ACCURATE EVALUATION OF ELECTROSTATICS FOR MACROMOLECULES IN SOLUTION [*]

I-LIANG CHERN[†], JIAN-GUO LIU[‡], AND WEI-CHENG WANG[§]

Abstract. Most biochemical processes involve macromolecules in solution. The corresponding electrostatics is of central importance for understanding their structures and functions. An accurate and efficient numerical scheme is introduced to evaluate the corresponding electrostatic potential and force by solving the governing Poisson-Boltzmann equation. This paper focuses on the following issues: (i) the point charge singularity problem, (ii) the dielectric discontinuity problem across a molecular surface, and (iii) the infinite domain problem. Green's function associated with the point charges plus a harmonic function is introduced as the zeroth order approximation to the solution to solve the point charge singularity problem. A jump condition capturing finite difference scheme is adopted to solve the discontinuity problem across molecule surfaces, where a body-fitting grid is used. The infinite domain problem is solved by mapping the outer infinite domain into a finite domain. The corresponding stiffness matrix is symmetric and positive definite, therefore, fast algorithm such as preconditioned conjugate gradient method can be applied for inner iteration. Finally, the resulting scheme is second order accurate for both the potential and its gradient.

1. Introduction. Most biochemical processes involve macromolecules such as protein or nucleotide in mobile ionic water solvent. The corresponding electrostatics is of central importance for understanding their structures and functions [5, 8, 28].

A common and practical model is to represent the macromolecule as a structured and polarized clusters of atoms, and to treat the ionic solvent as a continuum. Such a continuum model was pioneered by Debye and Hückel in 1924 [9, 19]. The coordinates $\{\boldsymbol{x}_i | i = 1, \cdots, N_m\}$ of these atoms can be obtained from X-ray crystallography pictures, and are available from the protein data bank at Brookhaven National Laboratory. As the macromolecule is immersed in water solvent, a "molecular surface" is formed, which separates the bimolecular region from the water region. In the water region, there is a thin ion exclusion layer around the molecular surface, and ions are mobile outside this layer. The construction of this molecular surface and the ion exclusion layer is usually done by moving a probe around the macromolecule. More precisely, the water region is defined to be the union of all spheres whose centers $\boldsymbol{y}$ satisfy $|\boldsymbol{y} - \boldsymbol{x}_i| > r_i + \sigma$ for all macromolecular atoms $\boldsymbol{x}_i$, where r_i and σ are respectively the radii of the ith atom and the probe. The mobile ion region is defined to be the exterior of the union of spheres $B(\boldsymbol{x}_i, r_i + \sigma_{ion})$ [20, 32]. The ion densities in the water solvent are assumed to obey the Boltzmann distribution law [9].

To write down the governing equation for the electrostatic potential for such a system, we follow the notation in Holst's paper [12]. Let us denote Ω_1 the macromolecule region, Γ_1 the molecular surface, Ω_2 the ion exclusion layer, Ω_3, the ionic solvent region, and Γ_2 the interface which separates Ω_2 and Ω_3. There could be several kinds of ions. For easy presentation of our numerical model, we assume there are only

two kinds of ions with opposite sign and same charge unit, and with total neutralized solution. The corresponding electrostatic potential Φ satisfies the Poisson-Boltzmann equation (PBE) [9, 27, 28]:

$$-\nabla \cdot [\epsilon(\boldsymbol{x})\nabla\Phi(\boldsymbol{x})] + K(\boldsymbol{x})\left(\frac{k_B T}{e_c}\right)\sinh\left(\frac{e_c\Phi(\boldsymbol{x})}{k_B T}\right) = 4\pi\sum_{i=1}^{N_m} q_i\delta(\boldsymbol{x} - \boldsymbol{x}_i). \quad (1)$$

$$\Phi(\infty) = 0. \quad (2)$$

Here, e_c the charge of an electron, k_B the Boltzmann constant, T the temperature, $\boldsymbol{x}_i$ the atomic location, q_i the atomic partial charge, the function $\epsilon(\boldsymbol{x})$ is the dielectric parameter,

$$\epsilon(\boldsymbol{x}) = \begin{cases} \epsilon^-, & \boldsymbol{x} \in \Omega_1, \\ \epsilon^+, & \boldsymbol{x} \in \Omega_2 \cup \Omega_3. \end{cases}$$

$$K(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in \Omega_1 \cup \Omega_2, \\ \overline{\kappa}^2, & \boldsymbol{x} \in \Omega_3, \end{cases}$$

where $\overline{\kappa} = \sqrt{\epsilon^+}\kappa$ is the modified Debye-Hückel parameter and κ is the Debye-Hückel parameter.

By introducing the dimensionless potential $\phi = \frac{e_c\Phi}{k_B T}$, we can rewrite the Poisson-Boltzmann equation as

$$-\nabla \cdot [\epsilon(\boldsymbol{x})\nabla\phi(\boldsymbol{x})] + K(\boldsymbol{x})\sinh(\phi(\boldsymbol{x})) = Q(\boldsymbol{x}) \quad (3)$$

where

$$Q(\boldsymbol{x}) = C\sum_{i=1}^{N_m} z_i\delta(\boldsymbol{x} - \boldsymbol{x}_i). \quad (4)$$

Typical molecules such as enzymes and proteins have length scale 10 to over 100 angstroms [12]. Typical numerical values of these parameters are [12]

$$\epsilon^- \approx 2, \epsilon^+ \approx 80,$$

$$5249.0 \leq C \leq 10500.0,$$

$$-1 \leq z_i \leq 1, \quad (5)$$

$$K^+ = \overline{\kappa}^2 = 8.486902807 \mathring{A}^{-2} I_s,$$

$$I_s \text{ (ionic strength) } \in [0, 10]$$

Previous numerical methods for PBE can be classified into finite difference methods [1, 10, 18, 23, 17, 26, 29, 12], finite element methods [6, 7], and adaptive finite element methods [13, 3]. To speed up the interior iteration in solving a corresponding linear system, multigrid [14] and multi-level methods [12] have been proposed. The nonlinear counterpart was solved successfully by the damped inexact-Newton's method [12].

Despite decades of effort on improvement of modeling by various groups of people mentioned above, accurate and efficient modeling for typical macromolecules is still very challenging for the following reasons.

1. Typical macromolecules contain hundreds to thousands of singular charges.
2. Usually, the molecular surface Γ_1 is quite irregular due to complicated macro-molecular folding; moreover, across which the dielectric function is discontinuous.
3. This is an infinite domain problem.

This paper focuses on the corresponding issues, namely, (i) the point charge singularity problem, (ii) the surface singularity problem, and (iii) the infinite domain problem.

Most finite difference methods mentioned above treated these singularity problems by smearing. The sources are then less singular and the coefficients become smoother. In finite element approaches, the point charge singularities are handled by taking averages with the test polynomials over mesh zones, and the surface discontinuities are taken care of by using interface-aligned meshes. These approaches are hard to achieve second order accuracy for both the potential and *the electric field*.

For the infinite domain problem, standard approach is to use a truncated finite domain which contains the macromolecule region, and its boundary is far from the molecule surface. A Dirichlet boundary condition is then applied. The Dirichlet data for the potential is taken to be either 0 or the solution of a linearized Poisson-Boltzmann equation. Such an approach requires a larger computation domain when the Debye parameter κ is small.

Here, we take a different approach. First, Green's function corresponding to the point charges plus a harmonic function in the molecule region are introduced as the zeroth order approximation to the solution. This approach takes care of the point charge singularity problem. It certainly also creates an extra surface discontinuity across the molecular surface. This problem together with the original surface discontinuity problem will be solved by using the jump condition capturing scheme [31], a finite difference scheme developed by one of the authors to handle surface singularities. The scheme uses a body-fitting grid and a skewed variable. The jump condition across the molecular surface is built in the finite difference scheme in the skewed variable naturally. The scheme is simple, and no interpolation is needed as that in the immersed interface method [21]. The resulting stiffness matrix is symmetric and positive definite. Hence, standard preconditioned conjugate gradient method can be applied to gain efficiency. More importantly, the scheme is second order for both the potential and its gradient. The second order property for the gradients is probably due to the central differencing and uniformity of the grids in the computational domain.

To treat the infinite domain problem, a grid patching technique is applied. The whole domain is divided into a ball (containing the molecule region) and its exterior. The grid in the exterior region is obtained through a mirror map. The grid inside the ball is an extension of the above body-fitting grid. The connection of these two grids is smooth only up to first order. The jump condition capturing scheme is applied again to solve the discontinuous curvature problem across the grid interface.

