

UC Irvine

UC Irvine Previously Published Works

Title

Improved Poisson-Boltzmann Methods for High-Performance Computing.

Permalink

<https://escholarship.org/uc/item/8p63h7rb>

Journal

Journal of Chemical Theory and Computation, 15(11)

ISSN

1549-9618

Authors

Wei, Haixin
Luo, Aaron
Qiu, Tianyin
et al.

Publication Date

2019-11-12

DOI

10.1021/acs.jctc.9b00602

Peer reviewed



HHS Public Access

Author manuscript

J Chem Theory Comput. Author manuscript; available in PMC 2020 May 01.

Published in final edited form as:

J Chem Theory Comput. 2019 November 12; 15(11): 6190–6202. doi:10.1021/acs.jctc.9b00602.

Improved Poisson-Boltzmann Methods for High-Performance Computing

Haixin Wei¹, Aaron Luo², Tianyin Qiu², Ray Luo^{1,2,3,*}, Ruxi Qi^{2,*}

¹Department of Materials Science and Engineering, University of California, Irvine, CA 92697, USA

²Department of Molecular Biology and Biochemistry, University of California, Irvine, CA 92697, USA

³Chemical and Biomolecular Engineering and Biomedical Engineering, University of California, Irvine, CA 92697, USA

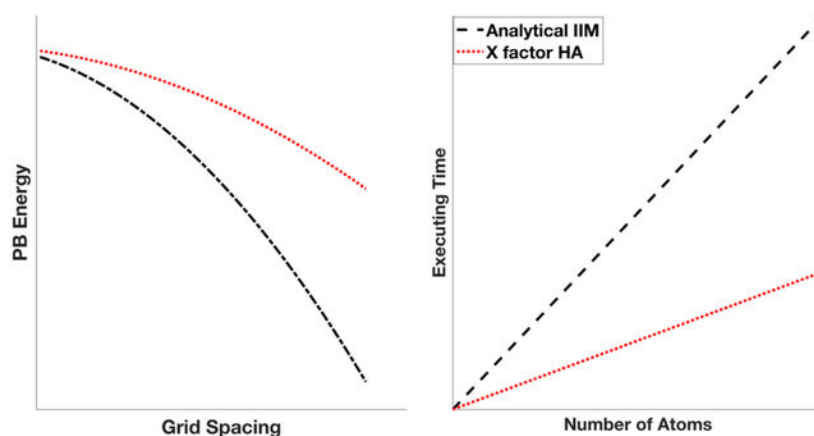
Abstract

Implicit solvent models based on the Poisson-Boltzmann equation (PBE) have been widely used to study electrostatic interactions in biophysical processes. These models often treat the solvent and solute regions as high and low dielectric continua, leading to a large jump in dielectrics across the molecular surface which is difficult to handle. Higher order interface schemes are often needed to seek higher accuracy for PBE applications. However, these methods are usually very liberal in the use of grid points nearby the molecular surface, making them difficult to use on high-performance computing platforms. Alternatively, the harmonic average (HA) method has been used to approximate dielectric interface conditions near the molecular surface with surprisingly good convergence and is well suited for high-performance computing. By adopting a 7-point stencil, the HA method is advantageous in generating simple 7-banded coefficient matrices, which greatly facilitate linear system solution with dense data parallelism, on high-performance computing platforms such as graphics processing unit (GPU). However, the HA method is limited due to its lower accuracy. Therefore, it would be of great interest for high-performance applications to develop more accurate methods while retaining the simplicity and effectiveness of the 7-point stencil discretization scheme. In this study, we have developed two new algorithms based on the spirit of the HA method by introducing more physical interface relations and imposing the discretized Poisson's equation to the second order, respectively. Our testing shows that, for typical biomolecules, the new methods significantly improve the numerical accuracy to that comparable to the second-order solvers, and with ~65% overall efficiency gain on widely available high-performance GPU platforms.

Graphical Abstract

*Please send correspondence to: ruxiq@uci.edu; rluo@uci.edu.

Supporting Information. 1. Quality of the χ approximation on the analytical octupole model and a small molecule DMP. 2. Both proposed methods are released in the PBSA program in the Amber 2019 and AmberTools 2019 releases.



High-order interface schemes are often needed for accurate modeling of electrostatic solvation interactions for biomolecules due to their complex electrostatic environments with Poisson-Boltzmann continuum solvent models. However, these methods demand significant amount of computing resources and are difficult to port to high-performance computing platforms. We have explored simpler strategies and developed new methods that give similar accuracy, but with dramatically reduced computing time.

1. Introduction

Electrostatic interactions are of great significance in biophysical processes, such as protein-protein and protein-ligand associations. Thus, accurate and efficient treatment of electrostatics is crucial to computational studies of biomolecular structures, dynamics, and functions. Since water molecules are always involved in biophysical processes, effectively modeling their electrostatic interactions with biomolecules has been under active exploration. Implicit solvation modeling scheme has been such an attempt, which models solvent molecules in a structureless dielectric medium but biomolecular solute in atomic details. In this scheme, the solvent and solute molecules are often modeled as high and low dielectric regions, respectively, or as Gaussian-based smooth dielectrics in an interface-free manner¹⁻³. Because most atoms are used to represent solvent molecules in explicit all-atom simulations of biomolecular systems under physiological conditions, modeling solvent molecules implicitly would allow higher computational efficiency without sacrificing the atomic resolution for the biomolecules. Among all the attempts, Poisson-Boltzmann equation (PBE) based implicit solvent models have proven to be among the most successful and are widely used in computational studies of biomolecules.

For biomolecular applications, numerically solving PBE is always inevitable because its analytical solution is only possible in a few specific cases with extremely simple solute geometries. Beside semi-analytical generalized Born approaches,⁴⁻¹⁵ most common numerical strategies include the following: finite-difference methods (FDM),¹⁶⁻³¹ finite-element methods³²⁻⁴¹ and boundary-element methods.⁴²⁻⁵⁹ FDM, out of its simplicity and physical transparency in the discretization process, is one of the most popular schemes. However, if a direct discretization is used without considering the discontinuity of the dielectric constant, the numerical solutions tend to have large errors, especially at locations

near the solute-solvent interface. In order to overcome this difficulty, Davis and McCammon proposed a harmonic average (HA) method to alleviate the dielectric discontinuity near the interface in 1991.⁶⁰ Since then further efforts have also been reported to develop better interface schemes, such as the immersed interface method (IIM) by Li and co-workers,^{61–64} and the matched interface and boundary (MIB) method by Wei and co-workers.^{65–68} The idea of IIM is to enforce the interface conditions into the finite-difference schemes at grid points near the interface; while that of MIB is to enforce the lowest-order jump condition repeatedly to achieve an high-order jump condition. In addition to these higher-order numerical methods, some other approaches have also been proposed to improve the implicit solvent models, such as using level set functions to define the interfaces for dielectric assignment,^{69–71} and coupling electrostatic and nonelectrostatic interactions within the implicit solvation treatment.^{72–79}

Apparently higher order interface schemes such as IIM and MIB in principle would lead to higher accuracy for PBE applications.^{80–81} However, they are expensive to use. For example, the IIM is quite involving for complex biomolecular environment due to its extensive use of the level set method to compute interface geometric properties.^{61–64} This is because the second-order solvers (e.g. IIM and MIB) are usually very liberal in the use of grid points nearby the interface to improve the numerical accuracy.^{61–68} For example IIM uses 27 grid points for interface grid points, leading to 27-banded matrices. This is different from the first-order HA method, which only uses six nearest grid points to set up discretized linear equations with 7-banded matrices. Interestingly, the HA method shows surprisingly good global convergence property, particularly at typically used coarse grids (e.g. 0.5 Å).

However, even with the advantages of simplicity and effectiveness, the HA method is clearly limited due to its lower accuracy. Thus, it would be of great interest if the accuracy of simpler 7-point-stencil methods such as HA can be further improved. Furthermore, the 7-banded matrix feature is also advantageous for high-performance computing platforms, such as graphics processing units (GPUs).⁸² GPUs have a parallel architecture that is suited for high-performance computation with dense data parallelism and have been used to accelerate linear PBE solutions for biomolecular systems with impressive speedup.^{82–84} The compact 7-banded matrix is more favorable in GPU implementation than the complex 27-banded matrix due to its denser data parallelism. In addition, in the latter case 20 bands out of the 27-banded matrix have non-zero entries only for grid points nearby the interface, which leads to huge waste of memory if implemented in the band-optimal diagonal matrix format on GPUs. Therefore, improving the accuracy of 7-point stencil discretization schemes such as HA while still maintaining the simplicity and effectiveness will greatly promote applications of PBE implicit modeling on modern GPU computing platforms.

In this study, we have developed two new algorithms in the 7-point stencil discretization scheme by introducing better physical interface relations and imposing the discretized Poisson's equation to the second order, respectively. To achieve this goal, we first re-derived the HA relations in a more general scheme and then developed further extensions from the original idea. Our testing data show that the new methods significantly improve the accuracy to that comparable to the second-order IIM. Meanwhile, the excellent convergence property of the HA method is still maintained. Finally, we present our implementation of the new

method onto GPU platforms, which shows impressive efficiency enhancement, making the new methods capable of high-performance biomolecular computational studies.

2. Methods

2.1 Finite Difference Method

As widely used for numerically solving partial differential equations (PDE), the FDM uses a uniform Cartesian grid to discretize the PDE. The grid points are numbered as (i, j, k) , where $i = 1, \dots, x_m, j = 1, \dots, y_m, k = 1, \dots, z_m$, and x_m, y_m and z_m are the numbers of points along the three axes. The grid spacing between neighboring points can be uniformly set to h .

To discretize Poisson's equation,

$$\nabla \cdot \epsilon \nabla \phi = -4\pi\rho, \quad (1)$$

the charge density $\rho(i, j, k)$ can be expressed as $q(i, j, k)/h^3$, where $q(i, j, k)$ is the total charge within the cubic volume centered at (i, j, k) . The final discretized Poisson's equation is,

$$\begin{aligned} & \left\{ \epsilon\left(i - \frac{1}{2}, j, k\right) [\phi(i-1, j, k) - \phi(i, j, k)] \right. \\ & + \epsilon\left(i + \frac{1}{2}, j, k\right) [\phi(i+1, j, k) - \phi(i, j, k)] \\ & + \epsilon\left(i, j - \frac{1}{2}, k\right) [\phi(i, j-1, k) - \phi(i, j, k)] \\ & + \epsilon\left(i, j + \frac{1}{2}, k\right) [\phi(i, j+1, k) - \phi(i, j, k)] \\ & + \epsilon\left(i, j, k - \frac{1}{2}\right) [\phi(i, j, k-1) - \phi(i, j, k)] \\ & \left. + \epsilon\left(i, j, k + \frac{1}{2}\right) [\phi(i, j, k+1) - \phi(i, j, k)] \right\} / h^2 = -4\pi q(i, j, k) / h^3, \end{aligned} \quad (2)$$

where, $\phi(i, j, k)$ is the potential at grid (i, j, k) , and $\epsilon\left(i - \frac{1}{2}, j, k\right)$ is the dielectric constant at the mid-point of the grids (i, j, k) and $(i-1, j, k)$, and all other ϕ and ϵ are defined similarly here.

