Version of Record: https://www.sciencedirect.com/science/article/pii/S0021999117306861
Manuscript_a3361180eeb9b3b641a5c633a215568b

A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation

Weihua Geng[a], Shan Zhao[b,*]

[a] *Department of Mathematics, Southern Methodist University, Dallas, TX 75275 USA*
[b] *Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487 USA*

Abstract

We present a new Matched Interface and Boundary (MIB) regularization method for treating charge singularity in solvated biomolecules whose electrostatics are described by the Poisson-Boltzmann (PB) equation. In a regularization method, by decomposing the potential function into two or three components, the singular component can be analytically represented by the Green's function, while other components possess a higher regularity. Our new regularization combines the efficiency of two-component schemes with the accuracy of the three-component schemes. Based on this regularization, a new MIB finite difference algorithm is developed for solving both linear and nonlinear PB equations, where the nonlinearity is handled by using the inexact-Newton's method. Compared with the existing MIB PB solver based on a three-component regularization, the present algorithm is simpler to implement by circumventing the work to solve a boundary value Poisson equation inside the molecular interface and to compute related interface jump conditions numerically. Moreover, the new MIB algorithm becomes computationally less expensive, while maintains the same second order accuracy. This is numerically verified by calculating the electrostatic potential and solvation energy on the Kirkwood sphere on which the analytical solutions are available and on a series of proteins with various sizes.

Keywords: Electrostatics; Poisson-Boltzmann equation; Elliptic interface problem; Green's function; Finite difference method; Matched interface and boundary (MIB).

1. Introduction

Electrostatic interactions between a biomolecule and its solvent environment play an important role in understanding the structure, function, dynamics, and mechanism of the biomolecules [28, 49]. Computing these interactions using explicit solvent models is computationally expensive, and a number of less costly implicit solvent models have been developed [16, 42]. Here we consider a model based on the Poisson-Boltzmann (PB) equation which treats the solute biomolecule as a low-dielectric medium with embedded atomic charges and the solvent as a high-dielectric medium with dissolved ions [3, 14, 17, 32]. The solute in general may be a protein or a complex system involving membranes [9] or nucleic acids [5].

Since the PB equation has analytic solution only for the simple geometries, such as sphere and rod, in practice the equation is solved numerically. Numerical methods for solving PB equation fall into two classes, (1) grid-based methods that discretize the entire domain, e.g. [2, 4, 8, 10, 13, 15, 28, 29, 36], and (2) boundary integral methods that discretize the molecular surface, e.g. [1, 7, 19, 24,

*Corresponding author
Email addresses: wgeng@smu.edu (Weihua Geng), szhao@ua.edu (Shan Zhao)

30, 34, 35, 48]. The numerical solution to the PB equation suffers many challenges such as (1) the interfacial molecular surface is geometrically complex; (2) the biomolecule is represented by singular point charges; (3) the dielectric function and electric field are discontinuous across the molecular surface; (4) the nonlinearity appears when ionic strength is strong; (5) the domain is unbounded. In treating these numerical challenges, particularly (1)-(3), a finite-difference based interface method named matched interface and boundary (MIB) method stands out [10, 21, 46]. The MIB method is designed to capture discontinuities of the solutions and coefficients of the differential equations by using local high order interpolation assisted with the interface jump conditions. Based on the MIB method for general interface problems [50–52], the MIB PB solver is further improved to be able to maintain its designed 2nd order of convergence and accuracy in the presence of geometric singularities of molecular surfaces [46].

A challenging issue in the MIB PB solver is the efficient treatment of charge singularities. Conventionally, in most finite difference based PB solvers, Dirac delta functions in the PB equation, or singular partial charges inside a biomolecule, are redistributed to their neighboring grid points. This low order treatment works well in commonly used PB solvers where the major error is due to the neglect of interface continuity conditions, but produces a pronounced error in the MIB PB solver, especially when the mesh size is large. This is because the grid points involved in the MIB interface treatments may also carry distributed charges, which results in an interference of geometric interface and charge singularities, and leads to an accuracy reduction [21].

The accurate treatment of charge singularities has been studied by many authors, giving rise to a series of regularization approaches [8, 11, 12, 21, 25, 45, 53]. In these methods, the potential function is decomposed into a singular component plus one or two other components. Satisfying a Poisson equation with the same PB singular sources, the singular component can be analytically solved as Coulomb potentials or Green's functions. After removing the singular component, the other potential components satisfy elliptic partial differential equations (PDEs) with a higher regularity, and the contribution of charge singularities is transferred into new interface jump conditions for these PDEs. Generally speaking, a three-component regularization [12, 21, 25, 45] is more accurate than a two-component method [11, 53], while the latter is more efficient.

The charge singularity has been resolved by using a three-component regularization [12] in the previous MIB PB algorithm [21]. The charge-induced interface jump conditions are perfectly compatible with the MIB techniques for complex interfaces [46], so that one can treat geometric and charge singularities on an equal footing. The resulting MIB PB solver is capable of eliminating the interference of the geometric and charge singularities in solving the PB equation, and is of second order convergence. The MIB PB method is able to deliver high numerical accuracy at a grid spacing as large as the van der Waals radius (about 1.2Å) for macromolecules [21].

Inspired by the original idea of charge singularity regularization from [21] and [8], the present work introduces a novel two-component formulation which combines the efficiency of two-component schemes with the accuracy of three-component methods. Compared with the previous three-component method [21], this new regularization becomes simpler to implement, because it circumvents the additional work of solving a boundary value Poisson equation inside the molecular interface and its related interface jump conditions computation for the regularized Poisson-Boltzmann equation. Moreover, the discontinuous function and flux jumps can be easily handled in the MIB discretization without introducing any additional difficulty. Thus, based on this new regularization, the MIB PB solver improves its efficiency, while maintains all good features including the same second order accuracy. More importantly, the reduced complexity largely simplifies the work needed to compute the electrostatic field and forces for molecular dynamics simulation under the MIB frame, which will be conducted in another related project.

The rest of the paper is organized as follows. Section 2 is devoted to the theoretical formulation

and computational algorithm. The two-component Green's function formulation of charge singularities is developed in the framework of the MIB method. The resulting two-component regularized MIB method is extensively validated in Section 3. The benchmark tests, such as the Kirkwood sphere [31] and a set of twenty four proteins, are used to examine the accuracy and test the speed of convergence of the proposed method. Comparisons with the previously established MIB PB solver are given. This article ends with a brief conclusion summarizing the main points.

2. Theory and algorithm

In this section, we will first briefly describe the Poisson-Boltzmann (PB) model. A summary of the matched interface and boundary method (MIB) will be offered next. Then, the treatment of charge singularity of the original three-component decomposition and the new two-component decomposition will be discussed in details. This section ends with discussions of the inexact-Newton's method for addressing the nonlinearity and the formulation for computing electrostatic free energy of solvation.

2.1. Poisson-Boltzmann model

The PB model is illustrated in Fig. 1(a). Consider a large domain Ω in $\mathbb{R}^3$ containing the solute protein. The domain Ω is divided by the molecule surface Γ into the molecule domain Ω^- with dielectric constant ϵ^- and the solvent domain Ω^+ with dielectric constant ϵ^+, i.e., $\Omega = \Omega^- \bigcup \Omega^+$. Denote the boundary of Ω as $\partial\Omega$. Charges in Ω^- are partial charges assigned to the centers of atoms by using force field, while charges in Ω^+ are mobile ions described by the Boltzmann distribution. For $\mathbf{r} \in \mathbb{R}^3$, applying Gauss's law to the charge distribution in both Ω^- and Ω^+ leads to the PB equation

$$-\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(\phi(\mathbf{r})) = \rho(\mathbf{r}), \tag{1}$$

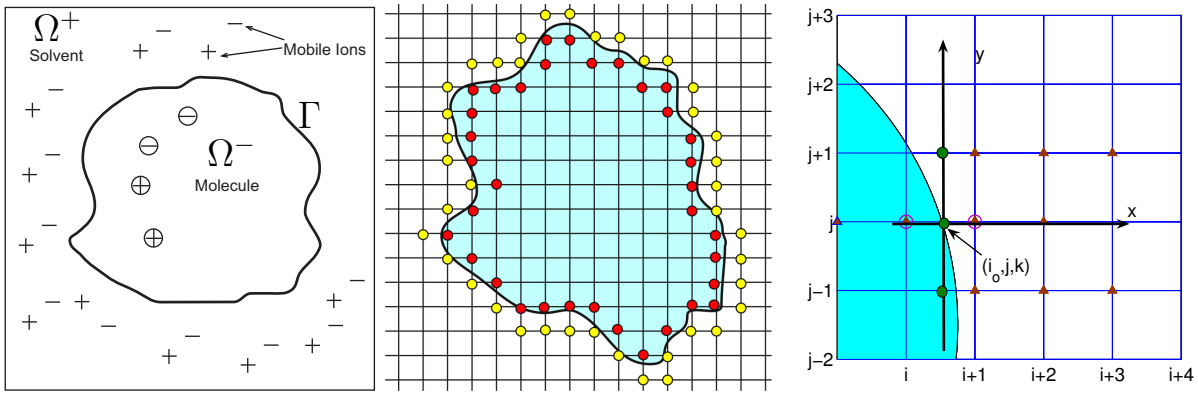
subject to interface continuity for the potential ϕ and flux density $\epsilon\phi_{\mathbf{n}}$,

$$[\phi]_\Gamma = 0 \text{ and } [\epsilon\phi_{\mathbf{n}}]_\Gamma = 0. \tag{2}$$

Here $\mathbf{n} = (n_x, n_y, n_z)$ is the outer normal direction of the interface Γ, $\phi_{\mathbf{n}} = \frac{\partial\phi}{\partial\mathbf{n}}$ is the directional derivative in $\mathbf{n}$, and the notation $[f]_\Gamma = f^+ - f^-$ is the difference of the function f cross the interface Γ.