2. Treatments of singularities.

2.1. Treatment of point charge singularities.
The solution of the Poisson-Boltzmann equation contains singular part due to the presence of the singular charge distribution $C \sum z_i \delta(\boldsymbol{x} - \boldsymbol{x}_i)$ inside Γ_1. In order to compute the solution correctly without smearing, we decompose the solution into regular and singular parts

$$\phi = \overline{\phi} + \tilde{\phi}, \tag{6}$$

where

$$\overline{\phi}(\boldsymbol{x}) = \begin{cases} \phi^*(\boldsymbol{x}) + \phi^0(\boldsymbol{x}) & \boldsymbol{x} \in \Omega_1 \\ 0 & \boldsymbol{x} \in \Omega_2 \cup \Omega_3 \end{cases}. \tag{7}$$

and ϕ^* is the potential in the free space induced by Q, i.e.

$$\phi^*(\boldsymbol{x}) = \begin{cases} C \sum_{i=1}^{m} \dfrac{1}{\epsilon_1} \dfrac{z_i}{4\pi} \dfrac{1}{|\boldsymbol{x} - \boldsymbol{x}_i|}, & \boldsymbol{x} \in R^3 \\ C \sum_{i=1}^{m} -\dfrac{1}{\epsilon_1} \dfrac{z_i}{2\pi} \log(|\boldsymbol{x} - \boldsymbol{x}_i|), & \boldsymbol{x} \in R^2 \end{cases}. \tag{8}$$

ϕ^0 is a harmonic function in Ω_1 satisfying

$$\begin{cases} \triangle \phi^0 = 0 & \text{in } \Omega_1 \\ \phi^0 = -\phi^* & \text{on } \Gamma_1. \end{cases} \tag{9}$$

The introduction of ϕ^0 is to force $[\overline{\phi}] = 0$ across Γ_1. This can avoid an over-floating problem in evaluating $\sinh(\phi)$ during nonlinear iteration when C is large (5). It is clear that $\overline{\phi}$ is a solution to (1) in $\Omega_1 \cup \Omega_2 \cup \Omega_3$ and satisfies the following jump conditions

$$\begin{aligned} &[\overline{\phi}]_{\Gamma_1} = 0, \\ &[\epsilon\overline{\phi}_n]_{\Gamma_1} = -\epsilon^- \nabla(\phi^* + \phi^0) \cdot \boldsymbol{n}|_{\Gamma_1}, \\ &[\overline{\phi}]_{\Gamma_2} = 0, \\ &[\epsilon\overline{\phi}_n]_{\Gamma_2} = 0. \end{aligned} \tag{10}$$

The equation for the correction potential $\tilde{\phi} = \phi - \overline{\phi}$ is therefore

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}(\boldsymbol{x})\right) + K(\boldsymbol{x})\sinh(\overline{\phi}(\boldsymbol{x}) + \tilde{\phi}(\boldsymbol{x})) = 0, \ \boldsymbol{x} \in \Omega_1 \cup \Omega_2 \cup \Omega_3, \tag{11}$$

$$\begin{aligned} &[\tilde{\phi}]_{\Gamma_1} = 0, \\ &[\epsilon\tilde{\phi}_n]_{\Gamma_1} = -[\epsilon\overline{\phi}_n]_{\Gamma_1} \\ &[\tilde{\phi}]_{\Gamma_2} = 0, \\ &[\epsilon\tilde{\phi}_n]_{\Gamma_2} = 0. \end{aligned} \tag{12}$$

or equivalently

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}(\boldsymbol{x})\right) + K(\boldsymbol{x})\sinh(\overline{\phi}(\boldsymbol{x}) + \tilde{\phi}(\boldsymbol{x})) = [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1}. \tag{13}$$

Since $K(\boldsymbol{x}) = 0$ on Ω_1 and $\overline{\phi}(\boldsymbol{x}) = 0$ on $\Omega_2 \cup \Omega_3$, (13) can be further simplified as

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}(\boldsymbol{x})\right) + K(\boldsymbol{x})\sinh(\tilde{\phi}(\boldsymbol{x})) = [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1}. \tag{14}$$

Note that $K(\boldsymbol{x}) = 0$ in Ω_1, the correction potential $\tilde{\phi}(\boldsymbol{x})$ is harmonic and therefore smooth in Ω_1. Thus the decomposition $\phi = \overline{\phi} + \tilde{\phi}$ has indeed separated the singular part from the solution.

In this formulation, we have effectively replaced the point charges singularity $Q(\boldsymbol{x})$ by the much milder surface charges singularity $[\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1}$ which can be handled accurately and efficiently by the jump condition capturing finite difference scheme described in section 2.3.

2.2. Nonlinear iteration for the correction potential. The nonlinear correction equation (14) can be solved by standard iterative methods such as monotone iteration and Newton's iteration. The monotone iteration for (14) is given by

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}^{l+1}(\boldsymbol{x})\right) + \lambda\tilde{\phi}^{l+1}(\boldsymbol{x}) = \lambda\tilde{\phi}^l(\boldsymbol{x}) - K(\boldsymbol{x})\sinh(\tilde{\phi}^l(\boldsymbol{x})) + [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1}. \quad (15)$$

and converges when $\lambda > 0$ is large enough and the right-hand side becomes monotonic in $\tilde{\phi}$

$$\lambda - K(\boldsymbol{x})\cosh(\tilde{\phi}(\boldsymbol{x})) \geq 0.$$

It is worth noting that the monotone iteration (15) can be interpreted as numerical integration of the gradient flow

$$\tilde{\phi}_t = -\frac{\delta E(\tilde{\phi})}{\delta\tilde{\phi}} = \nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}(\boldsymbol{x})\right) - K(\boldsymbol{x})\sinh(\tilde{\phi}(\boldsymbol{x})) + [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1}, \quad (16)$$

where $E(\tilde{\phi})$ is the convex energy functional

$$E(\tilde{\phi}) = \int_{R^n} \epsilon(\boldsymbol{x})(\nabla\tilde{\phi}(\boldsymbol{x}))^2 + K(\boldsymbol{x})\cosh(\tilde{\phi}(\boldsymbol{x})) - \int_{\Gamma_1} [\epsilon\overline{\phi}_n]\tilde{\phi}.$$

When we integrate the gradient flow (16) in t using explicit treatment for the nonlinear term and implicit treatment of the diffusion term, we get

$$\frac{\tilde{\phi}^{l+1} - \tilde{\phi}^l}{\Delta t} = \nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}^{l+1}(\boldsymbol{x})\right) - K(\boldsymbol{x})\sinh(\tilde{\phi}^l(\boldsymbol{x})) + [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1},$$

which is equivalent to (15) and converges for Δt sufficiently small.

The monotone iteration (15) only converges at a linear rate. For better performance, we can switch to Newton's iteration to accelerate convergence. The standard Newton's iteration for (14) is given by

$$\phi^l = \overline{\phi} + \tilde{\phi}^l \quad (17)$$

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla v^l\right) + K(\boldsymbol{x})\cosh(\phi^l)v^l = \nabla \cdot \left(\epsilon(\boldsymbol{x})\nabla\tilde{\phi}^l\right) - K(\boldsymbol{x})\sinh(\phi^l) + [\epsilon\overline{\phi}_n]_{\Gamma_1}\delta_{\Gamma_1} \quad (18)$$

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + v^l \quad (19)$$

A well known trick to control global convergence of Newton's method is through damping [2]. Since the direction v^n is indeed a descent direction for the functional $E(\phi)$,

$$E(\phi^l + \lambda^l v^l) < E(\phi^l) \text{ for small } \lambda^l > 0,$$

we can accelerate the convergence of the Newton's method globally by performing a line search to find a suitable damping parameter λ^l that minimizes $E(\phi^l + \lambda^l v^l)$ and replace (19) by

$$\tilde{\phi}^{l+1} = \tilde{\phi}^l + \lambda^l v^l. \tag{20}$$

An efficient search scheme for λ^l can be found in [2, 12]. When the linear system (18) is solved only approximately, a more sophisticated damped inexact Newton's method was also proposed to insure the approximate solution v^l is a descent direction [2, 12].