A key feature of the discretized PDE is that the coefficient matrix of the linear equation system is in a 7-banded structure, and thus can be naturally wrapped into the diagonal (DIA)⁸⁵ matrix format. The advantage lies in that for a banded matrix, the DIA format is the most efficient sparse matrix form for matrix-related linear operations in terms of both memory and CPU time. For example, comparing to the widely used compressed sparse row (CSR)⁸⁵ format, the DIA format uses only about half of the memory since it does not store the coefficient indices for the matrix. It is also advantageous for matrix operations, such as matrix-vector multiplications, as there is no need to retrieve index information from memory. The simplicity and orderliness in matrix operations in the banded format further promote the performance on GPUs as our recent analysis of multiple PB solvers has shown.⁸² Thus an important motivation of the current study is to explore how to preserve the banded structure in the coefficient matrix while achieving higher-order accuracy numerical solutions.

2.2 Harmonic Average Method in Terms of General Discretization Scheme

In Eqn (2), the dielectric constants are defined at the mid-points of all grid edges. As each edge is flanked by two grid points, it is natural to assign the dielectric constant to that of solute (or solvent) if both grid points belong to solute (or solvent). The complication arises when the two flanking grid points belong to different regions, i.e. the grid edge is across the solute/solvent interface.

A pioneering idea was the Harmonic Average (HA) method as proposed by Davis and McCammon in 1991. The HA method was derived from an infinite parallel plate capacitor model.⁶⁰ Although it is an excellent simple model to capture the basic physics of the PBE, it is limited in generality for further development. Thus, the HA relation is first re-derived in a more general scheme for discretized PDEs below.

As PBE is a 3-d (dimensional) second order PDE, the 7-point stencil is the simplest way to discretize the equation to achieve the second-order accuracy in both solvent and solute regions, but not at the interface. The goal of HA is to achieve an accuracy level as high as possible while still using the simplest 7-stencil scheme. To illustrate its idea, consider a 1-d interface illustrated in Figure 1. The general form of a discretized PDE in 1-d can be written as

$$\gamma_1\phi_1 + \gamma_2\phi_2 + \gamma_3\phi_3 + C = 0, \quad (3)$$

where ϕ_i and γ_i are potentials on the grid points and coefficients to be determined, respectively, and C is a constant.

First, consider the 1-d (or the x -direction) interface geometry in Figure 1A. Expressing ϕ_i in the Taylor expansion at the interface point yields

$$\begin{cases} \phi_1 = \phi - (1-a)h \cdot \phi_x^- + \frac{1}{2}(1-a)^2h^2 \cdot \phi_{xx}^- \\ \phi_2 = \phi + ah \cdot \phi_x^+ + \frac{1}{2}a^2h^2 \cdot \phi_{xx}^+ \\ \phi_3 = \phi + (1+a)h \cdot \phi_x^+ + \frac{1}{2}(1+a)^2h^2 \cdot \phi_{xx}^+ \end{cases}, \quad (4)$$

where “-” and “+” denote the solute (inside) and solvent (outside) regions, respectively. ϕ , ϕ_x and ϕ_{xx} denote the potential, the first and the second derivatives of the potential at the interface point, respectively. Similar expressions in y and z directions can also be obtained simply by changing subscripts. In the HA method, the following interface relations are assumed

$$\phi_x^+ = \frac{\epsilon_u}{\epsilon_v} \phi_x^-$$

$$\phi_{xx}^+ = \frac{\epsilon_u}{\epsilon_v} \phi_{xx}^- \quad (5)$$

Here, ϵ_u means the dielectric constant of the solute region (inside), and ϵ_v means that of the solvent region (outside). These interface relations actually imply a conductor-like approximation, i.e. with all tangential components assumed to be zero. Of course, this is different from a conductor-like solvent model with an infinitely large dielectric constant. Substituting Eqn (4) into Eqn (3) and applying the interface relations Eqn (5), we obtain

$$\begin{aligned} & (\gamma_1 + \gamma_2 + \gamma_3) \cdot \phi + \left[-(1-a)h\gamma_1 + \frac{\epsilon_u}{\epsilon_v}ah\gamma_2 + \frac{\epsilon_u}{\epsilon_v}(1+a)h\gamma_3 \right] \cdot \phi_x^- \\ & + \left[\frac{1}{2}(1-a)^2h^2\gamma_1 + \frac{1}{2}\frac{\epsilon_u}{\epsilon_v}a^2h^2\gamma_2 + \frac{1}{2}\frac{\epsilon_u}{\epsilon_v}(1+a)^2h^2\gamma_3 \right] \cdot \phi_{xx}^- + C = 0 \end{aligned} \quad (6)$$

Eqn (6) should be consistent with the PBE in 1-d so that the coefficients for the 0th- and 1st-order terms are all zero. These conditions lead to

$$\begin{cases} C = 4\pi q_2/h \\ \gamma_1 + \gamma_2 + \gamma_3 = 0 \\ -(1-a)h\gamma_1 + \frac{\epsilon_u}{\epsilon_v}ah\gamma_2 + \frac{\epsilon_u}{\epsilon_v}(1+a)h\gamma_3 = 0 \end{cases} \quad (7)$$

Eqn (7) cannot uniquely determine all coefficients. However, it is obvious that the coefficients used in the HA method, shown in Eqn (8) below, are a set of special solutions of Eqn (7),

$$\begin{cases} \gamma_1 = \frac{1}{\frac{a}{\epsilon_v} + \frac{1-a}{\epsilon_u}} \\ \gamma_2 = -\frac{1}{\frac{a}{\epsilon_v} + \frac{1-a}{\epsilon_u}} - \epsilon_o \\ \gamma_3 = \epsilon_v \end{cases} \quad (8)$$

For the interface geometry in Figure 1B, a similar result can also be obtained by the same argument as

$$\begin{cases} \gamma_1 = \epsilon_u \\ \gamma_2 = -\frac{1}{\frac{a}{\epsilon_u} + \frac{1-a}{\epsilon_v}} - \epsilon_u \\ \gamma_3 = \frac{1}{\frac{a}{\epsilon_u} + \frac{1-a}{\epsilon_v}} \end{cases} \quad (9)$$

It is clear from the above derivations that the HA method makes two vital approximations: utilizing the conductor-like interface conditions in Eqn (5) and imposing the discretization conditions only on the 0th- and 1st-order terms with the 2nd-order term ignored. The omission in principle does not guarantee that Eqn (6) is consistent with PBE at any finite value for the grid spacing, though it is possible as the grid spacing approaches zero. In this study, we aim to develop new methods by modifying these two approximations, while maintaining the simplicity of a 7-point stencil as in the HA method for GPU implementations.

2.3 First Modification: Imposing Discretization Conditions up to the Second Order

A straightforward modification can be proposed by imposing the discretization conditions in Eqn (6) to the 2nd-order, which yields

$$\left\{ \begin{array}{l} C = 4\pi q_2/h \\ \gamma_1 + \gamma_2 + \gamma_3 = 0 \\ -(1-a)h\gamma_1 + \frac{\epsilon_u}{\epsilon_v}ah\gamma_2 + \frac{\epsilon_u}{\epsilon_v}(1+a)h\gamma_3 = 0 \\ \frac{1}{2}(1-a)^2h^2\gamma_1 + \frac{1}{2}\frac{\epsilon_u}{\epsilon_v}a^2h^2\gamma_2 + \frac{1}{2}\frac{\epsilon_u}{\epsilon_v}(1+a)^2h^2\gamma_3 = h^2\epsilon_u \end{array} \right. \quad (10)$$

This replaces Eqn (7) in the HA method as the new coefficient relations for the interface in Figure 1A. It is clear that the 2nd-order term of Eqn (6) is also used as a constraint to determine the discretized equation, so that Eqn (6) is consistent with the PBE regardless the grid spacing used, different from the HA method. Here the right-hand side of the fourth equation above can be arbitrarily set to any nonzero constant. The choice of h^2 is for the simplicity of canceling the same factor on the left-hand side, and the use of ϵ_u is to make sure that the coefficients of the new discretized linear equation reduce to those of the standard 7-point stencil when there is no interface. It should be noted that even if the new discretization is imposed to the second order in each dimension, this is rather different from the second-order IIM, as IIM considers more detailed 3-d interface relations and utilizes a 27-point stencil.⁸⁰

Solving Eqn (10) yields,

$$\left\{ \begin{array}{l} \gamma_1 = \frac{2\epsilon_u}{\frac{\epsilon_u}{\epsilon_v}(1+a)a + (2+a)(1-a)} \\ \gamma_2 = -2\epsilon_u \frac{(1+a) + \frac{\epsilon_v}{\epsilon_u}(1-a)}{\frac{\epsilon_u}{\epsilon_v}(1+a)a + (2+a)(1-a)} \\ \gamma_3 = 2\epsilon_u \frac{a + \frac{\epsilon_v}{\epsilon_u}(1-a)}{\frac{\epsilon_u}{\epsilon_v}(1+a)a + (2+a)(1-a)} \end{array} \right. \quad (11)$$

Similarly, for the interface in Figure 1B, the following solutions can be obtained,

$$\left\{ \begin{array}{l} \gamma_1 = 2\epsilon_u \frac{a + \frac{\epsilon_u}{\epsilon_v}(1-a)}{(1+a)a + \frac{\epsilon_u}{\epsilon_v}(2+a)(1-a)} \\ \gamma_2 = -2\epsilon_u \frac{(1+a) + \frac{\epsilon_u}{\epsilon_v}(1-a)}{(1+a)a + \frac{\epsilon_u}{\epsilon_v}(2+a)(1-a)} \\ \gamma_2 = \frac{2\epsilon_u}{(1+a)a + \frac{\epsilon_u}{\epsilon_v}(2+a)(1-a)} \end{array} \right. \quad (12)$$

Obviously, the modification only scales the coefficients of the classical HA method. Thus, it is reasonable to assume this modification would improve the accuracy without jeopardizing its other properties, such as the stability and convergence behaviors. In addition, the proposed change retains the compact 7-banded structure of the final coefficient matrix, which is favorable for GPU implementations. For simplicity, we term this revised method as the second-order HA method and will refer to it throughout the manuscript.

