Research Article **Open Access**

Yi Jiang, Jinyong Ying, and Dexuan Xie*

A Poisson-Boltzmann Equation Test Model for Protein in Spherical Solute Region and its Applications

Abstract: The Poisson-Boltzmann equation (PBE) is one important implicit solvent continuum model for calculating electrostatics of protein in ionic solvent. Several numerical algorithms and program packages have been developed but verification and comparison between them remains an interesting topic. In this paper, a PBE test model is presented for a protein in a spherical solute region, along with its analytical solution. It is then used to verify a PBE finite element solver and applied to a numerical comparison study between a finite element solver and a finite difference solver. Such a study demonstrates the importance of retaining the interface conditions in the development of PBE solvers.

Keywords: Poisson-Boltzmann equation; continuum electrostatics; finite element method; finite difference method; electrostatic potential

MSC: 92-08, 92C40, 65N30, 65N06

DOI 10.2478/mlbmb-2014-0006
Received April 13, 2014; accepted August 16, 2014

1 Introduction

The Poisson-Boltzmann equation (PBE) is a widely-used implicit solvent continuum model for calculating protein electrostatics in ionic solvent [2, 9, 13, 15, 16, 21]. Several PBE numerical solvers and computer program packages have been developed, and applied to many biomolecular studies and simulations [3, 6, 10, 11, 17, 18], but verification and comparison among them is still an interesting research topic. So far, the Born ball model [4] was often employed to do such verification tests in the simplest case of a spherical solute region containing one central charge [8, 20]. The Kirkwood's dielectric sphere model [12] was applied to the case of a spherical solute region containing multiple point charges. However, its calculation is complex and produces truncation errors since its analytical solution is an infinite series in terms of Legendre's polynomials. Thus, it works only for a few point charges. Recently, a finite difference algorithm called the matched interface and boundary PBE solver (MIBPB) was proposed [7, 23]. With a special treatment of the general interface conditions, MIBPB works for the interface jumps and discontinuities too. To validate such a feature, several test models were constructed for a solute region containing a biomolecule and an irregular interface between the solute and solvent regions [7, 23]. However, since analytical solutions are jump discontinuous across the interface, they cannot be used to verify a PBE solver constructed from the continuous interface conditions. Hence, how to construct a PBE test model with the continuous interface conditions and a solute region containing a biomolecule remains an unsolved problem.

Yi Jiang: Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201-0413, USA
Jinyong Ying: Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201-0413, USA
***Corresponding Author: Dexuan Xie:** Department of Mathematical Sciences, University of Wisconsin-Milwaukee, Milwaukee, WI 53201-0413, USA, E-mial: dxie@uwm.edu

In fact, it is difficult to construct such a PBE test model even for a spherical solute region due to solution singularity. To overcome this difficulty, in this paper, we propose using solution decomposition techniques to construct PBE test models. In particular, following what was done in the construction of the Born ball PBE test model, we first construct a linear Poisson dielectric test model, and then modify it as a nonlinear PBE test model with a spherical solute region containing a protein. The key step is to split the solution u of the Poisson dielectric test model to a sum of two functions G and $\bar{u}$ with G being a known function (see (8)). Since G collects all the singularity points of u, we can simply select $\bar{u}$ as a twice continuously differentiable function within the solute and solvent regions, respectively (see (14)). In this way, the construction process is remarkably simplified.

In this paper, we construct a PBE test model (see (20)) and find its analytical solution in an algebraic expression for a protein hosted in a spherical solute region (see (21)). This PBE test model has the same PBE structure with one extra charge source term. Thus, it can be easily adapted for a PBE verification test. Another important feature of our PBE test model is that the solution range is allowed to be properly adjusted with a scaling parameter (see (22)). Due to this feature, our PBE test model works for any protein without causing any blow up problem in the calculation of the hyperbolic term $\sinh(u)$ of the PBE. In addition, similar to the PBE solution, its analytical solution has a singularity at each atomic position. Thus, different proteins may cause different levels of difficulties in its numerical solution. Hence, it may be valuable in a robustness comparison study of two different PBE solvers.

As an application example, in this paper, we used it to verify a PBE finite element program package we developed recently [20] using a protein with 488 atoms. In this study, we particularly constructed three nested quasi uniform tetrahedral meshes with 3,143, 25,131, and 200,009 vertices, respectively, and carried out numerical tests using the linear finite element method. The absolute and relative errors of the numerical solutions were calculated in the L_2 function norm. They were reduced almost quarterly as the mesh grid size was almost halved, which match well the finite element theory [5], and validate this PBE finite element program package.

As another application example, we did a comparison study between two PBE finite element and finite difference solvers. The main differences between the finite element and finite difference approaches lie on their different treatments of the Dirac delta point charge source terms and the interface conditions between the solute and solvent regions. Hence, for simplicity, it is sufficient for us to consider the Poisson dielectric test model for such a comparison study. In this study, we considered one commonly used finite difference method, which has been used in the popular PBE software program packages DELPHI [18], UHBD [6], APBS [1], and CHARMM [10, 11] for the calculation of solvation free energy, pK_a values, and electrostatic forces [2]. Such a finite difference method ignored the interface conditions, and used a uniform Cartesian grid with the interface between the solute and solvent regions being roughly approximated as a staircase line. Its numerical solution was reported to have a low accuracy for the Born and Kirkwood's dielectric sphere models [24]. However, no comparison test was done with the finite element method for a protein within the solute region. It is our new test model that makes such a test possible.

To do so, we programmed the finite element method for solving the Poisson dielectric test model and a tetrahedral mesh generator for a cubic domain Ω containing the unit spherical ball in Python, Fortran and C++ based on the finite element library `DOLFIN` from the `FEniCS` project [14] and the tetrahedral mesh generator `TetGen` (*http://wias-berlin.de/software/tetgen/*). We also programmed this PBE finite difference method (see [10, 17] for example) using a linear delta approximation function and the successive over-relaxation (SOR) method [22]. Numerical tests were done for four proteins with the number of atoms up to 4,173. Three different uniform meshes with grid sizes $h = 0.129, 0.0784, 0.0494$ were used for the finite difference tests. Three tetrahedral meshes were generated from our mesh generator for the finite element tests such that their numbers of vertices were close to that of the uniform meshes. Since each tetrahedral mesh was unstructured, its mesh size h was defined as the longest edge among all the tetrahedra. The mesh sizes of our three tetrahedral meshes were found to be 0.2851, 0.1656, 0.1137, respectively.