2.3. Treatment of surface singularities — the jump condition capturing scheme. Both the monotone iteration (15) and the Newton iteration (18) are based on solving the following linearized Poisson-Boltzmann equation

$$-\nabla \cdot \left(\epsilon(\boldsymbol{x}) \nabla \tilde{\phi}(\boldsymbol{x}) \right) + H(\boldsymbol{x}) \tilde{\phi}(\boldsymbol{x}) = f(\boldsymbol{x}), \ \boldsymbol{x} \in \Omega_1 \cup \Omega_2 \cup \Omega_3$$

$$[\tilde{\phi}] = g(\boldsymbol{x}), \ [\epsilon \tilde{\phi}_n] = h(\boldsymbol{x}), \ \boldsymbol{x} \in \Gamma_1. \tag{21}$$

$$[\tilde{\phi}] = 0, \ [\tilde{\phi}_n] = 0, \ \boldsymbol{x} \in \Gamma_2.$$

where $\epsilon(\boldsymbol{x})$ is discontinuous across Γ_1, and $H(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are discontinuous across Γ_2. The singularity across Γ_2 is not essential because the corresponding $\tilde{\phi}$ is C^1 across Γ_2. For easy presentation on the treatment of surface singularity, we lump the treatments of two singularities together and simply assume $\Gamma_1 = \Gamma_2$. Such a problem is known as the interface problem. In many practical application, we are also interested in computing the electrostatic field $\nabla \phi$. In this situation, the accuracy of the numerical solution is even more demanding. Some of the existing schemes which are first order accurate in ϕ are not adequate for this purpose. A standard finite element method with meshes aligned with interface can achieve second order for potential but only first order for its gradients. Finite difference approaches such as immersed boundary method [24], or immersed interface method [21] are also at most first order accurate for the gradients.

In our simulations, we have adopted the jump condition capturing finite difference scheme [31]. The resulting matrix is symmetric and positive definite and easy to invert. In addition, both the computed potential ϕ and electrostatic field $\nabla_h \phi|_{\Gamma_1}$ are observed to have second order accuracy in all our simulations. For readers' convenience, we briefly describe the jump condition capturing scheme in two space dimension here. The 3D scheme can be derived similarly [31].

2.3.1. Body-fitting grid and a skewed variable. Let us denote by (ξ^1, ξ^2) the variables in the computational domain, $\boldsymbol{X}(\xi^1, \xi^2) \in R^2$ the position vector in the physical space. We choose our coordinate system in such a way that one of the coordinate line $\xi^1 = $ constant is mapped to the interface Γ_1 which corresponds to $i = i_0$. We do not require the coordinate system to be orthogonal near the interface. Therefore such a coordinate system can be easily generated by smoothly deforming the interface from a chosen origin to a prescribed outer boundary.

We shall not discretize (21) in ξ variable. Instead, we shall discretize (21) in η variable, a new coordinate system in the computational domain in the skewed direction that aligned with the grid points generated by the ξ variables. More precisely, we define

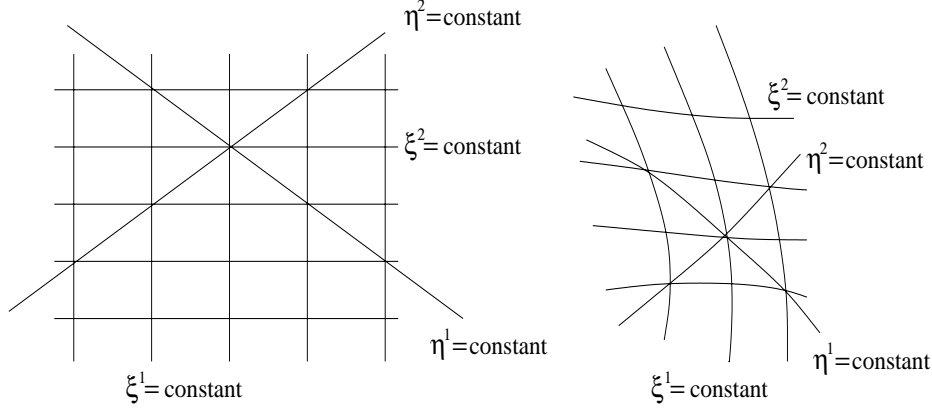$$\eta^1 = \frac{\xi^1 \Delta \xi^2 + \xi^2 \Delta \xi^1}{\sqrt{(\Delta \xi^1)^2 + (\Delta \xi^2)^2}} \tag{22}$$

$$\eta^2 = \frac{\xi^2 \Delta\xi^1 - \xi^1 \Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}} \tag{23}$$

$$\Delta\eta^1 = \Delta\eta^2 = \frac{2\Delta\xi^1\Delta\xi^2}{\sqrt{(\Delta\xi^1)^2 + (\Delta\xi^2)^2}} \tag{24}$$

See also Fig 1.

2.3.2. Finite difference discretization in the skewed variable. The linearized PB equation (21) in the η variables reads

$$-\partial_\mu(\epsilon\sqrt{\hat{g}}\hat{g}^{\mu\nu}\partial_\nu\tilde{\phi}) + \sqrt{\hat{g}}H(x)\tilde{\phi} = \sqrt{\hat{g}}f \tag{25}$$

where we have adopted the summation convention and

$$\hat{g}^{\mu\nu} = \langle \nabla\eta^\mu, \nabla\eta^\nu \rangle \tag{26}$$

$$\hat{g}_{\mu\nu} = \langle \frac{\partial \boldsymbol{X}}{\partial\eta^\mu}, \frac{\partial \boldsymbol{X}}{\partial\eta^\nu} \rangle \tag{27}$$

$$\hat{g} = \det(\hat{g}_{\mu\nu}) \tag{28}$$

$$\sqrt{\hat{g}} = \det(\frac{\partial \boldsymbol{X}}{\partial\boldsymbol{\eta}}) = \frac{1}{2}\left(\frac{\Delta\xi^1}{\Delta\xi^2} + \frac{\Delta\xi^2}{\Delta\xi^1}\right)\det(\frac{\partial \boldsymbol{X}}{\partial\boldsymbol{\xi}}) \tag{29}$$

$$\hat{g}^{\mu\gamma}\hat{g}_{\gamma\nu} = \delta^\mu_\nu \tag{30}$$

A crucial observation in [31] is that the jump conditions can be incorporated into

the following finite difference discretization

$$
\begin{aligned}
- \quad & \frac{1}{(\Delta\eta)^2}\left((\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{11})_{i+\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i+1,j+1}-\tilde{\phi}_{i,j}) - (\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{11})_{i-\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i,j}-\tilde{\phi}_{i-1,j-1})\right) \\
- \quad & \frac{1}{(\Delta\eta)^2}\left((\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{12})_{i+\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i,j+1}-\tilde{\phi}_{i+1,j}) - (\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{12})_{i-\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i-1,j}-\tilde{\phi}_{i,j-1})\right) \\
- \quad & \frac{1}{(\Delta\eta)^2}\left((\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{21})_{i-\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i,j+1}-\tilde{\phi}_{i-1,j}) - (\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{21})_{i+\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i+1,j}-\tilde{\phi}_{i,j-1})\right) \\
- \quad & \frac{1}{(\Delta\eta)^2}\left((\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{22})_{i-\frac{1}{2},j+\frac{1}{2}}(\tilde{\phi}_{i-1,j+1}-\tilde{\phi}_{i,j}) - (\epsilon\sqrt{\hat{g}_h}\hat{g}_h^{22})_{i+\frac{1}{2},j-\frac{1}{2}}(\tilde{\phi}_{i,j}-\tilde{\phi}_{i+1,j-1})\right) \\
= \quad & \sqrt{(\hat{g}_h)}_{i,j}(f_{i,j}-H_{i,j}\tilde{\phi}_{i,j})
\end{aligned}
\tag{31}
$$

The left-hand side of (31) is uniformly valid even in the presence of the interface and jump discontinuities of the coefficients $\epsilon(\boldsymbol{x})$ and $K(\boldsymbol{x})$. At the interface $i = i_0$, the right-hand side of (31) is replaced by

$$
\sqrt{(\hat{g}_h)}_{i,j}\left(\frac{1}{2}(f_{i^+,j}+f_{i^-,j}) - \frac{1}{2}(H_{i^+,j}\tilde{\phi}_{i^+,j}+H_{i^-,j}\tilde{\phi}_{i^-,j})\right)
$$
$$
+ \text{ terms involving } [\tilde{\phi}] \text{ and } [\epsilon\tilde{\phi}_n]
$$

