INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 12, Number 2, Pages 286–301

AN EFFECTIVE MINIMIZATION PROTOCOL FOR SOLVING A SIZE-MODIFIED POISSON-BOLTZMANN EQUATION FOR BIOMOLECULE IN IONIC SOLVENT

JIAO LI AND DEXUAN XIE

Abstract. The size-modified Poisson-Boltzmann equation (SMPBE) has been developed to consider ionic size effects in the calculation of electrostatic potential energy, but its numerical solution for a biomolecule remains a challenging research issue. To address this challenge, in this paper, we propose a solution decomposition formula and then develop an effective minimization protocol for solving the nonlinear SMPBE model by using finite element approximation techniques. As an application, a particular SMPBE numerical algorithm is constructed and programmed as a finite element program package for a biomolecule (e.g., protein and DNA) in a symmetric 1:1 ionic solvent. We also construct a nonlinear SMPBE ball model with an analytical solution and use it for validation of our new SMPBE numerical algorithm and program package. Furthermore, numerical experiments are made on a central charged ball model to show some physical features captured by the SMPBE model. Finally, they are made for six biomolecules with different net charges to demonstrate the computer performance of our SMPBE finite element program package. Numerical results show that the SMPBE model can capture some physical properties of an ionic solvent more reasonably, and can be solved more efficiently than the classic PBE model. As an application of the SMPBE model, free solvation energies were calculated and compared to the case of the PBE model.

Key words. Poisson-Boltzmann equation, variational minimization, finite element method, implicit solvent, electrostatic potential.

1. Introduction

The Poisson-Boltzmann equation (PBE) has been widely applied to the study of protein docking, ion channel modeling, and rational drug design [6, 12, 13, 20]. Despite its success in many applications, it has been well known to have some drawbacks in the calculation of electrostatic free energy for a highly charged biomolecule immersed in an ionic solvent with a high salt concentration, since it neglects ionic size effects on the biomolecular electrostatic free energy. As an improvement of the PBE model, a size-modified PBE (SMPBE) model was proposed by simply assuming that all water molecules and ions occupy the same space of a cube with side length e Λ [1]. It was recently extended to a case of nonuniform ionic sizes [2, 16, 18, 21]. However, even for the simple SMPBE model, the study of numerical solutions was only limited to a small biological molecule in a monovalent ion solution [10] and a case of one spheric ball containing a central charge or a small molecule with three atoms in a salt solution so far [2, 5, 26]. How to solve the SMPBE model effectively and efficiently for a large biomolecule in an ionic solvent remains a challenging research issue in the fields of computational biology, computational mathematics, and high performance scientific computing.

To address such a challenge, in this paper, we propose a solution decomposition formula (see Theorem 2.1), which naturally splits the SMPBE solution u, into

Received by the editors February 25, 2014.

2000 *Mathematics Subject Classification.* 92-08,65K10,65N30.

This research was partially supported by National Science Foundation, USA, through grant DMS-1226259, UWM Research Growth Initiative, and China Scholarship Council (for Jiao Li).

three component functions G, Ψ, and $\tilde{\Phi}$, respectively, according to the potential contributions from the biomolecular charges, the interface and boundary value conditions, and the ionic solvent charges. Since G collects all the singular points of u and is known in an analytical expression, the SMPBE numerical solution problem is remarkably simplified as the numerical solutions of one well defined nonlinear elliptic interface problem for $\tilde{\Phi}$ (see (5)) and one well defined linear elliptic interface problem for Ψ (see (6)). We then construct a nonlinear minimization problem and prove that this minimization problem has the same unique solution as the nonlinear interface problem so that $\tilde{\Phi}$ can be found through solving the minimization problem. In this way, we obtain an effective minimization protocol for solving the SMPBE model using finite element approximation techniques. As a result, it becomes possible for us to develop a global convergent iterative algorithm for solving the nonlinear SMPBE model. Note that setting a value of uniform size parameter Λ to be zero immediately reduces the SMPBE model to the classic PBE model. Hence, the SMPBE model contains the PBE model as a special case, and our work of this paper is an extension of our work on the classic PBE model [22, 23].

Different selections of linear and nonlinear iterative methods within our minimization protocol may result in different SMPBE numerical algorithms. As initial work, in this paper, we construct a particular SMPBE numerical algorithm for a biomolecule in a symmetric 1:1 ionic solvent by using a simple Newton minimization method. We then programmed this SMPBE algorithm as a finite element program package based on the FEniCS finite element library [17] and the molecular surface and volumetric mesh generation program package `GAMer` [24]. Our SMPBE program package is easy to use and portable on different computer systems because its main program is written in Python.

To verify our algorithm and program package, we construct a SMPBE ball model with an analytical solution (see (25)). We then solved it using our SMPBE program package. Numerical results validate our new SMPBE algorithm and program package, and show that a higher accuracy of the finite element solution can be achieved with a higher order of the finite element approximation.

We next made numerical experiments on a central charged ball immersed in the salt solution. Numerical results confirm that the SMPBE model can much better capture some physical features of ionic solvent than the classic PBE model.

Furthermore, we did numerical tests on six biomolecules with different net charges to demonstrate the computer performance of our new SMPBE program package. Although the SMPBE model is more complicated than the classic PBE model, our numerical results show that a SMPBE numerical solution could be found in less CPU time than the corresponding PBE numerical solution. For example, on one 2.4 GHz Intel Core i5 processor of a MacBook Pro, our SMPBE program package took only about 28 seconds for a biomolecule with 2124 atoms over a finite element mesh with 77,663 vertices; in contrast, the PBE program package took about 56 seconds for the corresponding PBE finite element solution [22].

Finally, as one important application, we calculated free solvation energies for the six biomolecules using the SMPBE program package and compared them with the ones calculated by our PBE program package [22]. Numerical results show that the PBE model always produced a smaller free solvation energy than the SMPBE model, and a value of the solvation energy becomes increasing when a larger value of Λ is used for the SMPBE model if Λ is treated as a purely scale parameter.

The remaining sections of the paper are outlined as follows. Section 2 introduces the SMPBE model and its solution decomposition. Section 3 presents the minimization protocol for solving the SMPBE model. Section 4 constructs a particular SMPBE algorithm from the minimization protocol. Section 5 reports the SMPBE program package and numerical results. The conclusions and future work are given in Section 6.

2. The SMPBE model

We consider a biomolecule with n_p atoms hosted in a region D_p with the boundary Γ being a molecular surface of the biomolecule. Here the position $\mathbf{r}_j$ and charge number Z_j of the jth atom of the biomolecule are given. We then select a sufficiently large ionic solvent region D_s surrounding D_p, and treat both D_p and D_s as two continuum media with different dielectric constants ϵ_p and ϵ_s satisfying that $0 < \epsilon_p < \epsilon_s$. We further assume that the solvent contains n different species of ions with z_i being the charge number of the ith ionic species, and each water molecule and each ion occupy the same space of a cube with side length Λ.

