Lecture Notes on Poisson-Nernst-Planck Modeling and Simulation of Biological Ion Channels

Jinn-Liang Liu

Department of Applied Mathematics, National Hsinchu University of Education, Hsinchu 300, Taiwan. E-mail: jinnliu@mail.nhcue.edu.tw

---

Abstract

The Poisson-Nernst-Planck (PNP) model is a basic continuum model for simulating ionic flows in an open ion channel. It is one of commonly used models in theoretical and computational studies of biological ion channels. The Poisson equation is derived from Coulomb's law in electrostatics and Gauss's theorem in calculus. The Nernst-Planck equation is equivalent to the convection-diffussion model.

3D PNP Projects:

Project A: Linear PNP, Domain: Box without Channel, Exact Solutions without Singular Charges.

Project B: Linear PNP, Domain: Cylinder in Box, Channel: Cylinder, with Exact Solutions without Singular Charges, Goal: Second-Order Convergence.

Project C: Linear GA PNP, Domain: GA in Box, Channel: GA, with Exact Solutions without Singular Charges.

Project D: Nonlinear GA PNP, Singualr Charges, Exact Solutions.

Project E: Nonlinear GA PNP, Singualr Charges, no Exact Solutions, Diffusion Function.

Project F: Nonlinear GA PNP, Singualr Charges, no Exact Solutions, Diffusion Function, van der Waas Potential.

Project G: Nonlinear GA PNP, Singualr Charges, no Exact Solutions, Diffusion Function, Finite Size Effects.

Project H: Poisson-Boltzmann (PB) Model.

Methods: FDM (Finite Difference Method), MIB (Matched Interface and Boundary Method).

Matrix: Nonsymmetric.

Solvers: CG, SOR, BiCGStab. 2011/9/16

---

8 May 2012

1 PNP Models

Biological ion channels seem to be a precondition for all living matter [17]. Ion channels are porous proteins across cell membranes that control many biological functions ranging from signal transfer in the nervous system to regulation of secretion of hormones. Understanding the mechanism of ionic flows within a channel as a function of ionic concentration, membrane potential, and the structure of the channel is a central problem in molecular biophysics [9]. The PNP model proposed by Eisenberg and coworkers [3,7] as a basic continuum model for simulating the ionic flow in an open ion channel is one of commonly used models in theoretical and computational studies of biological ion channels.

For modeling the flow of two species of ions through a channel, the steady-state PNP model reads as

$$\mathrm{P}: \; -\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) = \sum_{j=1}^{N_A} q_j\delta(\mathbf{r}-\mathbf{r}_j) + \sum_{i=1}^{2} q_iC_i + F, \quad \mathbf{r}\in\Omega \tag{1.1}$$

$$\mathrm{NP1}: \quad -\nabla\cdot\mathbf{J}_1(\mathbf{r}) = \quad F_1, \quad \mathbf{r}\in\Omega_s, \tag{1.2}$$

$$\mathrm{NP2}: \quad -\nabla\cdot\mathbf{J}_2(\mathbf{r}) = \quad F_2, \quad \mathbf{r}\in\Omega_s, \tag{1.3}$$

$$\mathbf{J}_i(\mathbf{r}) = -D_i(\mathbf{r})\left[\nabla C_i(\mathbf{r}) + \beta_i C_i(\mathbf{r})\nabla\phi(\mathbf{r})\right] \tag{1.4}$$

where ϕ is the electrostatic potential, ϵ the electric permittivity, N_A the total number of atomic (partial) charges q_j located (fixed) at $\mathbf{r}_j = (x_j, y_j, z_j)$ in the channel protein, $\delta(\mathbf{r}-\mathbf{r}_j)$ the delta function (and hence q_j are singular charges), C_i the concentration of an ion species i carrying charge q_i (for example, $q_{K^+} = +1e$, $q_{Cl^-} = -1e$), $\mathbf{J}_i$ the concentration flux (current density), D_i the spatially dependent diffusion coefficient, $\beta_i = q_i/(k_BT)$, k_B the Boltzmann constant, T the absolute temperature, and e the proton charge. Note that the diffusion coefficient D_i and the parameter β_i are related to the mobility coefficient μ_i by Einstein's relation $\mu_i = |\beta_i|\,D_i$. The domain $\overline{\Omega} = \overline{\Omega}_s \cup \overline{\Omega}_m$ consists of two subdomains, namely, the solvent subdomain $\overline{\Omega}_s$ and the biomolecular subdomain $\overline{\Omega}_m$. The electric permittivity has different values in subdomains

$$\epsilon(\mathbf{r}) = \epsilon_r\epsilon_0 = \begin{cases} \epsilon_s\epsilon_0, \; \forall\mathbf{r}\in\Omega_s \\ \epsilon_m\epsilon_0, \; \forall\mathbf{r}\in\Omega_m \end{cases} \tag{1.5}$$

where ϵ_0 is the vacuum permittivity, $\epsilon_r = \epsilon_s$ is the dielectric constant (relative permittivity) of the solvent, and $\epsilon_r = \epsilon_m$ is the dielectric constant of the molecules. In most occurrences, we shall omit ϵ_0 if there is no danger of confusion. We consider the domain as a cubical box

$$\mathrm{Box} = \Omega = (-20\text{Å}, 20\text{Å}) \times (-20\text{Å}, 20\text{Å}) \times (-20\text{Å}, 20\text{Å}). \tag{1.6}$$

The channel protein is embedded in the biomolecular subdomain, for which we consider

$$\text{Channel} : \begin{cases} \text{None for Project A,} \\ \text{Cylinder for Project B,} \\ \text{GA for Projects C, D, E.} \end{cases} \tag{1.7}$$

The model is nonlinear because the unknown functions ϕ, C_1, and C_2 are coupled together in (1.1)-(1.3).

Note that

$$F = F_i = 0 \tag{1.8}$$

in the real PNP model without reaction. They are usually not equal to zero if we want to construct exact solutions for ϕ and C_i in order to test whether our numerical methods and our code are correct before applying to a real model problem for which we know that the solutions exist but cannot be expressed in analytical forms (called analytical solutions).