In Eqs. (1) and (2), ϵ is a piecewise function for the dielectric constants in Ω^- and Ω^+. Here κ is the inverse Debye screening length measuring ionic strength, and its modified version $\bar{\kappa}$ is given as $\bar{\kappa}^2 = \epsilon^+\kappa^2$. The value of κ is nonzero in Ω^+ only and it can be related to ionic strength I in M or mol/L by the relation $\kappa^2 = 8.430325455I/\epsilon^+$. The source term ρ in Eq. (1) is the summation of the charge distribution in Ω^- using Delta function for N_c partial charges located at $\mathbf{r}_i$, i.e., $\rho(\mathbf{r}) = 4\pi C \sum_{i=1}^{N_c} q_i\delta(\mathbf{r} - \mathbf{r}_i)$, where q_i is the partial charge on the ith atom with units e_c and C is a constant to balance the units. For example, when $C = \frac{e_c}{k_B T}$, where e_c is the fundamental charge, k_B is the Boltzmann constant, and T is the absolute temperature, the potential ϕ is dimensionless. In our numerical implementation, the unit of the potential ϕ is $e_c/$Å and unit for the length is Å, which conveniently makes $C = 1$. After computing ϕ, one needs to multiply it with a factor 332.0716 for $T = 298$K to convert it to the popular unit of kcal/mol/e_c for free energy calculation. The reader can refer to [18, 26] for more details about definition and units of these coefficients.

Due to the singular partial charges in the source $\rho(\mathbf{r})$, the electrostatic potential $\phi(\mathbf{r})$ goes to infinity as $\mathbf{r} \to \mathbf{r}_i$. This brings considerable difficulties in numerical solution of the PB equation. We



(a) (b) (c)

Figure 1: (a) The PB model, (b) The MIB scheme illustrated in a 2-D setting with inside fictitious points in red and outside fictitious points in yellow, (c) The MIB scheme in finding the fictitious values at (i, j, k) and $(i + 1, j, k)$ in the cross section of $z = z_k$.

will present a new regularization approach to remove the singularity under the matched interface and boundary (MIB) framework. On the outer boundary $\partial\Omega$, a Dirichlet boundary condition can be assumed [26]

$$\phi(\mathbf{r}) = \phi_b(\mathbf{r}) := C \sum_{i=1}^{N_c} \frac{q_i e^{-\kappa|\mathbf{r}-\mathbf{r}_i|}}{\epsilon^+|\mathbf{r}-\mathbf{r}_i|}. \tag{3}$$

It is obviously that $\lim_{|\mathbf{r}|\to\infty} \phi(\mathbf{r}) = 0$, which is actually the physical boundary condition defined at the infinity. When $\partial\Omega$ is sufficiently distanced from the protein, Eq. (3) provides a fairly accurate Debye-Huckel approximation [26] to the analytical solution of the linearized PB equation. The accuracy of the Debye-Huckel boundary approximation, particularly for the nonlinear PB model, has been discussed in [6]. In the present study, the domain Ω is chosen to be large enough such that the boundary approximation error is negligible.

2.2. Matched interface and boundary (MIB) method

We use the linearized PB equation as in Eq. (4) to explain the key ideas of the MIB method for solving the elliptic interface problem with discontinuous coefficients

$$-\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r})\phi(\mathbf{r}) = \rho(\mathbf{r}). \tag{4}$$

The interface Γ divides the whole domain into two separated parts, Ω^- and Ω^+. The jump conditions across the interface are assumed to be

$$[\phi]_\Gamma = g_0(\mathbf{r}), \tag{5}$$
$$[\epsilon\phi_{\mathbf{n}}]_\Gamma = g_1(\mathbf{r}), \tag{6}$$

where the nonhomogeneous jump data $g_0(\mathbf{r})$ and $g_1(\mathbf{r})$ are either given or computable. We note that discontinuous function and flux jumps, i.e., nonzero $g_0(\mathbf{r})$ and $g_1(\mathbf{r})$, can be easily handled in the MIB scheme, without introducing any difficulty. The corresponding discretization matrix will be the same as that for the homogeneous case with $g_0(\mathbf{r}) = g_1(\mathbf{r}) = 0$, while the right hand side shall be modified to account for $g_0(\mathbf{r})$ and $g_1(\mathbf{r})$.

Consider a uniform Cartesian grid partition of the domain Ω. It is well known that the standard finite difference schemes lose their designed convergence near the interface and the interface jump conditions have to be used to restore the accuracy. To this end, all the grid points in Ω are classified into two types, the regular ones and the irregular ones. At the regular points away from the interface Γ, a central difference discretization of Eq. (4) is carried out, which involves a grid node (x_i, y_j, z_k) and its six neighboring points. An *irregular* grid point is defined as a node at which the standard finite difference scheme involves grid points across the interface, i.e., at least one of its six neighboring points is from the other side of the interface. In the MIB scheme, the finite difference approximations at irregular points will be modified by using fictitious values from the other side of the interface. For example, consider a two-dimensional (2D) cross section $z = z_k$ shown in Figure 1(c), and denote $\phi_{i,j,k} = \phi(x_i, y_j, z_k)$. We have the following modified finite difference approximation for the x derivative

$$\frac{\partial^2}{\partial x^2}\phi_{i+1,j,k} \approx \frac{1}{\Delta x^2}(f_{i,j,k} - 2\phi_{i+1,j,k} + \phi_{i+2,j,k}), \tag{7}$$

where $f_{i,j,k}$ is a fictitious value defined at (x_i, y_j, z_k). The modified finite difference approximations at irregular points maintain the second order of accuracy, provided that the fictitious values are accurately estimated.

In considering x, y, and z derivatives at all irregular points, it is sufficient to accurately generate two layers of fictitious values f surrounding Γ, one inside and one outside. Figure 1(b) shows an example in 2D for the MIB scheme with inside fictitious points in red and outside fictitious points in yellow. Physically, the fictitious values can be regarded as smooth extension of function values from the other side. Numerically, they are determined by discretizing jump conditions (5) and (6). In the MIB scheme, a fictitious value $f_{i,j,k}$ at (x_i, y_j, z_k) will be represented as a linear combination of function values on a set of neighboring nodes $\mathbb{S}_{i,j,k}$ and nonhomogeneous jump data (g_0, g_1)

$$f_{i,j,k} = \sum_{(x_I, y_J, z_K) \in \mathbb{S}_{i,j,k}} w_{I,J,K}\phi_{I,J,K} + w_0 g_0 + w_1 g_1. \tag{8}$$

The major task of a particular MIB approximation is to determine the points set $\mathbb{S}_{i,j,k}$ and the representation weights $w_{I,J,K}$, w_0, and w_1 via discretizing Eqs. (5) and (6). Finally, Eq. (8) will be substituted into Eq. (7) to modify the x derivative approximation. When all necessary x, y, and z derivatives are corrected in discretizing Eq. (4), a discretized linear algebraic system of equations is generated, in which the nonhomogeneous data (g_0, g_1) only affects the right hand side of the equation.

A key idea in the MIB fictitious value determination is to decompose the three-dimensional (3D) jump conditions (5) and (6) so that they can be imposed in a one-dimensional (1D) manner. In the following, we will describe the MIB interface treatment by considering two irregular points located at (i, j, k) and $(i + 1, j, k)$ in the cross section of $z = z_k$ as illustrated in Fig. 1(c).

Assume that the interface Γ intersects the grid line in the x-direction at a point (i_o, j, k), which is located between (i, j, k) and $(i + 1, j, k)$. To determine the fictitious values $f_{i,j,k}$ and $f_{i+1,j,k}$, a derivative in the normal direction of the interface at point (i_o, j, k) has to be treated in Eq. (6). It is convenient to introduce a local coordinates (ξ, η, ζ) such that ξ is along the normal direction ($\xi = \mathbf{n}$), and η and ζ are in the tangential plane. The coordinate transformation can be given as

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \mathbf{P} \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \tag{9}$$

where $\mathbf{P}$ is the transformation matrix

$$\mathbf{P} = \begin{bmatrix} \sin\psi\cos\theta & \sin\psi\sin\theta & \cos\psi \\ -\sin\theta & \cos\theta & 0 \\ -\cos\psi\cos\theta & -\cos\psi\sin\theta & \sin\psi \end{bmatrix}. \tag{10}$$

Here θ and ψ are the azimuth and zenith angles with respect to the normal direction ξ , respectively. By differentiating Eq. (5) along two tangential directions, η and ζ, we can generate two additional jump conditions

$$[\phi_\eta]_\Gamma = (\phi_x^+ p_{21} + \phi_y^+ p_{22} + \phi_z^+ p_{23}) - (\phi_x^- p_{21} + \phi_y^- p_{22} + \phi_z^- p_{23}) = \frac{\partial g_0}{\partial \eta} \tag{11}$$

$$[\phi_\zeta]_\Gamma = (\phi_x^+ p_{31} + \phi_y^+ p_{32} + \phi_z^+ p_{33}) - (\phi_x^- p_{31} + \phi_y^- p_{32} + \phi_z^- p_{33}) = \frac{\partial g_0}{\partial \zeta}, \tag{12}$$

where p_{ij} is the ith row and jth column component of the transformation matrix $\mathbf{P}$. The right hand side terms $\frac{\partial g_0}{\partial \eta}$ and $\frac{\partial g_0}{\partial \zeta}$ can be calculated when the analytical form of g_0 is given.

