Analysis on Some Basic Ion Channel Modeling Problems

Zhen Chao

University of Wisconsin-Milwaukee

Follow this and additional works at: https://dc.uwm.edu/etd

Part of the Mathematics Commons

Recommended Citation
Chao, Zhen, "Analysis on Some Basic Ion Channel Modeling Problems" (2020). Theses and Dissertations. 2477.
https://dc.uwm.edu/etd/2477

This Dissertation is brought to you for free and open access by UWM Digital Commons. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of UWM Digital Commons. For more information, please contact open-access@uwm.edu.
ANALYSIS ON SOME BASIC ION CHANNEL MODELING PROBLEMS

by

Zhen Chao

A Dissertation Submitted in
Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
in Mathematics

at
The University of Wisconsin–Milwaukee
August 2020
The modeling and simulation of ion channel proteins are essential to the study of many vital physiological processes within a biological cell because most ion channel properties are very difficult to address experimentally in biochemistry. They also generate a lot of new numerical issues to be addressed in applied and computational mathematics. In this dissertation, we mainly deal with some numerical issues that are arisen from the numerical solution of one important ion channel dielectric continuum model, Poisson-Nernst-Planck (PNP) ion channel model, based on the finite element approximation approach under different boundary conditions and unstructured tetrahedral meshes. In particular, we present the derivation of an improved PNP ion channel model using Dirichlet boundary value conditions and membrane surface charges, and obtain its variational formulations. To solve this PNP ion channel model numerically, we develop a fast finite element iterative method and program it as a software package by using effective numerical techniques. This work makes it possible for us to carry out numerical tests in order to study the affection of different boundary value conditions on the PNP numerical solutions.

To solve a PNP ion channel model by the finite element method, one important task is to generate an interface fitted unstructured tetrahedral mesh but it is very challenging to complete since the PNP ion channel model involves three physical regions – a protein region, a membrane region, and a solvent region, and the interfaces between these three regions are very complex. To address this mesh challenge, in this dissertation, we develop a new algorithm for generating a triangular surface mesh of a simulation box domain and a new algorithm for constructing a tetrahedral mesh of the membrane region, such that we can easily split a mesh of the simulation box domain into three submeshes – the meshes of
protein, membrane, and solvent regions in high quality. Remarkably, our membrane mesh generation algorithm works for an ion channel protein with an irregular ion channel pore provided that a triangular mesh of the interface between the membrane and protein regions does not have any hole. Furthermore, we implement these two new mesh algorithms based on the state-of-the-art package FEniCS, and then adapt them to one commonly-used ion channel mesh generation package.

With our PNP ion channel program package, we study the impacts of boundary value conditions, membrane surface changes, and simulation box sizes on the quality of a PNP ion channel model. Such studies are done numerically by using crystallographic molecular structures of ion channel proteins in a solution of multiple ionic species. We visualize the three dimensional electrostatic potential and ionic concentrations not only in color mapping on a cross-section of protein, membrane, or solvent region but also in two dimensional curves with curve values being the average values of potential and concentration functions over a block partition of the solvent region along the membrane normal direction.
To my parents,
my wife Jianmin,
and my daughter Aimee
# Table of Contents

List of Figures ........................................ viii

List of Tables ........................................... xi

1 Introduction ........................................... 1
   1.1 Motivation and current research ................. 1
   1.2 Dissertation outline ............................. 5

2 Possion-Nernst-Planck ion channel models .......... 6
   2.1 A simulation box domain partition .............. 6
   2.2 Poisson dielectric equation ..................... 7
   2.3 Nernst-Planck equation ........................... 9
   2.4 A Poisson-Nernst-Planck ion channel system .... 11
   2.5 Interface and boundary conditions .............. 13

3 Numerical solution of a Poisson-Nernst-Planck ion channel model .................................. 16
   3.1 A solution decomposition method .......... 16
   3.2 Variational formulation ......................... 18
   3.3 A finite element iterative solver ............ 20
   3.4 Visualization of potential and concentration functions in curves ................. 22

4 An improved ion channel finite element mesh generation scheme ........................................ 25
   4.1 Our mesh generation method .................... 25
   4.2 A new triangular surface mesh for a simulation box ........................................ 27
   4.3 A new scheme for generating membrane tetrahedral meshes .......................... 29
   4.4 Numerical results ................................ 32
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>An illustration of a domain partition of (2.2).</td>
<td>7</td>
</tr>
<tr>
<td>4.1</td>
<td>An illustration of the labels of protein, solvent, and membrane regions.</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>A cross-section view of the ion channel simulation box at $z = Z_1$. Here, the protein region is colored in green, the membrane region $D_m$ is split into two subdomains, $D_{m1}$ and $D_{m2}$, colored in yellow and red, respectively, and the channel pore area is marked in white, which belongs to the solvent region $D_s$.</td>
<td>31</td>
</tr>
<tr>
<td>4.3</td>
<td>An iterative search process for the tetrahedrons that belong to the membrane region. (a) Select a “seed” tetrahedron as the initial iteration. (b) A volume mesh generated after 16 iterations. (c) The membrane mesh generated at the end of the iteration.</td>
<td>31</td>
</tr>
<tr>
<td>4.4</td>
<td>(a) A molecular structure of a mVDAC1 ion channel (PDB ID: 3EMN). Here the $\alpha$-helix is colored in red. (b) A molecular surface mesh of mVDAC1.</td>
<td>32</td>
</tr>
<tr>
<td>4.5</td>
<td>(a, b, c) The subdomain meshes generated by Lu’s mesh package. (d, e, f) The subdomain meshes generated by our improved mesh package.</td>
<td>33</td>
</tr>
<tr>
<td>5.1</td>
<td>A cross-section representation of the ion channel-membrane system with the various regions for simulations.</td>
<td>35</td>
</tr>
<tr>
<td>5.2</td>
<td>Illustrations of gA ion channel (PDB ID: 1MAG) geometry using PyMOL visualization software, color map indicates atoms.</td>
<td>38</td>
</tr>
<tr>
<td>5.3</td>
<td>Illustrations of mVDAC1 ion channel (PDB ID: 3EMN) geometry using PyMOL visualization software, color map indicates atoms in (c, d, e, f), and the $\alpha$-helix is colored in red in (a, b).</td>
<td>40</td>
</tr>
</tbody>
</table>
5.4 Color mappings of the differences of the potentials and ionic concentrations generated by the PNPic model using Dirichlet and Neumann boundary value conditions on a cross-section $x = 0$ for $g_A$ in a salt solution of KCl (0.5 M/L) when $g_b = 1.946$ (i.e., 50mV). The protein and membrane regions are colored in green and yellow, respectively.

5.5 Differences of the potentials and ionic concentrations generated by the PNPic model using Dirichlet and Neumann boundary value conditions for $g_A$ in a salt solution of KCl (0.5 M/L) when $g_b = 1.946$ (i.e., 50mV) in 2D curves.

5.6 A comparison of the current-voltage curves generated by the PNPic model using Dirichlet and Neumann boundary value conditions for $g_A$ in a salt solution of KCl (0.5 M/L).

5.7 Difference of potentials and ionic concentrations generated by using Dirichlet and Neumann boundary value conditions. All results are generated by PNPic model for mVDAC1 in a mixture solution of KCl (0.5 M/L) and CaCl$_2$ (0.1 M/L) when $g_b = 3.892$ (i.e., 100mV).

5.8 A comparison of I-V curves for different boundary value conditions.

5.9 Differences of $u$ and $c_i$ generated by the PNPic model using Dirichlet and Neumann boundary value conditions for $g_A$ in a salt solution of KCl (0.5 M/L) on Meshes 1, 2, and 3 given in Table 5.3.

5.10 Differences of $u$ and $c_i$ generated by the PNPic model using Dirichlet and Neumann boundary value conditions for mVDAC1 in a mixture solution of KCl (0.5 M/L) and CaCl$_2$ (0.1 M/L) on Meshes 1, 2, and 3 given in Table 5.3.

5.11 Comparisons of the electrostatic potential $u$ and ion concentrations $c_i$ by PNPic model with Dirichlet boundary value conditions and membrane surface charges for $g_A$ in a salt solution of KCl (0.5 M/L) on a cross-section $x = 0$. The protein and membrane regions are colored in green and yellow, respectively. Here $g_b = 1.946$ (i.e., 50mV).

5.12 Comparisons of the electrostatic potential $u$ and concentrations $c_i$ generated by PNPic model using Dirichlet boundary value conditions and membrane surface charges for $g_A$ (Mesh 1) in a salt solution of KCl (0.5 M/L). Here $g_b = 1.946$ (i.e., 50mV).

5.13 Current-voltage curves for $g_A$ ion channel with different membrane surface charges in a salt solution of KCl (0.1 M/L).
Comparisons of the electrostatic potential $u$ and concentrations $c_i$ generated by the PNPic model using Dirichlet boundary value conditions and membrane surface charges for mVDAC1 in a mixture solution of KCl (0.1 M/L) and CaCl$_2$ (0.1 M/L). Here $g_b = 1.946$ (i.e., 50 mV).
LIST OF TABLES

2.1 Magnitudes & units of PNP parameters in SI unit system ........................... 12

5.1 Relative permittivity and diffusion coefficient constants used in our tests. ... 39
5.2 Parameter values of mesh generation .......................................................... 39
5.3 Box domain dimensions and related mesh data for the three meshes used in
numerical tests. ......................................................................................... 41
ACKNOWLEDGMENTS

I would never have been able to finish my dissertation without help from many people at the University of Wisconsin-Milwaukee (UWM). First and foremost, I want to express my deep and sincerest gratitude to my advisor Prof. Dexuan Xie for supervising my dissertation with guidance and patience and training me how to become a good researcher, I have learned a lot from him in many ways. His insightful advice and immense knowledge have been greatly supportive during the past four years. He also gave me a lot of advice and help when I was looking for a post-doctoral job. Besides my advisor, a big thanks also goes out to Prof. Ahmed Sameh for his inspired discussions.

I am taking this opportunity to thank Prof. Hans Volkmer, Prof. Peter Hinow, Prof. Chao Zhu, and Prof. Lei Wang for serving in my dissertation committee.

I’d like to thank Prof. Yimin Zou and my previous supervisor Prof. Guoliang Chen, who gave me lots of help and suggestion when I applied for the Ph.D. program at UWM. I especially want to thank Kathleen Wehrheim, Xianwei Van Harpen, Rebecca Bourn, Kimberly Lacking-Quinn, Suzanne Boyd, Jeb Willenbring, who are really nice people and provided great help during the past four years. I also would like to thank my friends Ye Guo, Lianlian Zhou, and Jingchun Chen, they helped me a lot in my daily life and made me feel happy in the United States.

My heartfelt thanks go to my wife, Jianmin Liu, for her support and love. She has put a lot of effort into taking care of the family and my daughter. Without her effort and encouragement, I can not finish my Ph.D. study. Also, I want to share this important moment with my lovely daughter, Aimee. I also would like to dedicate this research to my parents. All of them are, and will always be, my source of inspiration in my life.

Last but not the least, I would like to thank the Department of Mathematical Sciences for its services and support. I am really honored to study here, and I appreciated the support fund I received from UWM during my study.
Chapter 1

Introduction

1.1 Motivation and current research

All living things are composed of cells. Although cells have differences in structure and function, all of them have a membrane. The biological cell membrane separates the cell from its exterior environment. It is the protective barrier that prevents unwanted material from passing in the cell and plays an important function to transport materials into the cell to support necessary life functions. There are many different proteins embedded within the membrane that have various functions and structures. One important type of these proteins is the pore-forming membrane protein; because these proteins are concerned with ion transport, they are referred to as ion channels.

In living organisms, transport of ions through channel pores in cell membranes is a fundamental and important process to cell biology. It is facilitated by ion channels that utilize the ionic flux to perform various biological functions, such as blood pressure, cell-cell signaling, maintaining homeostasis, muscle contraction, etc. Moreover, the action of ion channels is responsible for most of what we perceive as reality in the form of sound, smell, sight, taste, touch, and forms the physiological basis for thought. They also represent important and challenging drug targets [50]. Therefore, well understanding of the ion channel is one of the utmost biomedical significance. Although there exist probably 300 types of ion channels [66], they all share some characteristics, such as ions pass through their channels by electrical potential and concentration gradient across the cell membrane, ion channels are selective, which means that only certain types of ions can pass through their channels, and ion channels play a role via their open and close states in response to chemical or electrical
stimuli.

The study of ion channels has been done for a long time, and great progress has been made in understanding the functions and structures of them, but there still exist lots of key open questions and basic issues [23], for example, there are many ion channels whose gating mechanisms are still very obscure. In fact, ion channel study is a multidisciplinary research including lots of fields of science, such as biology, chemical, mathematics. Thus, cooperation between different disciplines is very important, and it is necessary for us to develop new methods and tools to understand ion channel functions.