Introducing the Slotboom variable $\widehat{C}_i$ [19] by

$$C_i = \widehat{C}_i \exp(-\beta_i \phi) \tag{1.9}$$

the concentration flux is then reformulated to

$$\mathbf{J}_i = -D_i \exp(-\beta_i \phi) \nabla \widehat{C}_i = -\alpha_i \nabla \widehat{C}_i, \;\; \alpha_i = D_i \exp(-\beta_i \phi) \tag{1.10}$$

Consequently, the self-adjoint PNP is

$$-\nabla \cdot (\epsilon \nabla \phi) = \sum_{j=1}^{N_A} q_j \delta(\mathbf{r} - \mathbf{r}_j) + \sum_{i=1}^{2} q_i C_i, \tag{1.11}$$

$$-\nabla \cdot \mathbf{J}_i = \nabla \cdot \left[\alpha_i \nabla \widehat{C}_i \right] = 0. \tag{1.12}$$

Dirichlet boundary conditions (BCs) for both P and NP equations (1.1)-(1.3) are assumed, namely,

$$\phi(\mathbf{r}) = g(\mathbf{r}), \forall \mathbf{r} \in \partial\Omega, \tag{1.13}$$

$$C_i(\mathbf{r}) = g_i(\mathbf{r}), \forall \mathbf{r} \in \partial\Omega_s \backslash \Gamma, \tag{1.14}$$

where

$$\Gamma = \overline{\Omega}_s \cap \overline{\Omega}_m \tag{1.15}$$

is the interface set between $\overline{\Omega}_s$ and $\overline{\Omega}_m$. One of the main concerns of this lecture note is to properly handle the jump conditions associated with the interface.

Fig. 1 illustrates a VMD [10] simulation system of the KcsA channel with membrane, water, and ions [1]. The channel protein is in the central part of

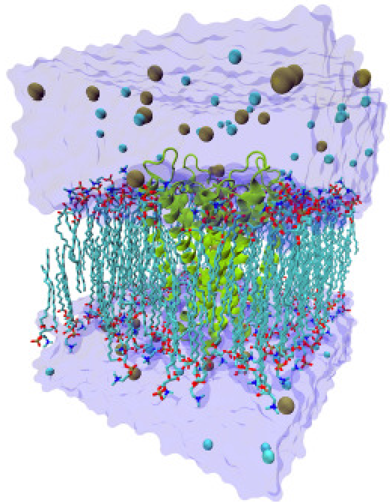

Fig. 1. VMD simulation system of the KcsA channel with membrane, water, and ions.

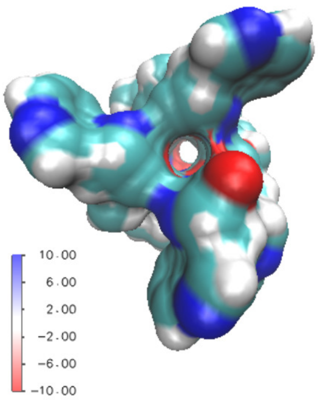

Fig. 2. Top view of GA channel.

the simulation domain (a box) as shown in green color. The membrane consists of bilipid layers shown in light blue surrounding the channel. The upper and lower regions represent the extracellular (outside of a cell) and intracellular (inside) solvent regions, respectively, that consist of water and ions. Fig. 2 is a top view of the Gramicidin A (GA) channel generated by the VMD program. Fig. 3 is a side view of the GA channel embedded in the membrane [21]. Fig. 4 is a cross section of a 3D PNP simulation domain for the GA channel [4].

2 Linear and Nonlinear PNP

For both linear and nonlinear PNP models, the exact solutions [21] are chosen to be

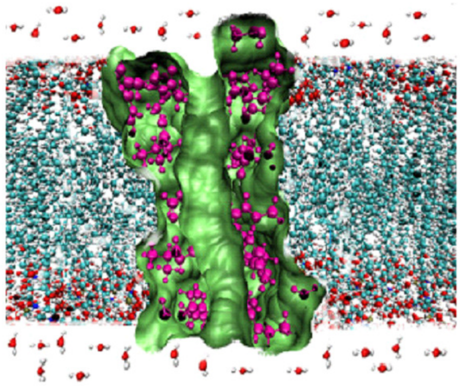

Fig. 3. Side view of the GA channel embedded in the membrane.

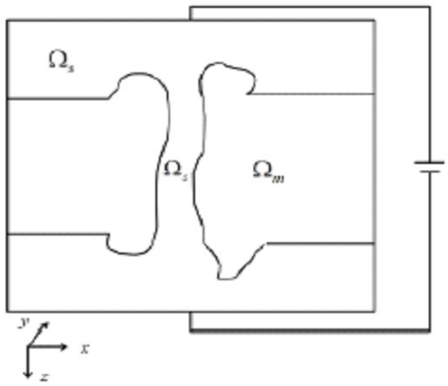

Fig. 4. A cross section of 3D PNP simulation domain for GA channel.

$$\text{ExactSolP()} \;:\; \phi^{\text{Ex}}(\mathbf{r}) = \cos x \cos y \cos z,\; \mathbf{r} \in \Omega \tag{2.1}$$

$$\text{ExactSolNP1()} \;:\; C_1^{\text{Ex}}(\mathbf{r}) = \begin{cases} 0, & \mathbf{r} \in \Omega_m \\ 0.2 \cos x \cos y \cos z + 0.3, & \mathbf{r} \in \Omega_s \end{cases} \tag{2.2}$$

$$\text{ExactSolNP2()} \;:\; C_2^{\text{Ex}}(\mathbf{r}) = \begin{cases} 0, & \mathbf{r} \in \Omega_m \\ 0.1 \cos x \cos y \cos z + 0.3, & \mathbf{r} \in \Omega_s \end{cases} \tag{2.3}$$

$$D_1 = 1,\; \beta_1 = 1,\; D_2 = 1,\; \beta_2 = -1,\; \text{for Projects A-D}, \tag{2.4}$$

$$\text{m\_diM} = \epsilon_m = 1,\; \text{m\_diS} = \epsilon_s = 80. \tag{2.5}$$

Note that the naming convention like ExactSolP() is used in our 3DPNP code in conjunction with the mathematical notation used in the lecture notes. The linear PNP model means that P, NP1, and NP2 (1.1)-(1.3) are decoupled (independent of each other). For example, the right hand sides of (1.1)-(1.3) are chosen as

$$\text{P:} \quad \sum_{i=1}^{2} q_i C_i = 0,$$

$$F = \begin{cases} 3\epsilon_m \cos x \cos y \cos z \text{ in } \Omega_m \\ 3\epsilon_s \cos x \cos y \cos z \text{ in } \Omega_s \end{cases} \tag{2.6}$$

$$\text{NP1:} \quad F_1 = -\nabla \cdot \mathbf{J}_1 = \nabla \cdot \left[D_1 \left(\nabla C_1^{\text{Ex}} + \beta_1 C_1^{\text{Ex}} \nabla \phi^{\text{Ex}} \right) \right] \tag{2.7}$$

