

# An Efficient Finite Element Solver for a Nonuniform size-modified Poisson-Nernst-Planck Ion Channel Model

Dexuan Xie\*

May 6, 2024

## Abstract

This paper presents an efficient finite element iterative method for solving a nonuniform size-modified Poisson-Nernst-Planck ion channel (SMPNPIC) model, along with a SMPNPIC program package that works for an ion channel protein with a three-dimensional crystallographic structure and an ionic solvent with multiple ionic species. In particular, the SMPNPIC model is constructed and then reformulated by novel mathematical techniques so that each iteration of the method only involves linear boundary value problems and nonlinear algebraic systems, circumventing the numerical difficulties caused by the strong nonlinearities, strong asymmetries, and strong differential equation coupling of the SMPNPIC model. To further improve the method's efficiency, an efficient modified Newton iterative method is adapted to the numerical solution of each related nonlinear algebraic system. Numerical results for a voltage-dependent anion channel (VDAC) and a mixture solution of four ionic species demonstrate the method's convergence, the package's high performance, and the importance of considering nonuniform ion size effects. They also partially validate the SMPNPIC model by the anion selectivity property of VDAC.

## 1 Introduction

Ion channels are a class of proteins embedded in the biological membrane. They are biological devices, the “valves” of cells, and the main controllers of many biological functions, including electrical signaling in the nervous system, heart and muscle contraction, and hormone secretion [9, 29]. To compute macroscopic ion channel kinetics (e.g. Gibbs free energy, electric currents, transport fluxes, membrane potential, and electrochemical potential), several Poisson-Nernst-Planck (PNP) ion channel models were developed and solved numerically in the past decades (see [7, 10, 12, 14, 16, 18, 30] for examples). We recently improved these models by introducing novel interface and boundary conditions, piecewise diffusion and bulk concentration functions, and membrane surface charge functions. To overcome the numerical difficulties caused by the solution singularities, exponential nonlinearities, multiple physical

---

\*Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI, 53201-0413, USA ([dxie@uwm.edu](mailto:dxie@uwm.edu), <http://www.uwm.edu/~dxie/>).

domain issues, and ionic concentration positivity constraints of a PNP ion channel model, we developed mathematical and computational techniques and used them to derive effective finite element iterative methods and corresponding software packages for solving our improved PNP ion channel models for an ion channel protein with a crystallographic three-dimensional molecular structure and an ionic solvent with multiple ionic species [6, 26, 27].

However, these PNP ion channel models do not consider any ion size effect in the calculation of ionic concentrations and electrostatic potential functions. Thus, they may work poorly in applications that require distinguishing ions by size. The importance of developing an ion channel model that can distinguish ions by their sizes has been well recognized in the fields of biochemistry, biophysics, and physiology. For example, if nerve cells lose their ability to distinguish sodium ions ( $\text{Na}^+$ ) and potassium ions ( $\text{K}^+$ ), which have the same charge but different sizes, life would stop in minutes. In biophysics, it is also known that an ionic solution that treats ions as spheres has quite different entropy and energy from an ionic solution that treats ions as volume-less points. To reflect ionic size effects in the calculation of electrostatic potential and ionic concentration functions, one size-modified PNP ion channel model and a related finite element solver were reported in [13, 21, 28]. After more than a decade, they still represent the best work that appeared in the literature due to the challenges in modeling and computing for a molecular structure of an ion channel protein embedded in a cell membrane. Lamentably, as an extension of one early size-modified Poisson-Boltzmann model reported in [3], the model treats water molecules and ions as different cubes to cause voids among ions and water molecules and a modeling redundancy problem as pointed out in [24]. Its iterative solver is also costly and may have a divergence problem since it involves nonlinear boundary value problems (see [28, Eq. (33)]) and solves each related nonlinear algebraic system by a simple fix-point iterative method (see [28, the first equation of Eq. (31)]). Thus, further developing size-modified PNP ion channel models and related numerical solvers remains an important research topic.

For a solution of  $n$  ionic species, a size-modified PNP ion channel model is a nonlinear system that consists of  $n$  size-modified Nernst-Planck boundary value problems and one Poisson boundary interface problem. Its solution gives  $n$  ionic concentration functions,  $\{c_i\}_{i=1}^n$ , and one electrostatic potential function,  $u$ . Here  $c_i$  denotes an ionic concentration function of species  $i$ . Since a Nernst-Planck equation is constructed from an electrostatic solvation free energy functional,  $F$ , different selections of  $F$  may lead to different Nernst-Planck equations. A modification of an early functional  $F$  given in [3] was applied for the construction of a size-modified Nernst-Planck equation in [13, Eq. (24)]. To fix the void and redundancy problems of this early functional  $F$ , we modified this  $F$  in [23, Eq. (23)] and applied our improved functional to the construction of a size-modified Poisson-Boltzmann equation for a protein surrounded by an ionic solvent in [23] and the construction of a size-modified Poisson-Boltzmann ion channel model in [24], respectively. Using this improved functional and the modeling techniques developed in our previous work [6, 24, 26, 27], we construct a new nonuniform ion size modified PNP ion channel model in this work. For clarity, this new model will be denoted by SMPNPIC in the remainder of this paper.

Because of SMPNPIC's complexities, developing an effective SMPNPIC finite element solver, which includes both iterative methods and related software packages, is greatly challenging. One major difficulty comes from the strong coupling of the  $n$  size-modified Nernst-Planck equations, in which, each equation depends not only on the  $n$  unknown ionic con-

centration functions but also on the gradients of these unknown functions. To overcome this difficulty, one effective technique is to transform the  $n$  ionic concentration functions  $\{c_i\}_{i=1}^n$  into  $n$  new functions, denoted by  $\{\bar{c}_i\}_{i=1}^n$ , so that the  $n$  size-modified Nernst-Planck boundary value problems are transformed into the  $n$  new boundary value problems such that each new problem depends only on one of  $\{\bar{c}_i\}_{i=1}^n$  and does not involve any gradients of  $u$  and  $\{c_i\}_{i=1}^n$ . Such a function transformation technique was used in the calculation of a semiconductor steady state by Slotboom [20]. Due to this reason, each  $\bar{c}_i$  is often referred to as a Slotboom variable. This technique was applied to the SMPNP cases in [13, 28] and the PNP ion channel cases in [6, 12, 26, 27]. To enable it to work for the SMPNPIC case, we start with a modification of the  $n$  nonlinear algebraic equations given in [24, Eq. (7)] to yield the  $n$  new algebraic equations that involve  $\{\bar{c}_i\}_{i=1}^n$ ,  $\{c_i\}_{i=1}^n$ , and  $u$ . From these new equations, we obtain the required function transformation and use it to derive the  $n$  transformed problems. Combining these  $n$  transformed problems with the  $n$  new algebraic equations and a new Poisson problem, we obtain an expanded nonlinear system with  $2n + 1$  unknown functions —  $\{\bar{c}_i\}_{i=1}^n$ ,  $\{c_i\}_{i=1}^n$ , and  $u$ . To this end, we can find a solution of SMPNPIC from a solution of this expanded system.

Even so, solving this expanded system numerically is still difficult due to the singularity of  $u$  at each atomic position of an ion channel protein. We overcome this singularity difficulty by adapting the solution decomposition technique developed in our prior work [6, 24, 25, 26, 27]. That is, we decompose  $u$  as a sum of the three potential functions,  $G$ ,  $\Psi$ , and  $\tilde{\Phi}$ , which are induced from atomic charges, the potential values and charges from interfaces and boundaries, and ionic charges, respectively. Since  $G$  is given in an algebraic expression and collects all the singularity points of  $u$ , both  $\Psi$  and  $\tilde{\Phi}$  can be found by solving linear interface boundary value problems without involving any solution singularity issues. Moreover,  $\Psi$  is independent of  $\tilde{\Phi}$  and  $\{c_i\}_{i=1}^n$ . Thus, we can calculate it before the search for  $\tilde{\Phi}$  and  $\{c_i\}_{i=1}^n$ . In this way, we can reformulate the expanded system as a nonlinear system for computing  $\tilde{\Phi}$ ,  $\{c_i\}_{i=1}^n$ , and  $\{\bar{c}_i\}_{i=1}^n$  without involving any potential function singularity.

We next develop a damped block iterative method for solving this nonlinear system to reduce the computational complexity and computer memory requirement. This method stands for one major advance we made in this work. Specifically, the  $2n + 1$  unknown functions of the system are grouped into three blocks: Block 1 corresponds to  $\{\bar{c}_i\}_{i=1}^n$ , Block 2 corresponds to  $\{c_i\}_{i=1}^n$ , and Block 3 corresponds to  $\tilde{\Phi}$  in their linear finite element approximations. In Block 1, we modify each transformed boundary value problem as a linear self-adjoint problem for computing one of  $\{\bar{c}_i\}_{i=1}^n$  by substituting  $\tilde{\Phi}$  and  $\{c_i\}_{i=1}^n$  with their current iterative values, resulting in  $n$  independent linear problems, which can be solved quickly one by one per iteration of Block 1. In Block 3, we derive one linear problem from a modification of the interface boundary value problem of  $\tilde{\Phi}$  by substituting  $\{c_i\}_{i=1}^n$  with their current iterative values. Thus, each iteration of Block 3 only needs to solve one linear problem. Hence, it can also be done quickly.

However, each iteration of Block 2 is very costly since it needs to solve a system of nonlinear algebraic equations for  $\{c_i\}_{i=1}^n$ , which we derive by substituting  $\{\bar{c}_i\}_{i=1}^n$  and  $\tilde{\Phi}$  with their current iterative values. Similarly to what was done in our previous work [24], we can split this large algebraic system into  $N$  independent small subsystems with  $N$  being the number of interior mesh points of a finite element mesh of the solvent domain  $D_s$ . Since each subsystem has  $n$  unknowns only, which is typically very small (such as 2, 3, or 4) in

most application cases, we can quickly solve it by constructing a modified Newton iterative method. In this way, each iteration of Block 2 can be done fast to yield an updated iterative value of  $\{c_i\}_{i=1}^n$ .

Clearly, we can solve each related linear algebraic system either directly (e.g. by the Gaussian elimination method) or iteratively (e.g. by the generalized minimal residual method using incomplete LU preconditioning (GMRES-ILU) in our current implementation). Moreover, from the classic iterative theory [17, Section 13.5] it can imply that our method can converge at a fast rate since it only uses three blocks in its definition. Because our method only involves well-defined linear boundary value problems and well-defined nonlinear algebraic systems, we can use it to derive a finite element solution of SMPNPIC without involving any difficulties caused by the nonlinearity, asymmetry, singularity, coupling, and concentration positive requirement of SMPNPIC.

