A Kernel-free Boundary Integral Method for the Nonlinear Poisson-Boltzmann Equation

Wenjun Ying [*]

Department of Mathematics, MOE-LSC and Institute of Natural Sciences, Shanghai Jiao Tong University, Minhang, Shanghai 200240, P. R. China.

Abstract

This work proposes a boundary integral formulation based Cartesian grid method for the nonlinear Poisson-Boltzmann (PB) interface problem in biophysics. The method (a) does not have the limitation associated with the standard finite difference method for complex interfaces, (b) avoids generation of any unstructured volume or surface grids needed by the finite element method, and (c) solves the nonlinear and variable coefficient PDE in the framework of boundary integral equations. The method solves the nonlinear PB equation with the Newton iterative method. It first reformulates the linearized PB equation in each Newton iteration as a Fredholm system of two boundary integral (BI) equations of the second kind, which is well-conditioned, and then solves the BI system with a Krylov subspace method. The evaluation of boundary and volume integrals involved in the solution of the BI system is done with a kernel-free method, which does not need to know the Green's function associated with the variable coefficient PB equation. The kernel-free method evaluates volume and boundary integrals by solving equivalent simple interface problems on Cartesian grids with either a fast Fourier transform based Poisson solver or a geometric multigrid preconditioned conjugate gradient solver, either of which involves computational work essentially linearly proportional to the number of nodes on the Cartesian grid. Numerical examples in both two and three space dimensions are presented to demonstrate the efficiency and accuracy of the proposed numerical method.

Key words: Poisson-Boltzmann equation, boundary integral method, kernel-free, Newton method, interface problem, Cartesian grid method

[*] Corresponding author.
Email address: `wying@sjtu.edu.cn` (Wenjun Ying).

1 Introduction

The Poisson-Boltzmann (PB) equation appears in many applications [14,22,25,27,38,39,41,54,57,66,68,70,75], which is known as the Gouy-Chapman theory in electrochemistry [14,27] and known as the Poisson-Boltzmann theory in biophysics [22,38]. In biophysics applications, the Poisson-Boltzmann equation models a solvated biomolecular system by dielectrically distinct regions with singular charges distributed in the molecular region. The Poisson-Boltzmann equation takes the form

$$\nabla \cdot \Big(\epsilon \nabla u\Big) - \chi(\mathbf{p}) \sum_{j=1}^{J} c_j q_j e^{-\beta q_j u} = -\sum_{i=1}^{I} q_i \delta(\mathbf{p} - \mathbf{p}_i). \tag{1}$$

Here, $\mathbf{p}$ is the space variable (a vector of space coordinates), $u = u(\mathbf{p})$ is the electrostatic potential, ϵ is a space-dependent and piecewise constant dielectric coefficient, the characteristic function vanishes ($\chi(\mathbf{p}) = 0$) in the molecule region (impenetrable to ions) and takes value one ($\chi(\mathbf{p}) = 1$) in the solvent region, c_j is the bulk density of the j^{th} mobile ion species with charge q_j, $\beta = 1/(\kappa_{\mathrm{B}} T)$ with κ_{B} be the Boltzmann constant and T be the absolute temperature, q_i is the singular charge located at point $\mathbf{p}_i$ within the solute region. For symmetric $1:1$ salt, the Poisson-Boltzmann equation reads

$$\nabla \cdot \Big(\epsilon \nabla u\Big) - \chi(\mathbf{p}) \kappa^2 \sinh(u) = -\sum_{i=1}^{I} q_i \delta(\mathbf{p} - \mathbf{p}_i), \tag{2}$$

where the reaction coefficient κ absorbs all of the related parameters. Subject to continuity conditions on the dielectric interface between the solute (molecule) and solvent regions, the PB model is often proposed as an interface problem.

Over the past decades, a large amount of numerical solution techniques for the PB interface problem have been developed [1,26,52,65]. We refer the interested readers to the review paper on numerical methods for the PB equation by Lu et. al. [51]. Due to the nonlinearity of the PB equation, the heterogeneity of the interface problem, which has different equations on different sides of the dielectric interface, and particularly the geometry complexity of the molecule surface, which may consist of tens of thousand to millions of atoms, nowadays it is still a challenging and difficult task to efficiently and accurately solve the PB interface problem even though after years of research.

Finite difference method [17,18,21,46,53,60,64,83,84], finite element method [5,16,19,20,33–37,69,74] and boundary element/integral method [9–11,28,47–50,81,82] are the three most widely used numerical methods for the PB equation.

Finite difference method is the most popular numerical method for the PB equation due to its simplicity in implementation. The solution by the standard finite difference method usually has low order accuracy due to its inflexibility with the complex geometry of the molecular surface (the dielectric interface) and its less accurate treatment for the interface conditions. To achieve high order accuracy for the PB equation, a few non-standard finite difference methods [18, 31, 46, 58, 59, 64, 83, 84] have been developed to take into account the complex molecule surface or the discontinuity of the potential flux across the dielectric interface. However, the coefficient matrix of the discrete system by the non-standard finite difference methods [18, 83, 84] is often non-symmetric and even indefinite, which leads to inefficient solution of the discrete equations. In Li et. al.'s work [46], which solves a boundary value problem of the nonlinear PB equation in two space dimensions, the linear constant coefficient PB equation appeared in a quasi-Newton iteration for the nonlinear PB equation is discretized with a finite difference interface method [44, 45] and the coefficient matrix of the resulting system is symmetric and positive definite, the same as that obtained by the standard finite difference method for the equation without boundary or interface. The linear system of discrete equations in Li et. al. [46] is solved with a fast Fourier transform (FFT) based fast elliptic solver while it is unclear how to extend the approach there for the nonlinear PB interface problem.

Finite element method gains popularity due to its geometry flexibility with the complicated molecule surface. Compared to other methods, finite element method may provide more rigorous convergence analysis [16], which makes it possible to develop a rigorous adaptive and local mesh refinement algorithm. However, the generation of body-fitted unstructured (quality) grids around the complicated molecule surface needed by the finite element method, especially in three space dimensions, is usually a difficult, expensive and time-consuming process. Even though there are already well-developed techniques and software packages for grid generation [13, 24], the computer time spent for the process is not negligible and should be taken into account when we develop a solver for the PB equation with complicated molecule surfaces. Another point that we would like to emphasize on is that the discrete finite element equations for the PB problem on unstructured grids in general can not be solved with an FFT or geometric multigrid based fast elliptic solver [12]. Instead, the system can be solved at best with an algebraic multigrid iterative method [33–37], which is usually much less efficient and less robust than FFT and geometric multigrid methods.

Boundary element/integral method may be theoretically regarded as the most efficient numerical method for the linear constant coefficient PB equation. For problems without involving volume integrals, the boundary element method reduces the dimension of the problem by one. Accelerated by fast matrix-vector multiplication algorithms [2, 3, 6, 28, 29, 48, 61–63, 73], the boundary element/integral method may involve computational work only essentially (up to a logarithm factor) linearly proportional to the number of unknowns on the dielectric interface. However, since there is no Green's function directly available for the nonlinear PB equation, the traditional boundary element/integral method is only applied to the linear constant coefficient PB equation. In addition, the computational work will increase by at least one order when volume integrals appear in the formulation. The boundary element/integral method involves singular and hyper-singular boundary integrals, improper evaluation of which affects the accuracy and stability of the method.

In this work, we will solve the nonlinear PB interface problem with the Newton method. We solve the linearized PB equation in the Newton iteration with a generalized boundary integral method, the so-called kernel-free boundary integral (KFBI) method [78–80]. The KFBI method does not need to know the kernels of boundary and volume integrals and the Green's function of the nonlinear PB equation. It evaluates both boundary and volume integrals in the same way, involving computational work essentially linearly proportional to the number of unknowns in the domain (instead of on the interface). The KFBI method accurately computes singular and hyper-singular boundary integrals appeared in the boundary integral formulation for the PB interface problem.

The KFBI method [78] as a generalization of the Cartesian grid-based evaluation method for boundary integrals by Mayo [55, 56] was proposed by Ying et. al. [78, 80] for variable coefficients elliptic partial differential equations. The KFBI method evaluates a boundary or volume integral by first solving an equivalent simple interface problem on a Cartesian grid and then interpolating the discrete solution on the grid to get values of the boundary or volume integral at points of interest on the interface/boundary of the problem. As the discretization of the equivalent interface problem and the interpolation of the grid-based solution for values on the interface take into account the geometry complexity and the discontinuity of the solution and/or its derivatives, the solution by the KFBI method has high order accuracy.

To solve the linearized PB interface problem in the Newton iteration with the KFBI method, we first reformulate the problem as a system of two boundary integral equations, which is well-conditioned as a Fredholm system of the second kind. We represent the dielectric interface by its intersection with the grid lines of an underlying Cartesian grid and discretize the boundary integral equations at the intersection points, following a Nyström-like approach [4]. This representation of the interface and discretization of the system avoid generation of any unstructured volume or surface grids. We solve the discrete boundary integral system with a Krylov subspace method, the generalized minimal residual (GMRES) method [67].

In each matrix-vector multiplication during the Krylov subspace iteration, we need to evaluate four boundary integrals, a single layer boundary integral, a double layer boundary integral, an adjoint double layer boundary integral and a hyper-singular boundary integral. The single layer and adjoint double layer boundary integrals are simultaneously evaluated by solving an equivalent interface problem, which is associated with the (linearized) partial differential operator on the solvent and solved with a full V-cycle geometric multigrid preconditioned conjugate gradient method. The double layer and hyper-singular boundary integrals are simultaneously evaluated by solving another equivalent interface problem, which is associated with the partial differential (in fact the Laplacian) operator on the solute (molecule) and solved with an FFT based Poisson solver. That is, each matrix-vector multiplication in the Krylov subspace iteration involves solution of two simple interface problems. Besides, we also need to evaluate the boundary and volume integrals appeared on the right hand side of the boundary integral system with the KFBI method but only need to evaluate them once for each solution of the system. As the discrete boundary integral system is well-conditioned, the number of Krylov subspace iterations is essentially independent of the system dimension or number of unknowns on the dielectric interface. Note that the Newton iteration is also independent of the system dimension. The overall computational work involved with the method for the nonlinear PB interface problem is essentially linearly proportional to the number of unknowns on the Cartesian grid that is used for solving the equivalent interface problems.

The remainder of this paper is organized as follows. In Section 2, we present the nonlinear Poisson-Boltzmann interface problem to be solved and describe its linearization by the Newton method. In Section 3, we reformulate the linearized Poisson-Boltzmann interface problem as a system of two boundary integral equations, which is to be solved with a Krylov subspace iterative method. In Section 4, we briefly describe the kernel-free evaluation method for boundary and volume integrals involved in the Krylov subspace iteration for the boundary integral system. In Section 5, we summarize the Newton-KFBI algorithm for the nonlinear PB interface problem. In Section 6, we present numerical examples in both two and three space dimensions to demonstrate the accuracy and efficiency of the method. Finally, in Section 7, we discuss on the advantages and possible further improvement or extension of the kernel-free boundary integral method for the PB equation or other physically more realistic (modified) PB models.