Over the past several decades, ion channel methodological progresses in X-ray crystallography and electron microscopy have led to tremendous progress in determining the structure of ion channels [14, 63] in high resolution, which can help us to better understand and capture some important properties, such as ion channel conductance, selectivity, and gating property. However, some microscopic activities that happen as ions pass through a channel, such as, conformational changes in the protein, solvation/desolvation of ions along the journey, are almost impossible to obtain by the experimental approach. Fortunately, some of the shortcomings of experimental methods can be obtained by computational approaches.

There exist a large number of computational methods for studies of ion channels. Here are three widely-used computational methods:

1) Molecular Dynamics (MD): it is a fully microscopic description with all atoms treated explicitly [8, 10]. The history of MD simulation for small biological molecules dates back to the 1970s [42]. MD simulations have been widely used to study functions of ion channels, such as ion conductance, channel gating, ion selectivity, or larger molecules, see [2, 24, 25, 40, 49] for more details. However, there exist some challenges in MD simulations, the accuracy of force fields (AMBER [70], CHARMM [30], GROMOS [60]) is not good enough due to that it is an approximation, and the effect of electronic polarization is not considered during the simulation. The overall MD simulation is too expensive to investigate ion currents, so it is very difficult to use MD simulation to run up to a long time scale of ion permeation pass through the membrane.

2) Poisson-Boltzmann (PB) equation: ion channels are very complicated systems with huge degrees of freedom, hence, they are challenging to model in atomistic detail. The PB equation is one popular continuum model for describing the electrostatic interaction around a charged molecule in ionic solution at the equilibrium state, a brief history of the PB theory can be found in [17]. In the PB model, the distribution of charges in the solution is assumed to satisfy Boltzmann distribution. A solution of the PB equation presents the
electrostatic potential and equilibrium concentration of ions in the solution. This model has been extensively used to estimate the free energy of transferring the ions from the bulk solution to the ion channel pore [29, 47, 71] and to calculate the electrostatic potential and ionic concentrations [55, 56]. Recently, a size-modified PB model has been studied for ion channels in a solution of multiple ionic species [73]. To solve the PB equation numerically, there exist several popular software packages, such as DelPhi [53], APBS [1], UHBD [39], PBSA [38], PBEQ [28], MIBPB [5], SDPBS [27]. However, these packages need to be modified significantly for modeling ion channels due to the complicated shape of the ion channel proteins.

3) Poisson-Nernst-Planck (PNP) equations: while the PB model only gives insights into the equilibrium state of an ion channel, PNP can be used to simulate the flow of ions across the biological membrane in a non-equilibrium state. It also can be used to study ion permeation events, which are expensive to obtain by using MD. Mathematical analyses of PNP equations have been studied for a long time since it was introduced by German theoretical physicists Nernst and Planck [48, 52] at the end of 19th century. The solution existence of the PNP equations was studied for the simple case in [26, 45]. For multiple ion species case, the existence and uniqueness of solution were studied in [34] for the 1D steady-state PNP system. Analytic solutions of PNP equations in the 1D case were found in [19, 20].

However, due to the nonlinearity of the strong coupled system of PNP equations, it is challenging to find the analytic solution of PNP equations in the realistic bimolecular context. Therefore, numerical methods are required to find approximate solutions in general. Several numerical techniques and algorithms for solving PNP equation were developed in the past two decades including finite difference methods [3, 4], recently, a second-order finite difference method by using matched interface and boundary method [76] has been developed to solve PNP system in gA ion channel [78]. The application of the finite difference method to the PNP system has some limitations on the description of the ionic channel structure. Finite volume method [41] is also very popular to solve PNP equations in some irregular domains, but it was still limited due to the low convergence rate. And a hybrid finite-difference/finite-volume method for solving PNP equations in irregular domains by utilizing a conservative discretization of the diffusive has been presented in [44]. Hollerbach et al. [21] simulated sodium chloride currents through the gramicidin A (gA) channel by using the spectral element method to solve PNP equations. Another important and widely used method is the finite element methods [37, 65]. Of these numerical methods, the finite element
method is the most suitable to handle irregular molecular surfaces of an ion channel protein and different types of boundary conditions. Current PNP models used Dirichlet boundary value conditions, ignored any effect of membrane surface charges. Recently, in order to reflect the affection of ions and ion channel proteins outside a simulation box, Xie and Lu [75] applied periodic boundary conditions to a PNP ion channel model. We also constructed an improved PNP ion channel model using Neumann boundary value condition and membrane surface charges in [74].

Each of the three approaches for ion channel simulations has limitations and advantages. The MD method is considered the most computationally expensive but also the most accurate. The PNP approaches are, in general, less computationally expensive but also provide less detail. They easily incorporate certain types of boundary value conditions that arise in physical systems. In addition, the accuracy of a PNP finite element solution depends on the quality of a related finite element mesh.

To solve a PNP ion channel model by the finite element method, we need an ion channel mesh generation package to generate an interface fitted unstructured tetrahedral mesh for a box simulation domain. Quite often, people give up the finite element method because they can not generate a tetrahedral mesh, which can fit well the three related physical regions, called the membrane, protein, and solvent region, respectively. Generally speaking, the first step of generating an ion channel mesh is to produce a high quality protein surface mesh. There exist several molecular surface mesh generation software packages, such as TMSmesh [6], NanoShaper [11], GAMer [77], and MolSurf [64]. Recently, in order to improve the efficiency, robustness, and mesh quality of the molecular surface mesh, Lu’s group [33] proposed an updated version of TMSmesh, TMSmesh 2.0, to handle large molecules and speedup the computer performance of TMSmesh. Then the box triangular surface mesh of the domain is needed to produce, this can be done by Gmesh [18] or FEniCS [35]. In [32], Liu et al. used the uniform mesh with the fixed density, which can not fit the interface between membrane and solvent region well, and may reduce the quality of the corresponding volume mesh. Once the triangular surface meshes of molecular and box domain are obtained, the volume mesh can be generated by the three dimensional (3D) volume mesh generation packages such as TetGen [62, 61], CGAL [15], and Gmsh [18]. Finally, the membrane region is needed to construct to simulate the effect of the membrane surface charges. In 2014, Tu et al. [67] used cylinders or balls to generate a membrane mesh manually because the structure of each ion channel is different. Recently, this method has been improved by Lu’s group [32] via the walk-detect method to yield a computer algorithm for generating the membrane mesh.
region mesh. Even so, this method works only for a single ion channel protein with the channel pore having a regular shape. It does not work well or even fails for an ion channels with a complicated channel shape.

1.2 Dissertation outline

Following what is done in [74, 75], in this dissertation, we present a finite element iterative method for solving the PNP ion channel model using Dirichlet boundary value conditions and a membrane surface charge density function. We then propose a new numerical scheme for generating a membrane mesh to improve the ion channel tetrahedron mesh generation package reported in [67, 32]. Furthermore, we implement our new PNP finite element iterative solver and new membrane mesh scheme as a software package so that we can explore some important numerical issues arisen from PNP ion channel simulations. The remaining parts of the dissertation are organized as follows:

In Chapter 2, we present the PNP ion channel model using Dirichlet boundary conditions and membrane surface charges. For clarity, this model is denoted by the PNPic model.

In Chapter 3, we follow the work in [74] to derive the variational forms related to the PNPic model and then use the novel solution decomposition method [72] to derive an efficient finite element method for solving the PNPic model approximately.

In Chapter 4, we present a new algorithm for generating a triangular surface mesh of a simulation box domain and a new algorithm for constructing a tetrahedral mesh of the membrane region. We then implement these two new mesh algorithms based on the state-of-the-art package FEniCS, and use them to modify Lu’s ion channel mesh generation package.

In Chapter 5, we present the results of numerical experiments to study the effects of boundary value conditions, simulation domain sizes, and membrane surface charges.

The conclusions and future work are given in Chapter 6.
Chapter 2

Possion-Nernst-Planck ion channel models

In this chapter, we describe Possion-Nernst-Planck ion channel (PNPic) equations and the related interface and boundary value conditions, and then derive a dimensionless PNPic model.

2.1 A simulation box domain partition

Let a rectangular box open domain \( \Omega \in \mathbb{R}^3 \) be defined by

\[
\Omega = \{(x, y, z) | L_{x_1} < x < L_{x_2}, L_{y_1} < y < L_{y_2}, L_{z_1} < x < L_{z_2}\},
\]

where \( L_{x_1}, L_{x_2}, L_{y_1}, L_{y_2}, L_{z_1}, \) and \( L_{z_2} \) are real numbers.

In order to construct the PNPic model, the domain \( \Omega \) is divided as follows:

\[
\Omega = D_p \cup D_m \cup D_s \cup \Gamma_m \cup \Gamma_p \cup \Gamma_{pm},
\]

where \( D_p \) represents a protein domain, \( D_s \) is a solvent domain, which contains a solution of ionic species, \( D_m \) is a membrane domain. \( \Gamma_m, \Gamma_p, \) and \( \Gamma_{pm} \) denote the interfaces between \( D_m \) and \( D_s, D_p \) and \( D_s, \) and \( D_p \) and \( D_m, \) respectively. We further split the interface \( \Gamma_m \) and
the boundary \( \partial \Omega \) by

\[
\Gamma_m = \Gamma_{mb} \cup \Gamma_{mt}, \quad \partial \Omega = \Gamma_D \cup \Gamma_{Ns} \cup \Gamma_{Nm},
\]

(2.3)

where \( \Gamma_{mb} \) and \( \Gamma_{mt} \) represent the bottom and top surfaces of membrane, \( \Gamma_{Ns} \) and \( \Gamma_{Nm} \) consist of the lateral surfaces of the boundary \( \partial D_s \) of \( D_s \) and \( \partial D_m \) of \( D_m \), respectively, and \( \Gamma_D \) consists of the bottom and top surfaces of \( \partial \Omega \).

A cross-section of the domain partitions (2.2) and (2.3) is shown in Figure 2.1(b). Here the normal direction of the membrane surface is set in the \( z \)-axis direction, and the membrane location is determined by two numbers \( Z_1 \) and \( Z_2 \). Figure (2.1(a)) is a view of the mesh of a box domain generated from our mesh software package while Figure (2.1(b)) is simply a diagram to illustrate the region partition and related interfaces and boundaries.

![Figure 2.1: An illustration of a domain partition of (2.2).](image)

2.2 Poisson dielectric equation

One of the cornerstones of electrostatics is setting up and solving problems described by the Poisson dielectric equation. Solving the Poisson equation amounts to finding the electric potential for a given charge distribution. Now, we give a brief description of it, the differential
form of time dependent Gauss’s law in electrostatic is:

\[ \nabla \cdot \epsilon(r) \mathbf{E}(r) = \rho(r), \]  

(2.4)

where \( \rho(r) \) is the total charge density that consists of both fixed and mobile charges, \( \epsilon(r) \) is permittivity of the medium, in reality, it maybe space-dependent [51]. \( \nabla \cdot \) is the divergence operator, and \( \mathbf{E} \) is an electric field induced by charge density \( \rho \). Generally, the electric permittivity \( \epsilon(r) \) is conventionally expressed as

\[ \epsilon(r) = \epsilon_r(r) \epsilon_0, \]

where \( \epsilon_r(r) \) is the relative permittivity, \( \epsilon_0 \) is the permittivity of vacuum.

Since \( \mathbf{E} \) is conservative, there exists an electrostatic potential function \( \phi \) such that

\[ \mathbf{E}(r) = -\nabla \phi(r). \]

Applying the above identity to (2.4), we obtain the Poisson dielectric equation as follows:

\[ -\nabla \cdot \epsilon_r(r) \nabla \phi(r) = \frac{\rho(r)}{\epsilon_0}. \]  

(2.5)

Normally, different subdomains have different charge densities. In the PNPic model, the charge density function \( \rho(r) \) is defined by

\[ \rho(r) = \begin{cases} 
0, & r \in D_m, \\
e_c \sum_{i=1}^{n} Z_i c_i(r), & r \in D_s, \\
\sum_{j=1}^{n_p} q_j \delta(r - r_j), & r \in D_p,
\end{cases} \]  

(2.6)

where \( q_j = z_j e_c \) with \( z_j \) being the valence of the \( j \)-th atom of the protein at position \( r_j \), \( e_c \) is the elementary charge, \( c_i \) is the concentration of the \( i \)-th ionic species, \( Z_i \) is the charge number of the \( i \)-th ionic species, \( n \) is the number of ionic species in the solvent, \( n_p \) is the number of atoms of the protein, and \( \delta(r - r_j) \) is the Dirac delta function [57].

Due to the membrane chemical compositions, both protein and membrane can have
different relative electrical permittivities from the relative permittivity of the solvent. So $\epsilon_r(r)$ is discontinuous across the different regions, and defined as a piecewise function by

$$
\epsilon_r(r) = \begin{cases} 
\epsilon_m, & r \in D_m, \\
\epsilon_s(r), & r \in D_s, \\
\epsilon_p, & r \in D_p,
\end{cases}
$$

(2.7)

where $\epsilon_m$ and $\epsilon_p$ are the dielectric functions in the membrane and protein regions, respectively, which can be chosen as constants, and $\epsilon_s(r)$ denotes the dielectric function of the solvent region, which can vary from the bulk region to the ion channel pore.