2.4 Second Modification: Relaxing the Conductor-like Interface Conditions

The interface relations in Eqn (5) are similar to those on a conductor surface,

$$\phi_{\tau}^{-} = 0, \quad (13)$$

where τ denotes the tangential direction of the electric field on the interface. However, the outside region obviously does not have an infinitely large dielectric constant ($\epsilon_v = 80$ for water). As a result, the tangential electric field cannot be exactly zero. The physical interface relations for the first derivatives of electric potential are,

$$\phi_x^{-} = \phi_{\xi}^{-} \cos(\hat{\xi}, \hat{x}) + \phi_{\tau}^{-} \cos(\hat{\tau}, \hat{x})$$

$$\phi_x^{+} = \frac{\epsilon_u}{\epsilon_v} \phi_{\xi}^{-} \cos(\hat{\xi}, \hat{x}) + \phi_{\tau}^{-} \cos(\hat{\tau}, \hat{x}) \quad (14)$$

where ξ denotes the normal direction of the electric field on the interface. The above equations show that the factor $\frac{\epsilon_u}{\epsilon_v}$ in Eqn (5) should be replaced by a more physical term, χ ,

because $\frac{\phi_x^{+}}{\phi_x^{-}}$ is no longer equal to $\frac{\epsilon_u}{\epsilon_v}$ as in the conductor-like interface,

$$\chi = \frac{\phi_x^{+}}{\phi_x^{-}} = \frac{\frac{\epsilon_u}{\epsilon_v} \phi_{\xi}^{-} \cos(\hat{\xi}, \hat{x}) + \phi_{\tau}^{-} \cos(\hat{\tau}, \hat{x})}{\phi_{\xi}^{-} \cos(\hat{\xi}, \hat{x}) + \phi_{\tau}^{-} \cos(\hat{\tau}, \hat{x})} \quad (15)$$

Now the challenge is to evaluate ϕ_{ξ}^{-} and ϕ_{τ}^{-} in Eqn (15). This requires a careful look at the conductor approximation. For a perfect conductor (ϵ_v goes to infinity), ϕ_C^{\pm} , the electric field generated by free charges (Coulomb field), and ϕ_{RF}^{\pm} , the electric field generated by induced charges (reaction field), satisfy the following relations on the interface,

$$\begin{aligned}(\phi_{\xi}^{\pm})_C &= \mp (\phi_{\xi}^{\pm})_{RF} \\ (\phi_{\tau}^{\pm})_C &= - (\phi_{\tau}^{\pm})_{RF}.\end{aligned}\tag{16}$$

The first equation of (16) shows that the normal field is zero inside the conductor (i.e. the solvent region) and is twice the magnitude of the Coulomb field outside the conductor (i.e. the solute region). The second equation of (16) shows that the tangential field is always zero as the two terms cancel out. However, for a dielectric medium such as water ($\epsilon_v = 80$), the above relations in the normal direction may still be similar for the two fields, while in the tangential direction the induced field cannot fully cancel the Coulomb field and thus a non-zero residual component exists. The tangential residual component may be small, but definitely not zero. Therefore, we use the following approximations to evaluate the normal and tangential components ϕ_{ξ}^{-} and ϕ_{τ}^{-} ,

$$\begin{aligned}\phi_{\xi}^{-} &\approx 2(\phi_{\xi}^{-})_C \\ \phi_{\tau}^{-} &\approx \frac{2\epsilon_u}{\epsilon_v}(\phi_{\tau}^{-})_C\end{aligned}\tag{17}$$

and thus χ can be approximated as,

$$\chi = \frac{\epsilon_u}{\epsilon_v} \frac{(\phi_{\xi}^{-})_C \cos(\hat{\xi}, \hat{x}) + (\phi_{\tau}^{-})_C \cos(\hat{\tau}, \hat{x})}{(\phi_{\xi}^{-})_C \cos(\hat{\xi}, \hat{x}) + \frac{\epsilon_u}{\epsilon_v} (\phi_{\tau}^{-})_C \cos(\hat{\tau}, \hat{x})}\tag{18}$$

It is obvious that the above relations reduce to the conductor-like interface relations when ϵ_v goes to infinity. While the first relation in Eqn (17) has a reasonable physical basis, the second relation is an approximation proposed after comparing with the simple sphere geometries that can be solved analytically, as many proteins are shaped (approximately) as spheres (globular proteins). The derivation is offered in the section 2.5, and we also provide validations, on an analytical octupole model imbedded in a sphere and a small molecule (DMP) with 13 atoms, in the Supporting Materials.

Once factor χ is obtained, substituting it into Eqn (8) and Eqn (9) yields

$$\begin{cases} \gamma_1 = \epsilon_v \frac{1}{a + \frac{1-a}{\chi}} \\ \gamma_2 = -\epsilon_v \frac{1}{a + \frac{1-a}{\chi}} - \epsilon_v \\ \gamma_3 = \epsilon_v \end{cases} \quad (19)$$

$$\begin{cases} \gamma_1 = \epsilon_u \\ \gamma_2 = -\epsilon_u \frac{1}{a + \chi(1-a)} - \epsilon_u \\ \gamma_3 = \epsilon_u \frac{1}{a + \chi(1-a)} \end{cases} \quad (20)$$

This modification would also keep the simple 7-banded coefficient matrix from a standard 7-point stencil, suitable for GPU implementations. Also, because this modification involves only a minor change (replacing $\frac{\epsilon_u}{\epsilon_v}$ with χ) to the classical HA method, we assume it would not compromise other properties, such as fast convergence, good stability etc., while improving its accuracy. In the following, we term this modification as the χ -factor HA method and will refer to it throughout the manuscript.

2.5 Analytical Test Cases and the Determination of χ Factor

An analytical model of a single dielectric sphere embedded with point charges was used to validate the accuracy and precision of the proposed methods. The analytical reaction and total fields of the inside region in spherical coordinates are,⁶²

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial r} = \sum_{l=0}^{\infty} \sum_{m=-l}^l -l \frac{4\pi}{2l+1} \frac{1}{R^{2l+1}} \frac{(l+1)(\epsilon_u - \epsilon_v)}{\epsilon_u(l\epsilon_u + (l+1)\epsilon_v)} Q_{lm} r^{l-1} Y_{lm}(\theta, \varphi)$$

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \theta} = \sum_{l=0}^{\infty} \sum_{m=-l}^l -l \frac{4\pi}{2l+1} \frac{1}{R^{2l+1}} \frac{(l+1)(\epsilon_u - \epsilon_v)}{\epsilon_u(l\epsilon_u + (l+1)\epsilon_v)} Q_{lm} r^l \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \theta}$$

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \varphi} = \sum_{l=0}^{\infty} \sum_{m=-l}^l -l \frac{4\pi}{2l+1} \frac{1}{R^{2l+1}} \frac{(l+1)(\epsilon_u - \epsilon_v)}{\epsilon_u(l\epsilon_u + (l+1)\epsilon_v)} Q_{lm} r^l \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \varphi}$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial r} = \sum_{l=0}^{\infty} \sum_{m=-l}^l -l \frac{\epsilon_v}{\epsilon_u} \frac{-4\pi(l+1)}{(l\epsilon_u + (l+1)\epsilon_v)} Q_{lm} \frac{1}{R^{l+2}} Y_{lm}(\theta, \varphi)$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \theta} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \theta}$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \varphi} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \varphi}$$

$$\text{with, } Q_{lm} = \sum_{k=1}^{N_q} q_k r_k^l Y_{lm}^*(\theta_k, \varphi_k) \quad (21)$$

where R is the radius of the sphere with the center set to the origin, N_q is the number of charges, and Y_{lm} is the spherical harmonic function.

With the above analytical results, we are able to estimate the new jump condition, i.e. the χ factor. From Eqn (21), we can obtain the electric fields of the inside region on the interface as,

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial r} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{2l+1} \frac{(l+1)(\varepsilon_u - \varepsilon_v)}{\varepsilon_u(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{l}{R^{l+2}} Y_{lm}(\theta, \varphi)$$

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \theta} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{2l+1} \frac{(l+1)(\varepsilon_u - \varepsilon_v)}{\varepsilon_u(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \theta}$$

$$\frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \varphi} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{2l+1} \frac{(l+1)(\varepsilon_u - \varepsilon_v)}{\varepsilon_u(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \varphi}$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial r} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{\varepsilon_v}{\varepsilon_u} \frac{-4\pi(l+1)}{(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+2}} Y_{lm}(\theta, \varphi)$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \theta} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \theta}$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \varphi} = \sum_{l=0}^{\infty} \sum_{m=-l}^l \frac{4\pi}{(l\varepsilon_u + (l+1)\varepsilon_v)} Q_{lm} \frac{1}{R^{l+1}} \frac{\partial Y_{lm}(\theta, \varphi)}{\partial \varphi} \quad (22)$$

From the analysis of Eqns (16) and (17), we have,

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial r} \approx 2 \frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial r} \approx 2 \frac{\partial \phi_C^-(r, \theta, \varphi)}{\partial r} \quad (23)$$

Comparing Eqn (23) with Eqn (22), we obtain,

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \theta} \approx -\frac{2\varepsilon_u}{\varepsilon_v} \frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \theta} \approx \frac{2\varepsilon_u}{\varepsilon_v} \frac{\partial \phi_C^-(r, \theta, \varphi)}{\partial \theta}$$

$$\frac{\partial \phi^-(r, \theta, \varphi)}{\partial \varphi} \approx -\frac{2\varepsilon_u}{\varepsilon_v} \frac{\partial \phi_{RF}^-(r, \theta, \varphi)}{\partial \varphi} \approx \frac{2\varepsilon_u}{\varepsilon_v} \frac{\partial \phi_C^-(r, \theta, \varphi)}{\partial \varphi} \quad (24)$$

Eqn (23) and (24) lead to the approximations used in Eqn (17).

Note that although Eqn (24) is a good approximation for spherical cases, its correctness and effectiveness for complex biomolecular systems can only be validated by actual numerical experiments. A good approximation should have at least the following two features. First, the estimated χ values follow the trend of true χ values relatively well, so they can capture more accurate interface geometries and field distributions. Second, the systematical error should be as small as possible. The tests and analyses of the approximation are shown in Supporting Materials.

3. Computation Details

The potentials at grid points are obtained by solving PBE numerically. Then the potential at any position (x,y,z) is obtained with a one-sided least-squares interpolation method.⁶² The linear Poisson-Boltzmann energy (PB energy) can then be obtained as the summation of the charges multiplying their respective reaction field potentials as

$$\Delta G = \frac{1}{2} \sum_{charges} q_i \phi_{RF} \quad (25)$$

For the analytical test cases, four off-center charge models were used, which are monopole, dipole, quadrupole, and octupole, respectively. The radius of the sphere was set to 1.6 Å, which is about the size of a carbon atom. The Cartesian coordinates of each charge model are listed in Table 1. The inside and outside dielectric constants were set to 1.0 and 80.0, respectively. The truncation order in Eqn (21) was chosen to be 120, which gives a precision of 10^{-6} .

In addition, a total of 573 biomolecular structures from the Amber PBSA benchmark suite were used to study the overall accuracy and efficiency of the new methods.²⁹ Of these molecules, eight small proteins of about 1000 atoms from the suite were selected to analyze the properties of the new method in more details, so that computational jobs could be handled on our local compute nodes at the finest grid spacings tested. These biomolecules were assigned with charges of Cornell *et al*⁸⁶ and the modified Bondi radii. The probe radius

was set to 1.4 Å. All testing jobs were performed with the following conditions unless specified otherwise. The convergence criterion of 10^{-4} was used for the biconjugate gradient (BiCG) linear solver. The default grid spacing was set to 0.5 Å for most calculations, except that in the convergence analysis it was set to a range of values of from 1.0 Å to 0.1 Å. The ratio of the grid dimension to the solute dimension (the *fillratio* keyword in Amber) was set to 1.5 and 4.0 for biomolecules and analytical test cases, respectively. No electrostatic focusing was applied. The free boundary condition was used to assign the finite-difference grid boundary potentials and the charge singularity was removed. The dielectric constants were set to 1.0 and 80.0 for solute and solvent, respectively. All other parameters were set as default in the PBSA module in Amber 18 package.^{25, 27, 29–30, 69–71, 82, 84, 87–96}

A BiCG linear system solver for GPUs was also implemented using the Nvidia CUDA Sparse Matrix (cuSPARSE) and CUSP libraries, which provide basic linear algebra procedures for sparse matrix operations.⁹⁷ The DIA matrix format was used for the non-symmetric coefficient matrix as in our previous publication.⁸² All GPU and CPU tests were conducted on a dedicated compute node with two Nvidia RTX 2080Ti GPU cards, one Intel Xeon E5–1620 v3 CPU, and 16GB main memory. Our time measurements for all solvers include all execution time of the solver routines, i.e. time elapsed on the device (GPU) and on the host (CPU) and also for transferring data between the device and the host.