2 The Nonlinear Poisson-Boltzmann Equation

In this section, we will present the model Poisson-Boltzmann equation to be solved. For simplicity, we only consider the PB equation for the symmetric $1:1$ salt. We assume the region occupied by the solute (molecule) and solvent

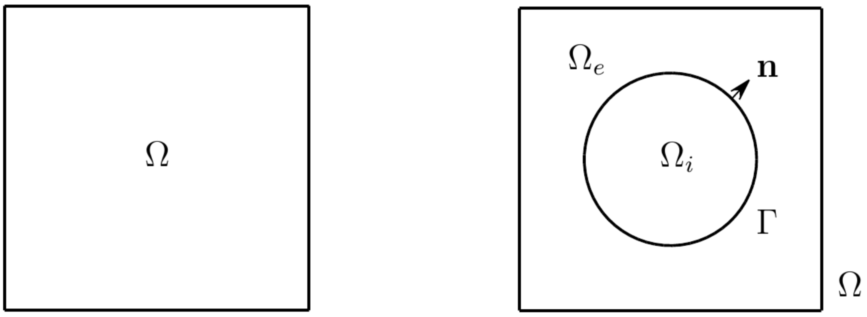

Fig. 1. A rectangular domain Ω separated by an interface Γ into two subdomains, the solute (molecule) region Ω_i and the solvent region Ω_e

is a rectangle.

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be the rectangle domain, which is the union of regions for the solute and solvent. Denote by Γ the dielectric interface (molecule surface) between the solute and solvent regions. Assume the boundary $\partial\Omega$ of the rectangle and the dielectric interface Γ have no intersection, i.e., $\Gamma \cap \partial\Omega = \emptyset$. Let Ω_i be the solute (molecule) region, which is the interior domain enclosed by the dielectric interface Γ. Denote by $\Omega_e \equiv \Omega \setminus \bar{\Omega}_i$ the solvent region, which is the complement of $\bar{\Omega}_i$ (the closure of Ω_i) in Ω. Let $\mathbf{n}$ be the unit outward normal, pointing from Ω_i to Ω_e, on the dielectric interface Γ. See Fig. 1 for an illustration of the domains Ω, $\Omega_{i,e}$ and the dielectric interface Γ.

Let $u_i = u_i(\mathbf{p})$ and $u_e = u_e(\mathbf{p})$ be the electrostatic potentials, defined in Ω_i and Ω_e, respectively. The nonlinear Poisson-Boltzmann equation for symmetric $1:1$ salt reads

$$\epsilon_i \triangle u_i = \rho_i \qquad \text{in } \Omega_i, \tag{3}$$

$$\epsilon_e \triangle u_e - \kappa^2 \sinh(u_e) = \rho_e \qquad \text{in } \Omega_e, \tag{4}$$

subject to the interface conditions

$$u_i - u_e = g \qquad \text{on } \Gamma, \tag{5}$$

$$\epsilon_i\, \partial_{\mathbf{n}} u_i - \epsilon_e\, \partial_{\mathbf{n}} u_e = J \qquad \text{on } \Gamma, \tag{6}$$

and the boundary condition

$$u_e = 0 \qquad \text{on } \partial\Omega. \tag{7}$$

Here, ϵ_i and ϵ_e are the dielectric constants; $\kappa > 0$ is a constant reaction coefficient; ρ_i and ρ_e are the charge sources; g and J are known functions defined on Γ only; $\partial_{\mathbf{n}} u_i$ and $\partial_{\mathbf{n}} u_e$ represent the normal derivatives of the unknown functions $u_i(\mathbf{p})$ and $u_e(\mathbf{p})$.

We regard the nonlinear Poisson-Boltzmann interface problem (3)-(7) as a regularized version of the classic one for (2), where both the potential and the flux are continuous across the interface (i.e., $g = 0$, $J = 0$) but the functions ρ_i and ρ_e may contain singular sources such as the Dirac delta functions, corresponding to point charges. In the regularized Poisson-Boltzmann interface problem (3)-(7), we assume the functions $\rho_i = \rho_i(\mathbf{p})$ and $\rho_e = \rho_e(\mathbf{p})$ are smooth (regular) functions defined on Ω_i and Ω_e, respectively, while the jumps of the potential and the flux are non-zero (i.e., $g \neq 0$ and $J \neq 0$). We refer to [17] for the regularization technique of the Poisson-Boltzmann equation.

We solve the nonlinear PB interface problem iteratively with the standard Newton method. For integer $m \geq 0$, given an approximate solution u_i^m and u_e^m, we look for a hopefully better approximate solution u_i^{m+1} and u_e^{m+1} by solving the linearized Poisson-Boltzmann equation

$$\epsilon_i \triangle u_i^{m+1} = \rho_i \quad \text{in } \Omega_i, \tag{8}$$

$$\epsilon_e \triangle u_e^{m+1} - \kappa^2 \cosh(u_e^m)\, u_e^{m+1} = \rho_e + \kappa^2 \sinh(u_e^m) - \kappa^2 \cosh(u_e^m)\, u_e^m \quad \text{in } \Omega_e, \tag{9}$$

subject to the interface conditions

$$u_i^{m+1} - u_e^{m+1} = g \qquad \text{on } \Gamma, \tag{10}$$

$$\epsilon_i\, \partial_{\mathbf{n}} u_i^{m+1} - \epsilon_e\, \partial_{\mathbf{n}} u_e^{m+1} = J \qquad \text{on } \Gamma, \tag{11}$$

and the boundary condition

$$u_e^{m+1} = 0 \qquad \text{on } \partial\Omega. \tag{12}$$

For $m = 0, 1, 2, \cdots$, we make the Newton iteration and terminate it when the difference between $u_{i,e}^m$ and $u_{i,e}^{m+1}$ is sufficiently small.

3 Boundary Integral Equation Formulation

In this section, we will reformulate the linearized PB interface problem (8)-(12) in the m^{th} Newton iteration as a system of two boundary integral equations.

First, we assume the approximate solution u_e^m has continuous extension onto Ω_i, i.e., $u_e^m = u_e^m(\mathbf{p})$ is continuous on $\bar{\Omega} = \bar{\Omega}_i \cup \bar{\Omega}_e$. We introduce two Green's functions $G = G(\mathbf{q}; \mathbf{p})$ and $K = K(\mathbf{q}; \mathbf{p})$ that satisfy

$$\begin{aligned} \triangle G(\mathbf{q}; \mathbf{p}) &= \delta(\mathbf{q} - \mathbf{p}), \qquad \mathbf{q} \in \Omega, \\ G(\mathbf{q}; \mathbf{p}) &= 0, \qquad \mathbf{q} \in \partial\Omega, \end{aligned}$$

and

$$\begin{aligned} \epsilon_e \triangle K(\mathbf{q};\mathbf{p}) - \kappa^2 \cosh(u_e^m)\, K(\mathbf{q};\mathbf{p}) = \epsilon_e \delta(\mathbf{q}-\mathbf{p}), \qquad & \mathbf{q} \in \Omega, \\ K(\mathbf{q};\mathbf{p}) = 0, \qquad & \mathbf{q} \in \partial\Omega, \end{aligned}$$

for each $\mathbf{p} \in \Omega$. Analytical expression of the Green's function $G(\mathbf{q};\mathbf{p})$ is not directly available and in general there is no closed form of the Green's function $K(\mathbf{q};\mathbf{p})$. We also note that the Green's function $K(\mathbf{q};\mathbf{p})$ varies during the Newton iteration. It depends on the approximate solution u_e^m and the index m. For conciseness, we omit the dependency of the Green's function $K(\mathbf{q};\mathbf{p})$.

Let $f_i = \rho_i/\epsilon_i$ and $f_e^m = \Big[\rho_e + \kappa^2 \sinh(u_e^m) - \kappa^2 \cosh(u_e^m)\, u_e^m\Big]/\epsilon_e$. We decompose the linearized PB interface problem (8)-(12) into a Dirichlet boundary value problem (BVP) and a Neumann boundary value problem. The Dirichlet BVP reads

$$\begin{cases} \triangle u_i^{m+1} = f_i & \text{in } \Omega_i \\ u_i^{m+1} = g + u_e^{m+1} & \text{on } \Gamma \end{cases} \tag{13}$$

and the Neumann BVP reads

$$\begin{cases} \epsilon_e \triangle u_e^{m+1} - \kappa^2 \cosh(u_e^m)\, u_e^{m+1} = \epsilon_e f_e^m & \text{in } \Omega_e \\ \epsilon_e \partial_{\mathbf{n}} u_e^{m+1} = \epsilon_i\, \partial_{\mathbf{n}} u_i^{m+1} - J & \text{on } \Gamma \\ u_e^{m+1} = 0 & \text{on } \partial\Omega \end{cases} . \tag{14}$$

In terms of the Green's function $G(\mathbf{q};\mathbf{p})$, we represent the solution $u_i^{m+1}(\mathbf{p})$ to the Dirichlet boundary value problem (13) as the sum of an interior volume integral and a double layer boundary integral

$$u_i^{m+1}(\mathbf{p}) = \int_{\Omega_i} G(\mathbf{q};\mathbf{p}) f_i(\mathbf{q})\, d\mathbf{q} + \int_\Gamma \frac{\partial G(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_q}} \varphi^{m+1}(\mathbf{q})\, ds_{\mathbf{q}} \quad \text{for } \ \mathbf{p} \in \Omega_i, \tag{15}$$

with the density $\varphi^{m+1}(\mathbf{p})$ satisfying

$$\begin{aligned} & \frac{1}{2}\varphi^{m+1}(\mathbf{p}) + \int_\Gamma \frac{\partial G(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_q}} \varphi^{m+1}(\mathbf{q})\, ds_{\mathbf{q}} \\ &= g + u_e^{m+1} - \int_{\Omega_i} G(\mathbf{q};\mathbf{p}) f_i(\mathbf{q})\, d\mathbf{q} \qquad \text{for } \ \mathbf{p} \in \Gamma. \end{aligned} \tag{16}$$

In terms of the Green's function $K(\mathbf{q};\mathbf{p})$, we represent the solution $u_e^{m+1}(\mathbf{p})$ to the Neumann boundary value problem (14) as the sum of an exterior volume integral and a single layer boundary integral

$$u_e^{m+1}(\mathbf{p}) = \int_{\Omega_e} K(\mathbf{q};\mathbf{p}) f_e^m(\mathbf{q})\, d\mathbf{q} - \int_\Gamma K(\mathbf{q};\mathbf{p}) \psi^{m+1}(\mathbf{q})\, ds_{\mathbf{q}} \quad \text{for } \ \mathbf{p} \in \Omega_e, \tag{17}$$

with the density $\psi^{m+1}(\mathbf{p})$ satisfying

$$-\frac{1}{2}\,\psi^{m+1}(\mathbf{p}) - \int_\Gamma \frac{\partial K(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_p}}\psi^{m+1}(\mathbf{q})\,ds_{\mathbf{q}}$$
$$= \mu\,\frac{\partial u_i^{m+1}}{\partial \mathbf{n_p}} - J/\epsilon_e - \frac{\partial}{\partial \mathbf{n_p}}\int_{\Omega_e} K(\mathbf{q};\mathbf{p})f_e^m(\mathbf{q})\,d\mathbf{q} \quad \text{for} \quad \mathbf{p}\in\Gamma. \tag{18}$$

Here, $\mu = \epsilon_i/\epsilon_e$.

We remark that the solution u_e^{m+1} to the Neumann BVP is well-defined and continuous in the rectangular domain Ω since the volume integral and the single layer boundary integral are continuous on $\bar{\Omega}$. As long as we choose the initial guess u_e^0 to be continuous on Ω (say, $u_e^0 = 0$), the approximate solution u_e^m always has continuous extension onto Ω_i. So we guarantee that the continuity assumption on u_e^m is satisfied all the time.