From (2.6), (2.7), and (2.5), the Poisson dielectric equation for ion channel cases can be defined by

$$
\begin{align*}
-\epsilon_p \Delta \phi(r) &= \frac{e_0}{\epsilon_0} \sum_{j=1}^{n_p} z_j \delta(r - r_j), & r \in D_p, \\
-\epsilon_m \Delta \phi(r) &= 0, & r \in D_m, \\
-\nabla \cdot \epsilon_s(r) \nabla \phi(r) &= \frac{e_0}{\epsilon_0} \sum_{i=1}^{n} Z_i c_i(r), & r \in D_s.
\end{align*}
$$

(2.8)

2.3 Nernst-Planck equation

The Nernst–Planck equation is a conservation of mass equation used to describe the flux of a charged chemical species in a fluid medium under the influence of both an ionic concentration gradient and an electric field. It is named after Walther Nernst and Max Planck.

Let’s start with a general derivation of the Nernst-Planck equation. Suppose there is a flux density $J_i$ of the $i$-th ionic species across the boundary $\partial \Omega_s$ of a solvent domain $\Omega_s$, then the total ionic amount flowing out the boundary is

$$
\int_{\partial \Omega_s} J_i \cdot dS.
$$

By the law of mass conservation, the above expression should be equal to the rate of decrease of the total amount of ions in the solvent domain $\Omega_s$, i.e.,

$$
- \frac{dQ_i}{dt} = \int_{\partial \Omega_s} J_i \cdot dS,
$$
where
\[ Q_i(t) = \int_{\Omega_s} c_i(t, \mathbf{r}) d\mathbf{r}, \]
c_i(t, \mathbf{r}) is the concentration of the i-th ionic species in the solvent. Thus, it has
\[ -\int_{\Omega_s} \frac{\partial c_i(t, \mathbf{r})}{\partial t} d\mathbf{r} = \int_{\partial \Omega_s} \mathbf{J}_i \cdot d\mathbf{S}. \]

Now, by the Divergence Theorem, we can get
\[ -\int_{\Omega_s} \frac{\partial c_i(t, \mathbf{r})}{\partial t} d\mathbf{r} = \int_{\Omega_s} \nabla \cdot \mathbf{J}_i d\mathbf{r}, \]
from which it implies that
\[ \frac{\partial c_i(t, \mathbf{r})}{\partial t} = -\nabla \cdot \mathbf{J}_i. \quad (2.9) \]
The above equation is the continuity equation, which states that the divergence of the flux density \( \mathbf{J}_i \) is equal to the negative rate of the ionic concentration \( c_i \) for the i-th ionic species.

When the flux density \( \mathbf{J}_i \) comes from two sources, the concentration gradient and the electric potential gradient, the effect of a concentration gradient of the i-th ionic species on ion flow can be described by Fick’s law of diffusion [16]:
\[ \mathbf{J}_{i,d}(t, \mathbf{r}) = -D_i(\mathbf{r}) \nabla c_i(t, \mathbf{r}), \quad (2.10) \]
where \( \mathbf{J}_{i,d} \) is the diffusion flux, \( D_i \) is the diffusion coefficient. And the effect of an electric potential gradient of the i-th ionic species is described by Ohm’s law of drift [46]:
\[ \mathbf{J}_{i,e}(t, \mathbf{r}) = -\mu_i \frac{q_i}{|q_i|} c_i(t, \mathbf{r}) \nabla \phi(t, \mathbf{r}), \quad (2.11) \]
where \( \mu_i \) is the mobility of the i-th ion. Since the diffusion and drift can be impeded by the same molecular processes, there exists a physical connection between the parameters \( \mu_i \) and \( D_i(\mathbf{r}) \), i.e., Einstein relation:
\[ \mu_i = \frac{|q_i|}{k_B T} D_i(\mathbf{r}), \]
where \( k_B \) and \( T \) are the Boltzmann constant and the absolute temperature, respectively.
Then, (2.11) becomes
\[ \mathbf{J}_{i,e}(t, \mathbf{r}) = -\frac{q_i}{k_B T} \mathbf{D}_i(\mathbf{r}) c_i(t, \mathbf{r}) \nabla \phi(t, \mathbf{r}). \] (2.12)

When both diffusional and electric field forces are present in (2.10) and (2.12), the total flux density,
\[ \mathbf{J}_i(t, \mathbf{r}) = \mathbf{J}_{i,d}(t, \mathbf{r}) + \mathbf{J}_{i,e}(t, \mathbf{r}), \]
is obtained in the expression
\[ \mathbf{J}_i(t, \mathbf{r}) = -\mathbf{D}_i(\mathbf{r}) \left[ \nabla c_i(t, \mathbf{r}) + \frac{q_i}{k_B T} c_i(t, \mathbf{r}) \nabla \phi(t, \mathbf{r}) \right]. \] (2.13)

Combining (2.13) with (2.9), we obtain the Nernst-Planck equation of the \( i \)th ionic species in SI units as follows:
\[ \frac{\partial c_i(t, \mathbf{r})}{\partial t} = \nabla \cdot \mathbf{D}_i(\mathbf{r}) \left[ \nabla c_i(t, \mathbf{r}) + \frac{Z_i e_c}{k_B T} c_i(t, \mathbf{r}) \nabla \phi(t, \mathbf{r}) \right]. \] (2.14)

2.4 A Poisson-Nernst-Planck ion channel system

We now consider the steady-state Poisson-Nernst-Planck ion channel system as follows:
\[
\begin{cases}
-\epsilon_m \Delta \phi(\mathbf{r}) = 0, & \mathbf{r} \in D_m, \\
-\nabla \cdot \epsilon_s(\mathbf{r}) \nabla \phi(\mathbf{r}) = \frac{\epsilon_0}{\epsilon_0} \sum_{i=1}^{n} Z_i c_i(\mathbf{r}), & \mathbf{r} \in D_s, \\
-\epsilon_p \Delta \phi(\mathbf{r}) = \frac{\epsilon_0}{\epsilon_0} \sum_{j=1}^{n_p} z_j \delta(\mathbf{r} - \mathbf{r}_j), & \mathbf{r} \in D_p, \\
\nabla \cdot \mathbf{D}_i(\mathbf{r}) \left[ \nabla c_i(\mathbf{r}) + \frac{Z_i e_c}{k_B T} c_i(\mathbf{r}) \nabla \phi(\mathbf{r}) \right] = 0, & \mathbf{r} \in D_s, \quad i = 1, 2, \ldots, n.
\end{cases}
\] (2.15)

where \( \phi \) is measured in Volts (V), \( c_i \) has the unit of the number of ions per cubic meter, \( e_c \), \( \epsilon_0 \), \( k_B \), and \( T \) are given in Table 2.1, and \( D_i \) has the unit m\(^2\)/s. To obtain a dimensionless form of the PNP ion channel system (2.15), we express the potential \( \phi \) as
\[ \phi(\mathbf{r}) = \frac{k_B T}{e_c} u(\mathbf{r}), \] (2.16)
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Name</th>
<th>Unit(abbr.)</th>
<th>Magnitude</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>Absolute temperature</td>
<td>Kelvin(K)</td>
<td>298.15</td>
</tr>
<tr>
<td>$k_B$</td>
<td>Boltzmann constant</td>
<td>Joule/Kelvin(J/K)</td>
<td>$1.380648813 \times 10^{-23}$</td>
</tr>
<tr>
<td>$\epsilon_0$</td>
<td>Permittivity of vacuum</td>
<td>Farad/Meter(F/m)</td>
<td>$8.854187817 \times 10^{-12}$</td>
</tr>
<tr>
<td>$\epsilon_e$</td>
<td>Elementary charge</td>
<td>Coulomb(C)</td>
<td>$1.602176565 \times 10^{-19}$</td>
</tr>
</tbody>
</table>

Table 2.1: Magnitudes & units of PNP parameters in SI unit system

where $u$ is a dimensionless potential function. Then the PNP ion channel system (2.15) can be changed as

$$
\begin{align*}
-\epsilon_m \Delta u(r) &= 0, \quad r \in D_m, \\
-\nabla \cdot \epsilon_s(r) \nabla u(r) &= \frac{e^2}{\epsilon_0 k_B T} \sum_{i=1}^{n} Z_i c_i(r), \quad r \in D_s, \\
-\epsilon_p \Delta u(r) &= \frac{e^2}{\epsilon_0 k_B T} \sum_{j=1}^{n_p} z_j \delta(r - r_j), \quad r \in D_p, \\
\nabla \cdot D_i(r) \left[ \nabla c_i(r) + Z_i c_i(r) \nabla u(r) \right] &= 0, \quad r \in D_s, \quad i = 1, 2, \cdots, n.
\end{align*}
$$

(2.17)

In molecular calculation, the coordinates of atoms are in unit of angstrom (Å). We change the length units from meter to angstrom. Correspondingly, the time is measured in picosecond (ps), diffusion coefficient $D_i(r)$ in units Å$^2$/ps, and the concentration $c_i(r)$ is measured in mole/liter (M/L). Then (2.17) can be rewritten as

$$
\begin{align*}
-\epsilon_m \Delta u(r) &= 0, \quad r \in D_m, \\
-\nabla \cdot \epsilon_s(r) \nabla u(r) &= \beta \sum_{i=1}^{n} Z_i c_i(r), \quad r \in D_s, \\
-\epsilon_p \Delta u(r) &= \alpha \sum_{j=1}^{n_p} z_j \delta(r - r_j), \quad r \in D_p, \\
\nabla \cdot D_i(r) \left[ \nabla c_i(r) + Z_i c_i(r) \nabla u(r) \right] &= 0, \quad r \in D_s, \quad i = 1, 2, \cdots, n,
\end{align*}
$$

(2.18)

where

$$
\alpha = \frac{10^{10} \epsilon_e^2}{\epsilon_0 k_B T}, \quad \beta = \frac{N_A e^2}{10^{17} \epsilon_0 k_B T},
$$

(2.19)

and $N_A = 6.02214076 \times 10^{23}$, which is the Avogadro constant (an estimate of the number of ions per mole.)
2.5 Interface and boundary conditions

In physics, the potential function and the normal derivative of the potential function should be continuous at the interface between different regions when the permittivity function $\epsilon_r$ is discontinuous across the interfaces. A membrane surface charge density function $\sigma(r)$ is provided on the membrane in order to consider the membrane surface charge effect. Thus, we have the following interface conditions:

\[
\begin{align*}
\mathbf{u}(s^-) &= \mathbf{u}(s^+), \quad \epsilon_p \frac{\partial \mathbf{u}(s^-)}{\partial n_p(s)} = \epsilon_s \frac{\partial \mathbf{u}(s^+)}{\partial n_p(s)}, & s \in \Gamma_p, \\
\mathbf{u}(s^-) &= \mathbf{u}(s^+), \quad \epsilon_m \frac{\partial \mathbf{u}(s^-)}{\partial n_m(s)} = \epsilon_s \frac{\partial \mathbf{u}(s^+)}{\partial n_m(s)} + \tau \sigma(s), & s \in \Gamma_m, \\
\mathbf{u}(s^-) &= \mathbf{u}(s^+), \quad \epsilon_p \frac{\partial \mathbf{u}(s^-)}{\partial n_p(s)} = \epsilon_m \frac{\partial \mathbf{u}(s^+)}{\partial n_p(s)} + \tau \sigma(s), & s \in \Gamma_{pm},
\end{align*}
\]

where
\[
\tau = \frac{10^{-12}c}{\epsilon_0 k_B T},
\]

$n_m$ and $n_p$ denote the unit outward normal direction of $D_m$ and $D_p$, respectively,

\[
u(s^+) = \lim_{t \to 0^+} \mathbf{u}(s + tn(s)),
\]

which are limits along a normal direction $n$ of a given region from inside to outside, $\sigma(s)$ is defined by

\[
\sigma(s) = \begin{cases} 
\sigma_t, & s \in \Gamma_{mt}, \\
\sigma_b, & s \in \Gamma_{mb}.
\end{cases}
\]

Moreover, since the channel walls are insulating, charged particles cannot penetrate them. To ensure this, we require that the flux satisfies the Robin boundary value conditions:

\[
D_i(s) \left( \frac{\partial c_i(s)}{\partial n_s(s)} + Z_i c_i(s) \frac{\partial \mathbf{u}(s)}{\partial n_s(s)} \right) = 0, \quad s \in \Gamma_p \cup \Gamma_m,
\]

where $n_s$ denotes the unit outward normal direction of the solvent region $D_s$.