Since Eq. (9) implies

$$\begin{bmatrix} \phi_\xi \\ \phi_\eta \\ \phi_\zeta \end{bmatrix} = \mathbf{p} \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix}, \tag{13}$$

Eqs. (6), (11) and (12) can be rewritten as follows:

$$\mathbf{C} \begin{bmatrix} \phi_x^+ \\ \phi_x^- \\ \phi_y^+ \\ \phi_y^- \\ \phi_z^+ \\ \phi_z^- \end{bmatrix} = \begin{bmatrix} g_1 \\ \frac{\partial g_0}{\partial \eta} \\ \frac{\partial g_0}{\partial \zeta} \end{bmatrix}, \tag{14}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix} = \begin{bmatrix} p_{11}\epsilon^+ & -p_{11}\epsilon^- & p_{12}\epsilon^+ & -p_{12}\epsilon^- & p_{13}\epsilon^+ & -p_{13}\epsilon^- \\ p_{21} & -p_{21} & p_{22} & -p_{22} & p_{23} & -p_{23} \\ p_{31} & -p_{31} & p_{32} & -p_{32} & p_{33} & -p_{33} \end{bmatrix}, \tag{15}$$

with $\mathbf{C}_i$ represents the ith row of matrix $\mathbf{C}$. For the three jump conditions given in (14), the matrix $\mathbf{C}$ and right hand side terms are known. However, none of these conditions is easy to implement because six derivatives, ϕ_x^+, ϕ_x^-, ϕ_y^+, ϕ_y^-, ϕ_z^+, and ϕ_z^-, are coupled together. For complex solvent-molecule interfaces of macromolecules, it is often very difficult to numerically evaluate some of these derivatives. Therefore, in the MIB method, we avoid calculating two most difficult derivatives by eliminating them from (14). Symbolically, we illustrate the idea by eliminating the lth and mth elements of the array (ϕ_x^+, ϕ_x^-, ϕ_y^+, ϕ_y^-, ϕ_z^+, ϕ_z^-), Eq. (14) then reduces to a single jump condition

$$(a\mathbf{C}_1 + b\mathbf{C}_2 + c\mathbf{C}_3) \begin{bmatrix} \phi_x^+ \\ \phi_x^- \\ \phi_y^+ \\ \phi_y^- \\ \phi_z^+ \\ \phi_z^- \end{bmatrix} = ag_1 + b\frac{\partial g_0}{\partial \eta} + c\frac{\partial g_0}{\partial \zeta}, \tag{16}$$

where

$$a = C_{2l}C_{3m} - C_{3l}C_{2m} \tag{17}$$
$$b = C_{3l}C_{1m} - C_{1l}C_{3m}$$
$$c = C_{1l}C_{2m} - C_{2l}C_{1m}.$$

Note that the jump condition (16) actually involves four derivatives out of (ϕ_x^+, ϕ_x^-, ϕ_y^+, ϕ_y^-, ϕ_z^+, ϕ_z^-), even though all six derivatives are shown in the equation.

A concrete example is discussed here to further illustrate Eq. (16). For the case shown in Fig. 1(c), we will generate a jump condition along x direction by keeping both ϕ_x^+ and ϕ_x^- in Eq. (14). One derivative will be eliminated for each of y and z directions. In y direction, ϕ_y^- is more difficult to compute at (i_o, j, k) than ϕ_y^+. Thus, ϕ_y^- will be eliminated in Eq. (14), while ϕ_y^+ in Eq. (14) will be calculated by using ϕ^+ at three auxiliary points $(i_o, j+1, k)$, (i_o, j, k), and $(i_o, j-1, k)$ (the green solid circles). Since these auxiliary points are not grid nodes, ϕ^+ values on auxiliary points need to be interpolated by using grid values exclusively from the Ω^+. For example, $\phi^+(x_{i_o}, y_{j-1}, z_k)$ will be approximated by ϕ values on $(i+1, j-1, k)$, $(i+2, j-1, k)$, and $(i+3, j-1, k)$. After interpolation, ϕ_y^+ is expressed as a linear combination of some surrounding grid values, and thus is not treated as an unknown in Eq. (14). Similarly, in z direction, one eliminates one derivative, say ϕ_z^-, and interpolates the other derivative, say ϕ_z^+. In practice, care is required to select appropriate interpolation schemes so that the resulting MIB matrix can be optimized for arbitrarily complex solvent-molecule interfaces with geometric singularities [46, 47]. For the present example, the jump condition (16) involves four derivatives ϕ_x^+, ϕ_x^-, ϕ_y^+, and ϕ_z^+, while the last two have been interpolated. Hence, Eq. (16) essentially reduces to a 1D jump condition along x direction.

For the present case, it is sufficient to discretize two x direction jump conditions (5) and (16) to determine the fictitious values $f_{i,j,k}$ and $f_{i+1,j,k}$. In particular, ϕ^+ and ϕ_x^+ at (i_o, j, k) will be approximated by the fictitious value $f_{i,j,k}$ and ϕ values at $(i+1, j, k)$ and $(i+2, j, k)$, and ϕ^- and ϕ_x^- are treated similarly. After solving $f_{i,j,k}$ and $f_{i+1,j,k}$ based on the discretized 1D jump conditions, we can express these fictitious values as a linear combination form as shown in Eq. (8) with the points set and representation weights now known. For example, for the case shown in Fig. 1(c), the points set $\mathbb{S}_{i,j,k}$ includes all brown triangles in the plane $z = z_k$, as well as another six points in the planes $z = z_{k-1}$ and $z = z_{k+1}$. The MIB interface treatment is systematically carried out to determine all fictitious values along any mesh line and at any interface intersection points. More details of this procedure in treating geometric complexity and singularity can be found in [46].

We note finally that in the discretization of jump conditions (5) and (16), fictitious values determination Eq. (8), and finite difference derivative correction Eq. (7), it does not matter if the nonhomogeneous data $g_0(\mathbf{r})$ and $g_1(\mathbf{r})$ are vanishing or not. By allowing $g_0(\mathbf{r})$ to be nonzero, this gives us extra flexibility in designing a new regularization approach for solving the PB equation.

2.3. Three-component regularization

We first review our previous three-component regularization procedure. Numerically, the second and higher-order numerical implementation of Delta functions on Cartesian grid points is feasible with appropriate interpolation schemes. However, the overlap of grid points carrying redistributed partial charges and those involved in the treatment of geometric interface singularities leads to an accuracy reduction. This interference becomes inevitable when a coarse mesh is pursued in an interface treatment. Therefore, a Green's function approach of the singular charges becomes attractive in the MIB interface scheme for the PB equation. In the existing MIB PB algorithm [21], a three-component regularization [12] is utilized. Such an approach decomposes the solution ϕ into regular part $\hat{\phi}$ and singular part $\bar{\phi}$. The latter consists of a fundamental solution to the

Poisson equation with the singular charge ϕ^*, and a harmonic function ϕ^0

$$\phi = \hat{\phi} + \bar{\phi}, \tag{18}$$

where $\bar{\phi}(\mathbf{r})$ is defined as

$$\bar{\phi}(\mathbf{r}) = \begin{cases} \phi^*(\mathbf{r}) + \phi^0(\mathbf{r}) & \mathbf{r} \in \Omega^-; \\ 0 & \mathbf{r} \in \Omega^+. \end{cases} \tag{19}$$

Here $\phi^*(\mathbf{r})$ is the fundamental solution to the Poisson equation with singular charges

$$\phi^*(\mathbf{r}) = G(\mathbf{r}) := C \sum_{i=1}^{N_c} \frac{q_i}{\epsilon^- |\mathbf{r} - \mathbf{r}_i|} \tag{20}$$

which is simply the Green's function $G(\mathbf{r})$. The harmonic component $\phi^0(\mathbf{r})$ satisfies a Laplace equation in Ω^-

$$\begin{cases} \nabla^2 \phi^0(\mathbf{r}) & = 0 \quad \text{in } \Omega^-; \\ \phi^0(\mathbf{r}) & = -G(\mathbf{r}) \quad \text{on } \Gamma. \end{cases} \tag{21}$$

The boundary value problem given by Eq. (21) is to be solved over an irregularly shaped domain Ω^- with possibly geometric singularities on its boundary. As a solution to the PB equation, the singular part $\bar{\phi}(\mathbf{r})$ is solved subject to the following jump conditions

$$[\bar{\phi}]_\Gamma = 0 \quad \text{and} \quad [\epsilon \bar{\phi}_{\mathbf{n}}]_\Gamma = -\epsilon^- \nabla (G + \phi^0) \cdot \mathbf{n}|_\Gamma. \tag{22}$$

Thus the equation for the correction potential $\hat{\phi}(\mathbf{r}) = \phi(\mathbf{r}) - \bar{\phi}(\mathbf{r})$ is source-free

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla \hat{\phi}(\mathbf{r})) + \bar{\kappa}^2 \sinh(\bar{\phi}(\mathbf{r}) + \hat{\phi}(\mathbf{r})) = 0, \quad \text{in } \Omega, \tag{23}$$

and can be rewritten as

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla \hat{\phi}(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(\hat{\phi}(\mathbf{r})) = 0, \quad \text{in } \Omega. \tag{24}$$

This equation is solved with jump conditions

$$[\hat{\phi}]_\Gamma = 0 \quad \text{and} \quad [\epsilon \hat{\phi}_{\mathbf{n}}]_\Gamma = -[\epsilon \bar{\phi}_{\mathbf{n}}]_\Gamma = \epsilon^- \nabla (G + \phi^0) \cdot \mathbf{n}|_\Gamma. \tag{25}$$

The interface problem given by Eq. (24) with interface jump conditions (25) and an appropriate far-field boundary condition $\hat{\phi} = \phi_b$ on $\partial\Omega$ can be solved by the MIB scheme. Since the interface Γ is not represented through grid nodes, in calculating boundary values for the harmonic component $\phi^0(\mathbf{r})$ and interface normal derivatives for the regular component $\hat{\phi}(\mathbf{r})$, sophisticated interpolations and finite difference approximations are required in numerical implementation [21].

2.4. Two-component regularization

A new two-component regularization is proposed here, which can circumvent the work for solving the Laplace equation, i.e., Eq. (21), of the three-component method [12, 21]. Inspired by the work by Cai et al. [8], we consider a decomposition of the electrostatic potential into a Coulomb component ϕ_C and a reaction field component ϕ_{RF} with $\phi = \phi_C + \phi_{\mathrm{RF}}$. The nonlinear PB equation equation (1) can then be rewritten as

$$-\nabla \cdot (\epsilon \nabla \phi_C(\mathbf{r})) - \nabla \cdot (\epsilon \nabla \phi_{RF}(\mathbf{r})) + \bar{\kappa}^2 \sinh(\phi_C(\mathbf{r}) + \phi_{RF}(\mathbf{r})) = \rho(\mathbf{r}), \quad \text{in } \Omega. \tag{26}$$

The Coulomb potential satisfies the free space Poisson's equation with the singular charges ρ

$$\begin{cases} -\epsilon^- \Delta \phi_C(\mathbf{r}) &= \rho(\mathbf{r}) \quad \text{in } \mathbb{R}^3; \\ \phi_C(\mathbf{r}) &= 0. \qquad \text{as } |\mathbf{r}| \to \infty. \end{cases} \tag{27}$$

Thus, ϕ_C is analytically given as $\phi_C(\mathbf{r}) = G(\mathbf{r})$, where the Green's function G is defined in Eq. (20). Note that $\nabla \cdot (\epsilon \nabla \phi_C) = 0$ everywhere in Ω^- or Ω^+ except locations having partial charges, since ϵ is a piecewise constant.