In addition, a proper selection of an initial iterate is essential to guarantee the convergence of our damped block iterative method. In [24], we report a size-modified Poisson-Boltzmann ion channel (SMPBIC) model and show that a SMPBIC solution can optimally describe an equilibrium status in the sense of minimizing an electrostatic free energy functional given in [24, Eq. (6)]. Interestingly, we observe that the SMPBIC model can be produced from SMPNPIC by substituting each function of  $\{\bar{c}_i\}_{i=1}^n$  with the corresponding bulk concentration constants, denoted by  $\{c_i^b\}_{i=1}^n$ . In this sense, the SMPBIC model can be regarded as an initial approximation of SMPNPIC. Thus, we can naturally select  $c_i^b$  as an initial guess to  $\bar{c}_i$  and a SMPBIC solution as an initial iterate of the method.

We implemented the damped block iterative method in Python and Fortran as a software package based on the state-of-the-art finite element library from the FEniCS project [11] and the software packages that we developed to solve the PNP ion channel models [6, 26, 27] and the SMPBIC models [24, 25]. We then demonstrate the performance of this software package by doing numerical tests via a crystallographic three-dimensional molecular structure of a voltage-dependent anion channel (VDAC) [2] and a mixture solution of four ionic species. This VDAC protein (PDB ID: 5XD0) is in the open state conformation and has an anion selectivity property. Thus, it gives us a good test case for validating SMPNPIC. Numerical results demonstrate a fast convergence rate of our damped block iterative method, the high performance of our software package, and the importance of considering nonuniform ion size effects. They also validate SMPNPIC by the VDAC anion selectivity property.

The rest of the paper is organized as follows. In Section 2, we present SMPNPIC. In Section 3, we reformulate SMPNPIC into the new nonlinear system. In Section 4, we construct the damped block iterative method for solving the new system. In Section 5, we present our iterative scheme for solving each related nonlinear algebraic system. In Section 6, we report the numerical test results. Conclusions are made in Section 7.

## 2 A nonuniform size-modified PNP ion channel model

In this section, we construct a nonuniform size-modified PNP ion channel model in the steady state. For clarity, we denote it by SMPNPIC. We start with a partition of an open box domain,  $\Omega$ , into a protein region,  $D_p$ , which hosts a three-dimensional molecular structure of an ion channel protein with  $n_p$  atoms, a solvent region,  $D_s$ , which contains a solution of

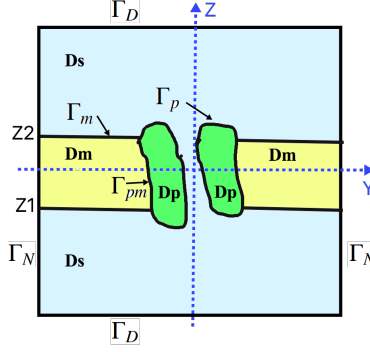


Figure 1: An illustration of the domain partition (1) and surface partition (??).

$n$  ionic species, and a membrane region,  $D_m$ , such that

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

where  $\Gamma_p$  is an interface between  $D_p$  and  $D_s$ ,  $\Gamma_m$  is an interface between  $D_m$  and  $D_s$ , and  $\Gamma_{pm}$  is an interface between  $D_p$  and  $D_m$ . In particular,  $\Omega$  is given by

$$\Omega = \{(x, y, z) \mid L_{x1} < x < L_{x2}, L_{y1} < y < L_{y2}, L_{z1} < z < L_{z2}\}, \quad (2)$$

with  $L_{x1}, L_{x2}, L_{y1}, L_{y2}, L_{z1}$ , and  $L_{z2}$  being real numbers in angstroms ( $\text{\AA}$ ), the origin of the rectangular coordinate system is set at the center of the ion channel protein, the bottom and top membrane surfaces are defined by the two plane surfaces  $z = Z1$  and  $z = Z2$ , respectively, and the normal direction of the membrane top surface coincides with the  $z$ -axis direction. See Figure 1 for an illustration.

We construct SMPNPIC as a system of  $n + 1$  partial differential equations for computing  $n$  ionic concentration functions,  $\{c_i\}_{i=1}^n$ , and a dimensionless electrostatic potential function,  $u$ . Here  $c_i$  denotes the concentration function of species  $i$ , which is defined in the solvent region  $D_s$  with the unit moles per liter (mol/L) while  $u$  is defined in the box domain  $\Omega$ . When  $u$  is given, we can get an electrostatic potential function,  $\Phi$ , in volts by

$$\Phi(\mathbf{r}) = \frac{k_B T}{e_c} u(\mathbf{r}), \quad \mathbf{r} \in \Omega,$$

where  $k_B$  is the Boltzmann constant in Joule/Kelvin,  $T$  is the absolute temperature in Kelvin, and  $e_c$  is the elementary charge in Coulomb.

Let  $\mu_i$  denote an electrochemical potential function of species  $i$ . It can be obtained as a variation of an electrostatic solvation free energy functional,  $F$ , with respect to  $c_i$ . A Nernst-Planck equation in the steady state is then defined by

$$\nabla \cdot \mathcal{D}_i(\mathbf{r}) \nabla \mu_i(\mathbf{r}) = 0, \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n, \quad (3)$$

where  $\nabla$  denotes the gradient operator (i.e.  $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$  for  $\mathbf{r} = (x, y, z)$ ) and  $\mathcal{D}_i$  is a diffusion function of species  $i$ . With an electrostatic solvation free energy functional,  $F$ , we can derive an expression of  $\mu_i$  as a variation of  $F$  with respect to  $c_i$ . Thus, different  $F$  may

lead to different Nernst-Planck equations. Using the expression of  $F$  given [24, Eq. (6)], we can derive an expression of  $\mu_i$  as follows:

$$\mu_i = k_B T \gamma \left[ Z_i u + \ln \left( \frac{c_i}{c_i^b} \right) - \frac{v_i}{v_0} \ln \left( 1 - \gamma \sum_{j=1}^n v_j c_j \right) \right], \quad i = 1, 2, \dots, n, \quad (4)$$

where  $v_i$ ,  $Z_i$ , and  $c_i^b$  denote the ion size, charge number, and bulk concentration of species  $i$ , respectively,  $v_0 = \min_{1 \leq i \leq n} v_i$ , and  $\gamma = 10^{-27} N_A$  with  $N_A$  denoting the Avogadro number. Since  $N_A$  is about  $6.02214129 \times 10^{23}$  as an estimation of the number of ions per mole, we can estimate  $\gamma$  by  $\gamma \approx 6.022 \times 10^{-4}$ . Note that all  $c_i$  and the term  $1 - \gamma \sum_{i=1}^n v_i c_i(\mathbf{r})$  are positive since the term  $\gamma \sum_{i=1}^n v_i c_i(\mathbf{r})$  gives a volume fraction of the ions in  $D_s$  while the term  $1 - \gamma \sum_{i=1}^n v_i c_i(\mathbf{r})$  gives a volume fraction of the water in  $D_s$ . Hence, the expression of  $\mu_i$  in (4) is well defined.

With (4), we can find the gradient vector  $\nabla \mu_i$  of  $\mu_i$  in the expression

$$\nabla \mu_i = k_B T \gamma \left[ Z_i \nabla u + \frac{1}{c_i} \nabla c_i + \frac{v_i}{v_0} \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} \right].$$

Substituting the above expression to (3), we derive a size-modified Nernst-Planck equation,

$$\nabla \cdot \mathcal{D}_i \left[ \nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) + \frac{v_i}{v_0} c_i(\mathbf{r}) \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} \right] = 0, \quad \mathbf{r} \in D_s, \quad (5)$$

and a size-modified flux vector,  $\mathbf{J}_i$ , of species  $i$  in the expression

$$\mathbf{J}_i(\mathbf{r}) = -\mathcal{D}_i \left[ \nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) + \frac{v_i}{v_0} c_i(\mathbf{r}) \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} \right], \quad \mathbf{r} \in D_s. \quad (6)$$

To get boundary conditions for each size-modified Nernst-Planck equation, we split the boundary  $\partial D_s$  of the solvent domain  $D_s$  into three parts — the interface part  $\Gamma_p \cup \Gamma_m$ , the bottom and top surface part  $\Gamma_D$ , and the side surface part  $\Gamma_N \cap \partial D_s$ . We then use the physical behaviors of the concentration functions on these three parts to derive three different kinds of boundary conditions as follows:

1. Robin boundary value conditions on the interface part  $\Gamma_p \cup \Gamma_m$ :

$$\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(\mathbf{s}) \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + \frac{v_i}{v_0} c_i(\mathbf{s}) \frac{\gamma \sum_{j=1}^n v_j \frac{\partial c_j(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})}}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{s})} = 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m \quad (7)$$

for  $i = 1, 2, \dots, n$ . Here,  $\mathbf{n}_s$  denotes the unit outward normal direction of the solvent region  $D_s$ ,  $\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = \nabla c_i(\mathbf{s}) \cdot \mathbf{n}_s$ , and  $\frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = \nabla u(\mathbf{s}) \cdot \mathbf{n}_s$ . The above equations come from the expression (6) of  $\mathbf{J}_i$  and the flux equations

$$\mathbf{J}_i(\mathbf{s}) \cdot \mathbf{n}_s(\mathbf{s}) = 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m, \quad i = 1, 2, \dots, n,$$

which describes that none of the ions can penetrate the interface to enter the protein and membrane regions.

2. Neumann boundary value conditions on the side surface part  $\Gamma_N \cap \partial D_s$ :

$$\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0, \quad \mathbf{s} \in \Gamma_N \cap \partial D_s, \quad i = 1, 2, \dots, n, \quad (8)$$

where  $\mathbf{n}_b$  denotes the unit outward normal direction of the box domain  $\Omega$ . The above equations indicate that none of the ions enter the solvent region  $D_s$  from the side surface of  $D_s$ .

3. Dirichlet boundary value conditions on the bottom and top surface part  $\Gamma_D$ :

$$c_i(\mathbf{s}) = g_i(\mathbf{s}), \quad \mathbf{s} \in \Gamma_D, \quad i = 1, 2, \dots, n, \quad (9)$$

where  $g_i$  is a boundary value function of species  $i$ . One simple selection of  $g_i$  is that  $g_i = c_i^b$ .