$$= D_1 \Delta C_1^{\text{Ex}} + D_1 \beta_1 \left(C_1^{\text{Ex}} \Delta \phi^{\text{Ex}} + \nabla C_1^{\text{Ex}} \cdot \nabla \phi^{\text{Ex}} \right) \tag{2.8}$$

$$= \text{D1*DelC1} + ... \tag{2.9}$$

$$\text{NP2:} \quad F_2 = D_2 \Delta C_2^{\text{Ex}} + D_2 \beta_2 \left(C_2^{\text{Ex}} \Delta \phi^{\text{Ex}} + \nabla C_2^{\text{Ex}} \cdot \nabla \phi^{\text{Ex}} \right) \tag{2.10}$$

m_Func2[i]=D1*DelC1+D1*beta1*(C1*DelPhi
+C1Dx*PhiDx+C1Dy*PhiDy+C1Dz*PhiDz) (2.11)

m_Func3[i]=D2*DelC2+D2*beta2*(C2*DelPhi
+C2Dx*PhiDx+C2Dy*PhiDy+C2Dz*PhiDz) (2.12)

3 Domain Notation for Projects B, C, D, E

With Figs. 3 and 4, the following domain notation is adopted in our model and in the 3DPNP code (for both Cylinder and GA Channels).

Define m_NodeTpye[i] =

(1) ‘P’ (the protein and membrane regions not including the channel wall),
(2) ‘C’ (the channel pore region not including the channel wall)
(3) ‘W’ (the channel wall of the pore region),
(4) ‘E’ (the extracellular solvent region not including the interface)
(5) ‘I’ (the intracellular solvent region not including the interface)
(6) ‘F’ (the interface between ‘P’ and ‘C’, ‘E’, or ‘I’)
(7) ‘1’ (the East side face (boundary) of the box in X axis)
(8) ‘2’ (the West side face of the box)
(9) ‘3’ (the South side face of the box in Y axis)
(10) ‘4’ (the North side face of the box)
(11) ‘5’ (the Down side face of the box in Z axis. The positive direction of Z is pointing upward. The origin is at the center of the channel or of the box.)
(12) ‘6’ (the Up side face of the box)

- ‘P’, ‘W’, ‘F’ $\subset \Omega_m$,
- ‘C’, ‘E’, ‘I’, $\subset \Omega_s$,
- ‘5’, ‘6’ $\subset \overline{\Omega}_s \cap \partial\Omega$; $\partial\Omega$ = the boundary of Ω, $\overline{\Omega}_s$ = the closure of Ω_s,
- ‘1’, ‘2’, ‘3’, ‘4’ $\subset \overline{\Omega}_s \cap \partial\Omega$ or $\subset \overline{\Omega}_m \cap \partial\Omega$,

- 'W' $\cup$ 'F' $= \Gamma$ (the interface $\overline{\Omega}_s \cap \overline{\Omega}_m$).

4 MIB for P

Discretization of the left hand side of (1.1) by the central finite difference method (FDM) yields

$$-\frac{\partial}{\partial x}\left(\epsilon(\mathbf{r})\frac{\partial \phi(x_i,y,z)}{\partial x}\right) \approx \frac{-\epsilon_{i-\frac{1}{2}}\phi_{i-1} + \left(\epsilon_{i-\frac{1}{2}} + \epsilon_{i+\frac{1}{2}}\right)\phi_i - \epsilon_{i+\frac{1}{2}}\phi_{i+1}}{\Delta x^2} \tag{4.1}$$

for all $(x_i, y, z) \in \Omega_m$ or Ω_s, $\frac{\partial \phi(x,y,z)}{\partial x} = \phi_x$, $\phi_i \approx \phi(x_i, y, z)$, $\Delta x = x_i - x_{i-1} = h$, and x_i are FD grid points. We assume a uniform partition of the box in each direction, i.e., $\Delta x = \Delta y = \Delta z = h$. To simplify the notation, we write (1.1) in 1D as

$$-\frac{\partial}{\partial x}\left(\epsilon(x)\frac{\partial \phi(x)}{\partial x}\right) = f \tag{4.2}$$

The second-order, denoted by $O(h^2)$ (convergence order is 2), central FD approximation of (4.2) is

$$\frac{-\epsilon_{i-\frac{1}{2}}\phi_{i-1} + \left(\epsilon_{i-\frac{1}{2}} + \epsilon_{i+\frac{1}{2}}\right)\phi_i - \epsilon_{i+\frac{1}{2}}\phi_{i+1}}{\Delta x^2} = f_i \tag{4.3}$$

For interface problems, we always assume that

$$x_{i-1} < \gamma = x_{i-\frac{1}{2}} < x_i, \tag{4.4}$$

i.e., every jump position $\gamma \in \Gamma =$ 'W' $\cup$ 'F' is at the middle of some neighboring grid points. We consider the following jump conditions for the P problem (1.1)

$$[\phi] = 0, \text{ for both linear and nonlinear PNP} \tag{4.5}$$

$$[\epsilon\phi_{\mathbf{n}}] = \begin{cases} \epsilon_m \nabla\phi \cdot \mathbf{n} - \epsilon_s \nabla\phi \cdot \mathbf{n} \neq 0, \text{ for linear,} \\ \epsilon_m \nabla\phi \cdot \mathbf{n} - \epsilon_s \nabla\phi \cdot \mathbf{n} = 0, \text{ for nonlinear,} \end{cases} \tag{4.6}$$

where $\mathbf{n}$ is an outward normal unit vector on Γ (see Fig. 5).

The jump is denoted by

$$[\phi] = \phi^+ - \phi^-,\ \phi^+ = \lim_{x\to\gamma^+}\phi(x),\ \phi^- = \lim_{x\to\gamma^-}\phi(x),\ \gamma^- \in \Omega_s,\ \ \gamma^+ \in \Omega_m. \tag{4.7}$$

Therefore, if

$$x_{i-1} = \text{'I', 'C' or 'E'} \in \Omega_s,\ x_{i-\frac{1}{2}} = \gamma,\ x_i = \text{'W' or 'F'} \in \Omega_m, \tag{4.8}$$

