2.10). Note that this new system is much easier to solve numerically than the original PNPic system since it avoids the solution singularity difficulties induced by atomic charges, and $\tilde{\Phi}$ is much smoother than u because the tough parts G and Ψ of u have been removed from the construction of $\tilde{\Phi}$.

In the remaining part of this paper, we only consider the steady state of PNPic. Since in the steady state c_i, u, and $\tilde{\Phi}$ become independent of time t, the system for $\tilde{\Phi}$ and c_i is simplified as n steady Nernst–Planck boundary value problems,

$$\begin{cases} \nabla\cdot\mathcal{D}_i(\mathbf{r}) \left[\nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r})\mathbf{w}(\mathbf{r}) + Z_i c_i(\mathbf{r})\nabla\tilde{\Phi}(\mathbf{r})\right] = 0, & \mathbf{r} \in D_s, \\ \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(\mathbf{s}) \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\ c_i(\mathbf{s}) = g_i(\mathbf{s}), & \mathbf{s} \in \Gamma_D, \\ \tilde{\Phi}(\mathbf{s}) \text{ is periodic}, & \mathbf{s} \in \Gamma_N, \end{cases} \tag{3.8}$$

for $i = 1, 2, \ldots, n$, plus one interface boundary value problem,

$$\begin{cases} \Delta\tilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r} \in D_m \cup D_p, \\ -\epsilon_s \Delta\tilde{\Phi}(\mathbf{r}) = \beta \sum_{i=1}^{n} Z_i c_i(\mathbf{r}), & \mathbf{r} \in D_s, \\ \tilde{\Phi}(\mathbf{s}^+) = \tilde{\Phi}(\mathbf{s}^-), \quad \epsilon_s \frac{\partial\tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_p \frac{\partial\tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_p, \\ \tilde{\Phi}(\mathbf{s}^+) = \tilde{\Phi}(\mathbf{s}^-), \quad \epsilon_s \frac{\partial\tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_m \frac{\partial\tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})}, & \mathbf{s} \in \Gamma_m, \\ \tilde{\Phi}(\mathbf{s}^-) = \tilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \frac{\partial\tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \frac{\partial\tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_{pm}, \\ \tilde{\Phi}(\mathbf{s}) = 0, & \mathbf{s} \in \Gamma_D, \\ \tilde{\Phi}(\mathbf{s}) \text{ is periodic}, & \mathbf{s} \in \Gamma_N. \end{cases} \tag{3.9}$$

When $\tilde{\Phi}$ is known, we obtain u by the formula

$$u(\mathbf{r}) = G(\mathbf{r}) + \Psi(\mathbf{r}) + \tilde{\Phi}(\mathbf{r}), \quad \mathbf{r} \in \Omega.$$

4. Variational formulations. One key step in the development of a finite element algorithm for solving the PNPic model is to derive the variational forms of interface boundary value problems (3.4) and (3.9) and Nernst–Planck system (3.8). In

this section, we obtain these forms and give them detailed proofs since their derivations are nontrivial due to the complicated interface conditions and periodic boundary value conditions. We then obtain a variational form of the system of (3.8) and (3.9). Furthermore, we simplify the variational form of (3.4) into a variational problem without involving any surface integral when the membrane permittivity constant ϵ_m is set to be equal to the protein permittivity constant ϵ_p.

Let $H^1(\Omega)$ and $H^1(D_s)$ be the Sobolev function spaces based on the box domain Ω and solvent region D_s, respectively [1]. We define their subspaces, $U, U_0, H_0^1(\Omega), V$, and V_0, as follows:

$$U = \{u \in H^1(\Omega) \mid \ u \text{ is periodic on } \Gamma_N \ \}, \quad U_0 = \{u \in U \mid u = 0 \text{ on } \Gamma_D\}, \tag{4.1}$$
$$H_0^1(\Omega) = \{v \in H^1(\Omega) \mid v = 0 \text{ on } \partial\Omega\},$$

$$V = \{v \in H^1(D_s) \mid \ v \text{ is periodic on } \Gamma_N \cap \partial D_s \ \}, \ V_0 = \{v \in V \mid v = 0 \text{ on } \Gamma_D\}. \tag{4.2}$$

We first present a variational form of the interface boundary value problem (3.9) in Theorem 4.1.

THEOREM 4.1. *The linear interface boundary value problem* (3.9) *has the following variational form:*

$$\textit{Find } \tilde{\Phi} \in U_0 \textit{ such that} \quad a(\tilde{\Phi}, v) = \beta \sum_{i=1}^{n} Z_i \int_{D_s} c_i v d\mathbf{r} \qquad \forall v \in U_0, \tag{4.3}$$

where U_0 is defined in (4.1) *and $a(\tilde{\Phi}, v)$ is defined by*

$$a(\tilde{\Phi}, v) = \epsilon_p \int_{D_p} \nabla\tilde{\Phi} \cdot \nabla v d\mathbf{r} + \epsilon_m \int_{D_m} \nabla\tilde{\Phi} \cdot \nabla v d\mathbf{r} + \epsilon_s \int_{D_s} \nabla\tilde{\Phi} \cdot \nabla v d\mathbf{r}. \tag{4.4}$$

Proof. We multiply the first and second equations of (3.9) with a test function $v \in U_0$; integrate it over D_p, D_m, and D_s, respectively; and then add them together to get

$$\begin{aligned} &-\epsilon_p \int_{D_p} \Delta\tilde{\Phi}(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} - \epsilon_m \int_{D_m} \Delta\tilde{\Phi}(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} - \epsilon_s \int_{D_s} \Delta\tilde{\Phi}(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} \\ &= \beta \sum_{i=1}^{n} Z_i \int_{D_s} c_i(\mathbf{r}) v(\mathbf{r}) d\mathbf{r}. \end{aligned}$$

Using Green's first identity, we can rewrite the above equation as

$$\begin{aligned} &\epsilon_p \int_{D_p} \nabla\tilde{\Phi}(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_m \int_{D_m} \nabla\tilde{\Phi}(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_s \int_{D_s} \nabla\tilde{\Phi}(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} \\ &= \epsilon_p \int_{\partial D_p} \frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \epsilon_m \int_{\partial D_m} \frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \epsilon_s \int_{\partial D_s} \frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \\ &+ \beta \sum_{i=1}^{n} Z_i \int_{D_s} c_i(\mathbf{r}) v(\mathbf{r}) d\mathbf{r}, \end{aligned} \tag{4.5}$$

where ∂D_p, ∂D_m, and ∂D_s denote the boundaries of D_p, D_m, and D_s and $\mathbf{n}_p$, $\mathbf{n}_m$, and $\mathbf{n}_s$ denote the unit outward normal vectors of D_p, D_m, and D_s, respectively. Note

that the normal vectors have the relations

$$\mathbf{n}_s = -\mathbf{n}_p \text{ on } \Gamma_p, \quad \mathbf{n}_s = -\mathbf{n}_m \text{ on } \Gamma_m, \ \mathbf{n}_m = -\mathbf{n}_p \text{ on } \Gamma_{pm},$$
$$\mathbf{n}_m = \mathbf{n}_b \text{ on } \Gamma_N \cap \partial D_m, \quad \mathbf{n}_s = \mathbf{n}_b \text{ on } \Gamma_N \cap \partial D_s,$$

and the boundaries ∂D_p, ∂D_m, and ∂D_s can be expressed as

$$\partial D_p = \Gamma_p \cup \Gamma_{pm}, \quad \partial D_m = \Gamma_m \cup (\Gamma_N \cap \partial D_m) \cup \Gamma_{pm}, \quad \partial D_s = \Gamma_m \cup \Gamma_p \cup \Gamma_D \cup (\Gamma_N \cap \partial D_s).$$

Hence, by $v = 0$ on Γ_D, the three surface integrals of (4.5) can be simplified as follows:

$$\begin{aligned}
\int_{\partial D_p} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} &= \int_{\Gamma_p} \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \int_{\Gamma_{pm}} \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s}, \\
\int_{\partial D_m} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} &= \int_{\Gamma_m} \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{\Gamma_{pm}} \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \\
&\quad + \int_{\Gamma_N \cap \partial D_m} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s}, \\
\int_{\partial D_s} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} &= -\int_{\Gamma_m} \frac{\partial \tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{\Gamma_p} \frac{\partial \tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \\
&\quad + \int_{\Gamma_N \cap \partial D_s} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s},
\end{aligned}$$

where $\mathbf{n}_b$ denotes the unit outward normal vector of the box domain Ω. Applying the above expressions and the interface conditions of (3.9)–(4.5), we obtain

$$\begin{aligned}
a(\tilde{\Phi}, v) = \beta \sum_{i=1}^{n} Z_i \int_{D_s} c_i v d\mathbf{r} &+ \epsilon_m \int_{\Gamma_N \cap \partial D_m} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \\
&+ \epsilon_s \int_{\Gamma_N \cap \partial D_s} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s}.
\end{aligned}$$