In (31), $\hat{g}_h^{\mu\nu}$, the 2nd order approximation of the metric tensor $\hat{g}^{\mu\nu}$, can be conveniently computed from the $\boldsymbol{X}_{i,j}$'s as follows

$$
\begin{aligned}
((\hat{g}_h)_{11})_{i+\frac{1}{2},j+\frac{1}{2}} &= \langle D_1\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}, D_1\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}\rangle \\
&= \langle \frac{\boldsymbol{X}_{i+1,j+1}-\boldsymbol{X}_{i,j}}{\Delta\eta}, \frac{\boldsymbol{X}_{i+1,j+1}-\boldsymbol{X}_{i,j}}{\Delta\eta}\rangle
\end{aligned}
$$

$$
\begin{aligned}
((\hat{g}_h)_{22})_{i+\frac{1}{2},j+\frac{1}{2}} &= \langle D_2\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}, D_2\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}\rangle \\
&= \langle \frac{\boldsymbol{X}_{i,j+1}-\boldsymbol{X}_{i+1,j}}{\Delta\eta}, \frac{\boldsymbol{X}_{i,j+1}-\boldsymbol{X}_{i+1,j}}{\Delta\eta}\rangle
\end{aligned}
$$

and

$$
((\hat{g}_h)_{12})_{i+\frac{1}{2},j+\frac{1}{2}} = ((\hat{g}_h)_{21})_{i+\frac{1}{2},j+\frac{1}{2}} = \langle D_1\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}, D_2\boldsymbol{X}_{i+\frac{1}{2},j+\frac{1}{2}}\rangle
$$

As long as the grids are non-crossing, that is $(\boldsymbol{X}_{i,j}, \boldsymbol{X}_{i+1,j}, \boldsymbol{X}_{i+1,j+1}, \boldsymbol{X}_{i,j+1})$ is positively oriented throughout for all (i,j)'s, then $\sqrt{\hat{g}_h} > 0$ and the $\hat{g}_h^{\mu\nu}$'s can be easily obtained from

$$
\hat{g}_h^{\mu\lambda}(\hat{g}_h)_{\lambda\nu} = \delta_\nu^\mu
\tag{32}
$$

The local truncation error is $O(\Delta\eta)$ on the interface and $O(\Delta\eta^2)$ elsewhere. From the numerical experiments we have conducted, it is observed that the resulting scheme is second order accurate. A remarkable feature about this scheme is that the resulting flux function is 2nd order accurate, even on the interface.

Another crucial point to use the skewed directions η^ν for discretization is that all 4 entries of the metric tensor $\hat{g}^{\mu\nu}$ are evaluated at the *same* location, namely $\boldsymbol{X}(\xi^1_{i+\frac{1}{2},j+\frac{1}{2}}, \xi^2_{i+\frac{1}{2},j+\frac{1}{2}})$. It follows from summation by parts (assuming Dirichlet boundary condition at the outermost boundary for simplicity) that,

$$
\begin{aligned}
-\langle\langle\tilde{\phi}, \nabla_h \cdot (\epsilon\nabla_h\tilde{\phi})\rangle\rangle &\stackrel{\text{def}}{=} -\sum_{i,j}\sqrt{\hat{g}}_{i,j}\tilde{\phi}_{i,j}\nabla_h \cdot (\epsilon\nabla_h\tilde{\phi})_{i,j} \\
&= \sum_{i,j}(\epsilon\sqrt{\hat{g}}\hat{g}^{\mu\nu}D_\mu\tilde{\phi}D_\nu\tilde{\phi})_{i+\frac{1}{2},j+\frac{1}{2}} \geq 0
\end{aligned}
$$

as long as each $(\hat{g}^{\mu\nu}_{i+\frac{1}{2},j+\frac{1}{2}})$ is a positive definite 2 by 2 matrix. In other words, the symmetry and positivity of this discretization is essentially unconditional.

2.4. Infinite domain problem and coordinate patching. In this subsection, we explain the idea of coordinate patching to handle the infinite domain problem. For simplicity, we assume that Γ_1 is contained in the unit disk $\{r \leq 1, 0 \leq \theta < 2\pi\}$, well separated from the boundary $r = 1$ and star-shaped with respect to the origin. More general domain can be handled by the same idea. For details, see [16].

Our computational domain will be the disk: $\{0 \leq \rho \leq 2, \ 0 \leq \theta < 2\pi\}$. We construct a map: $(\rho, \theta) \to (r(\rho, \theta), \theta)$ with the following properties:

1. $\rho = 1/2$ is mapped to the interface Γ_1, $\rho = 1 - \delta$ is mapped to $r = 1 - \delta$ for some small δ, $0 < \delta < 1/2$. This can be done by a suitable scaling because we have assumed Γ_1 is star shape.
2. The function $r(\rho, \theta)$ is defined by

$$r(\rho, \theta) = \begin{cases} \rho & 1 - \delta < \rho \leq 1 \\ 1/(2-\rho) & 1 \leq \rho < 2 \end{cases} \tag{33}$$

3. The function $r(\rho, \theta)$ is smooth everywhere except on $\rho = 1$.

The only non-smooth part of $r(\rho, \theta)$ is on $\rho = 1$, where its second derivative has jumps. In other words, we patch two coordinates together on $\{r = 1\}$.

In region $\{0 < \rho < 1 - \delta\}$, the jump condition capturing scheme is applied. In region $\{1 - \delta < \rho < 1\}$, the Laplacian is discretized by a standard five point formula. Smooth transition from the jump condition capturing scheme to the standard five point formula can be achieved by a symmetry preserving averaging as discussed in [31]. The treatment of pole (i.e. the origin) is also discussed there [31].

Now, we focus on the treatment across the coordinate patching interface $\{r = 1\}$. Roughly speaking, the patching results in discontinuity in the new diffusion coefficient $\epsilon\sqrt{\hat{g}}\hat{g}^{\mu\nu}$ since $\hat{g}^{\mu\nu}$ are discontinuous across coordinate charts. Therefore, it can be handled by the jump condition capturing scheme in the same way.

As in the treatment of the interface problem, we place the grids $i = i_M$, $j = 1, \cdots, n_\theta$ on $r = 1$. Notice that both $H(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are smooth on the outer region $r \geq 1 - \delta$. By standard elliptic regularity, $\tilde{\phi}$ is smooth there as well. Now consider the standard five point formula expressed in terms of (ρ, θ) on the outer region:

$$-\epsilon_2 \frac{\rho'(r)}{r} D_\rho(r\rho'(r)D_\rho\tilde{\phi}) + \frac{1}{r^2}D_\theta^2\tilde{\phi} + H(\boldsymbol{x})\tilde{\phi} = f(\boldsymbol{x}) \tag{34}$$

or equivalently

$$-\epsilon_2 D_\rho(r\rho'(r)D_\rho\tilde{\phi}) + \frac{1}{r\rho'(r)}D_\theta^2\tilde{\phi} + \frac{r}{\rho'(r)}H(\boldsymbol{x})\tilde{\phi} = \frac{r}{\rho'(r)}f(\boldsymbol{x}) \tag{35}$$

The discretization (35) is symmetric and has local truncation error $O(\Delta\rho^2)$ both on $\rho < 1$ and $\rho > 1$.

As to $i = i_M$, or $\rho = 1$, we observe that

$$(r\rho'(r)D_\rho\tilde{\phi})_{i_M+\frac{1}{2},j} = (r\rho'(r)\tilde{\phi}_\rho)_{i_M+\frac{1}{2},j} + O(\Delta\rho^2)$$

$$= (r\tilde{\phi}_r)_{i_M^+,j} + \frac{\Delta\rho}{2}\partial_\rho(r\rho'(r)\tilde{\phi}_\rho)_{i_M^+,j} + O(\Delta\rho^2)$$

and similarly

$$(r\rho'(r)D_\rho\tilde{\phi})_{i_M-\frac{1}{2},j} = (r\tilde{\phi}_r)_{i_M^-,j} - \frac{\Delta\rho}{2}\partial_\rho(r\rho'(r)\tilde{\phi}_\rho)_{i_M^-,j} + O(\Delta\rho^2)$$

It follows then

$$
\begin{aligned}
&-\epsilon^+ D_\rho(r\rho'(r)D_\rho\tilde{\phi})_{i_M,j} \\
= \quad &-\epsilon^+[(r\tilde{\phi}_r)]_{i_M,j} + \frac{\epsilon^+}{r\rho'(r)}\frac{1}{2}((\partial_\theta^2\tilde{\phi})_{i_M^+,j} + (\partial_\theta^2\tilde{\phi})_{i_M^-,j}) \\
&-\frac{1}{2}\frac{r}{\rho'(r)}((H\tilde{\phi})_{i_M^+,j} + (H\tilde{\phi})_{i_M^-,j}) + \frac{1}{2}\frac{r}{\rho'(r)}(f_{i_M^+,j} + f_{i_M^-,j}) + O(\Delta\rho)
\end{aligned}
\tag{36}
$$

where $[\cdot]_{i_M,j}$ denotes the jump across $r = 1$.