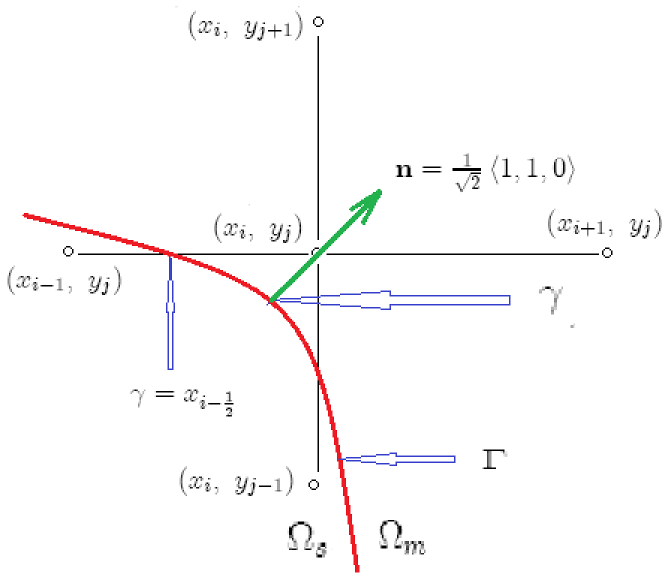

Fig. 5. Interface position γ.

then

$$[\epsilon] = \epsilon^+ - \epsilon^- = \epsilon_m - \epsilon_s. \tag{4.9}$$

For (4.6) in 1D, we have

$$[\epsilon\phi_{\mathbf{n}}] = \epsilon_m\nabla\phi\cdot\mathbf{n} - \epsilon_s\nabla\phi\cdot\mathbf{n} = \begin{cases} \epsilon_m\phi_x - \epsilon_s\phi_x, & \mathbf{n} = \langle 1,0,0\rangle \\ -\epsilon_m\phi_x + \epsilon_s\phi_x, & \mathbf{n} = \langle -1,0,0\rangle \,, \end{cases} = [\epsilon\phi_x] \tag{4.10}$$

The main ideas of the MIB (matched interface and boundary) method [21] for handling the jump problems are

(1) considering (4.2) as two different subproblems with two disjoint subdomains $x < \gamma$ and $x > \gamma$,
(2) taking the jump conditions (4.5) and (4.6) as the boundary conditions for each subproblem with respect to its subdomain,
(3) extending smoothly a function $\phi(x)$ defined on a subdomain to a 'fictitious' function $\Psi(x)$ defined on another subdomain, and
(4) applying Taylor's theorem to the jump conditions for joining two subproblems back to one.

Define the extension functions

$$F(x) = \begin{cases} \phi(x) \text{ if } x < \gamma \\ \Psi(x) \text{ if } x \geq \gamma \end{cases} \text{ or } G(x) = \begin{cases} \Psi(x) \text{ if } x \leq \gamma \\ \phi(x) \text{ if } x > \gamma \end{cases}. \tag{4.11}$$

Applying Taylor's theorem to $F(x)$ at the interface, we have

$$F(x_{i-1}) = F(\gamma) + F'(\gamma)(x_{i-1} - \gamma) + \frac{F''(\gamma)}{2!}(x_{i-1} - \gamma)^2 + O(h^3) \tag{4.12}$$

$$F(x_i) = F(\gamma) + F'(\gamma)(x_i - \gamma) + \frac{F''(\gamma)}{2!}(x_i - \gamma)^2 + O(h^3) \tag{4.13}$$

$$F(\gamma) = \frac{F(x_{i-1}) + F(x_i)}{2} + O(h^2) \tag{4.14}$$

Hence, for (4.5), we have

$$\phi^- = F(\gamma) = \frac{\phi_{i-1} + \Psi_i}{2} + O(h^2) \tag{4.15}$$

$$\phi^+ = G(\gamma) = \frac{\Psi_{i-1} + \phi_i}{2} + O(h^2) \tag{4.16}$$

$$[\phi] = \frac{\Psi_{i-1} + \phi_i}{2} - \frac{\phi_{i-1} + \Psi_i}{2} + O(h^2) \tag{4.17}$$

Similarly for (4.6), subtracting (4.12) from (4.13) gives

$$hF'(\gamma) = F(x_i) - F(x_{i-1}) + O(h^3) \tag{4.18}$$

$$\phi_x^- = F'(\gamma) = \frac{\Psi_i - \phi_{i-1}}{h} + O(h^2) \tag{4.19}$$

$$\phi_x^+ = G'(\gamma) = \frac{\phi_i - \Psi_{i-1}}{h} + O(h^2) \tag{4.20}$$

$$[\epsilon\phi_x] = \epsilon_m\epsilon_0\frac{\phi_i - \Psi_{i-1}}{h} - \epsilon_s\epsilon_0\frac{\Psi_i - \phi_{i-1}}{h} + O(h^2) \tag{4.21}$$

Therefore, by (4.17) and (4.21), the following equations

$$A_1\phi_{i-1} + A_2\Psi_i = A_3\Psi_{i-1} + A_4\phi_i - [\phi] \tag{4.22}$$

$$\epsilon^-\left(B_1\phi_{i-1} + B_2\Psi_i\right) = \epsilon^+\left(B_3\Psi_{i-1} + B_4\phi_i\right) - [\epsilon\phi_x] \tag{4.23}$$

represent FD approximations of (4.5) and (4.6), respectively, with local truncation errors of $O(h^2)$. Here, the weights are

$$A_1 = A_2 = A_3 = A_4 = \frac{1}{2}, \tag{4.24}$$

$$B_1 = \frac{-1}{h},\ B_2 = \frac{1}{h},\ B_3 = \frac{-1}{h},\ B_4 = \frac{1}{h}. \tag{4.25}$$

Solving (4.22) and (4.23) for the fictitious values Ψ_i and Ψ_{i+1}, we obtain

$$
\begin{aligned}
\Psi_{i-1} = & \frac{\left(\epsilon^- B_2 A_1 - \epsilon^- B_1 A_2\right)\phi_{i-1} - \left(\epsilon^- B_2 A_4 - \epsilon^+ B_4 A_2\right)\phi_i}{\left(\epsilon^- B_2 A_3 - \epsilon^+ B_3 A_2\right)} \\
& + \frac{\epsilon^- B_2 [\phi] - A_2 [\epsilon\phi_x]}{\left(\epsilon^- B_2 A_3 - \epsilon^+ B_3 A_2\right)} \\
= & C_1\phi_{i-1} + C_2\phi_i + C_0 \qquad (4.26) \\
\Psi_i = & \frac{-\left(\epsilon^+ B_3 A_1 - \epsilon^- B_1 A_3\right)\phi_{i-1} + \left(\epsilon^+ B_3 A_4 - \epsilon^+ B_4 A_3\right)\phi_i}{\left(\epsilon^+ B_3 A_2 - \epsilon^- B_2 A_3\right)} \\
& + \frac{-\epsilon^+ B_3 [\phi] + A_3 [\epsilon\phi_x]}{\left(\epsilon^+ B_3 A_2 - \epsilon^- B_2 A_3\right)} \\
= & D_1\phi_{i-1} + D_2\phi_i + D_0 \qquad (4.27)
\end{aligned}
$$