Different boundary conditions may be imposed on (2.17) depending on specific biological settings of ion channel problems. Setting $\Gamma_N = \Gamma_{Ns} \cup \Gamma_{Nm}$, we define the boundary conditions of $u$ and $c_i$ as follows:

- Dirichlet boundary conditions:
\[
\begin{aligned}
\begin{cases}
  u(s) = g(s), & s \in \partial \Omega, \\
  c_i(s) = g_i(s), & s \in \Gamma_D \cup \Gamma_{Ns}, \quad i = 1, 2, \ldots, n.
\end{cases}
\end{aligned}
\] (2.24)

- Mixed (Neumann) boundary conditions:

\[
\begin{aligned}
\begin{cases}
  u(s) = g(s), & s \in \Gamma_D, \\
  \frac{\partial u(s)}{\partial n(s)} = 0, & s \in \Gamma_N, \\
  c_i(s) = g_i(s), & s \in \Gamma_D, \\
  \frac{\partial c_i(s)}{\partial n(s)} = 0, & s \in \Gamma_{Ns}, \quad i = 1, 2, \ldots, n.
\end{cases}
\end{aligned}
\] (2.25)

where \( n \) denotes the unit outward normal direction of \( \Omega \). In calculation, \( g_i \) can be simply given by the bulk concentration \( c_i^b \) of species \( i \), which is selected to satisfy the electron neutrality

\[
\sum_{i=1}^{n} Z_i c_i^b = 0. \tag{2.26}
\]

In summary, the PNP ion channel (PNPic) system with Dirichlet boundary conditions consists of the Poisson dielectric boundary value problem:

\[
\begin{aligned}
\begin{cases}
  -\epsilon_p \Delta u(r) = \alpha \sum_{j=1}^{n_p} z_j \delta(r - r_j), & r \in D_p, \\
  -\epsilon_m \Delta u(r) = 0, & r \in D_m, \\
  -\nabla \cdot \epsilon_s(r) \nabla u(r) = \beta \sum_{i=1}^{n} Z_i c_i(r), & r \in D_s,
  \\
  u(s^-) = u(s^+), & \epsilon_p \frac{\partial u(s^-)}{\partial n_p(s)} = \epsilon_s(s^+) \frac{\partial u(s^+)}{\partial n_p(s)}, & s \in \Gamma_p, \\
  u(s^-) = u(s^+), & \epsilon_m \frac{\partial u(s^-)}{\partial n_m(s)} = \epsilon_s(s^+) \frac{\partial u(s^+)}{\partial n_m(s)} + \tau \sigma(s), & s \in \Gamma_m, \\
  u(s^-) = u(s^+), & \epsilon_p \frac{\partial u(s^-)}{\partial n_p(s)} = \epsilon_m \frac{\partial u(s^+)}{\partial n_p(s)}, & s \in \Gamma_{pm}, \\
  u(s) = g(s), & s \in \partial \Omega,
\end{cases}
\end{aligned}
\] (2.27)
and \( n \) Nernst-Planck boundary value problems:

\[
\begin{aligned}
\nabla \cdot D_i(\mathbf{r}) \left[ \nabla c_i(\mathbf{r}) + Z_i c_i(\mathbf{r}) \nabla u(\mathbf{r}) \right] &= 0, \quad \mathbf{r} \in D_s, \\
D_i(\mathbf{s}) \left( \frac{\partial c_i(s)}{\partial n_i(s)} + Z_i c_i(s) \frac{\partial u(s)}{\partial n_i(s)} \right) &= 0, \quad \mathbf{s} \in \Gamma_p \cup \Gamma_m, \\
c_i(\mathbf{s}) &= g_i(\mathbf{s}), \quad \mathbf{s} \in \Gamma_{Ns} \cup \Gamma_D,
\end{aligned}
\]

for \( i = 1, 2, \ldots, n \).
Chapter 3

Numerical solution of a
Poisson-Nernst-Planck ion channel model

In this chapter, to overcome the singular difficulty caused by the distribution function, we present a solution decomposition technique for the PNPic model with Dirichlet boundary value conditions, and obtain its variational formulations. Then we develop a fast finite element solver for solving them. In order to display the potential and concentration functions in two dimensional curves across the membrane, we introduce one scheme for computing a volume integral of a three dimensional function over a block from a block partition of the solvent region.

3.1 A solution decomposition method

In order to avoid the singularities caused by the Dirac Delta function in Poisson-Boltzmann equation, Xie [72] introduced a solution decomposition scheme. Now we apply this scheme to the Poisson dielectric boundary value problem (2.27) to decompose the electrostatic potential function $u$ into three functions:

$$u(r) = G(r) + \Psi(r) + \Phi(r), \quad (3.1)$$
where $G$ is the Green’s function, which can be given analytically

$$G(r) = \frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} \frac{z_j}{|r - r_j|}, \quad r \in \mathbb{R}^3. \quad (3.2)$$

$
\Psi(r)$ is the solution of a linear interface boundary value problem with the Dirichlet boundary value condition as follows:

$$\begin{cases}
\Delta \Psi(r) = 0, \quad r \in D_p \cup D_s \cup D_m, \\
\Psi(s^-) = \Psi(s^+), \quad \epsilon_p \frac{\partial \Psi(s^-)}{\partial n_p(s)} = \epsilon_p(s) \frac{\partial \Psi(s^+)}{\partial n_p(s)} + (\epsilon_s(s) - \epsilon_p) \frac{\partial G(s)}{\partial n_p(s)}, \quad s \in \Gamma_p, \\
\Psi(s^-) = \Psi(s^+), \quad \epsilon_m \frac{\partial \Psi(s^-)}{\partial n_m(s)} = \epsilon_m(s) \frac{\partial \Psi(s^+)}{\partial n_m(s)} + (\epsilon_s(s) - \epsilon_m) \frac{\partial G(s)}{\partial n_m(s)} + \tau \sigma(s), \quad s \in \Gamma_m, \\
\Psi(s^-) = \Psi(s^+), \quad \epsilon_p \frac{\partial \Psi(s^-)}{\partial n_p(s)} = \epsilon_m \frac{\partial \Psi(s^-)}{\partial n_p(s)} + (\epsilon_m - \epsilon_p) \frac{\partial G(s)}{\partial n_p(s)}, \quad s \in \Gamma_{pm}, \\
\Psi(s) = g(s) - G(s), \quad s \in \partial \Omega, 
\end{cases} \quad (3.3)$$

and $\Phi(r)$ is the solution of a linear interface boundary value problem with the Dirichlet boundary value condition as follows:

$$\begin{cases}
\Delta \Phi(r) = 0, \quad r \in D_p \cup D_m, \\
-\nabla \cdot \epsilon_s(r)\nabla \Phi(r) = \beta \sum_{i=1}^{n} Z_i c_i(r), \quad r \in D_s, \\
\Phi(s^-) = \Phi(s^+), \quad \epsilon_p \frac{\partial \Phi(s^-)}{\partial n_p(s)} = \epsilon_p(s) \frac{\partial \Phi(s^+)}{\partial n_p(s)}, \quad s \in \Gamma_p, \\
\Phi(s^-) = \Phi(s^+), \quad \epsilon_m \frac{\partial \Phi(s^-)}{\partial n_m(s)} = \epsilon_m(s) \frac{\partial \Phi(s^+)}{\partial n_m(s)}, \quad s \in \Gamma_m, \\
\Phi(s^-) = \Phi(s^+), \quad \epsilon_p \frac{\partial \Phi(s^-)}{\partial n_p(s)} = \epsilon_m \frac{\partial \Phi(s^-)}{\partial n_p(s)}, \quad s \in \Gamma_{pm}, \\
\Phi(s) = 0, \quad s \in \partial \Omega. 
\end{cases} \quad (3.4)$$

Actually, $G$ is the solution of the Poisson dielectric equation

$$-\epsilon_p \Delta G(r) = \alpha \sum_{j=1}^{n_p} z_j \delta(r - r_j), \quad r \in \mathbb{R}^3. \quad (3.5)$$

From (3.3), (3.4), and (3.5) we notice that $\Phi$ is independent of $G$ and $\Psi$, and $G$ and $\Psi$ are independent of $c_i$, so we can calculate $G$ and $\Psi$ first, then set $w(r) = G(r) + \Psi(r)$ to
rewrite the Nernst-Planck boundary value problems as follows:

\[
\begin{align*}
\nabla \cdot D_i(r) \left[ \nabla c_i(r) + Z_i c_i(r) \nabla w(r) + Z_i c_i(r) \nabla \tilde{\Phi}(r) \right] &= 0, \quad r \in D_s, \\
D_i(s) \left( \frac{\partial c_i(s)}{\partial n_i(s)} + Z_i c_i(s) \frac{\partial w(s)}{\partial n_i(s)} + Z_i c_i(s) \frac{\partial \tilde{\Phi}(s)}{\partial n_i(s)} \right) &= 0, \quad s \in \Gamma_p \cup \Gamma_m, \\
c_i(s) &= g_i(s), \quad s \in \Gamma_{Ns} \cup \Gamma_D,
\end{align*}
\]

for \( i = 1, 2, \ldots, n \).

By introducing the effective densities (also referred to as Slotboom variables in semiconductor literature)

\[
\bar{c}_i(r) = c_i(r) e^{Z_i u(r)}, \quad \bar{D}_i(r) = D_i(r) e^{-Z_i u(r)},
\]

the Nernst-Planck equations (3.6) with the Dirichlet boundary value conditions can be transformed as

\[
\begin{align*}
-\nabla \cdot (\bar{D}_i(r) \nabla \bar{c}_i(r)) &= 0, \quad r \in D_s, \\
\bar{D}_i(s) \left( \frac{\partial c_i(s)}{\partial n_i(s)} \right) &= 0, \quad s \in \Gamma_p \cup \Gamma_m, \\
\bar{c}_i(s) &= g_i(s) e^{Z_i g(s)}, \quad s \in \Gamma_{Ns} \cup \Gamma_D,
\end{align*}
\]

where \( i = 1, 2, \ldots, n \).

From the above system, we notice that the Slotboom transformation removes the convection term and results in self-adjoint Laplace equations. However, the transformed diffusion coefficient \( \bar{D}_i(r) \) could result in a large condition number for the stiffness matrix of a finite element approximate equation of (3.8). Furthermore, the Slotboom transformation may cause overflow problems in numerical implementation due to the exponential term.

### 3.2 Variational formulation

To solve the PNPic model defined in (2.27) and (2.28) by the finite element method, the equivalent variational problems of them are needed, they can be obtained by following method in [74].

Let \( H^1(\Omega) \) and \( H^1(D_s) \) be the Sobolev function spaces defined in \( \Omega \) and \( D_s \), respectively.
Then, we introduce their subspaces, $H^1_0(\Omega)$ and $V_0$, as follows:

$$H^1_0(\Omega) = \{ \nu \in H^1(\Omega) | \nu = 0 \text{ on } \partial\Omega \}, \quad V_0 = \{ \mu \in H^1(D_s) | \mu = 0 \text{ on } \Gamma_{Ns} \cup \Gamma_D \}.$$  

A finite element solution $\Psi$ of (3.3) can be found by solving the equivalent linear variational problem:

Find $\Psi \in H^1(\Omega)$ satisfying $\Psi(s) = g(s) - G(s)$ on $\partial\Omega$ such that

$$a(\Psi, \nu) = L_1(\nu) \quad \forall \nu \in H^1_0(\Omega),$$  

(3.9)

where

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

(3.10)

and

$$L_1(\nu) = \int_{\Gamma_p} \frac{\partial G(s)}{\partial n_p(s)} (\epsilon_s(s) - \epsilon_p) \nu(s) ds + \int_{\Gamma_m} \frac{\partial G(s)}{\partial n_m(s)} (\epsilon_s(s) - \epsilon_m) \nu(s) ds$$

$$+ (\epsilon_m - \epsilon_p) \int_{\Gamma_{pm}} \frac{\partial G(s)}{\partial n_p(s)} \nu(s) ds + \tau \int_{\Gamma_m} \sigma(s) \nu(s) ds.$$  

(3.11)

Specially, if $\epsilon_m = \epsilon_p$, we have

$$L_1(\nu) = \int_{D_s} \nabla G(r) \cdot \nabla((\epsilon_p - \epsilon_s(r)) \nu(r)) dr + \tau \int_{\Gamma_m} \sigma(s) \nu(s) ds.$$  

(3.12)

Actually, by the Green’s first identity, Gauss’s Theorem, $\Delta G(r) = 0$ in $D_s$, and $\nu = 0$ on $\partial\Omega$, the volume integral of (3.12) can be reformulated as follows:

$$\int_{D_s} \nabla G(r) \cdot \nabla((\epsilon_p - \epsilon_s(r)) \nu(r)) dr$$

$$= \int_{\partial D_s} (\epsilon_p - \epsilon_s(s)) \nu(r) \frac{\partial G(s)}{\partial n(s)} ds$$

$$= \int_{\Gamma_p} \frac{\partial G(s)}{\partial n_p(s)} (\epsilon_p - \epsilon_s(s)) \nu(s) ds + \int_{\Gamma_m} \frac{\partial G(s)}{\partial n_m(s)} (\epsilon_p - \epsilon_s(s)) \nu(s) ds$$

$$= \int_{\Gamma_p} \frac{\partial G(s)}{\partial n_p(s)} (\epsilon_s(s) - \epsilon_p) \nu(s) ds + \int_{\Gamma_m} \frac{\partial G(s)}{\partial n_m(s)} (\epsilon_s(s) - \epsilon_p) \nu(s) ds.$$
Due to the complexity of the molecular surface, it may be difficult to do the interface integral calculation. We can avoid this, we can difficulty by using (3.12) when $\epsilon_m = \epsilon_p$.