Under the above assumptions and notation, the SMPBE model can be obtained (see [16] for example) as follows:

$$(1) \qquad \begin{cases} -\epsilon_p \Delta \Phi(\mathbf{r}) = 4\pi e_c \sum\limits_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta \Phi(\mathbf{r}) - \dfrac{4\pi e_c \sum\limits_{i=1}^{n} z_i M_i e^{-z_i \Phi e_c/(k_B T)}}{1+\Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-z_k \Phi e_c/(k_B T)}} = 0, & \mathbf{r} \in D_s, \\ \Phi(\mathbf{s}^+) = \Phi(\mathbf{s}^-), \quad \varepsilon_s \dfrac{\partial \Phi(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \varepsilon_p \dfrac{\partial \Phi(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \Phi(\mathbf{s}) = \bar{g}, & \mathbf{s} \in \partial\Omega, \end{cases}$$

where Φ denotes the electrostatic potential density function, e_c is the elementary charge, $\delta_{\mathbf{r}_j}$ is the Dirac-delta distribution at point $\mathbf{r}_j$, M_i denotes the bulk concentration of the ith ionic species, k_B is the Boltzmann constant, T denotes the absolute temperature, $\mathbf{n}(\mathbf{s})$ is the unit outward normal vector of D_p, $\bar{g}$ is a given function, and $\partial\Omega$ denotes the boundary of a bounded open domain Ω defined by $\Omega = D_p \cup D_s \cup \Gamma$.

When Φ is available, the concentration function c_i of the ith ionic species can be estimated by the modified Boltzmann distributions

$$(2) \qquad c_i = \frac{M_i e^{-q_i \Phi/(k_B T)}}{1 + \Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-q_k \Phi/(k_B T)}} \quad \text{for } i = 1, 2, \ldots, n.$$

Clearly, using the variable change $u = \Phi e_c/(k_B T)$, we can reformulate (1) as the dimensionless form

$$(3) \qquad \begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum\limits_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) - \dfrac{\alpha \sum\limits_{i=1}^{n} z_i M_i e^{-z_i u}}{1+\Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-z_k u}} = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \dfrac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \dfrac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = g, & \mathbf{s} \in \partial\Omega, \end{cases}$$

where $\alpha = 4\pi e_c^2/(k_B T)$, $g = \bar{g} e_c/(k_B T)$, and u is called the dimensionless electrostatic potential.

Because of the Dirac delta distributions $\delta_{\mathbf{r}_j}$, the solution u of the SMPBE model (3) has singularities at $\mathbf{r}_j$ for $j = 1, 2, \ldots, n_p$, causing the difficulties in mathematical analysis and numerical solution. To overcome such difficulties, we obtain a solution decomposition formula in the following theorem.

Theorem 2.1. *Let u be a solution of the SMPBE model* (3)*. Then u can be split into three component functions, G, Ψ, and $\tilde{\Phi}$, in the form*

$$u = \tilde{\Phi} + \Psi + G, \tag{4}$$

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

$$\begin{cases} \Delta \tilde{\Phi}(\mathbf{r}) = 0, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta \tilde{\Phi}(\mathbf{r}) - \frac{\alpha \sum\limits_{i=1}^{n} z_i M_i e^{-z_i(\tilde{\Phi}+\Psi+G)}}{1+\Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-z_k(\tilde{\Phi}+\Psi+G)}} = 0, & \mathbf{r} \in D_s, \\ \tilde{\Phi}(\mathbf{s}^+) = \tilde{\Phi}(\mathbf{s}^-), \quad \epsilon_s \frac{\partial \tilde{\Phi}(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \frac{\partial \tilde{\Phi}(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \tilde{\Phi}(\mathbf{s}) = 0, & \mathbf{s} \in \partial\Omega, \end{cases} \tag{5}$$

Ψ is a solution of the linear interface problem

$$\begin{cases} \Delta \Psi(\mathbf{r}) = 0, & \mathbf{r} \in D_p \cup D_s, \\ \Psi(\mathbf{s}^+) = \Psi(\mathbf{s}^-), & \mathbf{s} \in \Gamma, \\ \epsilon_s \frac{\partial \Psi(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \frac{\partial \Psi(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_p - \epsilon_s) \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ \Psi(\mathbf{s}) = g - G(\mathbf{s}), & \mathbf{s} \in \partial\Omega, \end{cases} \tag{6}$$

and G is given by

$$G(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} \frac{Z_j}{|\mathbf{r} - \mathbf{r}_j|}. \tag{7}$$

Here $\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} = \nabla G \cdot \mathbf{n}$ with ∇G being given by

$$\nabla G(\mathbf{r}) = -\frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} Z_j \frac{\mathbf{r} - \mathbf{r}_j}{|\mathbf{r} - \mathbf{r}_j|^3}. \tag{8}$$

Proof. Since $\frac{1}{4\pi|\mathbf{r}-\mathbf{r}_j|}$ satisfies the equation $-\Delta G_j = \delta_{\mathbf{r}_j}$,

$$\Delta G = \frac{\alpha}{\epsilon_p} \sum_{j=1}^{n_p} Z_j \Delta \frac{1}{4\pi|\mathbf{r} - \mathbf{r}_j|} = -\frac{\alpha}{\epsilon_p} \sum_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}.$$

For $\mathbf{r} \in D_p$, we have $\Delta\Psi = 0$ and $\Delta\tilde{\Phi} = 0$. Thus,

$$-\epsilon_p \Delta u = -\epsilon_p(\Delta\tilde{\Phi} + \Delta\Psi + \Delta G) = -\epsilon_p(-\frac{\alpha}{\epsilon_p} \sum_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}) = \alpha \sum_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}.$$

For $\mathbf{r} \in D_s$, we can easily verify that

$$-\epsilon_s \Delta u = -\epsilon_s \Delta\tilde{\Phi} = \frac{\alpha \sum\limits_{i=1}^{n} z_i M_i e^{-z_i(\tilde{\Phi}+\Psi+G)}}{1 + \Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-z_k(\tilde{\Phi}+\Psi+G)}} = \frac{\alpha \sum\limits_{i=1}^{n} z_i M_i e^{-z_i u}}{1 + \Lambda^3 \sum\limits_{k=1}^{n} M_k e^{-z_k u}},$$

since $\Delta G = 0$ and $\Delta\Psi = 0$ in D_s.