Clearly, the normal vectors $\mathbf{n}_b = (\pm 1, 0, 0)$ and $(0, \pm 1, 0)$ on the four side surfaces of Γ_N. Thus, the surface integral $\int_{\Gamma_N \cap \partial D_s} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) d\mathbf{s}$ can be written as

$$\begin{aligned}
&\int_{\Gamma_N \cap \partial D_s} \frac{\partial \tilde{\Phi}(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v(\mathbf{s}) d\mathbf{s} \\
&= \int_{Lz1}^{Z1} \int_{Ly1}^{Ly2} \left[\frac{\partial \tilde{\Phi}(L_{x2}, y, z)}{\partial x} v(L_{x2}, y, z) - \frac{\partial \tilde{\Phi}(L_{x1}, y, z)}{\partial x} v(L_{x1}, y, z) \right] dydz \\
&\quad + \int_{Z2}^{Lz2} \int_{Ly1}^{Ly2} \left[\frac{\partial \tilde{\Phi}(L_{x2}, y, z)}{\partial x} v(L_{x2}, y, z) - \frac{\partial \tilde{\Phi}(L_{x1}, y, z)}{\partial x} v(L_{x1}, y, z) \right] dydz \\
&\quad + \int_{Lz1}^{Z1} \int_{Lx1}^{Lx2} \left[\frac{\partial \tilde{\Phi}(x, L_{y2}, z)}{\partial y} v(x, L_{y2}, z) - \frac{\partial \tilde{\Phi}(x, L_{y1}, z)}{\partial y} v(x, L_{y1}, z) \right] dxdz \\
&\quad + \int_{Z2}^{Lz2} \int_{Lx1}^{Lx2} \left[\frac{\partial \tilde{\Phi}(x, L_{y2}, z)}{\partial y} v(x, L_{y2}, z) - \frac{\partial \tilde{\Phi}(x, L_{y1}, z)}{\partial y} v(x, L_{y1}, z) \right] dxdz,
\end{aligned}$$

where $Z1$ and $Z2$ denote the starting and ending numbers of the membrane in the Z-axis direction, respectively. Since each test function v satisfies the periodic boundary conditions, the above expression becomes

$$\begin{aligned}
&\int_{\Gamma_N\cap\partial D_s}\frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial\mathbf{n}_b(\mathbf{s})}v(\mathbf{s})d\mathbf{s} \qquad (4.6)\\
&=\int_{Lz1}^{Z1}\int_{Ly1}^{Ly2}\left[\frac{\partial\tilde{\Phi}(L_{x2},y,z)}{\partial x}-\frac{\partial\tilde{\Phi}(L_{x1},y,z)}{\partial x}\right]v(L_{x1},y,z)dydz\\
&\quad+\int_{Z2}^{Lz2}\int_{Ly1}^{Ly2}\left[\frac{\partial\tilde{\Phi}(L_{x2},y,z)}{\partial x}-\frac{\partial\tilde{\Phi}(L_{x1},y,z)}{\partial x}\right]v(L_{x1},y,z)dydz\\
&\quad+\int_{Lz1}^{Z1}\int_{Lx1}^{Lx2}\left[\frac{\partial\tilde{\Phi}(x,L_{y2},z)}{\partial y}-\frac{\partial\tilde{\Phi}(x,L_{y1},z)}{\partial y}\right]v(x,L_{y1},z)dxdz \qquad (4.7)\\
&\quad+\int_{Z2}^{Lz2}\int_{Lx1}^{Lx2}\left[\frac{\partial\tilde{\Phi}(x,L_{y2},z)}{\partial y}-\frac{\partial\tilde{\Phi}(x,L_{y1},z)}{\partial y}\right]v(x,L_{y1},z)dxdz.
\end{aligned}$$

From the periodicity of $\tilde{\Phi}$ on Γ_N, it can imply that the partial derivatives $\frac{\partial\tilde{\Phi}}{\partial x}$ and $\frac{\partial\tilde{\Phi}}{\partial y}$ satisfy the following periodic boundary conditions:

$$\begin{aligned}
\frac{\partial\tilde{\Phi}(L_{x1},y,z)}{\partial x}&=\frac{\partial\tilde{\Phi}(L_{x2},y,z)}{\partial x}\quad\forall(y,z)\in D_1,\\
\frac{\partial\tilde{\Phi}(x,L_{y1},z)}{\partial y}&=\frac{\partial\tilde{\Phi}(x,L_{y2},z)}{\partial y}\quad\forall(x,z)\in D_2.
\end{aligned}$$

Applying the above equations to (4.6) immediately gives

$$\int_{\Gamma_N\cap\partial D_s}\frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial\mathbf{n}_b(\mathbf{s})}v(\mathbf{s})d\mathbf{s}=0. \qquad (4.8)$$

Similarly, we can prove that $\int_{\Gamma_N\cap\partial D_m}\frac{\partial\tilde{\Phi}(\mathbf{s})}{\partial\mathbf{n}_b(\mathbf{s})}v(\mathbf{s})\mathrm{d}\mathbf{s}=0$. This completes the proof. ∎

We next present a variational formulation of the Nernst–Planck system (3.8) in Theorem 4.2.

THEOREM 4.2. *The system* (3.8) *of n steady Nernst–Planck equations has the following variational form: Find* $c_i\in V$ *satisfying* $c_i=g_i$ *on* Γ_D *such that*

$$\int_{D_s}\mathcal{D}_i(\mathbf{r})\left(\nabla c_i(\mathbf{r})+Z_ic_i(\mathbf{r})\nabla u(\mathbf{r})\right)\nabla v_i(\mathbf{r})d\mathbf{r}=0\quad\forall v_i\in V_0,\quad i=1,2,\ldots,n, \qquad (4.9)$$

where V *and* V_0 *are given in* (4.2).

Proof. We multiply a test function $v_i\in V_0$ on both sides of the first equation of (3.8), integrate on the solvent region D_s, and use Green's first identity to get

$$\int_{\partial D_s}\mathcal{D}_i\left(\frac{\partial c_i(\mathbf{s})}{\partial\mathbf{n}_s(\mathbf{s})}+Z_ic_i\frac{\partial u(\mathbf{s})}{\partial\mathbf{n}_s(\mathbf{s})}\right)v_i(\mathbf{s})\mathrm{d}\mathbf{s}-\int_{D_s}\mathcal{D}_i\left(\nabla c_i+Z_ic_i\nabla u\right)\nabla v_i d\mathbf{r}=0. \qquad (4.10)$$

Since the boundary ∂D_s of D_s can be expressed as

$$\partial D_s=\Gamma_m\cup\Gamma_p\cup\Gamma_D\cup(\Gamma_N\cap\partial D_s),$$

we can use the second equation of (3.8) and $v_i = 0$ on Γ_D to get

$$\int_{\partial D_s} \mathcal{D}_i \left(\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} \right) v_i(\mathbf{s}) \mathrm{d}\mathbf{s} = \mathcal{D}_i \int_{\Gamma_N \cap \partial D_s} \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v_i(\mathbf{s}) \mathrm{d}\mathbf{s} + \mathcal{D}_i Z_i \int_{\Gamma_N \cap \partial D_s} c_i \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} v_i(\mathbf{s}) \mathrm{d}\mathbf{s} \quad \forall v_i \in U_0,$$

where we have used the fact that $\mathbf{n}_s = \mathbf{n}_b$ on Γ_N and $\mathcal{D}_i$ is a constant on the side surface $\Gamma_N \cap \partial D_s$. Clearly, from the periodicities of c_i and u, it can imply the periodicities of the partial derivatives $\frac{\partial c_i}{\partial x}$, $\frac{\partial c_i}{\partial y}$, $\frac{\partial u}{\partial x}$, and $\frac{\partial u}{\partial y}$ on the side surfaces $\Gamma_N \cap \partial D_s$ and Γ_N, respectively. Similarly to what is done in the proof of (4.8), we can use the periodicities of c_i, v_i, $\frac{\partial c_i}{\partial x}$, and $\frac{\partial c_i}{\partial y}$ on $\Gamma_N \cap \partial D_s$ and the periodicities of u, $\frac{\partial u}{\partial x}$, and $\frac{\partial u}{\partial y}$ on Γ_N to get

$$\int_{\Gamma_N \cap \partial D_s} \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v_i(\mathbf{s}) \mathrm{d}\mathbf{s} = 0, \quad \int_{\Gamma_N \cap \partial D_s} c_i \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v_i(\mathbf{s}) \mathrm{d}\mathbf{s} = 0.$$

Thus, we obtain

$$\int_{\partial D_s} \mathcal{D}_i \left(\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} \right) v_i(\mathbf{s}) \mathrm{d}\mathbf{s} = 0.$$

Applying the above equation to (4.10) gives the weak form (4.9). This completes the proof. ∎

Furthermore, a variational form of the interface boundary value problem (3.4) is presented in Theorem 4.3.

THEOREM 4.3. *The linear interface boundary value problem* (3.4) *has the following variational form: Find* $\Psi \in H^1(\Omega)$ *satisfying* $\Psi = g - G$ *on* Γ_D *and* $\Psi = -G$ *on* Γ_N *such that*

$$a(\Psi, v) = (\epsilon_s - \epsilon_p) \int_{\Gamma_p} \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + (\epsilon_s - \epsilon_m) \int_{\Gamma_m} \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + (\epsilon_m - \epsilon_p) \int_{\Gamma_{pm}} \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \quad \forall v \in H_0^1(\Omega), \tag{4.11}$$

where $\mathbf{n}_m$ *and* $\mathbf{n}_p$ *denote the unit outward normal vectors of* D_m *and* D_p*, respectively, and* $a(\cdot, \cdot)$ *is defined in* (4.4).

Proof. We multiply the first equation of (3.4) with a test function $v \in H_0^1(\Omega)$; integrate it over D_p, D_m, and D_s, respectively; and then add them together to get

$$\epsilon_p \int_{D_p} \Delta\Psi(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_m \int_{D_m} \Delta\Psi(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_s \int_{D_s} \Delta\Psi(\mathbf{r}) v(\mathbf{r}) \mathrm{d}\mathbf{r} = 0.$$

Applying Green's first identity to each of the above three integrals, we can get

$$\begin{aligned} &\epsilon_p \int_{D_p} \nabla\Psi(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_m \int_{D_m} \nabla\Psi(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} + \epsilon_s \int_{D_s} \nabla\Psi(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} \\ &= \epsilon_p \int_{\partial D_p} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \epsilon_m \int_{\partial D_m} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \epsilon_s \int_{\partial D_s} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s}. \end{aligned} \tag{4.12}$$

By $v = 0$ on $\Gamma_D \cup \Gamma_N$ (i.e., the boundary $\partial\Omega$), the three surface integrals of (4.12) can be simplified as follows:

$$\int_{\partial D_p} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} = \int_{\Gamma_p} \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} + \int_{\Gamma_{pm}} \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s},$$
$$\int_{\partial D_m} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} = \int_{\Gamma_m} \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{\Gamma_{pm}} \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s},$$
$$\int_{\partial D_s} \frac{\partial \Psi(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} = -\int_{\Gamma_m} \frac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{\Gamma_p} \frac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s}.$$

Applying the above expressions and the interface conditions of (3.4)–(4.12), we obtain (4.11). This completes the proof. ∎