4. Results and Discussion

We extended the classical HA method by two improvements, i.e. imposing the discretization conditions to the second order and relaxing the conductor-like interface relation, respectively. These efforts resulted in the second order and the χ -factor HA methods. In the following, we report the numerical experiments to study the quality of the approximations, and also compare their accuracy and performance with the classical HA method and the analytical IIM. Specifically, analytical models were first used to evaluate the quality of the approximations. Then eight small proteins were studied to evaluate their convergence behaviors, which refer to the convergence of PB energies as a function of grid spacing. This notation is used throughout the manuscript. Next the robustness and correctness of these methods were investigated using a large set of biomolecular structures at the widely used numerical conditions. Furthermore, the BiCG linear solver was implemented for the two new methods utilizing the compact 7-banded structure of the coefficient matrices to boost their efficiency on GPUs. Finally, both the improvement and limitation of the new methods are also discussed.

4.1 Analytical Evaluation of the Approximations

We first tested the new methods using analytical models from monopole to octupole and compared them with the classical HA method and the analytical IIM. The convergence trends of PB energies over the tested grid spacing for all the methods are shown in Figure 2. Clearly, all tests converge well except for the classical HA method on the monopole model. This indicates that the classical HA method may give inaccurate solvation energies even for simple spherical cases, yet the two new HA methods were found to overcome the limitation.

More detailed convergence properties are shown in Table 2, where the extrapolated PB energies at the grid spacing of zero are listed for all models by the four tested methods. The analysis confirms that the PB energies of all the four methods converge to the analytical values reasonably well except the monopole case with the classical HA method.

As interface plays a critical role in the PBE modeling, we further examined the abilities of these methods to capture the interface property by calculating the electric field on the molecular surface. The root mean squared error (RMSE) values were calculated by comparing the numerical electric fields with the analytical results obtained from Eqn (21). Here the surface electric field was interpolated with the one-dimensional method, shown to perform the best for these analytical systems.⁹⁸ Figure 3 indicates that all 7-stencil methods agree well with analytical values, though not as accurate as the analytical IIM as expected. It is also worth noting that the χ -factor HA method is better than the classical HA method on every test point, though not by much. In summary, it is clear that the two approximations work well for all tested analytical systems with no systematic biases.

4.2 Convergence Analysis of the New Methods

To assess the quality of the two new methods on biomolecules, we studied their numerical behaviors with eight small proteins. The convergence trends of PB energies for all the methods are shown in Figure 4 and the detailed extrapolation analysis is presented in Table 3. Figure 4 shows that all tested methods converge smoothly but to two different limiting values: the classical HA method and the second-order HA method converge to more negative values; and the χ -factor HA method and the analytical IIM converge to more positive values. More specifically, the difference between the extrapolated PB energies for the latter two methods are mostly within 1 kcal/mol as shown in Table 3. As the analytical IIM is the most accurate method of the four since there is no approximation in its discretization scheme,⁹⁹ this means the second-order and the classical HA methods tend to give more negative PB energies than the true values for molecular systems. Note that this behavior has been observed for the classical HA method in the analytical monopole test already (Figure 2). However, the behavior is a new observation for the second-order HA method since it performs well in the analytical tests. This shows that the second-order HA method tends to degenerate to the classical HA method when it is applied to complicated molecular geometries such as proteins. In contrast, the χ -factor HA method still maintains its excellent numerical behaviors even on proteins, and almost produces the same accuracy as the high-order IIM does with its 27-point stencil.

Another observation from Figure 4 is that the classical and the χ -factor HA methods have much flatter convergence curves than the second-order HA method. A closer look of the PB energies at the grid spacing of 0.5 Å in Table 4 shows that the relative errors with respect to methods' own limiting values are 0.1%, 0.14%, and 0.21%, respectively, for the classical, the χ -factor, and the second-order HA methods. This behavior is rather different from what is observed in the analytical spherical tests, where the second-order HA method converges the fastest among the three HA methods.

Taken all the observations together, we can conclude that when dealing with realistic molecular systems such as proteins, the χ -factor HA method preserves the overall good

convergence behavior of the classical HA method while it converges to the correct limiting values. Worth noting is that the χ -factor HA method performs even better than the IIM with overall flatter convergence curves. It is worth pointing out that the slower convergence of the second-order methods may result from their overly precise description of the local complex interface geometries in typical proteins.⁸⁰ It is likely that fuzzy handling of complex geometries as done in simpler methods may lead to a good apparent convergence trend simply because the methods are not designed to detect complex geometries. However, what the χ -factor HA method gained over the classical HA method is that it converges to the correct limiting values as those from the analytical IIM, while still preserving the good convergence trend of the classical HA method. As a result, its simplicity, good convergence property, and numerical stability make the χ -factor HA method an outstanding scheme among the four investigated methods in practices.

4.3 Robustness of the New Methods

To examine the robustness, i.e. the success rate, of these methods under typical numerical conditions, we computed the PB energies with 573 biomolecular structures from the Amber PBSA benchmark suite at the commonly used grid spacing of 0.5 Å. No failure to obtain PB energy (NaN or positive numbers as energy values) was observed for any of the 573 test cases, which confirms that our new methods are robust enough for various biomolecules.

To confirm the correctness of the new methods, we analyzed the correlations between the PB energies from the three HA methods with those from the analytical IIM. As shown in Figure 5, the linear fitting slopes are 1.004, 0.998, and 1.000 for the classical, the χ -factor, and the second-order HA methods, respectively. This indicates that compared to the IIM, the classical HA method has a systematic tendency to give more negative results and the χ -factor HA method reduces the error to about a half, and the second-order HA method has almost no systematic tendency at the tested condition. Interestingly, the second-order HA method is the closest to the IIM at the tested condition, which may seem to be more accurate than the other two HA methods. However, note that the calculations were carried out at the grid spacing of 0.5 Å, where these methods may deviate from the limiting values (i.e. at the grid spacing of 0.0 Å). While it is difficult and impractical to obtain extrapolated limiting PB energies for all 573 biomolecular structures, the results from the eight representative proteins as discussed in section 4.2 can serve as a guidance. As shown in Figure 4, the convergence trendlines of the IIM and the second-order HA method cross over near the grid spacing of 0.5 Å and thus their PB energies are closer to each other.

To quantitatively measure the accuracy of each method at the tested grid spacing, we calculated the relative error of the PB energy from each method (see Table 4) with respect to the IIM's limiting values (see Table 3). The medium relative errors are 0.6%, 0.2%, 0.1% and 0.13% for the classical, the χ -factor HA, the second-order HA, and the IIM, respectively. This means that at the grid spacing of 0.5 Å, a common choice for practical biomolecular applications, the χ -factor and the second-order HA methods are with very similar errors as the IIM. This further illustrates the impressive improvements achieved in the two new HA methods. It is interesting that the second-order HA method coincidentally behaves well at the widely used grid spacing even if its limiting convergence is closer to the

classical HA method. However, we expect that in a wider range of conditions, the χ -factor HA method would be more accurate and stable as it has better convergence behavior with various grid spacings as discussed above.

4.4 GPU Implementation

An important motivation in developing the new HA methods is to utilize their compact 7-banded structure in the coefficient matrices to improve computational efficiency on high-performance platforms such as GPUs. We implemented the BiCG linear solver on GPUs for the two new HA methods, as the BiCG algorithm is suitable for asymmetric matrices. The DIA matrix format was used to store the matrices in the new methods. In contrast the CSR matrix format was used in the analytical IIM. The timing results for 573 biomolecules are shown in Figure 6, with the timings of the BiCG solver implemented with the IIM on GPUs as the benchmark. The results show that the average solver time (Figure 6 top two) for the χ -factor and the second-order HA methods is about 35% and 39% of that of the IIM, respectively. This means that by using a more compact matrix structure, the solver time can be reduced by about two thirds. As simpler interface schemes, the new methods also reduce the setup time significantly (Figure 6 bottom two). This is because the IIM uses a multistep, complicated setup procedure,⁸⁰ which requires the interface geometry parameters up to the second order (curvatures) and the Coulomb field parameters of the second order (gradients). While for the two new methods, all these parameters (interface normal directions and Coulomb field) are required only to the first order. As a result, the intrinsically simpler HA methods leads to significant efficiency gain in PBE modeling on GPU platforms.

5. Conclusion

In this study, we explored new 7-stencil algorithms based on the spirit of the classic HA method for PBE modeling, namely, the χ -factor HA method and the second-order HA method by introducing better physical interface relations and imposing the discretized Poisson's equation to the second order. The new methods improve on the original one and still maintain the compact 7-banded structure for the discretized equations. As a result, the advantages of fast matrix operation and reduced memory requirement are obtained for numerical implementation, especially on GPUs for parallel computation.

We first tested our new methods on multiple analytical test cases to serve as a primary check for the quality of the approximations used in the new methods. The results showed that the new methods overcome the limitation of the classical HA method in the monopole case by giving more accurate limiting PB energy. In addition, the new methods also give correct electric field on the interface, showing that the new methods are overall consistent with theory for the tested analytical systems.

We further analyzed the convergence of the new methods on small proteins using the analytical IIM as reference, to study their advantages on biomolecules. The results show that the χ -factor HA method performs very well in both accuracy and convergence with its limiting PB energies mostly within 1 kcal/mol difference from the reference values. Therefore, its simplicity, good convergence property, and stability on complex protein molecules make χ -factor HA an outstanding scheme among the four investigated methods.

We next examined the robustness and correctness of the new methods with 573 biomolecular structures at the commonly used grid spacing of 0.5 Å. The results show that all tests passed with no failure, confirming that our new methods are robust enough for various complex interface geometries in biomolecules. Interestingly, the second-order HA method agrees the best with the IIM in terms of PB energies. Our further analysis of the relative error of the PB energies with respect to the IIM limiting values shows that two new HA methods are with very similar errors as the analytical IIM at the widely used numerical conditions. We further pointed out that though the second-order HA method performs well at the grid spacing of 0.5 Å, it is expected that the χ -factor HA method will be more accurate and stable in a wider range of application conditions.

Finally, we implemented the BiCG linear solver for the new HA methods and tested their performance on GPUs. The results show that by using a more compact 7-banded matrix structure, the solver time can be reduced by about two thirds, and the setup time are also significantly reduced over the IIM method. As a result, the intrinsic simple design of the HA methods leads to significant efficiency gain of PBE modeling on GPU platforms, which makes the new methods more suitable for high-performance biomolecular computational studies.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgement.

This work was supported by the National Institute of Health/NIGMS (GM093040 & GM079383).