Physically, the reaction field potential ϕ_{RF} can be interpreted as electrostatic field generated by the charges induced by transferring the environment surrounding the solute from ϵ^+ to ϵ^- (solvation) [8]. By subtracting (27) from (26), we have a regularized PB equation for ϕ_{RF}

$$-\nabla \cdot (\epsilon \nabla \phi_{RF}(\mathbf{r})) + \bar{\kappa}^2 \sinh(\phi_C(\mathbf{r}) + \phi_{RF}(\mathbf{r})) = 0, \quad \text{in } \Omega^- \cup \Omega^+. \tag{28}$$

Thus, ϕ_{RF} satisfies the following elliptic interface problem [11]

$$-\nabla \cdot (\epsilon^- \nabla \phi_{\mathrm{RF}}) = 0 \quad \text{in } \Omega^- \tag{29}$$

$$-\nabla \cdot (\epsilon^+ \nabla \phi_{\mathrm{RF}}) + \bar{\kappa}^2 \sinh(\phi_C + \phi_{\mathrm{RF}}) = 0 \quad \text{in } \Omega^+ \tag{30}$$

$$[\phi_{\mathrm{RF}}] = 0 \quad \text{on } \Gamma \tag{31}$$

$$\left[\epsilon \frac{\partial \phi_{\mathrm{RF}}}{\partial n}\right] = (\epsilon^- - \epsilon^+) \frac{\partial G}{\partial n} \quad \text{on } \Gamma \tag{32}$$

$$\phi_{\mathrm{RF}} = \phi_b - G \quad \text{on } \partial\Omega \tag{33}$$

However, this two-component regularization suffers several numerical difficulties. It is pointed out in [25] that the magnitudes of ϕ_C and ϕ_{RF} are much larger than that of ϕ, while ϕ_C and ϕ_{RF} have different signs. Given that ϕ_C is analytically calculated, a relatively small error in ϕ_{RF} will produce a relatively large error in ϕ. The amplifying factor is as large as $(\epsilon^+/\epsilon^- - 1)$ [25], which is about 79 for the present parameter setting in our numerical implementation by setting $\epsilon^+ = 80$ and $\epsilon^- = 1$. In addition, this formulation requires evaluating ϕ_C or G on all N grid points in Ω^+ at a computational cost $O(N^2)$, which is very expensive for large N (although multipole expansion like treecode or fast multipole method can reduce this cost to $O(N \log N)$ or $O(N)$ at the price of increased complexity in algorithm).

To overcome such numerical difficulties, Luo and his collaborators [8] proposed to solve the original potential ϕ instead of ϕ_{RF} in the domain Ω^+. Without resorting to the jump conditions, the finite difference scheme developed in [8] handles the dielectric interface by using an integral form of the PDE and discrete Green's functions, and is quite simple and effective.

In order to make the regularization approach of [8] being accessible to other finite difference or finite element methods, one has to rigorously formulate the interface jump conditions. To this end, we propose a new elliptic interface problem with discontinuous function and flux jumps for the two-component regularization. In particular, we define a regularized potential as

$$\tilde{\phi} = \begin{cases} \phi_{\mathrm{RF}} & \text{in } \Omega^- \\ \phi & \text{in } \Omega^+. \end{cases} \tag{34}$$

The jump conditions for $\tilde{\phi}$ can be derived based on those for ϕ and the definition $\phi = \phi_C + \phi_{RF}$

$$\phi^+ = \phi^-_{RF} + \phi^-_C, \quad \epsilon^+ \frac{\partial \phi^+}{\partial n} = \epsilon^- \frac{\partial \phi^-_{RF}}{\partial n} + \epsilon^- \frac{\partial \phi^-_C}{\partial n}, \quad \text{on } \Gamma. \tag{35}$$

Thus, the regularized PB equation of $\tilde{\phi}$ with corresponding interface and boundary conditions are given as

$$-\nabla \cdot (\epsilon^- \nabla \tilde{\phi}) = 0 \quad \text{in } \Omega^- \tag{36}$$

$$-\nabla \cdot (\epsilon^+ \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0 \quad \text{in } \Omega^+ \tag{37}$$

$$\left[\tilde{\phi}\right] = G \quad \text{on } \Gamma \tag{38}$$

$$\left[\epsilon \frac{\partial \tilde{\phi}}{\partial n}\right] = \epsilon^- \frac{\partial G}{\partial n} \quad \text{on } \Gamma \tag{39}$$

$$\tilde{\phi} = \phi_b \quad \text{on } \partial\Omega \tag{40}$$

Note that $\tilde{\phi}$ actually satisfies the same PB equation without the source

$$-\nabla \cdot (\epsilon \nabla \tilde{\phi}(\mathbf{r})) + \bar{\kappa}^2 \sinh(\tilde{\phi}(\mathbf{r})) = 0, \quad \text{in } \Omega^- \cup \Omega^+. \tag{41}$$

Numerically, one just needs to solve one PB interface problem given in (36) - (40). Then, the original potential is recovered as $\phi = \tilde{\phi}$ in Ω^+ and $\phi = \tilde{\phi} + G$ in Ω^-, where the Green's function G is analytically given.

The new regularization formulation could be discretized by using a finite difference or finite element method. The MIB discretization will be considered in this work, since the enforcement of nonhomogeneous jump conditions (38) and (39) does not require additional work than that for jump conditions (25) in the MIB framework. Hence, the MIB algorithm based on this new regularization is simpler than the one developed in [21]. In particular, one does not to solve an additional boundary value problem as in Eq. (21). Moreover, in jump conditions (25), the gradient of the harmonic component ϕ^0 has to be approximated numerically, which is non-trivial, because of the complex geometry of the molecular surface [21]. Whereas, the present jump conditions (38) and (39) are given analytically. Therefore, the present MIB algorithm is easier to implement. Furthermore, the proposed MIB PB solver is more efficient than the previous one [21] while still maintains the second order accuracy under the MIB scheme.

2.5. Nonlinearity

The PB equation has a nonlinear term $\bar{\kappa}^2 \sinh(\phi)$ charactering the ionic strength subject to the electrostatic potential. When the ionic strength and the electrostatic potential is weak under the physiological conditions, the linearized PB equation is more frequently used. However, in chemistry and engineering circumstances, nonlinearity needs to be taken care of. The inexact-Newton method [10, 27] developed for the original nonlinear PB equation can be simply applied to solve the regularized PB equation (41) in our two-component regularization.

The MIB discretization of the nonlinear PB equation results in an algebraic system of the form

$$F(u) = Lu + N(u) - f = 0, \tag{42}$$

where u represents a vector of all unknowns in the domain Ω, the matrix L is generated by the MIB scheme, the nonlinear term $N(u)$ is diagonal and $N_i(u) = N_i(u_i) = \bar{\kappa}^2 \sinh(u_i)$ for each i as the grid in Ω^+. Note that the discrete source f is not discretized from ρ, instead it is due to the nonhomogeneous parts of jump conditions (38) and (39) and the boundary data ϕ_b. The inexact-Newton method is quite an efficient way to solve the nonlinear system (42), whose formulation is given as

$$F'(u_n)v_n = -F(u_n) + r_n, \quad \frac{||r_n||}{||F(u_n)||} \leq \eta_n \tag{43}$$

$$u_{n+1} = u_n + v_n, \tag{44}$$

where F' is the Jacobian matrix $[\partial F_i(u)/\partial u_j]$ and takes the form $F'(u) = L + N'(u)$. Here, N' is the Jacobian matrix of $N(u)$ and is also diagonal $N'(u) = N_i'(u_i) = \bar{\kappa}^2 \cosh(u_i)$. It is easy to see that the inexact-Newton method is a two-layer iterative algorithm. The correction term v_n in outer iteration (44) is considered as a rough solution of inner iteration (43). The scheme converges linearly when η_n, the ratio of the residual r_n between the function value $F(u)$, is less than 1, and converges super-linearly as the sequence η_n has the property that $\lim_{n\to\infty} \eta_n = 0$. Compared with the traditional Newton's method, the inexact-Newton method gains efficiency by solving v_n in Eq. (43) inexactly.

2.6. Electrostatic free energy and solvation energy

As described by Sharp and Honig [44], the electrostatic free energy for a solvated biomolecule is calculated by

$$\Delta G_{\text{ele}} = \int_{\mathbb{R}^3} \left(\phi\rho + \Delta\Pi - \frac{1}{2}\epsilon|\mathbf{E}|^2 \right) d\mathbf{r} \tag{45}$$

where ϕ is the electrostatic potential, ρ is the fixed charge density represented as a smeared surface charge or as a collection of point charges, $\Delta\Pi$ is the excess osmotic pressure of the mobile ion cloud, and $\frac{1}{2}\epsilon|\mathbf{E}|^2$ is the electrostatic stress.

In our numerical validation in the next section, we report the solvation energy as

$$E_{\text{sol}} = \Delta G_{\text{ele}} - E_{\text{cou}} \approx \sum_{i=1}^{N_c} q_i \phi_{RF}(\mathbf{r}_i) \tag{46}$$

Here for simplification, we omit the energy components related to the mobile ion pressure and the electrostatic stress, whose contribution are relatively smaller but computationally more challenging as integrals of discontinuous functions. Details of evaluating the complete functional can be found in [20, 23, 44].