On the interface Γ, $\tilde{\Phi}$ and Ψ satisfy the interface conditions given in (5) and (6), respectively, while both $G(\mathbf{s})$ and $\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}$ are continuous. As a result, for any $\mathbf{s} \in \Gamma$, we have

$$u(s^+) = \tilde{\Phi}(s^+) + \Psi(s^+) + G(s) = \tilde{\Phi}(s^-) + \Psi(s^-) + G(s) = u(s^-)$$

and

$$\begin{aligned} \epsilon_p \frac{\partial u(s^-)}{\partial \mathbf{n}(\mathbf{s})} &= \epsilon_p \frac{\partial \tilde{\Phi}(s^-)}{\partial \mathbf{n}(\mathbf{s})} + \epsilon_p \frac{\partial \Psi(s^-)}{\partial \mathbf{n}(\mathbf{s})} + \epsilon_p \frac{\partial G(s)}{\partial \mathbf{n}(\mathbf{s})} \\ =&\epsilon_s \frac{\partial \tilde{\Phi}(s^+)}{\partial \mathbf{n}(\mathbf{s})} + \epsilon_s \frac{\partial \Psi(s^+)}{\partial \mathbf{n}(\mathbf{s})} - (\epsilon_p - \epsilon_s) \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} + \epsilon_p \frac{\partial G(s)}{\partial \mathbf{n}(\mathbf{s})} \\ =&\epsilon_s \frac{\partial}{\partial \mathbf{n}(\mathbf{s})} (\tilde{\Phi} + \Psi + G)(s^+) = \epsilon_s \frac{\partial u(s^+)}{\partial \mathbf{n}(\mathbf{s})}. \end{aligned}$$

Finally, for $\mathbf{s} \in \partial\Omega$, we get that

$$u(\mathbf{s}) = \tilde{\Phi}(\mathbf{s}) + \Psi(\mathbf{s}) + G(\mathbf{s}) = 0 + g - G(\mathbf{s}) + G(\mathbf{s}) = g \quad \forall \mathbf{s} \in \partial\Omega.$$

This completes the proof. □

It is interesting to note that with our solution decomposition (4), we have naturally split the potential u into the three parts G, Ψ, and $\tilde{\Phi}$ according to the potential contributions from the biomolecular charges, the interface and boundary value conditions, and the ionic solvent charges, respectively. Since G is known, it only needs to find Ψ and $\tilde{\Phi}$ to yield a solution u of the SMPBE model (3) without any singularity difficulty.

Clearly, setting $\Lambda = 0$ immediately reduces the SMPBE model (3) to the classic PBE model:

$$(9) \qquad \begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum\limits_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) - \alpha \sum\limits_{i=1}^{n} z_i M_i e^{-z_i u} = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = g, & \mathbf{s} \in \partial\Omega. \end{cases}$$

Thus, the PBE model can be regarded as a special case of the SMPBE model. Since the work on the PBE model has been done in [22], we only consider the SMPBE model using a positive value of Λ in this paper.

3. Our minimization protocol for solving SMPBE

In this section, we present a minimization protocol for solving the SMPBE model (3) numerically by the finite element method. For a given tetrahedral mesh partition of domain Ω, we can construct a Lagrange finite element space $\mathcal{M}$ as a finite dimensional subspace of the usual Sobolev function space $H^1(\Omega)$ such that each function in $\mathcal{M}$ is continuous [4]. We then set

$$\mathcal{M}_0 = \{v \in \mathcal{M} \mid v = 0 \text{ on } \partial\Omega\},$$

as a subspace of the Sobolev function space $H_0^1(\Omega)$.

With Green's first identity, we can formulate the linear interface problem (6) as the following finite element variational problem:

$$\text{Find } \Psi \in \mathcal{M} \text{ with } \Psi|_{\partial\Omega} = g - G \text{ such that}$$

$$a(\Psi, v) = (\epsilon_p - \epsilon_s) \int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} \quad \forall v \in \mathcal{M}_0, \tag{10}$$

where $a(u, v)$ is a bilinear form defined by

$$a(u, v) = \epsilon_p \int_{D_p} \nabla u(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} + \epsilon_s \int_{D_s} \nabla u(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r}. \tag{11}$$

Note that in the weak form (10), we have used the identity

$$\int_\Gamma \frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} v(\mathbf{s}) \mathrm{d}\mathbf{s} = -\int_{D_s} \nabla G(\mathbf{r}) \cdot \nabla v(\mathbf{r}) d\mathbf{r} \quad \forall v \in H_0^1(\Omega)$$

to avoid the difficulty that may be caused by the complexity of the interface Γ.

Similarly, the nonlinear interface problem (5) can be reformulated as the finite element nonlinear variational problem:

$$\text{Find } \tilde{\Phi} \in \mathcal{M}_0 \text{ such that}$$

$$a(\tilde{\Phi}, v) - \int_{D_s} \frac{\alpha \sum\limits_{i=1}^n z_i P_i e^{-z_i \tilde{\Phi}}}{1 + \Lambda^3 \sum\limits_{k=1}^n P_k e^{-z_k \tilde{\Phi}}} v d\mathbf{r} = 0 \quad \forall v \in \mathcal{M}_0, \tag{12}$$

where $a(u, v)$ is defined in (11), and P_i is defined by

$$P_i = M_i e^{-z_i(\Psi + G)}.$$

In calculation, we always suppose that Ψ has been found before solving the nonlinear variational problem (12). Thus, P_i can be treated as a given function.

Obviously, $a(u, v)$ is a bounded and symmetric bilinear form. The solution existence and uniqueness of the variational problem (10) can be easily followed from the regularity analysis of the elliptic interface problem [3, 8, 19] if the interface Γ is assumed to be of class C^2.

For the case of the nonlinear variational problem (12), we construct a minimization problem as follows

$$\min_{v \in \mathcal{M}_0} J(v) \tag{13}$$

where J is a functional defined by

$$J(v) = \frac{1}{2} a(v, v) + \frac{\alpha}{\Lambda^3} \int_{D_s} \ln(1 + \Lambda^3 \sum_{k=1}^n P_k e^{-z_k v}) d\mathbf{r}, \quad v \in \mathcal{M}_0. \tag{14}$$

We then obtain the following theorem.

Theorem 3.1. *The minimization problem* (13) *has a unique solution and is equivalent to the nonlinear variational problem* (12).

Proof. Clearly, J is continuous and twice Fréchet differentiable on $\mathcal{M}_0$. Thus, J is lower semicontinuous, and its first and second derivatives $J'(\phi)$ and $J''(\phi)$ at $\phi \in \mathcal{M}_0$ can be found as follows: For any $v, w \in H_0^1(\Omega)$,

$$J'(\phi)v = a(\phi, v) - \int_{D_s} \frac{\alpha \sum\limits_{i=1}^n z_i P_i e^{-z_i \phi}}{1 + \Lambda^3 \sum\limits_{k=1}^n P_k e^{-z_k \phi}} v d\mathbf{r}, \tag{15}$$

and

$$J''(\phi)(v,w) = a(w,v) + \int_{D_s} \frac{\alpha \sum_{i=1}^{n} z_i^2 P_i e^{-z_i\phi}}{1 + \Lambda^3 \sum_{k=1}^{n} P_k e^{-z_k\phi}} wv d\mathbf{r}$$

$$- \int_{D_s} \frac{\alpha\Lambda^3 (\sum_{i=1}^{n} z_i P_i e^{-z_i\phi})^2}{(1 + \Lambda^3 \sum_{k=1}^{n} P_k e^{-z_k\phi})^2} wv d\mathbf{r}. \tag{16}$$