After using the Slotboom transformation, the linear interface boundary value problem (3.4) can be formulated as the following nonlinear variational problem:

$$
\text{Find } \bar{\Phi} \in H^1_0(\Omega) \text{ such that } a(\bar{\Phi}, \nu) - L_2(\bar{\Phi}, \nu) = 0 \forall \nu \in H^1_0(\Omega),
$$

(3.13)

where

$$
a(\bar{\Phi}, \nu) = \epsilon_p \int_{D_p} \nabla \bar{\Phi}(r) \cdot \nabla \nu(r)dr + \epsilon_m \int_{D_m} \nabla \bar{\Phi}(r) \cdot \nabla \nu(r)dr + \int_{D_s} \nabla \bar{\Phi}(r) \cdot \nabla (\epsilon_s(r) \nu(r))dr
$$

(3.14)

and

$$
L_2(\bar{\Phi}, \nu) = \beta \sum_{i=1}^{n} Z_i \int_{D_s} e^{-Z_i(G(r) + \Psi(r) + \bar{\Phi}(r))} \bar{c}_i(r) \nu(r)dr.
$$

(3.15)

Finally, the Nernst-Planck boundary value problem of (3.6) can be formulated as the following variational problems:

$$
\text{Find } \bar{c}_i \in H^1(D_s) \text{ satisfying } \bar{c}_i = g_{i} e^{Z_i g} \text{ on } \Gamma_{N_s} \cup \Gamma_D \text{ such that }
$$

$$
\int_{D_s} D_i(r) \nabla \bar{c}_i(r) \cdot \mu_i(r)dr = 0, \forall \mu_i \in V_0, i = 1, 2, \cdots, n.
$$

(3.16)

### 3.3 A finite element iterative solver

In this section, we describe our finite element solver for the PNPic model based on the variational problems. Notice that the values of $c_i$ are zero in the domains $D_m$ and $D_p$, so $c_i$ can only be defined in the solvent domain $D_s$ while an electrostatic potential function $u$ in the whole box domain $\Omega$. In order to couple functions $u$ and $c_i$ when define Nernst-Planck equation, most of the researchers tried to extended $c_i$ from $D_s$ to $\Omega$ by setting $c_i(r) = 0$ at the mesh nodes outside $D_s$. But, this simple treatment may decay the accuracy of a PNP numerical solution significantly since $c_i$ can be nonzero outside $D_s$ on a layer of tetrahedra along the interfaces $\Gamma_p$ and $\Gamma_m$ due to the continuity of $c_i$. To improve this, we define a finite element function space for $c_i$ or $\bar{c}_i$ based on an irregular tetrahedral mesh of $D_s$ directly.
Let $U$ and $V$ be the finite dimensional subspaces of $H^1(\Omega)$ and $H^1(D_s)$, respectively, and

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

After calculating $G$ and $\Psi$ from (3.2) and (3.9), we let $w = G + \Psi$, and compute $\tilde{\Phi}$ and $\bar{c}_i$ by solving the following nonlinear system:

Find $\tilde{\Phi} \in U_0$ and $\bar{c}_i \in V$ satisfying $\bar{c}_i = g_i e^{Z_i g}$ on $\Gamma_{N_s} \cup \Gamma_D$ such that

$$\begin{cases}
\int_{D_s} D_i(r) e^{-Z_i(w + \tilde{\Phi})} \nabla \bar{c}_i(r) \cdot \nabla \nu_i(r) dr = 0, & \forall \nu_i \in V_0, i = 1, 2, \ldots, n, \\
a(\tilde{\Phi}, \nu_{n+1}) - \beta \sum_{j=1}^{n} Z_i \int_{D_s} \bar{c}_i(r) e^{-Z_i(w + \tilde{\Phi})} \nu_{n+1}(r) dr = 0, & \forall \nu_{n+1} \in U_0.
\end{cases}$$

After finding $\tilde{\Phi}$ and $\bar{c}_i$, we recover $u$ and $c_i$ by the formulas (3.1) and (3.7), respectively.

The nonlinear system (3.17) can be solved efficiently by adopting the nonlinear relaxation method reported in [74]. Initial iterates are essentially to determine the convergent property of the relaxation method, and how to choose good initial guesses $\bar{c}_i(0)$ and $\tilde{\Phi}(0)$ for the relaxation method is an important topic. Interestingly, when we let the slotboom variable $\bar{c}_i = c^b_i$, which is a bulk concentration function of the $i$th ion species, then

$$c_i = c^b_i e^{-Z_i u}, \quad i = 1, 2, \ldots, n,$$

and the nonlinear equation (3.9) can be simplified as follows:

Find $\tilde{\Phi} \in U_0$ such that $a(\tilde{\Phi}, \nu) - \beta \sum_{j=1}^{n} Z_j c^b_j \int_{D_s} e^{-Z_i(w + \tilde{\Phi})} \nu dr = 0 \forall \nu \in U_0.$

It can be easily shown that adding the solution $\tilde{\Phi}$ of above equation with $w$ gives a solution $\tilde{\Phi}^{PB}$ of a Poisson-Boltzmann (PB) ion channel model with Dirichlet boundary value.
conditions as follows:

\[
\begin{aligned}
-\epsilon_m \Delta u(r) &= 0, \quad r \in D_m, \\
-\nabla \cdot \epsilon_s(r) \nabla u(r) &= \beta \sum_{i=1}^{n} Z_i c_i^b e^{-Z_i u(r)}, \quad r \in D_s, \\
-\epsilon_p \Delta u(r) &= \alpha \sum_{j=1}^{n_p} z_j \delta(r - r_j), \quad r \in D_p, \\
u(s^-) &= u(s^+), \quad \epsilon_p \frac{\partial u(s^-)}{\partial \nu(p(s))} = \epsilon_s(s) \frac{\partial u(s^+)}{\partial \nu(p(s))}, \quad s \in \Gamma_p, \\
u(s^-) &= u(s^+), \quad \epsilon_m \frac{\partial u(s^-)}{\partial \nu(m(s))} = \epsilon_s(s) \frac{\partial u(s^+)}{\partial \nu(m(s))} + \tau \sigma(s), \quad s \in \Gamma_m, \\
u(s^-) &= u(s^+), \quad \epsilon_p \frac{\partial u(s^-)}{\partial \nu(p(s))} = \epsilon_m \frac{\partial u(s^+)}{\partial \nu(p(s))}, \quad s \in \Gamma_{pm}, \\
u(s) &= g(s), \quad s \in \partial \Omega,
\end{aligned}
\]  

(3.19)

Now, we can set \( \bar{c}_i(0) = c_i^b \) and \( \bar{\Phi}(0) = \bar{\Phi}^{PB} \) as a good initial guess for the relaxation method.

### 3.4 Visualization of potential and concentration functions in curves

After the calculation, we can visualize the values of 3D functions \( u \) (or \( c_i \)) on a cross-section of a region in a color mapping or in a line that passes through an ion channel pore as done in the current literature. However, such a visualization only reflects some values of \( u \) (or \( c_i \)) and may ignore some important values. To display a global property of \( u \) and \( c_i \) over a 3D region, we introduce a numerical method for calculating a set of average values of \( u \) (or \( c_i \)) over a block partition of the solvent region \( D_s \) along the normal direction of the membrane, which is the \( z \)-axis direction in this thesis. Using these average values, we can then plot a 2D curve. Such a curve is valuable in the study of potential and ionic distribution profiles passing through an ion channel pore and in the comparison of potential and ionic concentration functions generated by either different models or the same model but using different parameter values.

Let \( \hat{h} \) be a positive parameter, which can be used to control the quality of the 2D curve. Set \( m = m_1 + m_2 + m_3 \), and start with \( m + 2 \) partition numbers \( \{z_j\} \) of the interval \( [L_{z_1}, L_{z_2}] \).
as follows:

\[ z^0 < z^1 < \ldots < z^{m_1} < \ldots < z^{m_1+m_2} < \ldots < z^{m+1}, \]

where

\[ z^0 = L_{z_1}, \quad z^{m_1} = P_{z_1}, \quad z^{m_1+m_2} = P_{z_2}, \quad \text{and} \quad z^{m+1} = L_{z_2}. \]

In addition, the set \{z^j\}_{j=1}^m contains the important location numbers that we want to know the average values such as the membrane location numbers Z1 and Z2. We also can set a larger value of \(m_2\) to get more average values in the channel pore. Then we can build \(m\) mesh blocks \(B_j\) of \(D_s\) by

\[ B_j = ([z^j - \bar{h}/2, z^j + \bar{h}/2] \times [L_{x_1}, L_{x_2}] \times [L_{y_1}, L_{y_2}]) \cap D_s, \quad j = 1, 2, \ldots, m, \quad (3.20) \]

where \(z^j\) is the middle location of \(B_j\) in the \(z\)-direction, and \(B_j\) is allowed to be overlapped with its adjacent blocks in order to avoid missing any tetrahedron, and each \(B_j\) should have a nonzero volume \(\|B_j\|\) by adjusting the parameter \(\bar{h}\). It is clear that \(\bigcup_{j=1}^m B_j = D_s\). Then, we can set \(c_i^0 = c_i^b = c_i^m = c_i^t\) to obtain \(m + 2\) points \((z^j, c_i^j)\) for \(j = 0, 1, 2, \ldots, m, m + 1\) by

\[ c_i^j = \frac{1}{\|B_j\|} \int_{B_j} c_i^j(r) dr, \quad j = 1, 2, \ldots, m, \quad (3.21) \]

where \(c_i^b\) and \(c_i^t\) are bulk concentrations in the upper and lower bulk regions, respectively. Note that an electrostatic potential function \(u\) has positive and negative values. Hence, we rewrite \(u\) as the sum of a positive function \(u^+\) and a negative function \(u^-\) by the formulas

\[ u = u^+ + u^-, \quad u^+(r) = \frac{u(r) + |u(r)|}{2}, \quad u^-(r) = \frac{u(r) - |u(r)|}{2} \quad \forall r \in D_s. \quad (3.22) \]

Now, we can set \(u_0^\pm = g_b\) and \(u_{m+1}^\pm = g_t\), and calculate the \(m\) average values \(u_j^\pm\) of \(u^\pm\) by

\[ u_j^\pm = \frac{1}{\|B_j\|} \int_{B_j} u_j^\pm(r) dr, \quad j = 1, 2, \ldots, m, \quad (3.23) \]

which produce \(m + 2\) points \((z^j, u_j^\pm)\) for \(j = 0, 1, 2, \ldots, m, m + 1\).

After the above calculations, we get points \((z^j, c_i^j)\) and \((z^j, u_j^\pm)\) for \(j = 0, 1, 2, \ldots, m, m + 1\), then 2D curves can be plotted as the visualization of concentration functions \(c_i\) and electrostatic potential function \(u\) over \(D_s\) along the \(z\)-axis direction. Since the mesh \(\bar{D}_{s,\bar{h}}\)
is unstructured, each block $B_j$, as a submesh of $\bar{D}_{s,h}$, also has an irregular shape, thus, the volume integral is difficult to calculate. Fortunately, FEniCS has provided us with some useful functions to help us to do so. We can extract a mesh of $B_j$, denoted by $B_{j,h}$, from a mesh $\bar{D}_{s,h}$ by the FEniCS function SubMesh, for example. As a submesh of $\bar{D}_{s,h}$, $B_{j,h}$ consists of tetrahedrons from $\bar{D}_{s,h}$. Therefore, $c^j_i$ and $u^\pm_j$ can be calculated approximately by:

$$c^j_i \approx \frac{1}{\|B_{j,h}\|} \int_{B_{j,h}} c_i(r) dr = \frac{\sum_{T \in B_{j,h}} \int_T c_i(r) dr}{\sum_{T \in B_{j,h}} \|T\|}, \quad (3.24)$$

and

$$u^\pm_j \approx \frac{1}{\|B_{j,h}\|} \int_{B_{j,h}} u^\pm_j(r) dr = \frac{\sum_{T \in B_{j,h}} \int_T u^\pm_j(r) dr}{\sum_{T \in B_{j,h}} \|T\|}, \quad (3.25)$$

where $T$ denotes a tetrahedron of the mesh domain $B_{j,h}$. 
Chapter 4

An improved ion channel finite element mesh generation scheme

To solve a PNP ion channel model by the finite element method, one of the most important tasks is to generate an interface fitted unstructured tetrahedral mesh for a box simulation domain. Actually, it is very difficult to construct such a mesh due to the irregular shape of proteins. In the PNPic model, to handle different boundary conditions and equations are defined in different regions, we need to mark triangular surface meshes on the interfaces/boundaries and tetrahedra meshes in different regions. In this chapter, we describe our algorithms and implementations for generating these meshes, and numerical tests for showing the robustness and efficiency of our mesh generation package are presented.

