Journal of Computational Physics 322 (2016) 1–20

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# A nonlocal modified Poisson–Boltzmann equation and finite element solver for computing electrostatics of biomolecules

Dexuan Xie*, Yi Jiang

*Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI, 53201, USA*

ARTICLE INFO

*Article history:*
Received 26 February 2016
Received in revised form 15 June 2016
Accepted 16 June 2016
Available online 21 June 2016

*Keywords:*
Nonlocal dielectric model
Poisson–Boltzmann equation
Finite element method
Newton method
Electrostatics

ABSTRACT

The nonlocal dielectric approach has been studied for more than forty years but only limited to water solvent until the recent work of Xie et al. (2013) [20]. As the development of this recent work, in this paper, a nonlocal modified Poisson–Boltzmann equation (NMPBE) is proposed to incorporate nonlocal dielectric effects into the classic Poisson–Boltzmann equation (PBE) for protein in ionic solvent. The focus of this paper is to present an efficient finite element algorithm and a related software package for solving NMPBE. Numerical results are reported to validate this new software package and demonstrate its high performance for protein molecules. They also show the potential of NMPBE as a better predictor of electrostatic solvation and binding free energies than PBE.

## 1. Introduction

The nonlocal dielectric approach started from Dogonadze and Kornyshev's early investigation around 1970s to reflect polarization correlations of water molecules or the spacial-frequency dependence of a dielectric medium in an electrostatic continuum model [1–4]. Since then, many studies were done but only limited to the water solvent, and did not involve any biomolecule due to lack of effective numerical schemes [5–14]. This situation was changed about ten years ago when Hildebrandt et al. reformulated a linear nonlocal dielectric model for protein in water approximately as a system of coupled partial differential equations (PDEs) [15], and solved it numerically by a boundary element algorithm [16] and an explicit jump immersed interface method [17].

Recently, by different techniques than the ones used by Hildebrandt et al., a typical linear nonlocal dielectric model, called the Fourier–Lorentzian nonlocal model, was novelly reformulated as two coupled PDEs, resulting in a fast finite element solver [18]. These reformulation techniques were then applied to the construction of a linear nonlocal dielectric model for protein in water [19], and a nonlocal Poisson dielectric model for protein (or other biomolecules) in ionic solvent [20]. Moreover, an effective solution decomposition scheme was proposed in [20] to solve the nonlocal Poisson dielectric model without involving any singularity difficulty. Since this nonlocal Poisson dielectric model depends on a set of ionic concentration functions, different selections of ionic concentration functions may lead to different nonlocal dielectric models. In this sense, this work has established a general framework for us to develop nonlocal dielectric models and related numerical schemes.

The Poisson–Boltzmann equation (PBE) is one widely-used dielectric continuum model for protein in ionic solvent [21–27]. It is actually produced from the classic Poisson dielectric model through constructing the ionic concentration

* Corresponding author.
*E-mail address:* dxie@uwm.edu (D. Xie).

http://dx.doi.org/10.1016/j.jcp.2016.06.028
0021-9991/© 2016 Elsevier Inc. All rights reserved.

functions by the Boltzmann distribution functions [28,29]. Similarly, we can use the Boltzmann distribution functions to generate a nonlinear nonlocal dielectric model from the nonlocal Poisson dielectric model. For clarity, such a new model will be referred to as a nonlocal modified Poisson–Boltzmann equation (NMPBE) since it not only includes PBE as a special case but also can be directly obtained by simply adding nonlocal terms to PBE. Due to the reflection of the spacial-frequency dependence of a dielectric medium, NMPBE is expected to improve the quality of PBE in the prediction of electrostatics of an ionic solvated protein and in some difficult applications such as rational drug design and ion channel study [30,22,31,32]. But, to explore the applications of NMPBE, we need efficient and effective numerical algorithms and software packages for solving NMPBE.

NMPBE is turned out to be very difficult and expensive to solve due to its solution singularity caused by point charge terms, its strong nonlinearity caused by exponential nonlinear terms, and its nonlocal convolution terms. Fortunately, in the last five years, we developed numerical techniques and software packages to solve PBE and several linear nonlocal dielectric models [27,33,18,20,19]. We also developed a biomolecular interface-fitted tetrahedral mesh generator called GAMer-II as an extension of the molecular surface and volumetric mesh generation program package reported in [34]. This previous work provides a strong foundation for us to develop NMPBE numerical algorithms and software packages.

In this paper, we start with an overview on the derivation of our nonlocal Poisson dielectric model. We then use the solution decomposition scheme proposed in [20] to split the NMPBE solution as a sum of three functions: $G$, $\Psi$, and $\tilde{\Phi}$ (see (22)) with $G$ being a given function that collects all the solution singular points. As a result, both $\Psi$ and $\tilde{\Phi}$ become twice continuously differentiable within the solvent and solute regions. Hence, they can be computed numerically without any singularity difficulty. Since an efficient finite element algorithm for computing $\Psi$ was given in [20], one major job to do in this paper is to develop numerical algorithms for solving the nonlinear nonlocal interface problem that defines $\tilde{\Phi}$ (see (25)).

For clarity, we present our NMPBE finite element solver for a protein in a symmetric 1:1 ionic solution while it works for an ionic solution containing any number of ionic species. Especially, we well deal with a possible solution "blow-up" problem caused by the exponentially nonlinear terms according to the function truncation strategy from [27]. We also find the analytical expression of the convolution of $G$ (see (35)), and use it to improve the numerical accuracy of the algorithm for computing $\tilde{\Phi}$ reported in [20]. We next reformulate the nonlinear nonlocal interface problem as a nonlinear variational problem (see (40)), and obtain a Taylor expression (see Theorem 4.1) to yield a new Newton-type iterative algorithm for computing $\tilde{\Phi}$ in a variational form.

Typically, a Newton algorithm for solving a nonlinear and nonlocal dielectric model can be very costly due to each Newton iteration involving the calculation of two convolutions — one for each Newton iterate and the other one for each Newton search direction (see (45)). The novelty of our new modified Newton algorithm lies on the fact that we find an effective recursive formula (see (48)) to update one convolution quickly while treating the other convolution as a unknown function. In this way, each Newton iteration can be done efficiently without any direct calculation of convolutions. Consequently, the costs of computing $\tilde{\Phi}$ can be reduced sharply, resulting in an efficient NMPBE finite element solver.

Since the modified Newton iterative method converges locally [35], its convergence and performance strongly depend on the selection of an initial iterate. In this paper, we construct a good initial iterate from a linear nonlocal interface problem (see (32)). We also construct another initial iterate by using a PBE finite element solution (see (55)). This is plausible since NMPBE is a nonlocal variant of PBE, whose solution may be close to the PBE solution. Furthermore, to define an approximate boundary value function for NMPBE, we propose a nonlocal Debye–Hückel equation (see (36)), and obtain its analytical solution expression (see (36)). Our nonlocal Debye–Hückel equation can be regarded as an extension of the well know Debye–Hückel equation (see (38)), whose analytical solution is often employed as a boundary value function of PBE. Similarly, the analytical solution of our nonlocal Debye–Hückel equation can be selected as a boundary value function of NMPBE.

To validate NMPBE numerical solutions, we construct a nonlinear nonlocal Born test model (see (57)), which has the same analytical solution as the nonlocal point charge Born model given in [19, Page 181, Model A]. Because of an artificial charge source term, the parameters of this Born test model lose their physical senses. One of them can be simply used to adjust the solution range to avoid the solution "blow-up" problem. Its different values may lead to different validation tests.

We programmed our NMPBE numerical algorithm in Python, Fortran, and C++ based on the finite element library `DOLFIN` [36] and our PBE and linear nonlocal model program packages reported in [27,20]. With the Born test model, we numerically validated this NMPBE software package (see Fig. 1). Our NMPBE solver was also verified to retain a quadratic rate of convergence in terms of mesh size for a linear finite element method (see Table 2). We then made numerical experiments on twelve proteins with the number of atoms up to 11439 to demonstrate the high performance of our NMPBE software package (see Table 3). Numerical results showed that our modified Newton iterative method converged fast (see Table 4). For example, in the case of a protein (PDB ID 1A7M) with 2803 atoms, the residual norm of a system of nonlinear finite element equations was reduced from $O(10^6)$ to $O(10^{-7})$ in a total number of 31 iterations. The total CPU time for this test example was only about 1 minute on one 3.7 GHz Intel Xeon E5 processor of a Mac Pro workstation.

As an application, we revisited the calculation of electrostatic solvation and binding free energies from the point view of solution decomposition, resulting in the new formulas for computing electrostatic solvation and binding free energies (see (62) and (64)). We then calculated the electrostatic solvation free energy for the twelve protein molecules and the electrostatic binding free energy for a DNA-drug complex studied chemically in [37]. Numerical results showed the significant differences between the PBE and NMPBE in such calculations. Interestingly, the electrostatic solvation free energy generated by NMPBE was found to be an increasing function of the nonlocal parameter $\lambda$ when $\lambda$ is greater than 2 (see Fig. 2). This promising feature signifies the potential of NMPBE to become a better predictor than PBE in the calculation of solvation and binding free energies. In these tests, the binding free energy change rates computed by NMPBE matched the chemical experimental data given in [37] in a much higher accuracy than the ones computed by PBE. Finally, we calculated the electrostatics of a peptide-RNA complex, which was studied in [23], and the electrostatics of the peptide and RNA components. A comparison of the electrostatics in one binding site before the binding with that after the binding (see Fig. 6) further demonstrated the potential of NMPBE in the study of binding properties.

The remaining sections of the paper are outlined as follows. In Section 2, we present an overview on the derivation of our nonlocal Poisson dielectric model. In Section 3, we describe NMPBE and its solution decomposition scheme. In Section 4, we present the NMPBE finite element solver for the case of a symmetric 1:1 ionic solution. In Section 5, we report the NMPBE program package, numerical results, and application examples. The conclusions are made in Section 6.

## 2. The derivation of our nonlocal Poisson dielectric model

In this section, we present an overview on the derivation of our nonlocal Poisson dielectric model for a protein (or other biomolecules such as RNA and DNA) in an ionic solvent based on our previous work [18,20,19].

We start with the case of the whole space $\mathbb{R}^3$ full of water. Water is a typical dielectric medium. Each water molecule reacts as a dipole under an electrostatic field, $\mathbf{e}$, to yield a dielectric charge density $\gamma(\mathbf{r})$. Together with a fixed charge density $\rho(\mathbf{r})$, the electric field $\mathbf{e}$ can be estimated by Gauss's law in the differential form

$$\epsilon_0 \nabla\cdot\mathbf{e}(\mathbf{r}) = \gamma(\mathbf{r}) + \rho(\mathbf{r}) \quad \text{for } \mathbf{r} = (x, y, z) \in \mathbb{R}^3, \tag{1}$$

where $\epsilon_0$ is the permittivity of the vacuum, and $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$.

To avoid the difficulty of calculating $\gamma$, the classic linear dielectric theory (see [38,39] for example) has been established based on the basic relationships

$$\text{(a)}\quad \mathbf{d}(\mathbf{r}) = \epsilon_0 \varepsilon(\mathbf{r})\mathbf{e}(\mathbf{r}); \quad \text{(b)}\quad \mathbf{p}(\mathbf{r}) = \epsilon_0 \chi(\mathbf{r})\mathbf{e}(\mathbf{r}); \quad \text{(c)}\quad \mathbf{e}(\mathbf{r}) = -\nabla\Phi(\mathbf{r}), \tag{2}$$

where $\mathbf{d}$ and $\mathbf{p}$ denote the displacement and polarization fields, which are defined by

$$\text{(a)}\quad \nabla\cdot\mathbf{d}(\mathbf{r}) = \rho(\mathbf{r}); \qquad \text{(b)}\quad -\nabla\cdot\mathbf{p}(\mathbf{r}) = \gamma(\mathbf{r}), \tag{3}$$

$\varepsilon(\mathbf{r})$ is the dielectric permittivity function, $\chi(\mathbf{r})$ is the susceptibility function, and $\Phi$ is the electrostatic potential function. Applying (2a) and (2c) to (3a) gives the classic Poisson dielectric model:

$$-\nabla\cdot(\varepsilon(\mathbf{r})\nabla\Phi(\mathbf{r})) = \frac{1}{\epsilon_0}\rho(\mathbf{r}) \quad \forall \mathbf{r} \in \mathbb{R}^3 \quad \text{and } \Phi(\mathbf{r}) \to 0 \text{ as } |\mathbf{r}| \to \infty. \tag{4}$$

Experimental data [40] has verified Debye's ansatz that the relationships (2a) and (2b) depend on temporal frequency $\nu$. Since $\nu$ is proportional to spatial wave number $\xi$ for plane waves, the $\nu$-dependent relationships can be imitated in the Fourier wavevector space as follows:

$$\text{(a)}\quad \widehat{\mathbf{d}}(\xi) = \epsilon_0 \widehat{\varepsilon}(\xi)\widehat{\mathbf{e}}(\xi); \qquad \text{(b)}\quad \widehat{\mathbf{p}}(\xi) = \epsilon_0 \widehat{\chi}(\xi)\widehat{\mathbf{e}}(\xi), \tag{5}$$

where $\widehat{\varepsilon}(\xi)$, $\widehat{\chi}(\xi)$, $\widehat{\mathbf{d}}(\xi)$, $\widehat{\mathbf{p}}(\xi)$, and $\widehat{\mathbf{e}}(\xi)$ denote the Fourier transforms of $\varepsilon(\mathbf{r})$, $\chi(\mathbf{r})$, $\mathbf{d}(\mathbf{r})$, $\mathbf{p}(\mathbf{r})$ and $\mathbf{e}(\mathbf{r})$, respectively. The inverse Fourier transform and the convolution property of the Fourier transform lead to the nonlocal electrostatic relationships:

$$\text{(a)}\quad \mathbf{d}(\mathbf{r}) = \epsilon_0 \int_{\mathbb{R}^3} \varepsilon(\mathbf{r}-\mathbf{r}')\mathbf{e}(\mathbf{r}')\,d\mathbf{r}'; \quad \text{(b)}\quad \mathbf{p}(\mathbf{r}) = \epsilon_0 \int_{\mathbb{R}^3} \chi(\mathbf{r}-\mathbf{r}')\mathbf{e}(\mathbf{r}')\,d\mathbf{r}'. \tag{6}$$

Applying (6a) and (2c) to (3a), we obtain the nonlocal Poisson dielectric equation:

$$-\nabla\cdot \int_{\mathbb{R}^3} \varepsilon(\mathbf{r}-\mathbf{r}')\nabla\Phi(\mathbf{r}')\,d\mathbf{r}' = \frac{1}{\epsilon_0}\rho(\mathbf{r}) \quad \forall \mathbf{r} \in \mathbb{R}^3 \quad \text{and } \Phi(\mathbf{r}) \to 0 \text{ as } |\mathbf{r}| \to \infty. \tag{7}$$

Following Debye's temporal frequency dependent function [40] and [41, page 100], we set $\widehat{\varepsilon}(\xi)$ in the expression

$$\widehat{\varepsilon}(\xi) = \epsilon_\infty + \frac{\epsilon_s - \epsilon_\infty}{1 + \lambda^2|\xi|^2}, \tag{8}$$

where $\varepsilon_s$ and $\varepsilon_\infty$ are the static and optic values corresponding to $|\xi| = 0$ and $|\xi| \to \infty$, respectively, $\epsilon_s > \epsilon_\infty$, and $\lambda$ is a parameter for characterizing the polarization correlations of water molecules or the spacial-frequency dependence of a dielectric medium in a more general sense [42]. Note that the function $\widehat{\varepsilon}(\xi)$ of (8) is intended to reflect nonlocal response in the case of zero momentum [43,44]. The scalar parameter $\lambda$ can also be generalized as a matrix to reflect the anisotropy property of a dielectric. Currently, a value of $\lambda$ was mainly determined from experiments. For example, it was found to vary from 3 to 25 for different ionic solvents [18, Figure 2.1] and [16,45].