3. Validation

In this section, we validate our algorithm by providing simulation results. A Kirkwood sphere will be studied first, from which the analytic solution of electrostatic potential is available. Then, we report results on proteins of various sizes for the computation of electrostatic solvation free energy and binding energies. Throughout the section, we use the notation MIB to represent results from the original matched interface and boundary method with three-component regularization [10, 21] to solve the PB equation. The original MIB PB package is available online through a web server at http://weilab.math.msu.edu/MIBPB/. The new MIB method based on the two-component regularization will be denoted as the rMIB scheme. The binary versions of the rMIB algorithm on multiple platforms such as MacOS, Linux/Unix, and Windows, will be released on authors' home pages. In all computations, the length is reported with an unit Å, the electrostatic potential uses the unit of kcal/mol/e_c, and the electrostatic energy has the unit of kcal/mol.

3.1. Kirkwood sphere with analytical solutions

The PB equation has analytical solutions available only for simple geometry. Here we use analytic solution on a sphere to validate the accuracy of the proposed method in the following two cases. We computed both surface potential errors and solvation energy. To conveniently compare results from the MIB PB solver[21], the errors are reported in the form of the maximum error as

$$e_\phi = \max_{i=1}^{N_i} |\phi_{\text{true}}(x_i) - \phi_{\text{num}}(x_i)|$$

Table 1: Solving linear PB equation and Poisson equation on a spherical cavity with a centered charge: spherical radius $r = 2$, $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 1$ ($I = 9.48955$ mol/L) for PB equation and $\kappa = 0$ for Poisson equation. Centered charge is located at $(0, 0, 0)$ with unit charge e_c.

	Poisson Equation						PB Equation					
	MIB			rMIB			MIB			rMIB		
h	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}
1	4.5e-4		-81.96	4.5e-4		-81.96	6.3e-4		-82.65	6.3e-4		-82.65
1/2	7.3e-5	2.6	-81.97	7.3e-5	2.6	-81.97	1.2e-4	2.4	-82.66	1.2e-4	2.4	-82.66
1/4	2.1e-5	1.8	-81.98	2.1e-5	1.8	-81.98	3.9e-5	1.6	-82.67	3.9e-5	1.6	-82.67
1/8	5.6e-6	1.9	-81.98	5.7e-6	1.9	-81.98	1.1e-5	1.9	-82.67	1.1e-5	1.9	-82.67
1/16	1.5e-6	1.9	-81.98	1.4e-6	2.0	-81.98	2.7e-6	2.0	-82.67	2.6e-6	2.0	-82.67

where x_i are all the grid points near the surface (also called irregular points in the MIB fashion), and ϕ_{true} and ϕ_{num} are exact solution and the numerical solution by solving the PB equation. The exact solution is originally given from Kirkwood's early work for the case including the stain layer and the dielectric layer. The closed form for centric charge and spheric harmonics for the dielectric layer only are obtained in [18].

Case I: single centered charge
The first case is on a spherical cavity with one centered charge. The parameters of the case are included in the caption of table 1. Here we also reported results of the MIB PB solver [21] for comparison purpose. From table 1, we can see both methods achieved quite similar results in surface potential and solvation energy. It is shown solid 2nd order convergence in surface potential error e_ϕ. As far as the solvation energy E_{sol} is concerned, the result obtained at coarse grid (e.g. $h = 1.0$) is very close to that at the fine grid (e.g. $h = 1/16$). The newly proposed two-component regularization (rMIB) is as accurate as the original three-component regularization (MIB).

Table 2: Solving the nonlinear PB equation on a spherical cavity with a centered charge: $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 1$, and spherical radius $r = 2$. Centered charge is located at $(0, 0, 0)$ with unit charge e_c.

h	e_ϕ	ord.	E_{sol}
1	1.18e-3		-81.8928
1/2	3.23e-4	1.87	-81.9687
1/4	8.76e-5	1.88	-81.9784
1/8	2.35e-5	1.90	-81.9799
1/16	6.20e-6	1.92	-81.9801

We next investigate the capability of the current method in solving the nonlinear Poisson-Boltzmann equation. The configuration of the nonlinear PB equation on a sphere with analytical

solution is based on the reference [22] as

$$\phi(\mathbf{r}) = \begin{cases} \dfrac{1}{\varepsilon R} - \dfrac{1}{R} + \dfrac{1}{||\mathbf{r}||} & ||\mathbf{r}|| < R, \\ \dfrac{1}{\varepsilon||\mathbf{r}||} & ||\mathbf{r}|| > R. \end{cases} \tag{47}$$

$$\rho(\mathbf{r}) = \begin{cases} 4\pi\delta(\mathbf{r}) & ||\mathbf{r}|| < R, \\ \bar{\kappa}^2 \sinh(\dfrac{1}{\varepsilon||\mathbf{r}||}) & ||\mathbf{r}|| > R, \end{cases} \tag{48}$$

where $\varepsilon = \epsilon^+/\epsilon^-$ and R is the radius of the sphere. By defining the source term ρ of Eq. (1) according to Eq. (48), it can be verified that the solution in Eq. (47) satisfies Eq. (1) as well as the interface jump conditions in Eq. (2). Note that there are both singularity and non-smoothness involved in the present analytical solution. The singularity is from the source term as seen in Eq. (48) and the non-smoothness is due to the interface jump conditions in Eq. (2). Both situations could significantly reduce the accuracy of the spatial discretization numerically if they were not properly treated. In our study of the spherical case, the chosen sphere has radius $R = 2$Å with one centered charge of $1e_c$.

Similarly, the detail of the parameters is included in the caption of table 2. The configuration of the problem leads to identical solution as the linear case while the nonlinear challenge is well kept. The results in table 2 show nearly 2nd order convergence in surface potential and similar results in solvation energy, indicating the nonlinearity of the PB equation is adequately addressed.

Table 3: Solving PB equation and Poisson equation on a spherical cavity with six off-centered charges: spherical radius $r = 2$, $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 1$ for PB equation, and $\kappa = 0$ for Poisson equation. Unit charges e_c are located at (0.2,0.2,0.2), (0.5,0.5,0.5), (0.8,0.8,0.8), (-0.2,0.2,-0.2), (0.5,-0.5,0.5), (-0.8,-0.8,-0.8).

	Poisson Equation						PB Equation					
	MIB			rMIB			MIB			rMIB		
h	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}
1	3.9e-2		-3175.6	3.1e-2		-3181.4	3.8e-2		-3202.5	3.0e-2		-3209.0
1/2	1.1e-2	1.9	-3136.2	8.2e-3	1.9	-3137.7	1.1e-2	1.8	-3161.8	8.5e-3	1.8	-3163.3
1/4	1.6e-3	2.8	-3121.8	1.1e-3	2.9	-3122.1	1.7e-3	2.7	-3147.5	1.2e-3	2.9	-3147.8
1/8	4.6e-4	1.8	-3123.7	3.3e-4	1.7	-3123.7	4.8e-4	1.8	-3149.5	3.5e-4	1.7	-3149.5
1/16	9.0e-5	2.3	-3124.2	9.4e-5	1.8	-3124.2	9.8e-5	2.3	-3150.0	8.6e-5	2.0	-3150.0

Case II: six eccentric charges

The next case investigates a more complicated situation on the spherical cavity in which six eccentric charges are distributed. The analytical solution is still available but in the forms of spherical harmonics [18, 31]. The numerical results in table 3 again demonstrate the similar accuracy of both MIB and rMIB in achieving second order convergence on surface potentials. Note the solvation energy results show more variation between values from coarse grid ($h = 1, h = 1/2$) and values from fine grid ($h = 1/4$, $h = 1/8$, $h = 1/16$) due to the interaction of charge locations and the interface. A special treatment discussed in [41] can improves the accuracy reduction caused by the interface-charge interaction.

Table 4: Solving PB equation on a collection of 24 proteins with various sizes: $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 1$.

	MIB			rMIB		
PDB	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 1$	$h = 0.5$	$h = 0.25$
1ajj	-1145.01	-1140.56	-1140.84	-1137.40	-1139.48	-1140.58
2erl	-967.93	-952.80	-953.23	-965.87	-952.36	-953.06
1cbn	-309.96	-303.90	-304.11	-307.72	-303.33	-304.00
1vii	-915.63	-903.13	-904.52	-910.68	-902.31	-903.62
1fca	-1219.89	-1205.04	-1205.69	-1216.82	-1204.44	-1205.51
1bbl	-998.34	-988.56	-989.32	-994.16	-988.40	-989.09
2pde	-826.65	-822.92	-823.19	-818.01	-820.97	-822.88
1sh1	-757.41	-755.11	-754.80	-749.60	-753.99	-754.50
1vjw	-1254.76	-1241.67	-1242.26	-1251.02	-1241.07	-1242.05
1uxc	-1159.02	-1140.56	-1141.76	-1154.21	-1139.25	-1141.43
1ptq	-897.99	-874.38	-874.96	-893.86	-873.32	-874.71
1bor	-859.47	-854.41	-855.23	-854.12	-853.47	-855.01
1fxd	-3331.79	-3322.19	-3322.67	-3326.78	-3321.39	-3322.43
1r69	-1111.76	-1089.52	-1090.71	-1106.88	-1088.62	-1090.43
1mbg	-1372.82	-1354.04	-1354.91	-1370.14	-1353.31	-1354.68
1bpi	-1326.42	-1305.13	-1306.56	-1322.33	-1304.37	-1306.30
1hpt	-827.18	-812.49	-814.62	-825.00	-811.56	-814.37
451c	-1030.99	-1027.21	-1027.79	-1024.07	-1025.66	-1027.45
1svr	-1743.19	-1712.55	-1713.71	-1733.97	-1711.11	-1713.28
1frd	-2892.23	-2864.26	-2866.77	-2886.99	-2862.50	-2866.30
1a2s	-1933.93	-1922.23	-1923.12	-1929.74	-1921.20	-1923.23
1neq	-1767.05	-1732.89	-1735.18	-1759.10	-1731.71	-1734.75
1a63	-2419.16	-2377.39	-2378.95	-2406.88	-2374.41	-2378.26
1a7m	-2196.79	-2162.70	-2164.20	-2181.92	-2160.34	-2163.61

3.2. Electrostatics on Proteins

Next we solved the PB equations on proteins, which can be used to validate the accuracy and efficiency of the method on complex geometry and change distribution. For simulation on proteins, the molecular surface is generated by MSMS [43] and the partial charges for each atoms are assigned by CHARMM22 force field [37].