Combining each equation of (5) with the mixed boundary value conditions (7), (8), and (9), we obtain a size-modified Nernst-Planck boundary value problem as follows:

$$\left\{ \begin{array}{l} \nabla \cdot \mathcal{D}_i \left[ \nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) + \frac{v_i}{v_0} c_i(\mathbf{r}) \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} \right] = 0, \quad \mathbf{r} \in D_s, \\ \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(\mathbf{s}) \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + \frac{v_i}{v_0} c_i(\mathbf{s}) \frac{\gamma \sum_{j=1}^n v_j \frac{\partial c_j(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})}}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{s})} = 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\ \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0, \quad \mathbf{s} \in \Gamma_N \cap \partial D_s, \quad c_i(\mathbf{s}) = g_i(\mathbf{s}), \quad \mathbf{s} \in \Gamma_D, \quad i = 1, 2, \dots, n. \end{array} \right. \quad (10)$$

Following what was done in our previous work [24, 26, 24], we can obtain the following Poisson interface boundary value problem:

$$\left\{ \begin{array}{ll} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_m \Delta u(\mathbf{r}) = 0, & \mathbf{r} \in D_m, \\ -\epsilon_s \Delta u(\mathbf{r}) = \beta \sum_{i=1}^n Z_i c_i(\mathbf{r}), & \mathbf{r} \in D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_p, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_m \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_m(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_m(\mathbf{s})} + \tau \sigma, & \mathbf{s} \in \Gamma_m, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}_p(\mathbf{s})} = \epsilon_m \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}_p(\mathbf{s})}, & \mathbf{s} \in \Gamma_{pm}, \\ u(\mathbf{s}) = g(\mathbf{s}), & \mathbf{s} \in \Gamma_D, \\ \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_N, \end{array} \right. \quad (11)$$

where  $z_j$  and  $\mathbf{r}_j$  denote the charge number and position vector of atom  $j$ , respectively;  $\delta_{\mathbf{r}_j}$  is the Dirac delta distribution at  $\mathbf{r}_j$ ;  $\sigma$  denotes a membrane surface charge density function in microcoulomb per squared centimeter ( $\mu\text{C}/\text{cm}^2$ );  $u(\mathbf{s}^\pm) = \lim_{t \rightarrow 0^+} u(\mathbf{s} \pm t\mathbf{n}(\mathbf{s}))$  and  $\frac{\partial u(\mathbf{s}^\pm)}{\partial \mathbf{n}(\mathbf{s})} = \lim_{t \rightarrow 0^+} \frac{\partial u(\mathbf{s} \pm t\mathbf{n}(\mathbf{s}))}{\partial \mathbf{n}(\mathbf{s})}$  for  $\mathbf{n} = \mathbf{n}_p, \mathbf{n}_m$ ;  $\alpha$ ,  $\beta$ , and  $\tau$  are defined by

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

and  $g$  is a boundary value function. Here  $\epsilon_0$  is the permittivity of the vacuum in Farad/meter and  $e_c$  is the elementary charge in Coulomb. Using the parameter values from [22, Table 1], we can estimate the three model constants  $\alpha$ ,  $\beta$ , and  $\tau$  by

$$\alpha \approx 7042.9399, \quad \beta \approx 4.2414, \quad \tau \approx 4.392.$$

We now obtain SMPNPIC as a nonlinear system that consists of the  $n$  size-modified Nernst-Planck boundary value problems (10) and the Poisson interface boundary value problem (11). A solution of SMPNPIC gives the  $n$  ionic concentration functions  $\{c_i\}_{i=1}^n$  and electrostatic potential function  $u$ .

As a special case, by setting the ion sizes  $v_i = 0$  for  $i = 1, 2, \dots, n$  (i.e. without considering any ion size effect), the size-modified Nernst-Planck boundary value problems (10) can be reduced to the following Nernst-Planck boundary value problems:

$$\left\{ \begin{array}{ll} \nabla \cdot \mathcal{D}_i [\nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r})] = 0, & \mathbf{r} \in D_s, \\ \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} + Z_i c_i(\mathbf{s}) \frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\ \frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_N \cap \partial D_s, \\ c_i(\mathbf{s}) = g_i(\mathbf{s}), & \mathbf{s} \in \Gamma_D, \quad i = 1, 2, \dots, n. \end{array} \right. \quad (12)$$

A combination of the above problems with the Poisson problem (11) gives the PNP ion channel model reported in our previous work [26]. In this sense, SMPNPIC can be regarded as an extension of this model.

From a comparison of (10) with (12), it can be seen that each size-modified Nernst-Planck boundary value problem is not only much more strongly nonlinear and un-symmetric but also has been strongly coupled with the others. Hence, SMPNPIC is much more difficult to solve numerically than the PNP ion channel model. New mathematical and numerical techniques are required to develop an efficient iterative method for solving SMPNPIC and a related software package for computing a finite element solution of SMPNPIC on a computer.

### 3 Reformations of SMPNPIC

In this section, we present a mathematical transformation and use it to reformulate SMPNPIC as a new nonlinear system. We then introduce a solution decomposition to avoid the singularity difficulties caused by atomic charges.

Let  $c = (c_1, c_2, \dots, c_n)$  and  $\bar{c} = (\bar{c}_1, \bar{c}_2, \dots, \bar{c}_n)$ . For a given  $u$ , we construct the mathematical transformation from  $c$  into  $\bar{c}$  in two steps. In Step 1, we substitute the bulk concentration

$c_i^b$  of the expression (4) of the size-modified electrochemical potential function  $\mu_i$  with  $\bar{c}_i$ ; we then set  $\mu_i = 0$  to derive an equation of  $\bar{c}_i$  with  $c$  as follows:

$$Z_i u + \ln \left( \frac{c_i}{\bar{c}_i} \right) - \frac{v_i}{v_0} \ln \left( 1 - \gamma \sum_{j=1}^n v_j c_j \right) = 0. \quad (13)$$

In Step 2, we solve the above equation for  $\bar{c}_i$  to get the mathematical transformation

$$\bar{c}_i(\mathbf{r}) = \frac{c_i(\mathbf{r}) e^{Z_i u(\mathbf{r})}}{\left[ 1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r}) \right]^{v_i/v_0}}, \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n. \quad (14)$$

The above expression can also be written in another form of the equation of  $\bar{c}_i$  with  $c$ :

$$c_i(\mathbf{r}) - \bar{c}_i(\mathbf{r}) \left[ 1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r}) \right]^{v_i/v_0} e^{-Z_i u(\mathbf{r})} = 0. \quad (15)$$

Note that each  $\bar{c}_i$  depends on all the ionic concentration functions due to ionic size affections. But, when all  $v_i = 0$  (i.e. ignoring size affections), the above mapping is simplified as  $n$  independent mappings from  $c_i$  into  $\bar{c}_i$  as follows:

$$\bar{c}_i(\mathbf{r}) = c_i(\mathbf{r}) e^{Z_i u(\mathbf{r})}, \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n. \quad (16)$$

In this special case,  $\bar{c}_i$  is often referred to as a Slotboom variable since it was introduced by Slotboom in [20]. Due to this reason, we can refer to each function  $\bar{c}_i$  of (14) as a size-modified Slotboom variable.

With the mathematical transformation (14), we get the following theorem.

**Theorem 3.1** *The size-modified Nernst-Planck boundary value problem (10) can be reformulated by the mathematical transformation (14) as follows:*

$$\begin{cases} \nabla \cdot \hat{\mathcal{D}}_i(u, c) \nabla \bar{c}_i(\mathbf{r}) = 0, & \mathbf{r} \in D_s, \\ \frac{\partial \bar{c}_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\ \bar{c}_i(\mathbf{s}) = \bar{g}_i(\mathbf{s}), & \mathbf{s} \in \Gamma_D, \\ \frac{\partial \bar{c}_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0, & \mathbf{s} \in \Gamma_N \cap \partial D_s, \end{cases} \quad (17)$$

where  $c = (c_1, c_2, \dots, c_n)$ ,  $\hat{\mathcal{D}}_i$  is a transformed diffusion function in the expression

$$\hat{\mathcal{D}}_i(u, c) = \mathcal{D}_i(\mathbf{r}) e^{-Z_i u(\mathbf{r})} \left[ 1 - \gamma \sum_{i=1}^n v_i c_i(\mathbf{r}) \right]^{v_i/v_0}, \quad \mathbf{r} \in D_s, \quad (18)$$

and  $\bar{g}_i$  is a transformed boundary value function in the expression

$$\bar{g}_i(\mathbf{s}) = \frac{g_i(\mathbf{s}) e^{Z_i g(\mathbf{s})}}{\left[ 1 - \gamma \sum_{j=1}^n v_j g_j(\mathbf{s}) \right]^{v_i/v_0}}, \quad \mathbf{s} \in \Gamma_D. \quad (19)$$

*Proof.* Applying the gradient operation  $\nabla$  to (13), we can get

$$Z_i \nabla u(\mathbf{r}) + \frac{1}{c_i(\mathbf{r})} \nabla c_i(\mathbf{r}) - \frac{1}{\bar{c}_i(\mathbf{r})} \nabla \bar{c}_i(\mathbf{r}) + \frac{v_i}{v_0} \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} = 0.$$

Multiplying  $c_i$  on the both sides of the above equation, we can obtain

$$\nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) + \frac{v_i}{v_0} c_i(\mathbf{r}) \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} = \frac{c_i(\mathbf{r})}{\bar{c}_i(\mathbf{r})} \nabla \bar{c}_i(\mathbf{r}). \quad (20)$$

From (14) we can get the ratio  $c_i/\bar{c}_i$  in the expression

$$\frac{c_i(\mathbf{r})}{\bar{c}_i(\mathbf{r})} = e^{-Z_i u(\mathbf{r})} \left[ 1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r}) \right]^{v_i/v_0}.$$

Applying the above expression to (20), we get a relationship identity between  $c_i$  and  $\bar{c}_i$  in the equation

$$\begin{aligned} \nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) + \frac{v_i}{v_0} c_i(\mathbf{r}) \frac{\gamma \sum_{j=1}^n v_j \nabla c_j(\mathbf{r})}{1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})} \\ = e^{-Z_i u(\mathbf{r})} \left[ 1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r}) \right]^{v_i/v_0} \nabla \bar{c}_i(\mathbf{r}). \end{aligned} \quad (21)$$

Using this identity, we can transform the first equation of (10) into the first equation of (17).

We next derive the boundary conditions of (17) from those of (10).

Since  $e^{-Z_i u(\mathbf{r})} [1 - \gamma \sum_{j=1}^n v_j c_j(\mathbf{r})]^{v_i/v_0} > 0$ , we can use (21) and the Robin boundary value condition of (12) on the boundary  $\Gamma_p \cup \Gamma_m$  to get the Neumann boundary value condition

$$\frac{\partial \bar{c}_i(\mathbf{s})}{\partial \mathbf{n}_s(\mathbf{s})} = 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m, \quad i = 1, 2, \dots, n. \quad (22)$$

Using (21) and the Neumann boundary value conditions  $\frac{\partial u(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0$  on  $\Gamma_N$  and  $\frac{\partial c_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0$  on  $\Gamma_N \cap \partial D_s$ , we also can get the Neumann boundary value conditions  $\frac{\partial \bar{c}_i(\mathbf{s})}{\partial \mathbf{n}_b(\mathbf{s})} = 0$  on  $\Gamma_N \cap \partial D_s$ .