In PNP ion channel simulations, it is often to set $\epsilon_m = \epsilon_p$. In this case, the weak form (4.11) can be simplified as follows: Find $\Psi \in H^1(\Omega)$ satisfying $\Psi = g - G$ on Γ_D and $\Psi = -G$ on Γ_N such that

$$a(\Psi, v) = (\epsilon_s - \epsilon_p) \int_\Gamma \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} \quad \forall v \in H_0^1(\Omega), \tag{4.13}$$

where $\mathbf{n}$ denotes the unit outward normal direction of the protein-membrane region $D_{pm} = D_p \cup D_m \cup \Gamma_{pm}$, $\Gamma = \Gamma_m \cup \Gamma_p$, which is the interface between D_{pm} and D_s, and $a(u, v)$ is simplified as follows:

$$a(u, v) = \epsilon_p \int_{D_{pm}} \nabla u \cdot \nabla v d\mathbf{r} + \epsilon_s \int_{D_s} \nabla \tilde{\Phi} \cdot \nabla v d\mathbf{r}. \tag{4.14}$$

THEOREM 4.4. *Let the gradient vector ∇G be given in* (3.6). *If $\epsilon_m = \epsilon_p$ and $\Gamma = \Gamma_m \cup \Gamma_p$, then*

$$\int_\Gamma \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} = -\int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r}. \tag{4.15}$$

Proof. Using Green's first identity, $\Delta G = 0$ in D_s, $\partial D_s = \Gamma \cup \Gamma_D \cup (\Gamma_N \cap \partial D_s)$, and $v = 0$ on $\Gamma_D \cup \Gamma_N$, we get

$$\begin{aligned} 0 = \int_{D_s} \Delta G v d\mathbf{r} &= \int_{\partial D_s} \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} \\ &= \int_\Gamma \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} - \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r}. \end{aligned}$$

Since $\mathbf{n}_s = -\mathbf{n}$ on Γ, from the above expression, it gives the identity (4.15). This completes the proof. ∎

Applying (4.15) to the variational problem (4.13), we obtain another variational form of Ψ as follows: Find $\Psi \in H^1(\Omega)$ satisfying $\Psi = g - G$ on Γ_D and $\Psi = -G$ on Γ_N such that

$$a(\Psi, v) = (\epsilon_p - \epsilon_s) \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} \qquad \forall v \in H_0^1(\Omega). \tag{4.16}$$

The above weak form simplifies the numerical calculation of Ψ since it does not involve any surface integral. A surface integral can be more difficult to calculate than a

corresponding volume integral since a geometrical shape of the interface Γ is very complicated in an ion channel simulation.

In summary, we have obtained a variational form of the system of (3.8) and (3.9) as follows: Find $\tilde{\Phi} \in V_0$ and $c_i \in U$ with $c_i = g_i$ on Γ_D for $i = 1, 2, \ldots, n$ such that

$$\begin{cases} \int_{D_s} \mathcal{D}_i \left[\nabla c_i + Z_i c_i (\mathbf{w} + \nabla \tilde{\Phi})\right] \nabla v_i d\mathbf{r} = 0 & \forall v_i \in U_0 \text{ for } i = 1, 2, \ldots, n, \\ a(\tilde{\Phi}, v) - \beta \sum_{i=1}^{n} Z_i \int_{D_s} c_i v d\mathbf{r} = 0 & \forall v \in V_0, \end{cases} \tag{4.17}$$

where $\mathbf{w} = \nabla G(\mathbf{r}) + \nabla \Psi(\mathbf{r})$ with ∇G being given in (3.6) and Ψ is a solution of (4.11) (or (4.16) in the case that $\epsilon_m = \epsilon_p$).

5. A PNPic finite element solver. Let Ω_h be an interface fitted irregular tetrahedral mesh of a box domain Ω. We use Ω_h to construct two linear Lagrange finite element function spaces, $\mathcal{U}_1$ and $\mathcal{U}_2$, as two finite-dimensional subspaces of the function spaces $H^1(\Omega)$ and U, respectively. From Ω_h, we extract an irregular tetrahedral mesh, $D_{s,h}$, of D_s to construct two linear Lagrange finite element function spaces, $\mathcal{V}_1$ and $\mathcal{V}_2$, as two finite-dimensional subspaces of the function spaces $H^1(D_s)$ and V, respectively. We also define three subspaces, $\mathcal{U}_{1,0}$, $\mathcal{U}_{2,0}$, and $\mathcal{V}_{2,0}$, by

$$\mathcal{U}_{1,0} = \{u \in \mathcal{U}_1 \mid \quad u = 0 \text{ on } \partial\Omega\}, \quad \mathcal{U}_{2,0} = \{u \in \mathcal{U}_2 \mid u = 0 \text{ on } \Gamma_D\},$$
$$\mathcal{V}_{2,0} = \{v \in \mathcal{V}_2 \mid v = 0 \text{ on } \Gamma_D\}.$$

Here U and V have been defined in (4.1) and (4.2), respectively.

Since Ψ, $\tilde{\Phi}$, and c_i belong to three different finite element spaces, $\mathcal{U}_1$, $\mathcal{U}_2$, and $\mathcal{V}_2$, respectively, we construct three communication operators P_1, P_2, and P_3 by

$$P_1 : \mathcal{U}_2 \to \mathcal{U}_1, \quad P_2 : \mathcal{U}_1 \to \mathcal{V}_1, \quad P_3 : \mathcal{V}_2 \to \mathcal{U}_2.$$

For example, we map $\tilde{\Phi}$ from the periodic boundary constrained finite element space $\mathcal{U}_2$ onto the original finite element space $\mathcal{U}_1$ by linear operator P_1 to complete the addition of $\tilde{\Phi}$ with G and Ψ. Using these linear operators, we approximate the system (4.17) by a system of finite element equations as follows: Find $\tilde{\Phi} \in \mathcal{U}_{2,0}$ and $c_i \in \mathcal{V}_2$ satisfying $c_i = g_i$ on Γ_D for $i = 1, 2, \ldots, n$ such that

$$\begin{cases} \int_{D_s} \mathcal{D}_i \left[\nabla c_i + Z_i c_i \nabla P_2 (G + \Psi + P_1 \tilde{\Phi})\right] \nabla v_i d\mathbf{r} = 0 & \forall v_i \in \mathcal{V}_{2,0} \\ & \text{for } i = 1, 2, \ldots, n, \\ a(\tilde{\Phi}, v) - \beta \sum_{j=1}^{n} Z_j \int_{D_s} P_3 c_j v d\mathbf{r} = 0 & \forall v \in \mathcal{U}_{2,0}, \end{cases} \tag{5.1}$$

where G is given in (3.2) and Ψ has been calculated through solving a finite element approximation of the variational problem (4.11). For example, in the case that $\epsilon_m = \epsilon_p$, the finite element equation for computing Ψ is given as follows: Find $\Psi \in \mathcal{U}_1$ satisfying $\Psi = g - G$ on Γ_D and $\Psi = -G$ on Γ_N such that

$$a(\Psi, v) = (\epsilon_p - \epsilon_s) \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) \mathrm{d}\mathbf{r} \quad \forall v \in \mathcal{U}_{1,0}, \tag{5.2}$$

where the bilinear form $a(\cdot, \cdot)$ is given in (4.14).

We recall that the Slotboom variable transformation is defined by

$$c_i = e^{-Z_i u} \bar{c}_i, \quad i = 1, 2, \ldots, n, \tag{5.3}$$

where $\bar{c}_i$ denotes the ith Slotboom variable [47]. From the periodicity of u and c_i on $\Gamma_N \cap \partial D_s$, it can imply that $\bar{c}_i$ is periodic on $\Gamma_N \cap \partial D_s$. Using (5.3), we can get

$$\nabla c_i + Z_i c_i \nabla u = e^{-Z_i u} \nabla \bar{c}_i, \quad i = 1, 2, \ldots, n, \tag{5.4}$$

and then transform the system (5.1) into a new system of $\tilde{\Phi}$ and $\bar{c}_i$ as follows: Find $\tilde{\Phi} \in \mathcal{U}_{2,0}$ and $\bar{c}_i \in \mathcal{V}_2$ satisfying $\bar{c}_i = \bar{g}_i$ on Γ_D for $i = 1, 2, \ldots, n$ such that

$$\begin{cases} \int_{D_s} \mathcal{D}_i e^{-Z_i P_2 (G + \Psi + P_1 \tilde{\Phi})} \nabla \bar{c}_i \nabla v_i d\mathbf{r} = 0 & \forall v_i \in \mathcal{V}_{2,0} \\ & \text{for } i = 1, 2, \ldots, n, \\ a(\tilde{\Phi}, v) - \beta \sum_{i=1}^{n} Z_i \int_{D_s} e^{-Z_i (G + \Psi + P_1 \tilde{\Phi})} P_3 \bar{c}_i, v d\mathbf{r} = 0 & \forall v \in \mathcal{U}_{2,0}, \end{cases} \tag{5.5}$$

where $\bar{g}_i = e^{Z_i g} g_i$, which is derived from the boundary value conditions $u = g$ and $c_i = g_i$ on Γ_D. After finding $\bar{c}_i$, we recover c_i using (5.3) for $i = 1, 2, \ldots, n$.

We now construct a relaxation iterative scheme for solving the nonlinear finite element system (5.5) using the classic successive relaxation iterative techniques [42]. Let $\tilde{\Phi}^k$ and $\bar{c}_i^k$ denote the kth iterative approximations to $\tilde{\Phi}$ and $\bar{c}_i$, respectively. We define them for $k = 0, 1, 2, \ldots$ by

$$\bar{c}_i^{k+1} = \bar{c}_i^k + \omega(\bar{p}_i - \bar{c}_i^k), \quad i = 1, 2, \ldots, n, \tag{5.6}$$

$$\tilde{\Phi}^{k+1} = \tilde{\Phi}^k + \omega(\bar{q} - \tilde{\Phi}^k), \tag{5.7}$$

where $\bar{p}_i \in \mathcal{V}_2$ satisfying $\bar{p}_i = \bar{g}_i$ on Γ_D such that

$$\int_{D_s} D_i e^{-Z_i P_2 (G + \Psi + P_1 \tilde{\Phi}^k)} \nabla \bar{p}_i \nabla v_i d\mathbf{r} = 0 \quad \forall v_i \in \mathcal{V}_{2,0}, \quad i = 1, 2, \ldots, n, \tag{5.8}$$

and $\bar{q}$ is a solution of the nonlinear variational problem: Find $\bar{q} \in \mathcal{U}_{2,0}$ such that

$$a(\bar{q}, v) - \beta \sum_{i=1}^{n} Z_i \int_{D_s} e^{-Z_i (G + \Psi + P_1 \bar{q})} P_3 \bar{c}_i^{k+1} v d\mathbf{r} = 0 \quad \forall v \in \mathcal{U}_{2,0}, \tag{5.9}$$

$\bar{c}_i^0$ and $\tilde{\Phi}^0$ are given initial iterates, and ω is a relaxation parameter between 0 and 1.