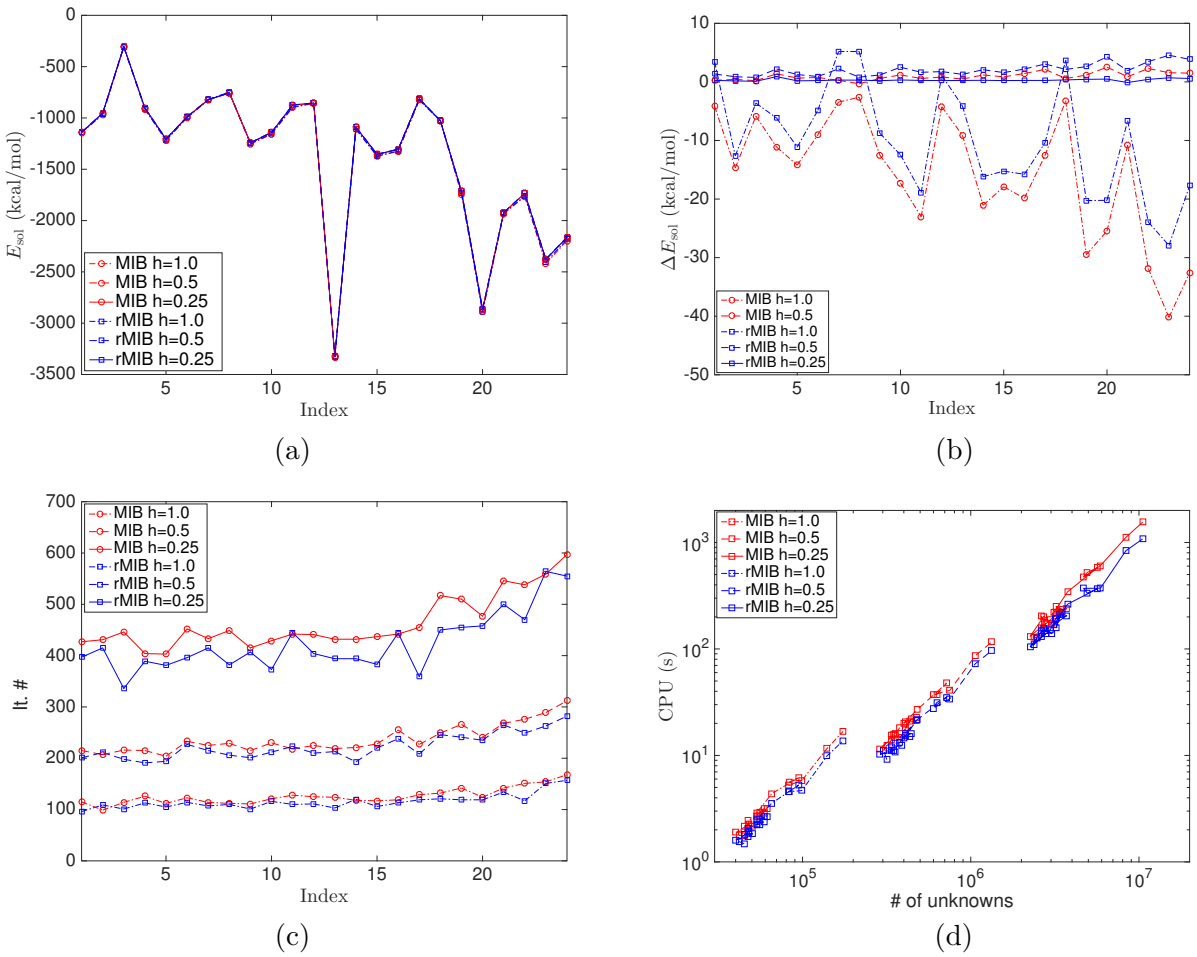


Figure 2: Solving PB equation on a collection of 24 proteins with various sizes: $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 1$: (a) solvation energy; (b) differences of salvation energy adjusted by solvation energy computed with MIB at fine grid $h = 0.25$; (c) number of iteration for the bi-conjugate iterative solver; (d) CPU time vs number of unknowns.

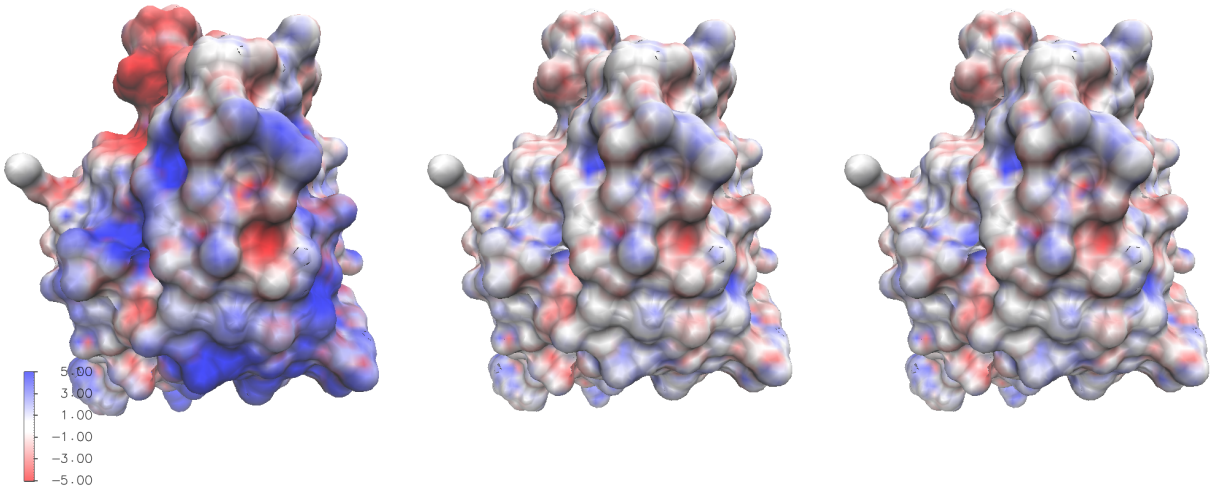
Case 1: Solvation energy computation for 24 proteins

For tests on proteins we first solve PB equation and compute solvation energy on a collection of 24 proteins as in [21]. The purpose of this test is to compare the accuracy and efficiency between MIB and rMIB PB solvers. The point we try to make is that rMIB obtain the same level of accuracy as MIB while it has improved efficiency since the work for solving a boundary value problem and some related interpolations is circumvented.

The solvation energies of the 24 proteins computed by MIB and rMIB at different mesh sizes are reported in Table 4 ordered by increased radius of gyration. With a glimpse, we could see that the results from both methods are consistent in two aspects: (1) at finest grid $h = 0.25$, the values from both methods are almost identical; (2) for both methods, when h varies from 1.0, 0.5, to 0.25, the results converge to the finest mash $h = 0.25$.

To better illustrate the consistency in accuracy of both methods. We plot all the solvation energies in Fig. 2(a). It can be seen that for both methods at various meshes, the values are very close to each other. To further show the difference of both methods at various meshes, we use the result of MIB at $h = 0.25$ as reference to compute the difference of the solvation energies by subtracting the reference. The results are shown in Fig. 2(b). In this figure, we can see result of rMIB at $h = 0.25$ (solid blue line with square) is almost the same as the reference thus it looks like a horizontal line across 0 values. Both methods converge to the result at the finest mesh $h = 0.25$. Result from rMIB at $h = 1.0$ (blue dashdot with square) is slightly better than that from MIB at $h = 1.0$ (red dashdot with circle), while result from MIB at $h = 0.5$ (red dash with circle) is slightly better than that from rMIB at $h = 0.5$ (red dash with square). Thus as far as accuracy is concerned, MIB and rMIB are similar.

We recorded the number of unknowns, number of iterations, and CPU time for all simulations. To save space, we did not list all the values in table but plot the results in two sub-figures. In Fig. 2(c), the number of iteration in solving the regularized PB equation is depicted for both methods and 24 proteins. It can be seen that solving PB equation with rMIB method has less number of iterations than with MIB method. Fig. 2(d) shows the CPU time comparison against number of unknowns for all proteins and all mesh sizes. For both MIB methods, the CPU time increases with respect to the number of unknowns, with a constant slope in loglog scale. We can also see that result from rMIB at various mesh size has obviously shortened CPU time compared with that from MIB. Quantitatively, rMIB achieved approximately 15-20% CPU time reduction while maintain the same level of accuracy. It worths mentioning the numerical implementation of rMIB is much more convenient than that of MIB.



(a) (b) (c)

Figure 3: Surface potential plotting of antibacterial protein RegIII lectin (4mth) after solving PB equations; $\kappa = 1$ for PB equations, $\epsilon^- = 1$, $\epsilon^+ = 80$; (a) Poisson equation, (b) linear PB equation, (c) nonlinear PB equation

Case 2: Surface potential study of protein 4mth [40]

This test case studies the surface potential of a protein after solving the PB equation. The protein we select here is RegIII lectin (PDB: 4mth) reported by Mukherjee et al. [40]. RegIII lectins, as antibacterial proteins, recognize their bacterial targets by binding peptidoglycan carbohydrate. The publication [40] revealed that the mechanistic basis for RegIII bactericidal activity is to bind membrane phospholipids and kill bacteria by forming a hexameric membrane-permeabilizing oligomeric pore.

Figure 3 shows the surface potential of RegIII lectin by solving Poisson equation (a), linear PB equation (b), and Nonlinear PB equation (c). Among all three sub-plots, we can see that Fig. 3(a) has the much more obvious red-blue contrast than that from Fig. 3(b)(c) because Poisson model does not have any screening effects caused by mobile ions while the PB model does. Figures 3(b) and 3(c) are almost the same because the electrostatic potential is small for the given κ. Note the $\sinh\phi \approx \phi$ for small ϕ and the boundary condition as given in Eq. (3) leads to small value of ϕ, and the Boltzmann term has nonzero values only outside the biomolecule. For reference, the electrostatic solvation energies are -2130.7835, -2161.6229, -2161.6231 kcal/mol for Poisson, Linear PB, and Nonlinear PB equations respectively. The fact that the energies did not vary very much explains why the surface potentials in Fig. 3(a-c) are similar in pattern.