Furthermore, we use (14) and the Dirichlet boundary condition  $u(\mathbf{s}) = g(\mathbf{s})$  on  $\Gamma_D$  to transform the Dirichlet boundary condition  $c_i(\mathbf{s}) = g_i(\mathbf{s})$  on  $\Gamma_D$  of (10) as

$$\bar{c}_i(\mathbf{s}) = \bar{g}_i(\mathbf{s}), \quad (23)$$

where  $\bar{g}_i$  has been defined in (19). This completes the proof.

Furthermore, we have the following theorem.

**Theorem 3.2** *If  $u$  and  $c$  are given, then each boundary value problem of (17) becomes self-adjoint and independent each other. Moreover, it has a unique solution, which is positive.*

*Proof.* When  $u$  and  $c$  are given, the transformed diffusion function  $\widehat{\mathcal{D}}_i$  becomes a known coefficient function of the boundary value problem (17). Thus, (17) becomes a linear symmetric boundary value problem with respect to  $\bar{c}_i$ . Hence, we can use the argument from [8, page 27] to claim that this boundary value problem has a unique positive solution. This completes the proof.

Following our previous work [26], we can split the electrostatic potential function  $u$  into three potential functions,  $G$ ,  $\Psi$ , and  $\tilde{\Phi}$ , such that

$$u(\mathbf{r}) = G(\mathbf{r}) + \Psi(\mathbf{r}) + \tilde{\Phi}(\mathbf{r}) \quad \forall \mathbf{r} \in \Omega, \quad (24)$$

where  $G$  is the potential induced by atomic charges,  $\Psi$  is the potential induced by charges and potentials on boundaries and interfaces, and  $\tilde{\Phi}$  is the potential induced by ionic charges. With (24), we can reformulate the Poisson interface boundary value problem (11) to derive the algebraic expression of  $G$  and the linear interface boundary value problems that define  $\Psi$  and  $\tilde{\Phi}$  as given in [26, Eq. 13, 15, 16]. Since  $G$  contains all the singularity points of  $u$ , these two problems do not involve any singularity points of  $u$ . Moreover,  $\Psi$  is independent of any ionic concentrations,  $\{c_i\}_{i=1}^n$ . Thus, we can first calculate  $\Psi$  and then treat it as a known function during a numerical calculation for  $\{c_i\}_{i=1}^n$  and  $\tilde{\Phi}$ .

Consequently, we combine (15) with (17) and the linear interface boundary value problem for computing  $\tilde{\Phi}$  as given in [26, Eq. 16] to derive an expanded nonlinear system for computing the  $2n + 1$  unknown functions  $\tilde{\Phi}$ ,  $\{c_i\}_{i=1}^n$ , and  $\{\bar{c}_i\}_{i=1}^n$ . In this way, we have avoided the singularity difficulty caused by atomic charges.

## 4 A damped block iterative method for computing a SMPNPIC finite element solution

In this section, we construct a damped block iterative method for solving the expanded system derived in the last paragraph of Section 3. With this method, we can obtain a finite element solution of SMPNPIC without involving any numerical difficulties caused by the strong nonlinearity, asymmetry, and singularity of SMPNPIC.

Because the box domain  $\Omega$  is split into a solvent region  $D_s$ , a protein region,  $D_p$ , and a membrane region,  $D_m$ , the potential function  $u$  is defined in  $\Omega$  subject to complex interface conditions among the three subregions  $D_s$ ,  $D_p$ , and  $D_m$ . On the other hand, each ionic concentration function,  $c_i$ , is defined in the solvent domain  $D_s$  only, resulting in a two physical domain issue in the calculation of a SMPNPIC finite element solution. To overcome the numerical difficulties caused by the interface conditions and two physical domain issue, effective techniques were developed in our previous work [6, 26, 27]. In this work, we use these techniques to generate an interface fitted irregular tetrahedral mesh,  $\Omega_h$ , of  $\Omega$  and a tetrahedral mesh,  $D_{s,h}$ , of  $D_s$  such that  $D_{s,h}$  is a sub mesh of  $\Omega_h$ . We then use these two meshes to construct two linear Lagrange finite element function spaces,  $\mathcal{U}$  and  $\mathcal{V}$ , as two finite

dimensional subspaces of the Sobolev function spaces  $H^1(\Omega)$  and  $H^1(D_s)$  [1], respectively. We further define the subspaces  $\mathcal{U}_0$  and  $\mathcal{V}_0$  of  $\mathcal{U}$  and  $\mathcal{V}$  by

$$\mathcal{U}_0 = \{u \in \mathcal{U} \mid u = 0 \text{ on } \Gamma_D\}, \quad \mathcal{V}_0 = \{v \in \mathcal{V} \mid v = 0 \text{ on } \Gamma_D\}.$$

To handle the operations involving functions from both  $\mathcal{U}$  and  $\mathcal{V}$ , we use the restriction operator,  $\mathcal{R} : \mathcal{U} \rightarrow \mathcal{V}$ , and prolongation operator,  $\mathcal{P} : \mathcal{V} \rightarrow \mathcal{U}$ , given in [26].

We now can obtain a finite element variational problem of (17) as follows: Find  $\bar{c}_i \in \mathcal{V}$  satisfying  $\bar{c}_i = \bar{g}_i$  on  $\Gamma_D$  such that

$$\int_{D_s} \widehat{\mathcal{D}}_i(u, c) \nabla \bar{c}_i(\mathbf{r}) \nabla v_i(\mathbf{r}) d\mathbf{r} = 0 \quad \forall v_i \in \mathcal{V}_0, \quad i = 1, 2, \dots, n, \quad (25)$$

where  $\bar{g}_i$  is given in (23),  $\widehat{\mathcal{D}}_i$  is defined in (18), and  $u = w + \tilde{\Phi}$  with  $w = G + \Psi$ . We also can get a linear finite element variational problem of the problem given in [26, Eq. 16] as follows: Find  $\tilde{\Phi} \in \mathcal{U}_0$  such that

$$a(\tilde{\Phi}, v) = \beta \sum_{j=1}^n Z_j \int_{D_s} \mathcal{P} c_j(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} \quad \forall v \in \mathcal{U}_0, \quad (26)$$

where  $a(\cdot, \cdot)$  denotes a bilinear form defined by

$$a(\tilde{\Phi}, v) = \epsilon_p \int_{D_p} \nabla \tilde{\Phi} \cdot \nabla v d\mathbf{r} + \epsilon_m \int_{D_m} \nabla \tilde{\Phi} \cdot \nabla v d\mathbf{r} + \epsilon_s \int_{D_s} \nabla \tilde{\Phi} \cdot \nabla v d\mathbf{r}. \quad (27)$$

Let  $(\bar{c}^k, c^k, \tilde{\Phi}^k)$  denote the  $k$ th iterate of our damped three-block iterative method. Here  $\bar{c}^k = (\bar{c}_1^k, \bar{c}_2^k, \dots, \bar{c}_n^k)$  and  $c^k = (c_1^k, c_2^k, \dots, c_n^k)$ , which denote the  $k$ th iterates in Blocks 1 and 2, respectively, while  $\tilde{\Phi}^k$  being the  $k$ th iterate in Block 3. We assume that  $\Psi$  has been found by solving a linear finite element equation given in [26, Eq. 29]. For clarity, we set  $w = G + \Psi$ . Here the algebraic expression of  $G$  has been given in [26, Eq. 13]. For a given initial iterate  $(\bar{c}^0, c^0, \tilde{\Phi}^0)$  with  $k = 0$ , we calculate the  $(k+1)$ th iterate  $(\bar{c}^{k+1}, c^{k+1}, \tilde{\Phi}^{k+1})$  in the following four steps:

**Step 1.** Define the first block update  $\bar{c}^{k+1}$  by

$$\bar{c}^{k+1}(\mathbf{r}) = \bar{c}^k(\mathbf{r}) + \omega[\bar{p}(\mathbf{r}) - \bar{c}^k(\mathbf{r})], \quad \mathbf{r} \in D_s, \quad (28)$$

where  $\omega$  is a damping parameter with  $\omega \in (0, 1)$  and  $\bar{p} = (\bar{p}_1, \bar{p}_2, \dots, \bar{p}_n)$  with  $\bar{p}_i \in \mathcal{V}$  being a solution of the linear finite element variational problem: Find  $\bar{p}_i \in \mathcal{V}$  satisfying  $\bar{p}_i = \bar{g}_i$  on  $\Gamma_D$  such that

$$\int_{D_s} \widehat{\mathcal{D}}_i(w + \tilde{\Phi}^k, c^k) \nabla \bar{p}_i \nabla v_i d\mathbf{r} = 0 \quad \forall v_i \in \mathcal{V}_0, \quad i = 1, 2, \dots, n, \quad (29)$$

which is a modification of (25) by substituting  $w + \tilde{\Phi}^k$  to  $u$  and  $c^k$  to  $c$ . Since the above  $n$  problems can be solved one by one, independently, Block 1 has been decoupled into  $n$  sub-blocks.

**Step 2.** Define the second block update  $c^{k+1}$  by

$$c^{k+1}(\mathbf{r}) = c^k(\mathbf{r}) + \omega [p(\mathbf{r}) - c^k(\mathbf{r})], \quad \mathbf{r} \in D_s, \quad (30)$$

where  $p = (p_1, p_2, \dots, p_n)$  with  $p_i \in \mathcal{V}$  satisfying the nonlinear algebraic system:

$$p_i(\mathbf{r}) - \bar{c}_i^{k+1}(\mathbf{r}) \left[ 1 - \gamma \sum_{j=1}^n v_j p_j(\mathbf{r}) \right]^{\frac{v_i}{v_0}} e^{-Z_i \mathcal{R}[w(\mathbf{r}) + \tilde{\Phi}^k(\mathbf{r})]} = 0, \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n, \quad (31)$$

which is a modification of (15) by substituting  $w + \tilde{\Phi}^k$  to  $u$  and  $\bar{c}_i^{k+1}$  to  $\bar{c}_i$ .

**Step 3.** Define the third block update  $\tilde{\Phi}^{k+1}$  by

$$\tilde{\Phi}^{k+1}(\mathbf{r}) = \tilde{\Phi}^k(\mathbf{r}) + \omega [q(\mathbf{r}) - \tilde{\Phi}^k(\mathbf{r})], \quad \mathbf{r} \in \Omega, \quad (32)$$

where  $q$  is a solution of the linear finite element variational problem: Find  $q \in \mathcal{U}_0$  such that

$$a(q, v) = \beta \sum_{j=1}^n Z_j \int_{D_s} \mathcal{P} c_j^{k+1}(\mathbf{r}) v d\mathbf{r} \quad \forall v \in \mathcal{U}_0, \quad (33)$$

which is a modification of (26) by substituting  $c_j^{k+1}$  to  $c_j$ .