4.1 Our mesh generation method

We consider a closed box domain

\[ \Omega = \{(x, y, z)|L_{x1} \leq x \leq L_{x2}, L_{y1} \leq y \leq L_{y2}, L_{z1} \leq x \leq L_{z2}\} \] (4.1)

and the partition

\[ \bar{\Omega} = \bar{D}_p \cup \bar{D}_m \cup \bar{D}_s, \]
where $\bar{D}_p$, $\bar{D}_m$, and $\bar{D}_s$ are the closures of $D_p$, $D_m$, and $D_s$, respectively. Our purpose is to generate tetrahedral meshes for subdomains $D_p$, $D_m$, and $D_s$ as well as triangular surface meshes for the interfaces $\Gamma_p$, $\Gamma_m$, and $\Gamma_{pm}$ such that they are the submeshes of a box mesh $\bar{\Omega}_h$ to satisfy

$$\bar{\Omega}_h = \bar{D}_{p,h} \cup \bar{D}_{m,h} \cup \bar{D}_{s,h}.$$  \hspace{1cm} (4.2)

We define the interface meshes by

$$\Gamma_{p,h} = \bar{D}_{p,h} \cap \bar{D}_{s,h}, \quad \Gamma_{m,h} = \bar{D}_{s,h} \cap \bar{D}_{m,h}, \quad \Gamma_{pm,h} = \bar{D}_{p,h} \cap \bar{D}_{m,h}.$$ 

Here, the normal direction of $\bar{D}_m$ is in the z-axis direction, and the location of $D_m$ is set between the two planes $z = Z_1$ and $z = Z_2$.

According to the methodology of volume mesh generation, the tetrahedral mesh for a 3D domain can be generated from triangular surface meshes of its boundary. Thus, the box boundary triangular surface mesh $\partial \bar{\Omega}_h$, the protein surface mesh $\partial \bar{D}_{p,h}$, and the membrane surface mesh $\partial \bar{D}_{m,h}$ are needed. However, due to the irregular and complicated interface $\Gamma_{pm,h}$, $\bar{D}_{m,h}$ is very difficult to produce directly. To avoid this difficulty, it is usually to first generate a good quality mesh for an expanded solvent region, $\hat{D}_s$, defined by

$$\hat{D}_s = \bar{D}_s \cup \bar{D}_m,$$

such that

$$\hat{\Omega} = \hat{D}_s \cup \bar{D}_p \cup \hat{\Gamma}_{sp},$$

where $\hat{\Gamma}_{sp}$ is exactly a molecule surface of an ion channel protein, which can be given approximately by the triangular surface mesh $\partial \bar{D}_{p,h}$. We next construct the triangular surface mesh $\partial \bar{\Omega}_h$ by

$$\partial \bar{\Omega}_h = \Gamma_{D,h} \cup \Gamma_{N,h},$$

where $\Gamma_{D,h}$ and $\Gamma_{N,h}$ are the triangular meshes of $\Gamma_D$ and $\Gamma_N$, respectively.

Using $\partial \bar{\Omega}_h$ and $\partial \bar{D}_{p,h}$, we can generate $\hat{D}_{s,h}$ and $\bar{D}_{p,h}$ such that

$$\hat{\Gamma}_{sp,h} = \hat{D}_{s,h} \cap \bar{D}_{p,h},$$

which is a triangular surface mesh of the interface $\hat{\Gamma}_{sp}$. The remaining work is to develop
algorithms to produce $\bar{D}_{m,h}$ as a submesh of $\hat{D}_{s,h}$, such that

$$\hat{D}_{s,h} = \bar{D}_{s,h} \cup \bar{D}_{m,h},$$

approximately. To do so, a scheme is used in the current mesh package developed by Lu’s group. In this scheme, a list of additional mesh nodes is pre-allocated on the two planes $z = Z1$ and $z = Z2$, as a major part of the mesh nodes of $\Gamma_{m,h}$, and a mesh of the four side surfaces of $D_m$ is also constructed as a submesh of $\partial \Omega_h$.

Our major improvements on this mesh generation algorithm include a new scheme for the box boundary mesh $\partial \Omega_h$ generation and a new scheme for generating the membrane region mesh $\bar{D}_{m,h}$. Moreover, we implement these two new schemes to yield a new mesh generation package, and then adopt it as a part of our PNPic software package.

### 4.2 A new triangular surface mesh for a simulation box

In this section, we present a new scheme for generating a high quality box surface mesh $\partial \Omega_h$. By this new scheme, the quality of the membrane and solvent region meshes can be improved significantly.

We set the surface mesh partition

$$\Gamma_{N,h} = \Gamma_{N,s,h} \cup \Gamma_{N,m,h},$$

where $\Gamma_{N,s,h}$ and $\Gamma_{N,m,h}$ are the lateral surface meshes of $\bar{D}_s$ and $\bar{D}_m$, respectively. To guarantee that the two surface meshes $\Gamma_{N,s,h}$ and $\Gamma_{N,m,h}$ can match well, we further set the surface mesh partition

$$\Gamma_{N,s,h} = \Gamma_{N,s1,h} \cup \Gamma_{N,s2,h},$$

$\Gamma_{N,s2,h}$ represents the layer between $\Gamma_{N,s1,h}$ and $\Gamma_{N,m,h}$, $\Gamma_{N,s1,h}$ denotes the upper and lower parts of $\Gamma_{N,s,h}$ except the layer. We then generate $\partial \Omega_h$ by the following steps:
• Step 1: Set the box domain $\bar{\Omega}$ by

$$
\begin{align*}
L_{x_1} &= P_{x_1} - \delta_{x_1}, & L_{x_2} &= P_{x_2} + \delta_{x_2}, \\
L_{y_1} &= P_{y_1} - \delta_{y_1}, & L_{y_2} &= P_{y_2} + \delta_{y_2}, \\
L_{z_1} &= P_{z_1} - \delta_{z_1}, & L_{z_2} &= P_{z_2} + \delta_{z_2},
\end{align*}
$$

where $\delta_{x_1}, \delta_{x_2}, \delta_{y_1}, \delta_{y_2}, \delta_{z_1}, \delta_{z_2}$ are the input positive parameters to control the size of the box domain, and the box $[P_{x_1}, P_{x_2}] \times [P_{y_1}, P_{y_2}] \times [P_{z_1}, P_{z_2}]$ holds the protein region $\bar{D}_{p,h}$.

• Step 2: Specify a box surface mesh of the box domain by the inputs $n_x, n_y, n_z$, which set the numbers of partitions in the $x$-, $y$-, and $z$-directions, respectively. We then simply set the number of partitions in the $z$-direction on the upper and lower parts of $\Gamma_{N_{s1},h}$ as $n_z/2$, the number of partitions in the $x$-, $y$-, and $z$-directions on the surface mesh $\Gamma_{N_{m},h}$ as $2n_x, 2n_y, \text{ and } n_z$. This setting leads to meshes of the membrane-protein region is finer than that of the solvent region and avoids any sharp triangular facet on the interface between the solvent and membrane regions. In addition, a parameter $d_z$ is introduced to control the height of $\Gamma_{N_{s2},h}$, which can be adjusted to further improve the quality of the mesh.

• Step 3: Generate the box triangular surface mesh data, vertices and triangles, and save these data for generating the volume mesh. In particular, we use the BoxMesh function from the state-of-the-art software package FEniCS [35] to generate the uniform volume meshes $\bar{\Omega}_{1,h}, \bar{\Omega}_{2,h}, \text{ and } \bar{\Omega}_{3,h}$ of the top, middle, and bottom parts of the box domain as follows

$$
\bar{\Omega}_{1,h} = \text{BoxMesh}((L_{x_1}, L_{y_1}, L_{z_1}), (L_{x_2}, L_{y_2}, Z1 - d_z), n_x, n_y, n_z), \\
\bar{\Omega}_{2,h} = \text{BoxMesh}((L_{x_1}, L_{y_1}, Z1), (L_{x_2}, L_{y_2}, Z2), 2n_x, 2n_y, n_z), \\
\bar{\Omega}_{3,h} = \text{BoxMesh}((L_{x_1}, L_{y_1}, Z2 + d_z), (L_{x_2}, L_{y_2}, L_{z_2}), n_x, n_y, n_z).
$$

We then obtain a mesh of the box domain $\bar{\Omega}$ by

$$
\bar{\Omega}_h = \bar{\Omega}_{1,h} \cup \bar{\Omega}_{2,h} \cup \bar{\Omega}_{3,h}.
$$
We then extract the surface meshes $\Gamma_{Ns1,h}$, $\Gamma_{Nm,h}$ and $\Gamma_{D,h}$ by the function BoundaryMesh from FEniCS. Consequently, we produce the surface mesh $\partial \bar{\Omega}_h$ by

$$\partial \bar{\Omega}_h = \Gamma_{Ns1,h} \cup \Gamma_{Ns2,h} \cup \Gamma_{Nm,h} \cup \Gamma_{D,h}.$$ 

### 4.3 A new scheme for generating membrane tetrahedral meshes

Once the box surface mesh $\partial \bar{\Omega}_h$ and the protein surface mesh $\partial \bar{D}_{p,h}$ are obtained, then the corresponding tetrahedral meshes $\tilde{D}_{s,h}$ and $\bar{D}_{p,h}$ can be generated by the volume mesh generation packages such as TetGen [62, 61]. We then need a scheme to determine $\bar{D}_{m,h}$ as a submesh of the extended solvent mesh $\tilde{D}_{s,h}$. However, developing such a scheme is difficult since the interface between the membrane and the solvent regions is not given in an explicit function expression. To avoid such a difficulty, we can construct a scheme to re-mark the tetrahedra of $\tilde{D}_{s,h}$ that belong to the membrane region $D_m$ from the label number 2 to 3 as illustrated in Figure 4.1. Using these region label data as shown in Figure 4.1(b), we can produce the triangular meshes of the related interfaces and boundaries.

![Figure 4.1](image)

Figure 4.1: An illustration of the labels of protein, solvent, and membrane regions.

The most recent scheme for constructing $\bar{D}_{m,h}$ is reported in [32]. In this scheme, the walk-and-detect algorithm [12] is used to construct the membrane region mesh $\bar{D}_{m,h}$. This
method works well when the channel pore has a smooth and regular shape. It does not work well or even fails for an ion channel with a complicated channel pore. To improve this scheme, we develop a new algorithm for constructing $\bar{D}_{m,h}$. We then show that our new algorithm works for an ion channel protein even with an irregular ion channel pore provided that a surface mesh of the interface region $\Gamma_{pm}$ does not have any hole.

We describe our new method in the following steps. Here we assume that $\Omega_h$ is produced, and the tetrahedrons of the protein region and the solvent-membrane region have been marked by label numbers 1 and 2, respectively (see Figure 4.1(a)).

- **Step 1**: Construct a partition of the membrane region $\bar{D}_m$ as

$$\bar{D}_m = \bar{D}_{m1} \cup \bar{D}_{m2},$$

where $\bar{D}_{m2}$ represents a subregion of $\bar{D}_m$ that intersects with a block region,

$$[l_{x1}, l_{x2}] \times [l_{y1}, l_{y2}] \times [Z1, Z2],$$

such that $\bar{D}_{m1}$ has a regular shape, where $l_{x1}$, $l_{x2}$, $l_{y1}$, and $l_{y2}$ are selected big enough not to touch the surface mesh of the protein. See Figure 4.2 for an illustration of this partition.

- **Step 2**: Produce a mesh $\bar{D}_{m1,h}$ of $D_{m1}$ by remarking the tetrahedrons of the extend solvent region that belong to this region from 2 to 3.

- **Step 3**: Find a “seed” tetrahedron in $\bar{D}_{m2,h}$ by using either the walk-and-detect method or a direct selection method. Numerical tests show that the efficiency of these two methods is almost the same. Figure 4.3(a) shows one “seed” tetrahedron, which is colored in purple.

- **Step 4**: Relabel all the tetrahedrons belonging to the subregion $D_{m2}$ starting from the “seed” tetrahedron via the following strategy: If the adjacent tetrahedron has label number 1 (i.e., in the protein region) or -1 (i.e., in the box boundary), the searching process is stopped; otherwise, this adjacent tetrahedron is re-labeled from 2 to 3, and continue to search for other adjacent tetrahedrons similarly until all the tetrahedrons belonging to $D_{m2}$ are found.
Figure 4.2: A cross-section view of the ion channel simulation box at $z = Z_1$. Here, the protein region is colored in green, the membrane region $D_m$ is split into two subdomains, $D_{m1}$ and $D_{m2}$, colored in yellow and red, respectively, and the channel pore area is marked in white, which belongs to the solvent region $D_s$.