For functions $\psi(\mathbf{q})$ and $\varphi(\mathbf{q})$ defined on the interface Γ, let

$$L_e\psi \equiv \int_\Gamma K(\mathbf{q};\mathbf{p})\psi(\mathbf{q})\,ds_{\mathbf{q}}, \qquad M_i\varphi \equiv \int_\Gamma \frac{\partial G(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_q}}\varphi(\mathbf{q})\,ds_{\mathbf{q}}$$

be the single layer and double layer boundary integrals,

$$M_e^*\psi \equiv \int_\Gamma \frac{\partial K(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_p}}\psi(\mathbf{q})\,ds_{\mathbf{q}}, \qquad N_i\varphi \equiv \frac{\partial}{\partial \mathbf{n_p}}\int_\Gamma \frac{\partial G(\mathbf{q};\mathbf{p})}{\partial \mathbf{n_q}}\varphi(\mathbf{q})\,ds_{\mathbf{q}}$$

be the adjoint double layer and hyper-singular boundary integrals, and

$$\mathcal{G}f_i \equiv \int_{\Omega_i} G(\mathbf{q};\mathbf{p})f_i(\mathbf{q})\,d\mathbf{q}, \qquad \mathcal{K}f_e^m \equiv \int_{\Omega_e} K(\mathbf{q};\mathbf{p})f_e^m(\mathbf{q})\,d\mathbf{q}.$$

be the interior and exterior volume integrals, respectively. Here, $\mathbf{n_p}$ denotes the unit outward normal vector at point $\mathbf{p}\in\Gamma$.

After substituting (15) into (18) and (17) into (16) and using the symbols introduced above, we write the system of boundary integral equations (16) and (18) as

$$\frac{1}{2}\,\varphi^{m+1} + M_i\varphi^{m+1} + L_e\psi^{m+1} = g + \mathcal{K}f_e^m - \mathcal{G}f_i, \tag{19}$$

$$\mu N_i\varphi^{m+1} + \frac{1}{2}\psi^{m+1} + M_e^*\psi^{m+1} = J/\epsilon_e + \partial_{\mathbf{n_p}}(\mathcal{K}f_e^m - \mu\,\mathcal{G}f_i). \tag{20}$$

In matrix-vector notation, the system reads

$$\begin{bmatrix} 1/2 + M_i & L_e \\ \mu N_i & 1/2 + M_e^* \end{bmatrix}\begin{bmatrix} \varphi^{m+1} \\ \psi^{m+1} \end{bmatrix} = \begin{bmatrix} g + \mathcal{K}f_e^m - \mathcal{G}f_i \\ J/\epsilon_e + \partial_{\mathbf{n_p}}(\mathcal{K}f_e^m - \mu\,\mathcal{G}f_i) \end{bmatrix}. \tag{21}$$

It is a Fredholm boundary integral system of the second kind. We may solve the corresponding discrete system after discretization, which is well-conditioned, with a Krylov subspace iterative method such as the generalized minimal residual (GMRES) method [67].

After solving the system (21) of boundary integral equations for the unknown densities φ^{m+1} and ψ^{m+1}, we may further get the solution to the linearized Poisson-Boltzmann interface problem (8)-(12) by

$$u_i^{m+1}(\mathbf{p}) = \mathcal{G} f_i + M_i \varphi^{m+1} \qquad \text{in } \Omega_i, \tag{22}$$

$$u_e^{m+1}(\mathbf{p}) = \mathcal{K} f_e^m - L_e \psi^{m+1} \qquad \text{in } \Omega_e. \tag{23}$$

4 The Kernel-Free Boundary Integral Method

Assume the boundary integral system (21) is solved with a Krylov subspace method [67]. We know the Green's functions are difficult to calculate or at least not directly available. A question now arising is how to evaluate the boundary and volume integrals encountered in computing the right hand side of the system and the matrix-vector multiplication during the Krylov subspace iteration.

We will evaluate the boundary and volume integrals with a kernel-free boundary integral (KFBI) method [78–80]. The KFBI method does not need to know any analytical form of the integral kernels and the Green's functions associated with the partial differential equations. Even it does not compute any approximation of the Green's functions. Instead, it only approximately computes the action of the Green's functions or integral kernels on the interface density or volume source functions.

The kernel-free boundary integral method replaces the evaluation of a boundary or volume integral by the solution of an equivalent simple interface problem, which is much easier to solve than the heterogeneous interface problem (8)-(12). The interface problem is discretized on a Cartesian grid of the rectangular domain and solved by an efficient method such as a fast Fourier transform or geometric multigrid based elliptic solver. After the simple interface problem on the Cartesian grid is solved, approximate values of the corresponding boundary or volume integral at discretization points of the interface are obtained by polynomial interpolation.

The interior volume integral $v_i = \mathcal{G} f_i$ is the solution to the following interface problem

$$\begin{cases} \triangle v_i = \begin{cases} f_i & \text{in } \Omega_i \\ 0 & \text{in } \Omega_e \end{cases}, \\ [v_i] = 0 \quad \text{on } \Gamma, \\ [\partial_{\mathbf{n}} v_i] = 0 \quad \text{on } \Gamma, \\ v_i = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{24}$$

In this work, a quantity with a square bracket such as $[v_i]$ and $[\partial_{\mathbf{n}} v_i]$ denotes the jump of the corresponding function across the interface, which equals the inside limit minus the outside limit. The exterior volume integral $v_e = \mathcal{K} f_e^m$ is the solution to the following interface problem

$$\begin{cases} \epsilon_e \triangle v_e - \kappa^2 \cosh(u_e^m) v_e = \begin{cases} 0 & \text{in } \Omega_i \\ \epsilon_e f_e^m & \text{in } \Omega_e \end{cases}, \\ [v_e] = 0 \quad \text{on } \Gamma, \\ [\partial_{\mathbf{n}} v_e] = 0 \quad \text{on } \Gamma, \\ v_e = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{25}$$

The double layer boundary integral $w_i = M\varphi^{m+1}$ is the solution to the following interface problem

$$\begin{cases} \triangle w_i = 0 \quad \text{on } \Omega \setminus \Gamma, \\ [w_i] = \varphi^{m+1} \quad \text{on } \Gamma, \\ [\partial_{\mathbf{n}} w_i] = 0 \quad \text{on } \Gamma, \\ w_i = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{26}$$

The single layer boundary integral $w_e = -L_e \psi^{m+1}$ is the solution to the following interface problem

$$\begin{cases} \epsilon_e \triangle w_e - \kappa^2 \cosh(u_e^m) w_e = 0 \quad \text{on } \Omega \setminus \Gamma, \\ [w_e] = 0 \quad \text{on } \Gamma, \\ [\partial_{\mathbf{n}} w_e] = \psi^{m+1} \quad \text{on } \Gamma, \\ w_e = 0 \quad \text{on } \partial\Omega. \end{cases} \tag{27}$$

Suppose the rectangular domain Ω is partitioned into a uniform Cartesian grid as shown in Figure 2 (a). Discretization of each interface problem above on the uniform Cartesian grid by the standard five-point finite difference method leads to a linear system of discrete equations [71]. To take into account the discontinuity of the solution or its derivatives on the interface Γ, only the right hand side of the linear system need to be corrected at irregular grid nodes, where finite difference stencils go across the interface and the local truncation errors of the finite difference discretization are large otherwise. As the coefficient matrix of the linear system is unchanged and the same as the one without the interface or without any discontinuity of the solution or its derivatives, the discrete interface equations can be solved very efficiently. We solve the discrete systems for the interface problems (24) and (26) with a fast Fourier transform (FFT) based Poisson solver and solve the discrete systems for the interface problems (25) and (27) with a full V-cycle geometric multigrid preconditioned conjugate gradient (PCG) iterative solver [12]. Each solver finds the solution to a discrete interface problem with computational work essentially (up to a logarithm factor for the FFT based Poisson solver) linearly proportional to the number of nodes on the Cartesian grid that covers the domain Ω.

Once we have values of a boundary or volume integral at the nodes of a Cartesian grid by solving its equivalent interface problem, we can obtain its value at any point on the interface by polynomial interpolation, which though needs to take into account the discontinuity of the integral or its derivatives across the interface. As a matter of fact, we can get not only the values of the volume and boundary integrals but also their normal derivatives on the interface, which include the hyper-singular boundary integral such as $N_i\varphi^{m+1}$ and the adjoint double layer boundary integral such as $M_e^*\psi^{m+1}$. To compute the normal derivative of a boundary or volume integral, we first interpolate the discrete solution on the Cartesian grid to get the first order partial derivatives and then compute the inner product of the gradient with a unit outward normal vector on the interface as the normal derivative. This is another advantage of the KFBI method over the traditional boundary integral method as it does not need to make any special treatment (such as the singularity subtraction and integral regularization [15, 72]) for the nearly singular or hyper-singular boundary integrals.

To numerically solve the boundary integral system (21), we need to discretize the dielectric interface Γ as well as the boundary integral system (21).

As for the traditional boundary integral method [4], we may discretize the interface Γ by a set of quasi-uniformly spaced points on it and correspondingly discretize the boundary integral system (21) at the points, following a Nyström-like approach for boundary integral equations. The interface discretization can be easily done in two space dimensions. But in three space dimensions it may still be a difficult or expensive process to find a set of quasi-uniformly spaced points on the molecule surface, which may be geomet-

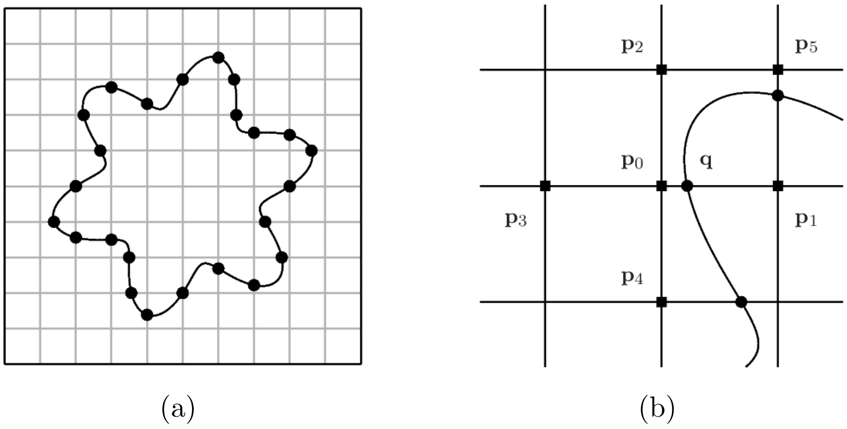

Fig. 2. Discretization of the domain Ω into a uniform Cartesian grid: (a) interface discretization by its intersection points with the Cartesian grid lines, (b) a six-point interpolation stencil $\{\mathbf{p}_k\}_{k=0}^{5}$ for computing the value of a boundary or volume integral at the intersection point $\mathbf{q}$.

rically very complicated.

To avoid the difficult or expensive generation process for an unstructured interface grid in three space dimensions, we represent and discretize the interface Γ by its intersection points with the underlying Cartesian grid that is used for solving the discrete interface problems; see Fig. 2 (a) for an illustration. Note we can always easily find all intersection points of the interface with a Cartesian grid. Even though the intersection points are not quasi-uniform, after it is combined with an equilibrating process, the representation is convenient and stable for interface interpolation and differentiation as the projection of the intersection points onto one coordinate axis or plane are locally uniform [79], which makes it much easier to find interpolation stencils than other approaches [23,43,45]. In fact, interpolation with data on locally uniform points in general yields more accurate results.