Following (4.3) by differencing $F(x)$ at the grid point x_{i-1} and differencing $G(x)$ at the grid point x_i, we obtain

$$
\frac{-\epsilon_{i-\frac{3}{2}}\phi_{i-2} + \left(\epsilon_{i-\frac{3}{2}} + \epsilon^-_{i-\frac{1}{2}}\right)\phi_{i-1} - \epsilon^-_{i-\frac{1}{2}}\Psi_i}{\Delta x^2} = f_{i-1} \qquad (4.28)
$$

$$
\frac{-\epsilon^+_{i-\frac{1}{2}}\Psi_{i-1} + \left(\epsilon^+_{i-\frac{1}{2}} + \epsilon_{i+\frac{1}{2}}\right)\phi_i - \epsilon_{i+\frac{1}{2}}\phi_{i+1}}{\Delta x^2} = f_i \qquad (4.29)
$$

Although Ψ_i and Ψ_{i-1} are called fictitious (ghost) values, they are real in implementation and defined by (4.26) and (4.27) via ϕ_{i-1} and ϕ_i. Consequently, (4.28) and (4.29) become

$$
\frac{-\epsilon_{i-\frac{3}{2}}\phi_{i-2} + \left(\epsilon_{i-\frac{3}{2}} + (1-D_1)\,\epsilon^-_{i-\frac{1}{2}}\right)\phi_{i-1} - D_2\epsilon^-_{i-\frac{1}{2}}\phi_i}{\Delta x^2} = f_{i-1} + \frac{\epsilon^-_{i-\frac{1}{2}} D_0}{\Delta x^2} \qquad (4.30)
$$

$$
\frac{-C_1\epsilon^+_{i-\frac{1}{2}}\phi_{i-1} + \left((1-C_2)\,\epsilon^+_{i-\frac{1}{2}} + \epsilon_{i+\frac{1}{2}}\right)\phi_i - \epsilon_{i+\frac{1}{2}}\phi_{i+1}}{\Delta x^2} = f_i + \frac{\epsilon^+_{i-\frac{1}{2}} C_0}{\Delta x^2} \qquad (4.31)
$$

or (by $\gamma = x_{i-\frac{1}{2}}$)

$$
\frac{-\epsilon_s\phi_{i-2} + \left(\epsilon_s + (1-D_1)\,\epsilon_s\right)\phi_{i-1} - D_2\epsilon_s\phi_i}{\Delta x^2} = f_{i-1} + \frac{\epsilon_s D_0}{\Delta x^2} \qquad (4.32)
$$

$$
\frac{-C_1\epsilon_m\phi_{i-1} + \left((1-C_2)\,\epsilon_m + \epsilon_m\right)\phi_i - \epsilon_m\phi_{i+1}}{\Delta x^2} = f_i + \frac{\epsilon_m C_0}{\Delta x^2} \qquad (4.33)
$$

$$
\begin{aligned}
& \frac{-C_1\epsilon_m\phi_{i-1,j} + \left((1-C_2)\,\epsilon_m + \epsilon_m\right)\phi_{ij} - \epsilon_m\phi_{i+1,j}}{\Delta x^2} \\
+ & \frac{-C_1\epsilon_m\phi_{i,j-1} + \left((1-C_2)\,\epsilon_m + \epsilon_m\right)\phi_{ij} - \epsilon_m\phi_{i,j+1}}{\Delta y^2} \\
& = f_{ij} + \frac{\epsilon_m C_0}{\Delta x^2} + \frac{\epsilon_m C_0}{\Delta y^2}, \text{ (for 2 jumps in 2D)}, \qquad (4.34)
\end{aligned}
$$

where

$$
\begin{aligned}
C_1 &= \frac{\epsilon^- B_2 A_1 - \epsilon^- B_1 A_2}{\epsilon^- B_2 A_3 - \epsilon^+ B_3 A_2} = \frac{\epsilon_s B_2 - \epsilon_s B_1}{\epsilon_s B_2 - \epsilon_m B_3} = \frac{2\epsilon_s}{\epsilon_m + \epsilon_s} \\
C_2 &= \frac{-\left(\epsilon^- B_2 A_4 - \epsilon^+ B_4 A_2\right)}{\epsilon^- B_2 A_3 - \epsilon^+ B_3 A_2} = \frac{-\epsilon_s B_2 + \epsilon_m B_4}{\epsilon_s B_2 - \epsilon_m B_3} = \frac{\epsilon_m - \epsilon_s}{\epsilon_m + \epsilon_s} \\
C_0 &= \frac{\epsilon^- B_2 [\phi] - A_2 [\epsilon \phi_x]}{\epsilon^- B_2 A_3 - \epsilon^+ B_3 A_2} = \frac{2\epsilon_s B_2 [\phi] - [\epsilon \phi_x]}{\epsilon_s B_2 - \epsilon_m B_3} = \frac{2\epsilon_s [\phi] - h [\epsilon \phi_x]}{\epsilon_m + \epsilon_s}
\end{aligned} \tag{4.35}
$$

$$
\begin{aligned}
D_1 &= \frac{-\left(\epsilon^+ B_3 A_1 - \epsilon^- B_1 A_3\right)}{\epsilon^+ B_3 A_2 - \epsilon^- B_2 A_3} = \frac{-\left(\epsilon_m B_3 - \epsilon_s B_1\right)}{\epsilon_m B_3 - \epsilon_s B_2} = \frac{-\left(\epsilon_m - \epsilon_s\right)}{\epsilon_m + \epsilon_s} \\
D_2 &= \frac{\epsilon^+ B_3 A_4 - \epsilon^+ B_4 A_3}{\epsilon^+ B_3 A_2 - \epsilon^- B_2 A_3} = \frac{\epsilon_m B_3 - \epsilon_m B_4}{\epsilon_m B_3 - \epsilon_s B_2} = \frac{2\epsilon_m}{\epsilon_m + \epsilon_s} \\
D_0 &= \frac{-\epsilon^+ B_3 [\phi] + A_3 [\epsilon \phi_x]}{\epsilon^+ B_3 A_2 - \epsilon^- B_2 A_3} = \frac{-2\epsilon_m B_3 [\phi] + [\epsilon \phi_x]}{\epsilon_m B_3 - \epsilon_s B_2} \\
&= \frac{-2\epsilon_m [\phi] - h [\epsilon \phi_x]}{\epsilon_m + \epsilon_s}
\end{aligned} \tag{4.36}
$$

It can be easily seen that (4.32) and (4.33) reduce to the standard FD equation (4.3) when $\epsilon_m = \epsilon_s$ (no jump). For $\epsilon_m = 1$, $\epsilon_s = 80$, and $[\phi] = 0$, we have

$$
C_1 = \frac{2 \cdot 80}{81},\ C_2 = \frac{-79}{81},\ C_0 = \frac{-h [\epsilon \phi_x]}{81},\ 1 - C_2 = \frac{2 \cdot 80}{81}, \tag{4.37}
$$

$$
D_1 = \frac{79}{81},\ D_2 = \frac{2}{81},\ D_0 = \frac{-h [\epsilon \phi_x]}{81},\ 1 - D_1 = \frac{2}{81}, \tag{4.38}
$$

which lead to a diagonally dominant matrix from (4.32) and (4.33).