Figure 4.3: An iterative search process for the tetrahedrons that belong to the membrane region. (a) Select a “seed” tetrahedron as the initial iteration. (b) A volume mesh generated after 16 iterations. (c) The membrane mesh generated at the end of the iteration.

- Step 5: Obtain the membrane mesh $\bar{D}_{m,h}$ by combining $\bar{D}_{m1,h}$ and $\bar{D}_{m2,h}$, i.e.,

$$\bar{D}_{m,h} = \bar{D}_{m1,h} \cup \bar{D}_{m2,h}.$$ 

The search process of Step 4 is easy to carry out because all the data of adjacent tetrahedrons exist (e.g., in a .neigh file when the volume mesh is generated by TetGen). Figure 4.3(b) displays the tetrahedrons (colored in purple) found in the membrane region after 16 iterations, and Figure 4.3(c) shows all the tetrahedrons (colored in purple) that make up the membrane region $D_m$. Once the membrane mesh is constructed, we save the data of vertices,
tetrahedrons, and tetrahedron labels to a .xml mesh file as required by FEniCS for a finite element mesh. We then can use this mesh directly for our PNPic package. In this way, we have obtained an improved mesh generation package.

4.4 Numerical results

As mentioned at the start of this chapter, Lu’s method works well when the channel pore has a smooth regular shape. However, it does not work well for ion channels with complicated channel pores. In this section, we present numerical tests to illustrate the robustness of our method compared to Lu’s method.

Figure 4.4(a) shows the cartoon backbone representation of the original mouse voltage-dependent anion-channel isoform (mVDAC1) ion channel. Figure 4.4(b) is the corresponding triangular surface meshes, which are generated by TMSmesh [6, 33] with parameters $h = 0.4$, $d = 0.9$, and $c = 0.9$.

Figure 4.5 displays a comparison of the membrane mesh $\bar{D}_{m,h}$ and the solvent mesh $\bar{D}_{s,h}$ generated by our improved mesh package with that by Lu’s mesh package for the original mVDAC1. From Figures 4.5(a, b, c) it can be seen that the membrane region mesh $\bar{D}_{m,h}$ generated by Lu’s mesh package contains the tetrahedra of $\bar{D}_{s,h}$ while the solvent region mesh $\bar{D}_{s,h}$ contains some sharp triangles on $\partial \bar{D}_{s,h}$. These low quality meshes may affect the accuracy of a finite element solution. Figures 4.5(d, e, f) display the meshes $\bar{D}_{m,h}$ and $\bar{D}_{s,h}$
Figure 4.5: (a, b, c) The subdomain meshes generated by Lu’s mesh package. (d, e, f) The subdomain meshes generated by our improved mesh package, which have much higher quality.
Chapter 5

Numerical studies on some ion channel modeling issues

In this chapter, we numerically study some important issues arisen from the PNP ion channel modeling. Our PNPic software package makes such a study possible. All numerical tests were done on an Intel Core i5 3.3 GHz MacBook Pro with 16GB RAM (macOS Catalina version 10.15.5) in double-precision floating-point format.

5.1 Preparation for numerical tests

From the physical point of view, the diffusion coefficient in the bulk region and the ion channel pore region can be different. In our PNPic software package, we use the diffusion coefficient defined by a piecewise expression as follows:

\[ D_i(r) = \begin{cases} 
D^p_i, & Z_1 + h < z < Z_2 - h \text{ (channel pore region)}, \\
D^p_i + (D^p_i - D^b_i)f_i(r), & Z_2 - h \leq z \leq Z_2 \text{ (upper buffer region)}, \\
D^p_i + (D^p_i - D^b_i)f_b(r), & Z_1 \leq z \leq Z_1 + h \text{ (lower buffer region)}, \\
D^b_i, & z < Z_1 \text{ or } z > Z_2 \text{ (bulk region)},
\end{cases} \]  

(5.1)
Figure 5.1: A cross-section representation of the ion channel-membrane system with the various regions for simulations.

where \( \mathbf{r} = (x, y, z) \), the functions \( f_b \) and \( f_t \) are defined by

\[
f_b(\mathbf{r}) = n \left( \frac{Z_1 + h - z}{h} \right)^{n+1} - (n + 1) \left( \frac{Z_1 + h - z}{h} \right)^n
\] (5.2)

and

\[
f_t(\mathbf{r}) = n \left( \frac{z - Z_2 + h}{h} \right)^{n+1} - (n + 1) \left( \frac{z - Z_2 + h}{h} \right)^n
\] (5.3)

with \( n \) is an integer (the default value is 9), \( h \) is the buffer size (the default value is 3), \( D^b_i \) and \( D^p_i \) denotes the diffusion coefficients of the \( i \)-th ion species in the bulk and channel pore regions, respectively, and \( D^p_i \) is given by

\[
D^p_i = \theta D^b_i \text{ for } 0 < \theta \leq 1.
\] (5.4)

A similar diffusion coefficient \( D_i \) can be found in [22].
We also use the following relative permittivity function $\epsilon_s$ in the solvent region:

$$
\epsilon_s(r) = \begin{cases} 
\epsilon^p_s, & Z_1 + h < z < Z_2 - h \text{ (channel pore region)}, \\
\epsilon^p_s + (\epsilon^p_s - \epsilon^b_s) f_t(r), & Z_2 - h \leq z \leq Z_2 \text{ (upper buffer region)}, \\
\epsilon^p_s + (\epsilon^p_s - \epsilon^b_s) f_b(r), & Z_1 \leq z \leq Z_1 + h \text{ (lower buffer region)}, \\
\epsilon^b_s, & z < Z_1 \text{ or } z > Z_2 \text{ (bulk region)},
\end{cases}
$$

where $\epsilon^b_s$ and $\epsilon^p_s$ denote the relative permittivities in the bulk and channel pore regions, respectively.

The boundary potential value function $g$ is defined by

$$
g(s) = \begin{cases} 
g_b(s), & z = L_{z_1} \text{ (bottom surface of } \Omega), \\
g_t(s), & z = L_{z_2} \text{ (top surface of } \Omega), \\
g_b(s) + (g_t(s) - g_b(s)) k(s), & L_{z_1} < z < L_{z_2} \text{ (lateral surface of } \Omega),
\end{cases}
$$

where $g_b$ and $g_t$ denote the given potential functions, $s = (x, y, z)$, and

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

The boundary concentration value function $g_i(r)$ can be set as

$$
g_i = c^b_i,
$$

where $c^b_i$ is the bulk concentration of the $i$-th ionic species in the solvent region $D_s$, which is
defined by

\[
c_i^b(r) = \begin{cases} 
    c_{i,p}^b(r), & \text{if } Z1 + h < z < Z2 - h \text{ (channel region)}, \\
    c_{i,p}^b(r) + (c_{i,p}^b(r) - c_{i,t}^b(r)) f_i(r), & \text{if } Z2 - h \leq z \leq Z2 \text{ (upper buffer region)}, \\
    c_{i,p}^b(r) + (c_{i,b}^b(r) - c_{i,t}^b(r)) f_b(r), & \text{if } Z1 \leq z \leq Z1 + h \text{ (lower buffer region)}, \\
    c_{i,b}^b(r), & \text{if } Z1 \leq z \leq Z1 + h \text{ (lower bulk region)}, \\
    c_{i,t}^b(r), & \text{if } Z1 \leq z \leq Z1 + h \text{ (upper bulk region)}, 
\end{cases}
\]  

(5.7)

where \(c_{i,p}^b\) is the concentration in the channel pore region (the default value is zero), \(c_{i,b}^b\) and \(c_{i,t}^b\) are the bulk concentrations in the upper and lower bulk regions, respectively.

Particularly, \(D_i^b, \epsilon_i^a, \epsilon_i^b, g_i, g_b, c_{i,b}^b, c_{i,p}^b, c_{i,b}^b, \) and \(c_{i,t}^b\) are given as constants for the \(i\)-th species.

To validate our PNPic software package, we did numerical tests on two realistic ion channel proteins: Gramicidin A (gA) [31, 59] and murine voltage-dependent anion channel isoforms (mVDAC1) [58].

The gA is a natural antibiotic peptides with 15 amino acids (552 atoms) that form cation-permeable channels like pores in biological membranes and lipid bilayers of bacteria and animal cells. Inorganic monovalent ions, such as sodium (Na\(^+\)) and potassium (K\(^+\)), can pass through the channel pores via diffusion. This process can destroy vital ion concentration gradients between membranes and lipid bilayers. The pore is about 4 Å in diameter. Over the last fifty years, this channel has been studied on its ion selectivity, structure [69], etc., producing a lot of experimental data [9, 54]. Thus, the gA channel is regarded as a useful model for validating ion channel models. Figure 5.2 shows the 3D structure of gA in three different forms.

The mVDAC1 is one of the voltage-dependent anion channel isoforms of the mammalian VDAC, which are the most abundant proteins in the mitochondrial membrane [58]. The high-resolution crystal 3D structure of mVDAC1 contains an 19 stranded \(\beta\) barrel with an \(\alpha\)-helix in the middle of the channel pore (see Figure 5.3) [68]. The pore is about 27 Å in diameter at the top and bottom mouths of the channel, but it narrows to about 14 Å near the helix at the center of the ion channel [7]. Although the structural information is available, the complexities of the conductance and the molecular mechanism are not understood well. Recent hypotheses show that the opening or closure of the channel could be controlled through a partial movement of the N-terminal helix with its flexible hinge region (amino
Figure 5.2: Illustrations of gA ion channel (PDB ID: 1MAG) geometry using PyMOL visualization software, color map indicates atoms.
acids 19–25) into or out of the pore. Although an inward α-helical movement could control the channel’s activity through the blockage, it fails to account for the channel’s ion selectivity, which an outward-movement model addresses more adequately [43]. Figure 5.3 shows the 3D structure of the mVDAC1 in three different forms, and the N-terminal α-helix is colored in red in Figure 5.3(a) and 5.3(b).

In our tests, the PDB files of proteins were downloaded from the Orientations of Proteins in Membranes (OPM) database [36] https://opm.phar.umich.edu, and converted to the PQR files by the online tool PDB2PQR [13], which prepares structures for further calculations by reconstructing some missing atoms and assigning atomic charges and radii from specified force fields. OPM provides us with proteins that have been rotated and transformed to match our requirement as illustrated in Figure 2.1, resulting in that $Z_1 = -12$ and $Z_2 = 12$ for mVDAC1 ion channel, $Z_1 = -11$ and $Z_2 = 6$ for gA ion channel.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\epsilon_p$</td>
<td>2</td>
<td>$D_i$ for K$^+$</td>
<td>0.196</td>
</tr>
<tr>
<td>$\epsilon_m$</td>
<td>2</td>
<td>$D_i$ for Ca$^{2+}$</td>
<td>0.0793</td>
</tr>
<tr>
<td>$\epsilon_{b_s}^i$</td>
<td>80</td>
<td>$D_i$ for Na$^+$</td>
<td>0.133</td>
</tr>
<tr>
<td>$c_{b,i,p}^b$</td>
<td>0</td>
<td>$D_i$ for NO$_3^-$</td>
<td>0.19</td>
</tr>
<tr>
<td>$D_i$ for Cl$^-$</td>
<td>0.203</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Relative permittivity and diffusion coefficient constants used in our tests.

Some constants used in our numerical tests are summarized in Table 5.1. These values are collected from the experimental data and the website https://www.aqion.de/site/194.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>gA</th>
<th>mVDAC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>Value</td>
<td></td>
</tr>
<tr>
<td>TMSmesh: [h, d, c]</td>
<td>[0.6, 0.9, 0.9]</td>
<td>[0.2, 0.9, 0.9]</td>
</tr>
<tr>
<td>TetGen: [q, a, T]</td>
<td>[1.3, 0, 0]</td>
<td>[1.4, 0, 0]</td>
</tr>
<tr>
<td>BoxMesh: [$\delta_{z_1}$, $\delta_{z_2}$]</td>
<td>[10, 10]</td>
<td>[10, 10]</td>
</tr>
<tr>
<td>BoxMesh: [$n_x$, $n_y$, $n_z$, $d_z$]</td>
<td>[20, 40, 40, 1]</td>
<td>[10, 20, 20, 2]</td>
</tr>
</tbody>
</table>

Table 5.2: Parameter values of mesh generation

The ion channel meshes used for our numerical tests were generated by the parameters in Table 5.2. Three different box domains defined in Table 5.3 were used to generate meshes,
Figure 5.3: Illustrations of mVDAC1 ion channel (PDB ID: 3EMN) geometry using PyMOL visualization software, color map indicates atoms in (c, d, e, f), and the α-helix is colored in red in (a, b).
and all the meshes in the channel pores were refined. To compare numerical results clearly, we used the method in Chapter 2 to depict the concentrations and potential functions in curves by using $m_1 = 10$, $m_2 = 30$, $m_3 = 10$, and $\bar{h} = 4$.