By Poincaré inequality and the facts that $\epsilon_s > \epsilon_p$ and

$$\frac{\alpha}{\Lambda^3} \int_{D_s} \ln(1 + \Lambda^3 \sum_{k=1}^{n} P_k e^{-z_k v}) d\mathbf{r} \geq 0,$$

there exists a constant $C > 0$ such that

$$J(v) = \frac{1}{2}a(v,v) + \frac{\alpha}{\Lambda^3} \int_{D_s} \ln(1 + \Lambda^3 \sum_{k=1}^{n} P_k e^{-z_k v}) d\mathbf{r}$$
$$\geq \epsilon_p \int_{\Omega} |\nabla v(\mathbf{r})|^2 d\mathbf{r} \geq \frac{\epsilon_p}{C} \|v\|^2_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega)$$

from which it implies $J(v) \to \infty$ as $\|v\|_{H^1(\Omega)} \to \infty$ (i.e., J is coercive on $H_0^1(\Omega)$).

Furthermore, it is easy to show that the difference of the last two terms of $J''(\phi)(v,v)$ in (16) is positive. Thus, for any $\phi \in \mathcal{M}_0$, we have

$$J''(\phi)(v,v) \geq \frac{1}{2}a(v,v) \geq \frac{\epsilon_p}{C} \|v\|^2_{H^1(\Omega)} \quad \forall v \in H_0^1(\Omega).$$

Hence, J is strictly convex. Therefore, from [11, Proposition 1.2, Page 35] it implies that the minimization problem (13) has a unique solution $u \in \mathcal{M}_0$. Moreover, the minimizer u satisfies $J'(u)v = 0$ for all $v \in H_0^1(\Omega)$, which is the same as (12). Hence, u is also a solution to (12). The equivalence can be directly followed from [25, Theorem 40.B, Page 194]. □

We are now in the position to construct our minimization protocol for solving the SMPBE model in the following algorithm.

Algorithm 1 (*Minimization Protocol for Solving SMPBE*) Let u be a finite element solution of the SMPBE model (3) on a Lagrange finite element space $\mathcal{M}$. It can be calculated in the following steps:

(1) *Calculate G by* (7) *and ∇G by* (8) *on $\mathcal{M}$.*
(2) *Solve the finite element variational problem* (10) *to get Ψ.*
(3) *Search for a minimizer $\tilde{\Phi}$ of the minimization problem* (13) *as a finite element solution of* (12).
(4) *Construct u by the decomposition formula $u = G + \Psi + \tilde{\Phi}$.*

4. A particular algorithm for solving SMPBE

Clearly, different linear and minimization solvers in Steps 2 and 3 of Algorithm 1 may lead to different numerical algorithms for solving the SMPBE model. For the purpose of numerically testing our minimization protocol, in this section, we only consider a symmetric 1:1 ionic solvent, and construct a Newton minimization algorithm for solving the minimization problem (13) as defined in Algorithm 2.

Algorithm 2. (*A Newton minimization method*) Let $\{\phi^{(k)}\}$ be a sequence of the Newton minimization method for solving the minimization problem (13). For a given initial guess $\phi^{(0)} \in \mathcal{M}_0$, do the following steps for $k = 0, 1, 2, \ldots$:

(1) Convergence Test*: If* $\|J'(\phi^{(k)})\| < \epsilon$ *(in our tests,* $\epsilon = 10^{-6}$*), stop the iteration and output* $\phi^{(k)}$ *as the numerical solution.*

(2) Determine a descent search direction p_k: *Select* $p_k \in \mathcal{M}_0$ *as a numerical solution of the Newton equation*

$$J''(\phi^{(k)})(p, v) = -J'(\phi^{(k)})v \quad \forall v \in \mathcal{M}_0. \tag{17}$$

(3) Selection of a steplength λ_k: *Set* $\lambda_k = 1$ *if*

$$J(\phi^{(k)} + \lambda_k p_k) \le J(\phi^{(k)}) \quad or \quad \|J'(\phi^{(k)} + \lambda_k p_k)\| \le \|J'(\phi^{(k)})\|; \tag{18}$$

otherwise, select λ_k *by a line search algorithm to satisfy* (18).

(4) Define the next iterate $\phi^{(k+1)}$: *Set* $\phi^{(k+1)} = \phi^{(k)} + \lambda_k p_k$.

In the case of the symmetric 1:1 ionic solvent, we have $n = 2$, $z_1 = 1, z_2 = -1$, and $M_1 = M_2 = M$. Thus, the SMPBE model (3) is simplified as

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} Z_j \delta_{\mathbf{r}_j}, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) + \frac{2M\alpha \sinh(u)}{1+2M\Lambda^3 \cosh(u)} = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \dfrac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_p \dfrac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = g, & \mathbf{s} \in \partial\Omega. \end{cases} \tag{19}$$

For the above SMPBE model, Ψ and G are the same as the ones defined in (6) and (7), but the variational form (12) of $\tilde{\Phi}$ has been simplified in the form

$$a(\tilde{\Phi}, v) + 2M\alpha \int_{D_s} \frac{\sinh(\tilde{\Phi} + \Psi + G)}{1 + 2M\Lambda^3 \cosh(\tilde{\Phi} + \Psi + G)} v d\mathbf{r} = 0. \tag{20}$$

Correspondingly, for $\Lambda > 0$, the functional J of (14) and its first and second Gâteaux derivatives of (15) and (16) are also simplified into the following forms

$$J(v) = \frac{1}{2} a(v, v) + \frac{\alpha}{\Lambda^3} \int_{D_s} \ln(1 + 2M\Lambda^3 \cosh(v + \Psi + G)) d\mathbf{r}, \tag{21}$$

$$J'(\phi)v = a(\phi, v) + 2M\alpha \int_{D_s} \frac{\sinh(\phi + \Psi + G)}{1 + 2M\Lambda^3 \cosh(\phi + \Psi + G)} v d\mathbf{r} \tag{22}$$

$$J''(\phi)(v, w) = a(w, v) + 2M\alpha \int_{D_s} \frac{2M\Lambda^3 + \cosh(\phi + \Psi + G)}{(1 + 2M\Lambda^3 \cosh(\phi + \Psi + G))^2} w v d\mathbf{r}$$

Clearly, the variational form (20) can be linearized as

$$a(\tilde{\Phi}, v) + \frac{2M\alpha}{1 + 2M\Lambda^3} \int_{D_s} \tilde{\Phi} v d\mathbf{r} = \frac{-2M\alpha}{1 + 2M\Lambda^3} \int_{D_s} (\Psi + G) v d\mathbf{r}, \tag{23}$$

whose finite element solution can be selected as a starting iterate $\phi^{(0)}$ for our Newton minimization algorithm.