We also discretize the boundary integral system (21) at the intersection points of the interface with the Cartesian grid. Different from the standard Nyström approach [4], we do not get values of the boundary and volume integrals with a numerical quadrature and instead interpolate the discrete solution on the Cartesian grid to get their values at the discretization points of the interface.

In addition to the interface representation and boundary integral system discretization, the KFBI method has three basic components: 1) calculation for jumps of the first few partial derivatives across the interface; 2) correction of the discrete interface equations at irregular grid nodes; 3) interpolation of the Cartesian grid based solution to discretization points of the boundary integral equation. Among the three components, the first is the most fundamental one. In terms of the jumps of partial derivatives, we can correct the right hand side of the discrete interface equations, which leads to small local truncation errors of the system at irregular grid nodes. In terms of the jumps of partial derivatives, we can interpolate the discrete solution on the Cartesian grid to get high order accurate values of the corresponding boundary or volume integral at discretization points of the interface.

Appendix A has some details on the computation for jumps of the first few partial derivatives of the solution to a unified interface problem in both two and three space dimensions. For details on the interface and boundary integral system discretization, the correction and solution of the discrete interface problems and the local polynomial interpolation, we refer the interested readers to [79, 80].

5 Algorithm Summary

We solve the nonlinear PB interface problem (3)-(7) with the Newton-KFBI method. We initialize the solutions u_i^0 and u_e^0 for the Newton iteration with zeros and solve the linearized PB interface problem (8)-(12) in the Newton iteration with the KFBI method. We represent the interface by its intersection points with an underlying Cartesian grid and discretize the boundary integral system (21) at the intersection points. We solve the discrete boundary integral equations for the unknown densities ψ^{m+1} and φ^{m+1} with the GMRES method. We always initialize the unknown densities for the GMRES iteration with zero and stop the iteration when the residual in the discrete ℓ^2-norm is less than a pre-specified absolute tolerance tol_{gmres}. In the KFBI evaluation for the boundary integrals $M_i\varphi^{m+1}$, $N_i\varphi^{m+1}$ and the interior volume integral $\mathcal{G}_i f_i$, an FFT-based fast Poisson solver is applied for solving the discrete Poisson equations on the Cartesian grid whose intersection points with the interface are used to represent the interface and discretize the boundary integral system. In the KFBI evaluation for the boundary integrals $L_e\psi^{m+1}$, $M_e^*\psi^{m+1}$ and the exterior volume integral $\mathcal{K}_e f_e^m$, a full V-cycle geometric multigrid preconditioned conjugate gradient (PCG) iteration is applied for solving the discrete interface problems. We terminate the PCG iteration when the residual in the maximum norm is less than a pre-specified absolute tolerance $\text{tol}_{multigrid}$. After the unknown densities ψ^{m+1} and φ^{m+1} are obtained, we compute the approximate solution u_i^{m+1} and u_e^{m+1} by (22)-(23). We stop the Newton iteration when the difference between the approximate solutions $(u_i^m,\, u_e^m)$ and $(u_i^{m+1},\, u_e^{m+1})$ in the discrete maximum norm is less than a pre-specified tolerance tol_{newton}.

6 Numerical Results

In this section, we present a few numerical examples for the nonlinear Poisson-Boltzmann interface problem with the Newton-KFBI method, which was implemented in custom codes written in the C++ computer language. The numerical experiments were all performed in double precision on a computer equipped with Intel(R) Xeon(R) 2.93GHz CPU.

In all examples, we fix the dielectric constants to be $\epsilon_i = 1$ and $\epsilon_e = 4$; and fix the reaction coefficient $\kappa = 1$. For examples in two space dimensions, we choose the domain Ω to be the square rectangle $\Omega = (-1, 1) \times (-1, 1)$ and choose the functions $\rho_i(\mathbf{p})$, $\rho_e(\mathbf{p})$, $g(\mathbf{p})$ and $J(\mathbf{p})$ so that the exact solutions to the nonlinear PB problem (3)-(7) read

$$\begin{cases} u_i(\mathbf{p}) = u_i(x, y) = e^{0.6x+0.8y} \qquad \text{for} \;\; \mathbf{p} = (x, y) \in \Omega_i \\ u_e(\mathbf{p}) = u_e(x, y) = \sin(\pi x)\sin(\pi y) \qquad \text{for} \;\; \mathbf{p} = (x, y) \in \Omega_e \end{cases}.$$

For examples in three space dimensions, we choose the domain Ω to be the rectangle $\Omega = (-1, 1) \times (-1, 1) \times (-1, 1)$ and choose the functions $\rho_i(\mathbf{p})$, $\rho_e(\mathbf{p})$, $g(\mathbf{p})$ and $J(\mathbf{p})$ so that the exact solutions to the nonlinear PB problem (3)-(7) read

$$\begin{cases} u_i(\mathbf{p}) = u_i(x, y, z) = e^{x-0.6y+0.8z} \qquad \text{for} \;\; \mathbf{p} = (x, y, z) \in \Omega_i \\ u_e(\mathbf{p}) = u_e(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z) \qquad \text{for} \;\; \mathbf{p} = (x, y, z) \in \Omega_e \end{cases}.$$

Each example has a different interface Γ in two or three space dimensions. In this work, we assume the interface in all examples is implicitly given as the zero level set of a smooth function.

In the solution of the nonlinear PB interface problem with the Newton-KFBI method, the tolerances for the Newton, GMRES and multigrid PCG iterations are fixed to be $\text{tol}_{newton} = 10^{-8}$, $\text{tol}_{gmres} = 10^{-8}$ and $\text{tol}_{multigrid} = 10^{-10}$. The tolerances are all absolute tolerances. In the numerical experiments, the initial guess for each of the Newton, GMRES and multigrid PCG iterations involved with the Newton-KFBI method is always set to be zero.

Numerical results for the examples are listed in Tables 1-5. In each table, the first column has the grid size, the second column shows the number of Newton iterations, the third column shows the maximum number of GMRES iterations needed for each boundary integral system (21) during the Newton iteration, the fourth column shows the maximum number of multigrid PCG iterations for the simple interface problems encountered during the GMRES iteration for (21), the fifth column shows the error of the numerical solution

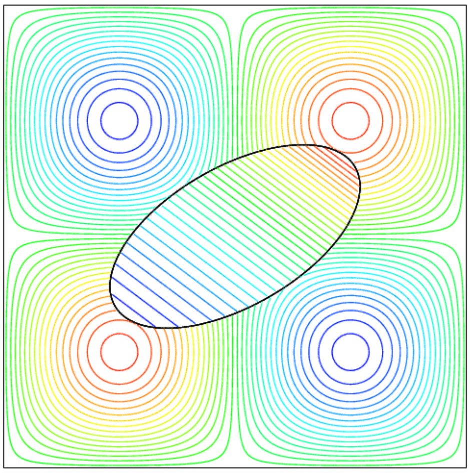

Fig. 3. Isolines of a numerical solution around the rotated ellipse (**Example 1**)

Table 1
Numerical results of **Example 1** in two space dimensions

grid size	#NEWTON	#GMRES	#PCG	$\|e_h\|_\infty$	CPU (sec)
128×128	5	14	6	2.07E-4	8.50E-1
256×256	5	14	6	5.06E-5	3.49E+0
512×512	5	14	6	1.27E-5	1.87E+1
1024×1024	5	14	6	3.25E-6	8.86E+1

in the discrete maximum norm and the last column has the CPU times in seconds.

Example 1. The interface Γ is the ellipse centered at the origin with radii $a = 1$ and $b = 1/2$, rotated by 30 degrees counter-clockwisely. This is, the interface Γ is given by

$$\Gamma = \Big\{ (x, y) \in \mathbb{R}^2 \,:\, (x \cos\theta + y \sin\theta)^2 + (-x \sin\theta + y \cos\theta)^2 - 1 = 0 \Big\},$$

with $\theta = \pi/6$. Numerical results of this example are listed in Table 1. Fig. 3 shows by isolines a numerical solution to the PB interface problem .

Example 2. The interface Γ is a two-ovals curve given by

$$\Gamma = \{ (x, y) \in \mathbb{R}^2 \,:\, (x^2 + y^2)^2 - 2a^2(x^2 - y^2) + a^4 - b^4 = 0 \}$$

with $a = 0.505$ and $b = 0.5$. Numerical results of this example are listed in Table 2. Fig. 4 shows by isolines a numerical solution to the PB interface problem .

Example 3. The interface Γ is a three-fold star-shaped curve in two space

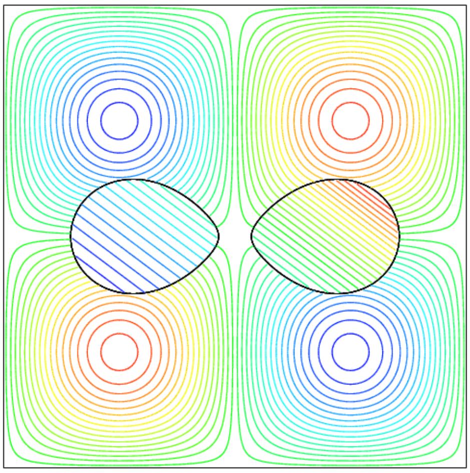

Fig. 4. Isolines of a numerical solution around the two-ovals curve (**Example 2**)

Table 2
Numerical results of **Example 2** in two space dimensions

grid size	#NEWTON	#GMRES	#PCG	$\|e_h\|_\infty$	CPU (sec)
128 × 128	4	17	6	1.90E-4	8.50E-1
256 × 256	4	16	6	4.73E-5	3.09E+0
512 × 512	4	15	6	1.18E-5	1.53E+1
1024 × 1024	4	15	6	2.93E-6	7.43E+1

Table 3
Numerical results of **Example 3** in two space dimensions

grid size	#NEWTON	#GMRES	#PCG	$\|e_h\|_\infty$	CPU (sec)
256 × 256	4	18	6	4.63E-5	4.05E+0
512 × 512	4	18	6	1.14E-5	1.93E+1
1024 × 1024	4	18	6	2.99E-6	9.37E+1
2048 × 2048	4	18	5	9.29E-7	3.80E+2

dimensions given by

$$\Gamma = \left\{ \mathbf{p} \in \mathbb{R}^2 \ : \ c - \sum_{i=1}^{3} \exp\left\{ -\frac{|\mathbf{p} - \mathbf{p}_i|^2}{r_i^2} \right\} = 0 \right\}.$$

Here, the centers $\mathbf{p}_1 = (1, -\sqrt{3}/3)\, a$, $\mathbf{p}_2 = (-1\,, -\sqrt{3}/3)\, a$, $\mathbf{p}_3 = (0\,, 2\sqrt{3}/3\,)\, a$ and the radii $r_i = a$ for $i = 1,\ 2,\ 3$ with $a = 0.3$ and $c = 0.75$. Numerical results of this example are listed in Table 3. Fig. 5 shows by isolines a numerical solution to the PB interface problem .