Numerical results show that the finite element solutions had a much higher accuracy than the finite difference solutions, and their accuracy was improved significantly as h was decreased. We also observed that the finite difference solutions had only a very limited accuracy, which could not be improved further through simply decreasing h (see Table 2). Similar numerical results were reported in [24, Tables 1 and 2] for the finite difference solutions generated from the program packages PBEQ and APBS in the case of solving the Born and Kirkwood's dielectric sphere models.

Furthermore, we repeated the finite difference tests using a cubic delta approximation function. It was found that the accuracy of the finite difference solution was close to that using the linear delta approximation function. This indicates that the finite difference errors are mainly related to the flux interface condition, which should be considered in order to improve the solution accuracy of a PBE finite difference solver. Such efforts were done in [7, 23, 24].

Finally, we repeated the numerical tests on one small mesh for the four protein test cases by using the quadratic finite element method. It was found that the accuracy of the finite element solution was sharply reduced. In this case, the quadratic finite element method involved about 260,000 mesh nodes, but its solution accuracy was close to the one generated by the linear finite element method on the mesh with 551,368 vertices. These tests showed a potential application of a higher order finite element method in solving PBE in order to avoid the difficulties of generating a large finite element mesh with a high quality.

The paper is organized as follows. In Section 2, we review the PBE model, the Poisson dielectric model and the Born ball PBE test model. In Section 3, we present the new Poisson dielectric and PBE test models. In Section 4, the new PBE test model is used to verify a PBE finite element program package. In Section 5, the new Poisson dielectric test model is applied to a comparison study, and finally, we make conclusions in Section 6.

2 PBE model and Born ball PBE test model

In this section, we introduce the PBE model and the Poisson dielectric model. We then show how the Born ball PBE test model is obtained. Let D_p be a bounded solute region surrounded by the solvent region D_s such that the whole space $\mathbb{R}^3$ satisfies the partition

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

where Γ denotes the interface between D_p and D_s. We assume that D_p hosts a biomolecule (e.g., a protein) consisting of n_p atoms, and is immersed in a symmetric 1:1 ionic solvent containing only sodium (N_a^+) and chloride (Cl^-) ions (a salt solution). In this case, the PBE model is defined by

$$\begin{cases} -\epsilon_p \Delta u = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j} & \text{in } D_p, \\ -\epsilon_s \Delta u + \kappa^2 \sinh(u) = 0 & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty, \end{cases} \tag{1}$$

where u is a dimensionless electrostatic potential function, ϵ_p and ϵ_s are two dielectric constants, α and κ^2 are two PBE constants, z_j and $\mathbf{r}_j$ are the charge number and position vector of the jth atom, respectively, and $\delta_{\mathbf{r}_j}$ is the Dirac delta distribution at $\mathbf{r}_j$. In SI units, α and κ^2 are given by

$$\alpha = \frac{e_c^2}{\epsilon_0 k_B T}, \quad \kappa^2 = 2I_s \frac{10^3 N_A e_c^2}{\epsilon_0 k_B T}, \tag{2}$$

where e_c is the elementary charge, ϵ_0 is the vacuum permittivity, I_s denotes the ionic strength, N_A is the Avogadro's number, k_B is the Boltzmann constant, and T is the absolute temperature.

Clearly, setting $\kappa = 0$ reduces the PBE model to the Poisson dielectric model

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j} & \text{in } D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) = 0 & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty. \end{cases} \tag{3}$$

For a spherical region D_p containing one central charge ze_c, the Poisson dielectric model becomes the Born ball model

$$\begin{cases} -\epsilon_p \Delta u = \alpha z \delta & \text{in } D_p, \\ -\epsilon_s \Delta u = 0 & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty, \end{cases} \tag{4}$$

where δ is the Dirac delta distribution at the origin, D_p, D_s and Γ are set as

$$D_p = \{\mathbf{r} \big| \, |\mathbf{r}| < a\}, \quad D_s = \{\mathbf{r} \big| \, |\mathbf{r}| > a\}, \quad \Gamma = \{\mathbf{r} \big| \, |\mathbf{r}| = a\},$$

with a given radius $a > 0$, and the analytical solution u can be found in the form

$$u(\mathbf{r}) = \begin{cases} \frac{\alpha z}{4\pi a}\left(\frac{1}{\epsilon_s} - \frac{1}{\epsilon_p}\right) + \frac{\alpha z}{4\pi \epsilon_p |\mathbf{r}|} & \text{in } D_p, \\ \frac{\alpha z}{4\pi \epsilon_s |\mathbf{r}|} & \text{in } D_s. \end{cases} \tag{5}$$

The Born ball PBE test model can then be constructed as follows:

$$\begin{cases} -\epsilon_p \Delta u = \alpha z \delta & \text{in } D_p, \\ -\epsilon_s \Delta u + \kappa^2 \sinh(u) = \rho_s(\mathbf{r}) & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty, \end{cases} \tag{6}$$

where $\rho_s(\mathbf{r})$ is defined by

$$\rho_s(\mathbf{r}) = \kappa^2 \sinh\left(\frac{\alpha z}{4\pi \epsilon_s |\mathbf{r}|}\right) \quad \text{for } \mathbf{r} \in D_s.$$

Obviously, when $\rho_s = 0$, the above test model is reduced to the Born ball PBE model. Hence, ρ_s can be regarded as an extra charge function added to the PBE model. It is clear that the analytical solution of the Born ball PBE test model (6) is given in (5). In calculation, u is treated as a unknown function while ρ_s is a given source function. Hence, the Born ball PBE test model is a nonlinear elliptic interface problem, which can be used to verify a PBE solver.