**Step 4.** Check the convergence: If the termination rules are satisfied:

$$\|\tilde{\Phi}^{k+1} - \tilde{\Phi}^k\|_{\Omega} < \epsilon, \max_{1 \leq i \leq n} \|\bar{c}_i^{k+1} - \bar{c}_i^k\|_{D_s} < \epsilon, \max_{1 \leq i \leq n} \|c_i^{k+1} - c_i^k\|_{D_s} < \epsilon, \quad (34)$$

stop the iteration and output  $u = w + \tilde{\Phi}^{k+1}$  and  $c = c^{k+1}$  as a finite element solution of SMPNPIC; otherwise, increase  $k$  by 1 and go back to Step 1.

In (34), we set  $\epsilon = 10^{-4}$  and define the function norms by

$$\|u\|_{\Omega} = \sqrt{\int_{\Omega} u(\mathbf{r}) d\mathbf{r}} \quad \text{for } u \in \mathcal{U}, \quad \|v\|_{D_s} = \sqrt{\int_{D_s} v(\mathbf{r}) d\mathbf{r}} \quad \text{for } v \in \mathcal{V}.$$

We also set the initial iterate  $(\bar{c}^0, c^0, \tilde{\Phi}^0)$  by

$$\bar{c}^0 = c^b, \quad c^0 = \xi, \quad \tilde{\Phi}^0 = q, \quad (35)$$

where  $c^b = (c_1^b, c_2^b, \dots, c_n^b)$ ,  $\xi = (\xi_1, \xi_2, \dots, \xi_n)$  with  $\xi_i \in \mathcal{V}$ , and  $q \in \mathcal{U}_0$  such that  $(q, \xi)$  is a solution of the following nonlinear finite element variational system:

$$\xi_i(\mathbf{r}) - c_i^b \left[ 1 - \gamma \sum_{j=1}^n v_j \xi_j(\mathbf{r}) \right]^{\frac{v_i}{v_0}} e^{-Z_i \mathcal{R}[w(\mathbf{r}) + q(\mathbf{r})]} = 0, \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n, \quad (36a)$$

$$a(q, v) - \beta \sum_{j=1}^n Z_j \int_{D_s} \mathcal{P} \xi_j(\mathbf{r}) v d\mathbf{r} = 0 \quad \forall v \in \mathcal{U}_0. \quad (36b)$$

From [24] it is known that a solution of the above nonlinear system can lead to a solution of the SMPBIC model, which describes an equilibrium status of the ion channel system. Thus, the initial iterate (35) is a natural selection for our damped block iterative method. We can solve the nonlinear system (36), approximately, by using the iterative method reported in [24].

With our damped block iterative method, we now can get a finite element solution of SMPNPIC by only solving the linear finite element equations (29) and (33) and the nonlinear algebraic system (31) without involving any numerical difficulties caused by the strong nonlinearity, asymmetry, and coupling of SMPNPIC.

The efficiency of our iterative method can be improved sharply provided that the equations of (29), (33), and (31) can be solved by efficient iterative algorithms. There exist several efficient iterative methods that we can select to solve the linear finite element equations (29) and (33). In this work, we select the generalized minimal residual method using incomplete LU preconditioning (GMRES-ILU) to do so by default.

We developed an efficient modified Newton iterative scheme for solving the nonlinear algebraic system (36a) in our previous work [24, Eq. (24)]. We observed that (36a) can be changed into the nonlinear algebraic system (31) by substituting the bulk concentration constant  $c_i^b$  with the function  $\bar{c}_i^{k+1}$ . Thus, we can modify this scheme as an efficient scheme for solving (31). It is this efficient scheme that has sharply improved the efficiency of our damped block iterative method. Due to its length in description, we describe this new scheme in the next section for clarity.

## 5 An iterative method for solving each related nonlinear algebraic system

In this section, we present an iterative method for solving each nonlinear algebraic system of (31). This scheme is a modification of the modified Newton iterative scheme given in [24, Eq. (24)]. Its construction is motivated by Theorem 5.1.

**Theorem 5.1** *Let  $D_{s,h}$  be a tetrahedral mesh of the solvent domain  $D_s$  and have the  $N_h$  mesh points,  $\mathbf{r}^{(\mu)}$  for  $\mu = 1, 2, \dots, N_h$ , from  $D_s$  and the boundary surface  $\Gamma_N \cap \partial D_s$ . For a linear Lagrange finite element function space,  $\mathcal{V}$ , defined on  $D_{s,h}$ , a solution of the nonlinear algebraic system (31) can be found equivalently from solving  $N_h$  nonlinear algebraic systems as follows: For  $\mu = 1, 2, \dots, N_h$ ,*

$$p_{i,\mu} - \bar{c}_i^{k+1}(\mathbf{r}^{(\mu)}) \left[ 1 - \gamma \sum_{j=1}^n v_j p_{j,\mu} \right]^{\frac{v_i}{v_0}} e^{-Z_i \mathcal{R}[w(\mathbf{r}^{(\mu)}) + \tilde{\Phi}^k(\mathbf{r}^{(\mu)})]} = 0, \quad i = 1, 2, \dots, n, \quad (37)$$

where  $p_{i,\mu}$  denotes a numerical approximation of the function value of  $p_i(\mathbf{r})$  at  $\mathbf{r} = \mathbf{r}^{(\mu)}$ .

*Proof.* According to the definition of a linear Lagrange finite element function space, for  $p_i \in \mathcal{V}$ , there exists a set of basis functions,  $\{\phi_\mu\}_{\mu=1}^{N_h}$ , such that  $p_i$  can be expressed by

$$p_i(\mathbf{r}) = \sum_{\mu=1}^{N_h} p_{i,\mu} \phi_\mu(\mathbf{r}), \quad \mathbf{r} \in D_s, \quad i = 1, 2, \dots, n. \quad (38)$$

Thus, we only need to determine the coefficient numbers  $\{p_{i,\mu}\}_{\mu=1}^{N_h}$  in order to find the function  $p_i$ . We can obtain the nonlinear system of (37) directly from (31) by setting  $\mathbf{r} = \mathbf{r}^\mu$  and then use them to determine the coefficient numbers  $\{p_{i,\mu}\}_{\mu=1}^{N_h}$ . This completes the proof.

For clarity, we rewrite each nonlinear algebraic system of (37) in the vector form:

$$\bar{F}(P_\mu) = \mathbf{0}, \quad \mu = 1, 2, \dots, N_h, \quad (39)$$

where  $P_\mu = (p_{1,\mu}, p_{2,\mu}, \dots, p_{n,\mu})$  and  $\bar{F} = (\bar{f}_1, \bar{f}_2, \dots, \bar{f}_n)$  with  $\bar{f}_i$  being given by

$$\bar{f}_i(P_\mu) = p_{i,\mu} - \bar{c}_i^{k+1}(\mathbf{r}^{(\mu)}) \left[ 1 - \gamma \sum_{j=1}^n v_j p_{j,\mu} \right]^{\frac{v_i}{v_0}} e^{-Z_i \mathcal{R}[w(\mathbf{r}^{(\mu)}) + \bar{\Phi}^k(\mathbf{r}^{(\mu)})]}. \quad (40)$$

Since each nonlinear system of (39) has only  $n$  unknown variables,  $\{p_{i,\mu}\}_{i=1}^n$ , and  $n$  is a small number (e.g.,  $n = 2, 3$ , or  $4$ ) in applications, we can solve it by the Newton iterative method:

$$P_\mu^{j+1} = P_\mu^j + \Upsilon_j, \quad j = 0, 1, 2, \dots, \quad (41)$$

where  $P_\mu^j$  denotes the  $j$ -th iterate of the Newton method,  $\Upsilon_j$  satisfies Newton's equation

$$J(P_\mu^j) \Upsilon_j = -\bar{F}(P_\mu^j), \quad (42)$$

and the initial iterate  $P_\mu^0$  is set as  $c^k$ . Here  $J$  denotes a  $n \times n$  Jacobian matrix of  $\bar{F}$  with the  $(i, j)$ -th entry being the partial derivative  $\partial \bar{f}_i / \partial p_{j,\mu}$  for  $i, j = 1, 2, \dots, n$ , which are found in the expression

$$\frac{\partial \bar{f}_i}{\partial p_{j,\mu}} = \begin{cases} 1 + \gamma \frac{v_i^2}{v_0} \bar{c}_i^{k+1}(\mathbf{r}^{(\mu)}) \left[ 1 - \gamma \sum_{j=1}^n v_j p_{j,\mu} \right]^{\frac{v_i}{v_0}-1} e^{-Z_i \mathcal{R}[w(\mathbf{r}^{(\mu)}) + \bar{\Phi}^k(\mathbf{r}^{(\mu)})]}, & j = i, \\ \gamma \frac{v_i v_j}{v_0} \bar{c}_i^{k+1}(\mathbf{r}^{(\mu)}) \left[ 1 - \gamma \sum_{j=1}^n v_j p_{j,\mu} \right]^{\frac{v_i}{v_0}-1} e^{-Z_i \mathcal{R}[w(\mathbf{r}^{(\mu)}) + \bar{\Phi}^k(\mathbf{r}^{(\mu)})]}, & j \neq i. \end{cases}$$

To avoid a possible numerical overflow problem in the implementation, we do a modification to the above expression as follows: For an upper bound,  $M$ , a value of  $-Z_i [w(\mathbf{r}^{(\mu)}) + \bar{\Phi}^k(\mathbf{r}^{(\mu)})]$  is truncated as  $M$  if the value is more than  $M$ . By default, we set  $M = 45$ . We then can solve each Newton equation of (42) directly by the Gaussian elimination method.

As soon as  $\|\Upsilon_j\| < \epsilon$ , we terminate the Newton iterative process, output  $P_\mu^{j+1}$  as a numerical solution of (39), and construct a numerical solution of the nonlinear algebraic system (31) by (38). By default, we set  $\epsilon = 10^{-8}$ .

## 6 SMPNPIC software package and numerical results

We developed a software package for computing a finite element solution of SMPNPIC in Python and Fortran based on the state-of-the-art finite element library from the FEniCS project [11] and our finite element program packages reported in [24, 25, 26]. This new

package includes the ion channel finite element mesh program package reported in [5] to generate all the required meshes. It also includes the Gaussian elimination method and the GMRES-ILU method as needed to solve each related linear finite element equation. By default, we select the GMRES-ILU method using the absolute and relative residual error tolerances as  $10^{-8}$ . We implemented the fast iterative scheme given in Section 5 to solve each nonlinear algebraic system of (31).