In practice, the ionic strength I_s is often given in *mole per liter (mol/L)*. In this case, the bulk concentration M is expressed by

$$M = 10^{-27} N_A I_s,$$

where N_A is the Avogadro constant, which is given as $N_A = 6.02214129 \times 10^{23}$.

Similar to the case of the PBE model [14], we can obtain the following approximate boundary function g for the SMPBE model (19):

$$g = \frac{e_c^2}{\epsilon_s k_B T} \sum_{j=1}^{n_p} \frac{e^{-\bar{\kappa}|\mathbf{r}-\mathbf{r}_j|/\sqrt{\epsilon_s(1+2M\Lambda^3)}}}{|\mathbf{r}-\mathbf{r}_j|} z_j. \tag{24}$$

5. Program package and numerical results

We programmed our SMPBE algorithm as a finite element program package in Python based on the library DOLFIN from the FEniCS project [17]. The mesh generation program package GAMer [24] was adapted as a Python module of our SMPBE program by the software development tool SWIG (*http://www.swig.org*). Thus, we can generate a finite element mesh directly within our program package for each input biomolecule represented in a PQR file. To improve the computer performance, we wrote Fortran subroutines for computing the mesh node values of functions G, ∇G, g, ln, sinh, and cosh, and converted them as Python modules by the Fortran to Python interface generator F2PY (*http://cens.ioc.ee/projects/f2py2e/*). The preconditioned conjugate gradient method with the ILU preconditioner from the PETSc library (*http://www.mcs.anl.gov/petsc/*) was used to solve a system of linear algebraic equations arisen from the finite element variational problem (10) and the Newton's equation (17).

All the numerical experiments were made on one 2.4 GHz Intel Core i5 processor of a MacBook Pro with 8GB memory. Here, we set the two iterative termination parameters from the PETSc library, *relative_tolerance* and *absolute_tolerance*, as 10^{-10}. All the numerical tests on the corresponding PBE model (9) were done by using the PBE program package [22].

5.1. Tests on a SMPBE ball model with analytical solution. To validate our SMPBE algorithm and program package, we construct the SMPBE ball model

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha Z \delta, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) + \frac{2M\alpha \sinh(u(\mathbf{r}))}{1+2M\Lambda^3 \cosh(u(\mathbf{r}))} = \rho_s(\mathbf{r}), & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}} = \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = \frac{\alpha Z}{4\pi\epsilon_s |\mathbf{s}|}, & \mathbf{s} \in \partial\Omega, \end{cases} \tag{25}$$

where $\Omega = \{\mathbf{r} : |\mathbf{r}| < R_s\}$, $D_p = \{\mathbf{r} : |\mathbf{r}| < R_p\}$, $D_s = \{\mathbf{r} : R_p < |\mathbf{r}| < R_s\}$, $\Gamma = \{\mathbf{r} : |\mathbf{r}| = R_p\}$, Z is a charge number, δ is the Dirac delta distribution at the origin, and the right hand side function ρ_s is defined by

$$\rho_s(\mathbf{r}) = 2M\alpha \sinh\left(\frac{\alpha Z}{4\pi\epsilon_s|\mathbf{r}|}\right)\left[1 + 2M\Lambda^3 \cosh\left(\frac{\alpha Z}{4\pi\epsilon_s|\mathbf{r}|}\right)\right]^{-1} \quad \text{for } \mathbf{r} \in D_s.$$

Clearly, the analytical solution of this SMPBE ball model is given by

$$u(\mathbf{r}) = \begin{cases} \frac{\alpha Z}{4\pi R_p}\left(\frac{1}{\epsilon_s} - \frac{1}{\epsilon_p}\right) + \frac{\alpha Z}{4\pi\epsilon_p|\mathbf{r}|}, & \mathbf{r} \in D_p, \\ \frac{\alpha Z}{4\pi\epsilon_s|\mathbf{r}|}, & \mathbf{r} \in D_s, \end{cases} \tag{26}$$

which is the same as the analytical solution of the Born ball model:

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha Z \delta, & \mathbf{r} \in D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) = 0, & \mathbf{r} \in D_s, \\ u(\mathbf{s}^+) = u(\mathbf{s}^-), \quad \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}} = \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}}, & \mathbf{s} \in \Gamma, \\ u(\mathbf{s}) = \frac{\alpha Z}{4\pi\epsilon_s |\mathbf{s}|}, & \mathbf{s} \in \partial\Omega. \end{cases} \tag{27}$$

To solve the SMPBE ball model (25), the only modification we made to the SMPBE program package was to subtract the term $\int_{D_s} \rho_s(\mathbf{r}) v d\mathbf{r}$ from the expressions (21) and (22) of $J(v)$ and $J'(\phi)v$, respectively. In this test, we set $R_p = 1$ and $R_s = 10$ to use the tetrahedral mesh with 2955 vertices from in [22]. $I_s = 0.1$ and 0.2 are two commonly used ionic strength values. Linear, quadratic, and cubic finite element methods were used. An initial guess of zero was set in the Newton minimization algorithm. Numerical results are reported in Table 1. Here Error_r denotes the relative error of a finite element solution u_h with respect to the analytical solution u as defined by

$$\text{Error}_r = \|u_h - u\|_{L^2(\Omega)} / \|u\|_{L^2(\Omega)}, \tag{28}$$

where $\|\cdot\|_{L^2(\Omega)}$ is the norm defined by $\|v\|_{L^2(\Omega)} = (\int_\Omega |v|^2 d\mathbf{r})^{\frac{1}{2}}$ for $v \in L^2(\Omega)$.

From Table 1 it can be seen that the accuracy of a finite element solution is high and can be improved significantly as the order of the finite element method is increased from 1 to 3. These numerical results well validated our SMPBE algorithm and program package.

Table 1. Relative errors of the finite element solutions of the SMPBE ball model (25) with $R_p = 1$, $R_s = 10$, and $\Lambda = 6$. Here, Error_r is defined in (28), and FEM stands for Finite Element Method.