Example 4. The interface Γ is an ellipsoid in three space dimensions given

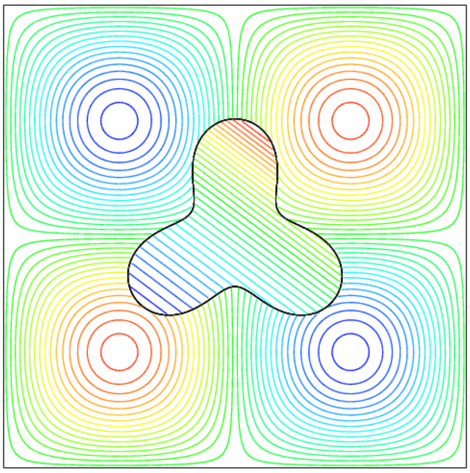

Fig. 5. Isolines of a numerical solution around the star-shaped curve (**Example 3**)

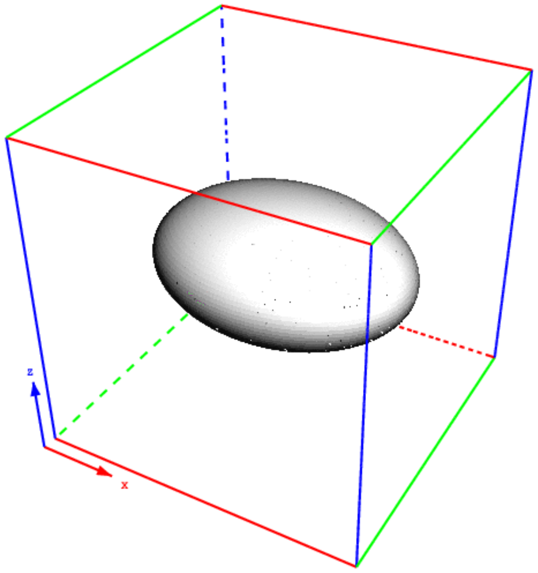

Fig. 6. The solute region is an ellipsoid (**Example 4**).

by

$$\Gamma = \Big\{ (x, y, z) \in \mathbb{R}^3 \, : \, \frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} - 1 = 0 \Big\},$$

with $a = 0.8$, $b = 0.6$ and $c = 0.4$. Fig. 6 shows the ellipsoid interface in the rectangle Ω. Numerical results of this example are listed in Table 4.

Example 5. The solute region Ω_i is a four-atoms molecule in three space dimensions given by

$$\Gamma = \Bigg\{ \mathbf{p} \in \mathbb{R}^3 \, : \, c - \sum_{i=1}^{4} \exp\Bigg\{ -\frac{|\mathbf{p} - \mathbf{p}_i|^2}{r_i^2} \Bigg\} = 0 \Bigg\}.$$

Here, the centers $\mathbf{p}_1 = (2\sqrt{3}/3 \, , 0 \, , -\sqrt{6}/6 \,) \, a$, $\mathbf{p}_2 = (-\sqrt{3}/3 \, , 1 \, , -\sqrt{6}/6 \,) \, a$,

Table 4
Numerical results of **Example 4** in three space dimensions

grid size	#NEWTON	#GMRES	#PCG	$\|e_h\|_\infty$	CPU (sec)
$64 \times 64 \times 64$	4	16	6	7.65E-4	3.33E+1
$128 \times 128 \times 128$	4	15	5	1.90E-4	2.46E+2
$256 \times 256 \times 256$	4	14	5	4.74E-5	2.15E+3
$512 \times 512 \times 512$	4	14	4	1.24E-5	2.09E+4

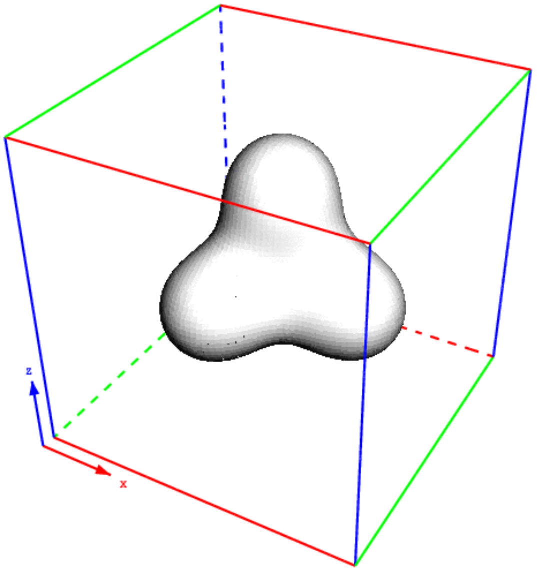

Fig. 7. The solute region Ω_i is a four-atoms molecule (**Example 5**).

Table 5
Numerical results of **Example 5** in three space dimensions

grid size	#NEWTON	#GMRES	#PCG	$\|e_h\|_\infty$	CPU (sec)
$64 \times 64 \times 64$	4	17	6	7.54E-4	6.52E+1
$128 \times 128 \times 128$	4	16	5	1.89E-4	3.60E+2
$256 \times 256 \times 256$	4	16	5	4.73E-5	2.86E+3
$512 \times 512 \times 512$	4	16	4	1.45E-5	2.53E+4

$\mathbf{p}_3 = (-\sqrt{3}/3\,, -1\,, -\sqrt{6}/6\,)\,a$, $\mathbf{p}_4 = (\,0\,, 0\,, \sqrt{6}/2\,)\,a$ and the radii $r_i = a$ for $i = 1,\ 2,\ 3,\ 4$ with $a = 0.4$ and $c = 0.6$. Fig. 7 shows the four-atoms molecule in the domain Ω. Numerical results of this example are listed in Table 5.

All the numerical tests consistently show that the number of either Newton or GMRES iterations involved with the Newton-KFBI method for the nonlinear Poisson-Boltzmann interface problem are essentially independent of the mesh parameter and the grid size; the numerical solutions generated by the Newton-

KFBI method in both two and three space dimensions have second order accuracy in the discrete maximum norm. The CPU times used by the computer scale linearly well with different grid sizes, indicating the algorithm has linear complexity.

7 Discussion

In this work, we propose a Newton-KFBI method for the nonlinear PB interface problem. The Newton method iteratively linearizes the nonlinear PB equation. The KFBI method solves the linearized PB equation in the formulation of boundary integral equations. The boundary integral system is well-conditioned and its solution by a Krylov subspace method can be done efficiently with the iteration number essentially independent of the mesh parameter and the system dimension. In each Krylov subspace iteration for the discrete equations, corresponding to the boundary integral system (21), one matrix vector multiplication is needed and each matrix vector multiplication involves the solution of two simple interface problems (26) and (27), both of which are much easier to solve than the heterogeneous interface problem (8)-(12). The first simple interface problem (26) is solved with an FFT based Poisson solver and the second simple interface problem (27) is solved with a full V-cycle geometric multigrid PCG solver. The overall computational work involved with the Newton-KFBI method is dominantly the number of Newton iterations times the number of Krylov subspace iterations times the work for those two interface problems. As the numbers of Newton iterations and Krylov subspace iterations are essentially independent of the mesh parameter and system dimension, the computational work is essentially linearly proportional to the number of nodes on the Cartesian grid that covers the rectangular domain.

One major advantage of the KFBI method for the PB equation is that it avoids generation of any body-fitted unstructured grids, which is especially a time-consuming or expensive process in three space dimensions where the dielectric interface may be very complicated. Another advantage of the KFBI method for evaluating boundary and volume integrals is that it yields accurate values at any point around the interface and in the domain and does not need to make any special treatment (such as singularity subtraction and integral regularization) for the nearly singular or hyper-singular boundary integrals as the traditional boundary element/integral method [7,8,15,30,32,40,72,76,77].

We may further improve the performance of the kernel-free boundary integral method by working with an adaptive and local mesh refinement algorithm. If the simple interface problems for boundary and volume integrals are solved with an adaptive algorithm, the overall computational work is expected to be significantly reduced and becomes essentially linearly proportional to the number of unknowns on the dielectric interface (instead of in the domain). We may also improve the kernel-free boundary integral solver for the nonlinear PB interface problem by combining together the Newton, GMRES and multigrid PCG iterations. In the current implementation, the three layers of iterations are kind of decoupled. A nice combination of the iterations may eliminate at least part of the inner iterations.

In this work, we only present the method for the PB interface problem. The methodology can also be extended for boundary value problems of the nonlinear PB equation, which may be subject to different boundary conditions such as the Neumann or Robin boundary condition [31].

The Newton-KFBI method is by no means limited to the PB equation (2) for symmetric 1 : 1 salt. It should be straightforwardly applicable for the general PB equation (1). The method may also work well for physically more realistic (modified) PB models [42]. We will report the application of the Newton-KFBI method for modified PB equations in future work.

Acknowledgments

Research of the author was supported in part by the National Science Foundation of the USA under Grant DMS–0915023, and is currently supported in part by the National Natural Science Foundation of China under Grants DMS–11101278, DMS–91130012 and the Young Thousand Talents Program of China.

A Calculation for the Jumps of Partial Derivatives

The interface problems (24)-(26) can be written as a unified interface problem of the following form

$$\begin{aligned} \triangle v - c\,v &= f \quad \text{in } \Omega \setminus \Gamma, && \text{(A.1)}\\ [v] &= \varphi \quad \text{on } \Gamma, && \text{(A.2)}\\ [\partial_{\mathbf{n}} v] &= \psi \quad \text{on } \Gamma. && \text{(A.3)} \end{aligned}$$

Here, the source function f is a piecewise smooth function, which vanishes on at least one side of the dielectric interface Γ; the coefficient c vanishes ($c = 0$) for the interface problems (24) and (26); $c = \epsilon_e^{-1}\kappa^2\cosh(u_e^m)$ for the interface problems (25) and (27). Since the Newton-KFBI method guarantees that the approximate solution u_e^m is continuous on $\bar{\Omega}$, we may assume the reaction coefficient c is continuous on $\bar{\Omega}$, too. In fact, the solution v to the interface problem above is the sum of a volume integral, a double layer boundary integral and a single layer boundary integral with their densities be f, φ and ψ, respectively.

A.1 Two Space Dimensions

This subsection describes the calculation for the jumps of partial derivatives of the solution v in two space dimensions.

Let $\boldsymbol{\tau}$ be a tangent vector at a point on the interface. Taking tangential derivative of the interface condition (A.2) along $\boldsymbol{\tau}$ yields

$$\partial_{\boldsymbol{\tau}}[v] = \partial_{\boldsymbol{\tau}}\varphi \qquad \text{on } \Gamma. \tag{A.4}$$

The two equations (A.3)-(A.4) will uniquely determine the jumps of the first order partial derivatives: $[v_x]$ and $[v_y]$.

Taking tangential derivative of interface condition (A.3) along $\boldsymbol{\tau}$ yields

$$\partial_{\boldsymbol{\tau}}[\partial_{\mathbf{n}} v] = \partial_{\boldsymbol{\tau}}\psi \qquad \text{on } \Gamma. \tag{A.5}$$

Taking tangential derivative of condition (A.4) yields

$$\partial_{\boldsymbol{\tau}\boldsymbol{\tau}}[v] = \partial_{\boldsymbol{\tau}\boldsymbol{\tau}}\varphi \qquad \text{on } \Gamma. \tag{A.6}$$

The partial differential equation (A.1) implies

$$[\triangle v] - c\,[v] = [f] \qquad \text{on } \Gamma. \tag{A.7}$$

Here, we have used the fact that the coefficient c is continuous across Γ. The three equations (A.5)-(A.7) will determine the jumps of the second order partial derivatives: $[v_{xx}],\ [v_{yy}],\ [v_{xy}]$.