Since $\tilde{\phi}$ has continuous first derivative and H, f and $\tilde{\phi}$ are continuous across $r = 1$, (36) reduces to

$$
-\epsilon^+ D_\rho(r\rho'(r)D_\rho\tilde{\phi})_{i_M,j} = \frac{\epsilon^+}{r\rho'(r)}(\partial_\theta^2\tilde{\phi})_{i_M,j} - \frac{r}{\rho'(r)}(H\tilde{\phi})_{i_M,j} + \frac{r}{\rho'(r)}f_{i_M,j} + O(\Delta\rho) \tag{37}
$$

which shows (35) is consistent with local truncation error $O(\Delta\rho)$ on $i = i_M$. The overall error resulting from the discretization (35) is thus $(\Delta\rho^2)$.

REMARKS.
1. We have explained the idea of coordinate patching to handle the infinite domain problem. It can also be applied to problems with more complex molecule surface.
2. In the example above, we may choose a smooth patching function: $r = \tan(\pi\rho/2)$, which is smooth everywhere and suitable for higher order discretizations.

2.5. Summary of the Algorithm. We summarize the numerical scheme for nonlinear Poisson Boltzmann equation as follows.

Step 1 Evaluate the free space Poisson kernel $\phi^*(\boldsymbol{x})$ and $\phi_n^*(\boldsymbol{x})$ for $\boldsymbol{x} \in \Gamma_1$ according to (8). When the number of atoms is large and efficient evaluation of the free space kernel becomes an issue, standard fast summation algorithm such as the tree code [22] or the fast Multipole method [11] can be adopted to gain efficiency.

Step 2 Evaluate $\phi_n^0(x)$ for $x \in \Gamma_1$ by solving the Laplace equation (9) in Ω_1. The jump conditions for the correction potential across Γ_1 is defined by $[\tilde{\phi}] = 0$ and $[\epsilon\tilde{\phi}] = \epsilon^-(\phi_n^* + \phi_n^0)$.

Step 3 Compute the correction potential $\tilde{\phi}$ according to (11) using the monotone iteration (15) and/or the damped Newton iteration (18). The resulting linearized Poisson Boltzmann equation (21) is discretized using the jump condition capturing scheme with the jump conditions defined in step 2. The initial trial of $\tilde{\phi}$ of the iteration is set to be zero.

Step 4 Evaluate $\phi = \overline{\phi} + \tilde{\phi}$ and $\nabla_h\phi = \nabla\phi^* + \nabla_h\phi^0 + \nabla_h\tilde{\phi}$ from (7), (8), (9) and the solution computed in step 3.

3. Numerical Examples. In the following numerical examples, the macromolecules have various shapes and roughly occupy the region $r < 50\mathring{A}$. We use different parameters in the regime specified in (5). We take the unit disk $\{\rho \leq 1\}$ as the computational domain with $n_r \times n_\theta$ uniform grids in polar coordinate. The circle $\rho = 5/9$ is mapped to the interface in physical domain and the linearized Poisson Boltzmann equation (11) is then discretized using the jump condition capturing scheme.

In example 1-3, we perform standard accuracy check. That is, we substitute the exact solution into the Poisson-Boltzmann equation to generate an additional source term and append it to the right hand side of the equation. Moreover, we impose the

$n_r \times n_\theta$	L^∞ error in ϕ	order	L^∞ error in $\epsilon^- \phi_n^-$	order
14×32	6.066E-03	—	1.214E-01	—
23×64	2.253E-03	1.429	6.372E-02	0.930
41×128	6.315E-04	1.835	1.747E-02	1.867
77×256	1.810E-04	1.803	4.097E-03	2.092
149×512	4.841E-05	1.903	9.481E-04	2.112

TABLE 1
Error and order of accuracy in ϕ and the flux for Example 1.

interface condition derived by the exact solution. In all these tests, we have observed 2nd order accuracy both in the potential ϕ and the electrostatic field $\nabla\phi$ at the cell interface in all practical regimes of the parameters I_s, C and z_i. In example 4, we conduct a genuine simulation with various configurations of the charge distributions. It is clear from our simulation that the resulting electrostatic field is well resolved and nontrivial at the cell interface.

EXAMPLE 1

In this test example, Ω_1 is given by a star-shaped region (Fig 2-3) $r < 50 * (1 + 0.2\cos(4\theta))$ with the exact solution:

$$u(x,y) = \begin{cases} 0.1\exp(x/2)\cos(y/2) & \text{inside } \Gamma_1 \\ 100\exp(-\overline{\kappa}/\sqrt{\epsilon^+}r) & \text{outside } \Gamma_1 \end{cases} \tag{38}$$

We take $\overline{\kappa}^2 = 2.0\mathring{A}^{-2}$, which corresponds to the ionic strength $I_s = 0.2357$. The result is listed in Table 1.
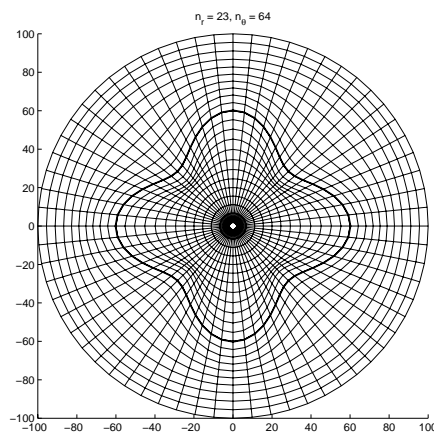

FIG. 2. *The shape of the macromolecule for Example 1*

EXAMPLE 2

The interface is given by an ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

with $a = 50 * 18/27\mathring{A}$ and $b = 50 * 10/27\mathring{A}$ (Fig4-5). Two charges of opposite sign that corresponds to $C = 10000$ and $z_i = \pm 1$ are located at $\boldsymbol{x}_i = (0, \mp 1)$ respectively.
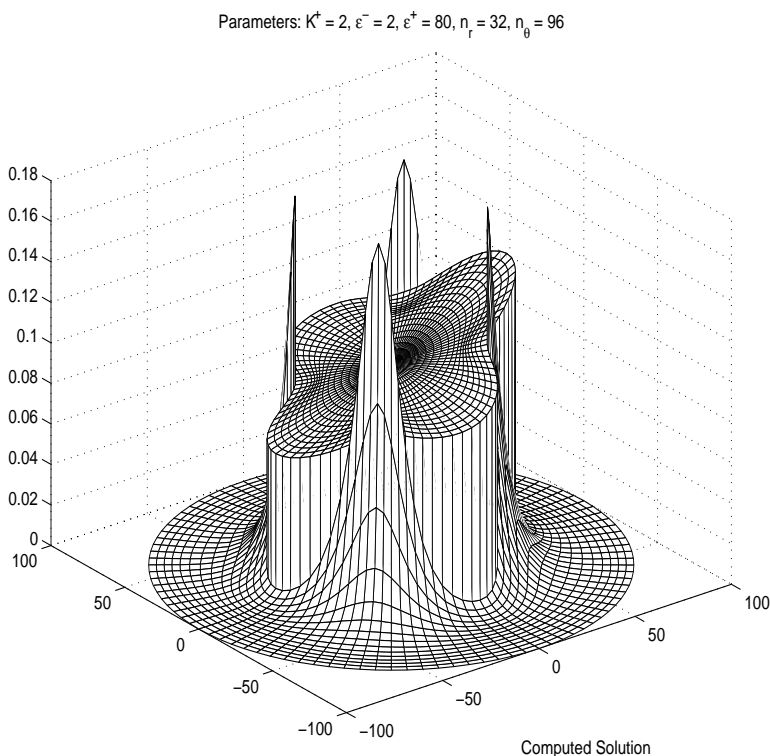
Parameters: $K^+ = 2, \epsilon^- = 2, \epsilon^+ = 80, n_r = 32, n_\theta = 96$

Computed Solution

FIG. 3. *Computed solution for Example 1*

$n_r \times n_\theta$	L^∞ error in ϕ	order	L^∞ error in $\epsilon^- \phi_n^-$	order
14×32	1.242E-02	—	2.208E-01	—
23×64	2.185E-03	2.507	4.968E-02	2.152
41×128	5.896E-04	1.890	1.315E-02	1.918
77×256	1.596E-04	1.885	3.681E-03	1.837

TABLE 2
Error and order of accuracy in ϕ and the flux for Example 2.