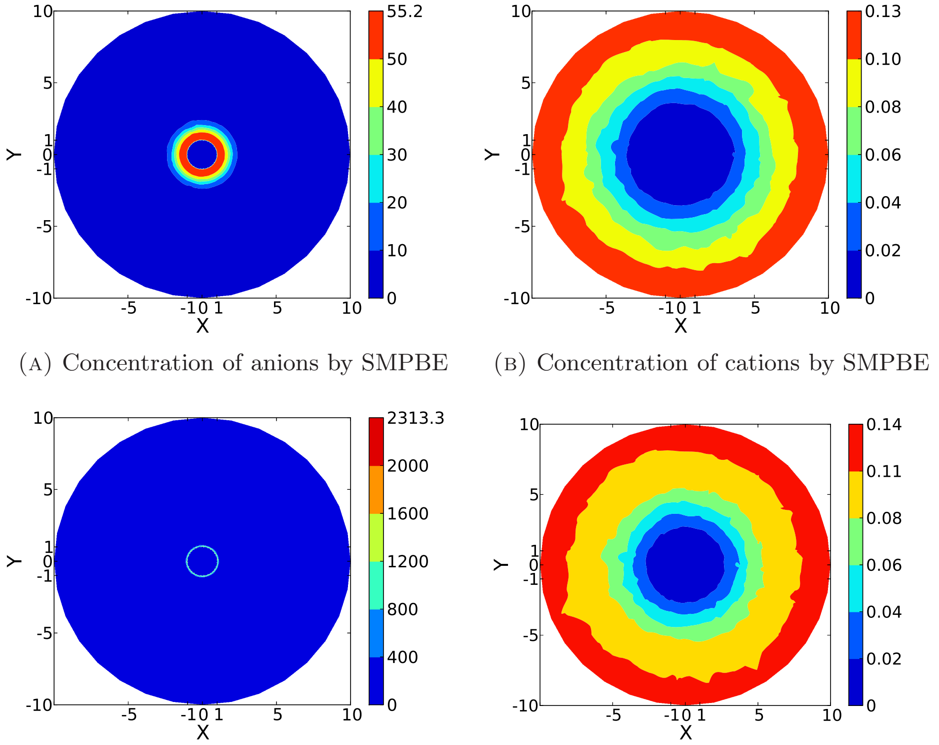
Order	Error_r $(Z = 1)$		Error_r $(Z = 3)$	
of FEM	$I_s = 0.1$	$I_s = 0.2$	$I_s = 0.1$	$I_s = 0.2$
$1st$	2.8890×10^{-3}	2.8925×10^{-3}	3.1599×10^{-3}	3.2098×10^{-3}
$2nd$	4.7170×10^{-4}	4.6829×10^{-4}	4.6535×10^{-4}	4.6385×10^{-4}
$3rd$	2.2023×10^{-4}	2.1976×10^{-4}	2.1854×10^{-4}	2.1837×10^{-4}

5.2. Tests on a central charged ball immersed in salt solution. To demonstrate that the SMPBE model can capture some physical features of an ionic solvent, we made numerical experiments on the SMPBE model (19) with $I_s = 0.2$, the solute region D_p being a unit ball, $D_p = \{\mathbf{r} \mid |\mathbf{r}| < 1\}$, with only one charge Ze_c at the center of the ball ($n_p = 1$), the solvent region $D_s = \{\mathbf{r} \mid 1 < |\mathbf{r}| < 10\}$, the boundary function g defined in (24), and $\Lambda = 3.11$. Such a value of Λ estimates the linear size of water molecule, which is calculated at temperature $298.15K$. At this temperature, the density of liquid water is 997.0479 Kg/m^3, and the molar mass of water is 18.01528 g/mol.

With the cubic finite element solution calculated on the same tetrahedral mesh as the one used in the previous subsection, we obtained the concentration functions c_1 and c_2 of anions and cations according to the expression of (2) with $n = 2$, $q_1 = -e_c$ and $q_2 = e_c$. One cross section of each concentration function from the xy coordinate plane is displayed on Figure 1 for both the SMPBE and PBE models.

From Plots (B) and (D) of Figure 1 it can be seen that both the SMPBE and PBE models captured such an ionic solvent feature of cations (e.g., sodium ions N_a^+): the cations are repelled away from the positive charged unit ball.

However, as shown in Plot (C), the PBE calculated concentration of anions (e.g., chloride ions Cl^-) had too high values around the unit ball within a very tiny ring, indicating that all the anions unrealistically accumulated on the spherical surface of the unit ball due to all the ions being treated as volumeless points.



(A) Concentration of anions by SMPBE

(B) Concentration of cations by SMPBE

(C) Concentration of anions by PBE

(D) Concentration of cations by PBE

Figure 1. A comparison of the concentrations calculated from the SMPBE and PBE models for a unit ball with one central point charge immersed in a symmetric 1:1 ionic solvent containing anions (e.g., chloride ion Cl^-) and cations (e.g., sodium ion N_a^+). Here $\Lambda = 3.11$ $\AA$, $I_s = 0.2$ M, $Z = 3$, $R_p = 1$, and $R_s = 10$.

In contrast, from Plot (A) it can be seen that the SMPBE calculated concentration of anions reached a saturation value 55.2 (i.e., $1/\Lambda^3$) around the spherical surface, which matched well what is claimed in Physics.

5.3. Tests on biomolecules with different net charges. To demonstrate the computer performance of our SMPBE program package, we made numerical experiments on six biomolecules with different net charges. These six biomolecules (three proteins, two DNA-protein complexes, and one DNA) were downloaded from the Protein Data Bank (PDB) (*http://www.rcsb.org*). Their PDB identifications (ID), net charges, and atom numbers are listed in Table 2.

Using the software tool PDB2PQR (*http://www.poissonboltzmann.org/pdb2pqr*), we converted each PDB file to a PQR file as an input file of our SMPBE program package. Six tetrahedral meshes were generated, respectively, by calling `GAMer` [24] within our program package for these six biomolecules located in the center parts of the six spherical ball domains Ω with radii 90.2, 127.5, 111.6, 149.7, 102.0, and 198.9 in units $\AA$ respectively. Their total numbers of vertices are listed in Table 2.

A tetrahedral mesh for the protein represented by PDB ID *1CBN* is displayed in Figure 2 as an example to show the challenge to generate a high quality finite

TABLE 2. The six biomolecules used in our numerical experiments.

PDB ID	Type	# Atom	Net charges	Mesh vertices
1CBN	Protein	642	$0e_c$	69949
1SVR	Protein	1433	$-2e_c$	52715
4PTI	Protein	892	$+6e_c$	73195
1AZQ	DNA-protein complex	1603	$-8e_c$	51380
1D3X	DNA	756	$-21e_c$	42358
1TC3	DNA-protein complex	2124	$-35e_c$	77663

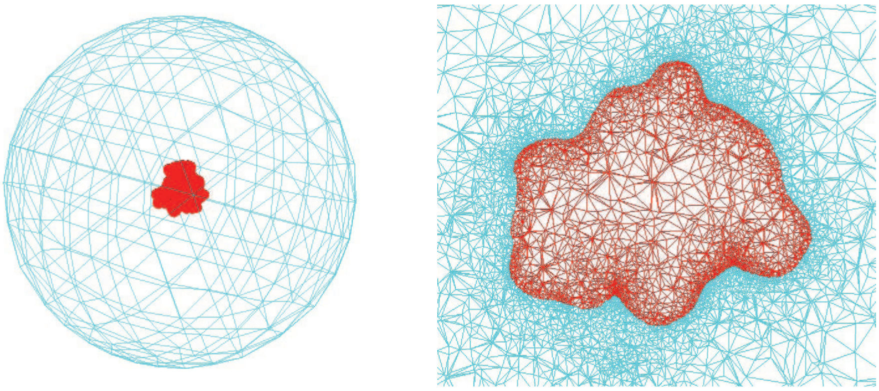


FIGURE 2. A tetrahedral mesh for a protein represented in PDB ID *1CBN*. Here the protein range D_p and solvent region D_s are marked in red and cyan, respectively. A part of cross section of the mesh is displayed in the right figure to show the mesh around the protein.

element mesh due to a complex molecular surface of a biomolecule. This figure is also used as a demonstration to show that the meshes generated from `GAMer` were satisfactory for our numerical experiments.