Let s be an independent variable for the local parametric representation of boundary Γ, assuming

$$x = x(s) \quad \text{and} \quad y = y(s).$$

The parameter could be x or y. Let

$$\mathbf{r} = (x, y)^{\mathrm{T}}, \quad \boldsymbol{\tau} = \frac{\partial \mathbf{r}}{\partial s}$$

and n_1, n_2 be the two components of the unit normal vector $\mathbf{n}$.

Equations (A.3)-(A.4) explicitly read

$$n_1[v_x] + n_2[v_y] = \psi, \tag{A.8}$$

$$\frac{\partial x}{\partial s}[v_x] + \frac{\partial y}{\partial s}[v_y] = \varphi_s. \tag{A.9}$$

Solving the two by two linear system yields the jumps of the first order partial derivatives.

Equations (A.5)-(A.7) explicitly read

$$n_1\frac{\partial x}{\partial s}[v_{xx}] + n_2\frac{\partial y}{\partial s}[v_{yy}] + \Big(n_1\frac{\partial y}{\partial s} + n_2\frac{\partial x}{\partial s}\Big)[v_{xy}] = r_1, \tag{A.10}$$

$$\Big(\frac{\partial x}{\partial s}\Big)^2[v_{xx}] + \Big(\frac{\partial y}{\partial s}\Big)^2[v_{yy}] + 2\frac{\partial x}{\partial s}\frac{\partial y}{\partial s}[v_{xy}] = r_2, \tag{A.11}$$

$$[v_{xx}] + [v_{yy}] - c\,[v] = [f]. \tag{A.12}$$

with

$$r_1 = \psi_s - \frac{\partial n_1}{\partial s}[v_x] - \frac{\partial n_2}{\partial s}[v_y] \quad \text{and} \quad r_2 = \varphi_{ss} - \frac{\partial^2 x}{\partial s^2}[v_x] - \frac{\partial^2 y}{\partial s^2}[v_y].$$

Solving the linear system consisting of the three equations above yields the jumps of the second order partial derivatives.

The right hand sides of equations (A.10)-(A.12) involve derivatives of the unit normal vector and the coordinates (x, y) with respect to the parameter s. Provided that the boundary Γ is given as the zero level set of a smooth function, these derivatives can be computed in terms of the level set function and its partial derivatives. The rest of this section will illustrate the computation.

Let $\Theta(x, y)$ be the smooth level set function such that the boundary Γ is given by

$$\Gamma = \{\,(x, y) \in \mathbb{R}^3 \,:\, \Theta(x, y) = 0\,\}.$$

Assume that $\Theta(x, y) > 0$ at point (x, y) outside the domain bounded by Γ. The two components of the unit outward normal $\mathbf{n} = (n_1, n_2)^{\mathrm{T}}$ at a point on Γ can be computed by

$$n_1 = \frac{\Theta_x}{\sqrt{\Theta_x^2 + \Theta_y^2}},$$

$$n_2 = \frac{\Theta_y}{\sqrt{\Theta_x^2 + \Theta_y^2}}.$$

For simplicity, in this appendix we only consider the case that the normal $\mathbf{n}$ has its magnitude-largest component in the direction along Z-axis while the computation for the other two cases is similar. This implies that $\Theta_z \neq 0$ and the surface can be locally represented by a Monge curve

$$y = H(x)$$

with $H(x)$ be a smooth function of x. In this case, the coordinate variable x is taken as the independent variable, i.e., $s = x$. Note that

$$0 = \Theta_x + \Theta_y \frac{\partial y}{\partial x} = \Theta_x + \Theta_y H_x.$$

We get the first order partial derivatives

$$H_x = -\frac{\Theta_x}{\Theta_y},$$

and the second order partial derivatives

$$H_{xx} = -\frac{(\Theta_{xx}\Theta_y^2 + \Theta_x^2\Theta_{yy}) - 2\,\Theta_x\Theta_y\Theta_{xy}}{\Theta_y^3}$$

The tangent vector is given by $\boldsymbol{\tau} = (1, H_x)^{\mathrm{T}} = (1, -\Theta_x/\Theta_z)^{\mathrm{T}}$ and the second order partial derivatives of y are given by $\partial^2 y/\partial x^2 = H_{xx}$.

In addition, the vector $\boldsymbol{\nu} = (\nu_1, \nu_2)^{\mathrm{T}}$ with components given by

$$\begin{aligned}\nu_1 &= \frac{H_x}{\sqrt{1 + H_x^2}},\\ \nu_2 &= -\frac{1}{\sqrt{1 + H_x^2}}\end{aligned}$$

is also a unit normal on Γ except that it points to the opposite direction of the outward normal $\mathbf{n}$ if Θ_y has a positive sign. We compute the two components of vector $\partial\boldsymbol{\nu}/\partial x$ by

$$\begin{aligned}\frac{\partial \nu_1}{\partial x} &= \frac{H_{xx}}{(1 + H_x^2)^{3/2}},\\ \frac{\partial \nu_2}{\partial x} &= \frac{H_x H_{xx}}{(1 + H_x^2)^{3/2}}.\end{aligned}$$

These two components are needed in evaluating the right hand sides of (A.10)-(A.12).

A.2 Three Space Dimensions

This subsection describes the calculation for the jumps of partial derivatives of the solution v to the unified interface problem (A.1)-(A.3) in three space dimensions.

Let $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ be two tangent vectors at a point on the interface Γ. Taking tangential derivatives of the interface condition (A.2) along $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ respectively yields

$$\partial_{\boldsymbol{\tau}_1}[v] = \partial_{\boldsymbol{\tau}_1}\varphi \qquad \text{on } \Gamma, \tag{A.13}$$
$$\partial_{\boldsymbol{\tau}_2}[v] = \partial_{\boldsymbol{\tau}_2}\varphi \qquad \text{on } \Gamma. \tag{A.14}$$

The three equations consisting of (A.3) and (A.13)-(A.14) will uniquely determine the jumps of the first order partial derivatives: $[v_x]$, $[v_y]$ and $[v_z]$.

Taking tangential derivatives of interface condition (A.3) along $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$ respectively yields

$$\partial_{\boldsymbol{\tau}_1}[\partial_{\mathbf{n}}v] = \partial_{\boldsymbol{\tau}_1}\psi \qquad \text{on } \Gamma, \tag{A.15}$$
$$\partial_{\boldsymbol{\tau}_2}[\partial_{\mathbf{n}}v] = \partial_{\boldsymbol{\tau}_2}\psi \qquad \text{on } \Gamma. \tag{A.16}$$

Taking tangential derivatives of conditions (A.13) and (A.14) yields

$$\partial_{\boldsymbol{\tau}_1\boldsymbol{\tau}_1}[v] = \partial_{\boldsymbol{\tau}_1\boldsymbol{\tau}_1}\varphi \qquad \text{on } \Gamma, \tag{A.17}$$
$$\partial_{\boldsymbol{\tau}_2\boldsymbol{\tau}_2}[v] = \partial_{\boldsymbol{\tau}_2\boldsymbol{\tau}_2}\varphi \qquad \text{on } \Gamma, \tag{A.18}$$
$$\partial_{\boldsymbol{\tau}_1\boldsymbol{\tau}_2}[v] = \partial_{\boldsymbol{\tau}_1\boldsymbol{\tau}_2}\varphi \qquad \text{on } \Gamma. \tag{A.19}$$

The partial differential equation (A.1) implies

$$[\triangle v] - c\,[v] = [f] \qquad \text{on } \Gamma. \tag{A.20}$$

Here, we have used the fact that the coefficient c is continuous across Γ, too. The six equations (A.15)-(A.20) will determine the jumps of the second order partial derivatives: $[v_{xx}]$, $[v_{yy}]$, $[v_{zz}]$, $[v_{yz}]$, $[v_{zx}]$, $[v_{xy}]$.

Let s_1 and s_2 be two independent variables for the local parametric representation of boundary Γ, assuming

$$x = x(s_1, s_2), \quad y = y(s_1, s_2) \quad \text{and} \quad z = z(s_1, s_2).$$

The pair of variables (s_1, s_2) could be (y, z), (z, x) or (x, y). Let

$$\mathbf{r} = (x, y, z)^{\mathrm{T}}, \quad \boldsymbol{\tau}_1 = \frac{\partial \mathbf{r}}{\partial s_1} \quad \text{and} \quad \boldsymbol{\tau}_2 = \frac{\partial \mathbf{r}}{\partial s_2}$$

and n_1, n_2 and n_3 be the three components of the unit normal vector $\mathbf{n}$.

Equations (A.3)-(A.14) explicitly read

$$n_1[v_x] + n_2[v_y] + n_3[v_z] = \psi, \tag{A.21}$$

$$\frac{\partial x}{\partial s_1}[v_x] + \frac{\partial y}{\partial s_1}[v_y] + \frac{\partial z}{\partial s_1}[v_z] = \varphi_{s_1}, \tag{A.22}$$

$$\frac{\partial x}{\partial s_2}[v_x] + \frac{\partial y}{\partial s_2}[v_y] + \frac{\partial z}{\partial s_2}[v_z] = \varphi_{s_2}. \tag{A.23}$$

Solving the three by three linear system yields the jumps of the first order partial derivatives.

Equations (A.15)-(A.16) explicitly read

$$\begin{aligned}
&n_1\frac{\partial x}{\partial s_1}[v_{xx}] + n_2\frac{\partial y}{\partial s_1}[v_{yy}] + n_3\frac{\partial z}{\partial s_1}[v_{zz}] \\
&+\Big(n_2\frac{\partial z}{\partial s_1} + n_3\frac{\partial y}{\partial s_1}\Big)[v_{yz}] + \Big(n_3\frac{\partial x}{\partial s_1} + n_1\frac{\partial z}{\partial s_1}\Big)[v_{zx}] + \Big(n_1\frac{\partial y}{\partial s_1} + n_2\frac{\partial x}{\partial s_1}\Big)[v_{xy}] \\
&= \psi_{s_1} - \frac{\partial n_1}{\partial s_1}[v_x] - \frac{\partial n_2}{\partial s_1}[v_y] - \frac{\partial n_3}{\partial s_1}[v_z]
\end{aligned} \tag{A.24}$$

and

$$\begin{aligned}
&n_1\frac{\partial x}{\partial s_2}[v_{xx}] + n_2\frac{\partial y}{\partial s_2}[v_{yy}] + n_3\frac{\partial z}{\partial s_2}[v_{zz}] \\
&+\Big(n_2\frac{\partial z}{\partial s_2} + n_3\frac{\partial y}{\partial s_2}\Big)[v_{yz}] + \Big(n_3\frac{\partial x}{\partial s_2} + n_1\frac{\partial z}{\partial s_2}\Big)[v_{zx}] + \Big(n_1\frac{\partial y}{\partial s_2} + n_2\frac{\partial x}{\partial s_2}\Big)[v_{xy}] \\
&= \psi_{s_2} - \frac{\partial n_1}{\partial s_2}[v_x] - \frac{\partial n_2}{\partial s_2}[v_y] - \frac{\partial n_3}{\partial s_2}[v_z].
\end{aligned} \tag{A.25}$$