In this software package, each diffusion function,  $\mathcal{D}_i$ , is set as a differentiable function given in [26, Eq. (9)]:

$$\mathcal{D}_i(\mathbf{r}) = \begin{cases} \mathcal{D}_{i,b}, & z < Z1 \text{ or } z > Z2, \\ \mathcal{D}_{i,c} + (\mathcal{D}_{i,c} - \mathcal{D}_{i,b})\mathcal{I}_t(\mathbf{r}), & Z2 - \eta \leq z \leq Z2, \\ \mathcal{D}_{i,c}, & Z1 + \eta \leq z \leq Z2 - \eta, \\ \mathcal{D}_{i,c} + (\mathcal{D}_{i,c} - \mathcal{D}_{i,b})\mathcal{I}_b(\mathbf{r}), & Z1 \leq z \leq Z1 + \eta, \end{cases} \quad (43)$$

where  $\mathcal{D}_{i,b}$  denotes a bulk diffusion constant of species  $i$ ,  $\mathcal{D}_{i,c}$  is a parameter for controlling a diffusion-limited conduction (or current) rate,  $\mathcal{I}_b$  and  $\mathcal{I}_t$  are two interpolation functions given in [21, eq. (27)], and  $\eta$  is a parameter for adjusting the buffering region size. We also used the boundary value functions  $g_i$  and  $g$  given in [26, Eq. (8)], from which we can get the transformed boundary value function  $\bar{g}_i$  of  $g_i$  in the piecewise expression:

$$\bar{g}_i(\mathbf{s}) = \begin{cases} \frac{c_i^b}{(1 - \gamma \sum_{j=1}^n v_j c_j^b)^{v_i/v_0}} e^{Z_i u_b} & \text{at } z = L_{z1} \text{ (bottom surface),} \\ \frac{c_i^b}{(1 - \gamma \sum_{j=1}^n v_j c_j^b)^{v_i/v_0}} e^{Z_i u_t} & \text{at } z = L_{z2} \text{ (top surface),} \end{cases} \quad (44)$$

where  $u_b$  and  $u_t$  denote the extracellular and intracellular potentials, respectively, and  $c_i^b$  a bulk concentration of species  $i$  for  $i = 1, 2, \dots, n$ .

To demonstrate the performance of our SMPNPIC software package, we did numerical tests using a crystallographic molecular structure of a voltage-dependent anion channel (VDAC) and a mixture of two salts, NaCl (sodium chloride) and KNO<sub>3</sub> (potassium nitrate), with four ionic species, Cl<sup>-</sup>, NO<sub>3</sub><sup>-</sup>, Na<sup>+</sup>, and K<sup>+</sup>. VDAC is the most abundant protein on the outer mitochondrial membrane as the main conduit for the entry and exit of ionic species, playing a crucial role in regulating cell survival and cell death and characterizing health and diseases [4, 15, 19]. We got a protein data bank (PDB) file, 5XD0.pdb, of VDAC from the Orientations of Proteins in Membranes (OPM) database (<https://opm.phar.umich.edu>) since in OPM, each VDAC molecular structure has been transformed to satisfy our assumptions illustrated in Figure 1. From the PDB file, we got the membrane location numbers  $Z1 = -11.5$  and  $Z2 = 11.5$ . We modified the PDB file as a PQR file from the PDB2PQR web server (<http://nbc-222.ucsd.edu/pdb2pqr-2.1.1/>) to get the data missed in 5XD0.pdb (such as hydrogen atoms, atomic charge numbers, and atomic radii). This PQR file is required by the mesh generation package to generate a triangular boundary surface mesh of the protein region  $D_p$ . VDAC has an anion-selectivity property [2], which we can use to validate SMPNPIC. Two views of a cartoon backbone representation of the VDAC molecular structure are displayed in Figure 2.

We constructed a box domain,  $\Omega$ , of (2) by using

$$L_{x1} = -59, \quad L_{x2} = 10, \quad L_{y1} = -32, \quad L_{y2} = 40, \quad L_{z1} = -45, \quad L_{z2} = 41.$$

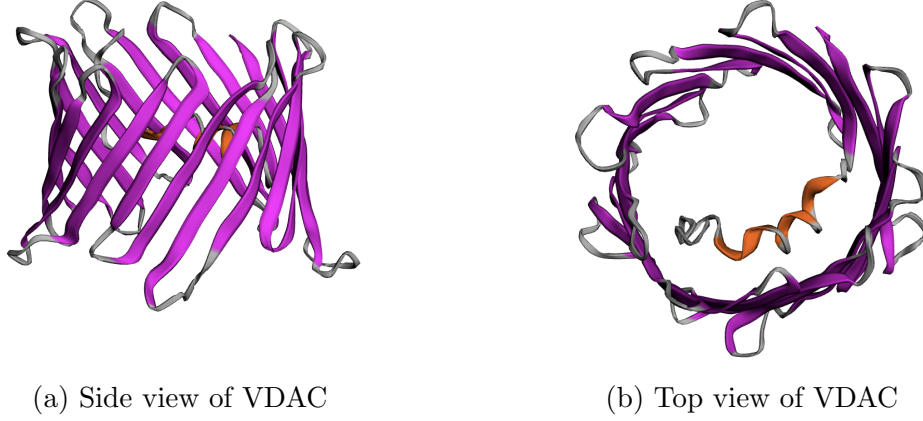


Figure 2: Two views of a crystallographic three-dimensional molecular structure of VDAC (PDB ID: 5XD0) in cartoon representations.

We then generated the interface fitted tetrahedral mesh  $\Omega_h$  of  $\Omega$ , the meshes  $D_{s,h}$ ,  $D_{p,h}$ , and  $D_{m,h}$  of the solvent region  $D_s$ , protein region  $D_p$ , and membrane region  $D_m$  and the interface surface meshes  $\Gamma_{p,h}$ ,  $\Gamma_{m,h}$ , and  $\Gamma_{pm,h}$  of the interfaces  $\Gamma_p$ ,  $\Gamma_m$ , and  $\Gamma_{pm}$ . For clarity, the set of these meshes is referred to as Mesh 1. We also generated another set of finer meshes, called Mesh 2. The major mesh data of Meshes 1 and 2 are listed in Table 1. A comparison of Meshes 1 and 2 is given in Figure 3.

	Number of vertices				Number of tetrahedra			
	$\Omega_h$	$D_{s,h}$	$D_{p,h}$	$D_{m,h}$	$\Omega_h$	$D_{s,h}$	$D_{p,h}$	$D_{m,h}$
Mesh 1	26760	11386	15506	6323	155433	46965	83702	24766
Mesh 2	38330	18672	21991	9386	227568	80312	109028	38228

Table 1: Mesh data for the two mesh sets Meshes 1 and 2.

Figure 3 shows that the box domain mesh  $\Omega_h$  has complex interfaces and the solvent domain mesh  $D_{s,h}$  has an irregular complex geometrical shape. To clearly display the complex interfaces, we have highlighted the solvent, protein, and membrane meshes,  $D_{s,h}$ ,  $D_{p,h}$ , and  $D_{m,h}$ , in grey, green, and yellow colors, respectively. From Figure 3, we can see that Mesh 2 is much more irregular and complex than Mesh 1. This implies that computing a SMPNPIC finite element solution on Mesh 2 may be more difficult and more costly than that on Mesh 1.

In the numerical tests, we used a mixed solution of 0.1 mol/L  $\text{KNO}_3$  and 0.1 mol/L  $\text{NaCl}$  and ordered the four ionic species  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$  from 1 to 4. Thus,  $c_1, c_2, c_3$ , and  $c_4$  denote the concentrations of  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$ , respectively. In this case, the bulk concentration  $c_i^b = 0.1$  mol/L for  $i = 1, 2, 3, 4$ ; the charge numbers  $Z_1 = -1, Z_2 = -1, Z_3 = 1$ , and  $Z_4 = 1$ . We estimated the ion size  $v_i$  of species  $i$  by the ball volume formula,  $v_i = 4\pi r_i^3/3$ , where  $r_i$  denotes a radius. From the website <https://bionumbers.hms.harvard.edu/bionumber.aspx?cid=108517>, we got the ionic radii of  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$  in Å as follows:

$$r_1 = 1.81, \quad r_2 = 2.64, \quad r_3 = 0.95, \quad r_4 = 1.33,$$

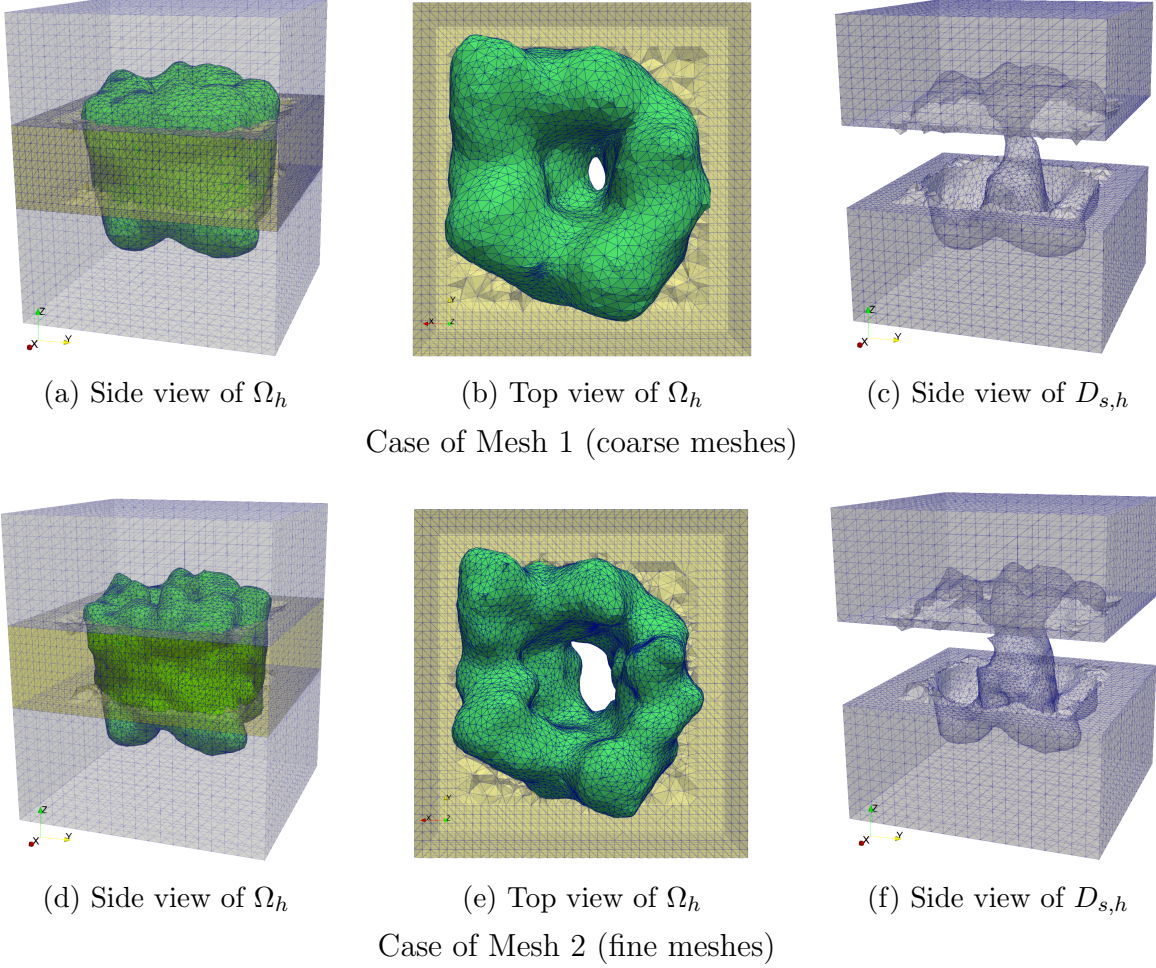


Figure 3: A comparison of the interface fitted tetrahedral box mesh  $\Omega_h$  and tetrahedral solvent domain mesh  $D_{s,h}$  in the case of Mesh 1 with those in the case of Mesh 2. Here the membrane region mesh  $D_{m,h}$  and protein region mesh  $D_{p,h}$  are colored in yellow and green, respectively, and the mesh data are given in Table 1.

from which it gives ionic sizes in  $\text{\AA}^3$  as follows:

$$v_1 = 24.8384, \quad v_2 = 77.0727, \quad v_3 = 3.5914, \quad v_4 = 9.8547.$$

This ionic solution is a good selection for us to demonstrate the importance of considering nonuniform ion sizes since it contains two anions with the same charge number  $-1$  ( $\text{Cl}^-$  and  $\text{NO}_3^-$ ) and two cations with the same charge number  $+1$  ( $\text{K}^+$  and  $\text{Na}^+$ ) while these ions have the same bulk concentration but significantly distinct sizes.