The numerical tests were done by using the linear finite element method, the boundary function g defined in (24), $I_s = 0.2$, and $\Lambda = 3.11$. An initial guess for the Newton minimization method was set as the corresponding numerical solution of the linear variational problem (23). We also simply treated Λ as a scaling parameter (i.e., ignoring its physical sense), and did numerical tests using $\Lambda = 6$ and 10. As comparisons, we also did tests on the corresponding PBE model using our PBE program package [22] with the same parameters as Algorithm 2. Computer performance data are reported in Table 3 and Figure 3. Here *Time* is the total amount of CPU time in seconds excluding the time for mesh generation. In fact, the same meshes were used in PBE and SMPBE tests. Thus, the time for mesh generation can be omitted in a performance comparison between PBE and SMPBE. One discussion on it can be found in [22] for the case of PBE.

From Table 3 we can see that while both the PBE and SMPBE program packages were efficient, the SMPBE package performed better than the PBE package in terms of the total CPU time. This result is beyond what we expected since the SMPBE model is a more complicated model than the PBE model so that each of its Newton iterations is more expensive to be computed than the case of the PBE model.

Table 3. Comparison of the our SMPBE program package with our PBE program package [22] in the total number of iterations of the Newton minimization method(Iter) and the total CPU time (Time in seconds).

PDB ID	PBE ($\Lambda = 0$)		SMPBE					
			$\Lambda = 3.11$		$\Lambda = 6$		$\Lambda = 10$	
	Iter	Time	Iter	Time	Iter	Time	Iter	Time
1CBN	8	25.00	6	21.31	5	19.41	4	17.70
1SVR	21	33.25	7	17.40	5	15.19	4	14.09
4PTI	10	29.91	9	27.74	8	26.03	6	22.53
1AZQ	19	29.74	8	18.32	7	17.00	5	14.97
1D3X	24	26.85	14	17.73	12	16.04	6	11.35
1TC3	21	55.57	12	38.47	11	36.88	6	28.40

Interestingly, as shown in Figure 3, our Newton minimization method achieved a faster rate of convergence for the SMPBE model than the PBE model, and the rate can be speeded up by using a larger value of the scaling parameter Λ, resulting in a sharp reduction of the total CPU time.

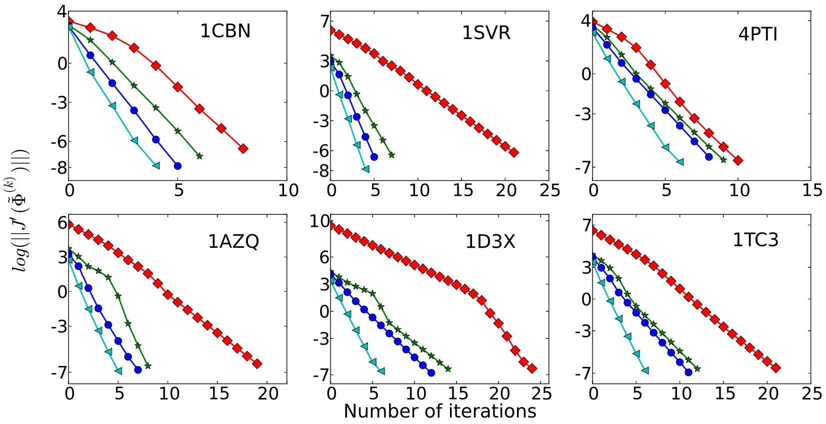


Figure 3. A comparison of the convergence rate of the Newton minimization method for the SMPBE model with that for the corresponding PBE model in terms of the total number of iterations. Here, the convergence rule is that $\|J'(\phi^{(k)})\| < 10^{-6}$ (see Algorithm 2), $I_s = 0.2$, the PBE case is marked in red diamonds, and the SMPBE using $\Lambda = 3.11, 6$, and 10 are marked in green stars, blue circles, and cyan triangles, respectively.

5.4. Solvation energy calculations. As one important application, we calculated the free solvation energy for the six biomolecules using the SMPBE program package and made a comparison with that calculated by our PBE program package [22]. The free solvation energy is an interesting physical quantity associated with the free energy changes of a charged system from the vacuum into a solvent. As a

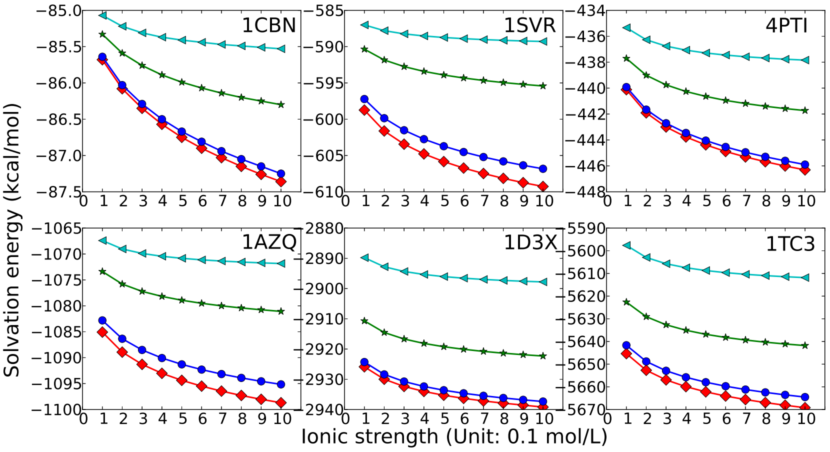


FIGURE 4. Comparison of the solvation free energy calculated by the PBE model (in red diamonds) with that by the SMPBE model using $\Lambda = 2$ (in blue circles), 6 (in green stars), or 10 (in cyan triangles). Here the tests were done with the ionic strength I_s being $1, 2, \ldots, 10$ in units 0.1mol/L, respectively.

result of our solution decomposition (4), it can be calculated by

$$F_{sol} = \frac{1.439 \times 10^{13}}{2} k_B T \sum_{j=1}^{n_p} Z_j(\tilde{\Phi}(\mathbf{r}_j) + \Psi(\mathbf{r}_j)). \tag{29}$$

Here the energy has the unit *kcal/mol* as the number 1.439×10^{13} is the energy unit converting constant from *erg* to *kcal/mol*.

With the finite element solutions produced from our SMPBE and PBE program packages, we calculated the solvation energies for the six biomolecules with different ionic strengths (from 0.1 *mol/L* to 1 *mol/L*) and different values of the uniform ionic size parameter Λ (from 0 to 10 Å, treating it as a purely scaling parameter). The results were plotted in Figure 4.