3 Our new PBE test model

In this section, we follow the construction of the Born ball PBE test model to construct our new PBE test model for a protein hosted in a spherical solute region D_p. The key step is to obtain a Poisson dielectric test model with a given analytical solution. To do so, we construct a Poisson dielectric test model in the form

$$\begin{cases} -\epsilon_p \Delta u(\mathbf{r}) = \alpha \sum_{j=1}^{n_p} z_j \delta_{\mathbf{r}_j} & \text{in } D_p, \\ -\epsilon_s \Delta u(\mathbf{r}) = f_s(\mathbf{r}) & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p \frac{\partial u(\mathbf{s}^-)}{\partial \mathbf{n}} = \epsilon_s \frac{\partial u(\mathbf{s}^+)}{\partial \mathbf{n}} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty, \end{cases} \tag{7}$$

where f_s is a function to be determined, which can be regarded as extra charges added to the Poisson dielectric model (3). We intend to find an analytical solution of the Poisson dielectric test model through properly selecting f_s.

It is difficult to search for a solution of (7) directly due to the solution singularities caused by the Dirac delta distributions $\{\delta_{\mathbf{r}_j}\}_{j=1}^{n_p}$. To avoid such a difficulty, we split the solution u of (7) to a sum of the functions G and $\bar{u}$,

$$u = \bar{u} + G, \tag{8}$$

where $\bar{u}$ is the solution of the elliptic interface problem

$$\begin{cases} -\epsilon_p \Delta \bar{u}(\mathbf{r}) = 0 & \text{in } D_p, \\ -\epsilon_s \Delta \bar{u}(\mathbf{r}) = f_s(\mathbf{r}) & \text{in } D_s, \\ \bar{u}(\mathbf{s}^-) = \bar{u}(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \bar{u}(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_s \dfrac{\partial \bar{u}(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_p)\dfrac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ \bar{u}(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty, \end{cases} \tag{9}$$

and G is given by

$$G(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} \frac{z_j}{|\mathbf{r} - \mathbf{r}_j|}, \tag{10}$$

which collects all the singular points of the solution u. Here $\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})}$ can be found in the expression

$$\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} = -\frac{\alpha}{4\pi\epsilon_p} \sum_{j=1}^{n_p} z_j \frac{(\mathbf{s} - \mathbf{r}_j)\cdot \mathbf{n}}{|\mathbf{s} - \mathbf{r}_j|^3}. \tag{11}$$

Thus, the problem becomes how to construct a solution $\bar{u}$ of the elliptic interface problem (9). Note that $\bar{u}$ is twice continuously differentiable within D_p and D_s, respectively, provided that f_s is a continuous function. Hence, we can construct $\bar{u}$ using the following expression

$$\bar{u}(\mathbf{r}) = \begin{cases} 0 & \text{for } \mathbf{r} \in D_p, \\ c(\mathbf{r}) \sin\left(\dfrac{|\mathbf{r}|^2}{a^2} - 1\right) & \text{for } \mathbf{r} \in D_s, \end{cases} \tag{12}$$

where $c(\mathbf{r})$ is a twice continuously differentiable function to be determined to satisfy the elliptic interface problem (9).

By expression (12), it is easy to verify that $\bar{u}$ satisfies

$$\bar{u}(\mathbf{s}^-) = \bar{u}(\mathbf{s}^+) = 0, \quad \epsilon_p \frac{\partial \bar{u}(\mathbf{s}^-)}{\partial \mathbf{n}(\mathbf{s})} = 0 \qquad \forall\, \mathbf{s} \in \Gamma.$$

We then use the facts $\mathbf{s}\cdot\mathbf{s} = |\mathbf{s}|^2 = a^2$ and $\mathbf{n}(\mathbf{s}) = \frac{\mathbf{s}}{a}$ to find that

$$\epsilon_s \frac{\partial \bar{u}(\mathbf{s}^+)}{\partial \mathbf{n}(\mathbf{s})} = \epsilon_s \nabla \bar{u}(\mathbf{s}^+)\cdot \mathbf{n}(\mathbf{s}) = \frac{2}{a^2}\epsilon_s c(\mathbf{s})\mathbf{s}\cdot\frac{\mathbf{s}}{a} = \frac{2}{a}\epsilon_s c(\mathbf{s}).$$

Thus, the second interface condition of (9) gives the equation of c:

$$0 = \frac{2}{a}\epsilon_s c(\mathbf{s}) + (\epsilon_s - \epsilon_p)\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} \qquad \text{on } \Gamma,$$

from which we obtain the expression of c on the interface Γ:

$$c(\mathbf{s}) = \frac{a(\epsilon_p - \epsilon_s)}{2\epsilon_s}\frac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} = \frac{\alpha(\epsilon_s - \epsilon_p)}{8\pi\epsilon_p\epsilon_s}\sum_{j=1}^{n_p}\frac{z_j(\mathbf{s} - \mathbf{r}_j)\cdot\mathbf{s}}{|\mathbf{s} - \mathbf{r}_j|^3} \qquad \forall \mathbf{s} \in \Gamma.$$

Hence, we can set $c(\mathbf{r})$ in the expression

$$c(\mathbf{r}) = \frac{\alpha(\epsilon_s - \epsilon_p)}{8\pi\epsilon_p\epsilon_s}\sum_{j=1}^{n_p}\frac{z_j(\mathbf{r} - \mathbf{r}_j)\cdot\mathbf{r}}{|\mathbf{r} - \mathbf{r}_j|^3} \quad \forall \mathbf{r} \in D_s. \tag{13}$$

Applying the above expression of c to (12) yields the expression of $\bar{u}$:

$$\bar{u}(\mathbf{r}) = \begin{cases} 0 & \text{for } \mathbf{r} \in D_p, \\ \frac{\alpha(\epsilon_s - \epsilon_p)}{8\pi\epsilon_p\epsilon_s} \sin\left(\frac{|\mathbf{r}|^2}{a^2} - 1\right) \sum_{j=1}^{n_p} \frac{z_j(\mathbf{r}-\mathbf{r}_j)\cdot\mathbf{r}}{|\mathbf{r}-\mathbf{r}_j|^3} & \text{for } \mathbf{r} \in D_s. \end{cases} \tag{14}$$

It is clear that the above $\bar{u}$ satisfies the first equation and the two interface conditions of (9), and approaches zero as $|\mathbf{r}| \to \infty$. It also satisfies the second equation of (9) by setting the function f_s in the form

$$f_s(\mathbf{r}) = -\epsilon_s \Delta\bar{u}(\mathbf{r}).$$

This provs that the function $\bar{u}$ of (14) is a solution of the elliptic interface problem (9) with the above selection of f_s.