We then do the inverse Fourier transform on the both sides of (8) to yield the commonly-used kernel function

$$\varepsilon(\mathbf{r}) = \epsilon_\infty \delta + (\epsilon_s - \epsilon_\infty) Q_\lambda(\mathbf{r}) \qquad \text{with} \quad Q_\lambda(\mathbf{r}) = \frac{e^{-|\mathbf{r}|/\lambda}}{4\pi\lambda^2|\mathbf{r}|}, \tag{9}$$

where $\delta$ denotes the Dirac-delta distribution at the origin. Applying the above expression to (7), we obtain the Fourier–Lorentzian nonlocal model for the water solvent:

$$\begin{cases} -\epsilon_0 \left[\epsilon_\infty \Delta\Phi(\mathbf{r}) + (\epsilon_s - \epsilon_\infty)\nabla \cdot \int_{\mathbb{R}^3} Q_\lambda(\mathbf{r}-\mathbf{r}')\nabla\Phi(\mathbf{r}')\,d\mathbf{r}'\right] = \rho(\mathbf{r}), & \mathbf{r} \in \mathbb{R}^3, \\ \Phi(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty. \end{cases} \tag{10}$$

A fast finite element algorithm for solving the above model can be found in [18].

Note that $Q_\lambda$ is also a Yukawa-type kernel [15,46] since it satisfies the equation

$$-\lambda^2 \Delta Q_\lambda(\mathbf{r}) + Q_\lambda(\mathbf{r}) = \delta, \quad \mathbf{r} \in \mathbb{R}^3.$$

Hence, doing the convolution of $\Phi$ on the both sides of the above equation and using the multiplication property of convolution, we can find that the convolution $\Phi * Q_\lambda$ satisfies the equation

$$-\lambda^2 \Delta w(\mathbf{r}) + w(\mathbf{r}) - \Phi(\mathbf{r}) = 0, \qquad \mathbf{r} \in R^3. \tag{11}$$

Here the convolution $\Phi * Q_\lambda$ is defined by

$$(\Phi * Q_\lambda)(\mathbf{r}) = \int\limits_{\mathbb{R}^3} Q_\lambda(\mathbf{r}-\mathbf{r}')\Phi(\mathbf{r}')d\mathbf{r}'.$$

We next consider an extension of the nonlocal Poisson dielectric model (7) to the case of a protein molecule (or other biomolecules such as RNA and DNA) in an ionic solvent — a water solution containing $n$ different ionic species. To do so, we decompose the whole space $\mathbb{R}^3$ as follows:

$$\mathbb{R}^3 = D_p \cup D_s \cup \Gamma,$$

where $D_p$, $D_s$, and $\Gamma$ denote a protein region, a solvent region, and an interface between $D_p$ and $D_s$, respectively. Based on the continuum implicit solvent approach [26], $D_p$ and $D_s$ are treated as continuum dielectric media with two different dielectric constants $\epsilon_p$ and $\epsilon_s$ (or called relative permittivity constants), respectively. We then set $\varepsilon$ as a function of two variables $\mathbf{r}$ and $\mathbf{r}'$ (see [19] for details) in the expression

$$\varepsilon(\mathbf{r}, \mathbf{r}') = \epsilon(\mathbf{r})\delta_{\mathbf{r}'} + \kappa(\mathbf{r}) Q_\lambda(\mathbf{r}-\mathbf{r}'), \tag{12}$$

where $\delta_{\mathbf{r}'}$ denotes the Dirac-delta distribution at $\mathbf{r}' \in \mathbb{R}^3$, which is defined by $\langle \delta_{\mathbf{r}'}, v\rangle = v(\mathbf{r}')$ for any test function $v$, and $\epsilon(\mathbf{r})$ and $\kappa(\mathbf{r})$ are defined by

$$\epsilon(\mathbf{r}) = \begin{cases} \epsilon_p, & \mathbf{r} \in D_p, \\ \epsilon_\infty, & \mathbf{r} \in D_s, \end{cases} \qquad \kappa(\mathbf{r}) = \begin{cases} 0, & \mathbf{r} \in D_p, \\ \epsilon_s - \epsilon_\infty, & \mathbf{r} \in D_s. \end{cases}$$

Since both $\epsilon$ and $\kappa$ are two piecewise functions, the nonlocal Poisson model (7) has to be redefined as an interface problem in order to make sense in differential equations.

To do so, we first use the classic linear dielectric theory [39] to get the interface conditions

$$\Phi(\mathbf{s}^-) = \Phi(\mathbf{s}^+), \qquad \mathbf{d}(\mathbf{s}^-)\cdot\mathbf{n}(\mathbf{s}) = \mathbf{d}(\mathbf{s}^+)\cdot\mathbf{n}(\mathbf{s}) \quad \forall \mathbf{s} \in \Gamma, \tag{13}$$

where $\mathbf{n}(\mathbf{s})$ denotes the unit outward normal vector of $D_p$, $\Phi(\mathbf{s}^\pm) = \lim_{t\to 0^+} \Phi(\mathbf{s} \pm t\mathbf{n}(\mathbf{s}))$, and $\mathbf{d}(\mathbf{s}^\pm) = \lim_{t\to 0^+} \mathbf{d}(\mathbf{s} \pm t\mathbf{n}(\mathbf{s}))$. We then apply (2c) and (12) to (6a) to gain the expressions of $\mathbf{d}$ in $D_p$ and $D_s$, respectively, as follows:

$$\mathbf{d}(\mathbf{r}) = \begin{cases} \epsilon_0\epsilon_p \nabla\Phi(\mathbf{r}), & \mathbf{r} \in D_p, \\ \epsilon_0\epsilon_\infty \nabla\Phi(\mathbf{r}) + \epsilon_0(\epsilon_s - \epsilon_\infty)\int_{\mathbb{R}^3} Q_\lambda(\mathbf{r}-\mathbf{r}')\nabla\Phi(\mathbf{r}')d\mathbf{r}', & \mathbf{r} \in D_s. \end{cases} \tag{14}$$

When a three-dimensional molecular structure of the protein and a concentration function $c_i$ of the $i$th ionic species are given, the charge density $\rho$ can be estimated by

$$\rho(\mathbf{r}) = \begin{cases} e_c \sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ e_c \sum\limits_{i=1}^{n} Z_i c_i(\mathbf{r}), & \mathbf{r} \in D_s, \end{cases} \tag{15}$$

where $n_p$ is the number of atoms of the protein, $\mathbf{r}_j$ and $z_j$ denote the position and charge number of atom $j$, respectively, $Z_i$ is the charge number of ionic species $i$, and $e_c$ is the elementary charge. With (3a), (13), (14), and (15), from (7) we can obtain the nonlocal Poisson dielectric model for protein in ionic solvent proposed in [20] as follows:

$$\begin{cases} -\epsilon_p \Delta\Phi(\mathbf{r}) = \frac{e_c}{\epsilon_0}\sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r}\in D_p, \\ -\epsilon_\infty \Delta\Phi(\mathbf{r}) - (\epsilon_s - \epsilon_\infty)\nabla\cdot \int_{\mathbb{R}^3} Q_\lambda(\mathbf{r}-\mathbf{r}')\nabla\Phi(\mathbf{r}')d\mathbf{r}' = \frac{e_c}{\epsilon_0}\sum\limits_{i=1}^{n} Z_i c_i(\mathbf{r}), & \mathbf{r}\in D_s, \\ \Phi(\mathbf{s}^-) = \Phi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial\Phi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial\Phi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty)\int_{\mathbb{R}^3} Q_\lambda(\mathbf{s}-\mathbf{r}')\nabla\Phi(\mathbf{r}')d\mathbf{r}'\cdot\mathbf{n}(\mathbf{s}), & \mathbf{s}\in\Gamma, \\ \Phi(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}|\to\infty, \end{cases} \tag{16}$$

where $\frac{\partial\Phi(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})} = \nabla\Phi(\mathbf{s})\cdot\mathbf{n}(\mathbf{s})$. By (11) and the multiplication property of convolution, the above interface problem can be further simplified as

$$\begin{cases} -\epsilon_p \Delta\Phi(\mathbf{r}) = \frac{e_c}{\epsilon_0}\sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r}\in D_p, \\ -\epsilon_\infty \Delta\Phi(\mathbf{r}) + \frac{\epsilon_s-\epsilon_\infty}{\lambda^2}[\Phi(\mathbf{r}) - (\Phi * Q_\lambda)(\mathbf{r})] = \frac{e_c}{\epsilon_0}\sum\limits_{i=1}^{n} Z_i c_i(\mathbf{r}), & \mathbf{r}\in D_s, \\ \Phi(\mathbf{s}^-) = \Phi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial\Phi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial\Phi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty)\frac{\partial(\Phi * Q_\lambda)(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s}\in\Gamma, \\ \Phi(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}|\to\infty. \end{cases} \tag{17}$$

The remaining issue is how to select ionic concentration functions $c_i$ for $i = 1, 2, \ldots, n$ to yield a particular nonlocal dielectric model for estimating the electrostatic potential $\Phi$ of an ionic solvated protein.

## 3. A nonlocal modified Poisson–Boltzmann equation

Clearly, different selections of ionic concentration functions $\{c_i\}$ may lead to different nonlocal dielectric continuum models. Note that the Boltzmann distribution function, $c_i(\mathbf{r}) = M_i e^{-\frac{Z_i e_c}{k_B T}\Phi(\mathbf{r})}$ for $i = 1, 2, \ldots, n$, has been known to be an optimal selection in the sense of minimizing a traditional electrostatic free energy [28,29]. Here, $M_i$ denotes a bulk concentration of the $i$th ionic species, $k_B$ is the Boltzmann constant, and $T$ is the absolute temperature. Hence, we can apply it to (17) to yield a nonlinear nonlocal dielectric continuum model as follows:

$$\begin{cases} -\epsilon_p \Delta\Phi(\mathbf{r}) = \frac{e_c}{\epsilon_0}\sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r}\in D_p, \\ -\epsilon_\infty \Delta\Phi(\mathbf{r}) + \frac{\epsilon_s-\epsilon_\infty}{\lambda^2}(\Phi - \Phi * Q_\lambda) - \frac{e_c}{\epsilon_0}\sum\limits_{i=1}^{n} Z_i M_i e^{-Z_i\frac{e_c}{k_B T}\Phi} = 0, & \mathbf{r}\in D_s, \\ \Phi(\mathbf{s}^-) = \Phi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial\Phi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial\Phi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty)\frac{\partial(\Phi * Q_\lambda)(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s}\in\Gamma, \\ \Phi(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}|\to\infty. \end{cases} \tag{18}$$

Clearly, setting $\epsilon_\infty = \epsilon_s$ reduces the above model to the classic PBE model:

$$\begin{cases} -\epsilon_p \Delta\Phi(\mathbf{r}) = \frac{e_c}{\epsilon_0}\sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r}\in D_p, \\ -\epsilon_s \Delta\Phi(\mathbf{r}) - \frac{e_c}{\epsilon_0}\sum\limits_{i=1}^{n} Z_i M_i e^{-Z_i\frac{e_c}{k_B T}\Phi} = 0, & \mathbf{r}\in D_s, \\ \Phi(\mathbf{s}^-) = \Phi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial\Phi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_s \frac{\partial\Phi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})}, & \mathbf{s}\in\Gamma, \\ \Phi(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}|\to\infty. \end{cases} \tag{19}$$

In other words, the nonlocal dielectric model (18) can be regarded as a modification of PBE with the nonlocal terms being added to (19). Hence, we call it the nonlocal modified Poisson–Boltzmann equation (NMPBE) for clarity.

In the SI (Le Système International d'Unités) units, $\Phi$ is measured in volts, and the values and units of physical parameters $\epsilon_0$, $e_c$, $T$, and $k_B$ are given in Table 1. In biomolecular simulation, length is measured in angstroms (Å), and $M_i$ is given in $c_i^b$ moles per liter. Thus, $\lambda$ has a value in angstroms, and $M_i$ is converted to a number density by

$$M_i = N_A c_i^b/liter = 10^3 N_A c_i^b/m^3 = 10^{-27} N_A c_i^b/\text{Å}^3,$$

where $c_i^b$ is a nonnegative scalar number, and $N_A = 6.02214129 \times 10^{23}$ is the Avogadro number.