Let  $\mathcal{D}_1^b$ ,  $\mathcal{D}_2^b$ ,  $\mathcal{D}_3^b$ , and  $\mathcal{D}_4^b$  denote the bulk diffusion constants of  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$ , respectively. From the website <https://www.aqion.de/site/194>, we got

$$\mathcal{D}_1^b = 0.203, \quad \mathcal{D}_2^b = 0.190, \quad \mathcal{D}_3^b = 0.133, \quad \mathcal{D}_4^b = 0.196.$$

We then set  $\mathcal{D}_{i,c} = \theta \mathcal{D}_{i,b}$  with  $\theta = 0.055$  to get the diffusion function  $\mathcal{D}_i$  given in (43). We also set  $\epsilon_p = 2$ ,  $\epsilon_m = 2$ ,  $\epsilon_s = 80$ ,  $u_t = 0$ , and  $u_b = 0$ . All the numerical tests were done on our

Mac Studio computer, which has one Apple M1 Max chip and 64 GB memory. We report the test results in Tables 2 and 3 and Figures 4 and 5.

	$\omega = 0.30$		$\omega = 0.35$		$\omega = 0.38$		$\omega = 0.40$		$\omega = 0.41$	
	Ite	CPU	Ite	CPU	Ite	CPU	Ite	CPU	Ite	CPU
Solve each related linear algebraic system by the Gaussian elimination method										
Mesh 1	41	36.02	35	29.14	32	35.28	31	30.31	30	26.37
Mesh 2	55	88.18	46	76.96	33	58.04	37	62.93	39	62.12
Solve each related linear algebraic system by the GMRES-ILU iterative method										
Mesh 1	53	16.09	48	14.89	64	19.03	87	25.4268	77	21.65
Mesh 2	109	57.42	89	45.63	104	52.66	75	39.01	74	38.49

Table 2: Convergence and performance of the damped iterative method defined in (28) to (34). Here Ite and CPU, respectively, denote the number of iterations and computer CPU time (in seconds) determined by the iteration termination rule (34) with  $\epsilon = 10^{-4}$ .

Table 2 shows that the convergence and performance of our damped block iterative method, in terms of the number of iterations and the computer CPU time, were affected by the mesh size, the damping parameter  $\omega$ , and the iterative errors produced from the GMRES-ILU method. It also compares the convergence and performance of our damped block iterative method using the Gaussian elimination method with that using the GMRES-ILU method. Even though the iterative errors of the GMRES-ILU method disturbed the convergence rate of our damped block iterative method and caused the number of iterations to fluctuate as the damping parameter  $\omega$  was increased from 0.30 to 0.41, our damped block iterative method using the GMRES-ILU method still reduced the total CPU time significantly. For example, in the test using Mesh 2 and  $\omega = 0.41$ , the total CPU time was reduced from about 62 seconds to about 38 seconds in spite of that the number of iterations was raised from 39 to 74. Furthermore, from Table 3, we can see that the GMRES-ILU method is more efficient than the Gaussian elimination method in the three major parts of our damped block iterative method. These test results motivated us to set the GMRES-ILU iterative method as the default linear solver in our SMPNPIC software package.

	Calculate $\Psi$		Solve (36) for $\tilde{\Phi}^0$ and $c^0$		Calculate $\bar{c}$ , $\tilde{\Phi}$ and $c$	
	GMRES	Direct	GMRES	Direct	GMRES	Direct
Mesh 1	0.28	2.28	0.19	2.20	21.65	26.37
Mesh 2	0.42	3.11	0.27	3.06	38.49	62.12

Table 3: The CPU time distribution (in seconds) over the three main parts of the damped iterative method. Here, GMRESS denotes the damped iterative method using the GMRES-ILU method, Direct denotes that using the Gaussian elimination method, and  $\omega = 0.41$ .

Figure 4 displays the concentrations of four ionic species  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{K}^+$ , and  $\text{Na}^+$  in color mapping on the cross-section ( $y = 0$ ) of the solvent region  $D_s$ . Here the membrane and protein areas are colored yellow and green, respectively. From Figure 4, we see that the anions ( $\text{Cl}^-$  and  $\text{NO}_3^-$ ) are mostly attracted into the channel pore of VDAC while the cations ( $\text{K}^+$

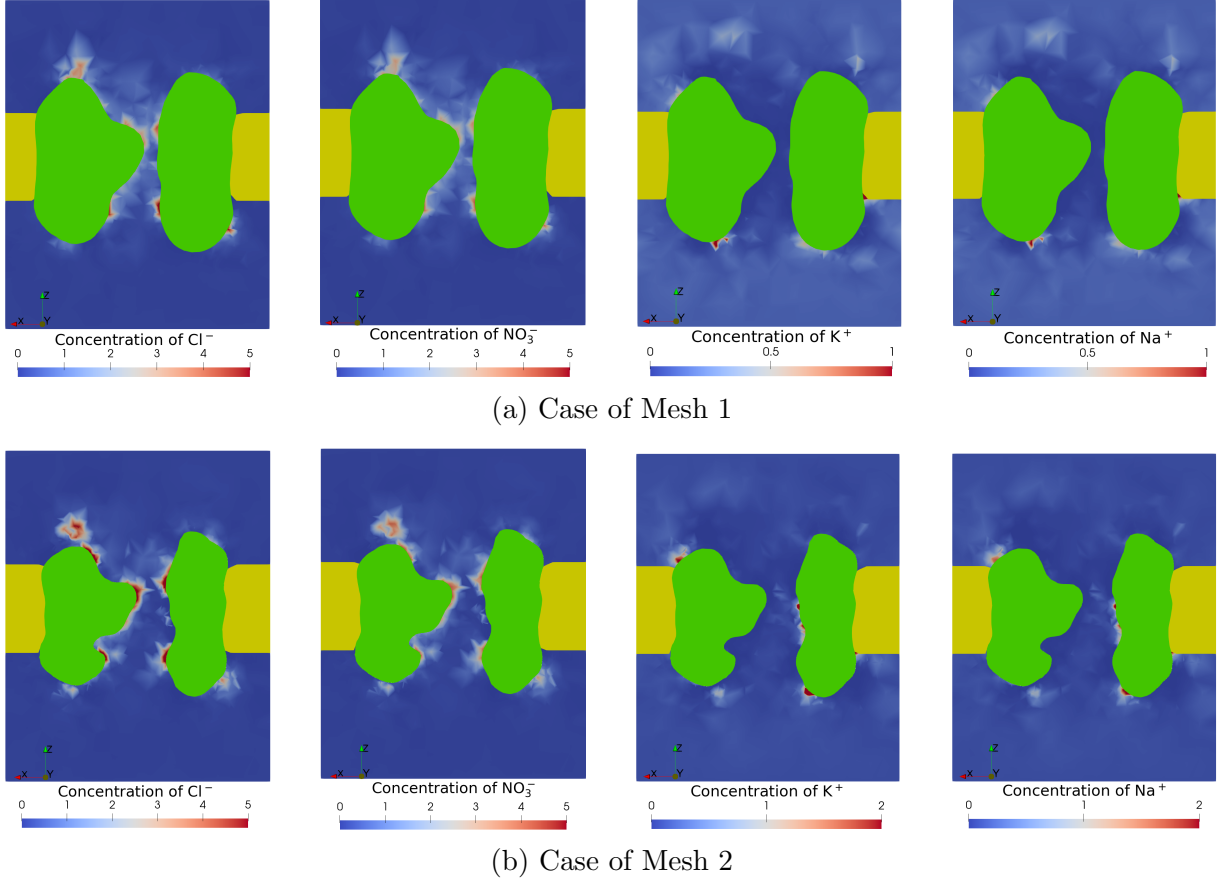


Figure 4: A color mapping of the ionic concentrations generated by our SMPNPIC software package on a cross-section ( $y = 0$ ) of the solvent region  $D_s$  for the VDAC (PDB ID: 5XD0) and the solution of four ionic species  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$ .

and  $\text{Na}^+$ ) are expelled away from the channel pore, confirming that our SMPNPIC can retain the anion selectivity property of VDAC. From Figure 4, we also see that the concentration of chloride ions ( $\text{Cl}^-$ ) has larger values than that of nitrate ions ( $\text{NO}_3^-$ ) within the ion channel pore area since a chloride ion is stronger in competition for space than a nitrate ion due to its smaller size (24.84 vs. 77.07 in  $\text{\AA}^3$ ). We further see that the concentration of potassium ions ( $\text{K}^+$ ) has larger values than that of sodium ions ( $\text{Na}^+$ ) since a potassium ion is harder to be expelled away from the ion channel pore than a sodium ion due to its larger size (9.85 vs. 3.59 in  $\text{\AA}^3$ ). These test results indicate that SMPNPIC can well reflect ion size effects for both anions and cations.