Case 3: Binding energy of BIV Tat Protein and BIV TAR RNA (PDB: 2a9x) [33]
We use this case to demonstrate the ability of rMIB to compute the binding energy between biomolecules. The biomolecules we choose are the BIV TAR RNA and BIV Tat Protein [33] in HIV Viral replication. The binding of the protein to the RNA plays a significant role in viral transcription and antiviral drug-design seek drug like molecule to inhibit the RNA-protein interaction. The work conducted by Leeper et al. [33] demonstrated that the cyclic peptide bound BIV TAR RNA with an affinity comparable to that of the RNA-binding domain of the Tat protein and inhibited protein binding to the RNA. Here we can use rMIB to compute the binding energy between the protein and RNA thus potentially provide important reference for the cyclic peptide related drug design.

We computed the binding energy between BIV Tat Protein and BIV TAR RNA at different mesh size as reported in Table 5. The detail of the configuration is included in the caption of the table. From the table, we can see the solvation energy and finally the binding energy computed are very consistent at different meshes, which indicate even at mesh size as coarse as $h = 1$, the numerical results are quite reliable. We also plot the surface potential as in the previous test case for the involved biomolecules in Fig. 4. Figure 4(a) and (b) are surface potential of RNA and protein respectively, and (c) is the surface potential of the complex. The helical structure of RNA can still be seen in (a) even on the molecular surface. Figure (d) is the opposite side of the protein, which binds to the RNA. The potential plot shows complimentary electrostatics (red-blue) and geometry (concave up-down).

Table 5: solvation energy and binding energy of 2a9x: $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 0.1257$ ($I = 0.15$ mol/L); the Coulombic energy of the complex, protein, and RNA are $E_{\text{coul}}^{\text{complex}} = -30530.79$kcal/mol, $E_{\text{coul}}^{\text{protein}} = -6008.01$kcal/mol, and $E_{\text{coul}}^{\text{RNA}} = -20807.43$ kcal/mol respectively.

h	E_{sol} (complex)	E_{sol} (protein)	E_{sol} (DNA)	ΔE_{sol}	ΔE_{bind}
1	-5816.38	-1021.94	-8893.39	4098.94	383.60
1/2	-5821.22	-1025.86	-8898.54	4103.18	387.84
1/4	-5823.39	-1026.27	-8900.52	4103.40	388.05

4. Conclusion

Inspired by the original idea of charge singularity regularization from [21] and [8], the present work provides a novel regularization scheme, which combines the efficiency of two-component regularization approaches with the accuracy of three-component methods. With singular component being analytically represented by the Green's function, a new matched interface and boundary (MIB) finite difference algorithm is developed for solving both linear and nonlinear PB equations.

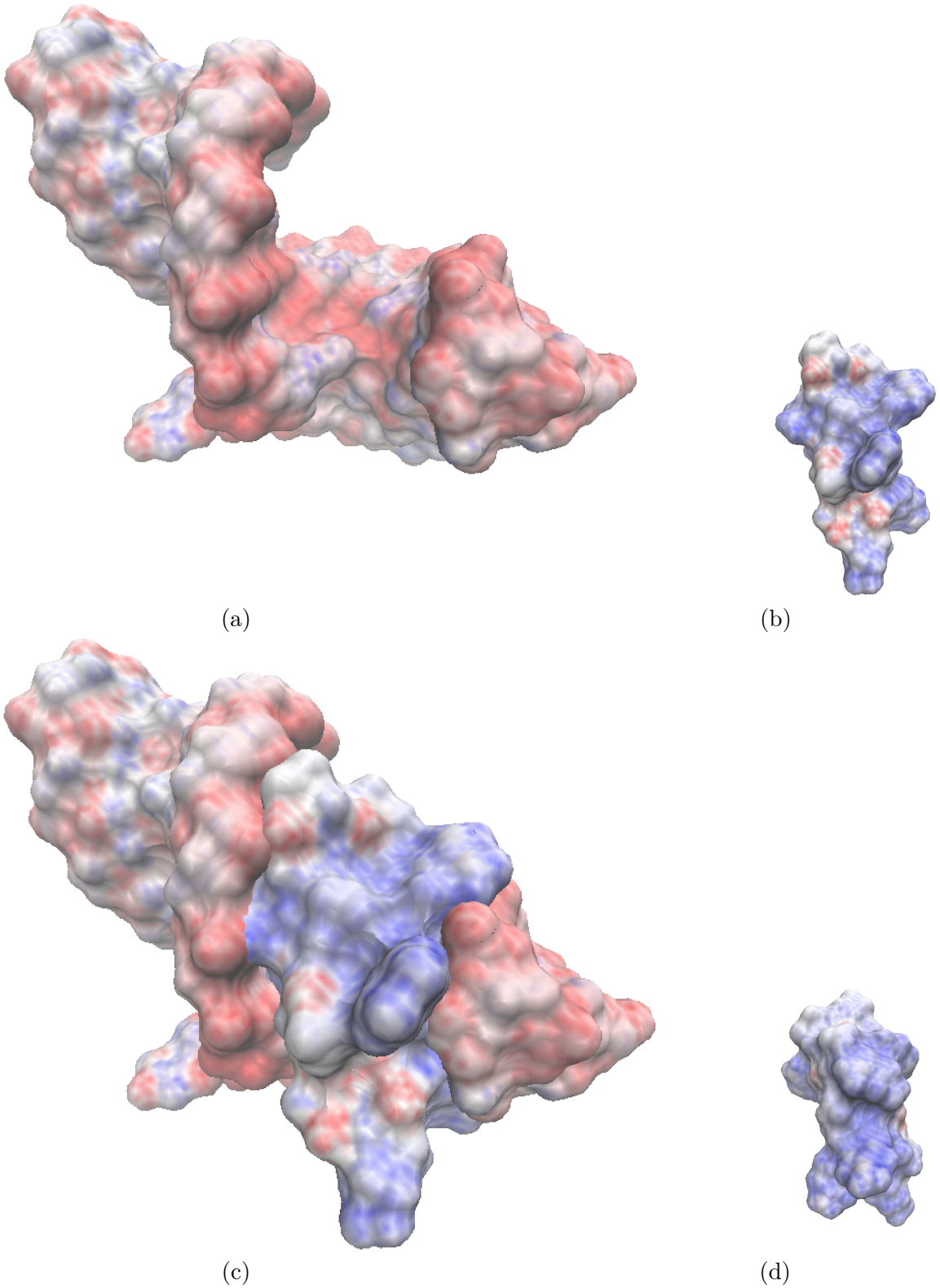


Figure 4: surface potential plot of the 2a9x: (a) RNA segment; (b) protein; (c) the RNA segment bound with the protein; (d) the opposite side (contacting surface in binding) of the protein.

Compared with the existing MIB Poisson-Boltzmann (PB) solver based on a three-component regularization, the present algorithm is simpler, easier to implement, and faster, while maintains the same second order accuracy. In particular, our new algorithm circumvents the work of solving a boundary value Poisson equation inside the molecular interface and its related interface jump conditions computation for the regularized PB equation. Our numerical results demonstrate the solid second order accuracy as well as the improved efficiency on Kirkwood sphere case in which the analytical solutions are available. We used the newly developed two-component MIB PB solver to compute the solvation energy of a series of 24 proteins, resolve the electrostatic surface potential of a RegIII lectin protein, and calculate the binding energy between BIV Tat Protein and BIV TAR RNA. These numerical results further validate the present method.

In comparison with the existing regularization approaches [8, 11, 12, 21, 25, 45, 53], our new scheme maintains the same accuracy level as the commonly used three-component methods [12, 21], while enjoying the efficiency of two-component methods [8, 11, 53] by solving only one partial differential equation (PDE). A subtle issue here is that the proposed approach defines a boundary value problem (BVP) with a discontinuous solution across the solute-solvent boundary, whereas the existing three-component methods [12, 21, 25, 45] have continuous solutions. Numerically, a discontinuous BVP is not necessarily more difficult to solve. This actually depends on the underlying spacial discretization method. If a finite difference interface method, such as the MIB scheme, is used, a discontinuous BVP can be simply handled as in a continuous BVP case. This is essentially why the present two-component rMIB method outperforms the three-component MIB method [21] in all aspects.

However, if a finite element method (FEM) is adopted for spatial discretization, the situation could be different. For solving a discontinuous BVP, some advanced FEMs, such as discontinuous Galerkin (DG) [15], have to be employed. Being well studied in the literature, the DG method has advantages over the conventional FEM for elliptic interface problems [38, 39], but may require more degree of freedoms in computation. If one prefers to the continuous FEM, the three-component regularizations [12, 21, 25, 45] are more suitable than the proposed two-component method. In particular, the regularization scheme of Ref. [45] could be a better choice, because of a continuous BVP and special designs for FEM.

In the future, we will take advantage of the reduced complexity to further attack the challenging problem of computing the electrostatic field and forces for molecular dynamics simulation under the MIB framework [20].

Acknowledgments

Geng's research was supported in part by the National Science Foundation (NSF) grants DMS-1318898 and DMS-1418957. Zhao's research was supported in part by the NSF grant DMS-1318898 and Simons foundation grant 524151.

References

[1] Bajaj, C., Chen, S.-C., and Rand, A. An efficient higher-order fast multipole boundary element solution for poissonboltzmann-based molecular electrostatics. *SIAM Journal on Scientific Computing 33*, 2 (2011), 826–848.

[2] Baker, N., Holst, M., and Wang, F. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II. Refinement at solvent-accessible surfaces in biomolecular systems. *Journal of Computational Chemistry 21*, 15 (2000), 1343–1352.

[3] Baker, N. A. Poisson-Boltzmann methods for biomolecular electrostatics. *Methods in Enzymology 383* (2004), 94–118.

[4] Baker, N. A., Sept, D., Joseph, S., Holst, M. J., and McCammon, J. A. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences of the United States of America 98*, 18 (2001), 10037–10041.