**Table 1**
Some physical parameters of the NMPBE model (18) in SI units.

| Parameter | Value | Unit (abbr.) | Name |
|---|---|---|---|
| $\epsilon_0$ | $8.854187817 \times 10^{-12}$ | Farad/Meter (F/m) | Permittivity of vacuum |
| $e_c$ | $1.602176565 \times 10^{-19}$ | Coulomb (C) | Elementary charge |
| $T$ | 298.15 | Kelvin (K) | Absolute temperature |
| $k_B$ | $1.380648813 \times 10^{-23}$ | Joule/Kelvin (J/K) | Boltzmann constant |

To simplify the analysis and calculation, we can use the variable change

$$u = \frac{e_c}{k_B T}\Phi, \tag{20}$$

to transform the NMPBE model (18) into the dimensionless form

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \frac{10^{10} e_c^2}{\epsilon_0 k_B T} \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta u + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2}[u - (u * Q_\lambda)] - \frac{10^{-17} N_A e_c^2}{\epsilon_0 k_B T} \sum_{i=1}^{n} Z_i c_i^b e^{-Z_i u} = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty)\frac{\partial (u * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}| \to \infty. & \end{cases} \tag{21}$$

However, the solution of (21) is singular at each atomic position $\mathbf{r}_j$ due to Dirac-delta distributions $\delta_{\mathbf{r}_j}$. To avoid such a singularity difficulty, we use the solution decomposition proposed in [20] to construct the NMPBE solution as

$$u(\mathbf{r}) = \Psi(\mathbf{r}) + \tilde{\Phi}(\mathbf{r}) + G(\mathbf{r}) \quad \forall \mathbf{r} \in \mathbb{R}^3, \tag{22}$$

where $G$ is given by

$$G(\mathbf{r}) = \frac{10^{10} e_c^2}{4\pi \epsilon_0 \epsilon_p k_B T} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|}, \tag{23}$$

$\Psi$ is a solution of the linear nonlocal interface problem

$$\begin{cases} \Delta \Psi(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta \Psi(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2}(\Psi - \Psi * Q_\lambda) = -\frac{\epsilon_s - \epsilon_\infty}{\lambda^2}(G - G * Q_\lambda), & \mathbf{r} \in D_s, \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} - \epsilon_\infty \frac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = (\epsilon_s - \epsilon_\infty)\frac{\partial (\Psi * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} + g_\Gamma(\mathbf{s}), & \mathbf{s} \in \Gamma, \\ \Psi(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}| \to \infty, & \end{cases} \tag{24}$$

and $\tilde{\Phi}$ is a solution of the nonlinear nonlocal interface problem

$$\begin{cases} \Delta \tilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta \tilde{\Phi} + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2}(\tilde{\Phi} - \tilde{\Phi} * Q_\lambda) - \frac{10^{-17} N_A e_c^2}{\epsilon_0 k_B T} \sum_{i=1}^{n} Z_i c_i^b e^{-Z_i(\Psi + G)} e^{-Z_i \tilde{\Phi}} = 0, & \mathbf{r} \in D_s, \\ \tilde{\Phi}(\mathbf{s}^-) = \tilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} - \epsilon_\infty \frac{\partial \tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = (\epsilon_s - \epsilon_\infty)\frac{\partial (\tilde{\Phi} * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \tilde{\Phi}(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}| \to \infty. & \end{cases} \tag{25}$$

Here $g_\Gamma$ is defined by

$$g_\Gamma(\mathbf{s}) = (\epsilon_s - \epsilon_\infty)\frac{\partial (G * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_\infty - \epsilon_p)\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}. \tag{26}$$

Clearly, the equations of (24) and (25) imply that $\Psi$ and $\tilde{\Phi}$ are twice continuously differentiable within the protein and solvent regions $D_p$ and $D_s$. Hence, they can be solved numerically by either a finite difference method or a finite element method.

## 4. The NMPBE finite element solver for symmetric 1:1 ionic solution

In this section, we present a NMPBE finite element solver for a symmetric 1:1 ionic solution (e.g., a salt solution containing sodium ($\mathrm{Na}^+$) and chloride ($\mathrm{Cl}^-$) ions). We choose this case not only for simplicity of presentation but also for its wide application in biomolecular simulation and bioengineering application. This solver can be extended to the general case in which a solution contains any number of ionic species.

In the case of a symmetric 1:1 ionic solution, we have that $n = 2$, $Z_1 = 1$, $Z_2 = -1$, and $c_1^b = c_2^b = I_s$ with $I_s$ denoting the ionic solvent strength [27]. Thus, (21) can be simplified as

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \frac{10^{10} e_c^2}{\epsilon_0 k_B T} \sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta u(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} [u(\mathbf{r}) - (u * Q_\lambda)(\mathbf{r})] + \kappa^2 \sinh(u(\mathbf{r})) = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty) \frac{\partial (u * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{r}) \to 0 \quad \text{as } |\mathbf{r}| \to \infty, & \end{cases} \tag{27}$$

where $\kappa^2$ is defined by

$$\kappa^2 = 2 \frac{10^{-17} N_A e_c^2}{\epsilon_0 k_B T} I_s. \tag{28}$$

With a bounded domain, $\Omega$, satisfying that $D_p \subset \Omega$, we define a boundary value problem by

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \frac{10^{10} e_c^2}{\epsilon_0 k_B T} \sum\limits_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta u(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} [u(\mathbf{r}) - (u * Q_\lambda)(\mathbf{r})] + \kappa^2 \sinh(u) = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty) \frac{\partial (u * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = g(\mathbf{s}), & \mathbf{s} \in \partial\Omega, \end{cases} \tag{29}$$

where $D_s = \Omega \setminus (D_p \cup \Gamma)$, $\partial\Omega$ is the boundary of $\Omega$, $g$ is a boundary value function, and $\kappa$ is defined in (28). The solution of the above boundary value problem is then constructed by the solution decomposition formula (22) with $\Psi$ and $\tilde{\Phi}$ being the solutions of the following two boundary value problems:

$$\begin{cases} \Delta \Psi(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta \Psi(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} (\Psi - \Psi * Q_\lambda) = -\frac{\epsilon_s - \epsilon_\infty}{\lambda^2} (G - G * Q_\lambda), & \mathbf{r} \in D_s, \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} - \epsilon_\infty \frac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = (\epsilon_s - \epsilon_\infty) \frac{\partial (\Psi * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} + g_\Gamma(\mathbf{s}), & \mathbf{s} \in \Gamma, \\ \Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s}), & \mathbf{s} \in \partial\Omega, \end{cases} \tag{30}$$

and

$$\begin{cases} \Delta \tilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta \tilde{\Phi}(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} (\tilde{\Phi} - \tilde{\Phi} * Q_\lambda) + \kappa^2 \sinh(\tilde{\Phi} + \Psi + G) = 0, & \mathbf{r} \in D_s, \\ \tilde{\Phi}(\mathbf{s}^-) = \tilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} - \epsilon_\infty \frac{\partial \tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = (\epsilon_s - \epsilon_\infty) \frac{\partial (\tilde{\Phi} * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \tilde{\Phi}(\mathbf{s}) = 0, & \mathbf{s} \in \partial\Omega. \end{cases} \tag{31}$$

Here $g_\Gamma$ has been defined in (26), and the boundary values function of (31) has been set to zero for the purpose of simplifying the calculation of $\tilde{\Phi}$. It has been proved that both (30) and (31) are well posed in [47].

Clearly, when $u$ is small enough, the nonlinear boundary value problem (31) can be linearized as

$$\begin{cases} \Delta \tilde{\Phi}_l(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_\infty \Delta \tilde{\Phi}_l(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} (\tilde{\Phi}_l - \tilde{\Phi}_l * Q_\lambda) + \kappa^2 \tilde{\Phi}_l(\mathbf{r}) = -\kappa^2 [G(\mathbf{r}) + \Psi(\mathbf{r})], & \mathbf{r} \in D_s, \\ \tilde{\Phi}_l(\mathbf{s}^-) = \tilde{\Phi}_l(\mathbf{s}^+), \quad \epsilon_p \frac{\partial \tilde{\Phi}_l(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} - \epsilon_\infty \frac{\partial \tilde{\Phi}_l(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = (\epsilon_s - \epsilon_\infty) \frac{\partial (\tilde{\Phi}_l * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \tilde{\Phi}_l(\mathbf{s}) = 0, & \mathbf{s} \in \partial\Omega, \end{cases} \tag{32}$$

whose solution can be selected as an initial guess to the solution of the nonlinear problem (31).

### *4.1. Scheme for computing $\Psi$*

Let $\mathcal{M}$ be a Lagrange finite element function space defined on a tetrahedral mesh of $\Omega$ such that $\mathcal{M} \subset H^1(\Omega)$ and each function of $\mathcal{M}$ is continuous. We then set

$$\mathcal{M}_0 = \{v \in \mathcal{M} \mid v = 0 \text{ on } \partial\Omega\} \subset H_0^1(\Omega).$$

Here $H^1(\Omega)$ and $H_0^1(\Omega)$ are two usual Sobolev function spaces [48]. The convolutions of $\Psi$, $g$, and $G$ with $Q_\lambda$ will be denoted by $u_1$, $\hat{g}$, and $\hat{G}$, respectively.

A finite element solution, $\Psi$, of the boundary value problem (30) can be found from solving the linear variational problem: Find $\underset{\sim}{\phi} = (\Psi, u_1) \in \mathcal{M} \times \mathcal{M}$ satisfying $\Psi(\mathbf{s}) = g(\mathbf{s}) - G(\mathbf{s})$ and $u_1(\mathbf{s}) = \hat{g}(\mathbf{s}) - \hat{G}(\mathbf{s})$ for all $\mathbf{s} \in \partial\Omega$ such that

$$a(\underset{\sim}{\phi}, \underset{\sim}{v}) = \ell_1(\underset{\sim}{v}) \qquad \forall \underset{\sim}{v} = (v_1, v_2) \in \mathcal{M}_0 \times \mathcal{M}_0, \tag{33}$$

where $\ell_1(\underset{\sim}{v}) = (\epsilon_\infty - \epsilon_s) \int_{D_s} \nabla \hat{G}(\mathbf{r}) \cdot \nabla v_1(\mathbf{r}) d\mathbf{r} + (\epsilon_p - \epsilon_\infty) \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v_1(\mathbf{r}) d\mathbf{r}$, and

$$\begin{aligned} a(\underset{\sim}{\phi}, \underset{\sim}{v}) = {} & \epsilon_p \int\limits_{D_p} \nabla\Psi \cdot \nabla v_1 d\mathbf{r} + \epsilon_\infty \int\limits_{D_s} \nabla\Psi \cdot \nabla v_1 d\mathbf{r} + (\epsilon_s - \epsilon_\infty) \int\limits_{D_s} \nabla u_1 \cdot \nabla v_1 d\mathbf{r} \\ & + \lambda^2 \int\limits_{\Omega} \nabla u_1 \cdot \nabla v_2 d\mathbf{r} + \int\limits_{\Omega} (u_1 - \Psi) v_2 d\mathbf{r}. \end{aligned} \tag{34}$$

In [20], an algorithm was proposed to solve (33) with $\hat{G}$ being calculated approximately as a solution of a boundary value problem. To improve this algorithm, using [19, Corollary A.1, Page 192], we can obtain the expression of $\hat{G}$ as

$$\hat{G}(\mathbf{r}) = \frac{10^{10} e_c^2}{4\pi \epsilon_0 \epsilon_p k_B T} \sum_{j=1}^{n_p} z_j \frac{1 - e^{-|\mathbf{r}-\mathbf{r}_j|/\lambda}}{|\mathbf{r} - \mathbf{r}_j|}. \tag{35}$$

We then can find the analytical expression of $\nabla\hat{G}$. Thus, we can calculate both $\hat{G}$ and $\nabla\hat{G}$ analytically in the algorithm for solving $\Psi$. Consequently, a more accurate numerical solution can be produced from our implementation than the one reported in [20].

### 4.2. Selection of boundary value functions

How to select the boundary value function $g$ and its convolution $\hat{g}$ is a research issue by itself. One simple selection is to set $g = \hat{g} = 0$ for a sufficiently large domain $\Omega$ based on the fact that $u(\mathbf{r}) \to 0$ and $(u * Q_\lambda)(\mathbf{r}) \to 0$ as $|\mathbf{r}| \to \infty$.

In the case of PBE, $g$ is commonly set as an analytical solution of the Debye–Hückel equation [31,49]. We extended this selection to the nonlocal case by constructing a nonlocal Debye–Hückel equation as follows:

$$-\epsilon_\infty \Delta u(\mathbf{r}) + \frac{\epsilon_s - \epsilon_\infty}{\lambda^2} [u(\mathbf{r}) - (u * Q_\lambda)(\mathbf{r})] + \kappa^2 u(\mathbf{r}) = \frac{10^{10} e_c^2}{\epsilon_0 k_B T} \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, \quad \mathbf{r} \in \mathbb{R}^3, \tag{36}$$

where $u(\mathbf{r}) \to 0$ as $|\mathbf{r}| \to \infty$, and $\kappa$ is given in (28). We then obtain its solution $u$ and convolution $\hat{u} = u * Q_\lambda$ as follows:

$$\begin{aligned} u(\mathbf{r}) = & \frac{10^{10} e_c^2}{4\pi \epsilon_\infty (\tau_2 - \tau_1) \epsilon_0 k_B T} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|} \left( \tau_2 e^{-\eta_1 |\mathbf{r}-\mathbf{r}_j|} - \tau_1 e^{-\eta_2 |\mathbf{r}-\mathbf{r}_j|} \right), \\ \hat{u}(\mathbf{r}) = & \frac{10^{10} e_c^2 \tau_1 \tau_2}{4\pi \epsilon_\infty (\tau_2 - \tau_1) \epsilon_0 k_B T} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|} \left( e^{-\eta_1 |\mathbf{r}-\mathbf{r}_j|} - e^{-\eta_2 |\mathbf{r}-\mathbf{r}_j|} \right), \end{aligned} \tag{37}$$

where $\tau_1$, $\tau_2$, $\eta_1$, and $\eta_2$ are defined in terms of $\xi = \sqrt{(\kappa^2\lambda^2 + \epsilon_s)^2 - 4\epsilon_\infty \lambda^2 \kappa^2}$ as follows:

$$\tau_1 = \frac{\kappa^2\lambda^2 + \epsilon_s - 2\epsilon_\infty - \xi}{2(\epsilon_s - \epsilon_\infty)}, \quad \tau_2 = \frac{\kappa^2\lambda^2 + \epsilon_s - 2\epsilon_\infty + \xi}{2(\epsilon_s - \epsilon_\infty)}, \quad \eta_1 = \frac{1}{\lambda}\sqrt{\frac{\kappa^2\lambda^2 + \epsilon_s + \xi}{2\epsilon_\infty}}, \quad \eta_2 = \frac{1}{\lambda}\sqrt{\frac{\kappa^2\lambda^2 + \epsilon_s - \xi}{2\epsilon_\infty}}.$$

The analytical expressions of (37) can be used to construct the boundary value functions $g$ and $\hat{g}$.