By default, we set that $\bar{c}_i^0 = c_i^b$, and $\tilde{\Phi}^0$ is a solution of the variational problem: Find $\tilde{\Phi}^0 \in \mathcal{U}_{2,0}$ such that

$$a(\tilde{\Phi}^0, v) - \beta \sum_{i=1}^{n} Z_i c_i^b \int_{D_s} e^{-Z_i (G + \Psi + P_1 \tilde{\Phi}^0)} v d\mathbf{r} = 0 \; \forall v \in \mathcal{U}_{2,0}. \tag{5.10}$$

We stop this iteration process whenever the following criteria hold:

$$\|\tilde{\Phi}^{k+1} - \tilde{\Phi}^k\| < \epsilon \quad \text{and} \quad \max_{1 \le i \le n} \|\bar{c}_i^{k+1} - \bar{c}_i^k\| < \epsilon, \tag{5.11}$$

where ϵ is a tolerance (e.g., $\epsilon = 10^{-5}$) and $\|\cdot\|$ denotes the L_2 norm.

In order to solve the nonlinear variational problem (5.9) in the kth iteration, we construct an iterative sequence, $\{q_k^j\}$, by

$$q_k^{j+1} = q_k^j + \xi_k^j, \qquad j = 0, 1, 2, \ldots, \tag{5.12}$$

where $q_k^0 = \tilde{\Phi}^k$ and ξ_k^j is a solution of the variational problem: Find $\xi_k^j \in \mathcal{U}_{2,0}$ such that

$$a(\xi_k^j, v) + \beta \int_{D_s} \sum_{i=1}^{n} Z_i^2 P_3 \bar{c}_i^{k+1} e^{-Z_i(G+\Psi+P_1 q_k^j)} \xi_k^j v d\mathbf{r} = \beta \int_{D_s} \sum_{i=1}^{n} Z_i e^{-Z_i(G+\Psi+P_1 q_k^j)} P_3 \bar{c}_i^{k+1} v d\mathbf{r} - a(q_k^j, v) \quad \forall v \in \mathcal{U}_{2,0}. \tag{5.13}$$

To get the initial iterate $\tilde{\Phi}^0$, we construct an iterative sequence, $\{q^j\}$, for solving the nonlinear variational problem (5.10) by

$$q^{j+1} = q^j + \xi^j, \qquad j = 0, 1, 2, \ldots, \tag{5.14}$$

where initial iterate q^0 is set as a solution of a linearized problem of (5.10),

$$a(\phi, v) + \beta \sum_{i=1}^{n} Z_i^2 c_i^b \int_{D_s} \phi v d\mathbf{r} = -\beta \sum_{i=1}^{n} Z_i^2 c_i^b \int_{D_s} (G + \Psi) v d\mathbf{r} \quad \forall v \in \mathcal{U}_{2,0}, \tag{5.15}$$

and ξ^j is a solution of the linear variational problem: Find $\xi^j \in \mathcal{U}_{2,0}$ such that

$$a(\xi^j, v) + \beta \int_{D_s} \sum_{i=1}^{n} Z_i^2 c_i^b e^{-Z_i(G+\Psi+P_1 q_k^j)} \xi^j v d\mathbf{r} = \beta \int_{D_s} \sum_{i=1}^{n} Z_i c_i^b e^{-Z_i(G+\Psi+P_1 q_k^j)} v d\mathbf{r} - a(q_k^j, v) \quad \forall v \in \mathcal{U}_{2,0}. \tag{5.16}$$

In (5.15), we have used the electroneutrality condition $\sum_{i=1}^{n} Z_i c_i^b = 0$.

In the iterative process of (5.12), we use the iteration stopping criterion,

$$\text{either} \quad j > \text{Ite\_max} \quad \text{or} \quad \|q_k^{j+1} - q_k^j\| < \tau, \tag{5.17}$$

where Ite_max denotes the maximum allowable number of iterations and τ is a tolerance. In calculation, we set Ite_max = 10 and $\tau = 10^{-5}$ by default. Similarly, we stop the iterative process of (5.14) whenever

$$\text{either} \quad j > \text{Ite\_max} \quad \text{or} \quad \|q^{j+1} - q^j\| < \tau. \tag{5.18}$$

For clarity, we summarize our relaxation iterative scheme in Algorithm 1.

Algorithm 1. *Our finite element relaxation iterative scheme for solving the steady state PNPic system of* (3.8) *and* (3.9) *for the electrostatic potential u and ionic concentrations c_i can be implemented in five steps:*

Step 1. *Initialization: Calculate G by* (3.2)*; calculate Ψ by solving a finite element approximation problem of* (4.11) *(or* (5.2) *when $\epsilon_m = \epsilon_p$); set the initial iterates $\bar{c}_i^0 = c_i^b$ for $i = 1, 2, \ldots, n$; calculate $\tilde{\Phi}^0$ as a solution of the nonlinear problem* (5.10) *by the iterative scheme* (5.14)*; and set $k = 0$.*

Step 2. *Define $\bar{c}_i^{k+1}$ by* (5.6) *with $\bar{p}_i$ being a solution of the linear variational problem* (5.8) *for $i = 1, 2, \ldots, n$.*

Step 3. *Define $\tilde{\Phi}^{k+1}$ by* (5.7) *with $\bar{q}$ being an iterate q_k^j of the iterative scheme* (5.12) *for solving the nonlinear variational problem* (5.9) *satisfying the iteration stop rule* (5.17)*.*

Step 4. *Check the convergence: If the iteration stop criteria of* (5.11) *hold, go to Step* 5 *with* $\bar{c}_i = \bar{c}_i^{k+1}$ *for* $i = 1, 2, \ldots, n$ *and* $\tilde{\Phi} = \tilde{\Phi}^{k+1}$*; otherwise, increase* k *by* 1*, and go back to Step* 2.

Step 5. *Define the steady state PNPic solution:* $u = G + \Psi + \tilde{\Phi}$ *and* $c_i = e^{-Z_i u}\bar{c}_i$ *for* $i = 1, 2, \ldots, n$.

Remark 1. The iterative scheme defined in (5.12) is a Newton iterative method for minimizing the functional

$$J(v) = \frac{1}{2}a(v, v) + \beta \int_{D_s} \sum_{i=1}^{n} \bar{c}_i^{k+1} e^{-Z_i(G+\Psi+P_1 v)} d\mathbf{r}.$$

It can be shown that the minimizer of J gives a solution of the nonlinear variational problem (5.9). This statement is true for the iterative scheme defined in (5.14) if Slotboom iterates $\bar{c}_i^{k+1}$ of J are replaced by the bulk concentrations c_i^b.

Remark 2. The iterative scheme of (5.14) is actually a finite element Newton iterative scheme for solving a PB ion channel model using the periodic boundary conditions given in (2.3). That is, this PB ion channel model is defined by the equations of (3.3), (3.4), and (3.5) using $c_i = c_i^b e^{-Z_i u}$ for $i = 1, 2, \ldots, n$. It can be shown that the solution u of this PB ion channel model can be constructed by

$$u = G + \Psi + \tilde{\Phi}^{PB}, \tag{5.19}$$

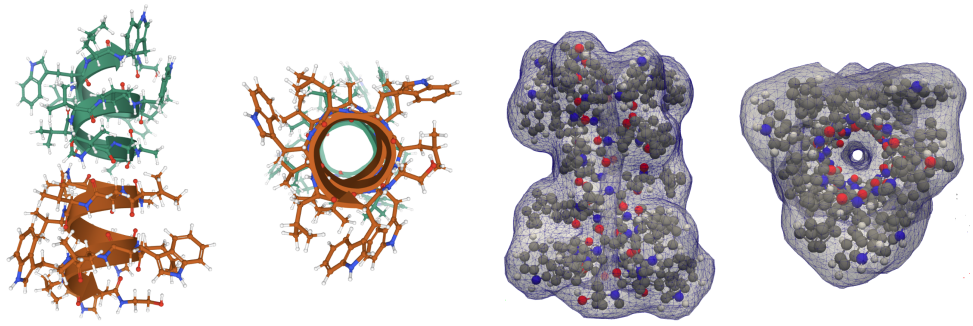
where $\tilde{\Phi}^{PB}$ denotes a solution of the nonlinear variational problem (5.10). This PB ion channel model and finite element solver are different from those reported in [24].