We next need to calculate $\Delta\bar{u}(\mathbf{r})$ to get the expression of f_s. A direct calculation of $\Delta\bar{u}(\mathbf{r})$ is intricate due to the complicated expression (14) of $\bar{u}(\mathbf{r})$. To avoid this difficulty, we start with (12) to get an expression of f_s in terms of c:

$$f_s(\mathbf{r}) = -\epsilon_s\left[\sin\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\Delta c(\mathbf{r}) + \frac{4}{a^2}\cos\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\nabla c(\mathbf{r})\cdot\mathbf{r} + \frac{6c(\mathbf{r})}{a^2}\cos\left(\frac{|\mathbf{r}|^2}{a^2}-1\right) - \frac{4|\mathbf{r}|^2 c(\mathbf{r})}{a^4}\sin\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\right]. \tag{15}$$

We then use the fact that $\Delta G = 0$ in D_s to obtain that

$$\Delta c(\mathbf{r}) = 0 \quad \forall \mathbf{r} \in D_s. \tag{16}$$

Furthermore, we find that

$$\nabla c(\mathbf{r})\cdot\mathbf{r} = \frac{\alpha(\epsilon_s-\epsilon_p)}{8\pi\epsilon_p\epsilon_s}\sum_{j=1}^{n_p} z_j\left(\frac{(2\mathbf{r}-\mathbf{r}_j)\cdot\mathbf{r}}{|\mathbf{r}-\mathbf{r}_j|^3} - \frac{3[(\mathbf{r}-\mathbf{r}_j)\cdot\mathbf{r}]^2}{|\mathbf{r}-\mathbf{r}_j|^5}\right). \tag{17}$$

Applying (16) and (17) to (15) gives the expression of f_s:

$$\begin{aligned} f_s(\mathbf{r}) = &\frac{\alpha(\epsilon_p-\epsilon_s)}{4\pi a^2\epsilon_p}\sum_{j=1}^{n_p} z_j\left[\cos\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\frac{7|\mathbf{r}|^2 - 5\mathbf{r}\cdot\mathbf{r}_j}{|\mathbf{r}-\mathbf{r}_j|^3} - \sin\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\frac{2|\mathbf{r}|^2(|\mathbf{r}|^2-\mathbf{r}\cdot\mathbf{r}_j)}{a^2|\mathbf{r}-\mathbf{r}_j|^3}\right. \\ &\left. -6\cos\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\frac{(|\mathbf{r}|^2-\mathbf{r}\cdot\mathbf{r}_j)^2}{|\mathbf{r}-\mathbf{r}_j|^5}\right] \qquad \forall \mathbf{r} \in D_s. \end{aligned} \tag{18}$$

We are now in the position to construct our new PBE test model. With f_s being given in (18) and $U(\mathbf{r})$ by

$$U(\mathbf{r}) = \frac{\alpha}{4\pi\epsilon_p}\sum_{j=1}^{n_p}\frac{z_j}{|\mathbf{r}-\mathbf{r}_j|} + \frac{\alpha(\epsilon_s-\epsilon_p)}{8\pi\epsilon_p\epsilon_s}\sin\left(\frac{|\mathbf{r}|^2}{a^2}-1\right)\sum_{j=1}^{n_p}\frac{z_j(\mathbf{r}-\mathbf{r}_j)\cdot\mathbf{r}}{|\mathbf{r}-\mathbf{r}_j|^3}, \tag{19}$$

we construct our new PBE test model as follows:

$$\begin{cases} -\epsilon_p\Delta u(\mathbf{r}) = \alpha\sum_{j=1}^{n_p} z_j\delta_{\mathbf{r}_j} & \text{in } D_p, \\ -\epsilon_s\Delta u(\mathbf{r}) + \kappa^2\sinh(u(\mathbf{r})) = f_s(\mathbf{r}) + \kappa^2\sinh(U(\mathbf{r})) & \text{in } D_s, \\ u(\mathbf{s}^-) = u(\mathbf{s}^+), \quad \epsilon_p\frac{\partial u(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_s\frac{\partial u(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ u(\mathbf{r}) \to 0 & \text{as } |\mathbf{r}| \to \infty. \end{cases} \tag{20}$$

Clearly, the analytical solution of our new PBE test model is the same as that of the Poisson dielectric test model (7). It is given in the expression

$$u(\mathbf{r}) = \begin{cases} \frac{\alpha}{4\pi\epsilon_p}\sum_{j=1}^{n_p}\frac{z_j}{|\mathbf{r}-\mathbf{r}_j|} & \text{in } D_p, \\ U(\mathbf{r}) & \text{in } D_s. \end{cases} \tag{21}$$

We observe that the solution u of our PBE test model (20) can be expressed as

$$u(\mathbf{r}) = \alpha\hat{u}(\mathbf{r}) \qquad \mathbf{r} \in \Omega, \tag{22}$$

where $\hat{u}$ denotes the solution of the nonlinear problem (20) using $\alpha = 1$. In this relationship, the constant α can be treated as a scaling parameter, with which we can properly adjust the solution range of our new PBE test model to avoid a potential blow up problem in the calculation of the hyperbolic term $\sinh(u)$.

In fact, the two extra charge terms f_s and $\kappa^2 \sinh(U(\mathbf{r}))$ may cause (20) to have a much larger solution range than the corresponding PBE model. Because of these two extra charge terms, the two PBE constants α and κ lost their original physical meanings. With a proper selection of α, our new PBE test model can work stably in its numerical implementation for any given protein.

Our new PBE test model (20) can be conveniently applied to the verification of a PBE numerical solver or a PBE program package, since in numerical tests, we only need to modify the parts of the solver or package that are related to the extra charge terms f_s and $\kappa^2 \sinh(U(\mathbf{r}))$.