Specially, when $\epsilon_\infty = \epsilon_s$, the nonlocal Debye–Hückel equation (36) can be reduced to the traditional Debye–Hückel equation

$$-\epsilon_s \Delta u(\mathbf{r}) + \kappa^2 u(\mathbf{r}) = \frac{10^{10} e_c^2}{\epsilon_0 k_B T} \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}, \quad \mathbf{r} \in \mathbb{R}^3. \tag{38}$$

Furthermore, we can get that $\tau_1 \to 1$, $\tau_2 \to +\infty$, $\eta_1 \to \kappa/\sqrt{\epsilon_s}$, and $\eta_2 \to 1/\lambda$ as $\epsilon_\infty \to \epsilon_s$. Thus, the analytical solution of (38) can be followed from (37) in the expression

$$u = \frac{10^{10} e_c^2}{4\pi \epsilon_s \epsilon_0 k_B T} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|} e^{-\frac{\kappa}{\sqrt{\epsilon_s}} |\mathbf{r}-\mathbf{r}_j|}. \tag{39}$$

Here we have omitted the details of calculation due to the length.

### 4.3. A modified Newton method for computing $\tilde{\Phi}$

We suppose that $U = G + \Psi$ is given before the calculation of $\tilde{\Phi}$. The nonlinear nonlocal boundary value problem (31) can be reformulated as the nonlinear variational problem:

$$\text{Find } \tilde{\Phi} \in \mathcal{M}_0 \text{ such that } \quad b(\tilde{\Phi}, v) = 0 \qquad \forall v \in \mathcal{M}_0, \tag{40}$$

where $b(\tilde{\Phi}, v)$ is defined by

$$\begin{aligned} b(\tilde{\Phi}, v) = {} & \epsilon_p \int\limits_{D_p} \nabla\tilde{\Phi}(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} + \epsilon_\infty \int\limits_{D_s} \nabla\tilde{\Phi}(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} \\ & + (\epsilon_s - \epsilon_\infty) \int\limits_{D_s} \nabla(\tilde{\Phi} * Q_\lambda) \cdot \nabla v d\mathbf{r} + \kappa^2 \int\limits_{D_s} \sinh(\tilde{\Phi} + U) v d\mathbf{r}. \end{aligned} \tag{41}$$

In [47], the nonlinear problem (40) has been proved to have a unique solution.

The following theorem sets up a base for constructing a Newton-like iterative method.

**Theorem 4.1.** *Let $b(\tilde{\Phi}, v)$ be defined in (41) on a finite element space $\mathcal{M}_0 \subset H_0^1(\Omega)$. For each given $v \in \mathcal{M}_0$, $b(\tilde{\Phi}, v)$ is Fréchet differentiable at $\tilde{\Phi} \in \mathcal{M}_0$, and its Fréchet derivative $b'(\tilde{\Phi}, v; p)$ at $p \in \mathcal{M}_0$ is given by*

$$\begin{aligned} b'(\tilde{\Phi}, v; p) = {} & \epsilon_p \int\limits_{D_p} \nabla p(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} + \epsilon_\infty \int\limits_{D_s} \nabla p(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} \\ & + (\epsilon_s - \epsilon_\infty) \int\limits_{D_s} \nabla(p * Q_\lambda) \cdot \nabla v d\mathbf{r} + \kappa^2 \int\limits_{D_s} p(\mathbf{r}) v(\mathbf{r}) \cosh(\tilde{\Phi} + U) d\mathbf{r}. \end{aligned} \tag{42}$$

*Moreover, $b(\Phi + p, v)$ has the Taylor expansion*

$$b(\tilde{\Phi} + p, v) = b(\tilde{\Phi}, v) + b'(\tilde{\Phi}, v; p) + O(\|p\|^2_{H^1(\Omega)}). \tag{43}$$

**Proof.** Set $\varphi(t) = b(\tilde{\Phi} + tp, v)$ for a real number $t$. Its first and second derivatives, denoted by $\varphi'$ and $\varphi''$ respectively, can be found as follows:

$$\varphi'(t) = \epsilon_p \int\limits_{D_p} \nabla p \cdot \nabla v d\mathbf{r} + \epsilon_\infty \int\limits_{D_s} \nabla p \cdot \nabla v d\mathbf{r} + (\epsilon_s - \epsilon_\infty) \int\limits_{D_s} \nabla(p * Q_\lambda) \cdot \nabla v d\mathbf{r} + \kappa^2 \int\limits_{D_s} pv \cosh(\tilde{\Phi} + U + tp) d\mathbf{r},$$

$$\varphi''(t) = \kappa^2 \int\limits_{D_s} ppv \sinh(\tilde{\Phi} + U + tp) d\mathbf{r}.$$

Since $\varphi(t)$ is twice continuously differentiable at $t = 0$, Taylor's Theorem follows that there exists a number $\theta$ between 0 and 1 such that

$$\varphi(1) = \varphi(0) + \varphi'(0) + \frac{1}{2}\varphi''(\theta). \tag{44}$$

Since $\tilde{\Phi}, U, p \in \mathcal{M}_0$, $\tilde{\Phi} + U + \theta p$ is continuous for any $\theta \in (0, 1)$. Thus, there exists a constant, $C$, independent of $p$, such that

$$\left| \int\limits_{D_s} p(\mathbf{r}) p(\mathbf{r}) v(\mathbf{r}) \sinh(\tilde{\Phi} + U + \theta p) d\mathbf{r} \right| \le C \|p\|^2_{H^1(\Omega)}.$$

Clearly, $\varphi(1) = b(\tilde{\Phi} + p, v)$, $\varphi(0) = b(\tilde{\Phi}, v)$, and $\varphi'(0) = b'(\tilde{\Phi}, v; p)$. With (44), we get

$$\frac{|b(\Phi + p, v) - b(\Phi, v) - b'(\tilde{\Phi}, v; p)|}{\|p\|_{H^1(\Omega)}} \le C \|p\|_{H^1(\Omega)},$$

from which we get

$$\lim_{\|p\|_{H^1(\Omega)} \to 0} \frac{|b(\Phi + p, v) - b(\Phi, v) - b'(\tilde{\Phi}, v; p)|}{\|p\|_{H^1(\Omega)}} = 0,$$

implying that $b(\Phi, v)$ is Fréchet differentiable, and $b'(\tilde{\Phi}, v; p)$ is its Fréchet derivative. Hence, (43) is followed from (44). This completes the proof. □

We now define the modified Newton iterative method for solving (40) as a sequence, $\{\tilde{\Phi}^{(k)}\}$, by induction. Suppose that an initial guess $\tilde{\Phi}^{(0)}$ is given. When the $k$th iterate $\tilde{\Phi}^{(k)}$ is known, and $\|p\|_{H^1(\Omega)}$ is sufficiently small, by (43), we get

$$b(\tilde{\Phi}^{(k)} + p, v) \approx b(\tilde{\Phi}^{(k)}, v) + b'(\tilde{\Phi}^{(k)}, v; p), \quad k = 0, 1, 2, \ldots,$$

from which we obtain the Newton equation in the linear variational problem:

$$\text{Find } p_k \in \mathcal{M}_0 \text{ such that } \quad b'(\tilde{\Phi}^{(k)}, v; p_k) = -b(\tilde{\Phi}^{(k)}, v) \qquad \forall v \in \mathcal{M}_0. \tag{45}$$

With the above solution $p_k$, we define $\tilde{\Phi}^{(k+1)}$ by

$$\tilde{\Phi}^{(k+1)} = \tilde{\Phi}^{(k)} + \omega_k p_k \quad \text{for } k = 0, 1, 2, \ldots, \tag{46}$$

where $\omega_k$ is a steplength determined by a line search algorithm such that $\tilde{\Phi}^{(k+1)}$ is a better approximation than $\tilde{\Phi}^{(k)}$ in the following sense:

$$\|F(\tilde{\Phi}^{(k+1)})\| \le \|F(\tilde{\Phi}^{(k)})\|, \quad k = 0, 1, 2, \ldots.$$

Here we have written the finite element equation (40) in the vector form

$$F(\tilde{\Phi}) = \mathbf{0} \quad \text{with } F = (b(\tilde{\Phi}, \varphi_1), b(\tilde{\Phi}, \varphi_2), \ldots, b(\tilde{\Phi}, \varphi_{N_h})),$$

with $N_h$ being the total number of interior mesh nodes and $\varphi_i$ denoting the $i$th basis function of $\mathcal{M}_0$. Thus, $\|F(\tilde{\Phi}^{(k)})\|$ is a residual norm of the above system defined by

$$\|F(\tilde{\Phi}^{(k)})\| = \left[ \sum_{i=1}^{N_h} |b(\tilde{\Phi}^{(k)}, \varphi_i)|^2 \right]^{1/2}. \tag{47}$$

In the line search algorithm, we always start with $\omega_k = 1$.

However, each Newton equation of (45) is very expensive to solve due to its two convolution terms: $p_k * Q_\lambda$ and $\tilde{\Phi}^{(k)} * Q_\lambda$. To speedup its numerical solution, we set

$$q_k = p_k * Q_\lambda, \qquad \zeta^{(k)} = \tilde{\Phi}^{(k)} * Q_\lambda,$$

and treat $q_k$ as an unknown function. Doing the convolution with $Q_\lambda$ on the both sides of (46), we obtain the recursive formula for computing the sequence $\{\zeta^{(k)}\}$:

$$\zeta^{(k+1)} = \zeta^{(k)} + \omega_k q_k \quad \text{for } k = 0, 1, 2, \ldots, \tag{48}$$

where $\zeta^{(0)}$ is an initial guess to be given (see Section 4.3 for a selection).

To determine $q_k$, we observe that $q_k$ satisfies the finite element equation,

$$\lambda^2 \int_\Omega \nabla q_k(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} + \int_\Omega (q_k(\mathbf{r}) - p_k(\mathbf{r})) v(\mathbf{r}) d\mathbf{r} = 0 \qquad \forall v \in \mathcal{M}_0. \tag{49}$$

We then use the above fact to modify the Newton equation (45) as a variational system without involving any convolution as follows:

$$\text{Find } \underset{\sim}{p_k} = (p_k, q_k) \in \mathcal{M}_0 \times \mathcal{M}_0 \text{ such that } A(\underset{\sim}{p_k}, \underset{\sim}{v}) = L(\underset{\sim}{v}) \quad \forall \underset{\sim}{v} \in \mathcal{M}_0 \times \mathcal{M}_0, \tag{50}$$

where $A(\underset{\sim}{p_k}, \underset{\sim}{v})$ and $L(\underset{\sim}{v})$ are defined by

$$\begin{aligned} A(\underset{\sim}{p_k}, \underset{\sim}{v}) = {} & \epsilon_p \int_{D_p} \nabla p_k(\mathbf{r}) \cdot \nabla v_1(\mathbf{r}) d\mathbf{r} + \epsilon_\infty \int_{D_s} \nabla p_k(\mathbf{r}) \cdot \nabla v_1(\mathbf{r}) d\mathbf{r} \\ & + (\epsilon_s - \epsilon_\infty) \int_{D_s} \nabla q_k(\mathbf{r}) \cdot \nabla v_1(\mathbf{r}) d\mathbf{r} + \lambda^2 \int_\Omega \nabla q_k(\mathbf{r}) \cdot \nabla v_2(\mathbf{r}) d\mathbf{r}, \\ & + \int_\Omega (q_k(\mathbf{r}) - p_k(\mathbf{r})) v_2(\mathbf{r}) d\mathbf{r} + \kappa^2 \int_{D_s} p_k v_1 \cosh(\tilde{\Phi}^{(k)} + U) d\mathbf{r}, \end{aligned}$$

$$L(\underset{\sim}{v}) = -\Big[(\epsilon_s - \epsilon_\infty)\int_{D_s} \nabla\zeta^{(k)}\cdot\nabla v_1 d\mathbf{r} + \epsilon_p \int_{D_p} \nabla\tilde{\Phi}^{(k)}(\mathbf{r})\cdot\nabla v_1(\mathbf{r})d\mathbf{r}$$
$$+\epsilon_\infty \int_{D_s} \nabla\tilde{\Phi}^{(k)}(\mathbf{r})\cdot\nabla v_1(\mathbf{r})d\mathbf{r} + \kappa^2\int_{D_s} v_1 \sinh(\tilde{\Phi}^{(k)}+U)d\mathbf{r}\Big].$$

Consequently, we can solve each modified Newton equation of (50) by a fast numerical algorithm, resulting in an efficient modified Newton iterative method.

From [20] it can be inferred that the modified Newton equation (50) has a unique solution so that our modified Newton iterative method is well defined.