6. Numerical results. We implemented Algorithm 1 in Python as a software package based on the state-of-the-art finite element library from the FEniCS project [35] and the PB finite element solver program package reported in [52]. We used the ion channel finite element mesh program package developed by Lu's research group [10, 30, 31] to generate interface fitted irregular tetrahedral meshes for a box domain Ω as illustrated in Figure 1. From a mesh of Ω, we extracted the meshes of solvent region D_s, membrane region D_m, and protein region D_p, denoted by $D_{s,h}, D_{m,h}$, and $D_{p,h}$, respectively. We then used these meshes to define the finite element function spaces $\mathcal{U}_1$ and $\mathcal{V}_1$. Furthermore, we modified $\mathcal{U}_1$ and $\mathcal{V}_1$ as the finite element function spaces $\mathcal{U}_2$ and $\mathcal{V}_2$ using the periodic boundary value conditions. In this software package, we set boundary value functions $g_i(\mathbf{r})$ and $g(\mathbf{r})$ with $\mathbf{r} = (x, y, z)$ for ionic concentration functions c_i and electrostatic potential function u, respectively, as follows:

$$g_i(\mathbf{r}) = \begin{cases} c_i^b & \text{at } z = L_{z1} \text{ (bottom)}, \\ c_i^b & \text{at } z = L_{z2} \text{ (top)}, \end{cases} \quad g(\mathbf{r}) = \begin{cases} u_b & \text{at } z = L_{z1} \text{ (bottom)}, \\ u_t & \text{at } z = L_{z2} \text{ (top)}, \end{cases} \tag{6.1}$$

where c_i^b is a bulk concentration of species i and the difference between electrostatic potential values u_b and u_t can be regarded as a voltage across the membrane. We also followed what was done in [49, equation (27)] to define the diffusion coefficient function $\mathcal{D}_i(\mathbf{r})$ with $\mathbf{r} = (x, y, z)$ by

$$\mathcal{D}_i(\mathbf{r}) = \begin{cases} D_{i,b}, & z < Z1 \text{ or } z > Z2 \text{ (bulk part)}, \\ D_{i,c} + (D_{i,c} - D_{i,b}) f_t(\mathbf{r}), & Z2 - \eta \le z \le Z2 \text{ (top buffer part)}, \\ D_{i,c}, & Z1 + \eta \le z \le Z2 - \eta \text{ (channel pore)}, \\ D_{i,c} + (D_{i,c} - D_{i,b}) f_b(\mathbf{r}), & Z1 \le z \le Z1 + \eta \text{ (bottom buffer part)}, \end{cases}$$



(a) Molecular structure of GA (b) Our protein region D_p fitting GA well

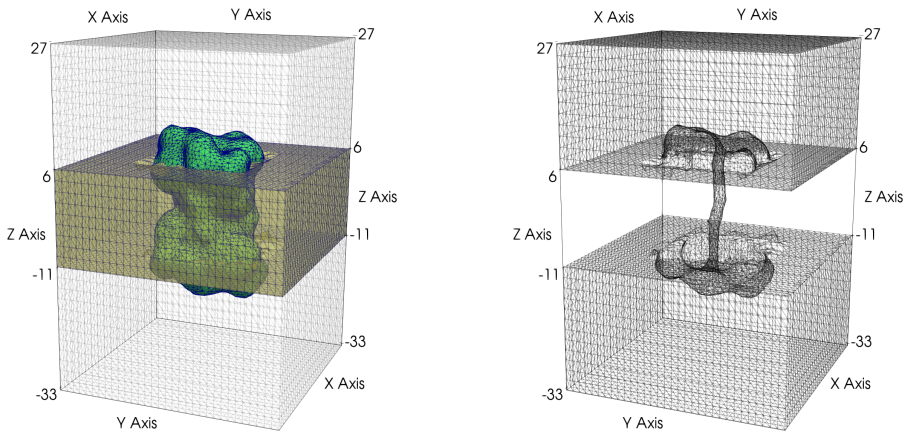
FIG. 3. (a) *Two views of GA (PDB identification code 1MAG) depicted in sticks for the molecular structure and cartoons for the two helical subunits.* (b) *Two views of our protein region D_p, along with the GA molecular structure depicted in balls for oxygen atoms (in red), nitrogen atoms (in blue), and carbon atoms (in gray).*

where $D_{i,b}$ and $D_{i,c}$ are the diffusion constants of species i for the bulk and channel pore regions, respectively; f_b and f_t are the interpolation functions given in [49, equation (27)] such that each diffusion function is sufficiently smooth in the solvent region D_s; and η is a parameter for adjusting the buffering region size. By default, each finite element equation of (5.8) and (5.13) is solved, approximately, by the generalized minimal residual method using incomplete LU preconditioning with the absolute and relative residual errors being less than 10^{-6}.

We did numerical tests on an ion channel protein, a gramicidin A (GA), in a solution of anions Cl^- and cations K^+ to demonstrate the convergence of our nonlinear relaxation iterative scheme and the computer performance of our program package. Here the charge numbers $Z_1 = 1$ and $Z_2 = -1$. The GA channel is a small protein 0.4 nm in diameter and 2.5 nm in length composed of symmetric dimers of two β-helical subunits. Two views of its molecular structure are given in Figure 3(a).

GA is an antibiotic peptide produced by *Bacillus brevis* and has been extensively studied in experiments and various modelings [3, 46]. Due to the cation-selective property and the simplicity in molecular structure compared with other ion channel proteins [2], the GA channel has been a typical molecular force probe to explore how changes in bilayer properties alter protein function [39]. With an X-ray crystallographic molecular structure [25] and the experimental data [12], the GA channel is often selected to construct numerical tests for validating PNP ion channel models [49, 54].

We downloaded the GA molecular structure file 1mag.pdb from the protein data bank (PDB, https://www.rcsb.org). We then derived its PQR file that contains the data missed in the PDB file, such as the hydrogen atoms, the atomic charge numbers, and the atomic radii. The total number n_p of atoms is 280. We rotated the ion channel and assembled it with a membrane, as illustrated in Figure 1, for a rectangular box Ω of dimensions $40 \times 40 \times 60$ defined by $L_{x1} = -20.323$, $L_{x2} = 19.677$, $L_{y1} = -20.0$, $L_{y2} = 20.0$, $L_{z1} = -33.421$, $L_{z2} = 26.579$, $Z1 = -11$, and $Z2 = 6$ for a membrane thickness of 17 Å. The meshes Ω_h and $D_{s,h}$ have 24686 and 15828 mesh points, respectively. We display them in Figure 4(a), (b) to show their geometrical complexities. Because of the periodic boundary conditions, the dimensions 24686 and 15828 of $\mathcal{U}_1$ and $\mathcal{V}_1$ were reduced to the dimensions 22541 and 14203 of $\mathcal{U}_2$ and $\mathcal{V}_2$, respectively.



(a) Mesh for the box domain Ω (b) Mesh for solvent region D_s

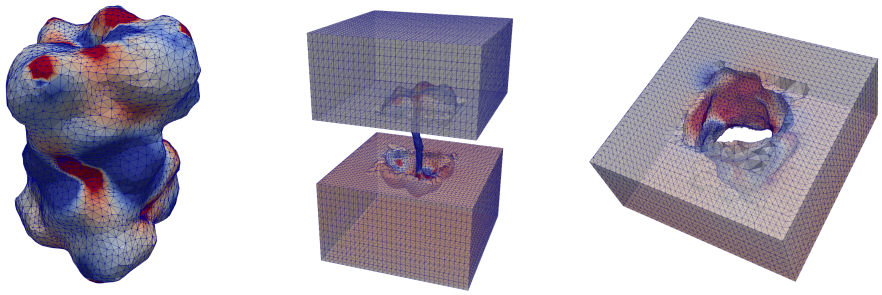
FIG. 4. *The interface fitted irregular tetrahedral meshes of the box domain Ω and solvent region D_s for the ion channel protein GA (PDB identification code 1MAG) for our numerical tests. Here the meshes of the membrane region D_m and protein region D_p are colored in yellow and green, respectively, for clarity.*

In the numerical tests, we set $\epsilon_s = 80$, $\epsilon_p = 2$, and $\epsilon_m = 2$; $D_{1,b} = 0.196$, $D_{1,c} = 0.0196$ (for K^+ ions), $D_{2,b} = 0.203$, and $D_{2,c} = 0.0203$ (for Cl^- ions); and $\eta = 3$ (for the diffusion coefficient function $\mathcal{D}_i(\mathbf{r})$). Since $\epsilon_m = \epsilon_p$, we calculated Ψ by solving the finite element variational problem (5.2). All the numerical tests were done on our iMac computer with one 4.2-GHz Intel core i7 processor and 64 GB memory.

One important feature of our PNPic software package is to be able to visualize the values of ionic concentrations c_i and electrostatic potential function u produced by our PNPic finite element solver in color mapping on a surface mesh of ion channel protein region D_p, membrane region D_m, or solvent region D_s. This feature makes our PNPic software package particularly useful in the study of ion channel properties. As an example, Figure 5 displays the values of u on the surface meshes of D_p, D_s, and D_m, respectively. The three surface mesh plots of Figure 5 also display the complicated shapes of the interfaces Γ_p, Γ_{pm}, and Γ_m. From Figure 3(b), it can be seen that our protein region D_p wraps well the molecular structure of GA.

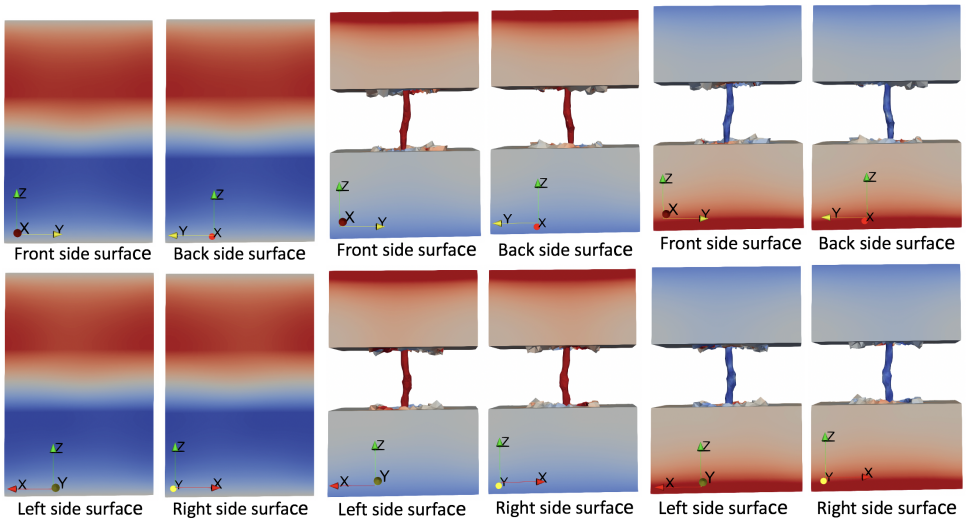
Figure 6 displays the boundary values of the electrostatic potential u and concentrations c_1 and c_2 on the four side surfaces Γ_N of the box domain Ω and the four side surfaces $\Gamma_N \cap \partial D_s$ of the solvent region D_s in color mapping. Here u, c_1 and c_2 were generated by our PNPic finite element software package using $u_b = -1$, $u_t = 1$, and $c_i^b = 0.5$ mol/L for $i = 1, 2$. The plots from this figure confirm that our PNPic finite element solution can well retain the periodic boundary value conditions (2.9).