By (4.22) and (4.23), we introduce two unknowns Ψ_{i-1} and Ψ_i in order to treat the two jump conditions $[\phi]$ and $[\epsilon \phi_x]$. If $[\phi] = 0$, we actually have only one jump condition $[\epsilon \phi_x]$ to take care of. Hence, we should let either $\Psi_i = \phi_i$ or $\Psi_{i-1} = \phi_{i-1}$ in (4.23). If we let $\Psi_i = \phi_i$, then (4.22) becomes

$$
A_1 \phi_{i-1} = A_3 \Psi_{i-1} - [\phi] \tag{4.39}
$$

which means that the fictitious value Ψ_{i-1} will cause an $O(h^2)$ error to approximate $[\phi]$ if (4.29) is in use. The next question is from which of (4.32) and (4.33) we should choose. Numerical results show that (4.33) is better. Nevertheless, if both $[\phi] \neq 0$ and $[\epsilon \phi_x] \neq 0$, we should use both.

Example 4.1. $\epsilon_m = \epsilon_s = 1$ (no jump), $\phi(\mathbf{r})$ is given as (2.1). m_XPts = 21, 41, 81, 161 $\Longrightarrow$ h = 2Å, 1Å, 0.5Å, 0.25Å. The naming convention of Table 4.1A-P represents the P equation of Project A. The conjugate-gradient method (CG) is used for solving matrix systems. The standard FD (4.3) for the box case without jumps yields $O(h^2)$ in the infinity error norm, i.e., $E_\infty = \max_{ijk}\left|\phi(x_i, y_j, z_k) - \phi_{ijk}\right|$, as shown in 4.1A-P.

Example 4.2. Cylinder, $\epsilon_m = 1$, $\epsilon_s = 80$, $[\phi] = 0$, $[\epsilon\phi_{\mathbf{n}}] \neq 0$, $\phi(\mathbf{r})$ is given as (2.1). Due to the interface condition (4.6), the resulting matrix systems are not symmetric, an SOR linear solver is used for this example. Numerical results in Table4.2B-P obtained by the MIB method (4.33) for the cylinder case with jumps also show an $O(h^2)$ convergence.

Table 4.1A-P. (4.3) & CG				Table 4.2B-P. (4.33) & SOR		
h in Å	E_∞	Order	Time	E_∞	Order	Time
2	0.4122			0.4442		
1	0.0877	2.23		0.0926	2.26	
0.5	0.0211	2.06		0.0227	2.03	
0.25	0.0052	2.02	1m44s	0.0057	1.99	5m41s

5 FDM for Linear NP in Primitive and Slotboom Forms

We first consider the NP equation in the primitive form, i.e. (1.2) or (1.3), and simplify it in 1D as

$$-\frac{\partial J}{\partial x} = \frac{\partial}{\partial x}\left(\left[D\left(\frac{\partial C}{\partial x} + \beta C\frac{\partial \phi}{\partial x}\right)\right]\right) = f. \tag{5.1}$$

Differencing (5.1) at x_i gives

$$\begin{aligned}
-\frac{\partial J(x_i, y, z)}{\partial x} &\approx -\frac{J_{i+\frac{1}{2}} - J_{i-\frac{1}{2}}}{\Delta x} && (5.2)\\
-J_{i+\frac{1}{2}} &\approx \left[D\left(\frac{\partial C}{\partial x} + \beta C\frac{\partial \phi}{\partial x}\right)\right]_{i+\frac{1}{2}} && (5.3)\\
&\approx \left[D_{i+\frac{1}{2}}\frac{C_{i+1} - C_i}{\Delta x} + D_{i+\frac{1}{2}}\beta_{i+\frac{1}{2}}\frac{C_{i+1} + C_i}{2}\frac{\phi_{i+1} - \phi_i}{\Delta x}\right] && (5.4)\\
-J_{i-\frac{1}{2}} &\approx \left[D_{i-\frac{1}{2}}\frac{C_i - C_{i-1}}{\Delta x} + D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\frac{C_i + C_{i-1}}{2}\frac{\phi_i - \phi_{i-1}}{\Delta x}\right] && (5.5)\\
-\frac{\partial J(x_i, y, z)}{\partial x} &\approx \frac{1}{\Delta x^2}\left[a_{i-1}C_{i-1} + a_i C_i + a_{i+1}C_{i+1}\right] && (5.6)
\end{aligned}$$

$$
\begin{aligned}
a_{i-1} &= D_{i-\frac{1}{2}} - D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\left(\phi_i - \phi_{i-1}\right)/2 \\
a_i &= -\left(D_{i-\frac{1}{2}} + D_{i+\frac{1}{2}}\right) - D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\left(\phi_i - \phi_{i-1}\right)/2 \\
&\quad + D_{i+\frac{1}{2}}\beta_{i+\frac{1}{2}}\left(\phi_{i+1} - \phi_i\right)/2 \\
a_{i+1} &= D_{i+\frac{1}{2}} + D_{i+\frac{1}{2}}\beta_{i+\frac{1}{2}}\left(\phi_{i+1} - \phi_i\right)/2
\end{aligned}
\tag{5.7}
$$

$$
\frac{a_{i-1}C_{i-1} + a_iC_i + a_{i+1}C_{i+1}}{\Delta x^2} = f_i \tag{5.8}
$$

The flux condition for the NP problems is

$$
\mathbf{J}\cdot\mathbf{n} = g \begin{cases} = 0 \text{ real PNP,} \\ \neq 0 \text{ with exact solutions,} \end{cases} \quad \text{on } \Gamma \tag{5.9}
$$

where the interface Γ is actually a part of the boundary $\partial\Omega_s$ (see Fig. 5). For this, we write in 1D as

$$
\begin{aligned}
\mathbf{J}\cdot\mathbf{n} &= \mathbf{J}\cdot\langle 1,0,0\rangle = J^x \\
J^x &= -D\left(\frac{\partial C}{\partial x} + \beta C\frac{\partial \phi}{\partial x}\right) = g \ \text{ at } \gamma.
\end{aligned}
\tag{5.10}
$$

Note that (5.10) is a BC for the NP problems not an interface condition. Moreover, it is usually called the Robin BC since it involves the data of both the unknown function C itself and its derivative $\frac{\partial C}{\partial x}$. If a BC is in terms of C only, it is then called a Dirichlet BC and is called a Neumann BC if in terms of $\frac{\partial C}{\partial x}$ only. We discuss the FD approximation of (5.10) in two cases.