In the calculation of $A(\cdot,\cdot)$ and $L(\cdot)$, the hyperbolic functions sinh and cosh may become huge when the iterates $\tilde{\Phi}^{(k)}$ are far away from the solution, which may cause a solution “blow-up” problem. Following what was done in [27], we use the following modified hyperbolic functions to simply deal with this problem:

$$\widehat{\sinh}(u) = \begin{cases} \sinh(u) & \text{if } |u| < \tau, \\ \sinh(\tau) & \text{if } u \geq \tau, \\ -\sinh(\tau) & \text{if } u \leq -\tau, \end{cases} \quad \widehat{\cosh}(u) = \begin{cases} \cosh(u) & \text{if } |u| < \tau, \\ \cosh(\tau) & \text{if } |u| \geq \tau, \end{cases} \tag{51}$$

where $\tau$ is a truncation parameter ($\tau = 85$ by default, which was set according to our numerical experience [27]).

Finally, we control the convergence of the modified Newton iterative algorithm using the termination rule

$$\|\tilde{\Phi}^{(k+1)} - \tilde{\Phi}^{(k)}\| < \sigma \quad \text{and} \quad \|\zeta^{(k+1)} - \zeta^{(k)}\| < \sigma, \tag{52}$$

where $\sigma$ is a convergence tolerance ($\sigma = 10^{-7}$ by default).

In conclusion, we have defined the modified Newton iterative method for solving the nonlinear nonlocal boundary value problem (31) as a sequence of pair iterates, $\tilde{\Phi}^{(k)}$ and $\zeta^{(k)}$, which are defined by (46) and (48), respectively, with $p_k$ and $q_k$ being a solution of the linear variational system (50).

### *4.4. Selection of initial iterates*

The efficiency of the modified Newton iterative method strongly depends on a selection of initial iterates $\tilde{\Phi}^{(0)}$ and $\zeta^{(0)}$. For a weak ionic solvent, we may set $\tilde{\Phi}^{(0)} = \zeta^{(0)} = 0$ since setting $\tilde{\Phi} = 0$ leads to a solution of a nonlocal dielectric model for biomolecule in water (i.e., (29) using $I_s = 0$).

Another selection is to set

$$\tilde{\Phi}^{(0)} = \tilde{\Phi}_l, \qquad \zeta^{(0)} = u_l, \tag{53}$$

where $\tilde{\Phi}_l$ and $u_l$ satisfy the linear variational system: Find $\underset{\sim}{\Phi_l} = (\tilde{\Phi}_l, u_l) \in \mathcal{M}_0 \times \mathcal{M}_0$ such that

$$a(\underset{\sim}{\Phi_l}, \underset{\sim}{v}) + \kappa^2 \int_{D_s} \tilde{\Phi}_l v_1 d\mathbf{r} = -\kappa^2 \int_{D_s} U v_1 d\mathbf{r} \quad \forall \underset{\sim}{v} = (v_1, v_2) \in \mathcal{M}_0 \times \mathcal{M}_0. \tag{54}$$

Here $a(\cdot,\cdot)$ has been defined in (34). The above variational system is a modification of the weak form of the linear nonlocal model (32) to avoid the calculation of convolutions.

We can also construct an initial iterate from a finite element solution, $u_{PBE}$, of PBE as follows. Since NMPBE is a modification of PBE, $u_{PBE}$ may be close to $u$. Thus, using (22) and $u_{PBE} = G + \Psi_{PBE} + \tilde{\Phi}_{PBE}$, we can get

$$\tilde{\Phi} \approx \tilde{\Phi}_{PBE} + \Psi_{PBE} - \Psi.$$

Hence, we can construct an initial guess $\tilde{\Phi}^{(0)}$ by

$$\tilde{\Phi}^{(0)} = \tilde{\Phi}_{PBE} + \Psi_{PBE} - \Psi. \tag{55}$$

To calculate the convolution $\zeta^{(0)}$ fast, we set

$$\zeta^{(0)} = w, \tag{56}$$

where $w \in \mathcal{M}_0$ is a finite element solution of the following variational problem:

$$\lambda^2 \int_\Omega \nabla w \cdot \nabla v d\mathbf{r} + \int_\Omega w v d\mathbf{r} = \int_\Omega \tilde{\Phi}^{(0)}(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} \qquad \forall v \in \mathcal{M}_0.$$

For clarity, we summarize the framework of our NMPBE finite element solver in Algorithm 1. Here we have set the initial guess constructed from PBE as a default choice since it was found to result in better performance than the other two choices in our numerical tests (see Table 4 in Section 5.2 for example).

**Algorithm 1** *(NMPBE finite element solver for symmetric 1:1 ionic solution).* Let $u$ be a finite element solution of the NMPBE model (29). It is calculated in the following six steps:

Step 1. Calculate $G$, $\hat{G}$, $\nabla G$, and $\nabla \hat{G}$ via formulas (23) and (35).
Step 2. Set the boundary value functions $g$ and $\hat{g}$ (by default, $g = \hat{g} = 0$).
Step 3. Solve the linear variational system (33) for $\Psi$.
Step 4. Select the initial iterates $\tilde{\Phi}^{(0)}$ and $\zeta^{(0)}$ (by default, use (55) and (56)).
Step 5. Solve the nonlinear variational problem (40) for $\tilde{\Phi}$ by our modified Newton iterative method with the iteration termination rule (52).
Step 6. Find $u$ by the solution decomposition: $u = \tilde{\Phi} + \Psi + G$.

## 5. Program package and numerical results

We developed a NMPBE program package according to Algorithm 1 in Python, Fortran, and C++ based on the finite element library DOLFIN [36], our PBE program package SDPBS [27,33], and our nonlocal linearized PBE software package [20]. Fortran subroutines were written for calculating $G$, $\hat{G}$, $\nabla G$, and $\nabla \hat{G}$ and converted as Python modules by the Fortran-to-Python interface generator f2py (*http://cens.ioc.ee/projects/f2py2e/*). We adopted the molecular surface fitted tetrahedral mesh generator GAMer-II from SDPBE to the NMPBE program package for generating a finite element tetrahedral mesh $\Omega_h$ of a cubic (or spherical) domain $\Omega$. A PQR file of a protein is an input file for implementation. It can be generated from a PDB file of the protein by using the program tool PDB2PQR (*http://www.poissonboltzmann.org/pdb2pqr/*) [50]. A PDB file can be downloaded from the Protein Data Bank (*http://www.rcsb.org/*).

In the numerical experiments, we used the parameters $\epsilon_p = 2$, $\epsilon_s = 80$, $\epsilon_\infty = 1.8$, $T = 298.15$, $I_s = 0.1$, the physical constants of Table 1, $\tau = 85$ for the "blow-up" test, and $\sigma = 10^{-7}$ for the convergence test of (52). Each related linear algebraic system was solved approximately either by GMRES (for an unsymmetrical system) or by the preconditioned conjugate gradient method (for a symmetrical system) from the PETSc library (*http://www.mcs.anl.gov/petsc*). Here, the relative residue error and the absolute residue error were set as $10^{-8}$, and the incomplete LU preconditioning was used. All the numerical experiments were made on one 3.7 GHz Intel Xeon E5 processor of Mac Pro with 64 GB of 1866 MHz memory.

### *5.1. Validation tests*

To validate our NMPBE numerical solutions, we modify a nonlocal point charge Born model given in [19, Page 181, Model A] as the following nonlinear nonlocal Born test model:

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha z \delta & \text{in } D_p, \\ -\epsilon_\infty \Delta u + \dfrac{\epsilon_s - \epsilon_\infty}{\lambda^2}[u - (u * Q_\lambda)(\mathbf{r})] + \kappa^2 \sinh(u(\mathbf{r})) = f(\mathbf{r}) & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_\infty \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_\infty)\frac{\partial (u * Q_\lambda)(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ u(\mathbf{r}) = g(\mathbf{s}) & \text{on } \partial\Omega, \end{cases} \tag{57}$$

where $\alpha$ is a constant, $z$ is a charge number, $D_p = \{\mathbf{r} \mid |\mathbf{r}| < a\}$, $\Gamma = \{\mathbf{r} \mid |\mathbf{r}| = a\}$, and $f$ is given by

$$f(\mathbf{r}) = \kappa^2 \sinh(g(\mathbf{r})) \text{ with } g(\mathbf{r}) = \frac{\alpha z}{4\pi \epsilon_p \epsilon_s |\mathbf{s}|}\left[\epsilon_p - (\epsilon_\infty - \epsilon_s) b_1 e^{\mu(a - |\mathbf{r}|)}\right].$$

From [19, Pages 182–183] we can obtain the analytical solution of (57):

$$u(\mathbf{r}) = \begin{cases} \frac{\alpha z}{4\pi \epsilon_p}\left(\frac{1}{|\mathbf{r}|} + \frac{1}{a\epsilon_s}[\epsilon_p - \epsilon_s - (\epsilon_\infty - \epsilon_s) b_1]\right) & \text{in } D_p, \\ \frac{\alpha z}{4\pi \epsilon_p \epsilon_s |\mathbf{r}|}\left[\epsilon_p - (\epsilon_\infty - \epsilon_s) b_1 e^{\mu(a - |\mathbf{r}|)}\right] & \text{in } D_s, \end{cases} \tag{58}$$

and its convolution $\hat{u} = u * Q_\lambda$ in the expression

$$\hat{u}(\mathbf{r}) = \frac{\alpha z}{4\pi \epsilon_p \epsilon_s |\mathbf{r}|}\left(\epsilon_p - \epsilon_\infty b_1 e^{\mu(a - |\mathbf{r}|)}\right), \quad \mathbf{r} \in D_s, \tag{59}$$

where $\mu$ and $b_1$ are defined by

$$\mu = \frac{1}{\lambda}\sqrt{\frac{\epsilon_s}{\epsilon_\infty}}, \quad b_1 = \frac{a\epsilon_s + \lambda(\epsilon_p - \epsilon_s)\sinh\frac{a}{\lambda}}{[a\sqrt{\epsilon_\infty \epsilon_s} + \lambda(\epsilon_\infty - \epsilon_s)]\sinh\frac{a}{\lambda} + a\epsilon_s \cosh\frac{a}{\lambda}}.$$

The Born test model (57) is particularly suitable for verification tests. From the solution expression (58) it can be seen that the solution $u$ is proportional to $\alpha$. Thus, the relative error $E_r = \|u - u_h\|/\|u\|$ of a finite element solution $u_h$ becomes independent of $\alpha$, where $\|\cdot\|$ is the norm of the function space $L^2(\Omega)$. This property is valuable for validation tests. In fact, because of the extra charge term $f$, the parameter $\alpha$ loses its physical sense. Instead, it mainly plays a role of controlling the

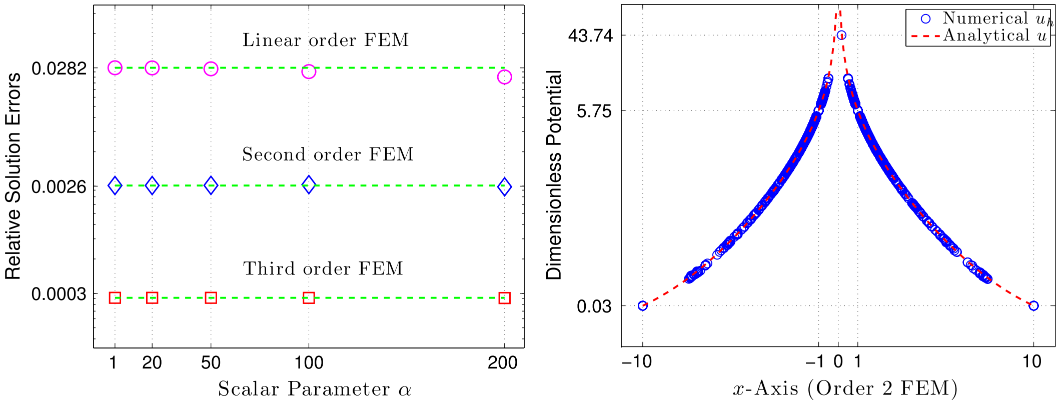


**Fig. 1.** Validation of our NMPBE solver and program package by the Born test model (57). **Left Plot:** Relative errors of the finite element solutions in terms of $\alpha$ and the order of the finite element method (FEM). **Right Plot:** A comparison of the finite element solutions (in blue circle) with the analytical solution (in red dash line).

**Table 2**
Errors and convergence order of the NMPBE solver for the Born test model (57) on the three nested tetrahedral meshes. Here, $N_h$ denotes the number of mesh nodes, and the order $\Upsilon_k$ of convergence rate is estimated via formula (60).

| $N_h$ | $h_{\max}$ | $h_{\min}$ | $\|u-u_h\|/\|u\|$ | $\|u-u_h\|$ | $L_2$ order $\Upsilon_k$ |
|---|---|---|---|---|---|
| 2,260 | 1.732 | 0.073 | $6.65 \times 10^{-3}$ | $1.03 \times 10^{-3}$ | – |
| 17,717 | 1.049 | 0.020 | $1.78 \times 10^{-3}$ | $2.71 \times 10^{-4}$ | 1.9263 |
| 139,993 | 0.579 | 0.010 | $4.41 \times 10^{-4}$ | $6.96 \times 10^{-5}$ | 1.9611 |

solution range of (57). Hence, with this property, we can do many validation tests using different values of $\alpha$. In addition, the NMPBE package can be easily adapted to solve (57). The only change to the package is to add the extra charge term $\int_{D_s} f v_1 d\mathbf{r}$ to the right hand side of the modified Newton equation (50). Therefore, the Born test model (57) is a good one for verifying a NMPBE solver.

We made numerical tests using $\lambda = 15$, $a = 1.0$, $\Omega = \{\mathbf{r} \mid |\mathbf{r}| < 10\}$, $\alpha = 1, 20, 50, 100,$ and 200, and a zero initial guess based on a tetrahedral mesh with 2955 vertices from [27]. The convolution $\hat{g}$ of $g$ was set by $\hat{g} = \hat{u}$ with $\hat{u}$ being given in (59). The first, second, and third order finite element methods were applied to the numerical solution of (57), resulting three linear systems with 5910, 47044, and 158118 unknowns, respectively. To achieve a higher numerical accuracy, each linear algebraic system from the first and second order finite element methods was solved by the LU factorization method while the ones from the cubic finite element method were still solved by the GMRES with ILU preconditioning due to memory limitation.