6. References

1. Li L; Li C; Zhang Z; Alexov E, On the dielectric “constant” of proteins: smooth dielectric function for macromolecular modeling and its implementation in DelPhi. *Journal of chemical theory and computation* 2013, 9 (4), 2126–2136. [PubMed: 23585741]
2. Chakravorty A; Jia Z; Peng Y; Tajilyato N; Wang L; Alexov E, Gaussian-Based Smooth Dielectric Function: A Surface-Free Approach for Modeling Macromolecular Binding in Solvents. *Frontiers in molecular biosciences* 2018, 5, 25. [PubMed: 29637074]
3. Li C; Jia Z; Chakravorty A; Pahari S; Peng Y; Basu S; Koirala M; Panday SK; Petukh M; Li L, DelPhi Suite: New Developments and Review of Functionalities. *Journal of Computational Chemistry* 2019.
4. Dominy BN; Brooks CL, Development of a generalized born model parametrization for proteins and nucleic acids. *Journal Of Physical Chemistry B* 1999, 103 (18), 3765–3773.
5. Onufriev A; Bashford D; Case DA, Modification of the generalized Born model suitable for macromolecules. *Journal Of Physical Chemistry B* 2000, 104 (15), 3712–3720.
6. Srinivasan J; Trevathan MW; Beroza P; Case DA, Application of a pairwise generalized Born model to proteins and nucleic acids: inclusion of salt effects. *Theoretical Chemistry Accounts* 1999, 101 (6), 426–434.
7. Tsui V; Case DA, Molecular dynamics simulations of nucleic acids with a generalized born solvation model. *J Am Chem Soc* 2000, 122 (11), 2489–2498.
8. Gallicchio E; Zhang LY; Levy RM, The SGB/NP hydration free energy model based on the surface generalized born solvent reaction field and novel nonpolar hydration free energy estimators. *Journal Of Computational Chemistry* 2002, 23 (5), 517–529. [PubMed: 11948578]

9. Lee MS; Salsbury FR; Brooks CL, Novel generalized Born methods. *J Chem Phys* 2002, 116 (24), 10606–10614.
10. Onufriev A; Case DA; Bashford D, Effective Born radii in the generalized Born approximation: The importance of being perfect. *Journal Of Computational Chemistry* 2002, 23 (14), 1297–1304. [PubMed: 12214312]
11. Gallicchio E; Levy RM, AGBNP: An analytic implicit solvent model suitable for molecular dynamics simulations and high-resolution modeling. *Journal of Computational Chemistry* 2004, 25 (4), 479–499. [PubMed: 14735568]
12. Onufriev A; Bashford D; Case DA, Exploring protein native states and large-scale conformational changes with a modified generalized born model. *Proteins-Structure Function and Bioinformatics* 2004, 55 (2), 383–394.
13. Li XF; Hassan SA; Mehler EL, Long dynamics simulations of proteins using atomistic force fields and a continuum representation of solvent effects: Calculation of structural and dynamic properties. *Proteins-Structure Function and Bioinformatics* 2005, 60 (3), 464–484.
14. Chen JH; Im WP; Brooks CL, Balancing solvation and intramolecular interactions: Toward a consistent generalized born force field. *J. Am. Chem. Soc* 2006, 128 (11), 3728–3736. [PubMed: 16536547]
15. Chocholousova J; Feig M, Implicit solvent simulations of DNA and DNA-protein complexes: Agreement with explicit solvent vs experiment. *Journal Of Physical Chemistry B* 2006, 110 (34), 17240–17251.
16. Klapper I; Hagstrom R; Fine R; Sharp K; Honig B, Focusing of Electric Fields in the Active Site of Copper-Zinc Superoxide Dismutase Effects of Ionic Strength and Amino Acid Modification. *Proteins: Struct., Funct., Genet* 1986, 1 (1), 47–59. [PubMed: 3449851]
17. Sharp K; Honig B, Lattice models of electrostatic interactions—the finite-difference Poisson-Boltzmann method. *Chem. Scr. A* 1989, 29, 71–74.
18. Davis ME; McCammon JA, Solving the Finite-Difference Linearized Poisson-Boltzmann Equation - a Comparison of Relaxation and Conjugate-Gradient Methods. *J. Comput. Chem* 1989, 10 (3), 386–391.
19. Nicholls A; Honig B, A Rapid Finite-Difference Algorithm, Utilizing Successive over-Relaxation to Solve the Poisson-Boltzmann Equation. *J. Comput. Chem* 1991, 12 (4), 435–445.
20. Luty BA; Davis ME; McCammon JA, Solving the Finite-Difference Nonlinear Poisson-Boltzmann Equation. *J. Comput. Chem* 1992, 13 (9), 1114–1118.
21. Holst M; Saied F, Multigrid Solution of the Poisson-Boltzmann Equation. *J. Comput. Chem* 1993, 14 (1), 105–113.
22. Forsten KE; Kozack RE; Lauffenburger DA; Subramaniam S, Numerical-Solution of the Nonlinear Poisson-Boltzmann Equation for a Membrane-Electrolyte System. *J. Phys. Chem* 1994, 98 (21), 5580–5586.
23. Im W; Beglov D; Roux B, Continuum Solvation Model: computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation. *Comput. Phys. Commun* 1998, 111 (1–3), 59–75.
24. Rocchia W; Alexov E; Honig B, Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions. *J. Phys. Chem. B* 2001, 105 (28), 6507–6514.
25. Luo R; David L; Gilson MK, Accelerated Poisson-Boltzmann calculations for static and dynamic systems. *J. Comput. Chem* 2002, 23 (13), 1244–1253. [PubMed: 12210150]
26. Bashford D, An Object-Oriented Programming Suite for Electrostatic Effects in Biological Molecules. *Lecture Notes in Computer Science* 1997, 1343, 233–240.
27. Lu Q; Luo R, A Poisson-Boltzmann dynamics method with nonperiodic boundary condition. *J. Chem. Phys* 2003, 119 (21), 11035–11047.
28. Prabhu NV; Zhu PJ; Sharp KA, Implementation and testing of stable, fast implicit solvation in molecular dynamics using the smooth-permittivity finite difference Poisson-Boltzmann method. *J Comput. Chem* 2004, 25 (16), 2049–2064. [PubMed: 15481091]
29. Wang J; Luo R, Assessment of Linear Finite-Difference Poisson-Boltzmann Solvers. *J. Comput. Chem* 2010, 31 (8), 1689–1698. [PubMed: 20063271]

30. Cai Q; Hsieh MJ; Wang J; Luo R, Performance of Nonlinear Finite-Difference Poisson-Boltzmann Solvers. *J Chem. Theory Comput* 2010, 6 (1), 203–211. [PubMed: 24723843]
31. Li L; Li C; Sarkar S; Zhang J; Witham S; Zhang Z; Wang L; Smith N; Petukh M; Alexov E, DelPhi: a comprehensive suite for DelPhi software and associated resources. *BMC biophysics* 2012, 5 (1), 9. [PubMed: 22583952]
32. Cortis CM; Friesner RA, Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes. *J. Comput. Chem* 1997, 18 (13), 1591–1608.
33. Baker N; Holst M; Wang F, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II. Refinement at solvent-accessible surfaces in biomolecular systems. *J. Comput. Chem* 2000, 21 (15), 1343–1352.
34. Holst M; Baker N; Wang F, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I. Algorithms and examples. *J. Comput. Chem* 2000, 21 (15), 1319–1342.
35. Shestakov AI; Milovich JL; Noy A, Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method. *J. Colloid Interface Sci* 2002, 247 (1), 62–79. [PubMed: 16290441]
36. Chen L; Holst MJ; Xu JC, The finite element approximation of the nonlinear Poisson-Boltzmann equation. *SIAM J. Numer. Anal* 2007, 45, 2298–2320.
37. Xie D; Zhou S, A new minimization protocol for solving nonlinear Poisson-Boltzmann mortar finite element equation. *Bit. Numer. Math* 2007, 47 (4), 853–871.
38. Friedrichs M; Zhou RH; Edinger SR; Friesner RA, Poisson-Boltzmann analytical gradients for molecular modeling calculations. *J Phys. Chem. B* 1999, 103 (16), 3057–3061.
39. Bond SD; Chaudhry JH; Cyr EC; Olson LN, A First-Order System Least-Squares Finite Element Method for the Poisson-Boltzmann Equation. *J Comput. Chem* 2010, 31 (8), 1625–1635. [PubMed: 19908291]
40. Lu B; Holst MJ; McCammon JA; Zhou YC, Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes I: Finite element solutions. *J Comput Phys* 2010, 229 (19), 6979–6994. [PubMed: 21709855]
41. Lu BZ; Zhou YC, Poisson-Nernst-Planck Equations for Simulating Biomolecular Diffusion-Reaction Processes II: Size Effects on Ionic Distributions and Diffusion-Reaction Rates. *Biophys J* 2011, 100 (10), 2475–2485. [PubMed: 21575582]
42. Miertus S; Scrocco E; Tomasi J, Electrostatic Interaction of a Solute with a Continuum - a Direct Utilization of Abinitio Molecular Potentials for the Prevision of Solvent Effects. *Chem. Phys* 1981, 55 (1), 117–129.
43. Hoshi H; Sakurai M; Inoue Y; Chujo R, Medium Effects on the Molecular Electronic-Structure .1. the Formulation of a Theory for the Estimation of a Molecular Electronic-Structure Surrounded by an Anisotropic Medium. *J. Chem. Phys* 1987, 87 (2), 1107–1115.
44. Zauhar RJ; Morgan RS, The Rigorous Computation of the Molecular Electric-Potential. *J. Comput. Chem* 1988, 9 (2), 171–187.
45. Rashin AA, Hydration Phenomena, Classical Electrostatics, and the Boundary Element Method. *J. Phys. Chem* 1990, 94 (5), 1725–1733.
46. Yoon BJ; Lenhoff AM, A Boundary Element Method for Molecular Electrostatics with Electrolyte Effects. *J. Comput. Chem* 1990, 11 (9), 1080–1086.
47. Juffer AH; Botta EFF; Vankeulen BAM; Vanderploeg A; Berendsen HJC, The Electric-Potential of a Macromolecule in a Solvent - a Fundamental Approach. *J. Comput. Phys* 1991, 97 (1), 144–171.
48. Zhou HX, Boundary-Element Solution of Macromolecular Electrostatics - Interaction Energy between 2 Proteins. *Biophys. J* 1993, 65 (2), 955–963. [PubMed: 8218918]
49. Bharadwaj R; Windemuth A; Sridharan S; Honig B; Nicholls A, The Fast Multipole Boundary-Element Method for Molecular Electrostatics - an Optimal Approach for Large Systems. *J. Comput. Chem* 1995, 16 (7), 898–913.
50. Purisima EO; Nilar SH, A Simple yet Accurate Boundary-Element Method for Continuum Dielectric Calculations. *J. Comput. Chem* 1995, 16 (6), 681–689.
51. Liang J; Subramaniam S, Computation of molecular electrostatics with boundary element methods. *Biophys. J* 1997, 73 (4), 1830–1841. [PubMed: 9336178]