The results are shown in Table 2. In this example, we have taken $\overline{\kappa}^2 = 100 \mathring{A}^{-2}$ and the regular part of the solution is given by

$$u(x,y) + \frac{C}{\epsilon^-} \sum_{i=1}^{2} \frac{z_i}{2\pi} \log(|\boldsymbol{x} - \boldsymbol{x}_i|) = \begin{cases} x^2 - 2y^2 & \text{inside } \Gamma_1 \\ 3\exp(-x^2 - y^2/2) & \text{outside } \Gamma_1 \end{cases} \qquad (39)$$

EXAMPLE 3

In this example, the interface is given by $r = 50 * (1 + 0.2\cos(6\theta))$ (Fig 6-7). Two positive and two negative point charges are located on $r = 50$, $\theta = 0, 2\pi/3, \pi$ and $5\pi/3$ respectively and correspond to $C = 5$ and $z_i = \pm 1$. Here we take $\overline{\kappa}^2 = 40 \mathring{A}^{-2}$
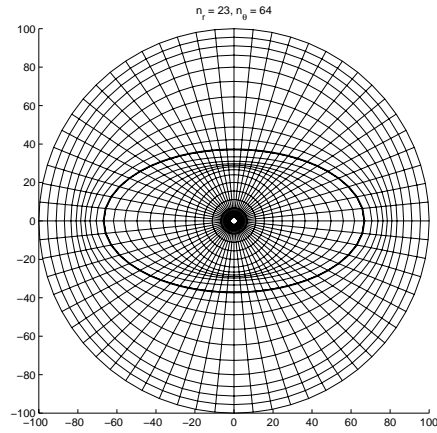
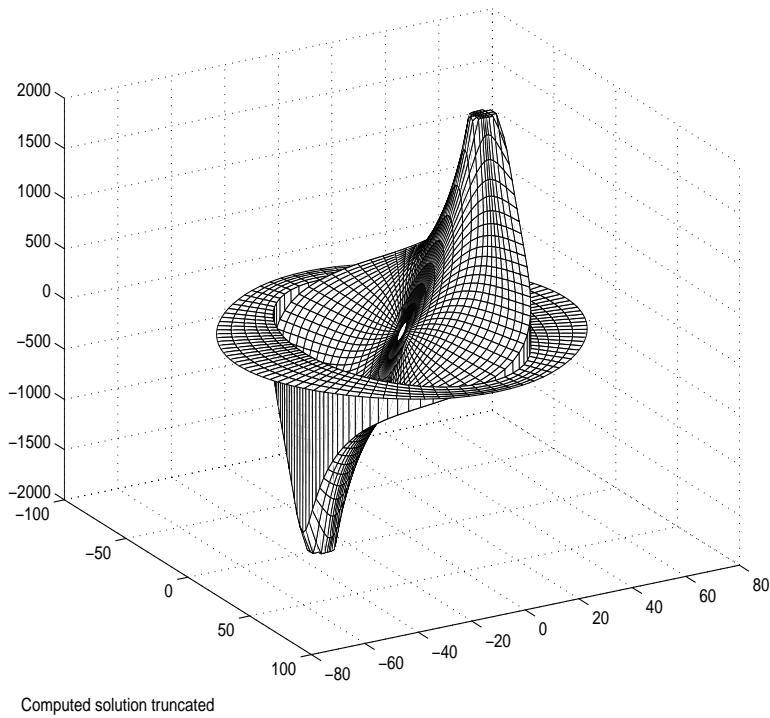FIG. 4. *The shape of the macromolecule for Example 2*

FIG. 5. *Computed solution for Example 2*

and the regular part of the exact solution as

$$u(x,y) + \frac{C}{\epsilon^-} \sum_{i=1}^{4} \frac{z_i}{2\pi} \log(|\boldsymbol{x} - \boldsymbol{x}_i|) = \begin{cases} x^2 + y^2 & \text{inside } \Gamma_1 \\ 0 & \text{outside } \Gamma_1 \end{cases} \tag{40}$$

The result is listed in Table 3. Note that in Fig 8, a typical error plot is given which

shows the error being continuous and piecewise smooth with a kink at the interface. This may explain why we are able to reconstruct the electrostatic field $\epsilon\nabla\phi$ to 2nd order accuracy even on Γ_1. See [31] for details.
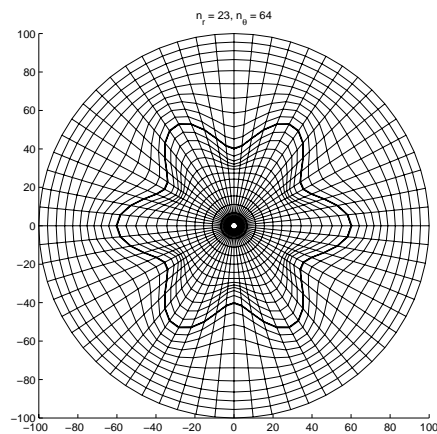

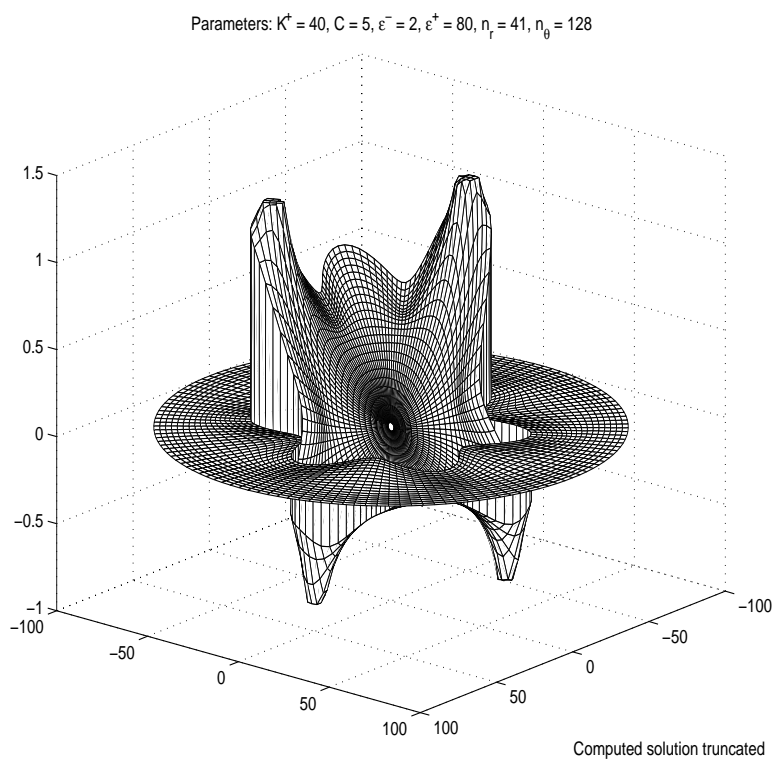FIG. 6. *The shape of the macromolecule for Example 3 and 4*
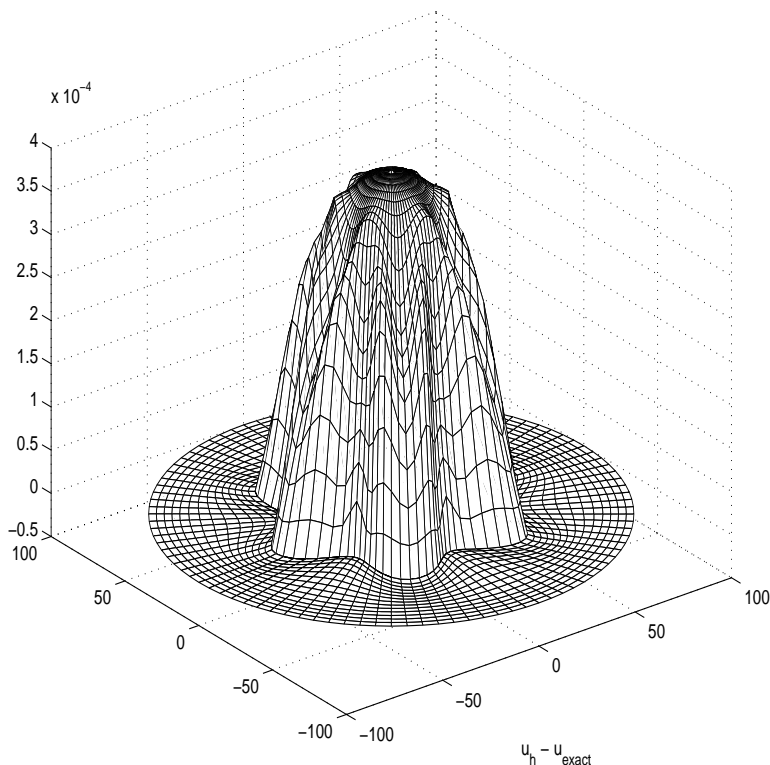