Fig. 1 reports the numerical results of our validation tests. From the left plot we can see that the relative errors $E_r$ had almost the same value for different $\alpha$, verifying that $E_r$ is independent of $\alpha$. Moreover, the relative errors $E_r$ were reduced almost one order as the order of the finite element method was increased by one. For example, in the case of $\alpha = 200$, it was found that $E_r = 2.34702 \times 10^{-2}$, $2.56656 \times 10^{-3}$, and $2.72852 \times 10^{-4}$ for the first, second, and third order finite element methods, respectively. These results well matched the finite element theory [48].

The right plot of Fig. 1 compares the quadratic finite element solution $u_h$ with the analytical solution $u$. Here, $\alpha = 1$, and the solution values were plotted in terms of the $x$-coordinate of the mesh vertices (except the ones with zero $x$ coordinates) according to the vertex ordering in which the magnitudes of vertices decrease on $[-10, 0]$ and increase on $[0, 10]$. From this plot it can be seen that $u_h$ almost matched $u$. These test results well validated our NMPBE solver and program package.

Furthermore, we constructed three nested tetrahedral meshes for $a = 1$ and a cubic domain $\Omega = [-3, 3]^3$ such that their mesh sizes $h_k \approx h_{k-1}/2$ for $k = 2, 3$. Here $h_1$ is a mesh size of the initial mesh with 2262 mesh nodes. We then estimated the convergence order of a linear finite element method approximately by

$$\Upsilon_k = \ln(\|u - u_{h_k}\|_2 / \|u - u_{h_{k-1}}\|_2) / \ln(h_k/h_{k-1}), \quad k = 2, 3. \tag{60}$$

In calculation of $\Upsilon_k$, we simply set as $h_k/h_{k-1} = 1/2$. Since the three tetrahedral meshes of $\Omega$ were unstructured, we calculated $h_{\max}$ and $h_{\min}$ as the maximum and minimum of all the diameters of tetrahedra, respectively. The numerical results were reported in Table 2.

From Table 2 we can see that $\Upsilon_k$ approached 2 as the mesh size decreased. This shows that our NMPBE solver can well retain a quadratical convergence rate of the linear finite element method in terms of mesh size [48].

### 5.2. Performance tests for proteins

To demonstrate the performance of our NMPBE finite element program package in terms of computer CPU time, we made numerical experiments on twelve protein molecules downloaded from the protein data bank. The PQR files of these proteins

**Table 3**
Performance of the NMPBE program package for the twelve proteins.

| PDB ID ($n_p, N_h$) | CPU time for five major parts of NMPBE (in seconds) | | | | | Total time (min.) |
|---|---|---|---|---|---|---|
| | Solve $\Psi$ by (33) | Solve $\tilde{\Phi}^{(0)}$ by (55) | Solve $\tilde{\Phi}$ by (40) | Find $G$, $\hat{G}$, $\nabla G$, $\nabla\hat{G}$ | Generate mesh | |
| 2LZX (488, 39644) | 2.17 | 10.98 | 48.59 | 1.93 | 8.80 | 1.21 |
| 1AJJ (513, 55566) | 3.60 | 22.86 | 99.40 | 2.17 | 12.51 | 2.35 |
| 1FXD (811, 58754) | 4.04 | 25.59 | 87.51 | 3.63 | 13.32 | 2.24 |
| 1HPT (852, 22492) | 1.09 | 9.38 | 24.54 | 1.47 | 6.48 | 0.72 |
| 4PTI (892, 67669) | 4.65 | 20.79 | 104.69 | 5.96 | 14.66 | 2.52 |
| 1SVR (1433, 30634) | 1.64 | 16.76 | 38.25 | 3.34 | 8.57 | 1.14 |
| 1A63 (2065, 38337) | 2.23 | 16.62 | 48.08 | 6.02 | 12.00 | 1.42 |
| 1CID (2783, 32852) | 1.86 | 20.22 | 45.57 | 6.95 | 10.45 | 1.43 |
| 1A7M (2803, 34502) | 1.95 | 14.02 | 47.14 | 7.35 | 10.94 | 1.36 |
| 2AQ5 (6024, 62026) | 4.31 | 43.28 | 112.60 | 28.40 | 20.47 | 3.49 |
| 1F6W (8243, 23803) | 1.16 | 13.30 | 35.02 | 14.91 | 16.50 | 1.35 |
| 1C4K (11439, 32436) | 1.85 | 22.90 | 60.19 | 28.22 | 23.84 | 2.30 |

**Table 4**
A comparison of the performance of the new modified Newton iterative method using an initial iterate $\tilde{\Phi}^{(0)}$ constructed from PBE (see (55) and (56)) with that from a linear nonlocal problem as given in (53). Here, PBE was solved by our PBE program package [27], and the residual norm $\|F(\tilde{\Phi}^{(k)})\|$ has been defined in (47).

| PDB ID | Start $\|F(\tilde{\Phi}^{(0)})\|$ | | Final $\|F(\tilde{\Phi}^{(k)})\|$ | | Iteration | | Time (min.) | |
|---|---|---|---|---|---|---|---|---|
| | By PBE | By (53) | By PBE | By (53) | By PBE | By (53) | By PBE | By (53) |
| 2LZX | $3.32 \times 10^{3}$ | $4.10 \times 10^{15}$ | $6.85 \times 10^{-7}$ | $6.24 \times 10^{-7}$ | 28 | 54 | 0.99 | 1.86 |
| 1AJJ | $2.07 \times 10^{8}$ | $3.58 \times 10^{17}$ | $4.46 \times 10^{-7}$ | $5.95 \times 10^{-7}$ | 33 | 58 | 2.03 | 3.15 |
| 1FXD | $6.47 \times 10^{4}$ | $8.44 \times 10^{15}$ | $7.54 \times 10^{-7}$ | $5.96 \times 10^{-7}$ | 28 | 55 | 1.89 | 3.01 |
| 1HPT | $2.03 \times 10^{4}$ | $2.56 \times 10^{18}$ | $9.18 \times 10^{-7}$ | $7.69 \times 10^{-7}$ | 29 | 59 | 0.56 | 0.94 |
| 4PTI | $5.87 \times 10^{5}$ | $1.55 \times 10^{17}$ | $4.44 \times 10^{-7}$ | $5.24 \times 10^{-7}$ | 29 | 57 | 2.09 | 3.82 |
| 1SVR | $4.60 \times 10^{5}$ | $1.14 \times 10^{25}$ | $7.57 \times 10^{-7}$ | $9.93 \times 10^{-7}$ | 29 | 75 | 0.92 | 1.95 |
| 1A63 | $2.54 \times 10^{5}$ | $6.90 \times 10^{17}$ | $9.77 \times 10^{-7}$ | $8.22 \times 10^{-7}$ | 27 | 54 | 1.08 | 1.61 |
| 1CID | $1.31 \times 10^{5}$ | $2.96 \times 10^{28}$ | $7.84 \times 10^{-7}$ | $1.20 \times 10^{-6}$ | 33 | 83 | 1.10 | 2.07 |
| 1A7M | $2.07 \times 10^{6}$ | $4.43 \times 10^{23}$ | $6.33 \times 10^{-7}$ | $7.56 \times 10^{-7}$ | 31 | 74 | 1.02 | 2.04 |
| 2AQ5 | $2.87 \times 10^{7}$ | $1.50 \times 10^{21}$ | $7.40 \times 10^{-7}$ | $7.41 \times 10^{-7}$ | 35 | 62 | 2.60 | 3.77 |
| 1F6W | $1.60 \times 10^{8}$ | $1.41 \times 10^{21}$ | $2.48 \times 10^{-6}$ | $3.39 \times 10^{-6}$ | 39 | 69 | 0.81 | 1.22 |
| 1C4K | $3.95 \times 10^{7}$ | $1.09 \times 10^{22}$ | $7.60 \times 10^{-6}$ | $7.88 \times 10^{-6}$ | 45 | 83 | 1.38 | 1.84 |

were determined by using PDB2PQR under the CHARMM forcefield. Each finite element tetrahedral mesh was generated from our `GAMer`-II with the default parameter values except two parameters: DomainRatio = 2 and DIM_SCALE = 1.5 (see [33, Appendix] for their definitions). Such a mesh setting resulted in a medium-resolution mesh, which was suitable not only for our performance tests but also for the applications discussed in Sections 5.3 and 5.4. The PDB ID, the number $n_p$ of atoms, and the number $N_h$ of mesh nodes were listed in Table 3. We used $\lambda = 15$ for all the tests.

Table 3 reports the CPU time distributions for the five main parts of our NMPBE program package. The linear equation of (33) was solved efficiently in about 1 to 5 seconds by the GMRES using the ILU preconditioning, showing the high efficiency of the linear iterative solver. The nonlinear finite element equation (40) was solved by our modified Newton iterative method using the initial iterate defined from the PBE solution as given in (55). Here, the numerical solutions of (33) and each modified Newton equation of (50) had the relative and absolute residue errors being less than $10^{-8}$. The total CPU time included the time for generating the mesh in an range from 0.72 to 3.49 minutes for these proteins with up to 11439 atoms, demonstrating the high performance of our NMPBE software package.

Table 4 reports the performance of our modified Newton iterative method for solving the nonlinear nonlocal variational problem (40) when the initial iterate $\tilde{\Phi}^{(0)}$ was set by PBE (see (55) and (56)) or by a linear nonlocal problem as given in (53). Here the CPU time includes the one spent on the calculation of $\tilde{\Phi}^{(0)}$. These numerical results demonstrate that our new modified Newton iterative method is efficient, and can converge fast. They also show that the initial iterate constructed by PBE can significantly improve the performance of the modified Newton iterative method using an initial guess as given in (53). For example, in the case of 1SVR (a protein with 1433 atoms) on a mesh with 30634 mesh nodes, which led to a nonlinear algebraic system with about 61268 (i.e., $30634 \times 2$) unknowns, by the PBE constructed initial iterate, the CPU time of computing $\tilde{\Phi}$ was reduced from about 1.95 minutes to 0.92 minute while the number of iterations was reduced from 75 to 29. Hence, the initial iterates constructed by PBE can be a default choice for our NMPBE software package.

### 5.3. *Electrostatic solvation free energy calculation*

The electrostatic solvation free energy $\Delta E_{ele}$ is one dominant part of the solvation free energy, which describes the free energy required to transfer a biomolecule from a solvent state to the vacuum state. In the implicit solvent continuum dielectric approach, it is commonly estimated in kilojoules per mole (kJ/mol) by

$$\Delta E_{ele} = \frac{N_A}{1000}\frac{1}{2}\int\limits_{\Omega} \rho_f(\mathbf{r})(\Phi_{sol} - \Phi_{vac})(\mathbf{r})d\mathbf{r}, \tag{61}$$

where $N_A$ is the Avogadro number, $\Phi_{sol}$ and $\Phi_{vac}$ denote the electrostatic potential functions in the solvent and vacuum states, respectively, and $\rho_f$ denotes the fixed charge density function. For $\rho_f = e_c \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j}$, we can use (20) and the solution decomposition (22) to reformulate (61) as a new formula for computing $\Delta E_{ele}$:

$$\Delta E_{ele} = \frac{N_A k_B T}{1000}\frac{1}{2}\sum_{j=1}^{n_p} z_j(\Psi + \tilde{\Phi})(\mathbf{r}_j), \tag{62}$$

where $\Psi$ and $\tilde{\Phi}$ are defined by (30) and (31), respectively. By this new formula, the electrostatic solvation free energy $\Delta E_{ele}$ can be calculated without any singularity difficulty.

With (62), we calculated the values of $\Delta E_{ele}$ for the twelve protein molecules from the previous subsection for $\lambda = 0.05, 0.1, 0.2, 0.5, 1, 2, 3, \ldots, 10, 15, 20, 25, \ldots, 70$ to check the behaviors of $\Delta E_{ele}$ as $\lambda \to 0$ or $\lambda \to \infty$. As a comparison, we also did the calculation via our PBE program package [27]. The results are reported in Fig. 2.

Fig. 2 shows that the electrostatic solvation energy difference $\Delta E_{ele}$ can be a bounded increasing function of the nonlocal parameter $\lambda$ when $\lambda > 2$. Its upper bound was indicated by a horizontal asymptote. Interestingly, $\Delta E_{ele}$ becomes a decreasing function of $\lambda$ for $0 < \lambda < 2$, and can approach the case of PBE as $\lambda \to 0$. This important feature may make NMPBE to become a better predictor than the classic PBE model in the calculation of electrostatic solvation free energy through a proper selection of $\lambda$.

From Fig. 2 we also see that $\Delta E_{ele}$ becomes less sensitive to the selection of $\lambda$ when $\lambda$ is large enough (e.g., $\lambda > 35$). Our test results suggest that a good selection range for $\lambda$ is from 10 to 30, which is consistent with the studies given in [5, Table 1] and [45,16], in which a typical $\lambda$-range of 10 to 20 Å was suggested.

Fig. 3 shows the numerical behavior of our NMPBE solver in the calculation of $\Delta E_{ele}$ with respect to mesh size. The tests were done for three different protein molecules on five different meshes. The number of mesh vertices was marked out on the horizontal line axis for each mesh. As the numbers of vertices was increased from about 34,000 to 760,000, the value of $\Delta E_{ele}$ varied smoothly, reflecting the convergent feature of our algorithm. This indicates that our solver can work well on a large scale mesh so that it can yield numerical solutions in high accuracy.

### *5.4. Binding free energy calculation*

Binding free energy calculation is another important application of a dielectric continuum model. PBE has been an important tool to do so, and has been applied to many bioengineering applications (such as computer-aided drug design) [51,23,52]. As an initial application of our NMPBE, we made tests on a DNA-drug complex (PDB ID 1D86) using the PQR files from [53]. The salt dependence of the binding constants for 1D86 was studied chemically in [37, Table 3].