52. Vorobjev YN; Scheraga HA, A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *J. Comput. Chem* 1997, 18 (4), 569–583.
53. Totrov M; Abagyan R, Rapid boundary element solvation electrostatics calculations in folding simulations: Successful folding of a 23-residue peptide. *Biopolymers* 2001, 60 (2), 124–133. [PubMed: 11455546]
54. Boschitsch AH; Fenley MO; Zhou HX, Fast boundary element method for the linear Poisson-Boltzmann equation. *J. Phys. Chem. B* 2002, 106 (10), 2741–2754.
55. Lu BZ; Cheng XL; Huang JF; McCammon JA, Order N algorithm for computation of electrostatic interactions in biomolecular systems. *Proc. Natl. Acad. Sci. U. S. A* 2006, 103 (51), 19314–19319. [PubMed: 17148613]
56. Lu BZ; Cheng XL; Hou TJ; McCammon JA, Calculation of the Maxwell stress tensor and the Poisson-Boltzmann force on a solvated molecular surface using hypersingular boundary integrals. *J Chem. Phys* 2005, 123 (8).
57. Lu BZ; Zhang DQ; McCammon JA, Computation of electrostatic forces between solvated molecules determined by the Poisson-Boltzmann equation using a boundary element method. *J Chem. Phys* 2005, 122 (21).
58. Lu BZ; Cheng XL; Huang JF; McCammon JA, An Adaptive Fast Multipole Boundary Element Method for Poisson-Boltzmann Electrostatics. *J Chem. Theory Comput* 2009, 5 (6), 1692–1699. [PubMed: 19517026]
59. Bajaj C; Chen SC; Rand A, An Efficient Higher-Order Fast Multipole Boundary Element Solution for Poisson-Boltzmann-Based Molecular Electrostatics. *Siam J Sci Comput* 2011, 33 (2), 826–848. [PubMed: 21660123]
60. Davis ME; Mccammon JA, Dielectric Boundary Smoothing in Finite-Difference Solutions of the Poisson Equation - an Approach to Improve Accuracy and Convergence. *J Comput. Chem* 1991, 12 (7), 909–912.
61. LeVeque RJ; Li Z, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal* 1994, 31, 1019–1044.
62. Wang J; Cai Q; Li Z-L; Zhao H-K; Luo R, Achieving energy conservation in Poisson-Boltzmann molecular dynamics: Accuracy and precision with finite-difference algorithms. *Chemical physics letters* 2009, 468 (4–6), 112–118. [PubMed: 20098487]
63. Liu X; Wang C; Wang J; Li Z; Zhao H; Luo R, Exploring a charge-central strategy in the solution of Poisson's equation for biomolecular applications. *Phys. Chem. Chem. Phys* 2013.
64. Wang C; Wang J; Cai Q; Li Z; Zhao H-K; Luo R, Exploring accurate Poisson-Boltzmann methods for biomolecular simulations. *Computational and Theoretical Chemistry* 2013, 1024, 34–44. [PubMed: 24443709]
65. Zhou YC; Zhao S; Feig M; Wei GW, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *J. Comput. Phys* 2006, 213 (1), 1–30.
66. Geng WH; Yu SN; Wei GW, Treatment of charge singularities in implicit solvent models. *J Chem. Phys* 2007, 127 (11).
67. Yu SN; Wei GW, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities. *Journal of Computational Physics* 2007, 227 (1), 602–632.
68. Zhou YC; Feig M; Wei GW, Highly accurate biomolecular electrostatics in continuum dielectric environments. *Journal of Computational Chemistry* 2008, 29, 87–97. [PubMed: 17508411]
69. Ye X; Wang J; Luo R, A revised density function for molecular surface calculation in continuum solvent models. *Journal of chemical theory and computation* 2010, 6 (4), 1157–1169. [PubMed: 24723844]
70. Wang J; Cai Q; Xiang Y; Luo R, Reducing Grid Dependence in Finite-Difference Poisson-Boltzmann Calculations. *Journal of Chemical Theory and Computation* 2012, 8 (8), 2741–2751. [PubMed: 23185142]
71. Botello-Smith WM; Liu XP; Cai Q; Li ZL; Zhao HK; Luo R, Numerical Poisson-Boltzmann model for continuum membrane systems. *Chem. Phys. Lett* 2013, 555, 274–281. [PubMed: 23439886]

72. Dzubiella J; Swanson JMJ; McCammon JA, Coupling nonpolar and polar solvation free energies in implicit solvent models. *J Chem. Phys* 2006, 124 (8).
73. Dzubiella J; Swanson JMJ; McCammon JA, Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Phys. Rev. Lett* 2006, 96 (8).
74. Cheng LT; Dzubiella J; McCammon JA; Li B, Application of the level-set method to the implicit solvation of nonpolar molecules. *J Chem. Phys* 2007, 127 (8).
75. Chen Z; Zhao S; Chun J; Thomas DG; Baker NA; Bates PW; Wei GW, Variational approach for nonpolar solvation analysis. *J Chem. Phys* 2012, 137 (8).
76. Thomas DG; Chun J; Chen Z; Wei GW; Baker NA, Parameterization of a geometric flow implicit solvation model. *J Comput. Chem* 2013, 34 (8), 687–695. [PubMed: 23212974]
77. Xiao L; Cai Q; Li Z; Zhao H; Luo R, A multi-scale method for dynamics simulation in continuum solvent models. I: Finite-difference algorithm for Navier–Stokes equation. *Chemical Physics Letters* 2014, 616–617 (Supplement C), 67–74.
78. Li Z; Xiao L; Cai Q; Zhao H; Luo R, A semi-implicit augmented IIM for Navier–Stokes equations with open, traction, or free boundary conditions. *Journal of Computational Physics* 2015, 297 (Supplement C), 182–193. [PubMed: 27087702]
79. Xiao L; Luo R, Exploring a multi-scale method for molecular simulation in continuum solvent model: Explicit simulation of continuum solvent as an incompressible fluid. *The Journal of Chemical Physics* 2017, 147 (21), 214112. [PubMed: 29221408]
80. Wei H; Luo R; Qi R, An efficient second-order poisson–boltzmann method. *Journal of computational chemistry* 2019.
81. Zhou Y; Zhao S; Feig M; Wei G-W, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. *Journal of Computational Physics* 2006, 213 (1), 1–30.
82. Qi R; Botello-Smith WM; Luo R, Acceleration of Linear Finite-Difference Poisson–Boltzmann Methods on Graphics Processing Units. *Journal of chemical theory and computation* 2017, 13 (7), 3378–3387. [PubMed: 28553983]
83. Colmenares J; Ortiz J; Rocchia W, GPU linear and non-linear Poisson–Boltzmann solver module for DelPhi. *Bioinformatics* 2014, 30 (4), 569–570. [PubMed: 24292939]
84. Qi R; Luo R, Robustness and Efficiency of Poisson–Boltzmann Modeling on Graphics Processing Units. *J. Chem. Inf. Model* 2019, 59 (1), 409–420. [PubMed: 30550277]
85. Buluç A; Fineman JT; Frigo M; Gilbert JR; Leiserson CE In Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks, *Proceedings of the twenty-first annual symposium on Parallelism in algorithms and architectures*, ACM: 2009; pp 233–244.
86. Cornell WD; Cieplak P; Bayly CI; Gould IR; Merz KM; Ferguson DM; Spellmeyer DC; Fox T; Caldwell JW; Kollman PA, A 2ND GENERATION FORCE-FIELD FOR THE SIMULATION OF PROTEINS, NUCLEIC-ACIDS, AND ORGANIC-MOLECULES. *J. Am. Chem. Soc* 1995, 117 (19), 5179–5197.
87. Case DA; Brozell SR; Cerutti DS; Cheatham TE III; Cruzeiro VWD; Darden TA; Duke RE; Ghoreishi D; Gohlke H; Goetz AW; Greene D; Harris R; Homeyer N; Izadi S; Kovalenko A; Lee TS; LeGrand S; Li P; Lin C; Liu J; Luchko T; Luo R; Mermelstein DJ; Merz KM; Miao Y; Monard G; Nguyen H; Omelyan I; Onufriev A; Pan F; Qi R; Roe DR; Roitberg A; Sagui C; Schott-Verdugo S; Shen J; Simmerling CL; Swails J; Walker RC; Wang J; Wei H; Wolf RM; Wu X; Xiao L; York DM; Kollman PA, AMBER 2018. University of California, San Francisco, 2018.
88. Botello-Smith WM; Luo R, Applications of MMPBSA to Membrane Proteins I: Efficient Numerical Solutions of Periodic Poisson–Boltzmann Equation. *J. Chem. Inf. Model.* 2015, 55 (10), 2187–2199. [PubMed: 26389966]
89. Wang J; Cai Q; Li Z-L; Zhao H-K; Luo R, Achieving energy conservation in Poisson-Boltzmann molecular dynamics: Accuracy and precision with finite-difference algorithms. *Chem. Phys. Lett* 2009, 468 (4–6), 112–118. [PubMed: 20098487]
90. Wang CH; Wang J; Cai Q; Li ZL; Zhao HK; Luo R, Exploring accurate Poisson-Boltzmann methods for biomolecular simulations. *Comput. Theor. Chem* 2013, 1024, 34–44. [PubMed: 24443709]

91. Wei H; Luo R; Qi R, An efficient second-order poisson–boltzmann method. *J. Comput. Chem* 2019, 40 (12), 1257–1269. [PubMed: 30776135]
92. Cai Q; Ye X; Wang J; Luo R, On-the-Fly Numerical Surface Integration for Finite-Difference Poisson-Boltzmann Methods. *J. Chem. Theory Comput* 2011, 7 (11), 3608–3619. [PubMed: 24772042]
93. Cai Q; Ye X; Luo R, Dielectric pressure in continuum electrostatic solvation of biomolecules. *Physical Chemistry Chemical Physics* 2012, 14 (45), 15917–15925. [PubMed: 23093365]
94. Xiao L; Cai Q; Ye X; Wang J; Luo R, Electrostatic forces in the Poisson-Boltzmann systems. *Journal of Chemical Physics* 2013, 139 (9).
95. Xiao L; Wang CH; Ye X; Luo R, Charge Central Interpretation of the Full Nonlinear PB Equation: Implications for Accurate and Scalable Modeling of Solvation Interactions. *Journal of Physical Chemistry B* 2016, 120 (33), 8707–8721.
96. Wang CH; Xiao L; Luo R, Numerical interpretation of molecular surface field in dielectric modeling of solvation. *Journal of Computational Chemistry* 2017, 38 (14), 1057–1070. [PubMed: 28318096]
97. Nvidia NVIDIA CUDA Sparse Matrix library. <https://developer.nvidia.com/cuspars> (accessed October 1st, 2016).
98. Wang C; Xiao L; Luo R, Numerical interpretation of molecular surface field in dielectric modeling of solvation. *Journal of computational chemistry* 2017, 38 (14), 1057–1070. [PubMed: 28318096]
99. Li Z; Ito K, *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*. Siam: 2006; Vol. 33.