4 Verification of a PBE finite element program package

In this section we use the PBE test model (20) to verify a PBE finite element program package developed in [20]. In the tests, the unbounded solvent region D_s was truncated as

$$D_s = \{\mathbf{r} | a < |\mathbf{r}| < A\},$$

to modify (20) as a boundary value problem with the Dirichlet boundary condition

$$u(\mathbf{s}) = U(\mathbf{s}) \qquad \text{on } \partial\Omega,$$

where a and A are two given positive numbers, $\partial\Omega$ denotes the boundary of the bounded domain $\Omega = \{\mathbf{r}|\ |\mathbf{r}| < A\}$, and U is the analytical solution of (20), which has been given in (19). By the PBE finite element algorithm [20], the solution u of this PBE test model using the above Dirichlet boundary condition on Ω was split as

$$u(\mathbf{r}) = G(\mathbf{r}) + \Psi(\mathbf{r}) + \tilde{\Phi}(\mathbf{r}) \qquad \forall\mathbf{r} \in \Omega, \tag{23}$$

where G is defined by (10), Ψ is the solution of the linear interface problem

$$\begin{cases} \Delta\Psi(\mathbf{r}) = 0 & \text{in } D_p \cup D_s \\ \Psi(\mathbf{s}^-) = \Psi(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial\Psi(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_s \dfrac{\partial\Psi(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} + (\epsilon_s - \epsilon_p)\dfrac{\partial G(\mathbf{s})}{\partial\mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ \Psi(\mathbf{s}) = U(\mathbf{s}) - G(\mathbf{s}) & \text{on } \partial\Omega, \end{cases} \tag{24}$$

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

$$\begin{cases} \Delta\tilde{\Phi}(\mathbf{r}) = 0 & \text{in } D_p, \\ -\epsilon_s\Delta\tilde{\Phi}(\mathbf{r}) + \kappa^2 \sinh(W(\mathbf{r}) + \tilde{\Phi}(\mathbf{r})) = \rho_s(\mathbf{r}) & \text{in } D_s, \\ \tilde{\Phi}(\mathbf{s}^-) = \tilde{\Phi}(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial\tilde{\Phi}(\mathbf{s}^-)}{\partial\mathbf{n}(\mathbf{s})} = \epsilon_s \dfrac{\partial\tilde{\Phi}(\mathbf{s}^+)}{\partial\mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ \tilde{\Phi}(\mathbf{s}) = 0 & \text{on } \partial\Omega. \end{cases} \tag{25}$$

Here $W = G + \Psi$, Ψ has been computed before solving (25), and ρ_s is defined by

$$\rho_s(\mathbf{r}) = f_s(\mathbf{r}) + \kappa^2 \sinh(U(\mathbf{r})).$$

To solve the above equations of (24) and (25) by the finite element method, the equations are reformulated as variational problems so that their interface conditions can be naturally treated. Furthermore, a modified Newton minimization scheme is developed to efficiently solve the nonlinear variational problem of $\tilde{\Phi}$ as a

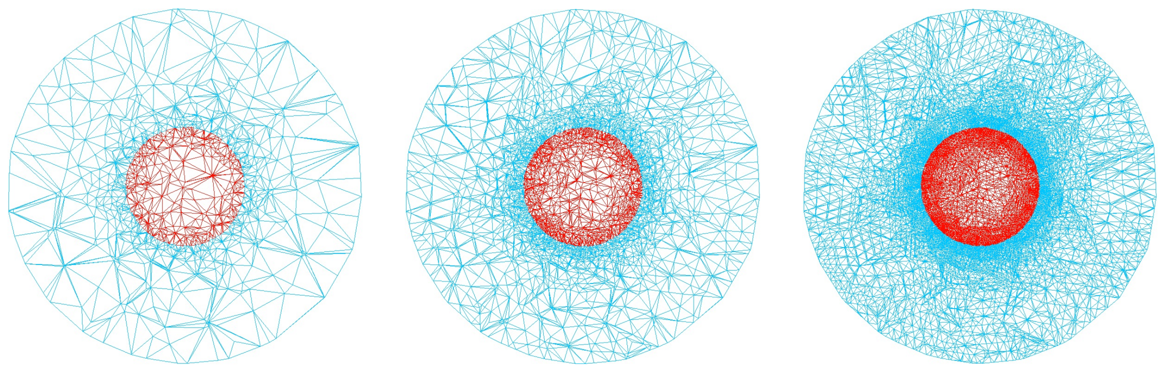

(a) 3,143 vertices **(b)** 25,131 vertices **(c)** 200,009 vertices

Figure 1: Cross-section views of the three nested tetrahedral meshes used for verification of the PBE finite element program package [20]. Here the meshes of solute region D_p are coloured in red.

variational minimization problem. All the related linear variational problems are solved efficiently by the preconditioned conjugate gradient (PCG) method with incomplete LU preconditioning. See [20] for the details.

We wrote a program for computing the extra charge term ρ_s and exact solution u. A few modifications were then made to the PBE finite element program package [20], making it work for numerically testing our PBE test model. We also wrote a tetrahedral mesh generator based on the tetrahedral mesh generator `TetGen` (*http://wias-berlin.de/software/tetgen/*) and the mesh function `Sphere()` from the `FEniCS` project [14]. In the numerical tests, we used

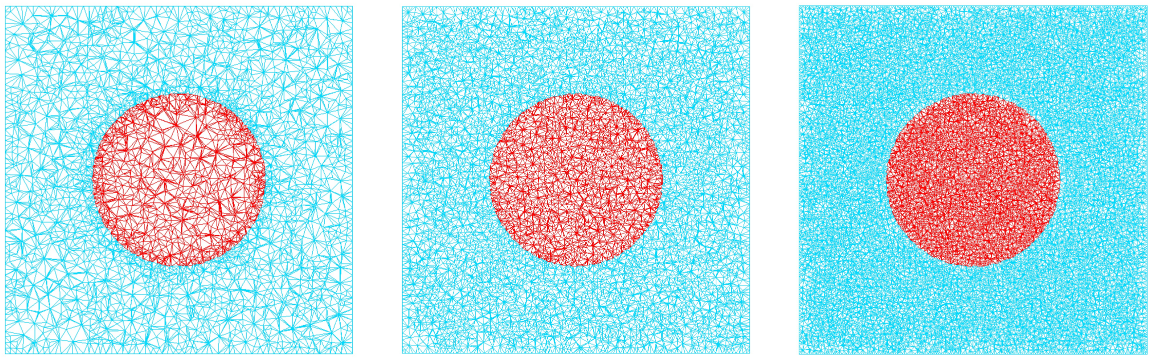
$$a = 17, \quad A = 51, \quad \epsilon_p = 2.0, \quad \epsilon_s = 78.54, \quad \alpha = 1, \quad \kappa^2 = 0.8482715835384875,$$

and a protein with 488 atoms (PDB ID 2LZX), which was downloaded from the Protein Data Bank (*http://www.rcsb.org*). Here the value of κ^2 was produced by using (2) with $T = 298.15$ and $I_s = 0.1$. As required by the PBE finite element program package, we converted the PDB file of 2LZX to a PQR file using the software tool PDB2PQR (*http://www.poissonboltzmann.org/pdb2pqr*). Our tetrahedral mesh generator was used to generate three nested tetrahedral meshes with 3,143, 25,131, and 200,009 vertices, respectively, whose mesh sizes were halved. One cross-section view of each mesh was displayed in Figure 1 to demonstrate these three meshes. Three finite element solutions were then calculated by the linear finite element method. Their relative and absolute errors were reported in Table 1.