Let $E(X, I_s)$ denote an electrostatic free energy of molecule $X$ in a solvent with the ionic strength $I_s \in (0, 1)$. To study the salt dependence of a complex $C$ consisting of molecules $A$ and $B$, a binding free energy, $E_b$, is defined by

$$E_b(I_s) = E(C, I_s) - E(A, I_s) - E(B, I_s). \tag{63}$$

In binding free energy calculation, $E(X, I_s)$ is set as the free energy from the water state to the salt solution state. Since the sum of $G$ and $\Psi$ gives the potential in the water state, with (20) and the solution decomposition (22), we can obtain a new formula for computing $E(X, I_s)$ in the expression

$$E(X, I_s) = \frac{\mathcal{C}}{2}\sum_{j=1}^{n_p} z_j \tilde{\Phi}_{X,I_s}(\mathbf{r}_j), \tag{64}$$

where $\mathcal{C} = N_A k_B T/4184$, and $\tilde{\Phi}_{X,I_s}$ denotes a solution of the nonlinear nonlocal problem (40) for molecule $X$ in a solvent with ionic strength $I_s$. By the constant 4.184, the unit of $E$ is transformed to kilocalorie per mole (kcal/mol) from kJ/mol.

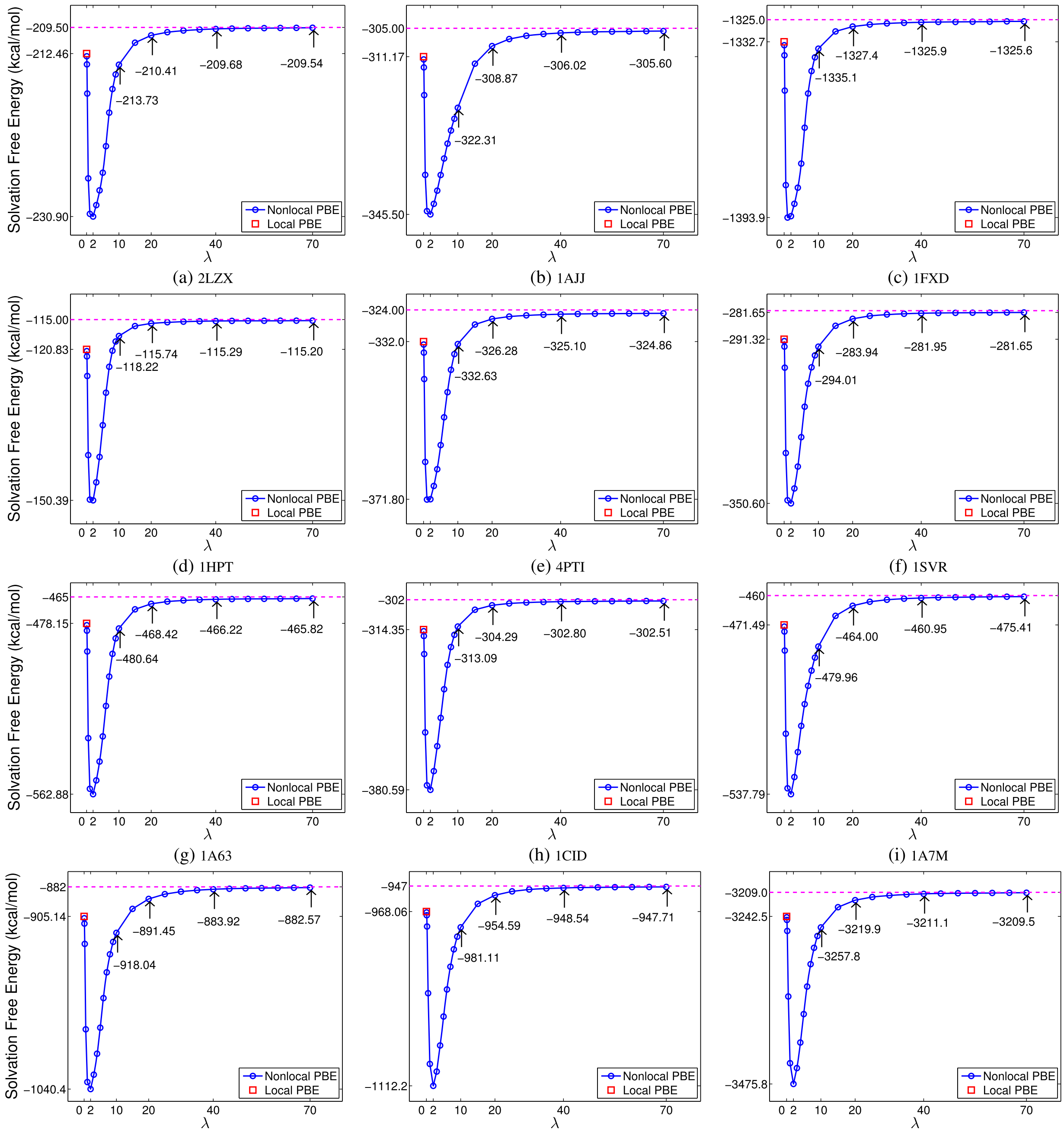
By variable change $\xi = \log_{10} I_s$ (or $\xi = \ln I_s$), the binding free energy $E_b$ can be transformed as a linear function of $\xi$ with the slope $m$ on an interval $[a_1, a_2]$ of $\xi$:

$$E_b = m\xi + b, \quad a_1 \le \xi \le a_2,$$

where $m$ can be estimated experimentally in a chemical laboratory [54].

To estimate the slope $m$ computationally, we select a steplength, $\tau > 0$, to construct a set of numbers, $\xi_j = a_1 + j\tau$ for $j = 0, 1, 2, \ldots, l$, with $l$ being the largest number satisfying $a_1 + l\tau \le a_2$. We next calculate a set of binding energies, $E_b(I_{s,j})$, with $I_{s,j} = 10^{\xi_j}$ for $j = 0, 1, 2, \ldots, l$. The slope $m$ is then estimated as the slope of a best fitted line determined by a linear regression program using the set of computed binding energies. We can also simply estimate the slope $m$ by the arithmetic average:

$$\tilde{\rho} = \frac{1}{l}\sum_{j=1}^{l} \frac{E_b(I_{s,j}) - E_b(I_{s,j-1})}{\tau}. \tag{65}$$



(a) 2LZX (b) 1AJJ (c) 1FXD

(d) 1HPT (e) 4PTI (f) 1SVR

(g) 1A63 (h) 1CID (i) 1A7M

(j) 2AQ5 (k) 1F6W (l) 1C4K

**Fig. 2.** The electrostatic solvation energy $\Delta E_{ele}$ (defined in (62)) as a function of the nonlocal parameter $\lambda$ (in Å) for the twelve proteins reported in Table 4. Here the PBE case was calculated by our PBE software package [27].

For the DNA-drug complex, a scaled slope, $m_s$, defined by using $\xi = \ln I_s$, was determined chemically as $m_s \approx -1.51$ in [37, Table 3]. Using the formula $m = -m_s N_A k_B T/(4184 \log_{10} e)$, we got $m \approx 2.06$ kcal/mol.

In our numerical tests, we set $a_1 = -1.5$, $a_2 = -0.1$ (i.e., $0.0316 \le I_s \le 0.7943$), $\tau = 0.1$, and $l = 14$. In the tests for the DNA-drug complex, $\lambda$ was set as 10, 11, ..., 18, and the three tetrahedral meshes had 57663, 62643, and 54797 vertices for the complex, DNA, and drug, respectively. We then estimated the slope $m$, and repeated the tests using our PBE program package [27]. The main results were reported in Figs. 4 and 5.

Fig. 4 shows that the arithmetic average $\tilde{\rho}$ of the slope can be an increasing function of $\lambda$. The best computed value 2.102 of $\tilde{\rho}$, which was reached at $\lambda = 13$, almost matches the experimental value of 2.06.

Fig. 5 compares the binding free energies computed by NMPBE and our PBE software package in best fitted lines. The slopes of the best fitted lines by NMPBE were closer to the experimental slope values than the ones by PBE, indicating that

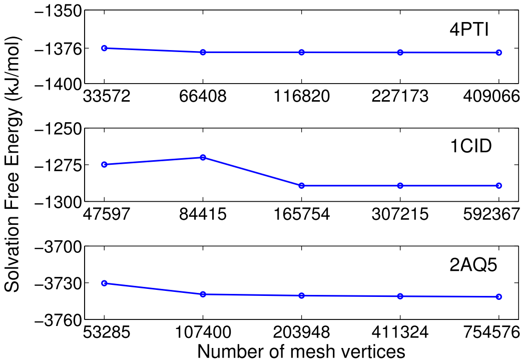


**Fig. 3.** Numerical stability of NMPBE solver in calculation of solvation free energy with respect to mesh size.

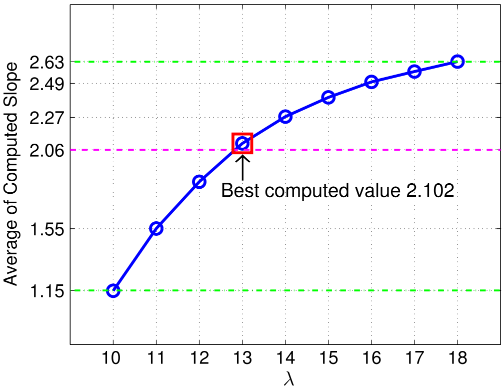


**Fig. 4.** The slope average $\tilde{\rho}$ computed by NMPBE as a function of λ (in Å) for the DNA-drug complex. Here $\tilde{\rho}$ is calculated by (65).

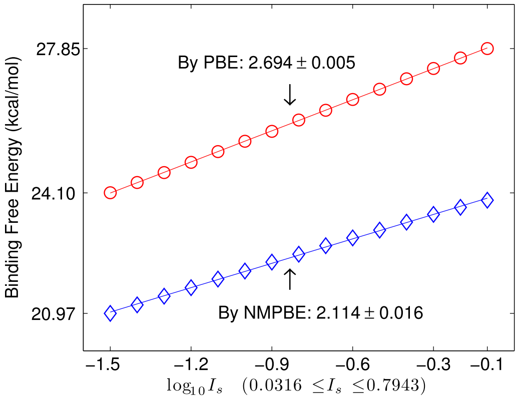
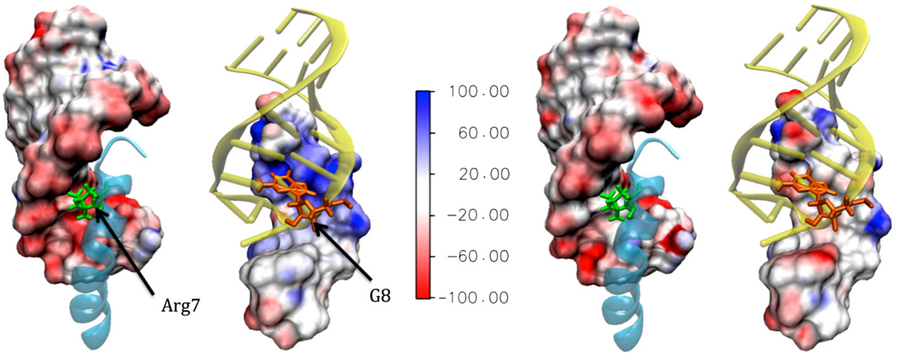


**Fig. 5.** A comparison of the computed binding free energies by NMPBE with those by our PBE package [27] in best fitted lines for the DNA-drug complex.

NMPBE has improved the quality of PBE in the prediction of binding free energy. Since the deviations of all the best fitted lines were small, the binding energy data generated by NMPBE or PBE matched the experimental observation — The binding energy $E_b$ is a linear function of $\xi$ with $\xi = \log_{10} I_s$ for $I_s \in [0.0316, 0.7943]$.

Finally, we studied the electrostatics on a binding site for a peptide-RNA complex. We got the PQR file of this complex and the PQR files of its peptide and RNA parts from the APBS website (*http://www.poissonboltzmann.org*). We also knew from [23] that the seventh residue Arg7 (Arginine) of the peptide bound with the eighth base G8 (guanine) of RNA. Hence, this peptide-RNA complex is a good example for us to compare the electrostatics predicted by NMPBE before the binding with that after the binding. In our NMPBE tests on this complex, we used $\lambda = 19$, $I_s = 0.1$, and the three tetrahedral meshes with 57072, 41960, and 47970 vertices for the peptide-RNA complex, the peptide, and the RNA, respectively. To clearly display the electrostatics at the binding site, the peptide and RNA parts were plotted in ribbon on the left and right plots in Fig. 6(A) and (B), respectively, along with the binding site being marked out by a molecular structure of Arg7 or G8 on each plot.



(A) Electrostatics before binding (B) Electrostatics after binding

**Fig. 6.** A comparison of electrostatics predicted by NMPBE before the binding (Figure (A)) with that after the binding (Figure (B)) on the surfaces of the RNA and peptide parts of a peptide-RNA complex given in [23]. Here, the seventh residue Arg7 of the peptide bound with the eighth base G8 of the RNA, Arg7 and G8 are plotted in molecular structure to mark out this binding site, and the blue, red, and white patches represent positive, negative, and neutral electrostatics, respectively. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Figs. 6(A) and 6(B) display the cases before and after the binding, respectively. From Fig. 6(A) we can see that NMPBE produced negative electrostatics (in red patches) on the binding site of RNA, and positive electrostatics (in blue patches) on the binding site of peptide. After the binding, as shown in Fig. 6(B), the electrostatics produced by NMPBE became almost neutral (in white patches) at the binding site. These test results well matched the experimental observation — Arg7 bound with G8 due to their hydrogen bonds [23, Figure 1]. They further demonstrate the value of NMPBE in the study of binding properties.