Figure 7 displays the convergence of our relaxation iterative scheme, defined in (5.6) and (5.7) in terms of iteration numbers and the performance of our software package in terms of computer CPU time, as a function of the relaxation parameter ω. Here we set $u_b = 1$, $u_t = 0$, and $c_1^b = c_2^b = 0.1$ mol/L. From the figure, it can be seen that the number of iterations was reduced from 36 at $\omega = 0.4$ to 15 at $\omega = 0.8$ and that the corresponding computer CPU time was reduced from 209 seconds to 86 seconds. These test results show that the convergence and performance



(a) Protein region D_p (b) Solvent region D_s (c) Membrane region D_m

FIG. 5. *The electrostatic potential u produced by the PNPic finite element solver on the triangular surface meshes of the protein, solvent, and membrane regions D_p, D_s, and D_m in color mapping from blue for -2 to red for 2.*



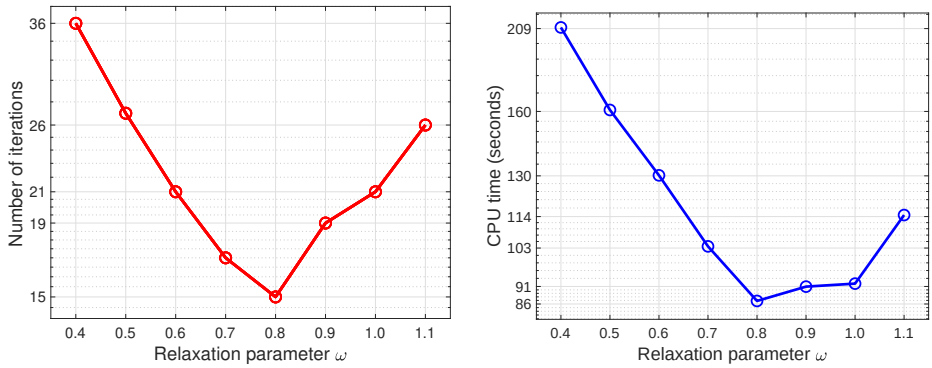
(a) Electrostatic potential u (b) K^+ concentration c_1 (c) Cl^- concentration c_2

FIG. 6. *The periodic boundary value conditions* (2.9) *well retained in the PNPic finite element solution (u, c_1, c_2). Here the color mapping ranges for u and c_i are $[-1, 1]$ and $[0, 1]$, respectively, from blue to red.*

of our relaxation iterative scheme can be improved sharply through properly selecting a relaxation parameter value.

Figure 8 reports the convergence processes of our PNPic relaxation iterative scheme. From the figure, it can be seen that the iteration errors for both $\tilde{\Phi}$ and c_i were reduced from 10^2 to 10^{-6} in 15 iterations, showing that our PNPic relaxation iterative scheme has a fast rate of convergence.

Figure 9 reports a convergence process of our Newton iterative scheme (5.12) for solving the nonlinear finite element equation of (5.9) for $\tilde{\Phi}$ at the initial iteration $k = 0$. Here the initial iterate $\tilde{\Phi}^0$ was generated by the modified Newton iterative scheme (5.14) for solving our PB ion channel model. From this figure, it can be seen that the iteration errors were reduced quickly from 10^6 to 10^{-6} in 16 iterations only. Furthermore, as the iteration number k was increased for $k \geq 1$, the total number



(a) Convergence (b) Performance

FIG. 7. *Convergence and performance of our relaxation iterative scheme* (5.6) *for solving the PNPic finite element system* (5.5) *as a function of* ω *for a GA (PDB identification code* 1*MAG) in the* 0.1 *molar KCl solution with* $u_b = 1$, *and* $u_t = 0$.

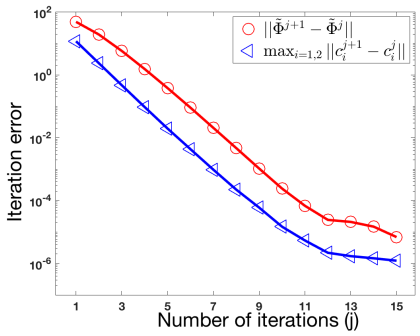


FIG. 8. *Iteration errors* $\max_{i=1,2} ||c_i^{j+1} - c_i^j||$ *and* $||\tilde{\Phi}^{j+1} - \tilde{\Phi}^j||$ *of iteration* j *for the PNPic relaxation iterative scheme defined in* (5.6) *and* (5.7) *using* $\omega = 0.8$.

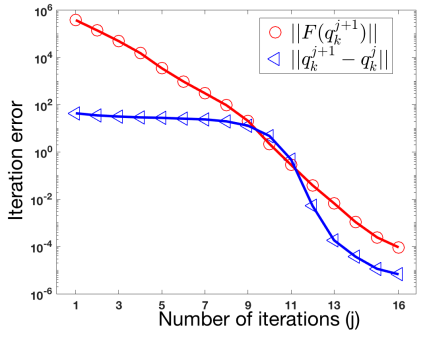


FIG. 9. *Iteration errors* $\|F(q_k^{j+1})\|$ *and* $\|q_k^{j+1} - q_k^j\|$ *of iteration* j *for Newton scheme* (5.12) *for finite element equation* $F(\tilde{\Phi}) = 0$ *of* (5.9) *at* $k = 0$.

of iterations determined by the criteria (5.11) was further reduced due to using the previous iterate $\tilde{\Phi}^k$ as the initial guess. It is this fast rate of convergence of our modified Newton iterative scheme that makes our PNPic relaxation iterative scheme particularly efficient.

Figure 10 displays the concentrations of anions Cl^- and cations K^+ and the electrostatic potential u on a cross section ($x = 0$) of the solvent region D_s in color mapping. Here we marked the membrane and protein regions in yellow and green colors, respectively, to clearly show the values in the solvent region D_s. From the figure, it can be seen that the electrostatic potential values are almost all negative (in blue) within the channel pore, repelling the anions Cl^- away from the channel pore (in blue) while attracting the cations K^+ to the channel pore (in red).

To visualize a three-dimensional concentration function as a curve across the channel pore, we construct a rectangular box domain B such that B contains the channel pore part fully. We then divide B uniformly into m sub-boxes, $\{B_j\}_{j=1}^m$, in

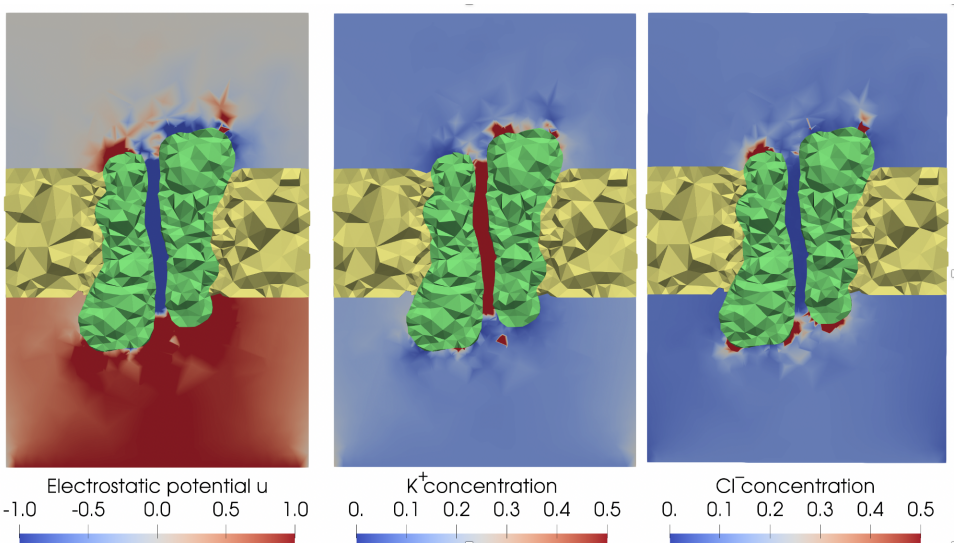


FIG. 10. *The electrostatic potential u and the concentrations c_1 and c_2 of K^+ and Cl^- ions in color mapping on a cross section ($x = 0$) of the solvent region D_s. Here the protein and membrane regions are colored in green and yellow, respectively; concentrations are in mol/L; and electrostatic potential u is in k_BT/e_c (≈ 0.0257 volts).*

TABLE 1

Parameter values for the boundary value functions g_i for $i = 1, 2$ and g defined in (6.1) and the performance of our PNPic finite element solver.

u_b	u_t	c_i^b	Iteration number	CPU time (seconds)
-1	1	0.5	15	86.10
-1	1	0.1	15	85.41
-3	3	0.5	24	140.86

the z-axis direction and calculate a volume integral as follows:

$$c_{i,j} = \int_{B_j} c_i(\mathbf{r}) d\mathbf{r}, \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, m, \tag{6.2}$$

where c_i has been set to be zero outside the solvent region D_s to ensure the definition of the above integrals. Clearly, c_{ij} gives the total amount of the ions of species i in the sub-box B_j. We next set z^j to be the z-coordinate of a midpoint of B_j to produce m points, $(z^j, c_{i,j})$ for $j = 1, \ldots, m$. Linking these points results in a curve of c_i as a function of z from z^1 to z^m. Clearly, such a curve provides us with a simple tool for visualizing the distribution of an ionic species within the channel pore. It can also be valuable for us to compare concentration functions.

We did numerical tests to study the effect of Dirichlet boundary value conditions on the concentrations c_1 and c_2. Here $B = [-1.791, 1.2125] \times [-0.8262, 1.6595] \times [-14.4, 10.6]$, and B was uniformly divided into 28 sub-boxes B_j (i.e., $m = 26$) to produce 26 points $(z_j, c_{i,j})$. We solved the PNPic model using three different boundary value functions as listed in Table 1, along with the performance data of our relaxation iterative scheme. A comparison of the concentrations is displayed in Figure 11.