Table 1: The absolute and relative errors of the finite element solutions u_h of the PBE test model (20) for the Protein with PDB ID 2LZX and 488 atoms on three different tetrahedral meshes. Here u is the analytical solution given in (21), and u_h denotes a numerical solution of the linear finite element method.

Mesh Data			Absolute Error	Relative Error
Mesh	# Vertices	# Tetrahedra	$\sqrt{\int_\Omega \lvert u_h - u\rvert^2 d\mathbf{r}}$	$\sqrt{\frac{\int_\Omega \lvert u_h-u\rvert^2 d\mathbf{r}}{\int_\Omega \lvert u\rvert^2 d\mathbf{r}}}$
Mesh 1	3,143	18,591	1.96133	3.65904×10^{-1}
Mesh 2	25,131	148,728	0.477399	9.23066×10^{-2}
Mesh 3	200,009	1,189,824	0.150239	2.91177×10^{-2}

From Table 1 it can be seen that the absolute and relative errors were reduced almost quarterly when the mesh sizes were almost reduced by half. These results reflect the convergence properties of the linear finite element method [5]. Hence, they validate the PBE finite element program package.



(a) 32,639 vertices **(b)** 139,967 vertices **(c)** 554,745 vertices

Figure 2: Cross-section views of the three tetrahedral meshes used in comparison tests between the finite element and finite difference methods. Here the solute region D_p is marked in red.

5 Finite Difference Method via Finite Element Method

As another application, in this section, we use the Poisson dielectric test model (7) to do a comparison study between the finite element and finite difference methods. For the sake of simplifying a construction of the finite difference method, we selected a cubic domain $\Omega = [-2, 2]\times[-2, 2]\times[-2, 2]$ to modify (7) as a boundary value problem with the Dirichlet boundary condition

$$u(\mathbf{s}) = U(\mathbf{s}) \qquad \text{on } \partial\Omega,$$

where U has been given in (19). Because of (8), we only need to solve the following boundary value problem

$$\begin{cases} -\epsilon_p \Delta \bar{u}(\mathbf{r}) = 0 & \text{in } D_p, \\ -\epsilon_s \Delta \bar{u}(\mathbf{r}) = f_s(\mathbf{r}) & \text{in } D_s, \\ \bar{u}(\mathbf{s}^-) = \bar{u}(\mathbf{s}^+), \quad \epsilon_p \dfrac{\partial \bar{u}(\mathbf{s}^-)}{\partial \mathbf{n}} = \epsilon_s \dfrac{\partial \bar{u}(\mathbf{s}^+)}{\partial \mathbf{n}} + (\epsilon_s - \epsilon_p)\dfrac{\partial G(\mathbf{s})}{\partial \mathbf{n}(\mathbf{s})} & \text{on } \Gamma, \\ \bar{u}(\mathbf{s}) = V(\mathbf{s}) & \text{on } \partial\Omega, \end{cases} \tag{26}$$

where V denotes the analytical solution of (9), which has been given in (14).

We wrote a finite element program for solving the test problem (26) based on the finite element library DOLFIN from the FEniCS project [14]. Here our mesh generation program was used to generate tetrahedral meshes for numerical tests.

We also wrote a finite difference program for solving the Poisson dielectric test model (7) following the PBE finite difference scheme used in the PBE finite difference program packages DELPHI [17, 18] and CHARMM [10, 11]. That is, the flux interface condition was ignored to simply approximate the Poisson dielectric test model (7) as a system of second order central finite difference equations as follows:

$$\begin{aligned} &\epsilon_{i+\frac{1}{2},j,k} u_{i+1,j,k} + \epsilon_{i-\frac{1}{2},j,k} u_{i-1,j,k} + \epsilon_{i,j+\frac{1}{2},k} u_{i,j+1,k} + \epsilon_{i,j-\frac{1}{2},k} u_{i,j-1,k} + \epsilon_{i,j,k+\frac{1}{2}} u_{i,j,k+1} + \epsilon_{i,j,k-\frac{1}{2}} u_{i,j,k-1} \\ &- \left(\epsilon_{i-\frac{1}{2},j,k} + \epsilon_{i+\frac{1}{2},j,k} + \epsilon_{i,j-\frac{1}{2},k} + \epsilon_{i,j+\frac{1}{2},k} + \epsilon_{i,j,k-\frac{1}{2}} + \epsilon_{i,j,k+\frac{1}{2}}\right) u_{i,j,k} = h^2 f_{i,j,k}, \end{aligned} \tag{27}$$

where $i, j, k = 0, 1, 2, \ldots, N$, $h = 4/N$, $u_{i,j,k}$ denotes a numerical value of the solution $u(x, y, z)$ at the grid node (x_i, y_j, z_k) with $x_i = -2 + ih$, $y_j = -2 + jh$, and $z_k = -2 + kh$, $\epsilon_{i+\frac{1}{2},j,k} = \epsilon(-2 + (i + \frac{1}{2})h, -2 + jh, -2 + kh)$ for the dielectric function $\epsilon(\mathbf{r})$ with $\mathbf{r} = (x, y, z) \in \Omega$, the other discrete values of ϵ are defined similarly, and $f_{i,j,k}$ denotes the value of the function $f(\mathbf{r})$ at the grid node (x_i, y_j, z_k). Here $\epsilon(\mathbf{r})$ and $f(\mathbf{r})$ are defined by