Case 1. $\mathbf{n} = \langle 1,0,0\rangle$, x_{i-1} = 'C', x_i = 'W', and $\gamma = x_{i-\frac{1}{2}}$.

Let

$$
J^x_{i-\frac{1}{2}} = -\left[D\left(\frac{\partial C}{\partial x} + \beta C\frac{\partial \phi}{\partial x}\right)\right]_{i-\frac{1}{2}} \tag{5.11}
$$

FD approximation of (5.10) at $\gamma = x_{i-\frac{1}{2}}$ is

$$
-D_{i-\frac{1}{2}}\frac{\Psi_i - C_{i-1}}{\Delta x} - D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\frac{\Psi_i + C_{i-1}}{2}\frac{\phi_i - \phi_{i-1}}{\Delta x} = g_{i-\frac{1}{2}} = J^x_{i-\frac{1}{2}} \tag{5.12}
$$

where Ψ_i is a fictitious value. We can extend the function $C(x)$ continuously from x_{i-1} = 'C' to x_i = 'W' by considering Ψ_i as an extra unknown that approximates the ghost value $C(x_i)$. This i^{th} FD equation and the $i-1^{\text{th}}$ equation (5.8) across the interface can be written respectively as

$$
d_i\Psi_i + d_{i-1}C_{i-1} = \Delta x \ g_{i-\frac{1}{2}} \tag{5.13a}
$$

$$
\frac{a_{i-2}C_{i-2} + a_{i-1}C_{i-1} + a_i\Psi_i}{\Delta x^2} = f_{i-1} \tag{5.13b}
$$

$$d_i = -D_{i-\frac{1}{2}} - D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\left(\phi_i - \phi_{i-1}\right)/2 \tag{5.13c}$$
$$d_{i-1} = D_{i-\frac{1}{2}} - D_{i-\frac{1}{2}}\beta_{i-\frac{1}{2}}\left(\phi_i - \phi_{i-1}\right)/2.$$

Case 2. $\mathbf{n} = \langle -1, 0, 0\rangle$, x_i = 'W', x_{i+1} = 'C', and $\gamma = x_{i+\frac{1}{2}}$. Similarly, we have

$$d_i\Psi_i + d_{i+1}C_{i+1} = \Delta x \; g_{i+\frac{1}{2}} \tag{5.14a}$$
$$\frac{a_i\Psi_i + a_{i+1}C_{i+1} + a_{i+2}C_{i+2}}{\Delta x^2} = f_{i+1} \tag{5.14b}$$

$$d_i = -D_{i+\frac{1}{2}} + D_{i+\frac{1}{2}}\beta_{i+\frac{1}{2}}\left(\phi_{i+1} - \phi_i\right)/2 \tag{5.14c}$$
$$d_{i+1} = D_{i+\frac{1}{2}} + D_{i+\frac{1}{2}}\beta_{i+\frac{1}{2}}\left(\phi_{i+1} - \phi_i\right)/2.$$

We next consider the Slotboom form of NP (1.14) with (1.11) and (1.12). In 1D, it reads as

$$-\frac{\partial J}{\partial x} = \frac{\partial}{\partial x}\left(\alpha\frac{\partial \widehat{C}}{\partial x}\right) = f \tag{5.15}$$

and the FD equation at $x = x_i$ is

$$\frac{\alpha_{i-\frac{1}{2}}\widehat{C}_{i-1} - \left(\alpha_{i+\frac{1}{2}} + \alpha_{i-\frac{1}{2}}\right)\widehat{C}_i + \alpha_{i+\frac{1}{2}}\widehat{C}_{i+1}}{\Delta x^2} = f_i. \tag{5.16}$$

Corresponding to (5.10) and (5.12), we have respectively

$$J^x = -\alpha\frac{\partial \widehat{C}}{\partial x} = g \quad \text{at } \gamma \tag{5.17}$$

$$-\alpha_{i-\frac{1}{2}}\frac{\widehat{\Psi}_i - \widehat{C}_{i-1}}{\Delta x} = g_{i-\frac{1}{2}}, \tag{5.18}$$

Eq. (5.17) is a Neumann BC. If a Dirichlet BC is considered, we then have

$$\widehat{C} = \widehat{g}_D \quad \text{at } \gamma \Longrightarrow \widehat{\Psi}_i = \widehat{g}_{Di} \text{ or} \tag{5.19}$$
$$C = g_D,\ \Psi_i = g_{Di} \text{ (primitive).}$$

The method (5.12) (or (5.18)) alone to treat the Robin (or Neumann) BC is usually unstable due to many undefined normal vectors $\mathbf{n}$ at corner points. To stabilize the method, we make connections between the adjacent points of 'W's and 'F's. For this, in addition to (5.12), we impose

$$-\frac{\Psi_i + \Psi_{i-1}}{2} = -C_{i-\frac{1}{2}} \tag{5.20a}$$

$$-\Psi_i + \Psi_{i-1} = 0, \text{ if } C_{i-\frac{1}{2}} \text{ is not given.} \tag{5.20b}$$

All Robin (with stabilization for the primitive form), Neumann (with stabilization for the Slotboom form), and Dirichlet BCs are implemented for both GA and cylinder. On the interface Γ, we should use either Robin or Neumann BCs. Dirichlet BCs are used only for testing the code. All numerical results are good as shown in the following tables.

Example 5.1. Primitive, GA, $\epsilon_m = 1$, $\epsilon_s = 80$, $[\phi] = 0$, $[\epsilon\phi_{\mathbf{n}}] \neq 0$. Numerical results for the Poisson problem are shown in Table 5.1C-P with good $O(h^2)$ convergence where numerical results of the same exact solution (2.1) used in [21] for the GA channel are also presented for comparison. Note that the MIB method of Wei et al. [21] requires more than 27 FD grid points whereas ours requires only 7 under the assumption (4.4). The method with (5.8), (5.12), and (5.19) in the primitive form gave perfect results as shown in Table 5.1 for all PNP problems.