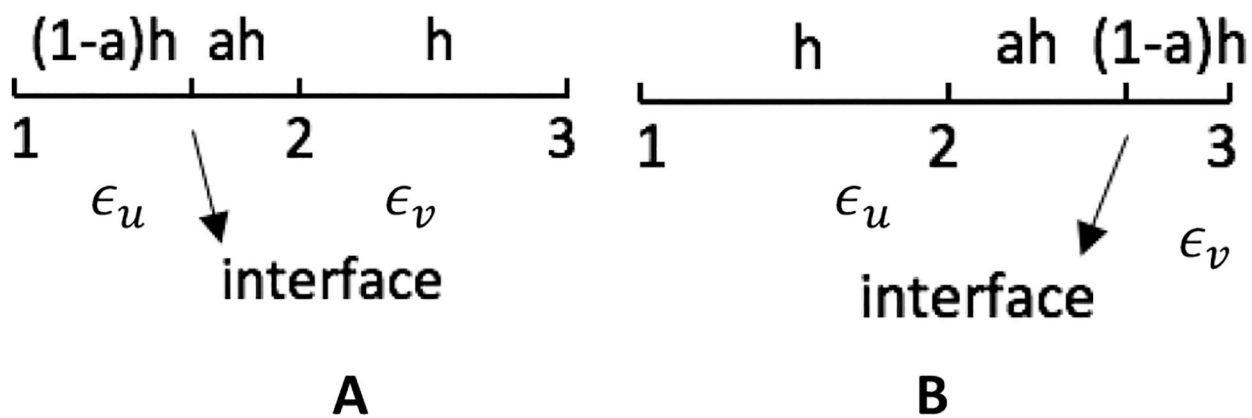
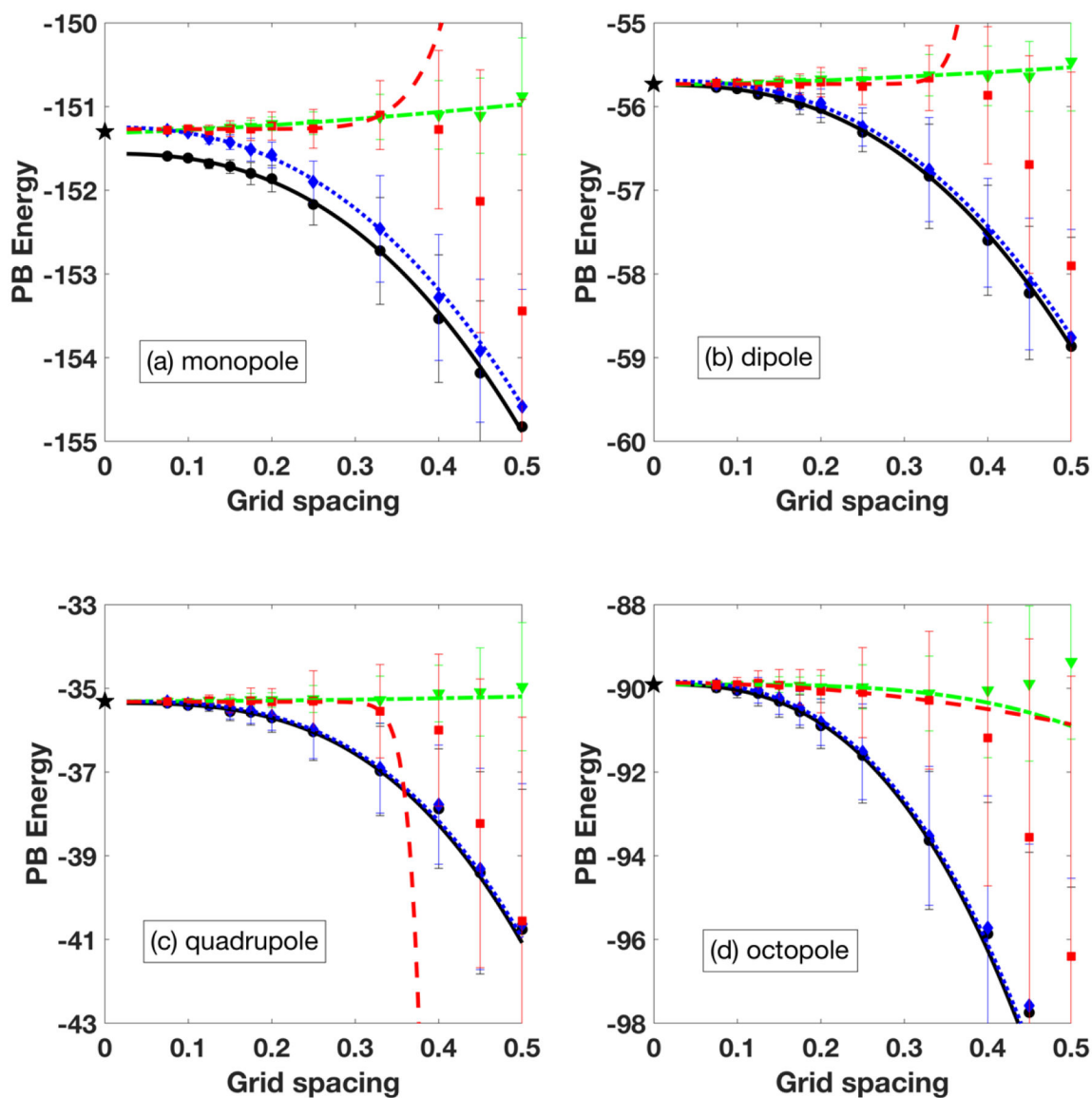


Figure 1. Two types of interface geometries. The left side of the interface is considered the solute region with the dielectric constant ϵ_u , and the right side the solvent region with the dielectric constant ϵ_v . h is the grid spacing and $0 < a < 1$.



— HA method - - - 2nd-order HA method ... χ -factor HA method - · - IIM method

Figure 2.

PB energies (kcal/mol) versus grid spacing (Å) for analytical sphere models. Energy results were obtained by averaging energy values from 30 runs of different grid orientations/offsets of the molecules, and error bars denote standard deviations of the 30 sampled energies. All trend lines were obtained by fitting data to functions in the form of $y=a*x^b+c$. Only data points at grid spacing 0.33 Å were included in the fitting. The stars on the y axes denote the analytical values.

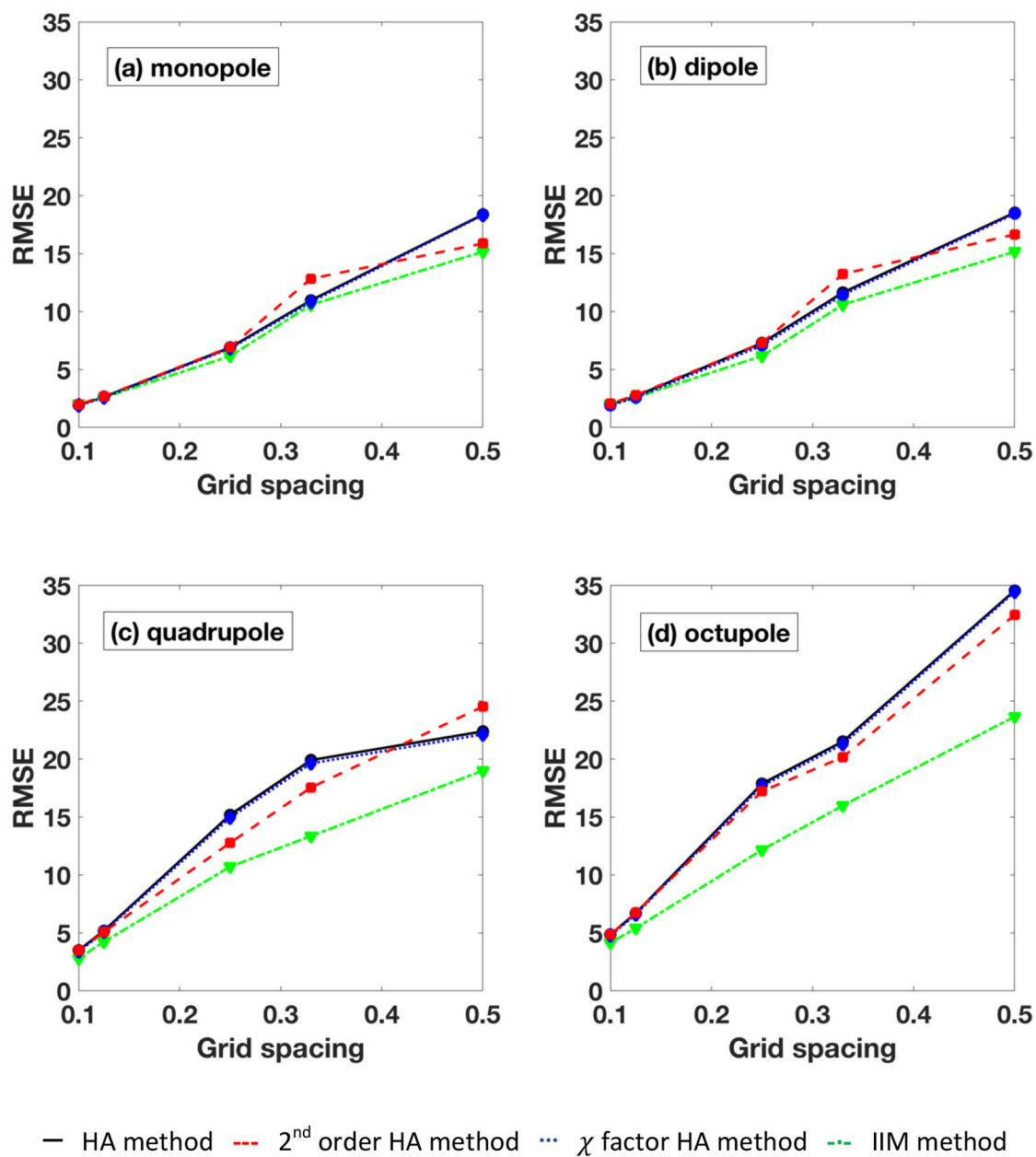


Figure 3.

RMSEs of normal surface electric fields between the computed and the analytical values (kcal/mol-e-Å) versus grid spacing (Å). The surface fields were obtained by one-dimensional interpolation.