$$\epsilon(\mathbf{r}) = \begin{cases} \epsilon_p & \forall \mathbf{r} \in D_p, \\ \epsilon_s & \forall \mathbf{r} \in D_s, \end{cases} \quad \text{and} \quad f(\mathbf{r}) = \begin{cases} \alpha \displaystyle\sum_{j=1}^{n_p} z_j \delta_h(\mathbf{r} - \mathbf{r}_j) & \forall \mathbf{r} \in D_p, \\ f_s(\mathbf{r}) & \forall \mathbf{r} \in D_s, \end{cases}$$

where $\delta_h(\mathbf{r}-\mathbf{r}_j)$ denotes an approximation of the Dirac delta distribution $\delta_{\mathbf{r}_j}$. By an approximate delta function ϕ [19, Table 1, Page 512], δ_h is set as

$$\delta_h(\mathbf{r}) = \frac{1}{h^3}\phi(x/h)\phi(y/h)\phi(z/h). \tag{28}$$

In numerical tests, we used the linear approximate delta function

$$\phi(x) = \begin{cases} 1 - |x| & 0 \le |x| < 1, \\ 0 & |x| > 1. \end{cases} \tag{29}$$

and the cubic approximate delta function

$$\phi(x) = \begin{cases} 1 - \frac{1}{2}|x| - |x|^2 + \frac{1}{2}|x|^3 & 0 \le |x| < 1, \\ 1 - \frac{11}{6}|x| + |x|^2 - \frac{1}{6}|x|^3 & 1 \le |x| < 2, \\ 0 & 2 \le |x|. \end{cases} \tag{30}$$

We solved the finite difference system (27) by the SOR method [17, 22] using the Dirichlet boundary condition

$$u(\mathbf{s}) = U(\mathbf{s}) \qquad \mathbf{s} \in \partial\Omega, \tag{31}$$

where U is given in (19). In the numerical tests, we set $N = 31, 51$, and 81, which gave the grid size $h = 0.1290, 0.0784$, and 0.0494, respectively.

Four proteins represented in PDB IDs 2LZX, 1UCS, 1AQ5, and 1HB8 were downloaded from the Protein Data Bank for our numerical tests. Their 488, 997, 2292, and 4173 fixed point charges were relocated into the unit ball region D_p through scaling their atomic positions, respectively. Three tetrahedral meshes were generated from our mesh generation program for the finite element numerical tests. Each of them had almost the same number of vertices as the number of mesh nodes of the corresponding finite difference uniform mesh. Since it was unstructured, its grid size h (defined as the longest edge of all the tetrahedra) was found to be 0.2851, 0.1656, and 0.1137, respectively, which was much larger than the corresponding mesh size of the finite difference method. One cross-section view was displayed in Figure 2 to demonstrate these three meshes.

To compare the solution accuracy, we calculated the relative error E_{re} according to the formula

$$E_{re} = \sqrt{\left(\sum_{i=1}^{N} |u(\mathbf{r}^i) - u_h(\mathbf{r}^i)|^2\right) / \sum_{i=1}^{N} |u(\mathbf{r}^i)|^2}, \tag{32}$$

where u and u_h denote the analytic and numerical solutions, respectively, and $\mathbf{r}^i$ denotes the ith grid node. Numerical results are reported in the Table 2.

From Table 2 it can be seen that the finite element method was much more accurate than the finite difference method defined in (27). In particular, such a finite difference method only had a low accuracy due to its ignoring the flux interface condition, whose relative errors might be increased even if the mesh size h was decreased from 0.129 to 0.0494. Similar numerical results were reported in [24, Tables 1 and 2] for the finite difference solutions generated from the program packages PBEQ and APBS in the case of solving the Born and Kirkwood's dielectric sphere models. Correspondingly, the finite element method was convergent, as its relative errors were found to be reduced as h was decreased in all the numerical tests.

We also found from Table 2 that the relative errors of the finite difference method of (27) might not be reduced when the linear approximate delta function (29) was replaced by a more accurate cubic approximate delta function (30). This implies that the ignorance of the flux interface condition was one major factor that affects the accuracy of the finite difference method. Hence, to improve the accuracy of a finite difference method, it is essential to consider the flux interface condition. Such efforts were done in [7, 23, 24].

From Table 2 we further noted that even with the grid size $h = 0.2851$, the finite element method still produced a much smaller relative error than the finite difference method using the grid size $h = 0.0494$. This suggests that the finite element method may take less CPU time than the finite difference method of (27) to generate a numerical solution with the same accuracy requirement.

Table 2: Comparison of the solution accuracy of the finite difference method with that of the linear finite element method for solving the Poisson dielectric test model (7). Here, $E_{re,1}$ and $E_{re,2}$ denote the relative errors of the finite difference method using the linear and cubic approximate delta functions (29) and (30), respectively, the mesh grid size h of the finite element method is defined as the longest edge among all the tetrahedra of a mesh, and the relative error E_{re} is defined in (32).

Protein	Finite Difference Method of (27)				Finite Element Method		
(#Atoms)	h	#Vertices	$E_{re,1}$	$E_{re,2}$	h	#Vertices	E_{re}
	0.1290	32,768	5.27×10^{-2}	5.50×10^{-2}	0.2851	32,639	1.47×10^{-2}
2LZX	0.0784	140,608	5.31×10^{-2}	5.21×10^{-2}	0.1656	139,967	9.70×10^{-3}
(488)	0.0494	551,368	4.67×10^{-2}	4.54×10^{-2}	0.1137	554,745	7.10×10^{-3}
	0.1290	32,768	3.71×10^{-1}	3.82×10^{-1}	0.2851	32,639	6.22×10^{-2}
1UCS	0.0784	140,608	5.42×10^{-1}	5.53×10^{-1}	0.1656	139,967	4.13×10^{-2}
(997)	0.0494	551,368	5.38×10^{-1}	5.55×10^{-1}	0.1137	554,745	2.25×10^{-2}
	0.1290	32,768	1.06×10^{-1}	1.05×10^{-1}	0.2851	32,639	3.49×10^{-2}
1AQ5	0.0784	140,608	9.00×10^{-2}	9.27×10^{-2}	0.1656	139,967	2.24×10^{-2}
(2292)	0.0494	551,368	1.06×10^{-1}	1.09×10^{-1}	0.1137	554,745	1.45×10^{-2}
	0.1290	32,768	1.16×10^{-1}	1.28×10^{-1}	0.2851	32,639	3.47×10^{-2}
1HB8	0.0784	140,608	1.70×10^{-1}	1.62×10^{-1}	0.1656	139,967	2.11×10^{-2}
(4173)	0.0494	551,368	2.93×10^{-1}	2.93×10^{-1}	0.1137	554,745	1.29×10^{-2}