<table>
<thead>
<tr>
<th></th>
<th>$[\delta x_1, \delta x_2, \delta y_1, \delta y_2, \delta z_1, \delta z_2]$</th>
<th>gA</th>
<th>mVDAC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh 1</td>
<td>[10, 10, 10, 10, 10, 10]</td>
<td>68614 Vertices 385740 Tetrahedra</td>
<td>49798 Vertices 300921 Tetrahedra</td>
</tr>
<tr>
<td>Mesh 2</td>
<td>[30, 30, 30, 30, 10, 10]</td>
<td>72157 Vertices 410080 Tetrahedra</td>
<td>52085 Vertices 316009 Tetrahedra</td>
</tr>
<tr>
<td>Mesh 3</td>
<td>[50, 50, 50, 50, 10, 10]</td>
<td>76019 Vertices 437349 Tetrahedra</td>
<td>53084 Vertices 322728 Tetrahedra</td>
</tr>
</tbody>
</table>

Table 5.3: Box domain dimensions and related mesh data for the three meshes used in numerical tests.

### 5.2 A comparison study on boundary value conditions

Several different boundary value conditions are used in PNP ion channel models. In this section, we present a comparison study on two commonly-used boundary value conditions — the Dirichlet and Neumann boundary value conditions that are stated in Chapter 2. The purpose of such a study is to show how boundary value conditions to affect the quality of a PNPic model. We used two ion channels, gA and mVDAC1, and Mesh 1 in Table 5.3 to do numerical tests, the membrane surface charge $\sigma_t = \sigma_b = 0$, and did the tests.

For gA in a symmetric salt solution of KCl (0.5 M/L) when $g_b = 1.946$ (i.e., 50mV) and $\theta = 0.055$, we calculated the differences between the electrostatic potentials $u$ and concentrations $c_i$ generated by the PNPic model using the Dirichlet and Neumann boundary value conditions, and plotted them in Figures 5.4 and 5.5. From Figure 5.4(a) it can be seen that there is a significant difference between potential results, especially in the solvent region. However, as shown in Figures 5.4(b) and 5.4(c), the differences in concentrations are not big except that the concentration of $K^+$ has significant changes in the entrance of the channel pore. In Figure 5.5 we depicted the differences in 2D curves, from which, we now can clearly see the trends of the differences in the whole solvent region.

We further produced current-voltage (I-V) curves using the PNPic models, and reported them in Figures 5.6 and 5.8. From these figures we can see that different boundary conditions can significantly affect the I-V curve calculation.
Figure 5.4: Color mappings of the differences of the potentials and ionic concentrations generated by the PNPic model using Dirichlet and Neumann boundary value conditions on a cross-section $x = 0$ for $g_A$ in a salt solution of KCl (0.5 M/L) when $g_b = 1.946$ (i.e., 50mV). The protein and membrane regions are colored in green and yellow, respectively.

Figure 5.5: Differences of the potentials and ionic concentrations generated by the PNPic model using Dirichlet and Neumann boundary value conditions for $g_A$ in a salt solution of KCl (0.5 M/L) when $g_b = 1.946$ (i.e., 50mV) in 2D curves.

5.3 Simulation box size affection on ion channel simulation

The simulation box domain size is another possible affection factor for PNP ion channel simulations. To explore such an issue numerically, we set $\sigma_b = \sigma_t = 0$, $\epsilon^p = 80$, $g_t = 0$, $g_b = 3.892$ (i.e., 100 mV), and $\theta = 0.055$ and $\theta = 0.035$ for $g_A$ and mVDAC1 cases,
Figure 5.6: A comparison of the current-voltage curves generated by the PNPic model using Dirichlet and Neumann boundary value conditions for gA in a salt solution of KCl (0.5 M/L).

Figure 5.7: Difference of potentials and ionic concentrations generated by using Dirichlet and Neumann boundary value conditions. All results are generated by PNPic model for mVDAC1 in a mixture solution of KCl (0.5 M/L) and CaCl\(_2\) (0.1 M/L) when \(g_b = 3.892\) (i.e., 100mV).

respectively, to control the value of diffusion constant \(D_p^i\) inside the channel pore. We used a salt solution with 0.5 M/L KCl for gA case and a mixture of 0.5 M/L KCl and 0.1 CaCl\(_2\) for mVDAC1 case. The numerical test results are reported in Figures 5.9 and 5.10.

The differences in electrostatic potential and ionic concentrations are displayed in Figure 5.9 for the gA simulations. From this figure we can see that the absolute difference on potential and ionic concentration was reduced as the box domain size was increased. For example, the largest difference in potential and concentrations are 0.9 and 0.16, respectively,
in the case of using Mesh 1, which are larger than that in the case of using a larger mesh, Mesh 3. As shown in Figure 5.10, the mVDAC1 case has the same trend as the gA case. These test results illustrate that the effects of boundary value conditions on the four side surfaces are insensitive on the PNPic model when the box dimension in the $x/y$-axis direction sufficiently exceeds the width of ion channel.

5.4 Effects of membrane surface charges

To study the effect of membrane surface charges, we did numerical tests by the PNPic model using the Dirichlet boundary value conditions and Mesh 1 defined in Table 5.3. The PNPic model was solved for a symmetric bulk salt concentration, i.e., $c_{i,p}^b = 0$ and $c_{i,t}^b = c_{i,b}^b$.

The electrostatic potential and concentrations on the cross-section $x = 0$ are shown in Figure 5.11 for different membrane surface charges under an applied voltage $g_b = 1.946$ (i.e., 50mV) for the gA ion channel in a salt solution of KCl (0.5 M/L). We also displayed the electrostatic potentials and ionic concentrations in 2D curves in Figure 5.12. From these figures, we can see that adding negative charges on one side surface of membrane attracted more cations $K^+$, and caused a larger negative potential in magnitude near this membrane surface. And adding positive charges on membrane surface increased positive potential, and attracted more anions $Cl^-$ near the membrane surface, but can not enter the channel pore due to the cation selectivity of gA channel.

Furthermore, from Figure 5.12 it can be seen that the positive membrane surface charge $\sigma_b = 20$ significantly increased the positive electrostatic potential from 1.5 to 3 (see Fig-
Figure 5.9: Differences of $u$ and $c_i$ generated by the PNPic model using Dirichlet and Neumann boundary value conditions for gA in a salt solution of KCl (0.5 M/L) on Meshes 1, 2, and 3 given in Table 5.3.
Figure 5.10: Differences of $u$ and $c_i$ generated by the PNPic model using Dirichlet and Neumann boundary value conditions for mVDAC1 in a mixture solution of KCl (0.5 M/L) and CaCl$_2$ (0.1 M/L) on Meshes 1, 2, and 3 given in Table 5.3.
Figure 5.11: Comparisons of the electrostatic potential $u$ and ion concentrations $c_i$ by PNPic model with Dirichlet boundary value conditions and membrane surface charges for gA in a salt solution of KCl (0.5 M/L) on a cross-section $x = 0$. The protein and membrane regions are colored in green and yellow, respectively. Here $g_b = 1.946$ (i.e., 50mV).

ures 5.12(b) and 5.12(d)), and anionic concentrations from 0 to 12 (see Figures 5.12(f) and 5.12(h)), and the anion Cl concentrations are essentially zero inside the channel for all cases. These numerical tests further validate that our PNPic model can predicate the important properties of gA channel, such as cation selectivity.

We also calculated the I-V curves of gA channel to further explore the effects of surface charges. The numerical results were plotted in Figure 5.13. They show that current-voltage
(a) $\sigma_b = 0$, $\sigma_t = 0$

(b) $\sigma_b = 20$, $\sigma_t = 20$

(c) $\sigma_b = -20$, $\sigma_t = -20$

(d) $\sigma_b = 20$, $\sigma_t = -20$

(e) $\sigma_b = 0$, $\sigma_t = 0$

(f) $\sigma_b = 20$, $\sigma_t = 20$

(g) $\sigma_b = -20$, $\sigma_t = -20$

(h) $\sigma_b = 20$, $\sigma_t = -20$

Figure 5.12: Comparisons of the electrostatic potential $u$ and concentrations $c_i$ generated by PNPic model using Dirichlet boundary value conditions and membrane surface charges for gA (Mesh 1) in a salt solution of KCl (0.5 M/L). Here $g_b = 1.946$ (i.e., 50mV).
Figure 5.13: Current-voltage curves for gA ion channel with different membrane surface charges in a salt solution of KCl (0.1 M/L).

curves were affected significantly by membrane surface charges. Here a symmetrical 0.1M bulk concentration, i.e., \( c_{i,a}^b = c_{i,b}^b = 0.1 \text{ M/L} \) was used. The presence of negative charges on the membrane surface increased the current value that passes through the ion channel pore, because more cations \( K^+ \) were attracted into the channel pore. Similarly, positive charges on the membrane surface can expel cations, producing a lower current. These test results are in agreement with experiment observations [54].

We also did tests for mVDAC1 in a mixture solution of KCl (0.1 M/L) and CaCl\(_2\) (0.1 M/L). Here we set \( \theta = 0.35 \) and \( g_b = 1.946 \) (i.e., 50 mV). The numerical results were displayed in Figure 5.14. They are similar to the case of gA ion channel.
Figure 5.14: Comparisons of the electrostatic potential $u$ and concentrations $c_i$ generated by the PNPic model using Dirichlet boundary value conditions and membrane surface charges for mVDAC1 in a mixture solution of KCl (0.1 M/L) and CaCl$_2$ (0.1 M/L). Here $g_b = 1.946$ (i.e., 50 mV).
Chapter 6

Conclusions and future work

In this dissertation, an effective finite element algorithm for solving a PNP ion channel model using Dirichlet boundary value conditions and an improved ion channel finite element mesh algorithm have been presented. They have been implemented as a high quality ion channel simulation software package. By using this package, many numerical tests were done to explore some important issues arisen from ion channel simulations by PNP ion channel models. Here is a summary of the main results of this dissertation:

- A PNP ion channel (PNPic) model using Dirichlet boundary value condition and membrane surface charge is derived.
- An effective PNPic finite element solver is presented.
- New numerical algorithms for generating a triangular surface mesh of the simulation box domain and for constructing a tetrahedral mesh of the membrane region are presented and implemented for an improved ion channel mesh generation package.
- The effects of boundary value conditions, simulation box domain sizes, and membrane surface charges on PNPic modeling are explored numerically.

Based on the current work, we will continue to do the following work in the future:

- In order to take the advantages of both finite difference and finite element methods, a new finite element and finite difference hybrid method for solving a PNP ion channel model will be developed.
- The PNPic model is effective but ignores the finite volume effect of ion particles. In fact, it has been known that ionic size is important for ion transport in highly confined channels. So an improved PNPic model needs to be constructed to consider ionic size effects.
- When we solve the PNPic system numerically, many highly indefinite linear systems...
may be produced, which cannot be solved by any current iterative method such as the GMRES method preconditioned by ILU due to very poor convergent rates. In order to deal with this numerical issue, we will develop preconditioners and new iterative algorithms for our PNPic finite element solver.
Bibliography


[53] W. Rocchia, E. Alexov, and B. Honig. Extending the applicability of the nonlinear
Poisson-Boltzmann equation: multiple dielectric constants and multivalent ions. *The

Membrane surface-charge titration probed by gramicidin A channel conductance. *Bio-

[55] B. Roux, T. Allen, S. Berneche, and W. Im. Theoretical and computational models of

[56] B. Roux, S. Berneche, and W. Im. Ion channels, permeation, and electrostatics: insight


[58] M. J. Sampson, R. S. Lovell, and W. J. Craigen. The murine voltage-dependent anion
channel gene family conserved structure and function. *Journal of Biological Chemistry*,


W. F. van Gunsteren. Definition and testing of the GROMOS force-field versions 54A7

[61] H. Si. TetGen-A quality tetrahedral mesh generator and three-dimensional delaunay tri-
81, 2006.

tetrahedralizations. *Proceedings of the 14th International Meshing Roundtable*, pages


CURRICULUM VITAE

Education

- Ph.D. in Applied Mathematics  
  UWM, Milwaukee, WI.  
  Sep. 2016 - Aug. 2020

- Visiting Scholar  
  UWM, Milwaukee, WI.  

- Doctoral study in Applied Mathematics  
  East China Normal University, Shanghai, China.  
  Sep. 2013 - Sep. 2015

- M.S. in Applied Mathematics  
  Wenzhou University, Wenzhou, China.  
  Jun. 2013

- B.S. in Applied Mathematics  
  Hunan University of Science and Technology, Xiangtan, China.  
  Jun. 2009

Selected List of Awards

1. Distinguished Graduate Student Fellowship, Graduate school, UWM, August 2018.

2. Morris and Miriam Marden Award in Mathematics, Department of Mathematical Sciences, UWM, May 2020.

3. AMS graduate student travel award, AMS, January 2020.


5. Chancellor’s Graduate Student Award, UWM, September 2018, September 2019.


7. Graduate Student Conference Travel Award, Department of Mathematical Sciences, UWM, April 2017, May 2018, June 2019.

**Selected List of Publications**

1. Dexuan Xie, **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*, under revision.