Figure 5 displays the 2D mapping curves of the concentrations of  $\text{Cl}^-$ ,  $\text{NO}_3^-$ ,  $\text{Na}^+$ , and  $\text{K}^+$ , which we produced by using the 2D mapping scheme in [26]. Such a 2D curve visualizes the distribution profile of an ionic species within the solvent domain  $D_s$ . Here the  $z$ -axis direction coincides with the channel pore direction; the channel pore area is between  $z = -28$  and  $z = 23$ ; the membrane area is between  $z = Z1$  and  $z = Z2$  with  $Z1 = -11.5$  and  $Z2 = 11.5$ ; the central part of the channel pore that links to the membrane and the other parts of the channel pore are highlighted in light-cyan and green, respectively. From Figure 5, we can

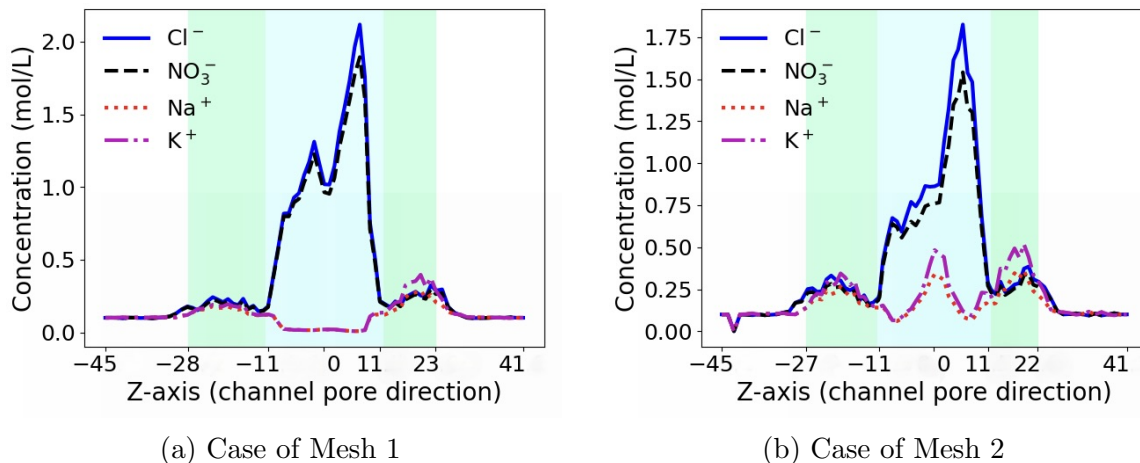


Figure 5: Mesh size influence on the ionic concentrations calculated by SMPNPIC.

clearly show that the concentrations of  $\text{Cl}^-$  and  $\text{K}^+$  have larger values than those of  $\text{NO}_3^-$  and  $\text{Na}^+$ , especially within the central part of the channel pore, due to ionic size effects. These test results confirm that ionic sizes have significant impacts on the values of ionic concentrations. They also indicate that the anion selectivity happens mostly within the central part of an ion channel pore.

## 7 Conclusions

In this work, we have introduced a nonuniform ion size-modified Poisson-Nernst-Planck ion channel model, referred to as SMPNPIC, and developed an efficient SMPNPIC finite element iterative method and a corresponding software package for an ion channel protein with a three-dimensional crystallographic structure and a solution with multiple ionic species. The numerical tests conducted on a voltage-dependent anion-channel (VDAC) and a solution with four ionic species demonstrate the fast convergence of the method and the high performance of the package. Moreover, these tests demonstrate the importance of considering nonuniform ion size effects and validate SMPNPIC by the anion selectivity property of VDAC.

SMPNPIC is a system of ion size-modified Nernst-Planck equations and Poisson equations for computing multiple ionic concentration functions and one electrostatic potential function. However, due to these equations' strong nonlinearity, asymmetry, and coupling, solving SMPNPIC numerically is much more challenging than any current Poisson-Nernst-Planck ion channel models. In this work, we have overcome these difficulties through developing effective mathematical and computational techniques while overcoming the other numerical difficulties caused by solution singularities, exponential nonlinearities, multiple physical domains, and ionic concentration positivity requirements by adapting the techniques developed in our previous work. As a result, we can find a SMPNPIC finite element solution by only solving linear boundary value problems and nonlinear algebraic equation systems. Moreover, we have developed efficient iterative methods for solving these linear problems and nonlinear algebraic systems, sharply improving the efficiency of our SMPNPIC finite element solver.

In the future, we will further improve the efficiency and usage of our SMPNPIC software

package and extend its application to the calculation of various ion channel kinetics, such as Gibbs free energy change, membrane potential, and electric currents. These developments will turn our package into a highly valuable tool for biochemists, biophysicists, and medical scientists to study and simulate ion channel proteins.

## Acknowledgements

This work was partially supported by the National Science Foundation, USA, through award number DMS-2153376, and the Simons Foundation, USA, through research award 711776.

## References

- [1] R.A. Adams and J.J.F. Fournier. *Sobolev Spaces*, volume 140 of *Pure and Applied Mathematics*. Amsterdam). Elsevier/Academic Press, Amsterdam,, second edition, 2003.
- [2] Seung Cheol Baek, Thien-Hoang Ho, Hyun Woo Lee, Won Kyeong Jung, Hyo-Seung Gang, Lin-Woo Kang, and Hoon Kim. Improvement of enzyme activity of  $\beta$ -1, 3-1, 4-glucanase from *paenibacillus* sp. X4 by error-prone PCR and structural insights of mutated residues. *Applied microbiology and biotechnology*, 101:4073–4083, 2017.
- [3] Itamar Borukhov, David Andelman, and Henri Orland. Steric effects in electrolytes: A modified Poisson-Boltzmann equation. *Physical Review Letters*, 79(3):435–438, 1997.
- [4] Amadou KS Camara, Yirphan Zhou, Emad Tajkhorshid, and Wai-Meng Kwok. Mitochondrial VDAC1: A key gatekeeper as potential therapeutic target. *Frontiers in Physiology*, 8(460):1–18, 2017.
- [5] Zhen Chao, Sheng Gui, Benzhou Lu, and Dexuan Xie. Efficient generation of membrane and solvent tetrahedral meshes for finite element ion channel calculation. *International Journal of Numerical Analysis and Modeling*, 19(6):887–906, 2022.
- [6] Zhen Chao and Dexuan Xie. An improved Poisson-Nernst-Planck ion channel model and numerical studies on effects of boundary conditions, membrane charges, and bulk concentrations. *Journal of Computational Chemistry*, 42(27):1929–1943, 2021.
- [7] Duan-pin Chen, James Lear, and Robert S Eisenberg. Permeation through an open channel: Poisson-Nernst-Planck theory of a synthetic ionic channel. *Biophysical Journal*, 72(1):97–116, 1997.
- [8] L.C. Evans. *Partial Differential Equations*, volume 19 of *Graduate studies in mathematics*. American Mathematical Society, 1998.
- [9] Bertil Hille. *Ion Channels of Excitable Membranes*. Sinauer Associates, Sunderland, Massachusettes, 2001.

- [10] Maria G. Kurnikova, Rob D. Coalson, Peter Graf, and Abraham Nitzan. A lattice relaxation algorithm for three-dimensional Poisson-Nernst-Planck theory with application to ion transport through the Gramicidin A channel. *Biophysical Journal*, 76(2):642–56, 1999.
- [11] A. Logg, K.-A. Mardal, and G. N. Wells, editors. *Automated Solution of Differential Equations by the Finite Element Method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer Verlag, 2012.
- [12] Benzhuo Lu, Michael J Holst, J Andrew McCammon, and YC Zhou. Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes I: Finite element solutions. *Journal of computational physics*, 229(19):6979–6994, 2010.
- [13] Benzhuo Lu and YC Zhou. Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes II: size effects on ionic distributions and diffusion-reaction rates. *Biophysical Journal*, 100(10):2475–2485, 2011.
- [14] Sanjay R Mathur and Jayathi Y Murthy. A multigrid method for the Poisson-Nernst-Planck equations. *International Journal of Heat and Mass Transfer*, 52(17-18):4031–4039, 2009.
- [15] Kyle S McCommis and Christopher P Baines. The role of VDAC in cell death: Friend or foe? *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1818(6):1444–1450, 2012.
- [16] Glenn Moy, Ben Corry, Serdar Kuyucak, and Shin-Ho Chung. Tests of continuum theories as models of ion channels. I. Poisson- Boltzmann theory versus Brownian dynamics. *Biophysical Journal*, 78(5):2349–2363, 2000.
- [17] J. M. Ortega and W. C. Rheinboldt. *Iterative Solution of Nonlinear Equations in Several Variables*. Academic Press, 1970.
- [18] B Roux, T Allen, S Berneche, and W Im. Theoretical and computational models of biological ion channels. *Q Rev Biophys*, 37:15–103, 2004.
- [19] Varda Shoshan-Barmatz, Vito De Pinto, Markus Zweckstetter, Ziv Raviv, Nurit Keinan, and Nir Arbel. VDAC, a multi-functional mitochondrial protein regulating cell life and death. *Molecular aspects of medicine*, 31(3):227–285, 2010.
- [20] Jan W Slotboom. Computer-aided two-dimensional analysis of bipolar transistors. *IEEE Transactions on Electron Devices*, 20(8):669–679, 1973.
- [21] Bin Tu, Minxin Chen, Yan Xie, Linbo Zhang, Bob Eisenberg, and Benzhuo Lu. A parallel finite element simulator for ion transport through three-dimensional ion channel systems. *Journal of computational chemistry*, 34:2065–2078, 2013.
- [22] D. Xie. New solution decomposition and minimization schemes for Poisson-Boltzmann equation in calculation of biomolecular electrostatics. *J. Comput. Phys.*, 275:294–309, 2014.

- [23] Dexuan Xie. New finite element iterative methods for solving a nonuniform ionic size modified Poisson-Boltzmann equation. *International Journal of Numerical Analysis and Modeling*, 14(4-5):688–711, 2017.
- [24] Dexuan Xie. An efficient finite element iterative method for solving a nonuniform size modified Poisson-Boltzmann ion channel model. *Journal of Computational Physics*, 470:111556, December 2022.
- [25] Dexuan Xie, Said H. Audi, and Ranjan K. Dash. A size modified Poisson-Boltzmann ion channel model in a solvent of multiple ionic species: Application to VDAC. *Journal of Computational Chemistry*, 41(3):218–231, 2020.
- [26] Dexuan Xie and Zhen Chao. A finite element iterative solver for an improved PNP ion channel model by Neumann boundary condition and membrane surface charge. *Journal of Computational Physics*, 423:109915, 2020.
- [27] Dexuan Xie and Benzhuo Lu. An effective finite element iterative solver for a Poisson-Nernst-Planck ion channel model with periodic boundary conditions. Accepted, September 2020.
- [28] Yan Xie, Jie Cheng, Benzhuo Lu, and Linbo Zhang. Parallel adaptive finite element algorithms for solving the coupled electro-diffusion equations. *Computational and Mathematical Biophysics*, 1(2013):90–108, 2013.
- [29] Jie Zheng and Matthew C Trudeau. *Handbook of ion channels*. CRC Press, 2015.
- [30] Qiong Zheng, Duan Chen, and Guo-Wei Wei. Second-order Poisson-Nernst-Planck solver for ion transport. *Journal of computational physics*, 230(13):5239–5262, 2011.