Figure 11 shows that changing the boundary value function of an electrostatic potential u (i.e., changing a voltage across the membrane) has an impact on concentration functions within and near the channel pore. We also see that changing the

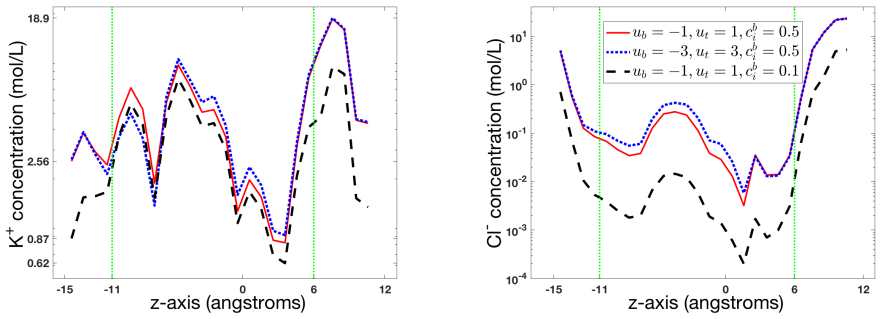


FIG. 11. *A comparison of the concentrations of K^+ and Cl^- ions within and near the channel pore ($-11 < z < 6$) generated by the PNPic model for GA (PDB identification code 1MAG) using three different boundary value functions g_i and g defined in* (6.1).

bulk concentrations c_i^b caused significant changes outside the channel pore for cations K^+ and inside the channel pore for anions Cl^-.

The test results of Figures 10 and 11 validate our PNPic model since they clearly describe the distribution patterns of cations and anions, which match the well-known fact that the GA is cation selective.

Finally, as an application of PNPic, we present a new formula for computing the electric current across the membrane and compare computed values with experimental data. It is known that the electric current I_S passing a cross section S of the channel pore can be calculated by

$$I_S = -\frac{e_c N_A}{10^3} \sum_{i=1}^{n} Z_i D_{i,c} \int_S \left[\frac{\partial c_i(\mathbf{s})}{\partial z} + Z_i c_i(\mathbf{s}) \frac{\partial u(\mathbf{s})}{\partial z} \right] d\mathbf{s} \tag{6.3}$$

provided that the normal direction of the cross section S coincides with the z-axis direction, each ionic concentration c_i is measured in mol/L, $D_{i,c}$ is a diffusion coefficient within the channel pore in Å/ps (picosecond), and the current is measured in pA (picoampere). In the steady state, I_S only varies with the cross surface S within the channel pore since both $\frac{\partial c_i(\mathbf{s})}{\partial z}$ and $\frac{\partial u(\mathbf{s})}{\partial z}$ with $\mathbf{s} = (x, y, z)$ are independent of z. In calculation, different values of I_S can be derived due to either numerical errors or S having different sizes. Thus, an average value I_{ave} of I_S is often calculated using several cross sections. However, for an irregular tetrahedral mesh of the solvent region D_s, the calculation of I_S is difficult since the calculation of a surface integral over S requires a mesh of S and an interpolation of both $\frac{\partial c_i(\mathbf{s})}{\partial z}$ and $\frac{\partial u(\mathbf{s})}{\partial z}$ onto this surface mesh, which are very difficult tasks to be done numerically. To avoid these difficulties, we present a new formula for computing I_{ave} as follows:

$$I_{ave} = -\frac{\theta}{h_B} \frac{e_c N_A}{10^3} \sum_{i=1}^{n} Z_i D_{i,b} \int_B \left[\frac{\partial c_i(\mathbf{r})}{\partial z} + Z_i c_i(\mathbf{r}) \frac{\partial u(\mathbf{r})}{\partial z} \right] d\mathbf{r}, \tag{6.4}$$

where B is a piece of the ion channel pore with height h_B in the z-axis direction, $0 < \theta \leq 1$, and $D_{i,b}$ is the diffusion coefficient of species i in the bulk solution region. Here $D_{i,c}$ has been set as $D_{i,c} = \theta D_{i,b}$.

TABLE 2

A comparison of the currents estimated by our new formula (6.4) *with the experimental data reported in* [12] *for GA (PDB identification code* 1*MAG) in a* 0.1 *molar NaCl solution. Here voltages are in mV and currents in pA.*

Voltage across the membrane	50	100	150	200
Averaged current by formula (6.4)	0.5878	1.2026	1.8430	2.5072
Experimental current reported in [12]	0.65	1.2	1.71	2.12
Relative error	0.0956	0.0022	0.0778	0.1826

In fact, since $B \approx S \times [z1, z2]$ with $z2 - z1 = h_B$, we can get that

$$\int_B \left[\frac{\partial c_i(\mathbf{r})}{\partial z} + Z_i c_i(\mathbf{r})\frac{\partial u(\mathbf{r})}{\partial z}\right] d\mathbf{r} \approx \int_{z1}^{z2}\int_S \left[\frac{\partial c_i(\mathbf{s})}{\partial z} + Z_i c_i(\mathbf{s})\frac{\partial u(\mathbf{s})}{\partial z}\right] d\mathbf{s}dz$$
$$= h_B \int_S \left[\frac{\partial c_i(\mathbf{s})}{\partial z} + Z_i c_i(\mathbf{s})\frac{\partial u(\mathbf{s})}{\partial z}\right] d\mathbf{s},$$

where we have used the fact that the surface integral is independent of z. Applying the above identity to (6.3), we show that I_{ave} is an approximation to I_S.

In the tests, we set B with the bottom surface at $z = -8$ and the top surface at $z = 2$ since the buffer size η was set as 3 (i.e., $h_B = 10$ Å), $c_i^b = 0.1$ mol/L, $\theta = 0.0245$, $u_t = 0$, and $u_b = 50, 100, 150$, and 200 mV (1 mV = 0.001 volts). The test results are reported in Table 2. From these test results, it can be seen that the currents computed by our PNPic finite element software package match well the experimental data reported in [12]. These test results further validate our PNPic model and software package.

7. Conclusions. We have presented a new PNP ion channel model using periodic boundary value conditions, called PNPic, and developed an effective finite element relaxation iterative algorithm for solving PNPic. We then implemented this PNPic finite element algorithm as a software package for the calculation of electrostatic potential density function, ionic concentration functions, and the distribution of ions and electric current within an ion channel pore. This PNPic software package works for an ion channel protein with a three-dimensional X-ray crystallographic molecular structure in an ionic solvent with multiple ionic species.

In particular, because of the periodic boundary value conditions, our PNPic model can reflect the influence of ion channels from outside a simulation box on the calculation of ionic concentrations and an electrostatic potential. Using our solution decomposition scheme, we simplify the PNPic system as a new system that does not involve any singularity and can be much easier to solve numerically so that the complexity of PNPic is reduced remarkably. We also show that the accuracy of the finite element solver can be well retained by using the Slotboom variable transformation technique. We have developed an efficient modified Newton iterative scheme for solving each nonlinear finite element equation that is generated from the Slotboom variable transformation. Through constructing proper communication operators, we have successively carried out function operations between different finite element function spaces, which are defined on different physical domains (a solvent region for ionic concentrations and a box domain for potential functions) and subject to periodic boundary constraints. As applications, we have obtained new formulas for visualizing the distribution of an ionic species within the channel pore in a simple curve (see (6.2)) and for computing the electric current passing on average a cross section of an ion channel pore (see (6.4)). Moreover, we did numerical tests on an ion channel

protein and reported the numerical results that demonstrate the convergence and performance of our PNPic finite element solver. Finally, we validated our PNPic model using the cation selectivity property of an ion channel protein and the experimental data from a chemical laboratory.

In this work, we have mainly focused on the presentation of our new PNPic model and its effective finite element solver and only reported numerical results on a small ion channel protein in a symmetric 1:1 ionic solvent. But our PNPic software package can be applied to the calculation of electrostatic potential and ionic concentrations for a large ion channel protein in ionic solvents with multiple species. It also can be used to study the various properties of our PNPic model. For example, we will study how and to what extent the periodic boundary value conditions can affect ion transport and electric current across membrane or within an ion channel pore. Moreover, our PNPic software package can be used to make various numerical experiments to justify the novelty and advantage of our PNPic model in comparison to those reported in [36, 49]. We will further improve the convergence and performance of our PNPic finite element solver using other advanced numerical techniques to make our PNPic software package a powerful tool for ion channel simulations.

Finally, it is worth noting that a repetition of one type of ion channel protein along the membrane, as done in our construction of periodic boundary value conditions, has been routinely used in state-of-the-art molecular dynamics for calculating long-range electrostatic interactions by means of a simulation box containing a single protein molecule. This treatment reduces the complexity of membrane modeling remarkably, making it possible for us to count the electrostatic interactions outside a simulation box. On the other hand, it does produce modeling errors since a real cell membrane consists of various ion channel proteins as passage conduits for different ionic species. In order to improve the reliability of our PNPic model in the calculation of electrostatics and ionic concentrations, it is important to estimate such modeling errors either theoretically or numerically via the experimental data from chemical laboratories and molecular dynamics simulations. We plan to do so in the future.

REFERENCES

[1] R. Adams and J. Fournier, *Sobolev Spaces*, 2nd ed., Pure Appl. Math. 140, Elsevier/Academic Press, Amsterdam, 2003.

[2] O. S. Andersen and R. E. Koeppe, *Molecular determinants of channel function*, Physiol. Rev., 72 (1992), pp. S89–S158.

[3] O. S. Andersen, R. E. Koeppe, and B. Roux, *Gramicidin channels*, IEEE Trans. NanoBiosci., 4 (2005), pp. 10–20.

[4] D. Boda, M. Valiskó, D. Henderson, R. S. Eisenberg, D. Gillespie, and W. Nonner, *Ionic selectivity in L-type calcium channels by electrostatics and hard-core repulsion*, J. Gen. Physiol., 133 (2009), pp. 497–509.

[5] W. M. Botello-Smith and R. Luo, *Applications of MMPBSA to membrane proteins* I: *Efficient numerical solutions of periodic Poisson-Boltzmann equation*, J. Chem. Inf. Model., 55 (2015), pp. 2187–2199.

[6] S. Brenner and L. Scott, *The Mathematical Theory of Finite Element Methods*, 3rd ed., Springer-Verlag, New York, 2008.