## 6. Conclusions

In this paper, we have presented an efficient finite element numerical algorithm and a related program package for solving a nonlinear nonlocal dielectric model, called the nonlocal modified PBE (NMPBE). In order to overcome the difficulties of solution singularity caused by atomic charges, we constructed the algorithm according to the solution decomposition in which the NMPBE solution is split as a sum of three functions $G$, $\Psi$, and $\tilde{\Phi}$. We then proposed a novel modified Newton iterative algorithm for solving the nonlinear and nonlocal boundary value problem that defines $\tilde{\Phi}$, together with two good initial iterates constructed from a linear NMPBE model and the PBE solution. We also proposed a nonlocal Debye–Hückel equation and obtained its analytical solution and its convolution function, which can be used to construct a non-trivial boundary value function for NMPBE. In addition, we obtained an analytical expression of the convolution of $G$, and used it to improve the numerical accuracy for computing $\Psi$. Moreover, numerical results on a nonlinear nonlocal Born ball test model and various proteins were reported. They not only validated our NMPBE numerical algorithms and program package but also demonstrated the efficiency of our new modified Newton iterative algorithm and the high performance of our NMPBE software package. Finally, the application test results on proteins, a peptide-RNA complex, and a DNA-drug complex showed the potential of NMPBE as a better predictor of electrostatic solvation and binding free energies than PBE.

We noted that several other variants of PBE were developed recently to remedy the weaknesses of PBE caused by ignoring ionic size effects and polarization correlations among water molecules. They include the size modified PBE models [55, 28], the dipolar Poisson–Boltzmann–Langevin models [56–58], a generalized PBE model by differential geometry [59], a generalized PBE model by ionic concentration functions [60,61], and the Poisson–Fermi model [62–64]. NMPBE stands for our initial efforts to incorporate polarization correlations into the improvement of PBE. In the future, we plan to further improve it to reflect ionic size effects or other important physical/chemical effects in our calculation of electrostatic potential energies.

## Acknowledgement

This work was partially supported by the National Science Foundation, USA, through grant DMS-1226259.

## References

[1] The Chemical Physics of Solvation. Part A: Theory of Solvation, R.R. Dogonadze, E. Kálmán, A.A. Kornyshev, J. Ulstrup (Eds.), Studies in Physical and Theoretical Chemistry, vol. 38, Elsevier Science Ltf, Amsterdam, 1985.
[2] R.R. Dogonadze, A.A. Kornyshev, Polar solvent structure in the theory of ionic solvation, J. Chem. Soc. Faraday Trans. 2: Mol. Chem. Phys. 70 (1974) 1121–1132.
[3] A.A. Kornyshev, A.I. Rubinshtein, M.A. Vorotyntsev, Model nonlocal electrostatics. I, J. Phys. C, Solid State Phys. 11 (1978) 3307.
[4] A. Kornyshev, G. Sutmann, Nonlocal dielectric saturation in liquid water, Phys. Rev. Lett. 79 (1997) 3435–3438.
[5] M. Basilevsky, D. Parsons, An advanced continuum medium model for treating solvation effects: nonlocal electrostatics with a cavity, J. Chem. Phys. 105 (9) (1996) 3734–3746.

[6] M. Basilevsky, D. Parsons, Nonlocal continuum solvation model with exponential susceptibility kernels, J. Chem. Phys. 108 (1998) 9107–9113.
[7] M. Basilevsky, D. Parsons, Nonlocal continuum solvation model with oscillating susceptibility kernels: a nonrigid cavity model, J. Chem. Phys. 108 (1998) 9114–9123.
[8] P. Bopp, A. Kornyshev, G. Sutmann, Static nonlocal dielectric function of liquid water, Phys. Rev. Lett. 76 (1996) 1280–1283.
[9] S. Buyukdagli, T. Ala-Nissila, Microscopic formulation of nonlocal electrostatics in polar liquids embedding polarizable ions, Phys. Rev. E 87 (2013) 063201.
[10] A. Kornyshev, G. Sutmann, Nonlocal dielectric function of water: how strong are the effects of intramolecular charge form factors?, J. Mol. Liq. 82 (1999) 151–160.
[11] J. Rottler, B. Krayenhoff, Numerical studies of nonlocal electrostatic effects on the sub-nanoscale, J. Phys. Condens. Matter 21 (25) (2009) 255901.
[12] A. Rubinstein, S. Sherman, Influence of the solvent structure on the electrostatic interactions in proteins, Biophys. J. 87 (3) (2004) 1544–1557.
[13] B. Sahin, B. Ralf, Nonlocal and nonlinear electrostatics of a dipolar coulomb fluid, J. Phys. Condens. Matter 26 (28) (2014) 285101.
[14] L. Scott, M. Boland, K. Rogale, A. Fernández, Continuum equations for dielectric response to macro-molecular assemblies at the nano scale, J. Phys. A, Math. Gen. 37 (2004) 9791–9803.
[15] A. Hildebrandt, R. Blossey, S. Rjasanow, O. Kohlbacher, H.-P. Lenhof, Novel formulation of nonlocal electrostatics, Phys. Rev. Lett. 93 (10) (2004) 108104.
[16] A. Hildebrandt, R. Blossey, S. Rjasanow, O. Kohlbacher, H. Lenhof, Electrostatic potentials of proteins in water: a structured continuum approach, Bioinformatics 23 (2) (2007) e99–e103.
[17] S. Weggler, V. Rutka, A. Hildebrandt, A new numerical method for nonlocal electrostatics in biomolecular simulations, J. Comput. Phys. 229 (11) (2010) 4059–4074.
[18] D. Xie, Y. Jiang, P. Brune, L. Scott, A fast solver for a nonlocal dielectric continuum model, SIAM J. Sci. Comput. 34 (2) (2012) B107–B126.
[19] D. Xie, H. Volkmer, A modified nonlocal continuum electrostatic model for protein in water and its analytical solutions for ionic Born models, Commun. Comput. Phys. 13 (1) (2013) 174–194.
[20] D. Xie, Y. Jiang, L. Scott, Efficient algorithms for a nonlocal dielectric model for protein in ionic solvent, SIAM J. Sci. Comput. 35 (6) (2013) B1267–B1284.
[21] N.A. Baker, Poisson–Boltzmann methods for biomolecular electrostatics, Methods Enzymol. 383 (2003) 94–118.
[22] F. Fogolari, A. Brigo, H. Molinari, The Poisson–Boltzmann equation for biomolecular electrostatics: a tool for structural biology, J. Mol. Recognit. 15 (6) (2002) 377–392.
[23] C. García-García, D. Draper, Electrostatic interactions in a peptide-RNA complex, J. Mol. Biol. 331 (1) (2003) 75–88.
[24] F. Fogolari, G. Esposito, P. Viglino, H. Molinari, Molecular mechanics and dynamics of biomolecules using a solvent continuum model, J. Comput. Chem. 22 (15) (2001) 1830–1842.
[25] P. Kollman, et al., Calculating structures and free energies of complex molecules: combining molecular mechanics and continuum models, Acc. Chem. Res. 33 (12) (2000) 889–897.
[26] B. Roux, T. Simonson, Implicit solvent models, Biophys. Chem. 78 (1999) 1–20.
[27] D. Xie, New solution decomposition and minimization schemes for Poisson–Boltzmann equation in calculation of biomolecular electrostatics, J. Comput. Phys. 275 (2014) 294–309.
[28] B. Li, Continuum electrostatics for ionic solutions with non-uniform ionic sizes, Nonlinearity 22 (4) (2009) 811–833.
[29] D. Xie, J. Li, A new analysis of electrostatic free energy minimization and Poisson–Boltzmann equation for protein in ionic solvent, Nonlinear Anal., Real World Appl. 21 (2015) 185–196.
[30] N. Baker, Improving implicit solvent simulations: a Poisson-centric view, Curr. Opin. Struct. Biol. 15 (2005) 137–143.
[31] B. Lu, Y. Zhou, M. Holst, J. McCammon, Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications, Commun. Comput. Phys. 3 (5) (2008) 973–1009.
[32] C.L. Vizcarra, S.L. Mayo, Electrostatics in computational protein design, Curr. Opin. Chem. Biol. 9 (2005) 622–626.
[33] Y. Jiang, Y. Xie, J. Ying, D. Xie, Z. Yu, SDPBS web server for calculation of electrostatics of ionic solvated biomolecules, Mol. Based Math. Biol. 3 (2015) 179–196.
[34] Z. Yu, M. Holst, Y. Cheng, J. McCammon, Feature-preserving adaptive mesh generation for molecular shape modeling and simulation, J. Mol. Graph. Model. 26 (8) (2008) 1370–1380.
[35] J. Nocedal, S. Wright, Numerical Optimization, 2nd edition, Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.
[36] A. Logg, G.N. Wells, J. Hake, DOLFIN: a C++/Python finite element library, in: Automated Solution of Differential Equations by the Finite Element Method, in: Lect. Notes Comput. Sci. Eng., vol. 84, Springer, Heidelberg, 2012, pp. 173–225, Ch. 10.
[37] K.J. Breslauer, D.P. Remeta, W.-Y. Chou, R. Ferrante, J. Curry, D. Zaunczkowski, J.G. Snyder, L.A. Marky, Enthalpy-entropy compensations in drug-DNA binding studies, Proc. Natl. Acad. Sci. 84 (24) (1987) 8922–8926.
[38] P. Debye, Polar Molecules, Dover, New York, 1945.
[39] D. Griffiths, Introduction to Electrodynamics, 3rd edition, Prentice Hall, New Jersey, 1999.
[40] U. Kaatze, R. Behrends, R. Pottel, Hydrogen network fluctuations and dielectric spectrometry of liquids, J. Non-Cryst. Solids 305 (1) (2002) 19–28.
[41] M. Basilevsky, G. Chuev, Nonlocal solvation theories, in: B. Mennucci, R. Cammi (Eds.), Continuum Solvation Models in Chemical Physics: From Theory to Applications, Wiley, 2008, pp. 94–109.
[42] H. Yada, M. Nagai, K. Tanaka, The intermolecular stretching vibration mode in water isotopes investigated with broadband terahertz time-domain spectroscopy, Chem. Phys. Lett. 473 (4–6) (2009) 279–283.
[43] O. Dolgov, D. Kirzhnits, E. Maksimov, On an admissible sign of the static dielectric function of matter, Rev. Mod. Phys. 53 (1) (1981) 81–94.
[44] A.C. Maggs, R. Everaers, Simulating nanoscale dielectric response, Phys. Rev. Lett. 96 (23) (2006) 230603.
[45] A. Hildebrandt, Biomolecules in a structured solvent: a novel formulation of nonlocal electrostatics and its numerical solution, PhD thesis, Saarlandes University, Saarbrücken, Germany, February 2005.
[46] A. Kornyshev, A. Nitzan, Effect of overscreening on the localization of hydrated electrons, Z. Phys. Chem. 215 (6) (2001) 701–715.
[47] L.R. Scott, D. Xie, Analysis of a nonlocal Poisson–Boltzmann equation, Technical Report TR-2016-01, University of Chicago, 2016.
[48] S. Brenner, L. Scott, The Mathematical Theory of Finite Element Methods, 3rd edition, Springer-Verlag, New York, 2008.
[49] W. Rocchia, Poisson–Boltzmann equation boundary conditions for biological applications, Math. Comput. Model. 41 (10) (2005) 1109–1118.
[50] T. Dolinsky, J. Nielsen, J. McCammon, N. Baker, PDB2PQR: an automated pipeline for the setup of Poisson–Boltzmann electrostatics calculations, Nucleic Acids Res. 32 (suppl 2) (2004) W665.
[51] C. Bertonati, B. Honig, E. Alexov, Poisson–Boltzmann calculations of nonspecific salt effects on protein-protein binding free energies, Biophys. J. 92 (6) (2007) 1891–1899.
[52] D. Sitkoff, K.A. Sharp, B. Honig, Accurate calculation of hydration free energies using macroscopic solvent models, J. Phys. Chem. 98 (7) (1994) 1978–1988.
[53] M.O. Fenley, R.C. Harris, B. Jayaram, A.H. Boschitsch, Revisiting the association of cationic groove-binding drugs to DNA using a Poisson–Boltzmann approach, Biophys. J. 99 (3) (2010) 879–886.
[54] G.S. Manning, The molecular theory of polyelectrolyte solutions with applications to the electrostatic properties of polynucleotides, Q. Rev. Biophys. 11 (02) (1978) 179–246.

[55] I. Borukhov, D. Andelman, H. Orland, Steric effects in electrolytes: a modified Poisson–Boltzmann equation, Phys. Rev. Lett. 79 (3) (1997) 435.

[56] P. Koehl, M. Delarue, AQUASOL: an efficient solver for the dipolar Poisson–Boltzmann–Langevin equation, J. Chem. Phys. 132 (6) (2010) 064101.

[57] P. Koehl, F. Poitevin, H. Orland, M. Delarue, Modified Poisson–Boltzmann equations for characterizing biomolecular solvation, J. Theor. Comput. Chem. 13 (03) (2014) 1440001.

[58] P. Koehl, H. Orland, M. Delarue, Beyond the Poisson–Boltzmann model: modeling biomolecule-water and water–water interactions, Phys. Rev. Lett. 102 (8) (2009) 087801.

[59] G.-W. Wei, Multiscale, multiphysics and multidomain models I: basic theory, J. Theor. Comput. Chem. 12 (08) (2013).

[60] D. Chen, Modeling and computation of heterogeneous implicit solvent and its applications for biomolecules, Mol. Based Math. Biol. 2 (1) (2014).

[61] B. Li, J. Wen, S. Zhou, Mean-field theory and computation of electrostatics with ionic concentration dependent dielectrics, Commun. Comput. Phys. 14 (1) (2016) 249–271.

[62] M.Z. Bazant, B.D. Storey, A.A. Kornyshev, Double layer in ionic liquids: overscreening versus crowding, Phys. Rev. Lett. 106 (2011) 046102.

[63] C.D. Santangelo, Computing counterion densities at intermediate coupling, Phys. Rev. E 73 (4) (2006) 041512.

[64] G. Tresset, Generalized Poisson–Fermi formalism for investigating size correlation effects with multiple ions, Phys. Rev. E 78 (6) (2008) 061506.