[5] Beard, D. A., and Schlick, T. Modeling salt-mediated electrostatics of macromolecules: the discrete surface charge optimization algorithm and its application to the nucleosome. *Biopolymers 58* (2001), 106–115.

[6] Boschitsch, A. H., and Fenley, M. O. A new outer boundary formulation and energy corrections for the nonlinear Poisson-Boltzmann equation. *Journal of Computational Chemistry 28*, 5 (2007), 909–21.

[7] Boschitsch, A. H., Fenley, M. O., and Zhou, H.-X. Fast boundary element method for the linear Poisson-Boltzmann equation. *The Journal of Physical Chemistry B 106*, 10 (2002), 2741–2754.

[8] Cai, Q., Wang, J., Zhao, H.-K., and Luo, R. On removal of charge singularity in poissonboltzmann equation. *The Journal of Chemical Physics 130*, 14 (2009).

[9] Callenberg, K. M., Choudhary, O. P., de Forest, G. L., Gohara, D. W., Baker, N. A., and Grabe, M. Apbsmem: A graphical interface for electrostatic calculations at the membrane. *PLoS ONE 5*, 9 (09 2010), 1–12.

[10] Chen, D., Chen, Z., Chen, C., Geng, W. H., and Wei, G. W. MIBPB: A software package for electrostatic analysis. *J. Comput. Chem. 32* (2011), 657 – 670.

[11] Chen, L., Holst, M., and Xu, J. The finite element approximation of the nonlinear poisson-boltzmann equation. *SIAM Journal of Numerical Analysis 45* (2007), 2298–2320.

[12] Chern, I. L., Liu, J.-G., and Weng, W.-C. Accurate evaluation of electrostatics for macromolecules in solution. *Methods and Applications of Analysis 10*, 2 (2003), 309–28.

[13] Davis, M. E., Madura, J. D., Sines, J., Luty, B. A., Allison, S. A., and McCammon, J. A. Diffusion-controlled enzymatic reactions. *Methods in Enzymology 202* (1991), 473–497.

[14] Davis, M. E., and McCammon, J. A. Electrostatics in biomolecular structure and dynamics. *Chemical Reviews 94* (1990), 509–21.

[15] Deng, W., Zhufu, X., Xu, J., and Zhao, S. A new discontinuous galerkin method for the nonlinear poisson-boltzmann equation. *Applied Mathematics Letters 257* (2015), 1000–1021.

[16] Feig, M., and Brooks III, C. L. Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Curr Opin Struct Biol. 14* (2004), 217 – 224.

[17] Fogolari, F., Brigo, A., and Molinari, H. The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology. *Journal of Molecular Recognition 15*, 6 (2002), 377–92.

[18] Geng, W. A boundary integral poisson-boltzmann solvers package for solvated bimolecular simulations. *Molecular Based Mathematical Biology 3* (2015), 43–58.

[19] Geng, W., and Krasny, R. A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules. *Journal of Computational Physics 247*, 0 (2013), 62 – 78.

[20] Geng, W., and Wei, G. W. Multiscale molecular dynamics using the matched interface and boundary method. *J Comput. Phys. 230*, 2 (2011), 435–457.

[21] Geng, W., Yu, S., and Wei, G. W. Treatment of charge singularities in implicit solvent models. *Journal of Chemical Physics 127* (2007), 114106.

[22] Geng, W., and Zhao, S. Fully implicit ADI schemes for solving nonlinear Poisson-Boltzmann equation. *Molecular Based Mathematical Biology 1* (2013), 109–123.

[23] Gilson, M. K., Davis, M. E., Luty, B. A., and McCammon, J. A. Computation of electrostatic forces on solvated molecules using the Poisson-Boltzmann equation. *Journal of Physical Chemistry 97*, 14 (1993), 3591–3600.

[24] Greengard, L., Gueyffier, D., Martinsson, P.-G., and Rokhlin, V. Fast direct solvers for integral equations in complex three-dimensional domains. *Acta Numerica 18* (005 2009), 243–275.

[25] Holst, M., McCammon, J., Yu, Z., Zhou, Y., and Zhu, Y. Adaptive finite element modeling techniques for the poisson-boltzmann equation. *Communication in Computational Physics 11* (2012), 179–214.

[26] Holst, M. J. *The Poisson-Boltzmann Equation: Analysis and Multilevel Numerical Solution.* PhD thesis, UIUC, 1994.

[27] Holst, M. J., and Saied, F. Numerical solution of the nonlinear poissonboltzmann equation: Developing more robust and efficient methods. *Journal of Computational Chemistry 16*, 3 (1995), 337–364.

[28] Honig, B., and Nicholls, A. Classical electrostatics in biology and chemistry. *Science 268*, 5214 (1995), 1144–9.

[29] Im, W., Beglov, D., and Roux, B. Continuum solvation model: electrostatic forces from numerical solutions to the Poisson-Boltzmann equation. *Computer Physics Communications 111*, 1-3 (1998), 59–75.

[30] Juffer, A., E., B., van Keulen, B., van der Ploeg, A., and Berendsen, H. The electric potential of a macromolecule in a solvent: a fundamental approach. *J. Comput. Phys. 97* (1991), 144–171.

[31] Kirkwood, J. G. Theory of solution of molecules containing widely separated charges with special application to zwitterions. *J. Comput. Phys. 7* (1934), 351 – 361.

[32] Klapper, I., Hagstrom, R., Fine, R., Sharp, K., and Honig, B. Focussing of electric fields in the active site of cu-zn superoxide dismutase: Effects of ionic strength and amino acid modification. *Protein 1* (1986), 47 – 59.

[33] Leeper, T. C., Athanassiou, Z., Dias, R. L. A., Robinson, J. A., and Varani, G. Tar rna recognition by a cyclic peptidomimetic of tat protein. *Biochemistry 44*, 37 (2005), 12362–12372. PMID: 16156649.

[34] Liang, J., and Subranmaniam, S. Computation of molecular electrostatics with boundary element methods. *Biophys. J. 73* (1997), 1830–1841.

[35] Lu, B., Cheng, X., and McCammon, J. A. new-version-fast-multipole-method accelerated electrostatic calculations in biomolecular systems. *Journal of Computational Physics 226*, 2 (2007), 1348 – 1366.

[36] Luo, R., David, L., and Gilson, M. K. Accelerated Poisson-Boltzmann calculations for static and dynamic systems. *Journal of Computational Chemistry 23*, 13 (2002), 1244–53.

[37] MacKerell, A. D., Jr., Bashford, D., Bellott, M., R. L. Dunbrack, J., Evanseck, J. D., Field, M. J., Fischer, S., Gao, J., Guo, H., Ha, S., Joseph-McCarthy, D., Kuchnir, L., Kuczera, K., Lau, F. T. K., Mattos, C., Michnick, S., Ngo, T., Nguyen, D. T., Prodhom, B., Reiher, W. E., Roux, B., Schlenkrich, M., Smith, J. C., Stote, R., Straub, J., Watanabe, M., Wirkiewicz-Kuczera, J., Yin, D., and Karplus, M. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B 102*, 18 (1998), 3586–3616. PMID: 24889800.

[38] Mu, L., Wang, J., Wei, G., Ye, X., and Zhao, S. Weak galerkin methods for second order elliptic interface problems. *Journal of computational physics 250* (2013), 106–125.

[39] Mu, L., Wang, J., Ye, X., and Zhao, S. A new weak galerkin finite element method for elliptic interface problems. *Journal of Computational Physics 325* (2016), 157–173.

[40] Mukherjee, S., Zheng, H., Derebe, M. G., Callenberg, K. M., Partch, C. L., Rollins, D., Propheter, D. C., Rizo, J., Grabe, M., Jiang, Q.-X., and Hooper, L. V. Antibacterial membrane attack by a pore-forming intestinal C-type lectin. *Nature 505*, 7481 (01 2014), 103–107.

[41] Nguyen, D. D., Wang, B., and Wei, G.-W. Accurate, robust and reliable calculations of poisson-boltzmann binding energies, (2016). *Journal of Computational Chemistry* (2017).

[42] Roux, B., and Simonson, T. Implicit solvent models. *Biophysical Chemistry 78*, 1-2 (1999), 1–20.

[43] Sanner, M. F., Olson, A. J., and Spehner, J. C. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers 38* (1996), 305–320.

[44] Sharp, K. A., and Honig, B. Calculating total electrostatic energies with the nonlinear Poisson-Boltzmann equation. *Journal of Physical Chemistry 94* (1990), 7684–7692.

[45] Xie, D. New solution decomposition and minimization schemes for poisson-boltzmann equation in calculation of biomolecular electrostatics. *Journal of Computational Physics 275* (2014), 294–309.

[46] Yu, S., Geng, W., and Wei, G. W. Treatment of geometric singularities in implicit solvent models. *Journal of Chemical Physics 126* (2007), 244108.

[47] Yu, S. N., and Wei, G. W. Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities. *J. Comput. Phys. 227* (2007), 602–632.

[48] Zhang, B., Lu, B., Cheng, X., Huang, J., Pitsianis, N. P., Sun, X., and McCammon, J. A. Mathematical and numerical aspects of the adaptive fast multipole poisson-boltzmann solver. *Communications in Computational Physics 13*, 1 (001 2013), 107–128.

[49] Zhang, Z., Witham, S., and Alexov, E. On the role of electrostatics in proteinprotein interactions. *Physical Biology 8*, 3 (2011), 035001.

[50] Zhao, S., and Wei, G. W. High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces. *J. Comput. Phys. 200*, 1 (2004), 60–103.

[51] Zhou, Y. C., and Wei, G. W. On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method. *J. Comput. Phys. 219*, 1 (2006), 228–246.

[52] Zhou, Y. C., Zhao, S., Feig, M., and Wei, G. W. High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *J. Comput. Phys. 213*, 1 (2006), 1–30.

[53] Zhou, Z., Payne, P., Vasquez, M., Kuhn, N., and Levitt, M. Finite-difference solution of the Poisson-Boltzmann equation: complete elimination of self-energy. *Journal of Computational Chemistry 17* (1996), 1344–1351.