Figure 4 shows that the solvation energy F_{sol} is an increase function of Λ for each fixed value of the ionic strength I_s while a decrease function of I_s for each fixed value of Λ. Thus, the PBE model produced smaller values of F_{sol} than the SMPBE model. It is also interesting to note that the difference between the solvation energies calculated by the SMPBE and PBE models become larger for a larger value of I_s so that the largest difference occurred at $\Lambda = 10$ *Å* and $I_s = 1$ *mol/L* for each biomolecule. Among all the six biomolecules, the largest difference was 60 *kcal/mol* in the case of 1TC3, but, in comparison to the large magnitude (5670) of the solvation energy, it was relatively very small ($60/5670 \approx 0.01058$).

6. Conclusions and future works

In this paper, we have proposed a solution decomposition formula such that the SMPBE solution u is split as a sum of three component functions G, Ψ, and $\tilde{\Phi}$. Here, these three functions are defined naturally according to the potential contributions from the biomolecular charges, the interface and boundary value conditions, and the ionic solvent charges. As an application of this solution decomposition formula, we have developed an effective minimization protocol using finite element approximation techniques to numerically solve the nonlinear SMPBE model for a biomolecule (e.g., protein or DNA) immersed in an ionic solvent containing n different kinds of ions. This general minimization protocol makes it possible to construct different fast SMPBE numerical solvers through selecting different linear/nonlinear iterative solvers or different minimization algorithms for computing Ψ and $\tilde{\Phi}$. As an example, in this paper, we have obtained one particular SMPBE algorithm and programmed it as a computer program package in the case of a biomolecule in a symmetric $1:1$ ionic solvent (i.e., the case of $n=2$ with two opposite unit charged ions), which is often studied in literature. This particular SMPBE algorithm and program package were numerically tested and validated on a SMPBE ball model we constructed with a given analytical solution in this paper. The high performance of our SMPBE program package was demonstrated for six biomolecules with different net charges. Furthermore, we have done comparison tests between the SMPBE and PBE models, showing that the SMPBE model is a more reasonable than the PBE model in the calculation of electrostatic solvation free energies.

We note that two other different solution decomposition formulas were proposed for the PBE model in the case of a symmetric 1:1 ionic solvent [7, 9, 15, 27]. Even in this special case, they are very different from ours. For example, in the one proposed in [9], the component function Ψ was defined only within the protein range D_p, while the one from [7] used different boundary value functions from ours in the construction of the elliptic interface problems of Ψ and $\tilde{\Phi}$.

In the future, we plan to do more theoretical and numerical studies on the SMPBE model to further understand its physical and mathematical properties. We will also carry out a theoretical analysis on the Newton minimization method to understand why it has a faster rate of the convergence in the case of the SMPBE model than in the case of the PBE model. Moreover, we will look for a more efficient mesh generator and more efficient linear and nonlinear iterative methods for calculating Ψ and $\tilde{\Phi}$ to further improve the performance of our SMPBE finite element algorithm and program package.

References

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

[2] A. H. Boschitsch and P.V. Danilov, Formulation of a new and simple nonuniform size-modified Poisson-Boltzmann description, J. Comput. Chem., 33 (2012) 1152-1164.

[3] J.H. Bramble and J.T. King, A finite element method for interface problems in domains with smooth boundaries and interfaces, Adv. Comput. Math., 6 (1996) 109-138.

[4] S.C. Brenner and L.R. Scott, The Mathematical Theory of Finite Element Methods, volume 15 of Texts in Applied Mathematics, Springer-Verlag, New York, 2nd edition, 2002.

[5] J.H. Chaudhry, S.D. Bond and L.N. Olson, Finite element approximation to a finite-size modified Poisson-Boltzmann equation, J. Sci. Comput., 47 (2011) 347-364.

[6] J. Chen, C. L Brooks III and J. Khandogin, Recent advances in implicit solvent-based methods for biomolecular simulations, Curr. Opin. Struct. Biol., 18 (2008) 140-148.

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

[8] Z. Chen and J. Zou, Finite element methods and their convergence for elliptic and parabolic interface problems, Numer. Math., 79 (1998) 175-202.

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

[10] V. B Chu, Y. Bai, J. Lipfert, D. Herschlag and S. Doniach, Evaluation of ion binding to DNA duplexes using a size-modified Poisson-Boltzmann theory, Biophys. J., 93 (2007) 3202-3209.

[11] I. Ekeland, R. Témam, Convex Analysis and Variational Problems, SIAM, Philadelphia, 1999.

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

[13] F. Fogolari, A. Brigo and H. Molinari, The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology, J Mol. Recognit., 15 (2002) 377-392.

[14] M.J. Holst, The Poisson-Boltzmann equation: Analysis and multilevel numerical solution, PhD thesis, University of Illinois at Urbana-Champaign, 1994.

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

[16] B. Li, Continuum electrostatics for ionic solutions with non-uniform ionic sizes, Nonlinearity, 22 (2009) 811-833.

[17] A. Logg, G. N. Wells and J. Hake, *DOLFIN: A C++/Python finite element library.* In Automated Solution of Differential Equations by the Finite Element Method, volume 84 of Lect. Notes Comput. Sci. Eng., Chapter 10, 173-225. Springer, Heidelberg, 2012.

[18] B. Lu and Y. Zhou, Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes II: Size effects on ionic distributions and diffusion-reaction rates, Biophys. J., 100 (2011) 2475-2485.

[19] G. Savaré, Regularity results for elliptic equations in Lipschitz domains, J. Func. Anal., 152 (1998) 176-201.

[20] J. Tomasi, B. Menucci and R. Cammi, Quantum mechanical continuum solvation models, Chem. Rev., 105 (2005) 2999-3093.

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

[22] D. Xie, New solution decomposition and minimization schemes for Poisson-Boltzmann equation in calculation of biomolecular electrostatics, J. Comput. Phys., 275 (2014) 294-309.

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

[24] Z. Yu, M. J. Holst, Y. Cheng and J. A. McCammon, Feature-preserving adaptive mesh generation for molecular modeling, J. Mol. Graph. Model., 26 (2008) 1370-1380.

[25] E. Zeidler, Nonlinear Functional Analysis and Its Applications III: Variational Methods and Optimization, Volume 3. Springer Verlag, 1985.

[26] S. Zhou, Z. Wang and B. Li, Mean-field description of ionic size effects with nonuniform ionic sizes: A numerical approach, Phys. Rev. E, 84 (2011) 021901.

[27] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn and M. Levitt, Finite-difference solution of the Poisson-Boltzmann equation: Complete elimination of self-energy, J. Comput. Chem., 17 (1996) 1344-1351.

School of Mathematics and Computing Science, Changsha University of Science and Technology, Changsha, Hunan, 410004, P.R. China
E-mail: lijiaoflying@163.com

Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201-0413, USA
E-mail: dxie@uwm.edu
URL: http://www.uwm.edu/~dxie