Table 5.1C-P. Ours vs Wei's MIB				
	Ours (7-pt)		Wei's (> 27-pt)	
h in Å	E_∞	Order	E_∞	Order
2	0.4466			
1	0.0922	2.28	0.1400	
0.5	0.0228	2.02	0.0271	2.36
0.25	0.0057	2.00	0.0152	0.84

Table 5.1. Primitive, GA, Linear								
	Dirichlet				Robin			
h in Å	P	NP1	NP2	Time	P	NP1	NP2	Time
2	0.4466	1.0203	1.1903		0.4466	1.0302	1.4471	
1	0.0922	0.0457	0.0360		0.0922	0.0451	0.0434	
0.5	0.0228	0.0103	0.0072	1m14s	0.0228	0.0103	0.0081	1m14s
0.25	0.0057	0.0025	0.0017	10m28s	0.0057	0.0025	0.0018	10m31s

Example 5.2. Slotboom, GA, $\epsilon_m = 1$, $\epsilon_s = 80$, $[\phi] = 0$, $[\epsilon\phi_{\mathbf{n}}] \neq 0$. The method with (5.16), (5.18), and (5.19) in the Slotboom form also gave good results as shown in Table 5.2 for all PNP problems. Note that the CPU time

is 10m24s with the SOR relaxation parameter $\omega = 1.9$ whereas it took 30m42s (not shown) with $\omega = 1.2$ in the subroutine SOR_3DCA().

Table 5.2. Slotboom, GA, Linear								
	Dirichlet				Neumann			
h in Å	P	NP1	NP2	Time	P	NP1	NP2	Time
2	0.4466	0.8265	0.2276		0.4466	0.7420	0.2715	
1	0.0922	0.0841	0.0364		0.0922	0.0812	0.0387	
0.5	0.0228	0.0187	0.0077	1m14s	0.0228	0.0195	0.0095	1m14s
0.25	0.0057	0.0045	0.0018	10m24s	0.0057	0.0047	0.0024	10m34s

Example 5.3. Slotboom, Cylinder, $\epsilon_m = 1$, $\epsilon_s = 80$, $[\phi] = 0$, $[\epsilon\phi_{\mathbf{n}}] \neq 0$.

Table 5.3. Slotboom, Cylinder, Linear						
	Dirichlet		Neumann			
h in Å	P	NP1	P	NP1	NP2	Time
2	0.4442	0.6488	0.4442	0.7414	0.2545	
1	0.0925	0.0847	0.0925	0.0831	0.0382	
0.5	0.0229	0.0188	0.0229	0.0189	0.0086	
0.25	0.0057	0.0046	0.0057	0.0046	0.0020	6m25s

As mentioned in [21], there is another way to implement the flux on Γ, namely, the Boundary Condition II in [21],

$$\mathbf{J} = \mathbf{g} \begin{cases} = \mathbf{0} \text{ real PNP}, \\ \neq \mathbf{0} \text{ with exact solutions}, \end{cases} \quad \text{on} \quad \Gamma. \tag{5.21}$$

References

[1] A. Aksimentiev, M. Sotomayor, and D. Wells, Membrane proteins tutorial, Theoretical and Computational Biophysics Group, University of Illinois at Urbana-Champaign (2009).

[2] I. Borukhov, D. Andelman, and H. Orland, Steric effects in electrolytes: A modified Poisson-Boltzmann equation, Phys. Rev. Lett. 79 (1997) 435–438.

[3] D. P. Chen, V. Barcilon, and R. S. Eisenberg, Constant fields and constant gradients in open ionic channels, Biophys. J. 61 (1992) 1372–1393.

[4] D. Chen, and G.-W. Wei, Quantum dynamicsin continuum for ion channel transport, preprint (2010).

[5] R.-C. Chen and J.-L. Liu, An accelerated monotone iterative method for the quantum-corrected energy transport model, J. Comp. Phys., 227 (2008) pp. 6266-6240.

[6] I-L. Chern, J.-G. Liu, and W.-C. Wang, Accurate evaluation of electrostatics for macromolecules in solution, Methods Appl. Anal. 10 (2003) 309–328.

[7] R. S. Eisenberg, Multiple scales in the simulation of ion channels and proteins, J. Phys. Chem. C 114 (2010) 20719–20733.

[8] W. Geng, S. Yu, and G. Weia, Treatment of charge singularities in implicit solvent models, J. Chem. Phys. 127 (2007) 114106.

[9] B. Hille, Ionic Channels of Excitable Membranes, 3rd Ed., Sinauer Associates Inc., Sunderland, MA, 2001.

[10] W. Humphrey, A. Dalke, and K. Schulten, VMD - Visual Molecular Dynamics, J. Molec. Graphics, 14 (1996) 33-38.

[11] M. G. Kurnikova, R. D. Coalson, P. Graf, and A. Nitzan, A lattice relaxation algorithm for three-dimensional Poisson-Nernst-Planck theory with application to ion transport through the Gramicidin A channel, Biophys. J. 76 (1999) 642–656.

[12] R. J. LeVeque, Finite Difference Methods for Differential Equations, Lecture Notes, University of Washington, 2005.

[13] J.-L. Liu, Lecture Notes on Numerical Methods for Partial Differential Equations, 2011.

[14] B. Lu and J. A. McCammon, Molecular surface-free continuum model for electrodiffusion processes, Chem. Phys. Lett. 451 (2008) 282–286.

[15] B. Lu and Y. C. Zhou, Poisson-Nernst-Planck Equations for simulating biomolecular diffusion-reaction processes II: Size effects on ionic distributions and diffusion-reaction rates, Preprint (2011).

[16] P. Mark and L. Nilsson, Structure and dynamics of the TIP3P, SPC, and SPC/E water models at 298 K, J. Phys. Chem. A 105 (2001) 9954-9960.

[17] Popular Information, Nobelprize.org. 18 Jan 2011, http://nobelprize.org/nobel_prizes/chemistry/laureates/2003/public.html.

[18] N. A. Simakov and M. G. Kurnikova, Soft wall ion channel in continuum representation with application to modeling ion currents in α-Hemolysin, J. Phys. Chem. B 114 (2010) 15180–15190.

[19] J. W. Slotboom, Computer-aided two-dimensional analysis of bipolar transistors, IEEE Trans. Elec. Dev. ED-20 (1973) 669-679.

[20] J. A. Te and T. Ichiye, Temperature and pressure dependence of the optimized soft-sticky dipole-quadrupole-octupole water model, J. Chem. Phys. 132 (2010) 114511.

[21] Q. Zheng, D. Chen, and G.-W. Wei, Second-order Poisson Nernst-Planck solver for ion channel transport, J. Comp. Phys. 230 (2011) 5239-5262.