[7] J. H. Chaudhry, J. Comer, A. Aksimentiev, and L. N. Olson, *A stabilized finite element method for modified Poisson-Nernst-Planck equations to determine ion flow through a nanopore*, Commun. Comput. Phys., 15 (2014), pp. 93–125.

[8] D.-P. Chen, J. Lear, and R. S. Eisenberg, *Permeation through an open channel: Poisson-Nernst-Planck theory of a synthetic ionic channel*, Biophys. J., 72 (1997), pp. 97–116.

[9] M. Chen and B. Lu, *TMSMESH: A robust method for molecular surface mesh generation using a trace technique*, J. Chem. Theory Comput., 7 (2010), pp. 203–212.

[10] M. Chen, B. Tu, and B. Lu, *Triangulated manifold meshing method preserving molecular surface topology*, J. Mol. Graph. Model., 38 (2012), pp. 411–418.
[11] I. Chern, J. Liu, and W. Wang, *Accurate evaluation of electrostatics for macromolecules in solution*, Methods Appl. Anal., 10 (2003), pp. 309–328.
[12] C. D. Cole, A. S. Frost, N. Thompson, M. Cotten, T. A. Cross, and D. D. Busath, *Noncontact dipole effects on channel permeation.* VI. 5*F- and* 6*F-Trp gramicidin channel currents*, Biophys. J., 83 (2002), pp. 1974–1986.
[13] R. S. Eisenberg, *Ionic channels in biological membranes: Natural nanotubes*, Acc. Chem. Res., 31 (1998), pp. 117–123.
[14] A. Flavell, J. Kabre, and X. Li, *An energy-preserving discretization for the Poisson-Nernst-Planck equations*, J. Comput. Electron., 16 (2017), pp. 431–441.
[15] A. Flavell, M. Machen, R. S. Eisenberg, J. Kabre, C. Liu, and X. Li, *A conservative finite difference scheme for Poisson-Nernst-Planck equations*, J. Comput. Electron., 13 (2014), pp. 235–249.
[16] H. Gao and P. Sun, *A linearized local conservative mixed finite element method for Poisson-Nernst-Planck equations*, J. Sci. Comput., 77 (2018), pp. 793–817.
[17] D. Gillespie, *A review of steric interactions of ions: Why some theories succeed and others fail to account for ion size*, Microfluid. Nanofluid., 18 (2015), pp. 717–738.
[18] D. Gillespie, W. Nonner, and R. S. Eisenberg, *Coupling Poisson–Nernst–Planck and density functional theory to calculate ion flux*, J. Phys. Condens. Matter, 14 (2002), pp. 12129–12145.
[19] H. K. Gummel, *A self-consistent iterative scheme for one-dimensional steady state transistor calculations*, IEEE Trans. Electron Devices, 11 (1964), pp. 455–465.
[20] D. He and K. Pan, *An energy preserving finite difference scheme for the Poisson-Nernst-Planck system*, Appl. Math. Comput., 287 (2016), pp. 214–223.
[21] U. Hollerbach, D.-P. Chen, and R. S. Eisenberg, *Two- and three-dimensional Poisson-Nernst-Planck simulations of current flow through gramicidin A*, J. Sci. Comput., 16 (2001), pp. 373–409.
[22] T.-L. Horng, T.-C. Lin, C. Liu, and R. S. Eisenberg, *PNP equations with steric effects: A model of ion flow through channels*, J. Phys. Chem. B, 116 (2012), pp. 11422–11441.
[23] W. Im and B. Roux, *Ion permeation and selectivity of OmpF porin: A theoretical study based on molecular dynamics, Brownian dynamics, and continuum electrodiffusion theory*, J. Mol. Biol., 322 (2002), pp. 851–869.
[24] N. Ji, T. Liu, J. Xu, L. Shen, and B. Lu, *A finite element solution of lateral periodic Poisson-Boltzmann model for membrane channel proteins*, Int. J. Mol. Sci., 19 (2018), 695.
[25] R. R. Ketchem, K. C. Lee, S. Huo, and T. A. Cross, *Macromolecular structural elucidation with solid-state NMR-derived orientational constraints*, J. Biomol. NMR, 8 (1996), pp. 1–14.
[26] M. G. Kurnikova, R. D. Coalson, P. Graf, and 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 (1999), pp. 642–656.
[27] H. Liu and Z. Wang, *A free energy satisfying finite difference method for Poisson-Nernst-Planck equations*, J. Comput. Physics, 268 (2014), pp. 363–376.
[28] J.-L. Liu and R. S. Eisenberg, *Poisson-Nernst-Planck-Fermi theory for modeling biological ion channels*, J. Chem. Phys., 141 (2014), 12B640_1.
[29] P. Liu, X. Ji, and Z. Xu, *Modified Poisson-Nernst-Planck model with accurate Coulomb correlation in variable media*, SIAM J. Appl. Math., 78 (2018), pp. 226–245.
[30] T. Liu, S. Bai, B. Tu, M. Chen, and B. Lu, *Membrane-channel protein system mesh construction for finite element simulations*, Comput. Math. Biophys., 1 (2015), https://doi.org/10.1515/mlbmb-2015-0008
[31] T. Liu, M. Chen, and B. Lu, *Efficient and qualified mesh generation for Gaussian molecular surface using adaptive partition and piecewise polynomial approximation*, SIAM J. Sci. Comput., 40 (2018), pp. B507–B527.
[32] W. Liu, X. Tu, and M. Zhang, *Poisson-Nernst-Planck systems for ion flow with density functional theory for hard-sphere potential:* I–V *relations and critical potentials. Part* II: *numerics*, J. Dynam. Differential Equations, 24 (2012), pp. 985–1004.
[33] X. Liu and B. Lu, *Incorporating Born solvation energy into the three-dimensional Poisson-Nernst-Planck model to study ion selectivity in KcsA K+ channels*, Phys. Rev. E, 96 (2017), 062416.
[34] X. Liu, Y. Qiao, and B. Lu, *Analysis of the mean field free energy functional of electrolyte solution with nonhomogenous boundary conditions and the generalized PB/PNP equations*

with inhomogeneous dielectric permittivity, SIAM J. Appl. Math., 78 (2018), pp. 1131–1154.

[35] A. Logg, K.-A. Mardal, and G. N. Wells, eds., *Automated Solution of Differential Equations by the Finite Element Method*, Lect. Notes Comput. Sci. Eng. 84, Springer-Verlag, New York, 2012.

[36] B. Lu, M. J. Holst, J. A. McCammon, and Y. Zhou, *Poisson-Nernst-Planck equations for simulating biomolecular diffusion–reaction processes* I: *Finite element solutions*, J. Comput. Phys., 229 (2010), pp. 6979–6994.

[37] B. Lu and Y. C. Zhou, *Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes* II: *Size effects on ionic distributions and diffusion-reaction rates*, Biophys. J., 100 (2011), pp. 2475–2485.

[38] B. Lu, Y. C. Zhou, G. A. Huber, S. D. Bond, M. J. Holst, and J. A. McCammon, *Electrodiffusion: A continuum modeling framework for biomolecular systems with realistic spatiotemporal resolution*, J. Chem. Phys., 127 (2007), 10B604.

[39] J. A. Lundbæk, *Regulation of membrane protein function by lipid bilayer elasticity: A single molecule technology to measure the bilayer properties experienced by an embedded protein*, J. Phys. Condens. Matter, 18 (2006), pp. S1305–S1344.

[40] S. R. Mathur and J. Y. Murthy, *A multigrid method for the Poisson-Nernst-Planck equations*, Internat. J. Heat Mass Transfer, 52 (2009), pp. 4031–4039.

[41] M. S. Metti, J. Xu, and C. Liu, *Energetically stable discretizations for charge transport and electrokinetic models*, J. Comput. Phys., 306 (2016), pp. 1–18.

[42] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[43] Y. Qiao, C. Lian, B. Lu, and J. Wu, *Modeling selective ion adsorption into cylindrical nanopores*, Chem. Phys. Lett., 709 (2018), pp. 116–124.

[44] Y. Qiao, X. Liu, M. Chen, and B. Lu, *A local approximation of fundamental measure theory incorporated into three dimensional Poisson-Nernst-Planck equations to account for hard sphere repulsion among ions*, J. Statist. Phys., 163 (2016), pp. 156–174.

[45] B. Roux, T. Allen, S. Berneche, and W. Im, *Theoretical and computational models of biological ion channels*, Q. Rev. Biophys., 37 (2004), pp. 15–103.

[46] B. Roux and M. Karplus, *Molecular dynamics simulations of the gramicidin channel,* Annu. Rev. Biomol. Struct. Dyn. 23 (1994), pp. 731–761.

[47] J. W. Slotboom, *Computer-aided two-dimensional analysis of bipolar transistors*, IEEE Trans. Electron Devices, 20 (1973), pp. 669–679.

[48] S. Tanizaki and M. Feig, *A generalized Born formalism for heterogeneous dielectric environments: Application to the implicit modeling of biological membranes*, J. Chem. Phys., 122 (2005), 124706.

[49] B. Tu, M. Chen, Y. Xie, L. Zhang, R. S. Eisenberg, and B. Lu, *A parallel finite element simulator for ion transport through three-dimensional ion channel systems*, J. Comput. Chem., 34 (2013), pp. 2065–2078.

[50] B. Tu, Y. Xie, L. Zhang, and B. Lu, *Stabilized finite element methods to simulate the conductances of ion channels*, Comput. Phys. Commun., 188 (2015), pp. 131–139.

[51] G.-W. Wei, Q. Zheng, Z. Chen, and K. Xia, *Variational multiscale models for charge transport*, SIAM Rev., 54 (2012), pp. 699–754.

[52] D. Xie, *New solution decomposition and minimization schemes for Poisson-Boltzmann equation in calculation of biomolecular electrostatics*, J. Comput. Phys., 275 (2014), pp. 294–309.

[53] D. Xie, H. W. Volkmer, and J. Ying, *Analytical solutions of nonlocal Poisson dielectric models with multiple point charges inside a dielectric sphere*, Phys. Rev. E, 93 (2016), 043304.

[54] Q. Zheng, D. Chen, and G.-W. Wei, *Second-order Poisson-Nernst-Planck solver for ion transport*, J. Comput. Phys., 230 (2011), pp. 5239–5262.