Equations (A.17)-(A.19) explicitly read

$$\begin{aligned}
&\Big(\frac{\partial x}{\partial s_1}\Big)^2[v_{xx}] + \Big(\frac{\partial y}{\partial s_1}\Big)^2[v_{yy}] + \Big(\frac{\partial z}{\partial s_1}\Big)^2[v_{zz}] \\
&+ 2\frac{\partial y}{\partial s_1}\frac{\partial z}{\partial s_1}[v_{yz}] + 2\frac{\partial z}{\partial s_1}\frac{\partial x}{\partial s_1}[v_{zx}] + 2\frac{\partial x}{\partial s_1}\frac{\partial y}{\partial s_1}[v_{xy}] \\
&= \varphi_{s_1 s_1} - \frac{\partial^2 x}{\partial s_1^2}[v_x] - \frac{\partial^2 y}{\partial s_1^2}[v_y] - \frac{\partial^2 z}{\partial s_1^2}[v_z],
\end{aligned} \tag{A.26}$$

$$\begin{aligned}
&\Big(\frac{\partial x}{\partial s_2}\Big)^2[v_{xx}] + \Big(\frac{\partial y}{\partial s_2}\Big)^2[v_{yy}] + \Big(\frac{\partial z}{\partial s_2}\Big)^2[v_{zz}] \\
&+ 2\frac{\partial y}{\partial s_2}\frac{\partial z}{\partial s_2}[v_{yz}] + 2\frac{\partial z}{\partial s_2}\frac{\partial x}{\partial s_2}[v_{zx}] + 2\frac{\partial x}{\partial s_2}\frac{\partial y}{\partial s_2}[v_{xy}] \\
&= \varphi_{s_2 s_2} - \frac{\partial^2 x}{\partial s_2^2}[v_x] - \frac{\partial^2 y}{\partial s_2^2}[v_y] - \frac{\partial^2 z}{\partial s_2^2}[v_z]
\end{aligned} \tag{A.27}$$

and

$$
\begin{aligned}
&\frac{\partial x}{\partial s_1}\frac{\partial x}{\partial s_2}[v_{xx}] + \frac{\partial x}{\partial s_1}\frac{\partial y}{\partial s_2}[v_{xy}] + \frac{\partial x}{\partial s_1}\frac{\partial z}{\partial s_2}[v_{xz}] \\
&\frac{\partial y}{\partial s_1}\frac{\partial x}{\partial s_2}[v_{yx}] + \frac{\partial y}{\partial s_1}\frac{\partial y}{\partial s_2}[v_{yy}] + \frac{\partial y}{\partial s_1}\frac{\partial z}{\partial s_2}[v_{yz}] \\
&\frac{\partial z}{\partial s_1}\frac{\partial x}{\partial s_2}[v_{zx}] + \frac{\partial z}{\partial s_1}\frac{\partial y}{\partial s_2}[v_{zy}] + \frac{\partial z}{\partial s_1}\frac{\partial z}{\partial s_2}[v_{zz}] \\
= {}&\varphi_{s_1 s_2} - \frac{\partial^2 x}{\partial s_1 \partial s_2}[v_x] - \frac{\partial^2 y}{\partial s_1 \partial s_2}[v_y] - \frac{\partial^2 z}{\partial s_1 \partial s_2}[v_z]. \qquad \text{(A.28)}
\end{aligned}
$$

Equation (A.20) is explicitly written as

$$[v_{xx}] + [v_{yy}] + [v_{zz}] = c\,[v] + [f]. \qquad \text{(A.29)}$$

Solving the six by six linear system consisting of equations (A.24)-(A.29) yields the jumps of the second order partial derivatives.

The right hand sides of equations (A.24)-(A.29) involve derivatives of the unit normal vector and the coordinates (x, y and z) with respect to the parameters s_1 and s_2. Provided that the boundary Γ is given as the zero level set of a smooth function, these derivatives can also be computed in terms of the level set function and its partial derivatives. The rest of this section will illustrate the computation.

Let $\Theta(x, y, z)$ be the smooth level set function such that the boundary Γ is given by

$$\Gamma = \{\, (x, y, z) \in \mathbb{R}^3 \,:\, \Theta(x, y, z) = 0 \,\}.$$

Assume that $\Theta(x, y, z) > 0$ at point (x, y, z) outside the domain bounded by Γ. The three components of the unit outward normal $\mathbf{n} = (n_1, n_2, n_3)^{\mathrm{T}}$ at a point on Γ can be computed by

$$
\begin{aligned}
n_1 &= \frac{\Theta_x}{\sqrt{\Theta_x^2 + \Theta_y^2 + \Theta_z^2}}, \\
n_2 &= \frac{\Theta_y}{\sqrt{\Theta_x^2 + \Theta_y^2 + \Theta_z^2}}, \\
n_3 &= \frac{\Theta_z}{\sqrt{\Theta_x^2 + \Theta_y^2 + \Theta_z^2}}.
\end{aligned}
$$

For simplicity, in this appendix we only consider the case that the normal $\mathbf{n}$ has its magnitude-largest component in the direction along Z-axis while the computation for the other two cases is similar. This implies that $\Theta_z \neq 0$ and the surface can be locally represented by a Monge patch

$$z = H(x, y)$$

with $H(x,y)$ be a smooth function of x and y. In this case, the coordinate variables, x and y, are taken as the parameters, i.e., $s_1 = x$ and $s_2 = y$. Note that

$$\begin{aligned} 0 &= \Theta_x + \Theta_z \frac{\partial z}{\partial x} = \Theta_x + \Theta_z H_x, \\ 0 &= \Theta_y + \Theta_z \frac{\partial z}{\partial y} = \Theta_y + \Theta_z H_y. \end{aligned}$$

We get the first order partial derivatives

$$\begin{aligned} H_x &= -\frac{\Theta_x}{\Theta_z}, \\ H_y &= -\frac{\Theta_y}{\Theta_z}, \end{aligned}$$

and the second order partial derivatives

$$\begin{aligned} H_{xx} &= -\frac{\Theta_x^2\Theta_{zz} + \Theta_{xx}\Theta_z^2 - 2\,\Theta_x\Theta_{xz}\Theta_z}{\Theta_z^3}, \\ H_{yy} &= -\frac{\Theta_y^2\Theta_{zz} + \Theta_{yy}\Theta_z^2 - 2\,\Theta_y\Theta_{yz}\Theta_z}{\Theta_z^3}, \\ H_{xy} &= -\frac{\Theta_z(\Theta_{xy}\Theta_z - \Theta_x\Theta_{yz} - \Theta_y\Theta_{xz}) + \Theta_x\Theta_y\Theta_{zz}}{\Theta_z^3}. \end{aligned}$$

The tangent vectors are given by $\boldsymbol{\tau}_1 = (1, 0, H_x)^{\mathrm{T}} = (1, 0, -\Theta_x/\Theta_z)^{\mathrm{T}}$ and $\boldsymbol{\tau}_2 = (0, 1, H_y)^{\mathrm{T}} = (0, 1, -\Theta_y/\Theta_z)^{\mathrm{T}}$ and the second order partial derivatives of z are given by $\partial^2 z/\partial x^2 = H_{xx}$, $\partial^2 z/\partial y^2 = H_{yy}$ and $\partial^2 z/\partial x\partial y = H_{xy}$.

In addition, the vector $\boldsymbol{\nu} = (\nu_1, \nu_2, \nu_3)^{\mathrm{T}}$ with components given by

$$\begin{aligned} \nu_1 &= \frac{H_x}{\sqrt{1 + H_x^2 + H_y^2}}, \\ \nu_2 &= \frac{H_y}{\sqrt{1 + H_x^2 + H_y^2}}, \\ \nu_3 &= -\frac{1}{\sqrt{1 + H_x^2 + H_y^2}} \end{aligned}$$

is also a unit normal on Γ except that it points to the opposite direction of the outward normal $\mathbf{n}$ if Θ_z has a positive sign. We compute the three components of vector $\partial\boldsymbol{\nu}/\partial x$ by

$$\begin{aligned}
\frac{\partial\nu_1}{\partial x} &= \frac{H_{xx} + H_y(H_{xx}H_y - H_xH_{xy})}{(1 + H_x^2 + H_y^2)^{3/2}}, \\
\frac{\partial\nu_2}{\partial x} &= \frac{H_{xy} + H_x(H_{xy}H_x - H_yH_{xx})}{(1 + H_x^2 + H_y^2)^{3/2}}, \\
\frac{\partial\nu_3}{\partial x} &= \frac{H_xH_{xx} + H_yH_{xy}}{(1 + H_x^2 + H_y^2)^{3/2}},
\end{aligned}$$

and the three components of vector $\partial\boldsymbol{\nu}/\partial y$ by

$$\begin{aligned}
\frac{\partial\nu_1}{\partial y} &= \frac{H_{xy} + H_y(H_{xy}H_y - H_xH_{yy})}{(1 + H_x^2 + H_y^2)^{3/2}}, \\
\frac{\partial\nu_2}{\partial y} &= \frac{H_{yy} + H_x(H_{yy}H_x - H_yH_{xy})}{(1 + H_x^2 + H_y^2)^{3/2}}, \\
\frac{\partial\nu_3}{\partial y} &= \frac{H_xH_{xy} + H_yH_{yy}}{(1 + H_x^2 + H_y^2)^{3/2}}.
\end{aligned}$$

These six components are needed in evaluating the right hand sides of (A.24)-(A.25).

References

[1] S. A. Allison, J. J. Sines, A. Wierzbicki, Solutions of the full Poisson-Boltzmann equation with application to diffusion-controlled reactions, J. Phys. Chem. 93 (1989) 5819.

[2] B. Alpert, G. Beylkin, R. Coifman, V. Rokhlin, Wavelet-like bases for the fast solutions of second-kind integral equations, SIAM J. Sci. Comput. 14 (1) (1993) 159–184.

[3] M. D. Altman, J. P. Bardhan, B. Tidor, J. K. White, FFTSVD: a fast multiscale boundary element method solver suitable for Bio-MEMS and biomolecule simulation, IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems 25 (2) (2006) 274–284.

[4] K. E. Atkinson, The Numerical Solution of Integral Equations of the Second Kind, Cambridge University Press, Cambridge, UK, 1997.

[5] N. Baker, M. Holst, 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) 1343.

[6] J. Barnes, P. Hut, A hierarchical $O(n \log n)$ force-calculation algorithm, Nature 324 (4) (1986) 85–103.

[7] J. T. Beale, A grid-based boundary integral method for elliptic problems in three-dimensions, SIAM J. Numer. Anal. 42 (2004) 599–620.

[8] J. T. Beale, M. C. Lai, A method for computing nearly singular integrals, SIAM J. Numer. Anal. 38 (6) (2001) 1902–1925.

[9] A. J. Bordner, G. A. Huber, Boundary element solution of the linear Poisson-Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution, J. Comput. Chem. 24 (3) (2003) 353–367.

[10] A. H. Boschitsch, M. O. Fenley, Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzmann equation, J. Comput. Chem. 25 (7) (2004) 935–955.

[11] A. H. Boschitsch, M. O. Fenley, H. X. Zhou, Fast boundary element method for the linear Poisson-Boltzmann equation, J. Phys. Chem. B 106 (10) (2002) 2741–2754.

[12] W. L. Briggs, V. E. Henson, S. F. McCormick, A Multigrid Tutorial, SIAM Books, Philadelphia, 2000, second edition.

[13] G. F. Carey, Computational Grids: Generations, Adaptation & Solution Strategies, CRC, 1997.

[14] D. L. Chapman, A contribution to the theory of electrocapollarity, Phil. Mag. 25 (1913) 475–481.

[15] J. T. Chen, H. K. Hong, Review of dual boundary element methods with emphasis on hypersingular integrals and divergent series, Appl. Mech. Rev. 52 (1) (1999) 17–33.

[16] L. Chen, M. Holst, J. Xu, The finite element approximation of the nonlinear Poisson-Boltzmann equation, SIAM J. Numer. Anal. 45 (6) (2007) 2298–2320.