FIG. 7. *Computed solution for Example 3*

FIG. 8. *Error of the computed solution for Example 3*

$n_r \times n_\theta$	L^∞ error in ϕ	order	L^∞ error in $\epsilon^- \phi_n^-$	order
14×32	3.438E-03	—	1.227E-01	—
23×64	8.859E-04	1.956	3.388E-02	1.857
41×128	2.232E-04	1.989	8.566E-03	1.984
77×256	5.604E-05	1.994	2.148E-03	1.996

TABLE 3
Error and order of accuracy in ϕ and the flux for Example 3.

EXAMPLE 4

In this example, we perform an actual simulation on the Poisson-Boltzmann equation. The interface is the same as in Example 3. Here we take $\overline{\kappa}^2 = 1.27\mathring{A}^{-2}$ and $C = 15,000$. In case (a), we put six charges with alternating sign corresponding to $z_i = \pm 1$. In case (b), z_i's are randomly chosen between ± 1 and sum to zero. The results are plotted in Fig 9-11.

4. Conclusion. We propose an accurate numerical procedure to evaluate electrostatic potential and force induced by a macromolecule in solution. The main ingredients of this numerical procedure include the follows.

   (i) It uses point charge potential induced by the macromolecule in free space to separate the effect of point charges from those of molecular surface and nonlinear response of mobile ions.

Parameters: $K^+ = 1.27$, $C = 15000$, $\varepsilon^- = 2$, $\varepsilon^+ = 80$, $n_r = 32$, $n_\theta = 96$
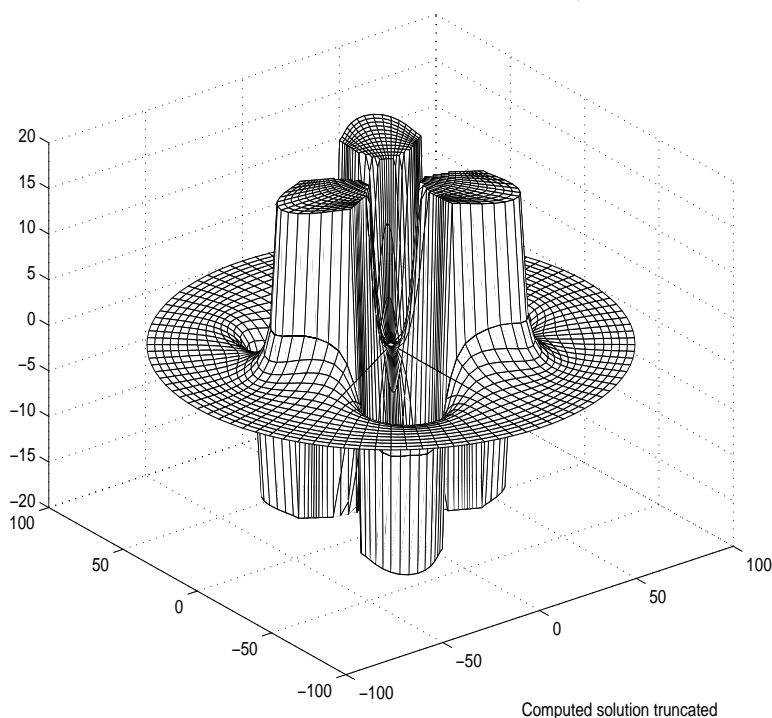

Computed solution truncated

FIG. 9. *Computed solution for Example 4a*

(ii) It uses a jump condition capturing scheme with body-fitting grid to solve the discontinuity problem of the dielectric coefficient across molecular surface.

(iii) A grid patching technique is used to handle the infinite domain problem.

The resulting stiffness matrix is symmetric and positive definite, thus standard preconditioned conjugate gradient method can be used to achieve linear convergent rate for inner iteration. More importantly, the scheme is second-order accurate for both potential and its gradient, as shown in all numerical tests.

REFERENCES

[1] S. A. ALLISON, J. J. SINES AND A. WIERZBICKI, *Solutions of the full Poisson-Boltzmann equation with application to diffusion-controlled reactions,* J. Phys. Chem., 93, 5819 (1989).

[2] R. E. BANK AND D. J. ROSE, *Global approximate Newton methods,* Numer. Math., 37, 279 (1981).

[3] N. BAKER, M. HOLST, AND F. WANG, *Adaptive Multilevel Finite Element Solution of the Poisson- Boltzmann Equation; II: Refinement at Solvent Accessible Surfaces in Biomolecular Systems,* J. Comput. Chem., 21 (2000), pp. 1343-1352.

[4] N. A. BAKER, D. SEPT, M. J. HOLST AND J. A. MCCAMMON, *The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers,* IBM J. Res. & Dev., 45:3/4, 427 (2001).

[5] J. M. BRIGGS AND J. A. MCCAMMON, *Computation unravels mysteries of molecular biophysics,* Computers in Physics, 6:3, 238 (1990).

[6] C. M. CORTIS AND R. A. FRIESNER, *An automatic three-dimensional finite element mesh generation system for the Poisson-Boltzmann equation,* J. Comput. Chem., 18:13 (1997), pp. 1570–1590.
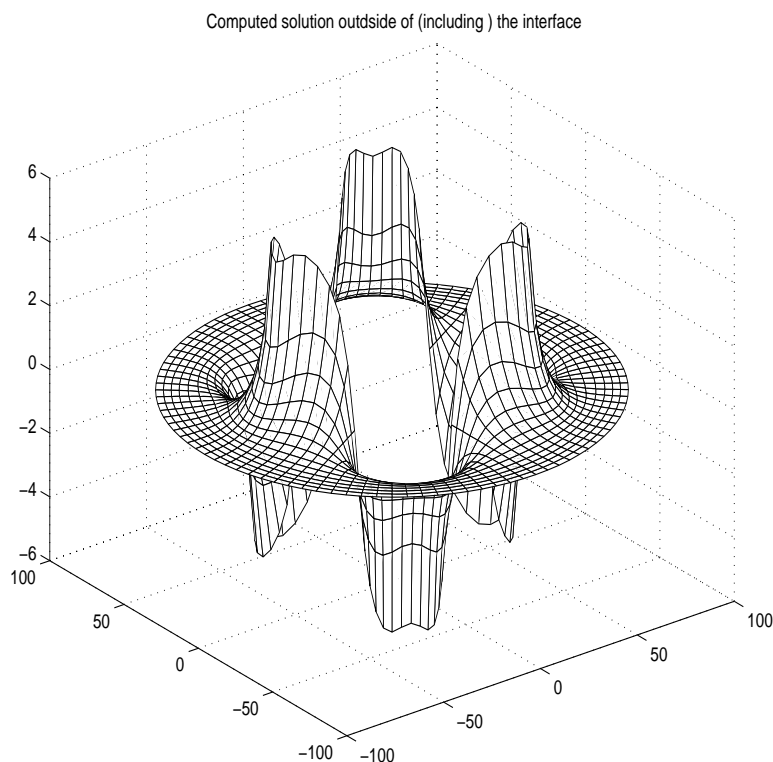
Computed solution outdside of (including) the interface

FIG. 10. *Computed solution outside of Γ_1 for Example 4a*

[7] C. M. CORTIS AND R. A. FRIESNER, *Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes,* J. Comput. Chem., 18:13 (1997), pp. 1591–1608.