Finally, we repeated the numerical tests on the mesh with 32,639 vertices by the quadratic finite element method. In this case, the number of mesh nodes was increased to 259,998, but the relative errors of the finite element solutions for the four protein cases were sharply reduced to 0.0047, 0.0248, 0.0135, and 0.0133, respectively, which were close to the results generated by the mesh with 551,368 vertices. Currently, generating a large finite element mesh costs much more CPU time than solving a finite element equation. A higher order finite element method may make it possible to generate a highly accurate PBE numerical solution on a small mesh while reducing the total CPU time.

6 Conclusions

In this paper, we have presented a PBE test model and its analytical solution construction process as well as its applications. This model retains the PBE structure except for one extra charge source term, and its analytical solution has a concise algebraic expression. Thus, it can be easily adapted to a PBE program package for verification tests. Like the PBE solution, the analytical solution of our PBE test model satisfies the continuous interface conditions and has a singularity at each atomic position. Different proteins may cause different levels of difficulties in the numerical solution of our PBE test model. Because of this feature, our PBE test model is valuable not only in a verification test of a PBE solver/program package but also in a robustness comparison study of two different PBE solvers. Furthermore, we have shown in this paper that the solution range of our PBE test model can be adjusted simply with a scaling parameter. Hence, our PBE test model can work stably for any protein without causing any blow up problem in its computer implementation.

To demonstrate the application of our PBE and Poisson dielectric test models, we have reported the numerical results made from verification tests on one PBE finite element program package that we developed recently, and from comparison tests between a finite element solver and a finite difference solver that ignores the flux continuous interface condition. To carry out these numerical tests, we wrote a tetrahedral mesh generation program, a program for a finite element solver, and a program for this finite difference solver.

Currently, a PBE solver/program package is primarily verified by the simple Born and Kirkwood's dielectric sphere models. Several test models that worked for protein were constructed for validating the MIBPB algorithms but are suitable only for interface jumps and discontinuities. Since the interface conditions of PBE are mostly continuous, it is important to have a PBE test model that works for protein while preserving the continuous interface conditions. Our PBE test model is the first of such models. We expect it to be particularly valuable in the numerical study of PBE numerical algorithms and program packages.

Acknowledgement: This work was partially supported by the National Science Foundation, USA, through grant DMS-1226259 and the UWM Research Growth Initiative.

References

[1] N. A. Baker, D. Sept, S. Joseph, M. Holst, and J. A. McCammon, *Electrostatics of nanosystems: Application to microtubules and the ribosome*, Proc. Natl. Acad. Sci. USA **98** (2001), no. 18, 10037–10041.

[2] N.A. Baker, *Improving implicit solvent simulations: a Poisson-centric view*, Curr. Opin. Struc. Biol. **15** (2005), 137–143.

[3] N.A. Baker, D. Sept, M. Holst, and J. A. McCammon, *The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers*, IBM Journal of Research and Development **45** (2001), 427–438.

[4] M.Z. Born, *Volumen und hydratationswärme der ionen*, Zeitschrift für Physik A Hadrons and Nuclei **1** (1920), no. 1, 45–48.

[5] S.C. Brenner and L.R. Scott, *The mathematical theory of finite element methods*, third ed., Springer-Verlag, New York, 2008.

[6] M.E. Davis, J.D. Madura, B.A. Luty, and J.A. McCammon, *Electrostatics and diffusion of molecules in solution: Simulations with the University of Houston Browian dynamics program*, Comp. Phys. Comm. **62** (1991), 187–197.

[7] W. Geng, S. Yu, and G. Wei, *Treatment of charge singularities in implicit solvent models*, The Journal of chemical physics **127** (2007), 114106.

[8] M. Holst, J.A. McCammon, Z. Yu, Y. Zhou, and Y. Zhu, *Adaptive finite element modeling techniques for the Poisson-Boltzmann equation*, Communications in computational physics **11** (2012), no. 1, 179.

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

[10] W. Im, D. Beglov, and B. Roux, *Continuum solvation model: electrostatic forces from numerical solutions to the Poisson-Bolztmann equation*, Comput. Phys. Comm. **111** (1998), 59–75.

[11] S. Jo, M. Vargyas, J. Vasko-Szedlar, B. Roux, and W. Im, *PBEQ-Solver for online visualization of electrostatic potential of biomolecules*, Nucleic Acids Research **36** (2008), W271–W275.

[12] J.G. Kirkwood, *Theory of solutions of molecules containing widely separated charges with special application to zwitterions*, The Journal of Chemical Physics **2** (1934), 351.

[13] P. Koehl, *Electrostatics calculations: Latest methodological advances*, Current Opinion in Structural Biology **16** (2006), no. 2, 142–151.

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

[15] B. Lu, Y. Zhou, M.J. Holst, and J.A. McCammon, *Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications*, Commun. Comput. Phys. **3** (2008), no. 5, 973–1009.

[16] M.T. Neves-Petersen and S.B. Petersen, *Protein electrostatics: A review of the equations and methods used to model electrostatic equations in biomolecules – Applications in biotechnology*, Biotech. Annu. Rev. **9** (2003), 315–395.

[17] A. Nicholls and B. Honig, *A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson-Boltzmann equation*, J. Comp. Chem. **12** (1991), 435–445.

[18] W. Rocchia, E. Alexov, and B. Honig, *Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions*, J. Phys. Chem. B **105** (2001), 6507–6514.

[19] J. Waldén, *On the approximation of singular source terms in differential equations*, Numerical Methods for Partial Differential Equations **15** (1999), no. 4, 503–520.

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

[21] 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.

[22] D. M. Young, *Iterative solution of large linear system*, Academic press, New York, 1971.

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

[24] Y. Zhou, M. Feig, and G. Wei, *Highly accurate biomolecular electrostatics in continuum dielectric environments*, Journal of Computational Chemistry **29** (2008), no. 1, 87–97.