[17] I.-L. Chern, J.-G. Liu, W.-C. Wang, Accurate evaluation of electrostatics for macro-molecules in solution, Methods and Applications of Analysis 10 (2003) 309–328.

[18] I.-L. Chern, Y. C. Shu, Coupling interface method for elliptic interface problems, J. Comput. Phys. 225 (2) (2007) 2138–2174.

[19] C. M. Cortis, R. A. Friesner, An automatic three-dimensional finite element mesh generation system for the Poisson-Boltzmann equation, J. Comput. Chem. 18 (1997) 1570.

[20] C. M. Cortis, R. A. Friesner, Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes, J. Comput. Chem. 18 (1997) 1591.

[21] M. E. Davis, J. A. McCammon, Solving the finite difference linearized Poisson-Boltzmann equation: a comparison of relaxation and conjugate gradient methods, J. Comput. Chem. 10 (1989) 386.

[22] M. E. Davis, J. A. McCammon, Electrostatics in biomolecular structure and dynamics, Chem. Rev. 90 (3) (1990) 509–521.

[23] S. Z. Deng, K. Ito, Z. Li, Three-dimensional elliptic solvers for interface problems and applications, J. Comput. Phys. 184 (2003) 215–243.

[24] H. Edelsbrunner, Geometry and Topology for Mesh Generation, Cambridge University Press, 2006.

[25] E. Frackowiak, Carbon materials for the electrochemical storage of energy in capacitors, Carbon 39 (2001) 937.

[26] M. K. Gilson, K. A. Sharp, B. H. Honig, Calculating the electrostatic potential of molecules in solution: method and error assessment, J. Comput. Chem. 9 (1988) 327.

[27] G. Gouy, Constitution of the electric charge at the surface of an electrolyte, J. Phys. 9 (1910) 457–468.

[28] L. F. Greengard, J. F. Huang, A new version of the fast multipole method for screened coulomb interactions in three dimensions, J. Comput. Phys. 180 (2) (2002) 642–658.

[29] W. Hackbusch, Z. P. nowak, On the fast matrix multiplication in the boundary element method by panel clustering, Numer. Math. 54 (4) (1989) 463–491.

[30] S. Hao, A. H. Barnett, P. G. Martinsson, P. Young, High-order accurate methods for Nyström discretization of integral equations on smooth curves in the plane, Adv. Comput. Math. 40 (2014) 245–272.

[31] A. Helgadóttir, F. Gibou, A Poisson-Boltzmann solver on irregular domains with Neumann or Robin boundary conditions on non-graded adaptive grid, J. Comput. Phys. 230 (10) (2011) 3830–3848.

[32] J. Helsing, R. Ojala, On the evaluation of layer potentials close to their sources, Journal of Computational Physics 227 (5) (2008) 2899–2921.

[33] M. Holst, Multilevel methods for the Poisson-Boltzmann equation, Ph.D. thesis, University of Illinois and Urbana-Champaign (1993).

[34] M. Holst, N. Baker, F. Wang, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I. algorithms and examples, J. Comput. Chem. 21 (2000) 1319.

[35] M. Holst, J. A. McCammon, Z. Yu, Y. C. Zhou, Y. Zhu, Adaptive finite element modeling techniques for the Poisson-Boltzmann equation, Commun. in Comput. Phys. 11 (2012) 179–214.

[36] M. Holst, F. Saied, Multigrid solution of the Poisson-Boltzmann equation, J. Comput. Chem. 14 (1993) 105.

[37] M. Holst, F. Saied, Numerical solution of the nonlinear Poisson-Boltzmann equation: developing more robust and efficient methods, J. Comput. Chem. 16 (1994) 337.

[38] B. Honig, A. Nicholls, Classical electrostatics in biology and chemistry, Science 268 (1995) 1144–1149.

[39] A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, H. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach, J. Comput. Phys. 97 (1991) 144.

[40] A. Klöckner, A. Barnett, L. Greengard, M. O'Neil, Quadrature by expansion: A new method for the evaluation of layer potentials, Journal of Computational Physics 252 (2013) 332–349.

[41] R. Kotz, Principles and applications of electrochemical capacitors, Electrochim. Acta 45 (2000) 2483.

[42] C.-C. Lee, H. Lee, Y. Hyon, T.-C. Lin, C. Liu, New Poisson-Boltzmann type equations: one-dimensional solutions, Nonlinearity 24 (2011) 431–458.

[43] S. Leung, H. Zhao, A grid based particle method for moving interface problems, Journal of Computational Physics 228 (8) (2009) 2993–3024.

[44] Z. Li, A fast iterative algorithm for elliptic interface problems, SIAM J. Numer. Anal. 35 (1) (1998) 230–254.

[45] Z. Li, K. Ito, The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains, SIAM Frontiers in Applied Mathematics, 2006.

[46] Z. Li, C. V. Pao, Z. Qiao, A finite difference method and analysis for 2D nonlinear Poisson-Boltzmann equations, J. Scientific Computing 30 (1) (2007) 61–81.

[47] J. Liang, S. Subramaniam, Computation of molecular electrostatics with boundary elmement methods, Biophys. J. 73 (4) (1997) 1830–1841.

[48] B. Z. Lu, X. L. Cheng, J. F. Huang, J. A. McCammon, Order N algorithm for computation of electrostatic interactions in biomolecular systems, Proc. Natl. Acad. Sci. U. S. A. 103 (51) (2006) 19314–19319.

[49] B. Z. Lu, J. A. McCammon, Improved boundary element methods for Poisson-Boltzmann electrostatic potential and force calculations, J. Chem. Theory Comput. 3 (3) (2007) 1134–1142.

[50] B. Z. Lu, D. Q. Zhang, J. A. McCammon, Computation of electrostatic forces between solvated molecules determined by the Poisson-Boltzmann equation using a boundary element method, J. Chem. Phys. 122 (21) (2005) 214102.

[51] B. Z. Lu, Y. Zhou, M. Holst, J. McCammon, Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications, Communications in Computational Physics 3 (5) (2008) 973–1009.

[52] R. Luo, L. David, M. Gilson, Accelerated Poisson-Boltzmann calculations for static and dynamic systems, J. Comput. Chem. 23 (2002) 1244.

[53] B. Luty, M. Davis, J. McCammon, Solving the finite-difference non-linear Poisson-Boltzmann equation, J. Comput. Chem. 13 (1992) 1114.

[54] J. Lyklema, Fundamentals of Interface and Colloid Science, Vol. 2, Academic Press, 1995.

[55] A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions, SIAM J. Numer. Anal. 21 (1984) 285–299.

[56] A. Mayo, Fast high order accurate solution of Laplace's equation on irregular regions, SIAM J. Sci. Statist. Comput. 6 (1985) 144–157.

[57] S. McLaughlin, The electrostatic properties of membranes, Ann. Rev. Biophys. Biophys. Chem. 18 (1989) 113.

[58] M. Mirzadeh, M. Theillard, F. Gibou, A second-order discretization of the nonlinear Poisson-Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids, J. Comp. Phy. 230 (2011) 2125–2140.

[59] M. Mirzadeh, M. Theillard, A. Helgadóttir, D. Boy, F. Gibou, An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations, Commun. Comput. Phys. 13 (1) (2013) 150–173.

[60] A. Nicholls, B. Honig, A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation, J. Comput. Chem. 12 (1991) 435.

[61] E. T. Ong, K. H. Lee, K. M. Lim, A fast algorithm for three-dimensional electrostatics analysis: fast Fourier transform on multipoles (FFTM), Int. J. Numer. Methods Engr. 61 (5) (2004) 633–656.

[62] E. T. Ong, K. M. Lim, K. H. Lee, H. P. Lee, A fast algorithm for three-dimensional potential fields calculation: fast Fourier transform on multipoles, J. Comput. Phys. 192 (1) (2003) 244–261.

[63] J. Phillips, J. K. White, A precorrected-FFT method for electrostatic analysis of complicated 3-D structures, IEEE Transactions on Computer-Aided Design of Integrated Circuits Systems 16 (1997) 1059–1072.

[64] Z. Qiao, Z. Li, T. Tang, A finite difference scheme for solving the nonlinear Poisson-Boltzmann equation modeling charged spheres, J. Comput. Math. 24 (2006) 252–264.

[65] A. A. Rashin, J. Malinsky, New method for the computation of ionic distribution around rod-like polyelectrolytes with the helical distribution of charges. i. general approach and a nonlinearized Poisson-Boltzmann equation, J. Comput. Chem. 12 (1991) 981.

[66] W. B. Russel, D. A. Saville, W. Schowalter, Colloidal Dispersions, Cambridge University Press, 1989.

[67] Y. Saad, Iterative Methods for Sparse Linear Systems, second edition, Society for Industrial and Applied Mathematics, 2003.

[68] K. A. Sharp, B. Honig, Electrostatic interactions in macromolecules: theory and applications, Ann. Rev. Biophys. Biophys. Chem. 19 (1990) 301.

[69] A. I. Shestakov, J. L. Milovich, A. Noy, Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method, J. Colloid Interface Sci. 247 (1) (2002) 62–79.

[70] P. Simon, Y. Gogotsi, Materials for electrochemical capacitors, Nature Mater. 7 (2008) 845.

[71] J. Strikwerda, Finite Difference Schemes and Partial Differential Equations, Society for Industrial and Applied Mathematics, 2007.

[72] M. Tanaka, V. Sladek, J. Sladek, Regularization techniques applied to boundary element method, AMSE Appl. Mech. Rev. 47 (1994) 457–499.

[73] J. Tausch, J. White, Multiscale bases for the sparse representation of boundary integral operators on complex geometry, SIAM J. Sci. Comput. 24 (5) (2003) 1610–1629.

[74] D. Xie, S. Zhou, A new minimization protocol for solving nonlinear Poisson-Boltzmann mortar finite element equation, BIT 47 (2007) 853–871.

[75] R.-J. Yang, L.-M. Fu, Y.-C. Lin, Electroosmotic flow in microchannels, J. Colloid Interface Sci. 239 (2001) 98.

[76] L. Ying, G. Biros, D. Zorin, A high-order 3d boundary integral equation solver for elliptic PDEs in smooth domains, Journal of Computational Physics 219 (1) (2006) 247–275.

[77] W.-J. Ying, J. T. Beale, A fast accurate boundary integral method for potentials on closely packed cells, Communications in Computational Physics 14 (4) (2013) 1073–1093.

[78] W.-J. Ying, C. S. Henriquez, A kernel-free boundary integral method for elliptic boundary value problems, J. Comp. Phy. 227 (2) (2007) 1046–1074.

[79] W.-J. Ying, W.-C. Wang, A kernel-free boundary integral method for implicitly defined surfaces, Journal of Computational Physics 252 (2013) 606–624.

[80] W.-J. Ying, W.-C. Wang, A kernel-free boundary integral method for variable coefficients elliptic PDEs, Communications in Computational Physics 15 (4) (2014) 1108–1140.

[81] B. Yoon, A. Lenhoff, A boundary element method for molecular electrostatics with electrolyte effects, J. Comput. Chem. 11 (1990) 1080.

[82] R. J. Zauhar, R. S. Morgan, A new method for computing the macromolecular electric potential, J. Mol. biol. 186 (4) (1985) 815–820.

[83] Y. C. Zhou, Matched interface and boundary (MIB) method and its applications to implicit solvent modeling of biomolecules, Ph.D. thesis, Michigan State University (2006).

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