[8] M. E. DAVIS AND J. A. MCCAMMON, *Electrostatics in Biomolecular Structure and Dynamics,* Chem. Rev., 94 (1990), pp. 7684–7692.

[9] P. DEBYE AND E. HÜCKEL, Physik. Z., 24, 185 (1923).

[10] M. GILSON, K. SHARP AND B. HONIG, *Calculating the electrostatic potential of molecules in solution: method and error assessment,* J. Comp. Chem., 9, 327 (1987).

[11] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulation,* J. Comput. Phys., 73, 325 (1987).

[12] M. HOLST, *Multilevel methods for the Poisson-Boltzmann equation,* Ph.D. thesis, Numerical Computing Group, University of Illinois at Urbana-Champaign, 1993, Also published as Technical Report UIUCDCS-R-03-1821.

[13] M. HOLST, N. BAKER, AND F. WANG, *Adaptive Multilevel Finite Element Solution of the Poisson-Boltzmann Equation; I: Algorithms and Examples,* J. Comput. Chem., 21 (2000), pp. 1319–1342.

[14] M. HOLST AND F. SAIED, *Multigrid solution of the Poisson-Boltzmann equation,* J. of Comp. Chem., 14, 105 (1993).

[15] B. HONIG AND A. NICHOLLS, *Classical Electrostatics in Biology and Chemistry,* Science, 268 (1995), pp. 1144–1149.

[16] Y. L. HUANG AND W. C. WANG, *A monotone jump condition capturing scheme for elliptic interface problems on irregular domains,* in preparation, 2003.

[17] A. NICHOLLS AND B. HONIG, *A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation,* J. Comp. Chem., 12, 435 (1991).

[18] B. JAYARAM, K. A. SHARP, AND B. HONIG, *The electrostatic potential of B-DNA,* Biopolymers, 28, 975 (1989).

[19] J.G. KIRKWOOD, J. Chem., Phys., 2, 351 (1934).

[20] B. LEE AND F. M. RICHARDS, *The Interpretation of Protein Structures: Estimation of Static*

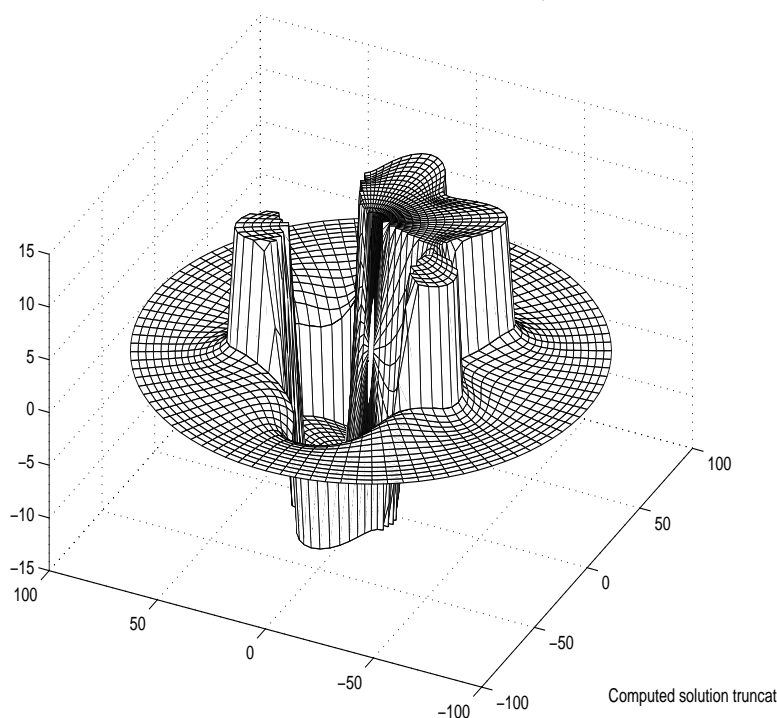Parameters: $K^+ = 1.27$, $C = 9773$, $\varepsilon^- = 2$, $\varepsilon^+ = 80$, $n_r = 32$, $n_\theta = 96$

Fig. 11. *Computed solution for Example 4b*

Accessibility, J. Mol. Biol., 55 (1971), pp. 379–400.

[21] R.J. LeVeque and Z. Li, *The immersed interface method for elliptic equations with discontinuous coefficients and singular sources,* SIAM J. Numer. Anal., 31, 1019 (1994)

[22] K. Lindsay and R. Krasny, *A particle method and adaptive treecode for vortex sheet motion in three-dimensional flow,* J. of Comp. Phys., 172 (2001), pp. 879–907.

[23] B. Luty, M. Davis and J. McCammon, *Electrostatic energy calculations by a finite-difference method: rapid calculation of charge-solvent interaction energies,* J. Comp. Chem., 13, 768 (1992).

[24] C.S. Peskin, *Numerical analysis of blood flow in the heart,* J. Comput. Phys., 25, 220 (1977)

[25] W.H. Press, B.P. Flannery, S.A. Teukolsky and W.T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing,* 2nd ed., Cambridge University Press, New York, 1996.

[26] A. A. Rashin and J. Malinsky, *New method for the computation of ionic distribution around rod-like polyelectrolytes with helical distribution of charges. I. General approach and a nonlinearized Poisson-Boltzmann equation,* J. Comput. Chem., 12:8 (1991), pp. 981–993.

[27] R.B. Setlow and E.C. Pollard, *Molecular Biophysics,* Addison-Wesley, Boston, 1962.

[28] K. A. Sharp and B. Honig, Annu. Rev. Biophys. Biophys. Chem., 94, 7684 (1990).

[29] K. A. Sharp and B. Honig, *Calculating total electrostatic energies with the nonlinear Poisson-Boltzmann equation,* J. Phys. Chem., 94 (1990), pp. 7684–7692.

[30] B.F. Smith, P.E. Bjorstad, W.D. Gropp, *Domain Decomposition : Parallel Multilevel Methods for Elliptic Partial Differential Equations,* Cambridge University Press, New York, 1996.

[31] W. Wang, *A Jump Condition Capturing Scheme for Elliptic Interface Problems,* to appear in SIAM J. Sci. Comp. (2003).

[32] T. You and D. Bashford, *An analytical algorithm for the rapid determination of the solvent accessibility of points in a three-dimensional lattice around a solute molecule,* J. Comput. Chem., 16:6, 753 (1995).
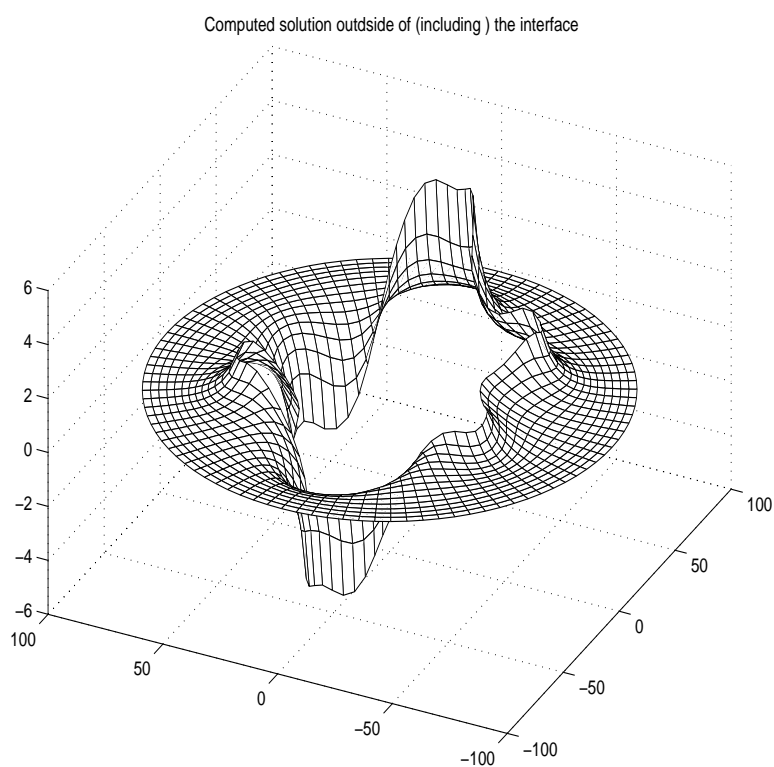
Computed solution outdside of (including) the interface

Fig. 12. *Computed solution outside of Γ_1 for Example 4b*