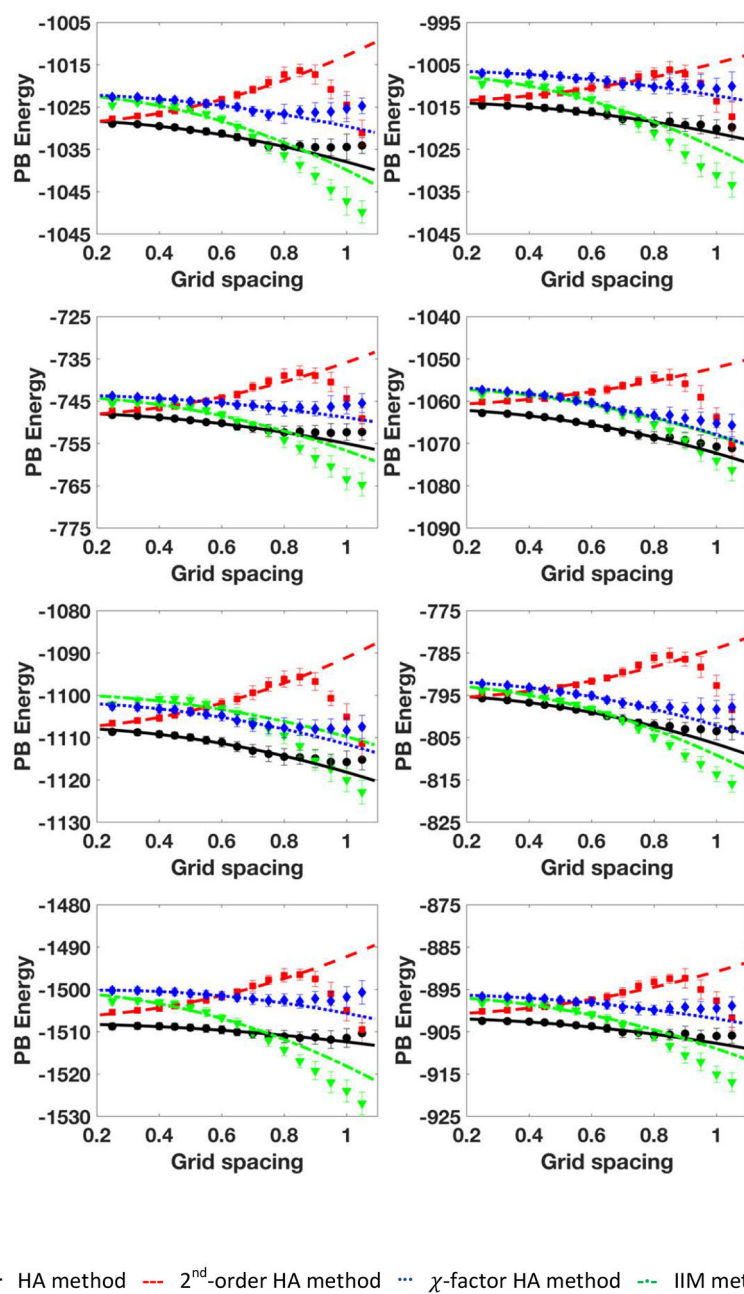


Figure 4.

PB energies (kcal/mol) versus grid spacing (\AA) for the 8 selected protein molecules. Energy results are obtained by averaging energy values from 30 runs of different grid orientations/offsets of the molecules, and error bars denote standard deviations of the 30 sampled energies. All trend lines were obtained by fitting data to functions in the form of $y=a*x^b+c$. Only data points at grid spacing 0.5 \AA were included in the fitting.

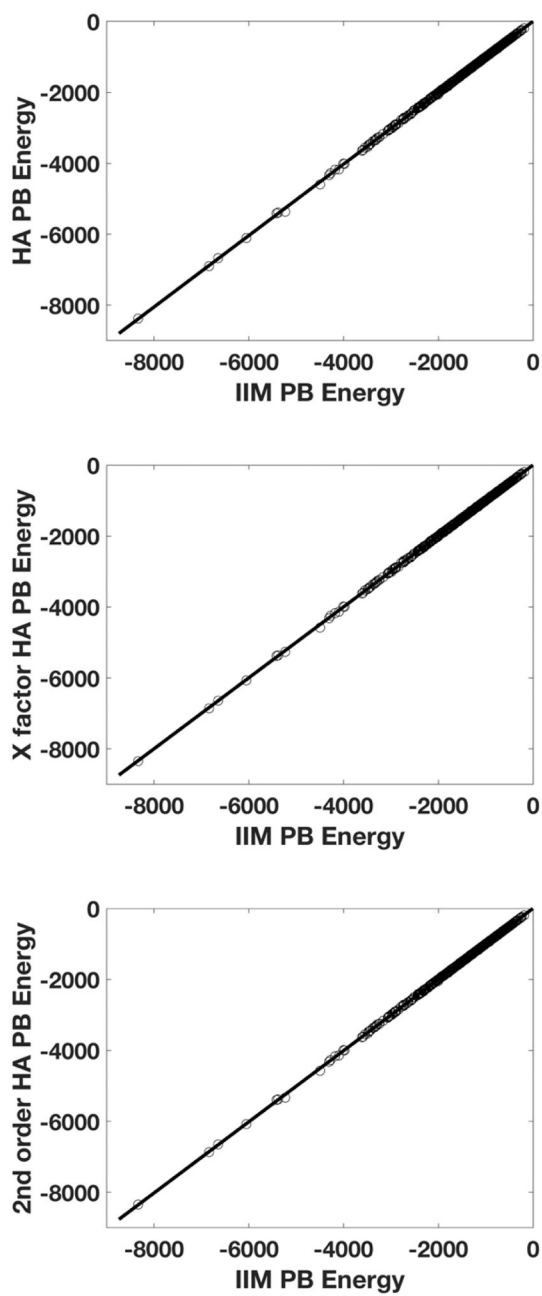


Figure 5. Tests of PB energies (kcal/mol) at 0.5 Å for 573 biomolecules. The data were fitted linearly and the intercepts were set to zero.

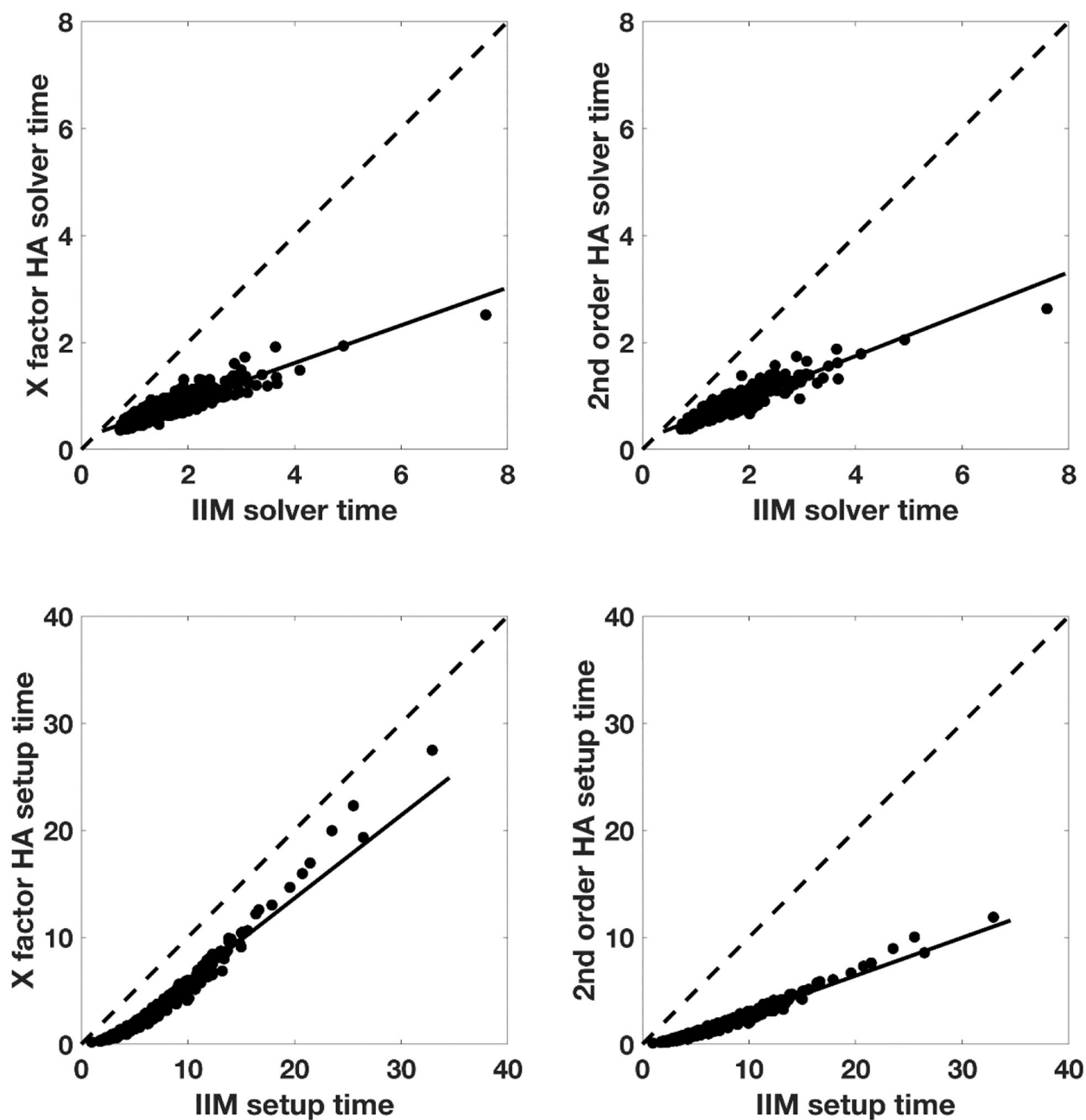


Figure 6.

Timing (seconds) tests of GPU implementation at 0.5 \AA for 573 biomolecules. The data were fitted linearly. The fitting line of the χ factor HA method solver time is $y=0.3526x+0.2069$, and that of the second-order HA method is $y=0.3917x+0.1819$. The fitting line of the χ factor HA method setup time is $y=0.7397x-2.405$, and that of the second-order HA method is $y=0.3355x+0.1703$. The dashed lines are diagonal.

Table 1.

Cartesian coordinates of charges for the analytical monopole, dipole, quadrupole, and octupole models.

Model	Positive Charge Coordinates	Negative Charge Coordinates
Monopole	(-0.10, -0.10, 0.90)	
Dipole	(-0.10, -0.10, 0.90)	(-0.10, -0.10, -0.90)
Quadrupole	(0.90, -0.90, -0.10)	(0.90, 0.90, -0.10)
	(-0.10, 0.90, -0.10)	(-0.10, -0.90, -0.10)
Octupole	(0.90, -0.10, 0.90)	(0.90, 0.90, 0.90)
	(-0.10, 0.90, 0.90)	(-0.10, -0.10, 0.90)
	(0.90, -0.10, -0.10)	(0.90, 0.90, -0.10)
	(-0.10, 0.90, -0.10)	(-0.10, -0.10, -0.10)

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 2.

Extrapolated and analytical PB energies (kcal/mol) for the analytical sphere models. Extrapolated values are from Figure 2. Analytical PB energies were calculated using Mathematica 7.0.

Methods	HA	2 nd order HA	χ factor HA	IIM	Analytical
Monopole	-151.6	-151.3	-151.2	-151.3	-151.3
Dipole	-55.74	-55.74	-55.69	-55.73	-55.73
Quadrupole	-35.35	-35.31	-35.32	-35.32	-35.31
Octupole	-89.91	-89.91	-89.84	-89.87	-89.91

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 3.

Extrapolated PB energies (kcal/mol, at the limiting grid spacing of 0 Å) for the 8 selected protein molecules from Figure 4.

Methods	HA	2 nd order HA	χ factor HA	IIM
1aci	-1029	-1028	-1025	-1024
1ah9	-1015	-1013	-1010	-1009
1b22	-747.9	-747.5	-745.5	-745.4
1aw0	-1063	-1060	-1059	-1058
1bbi	-1108	-1107	-1105	-1101
1bdc	-795.2	-794.9	-793.6	-793.8
1bw6	-1509	-1506	-1503	-1503
1c75	-902.4	-900.4	-898.4	-898.2

Table 4.

PB energies (kcal/mol) and standard deviations (kcal/mol in parenthesis) for the 8 selected protein molecules at the grid spacing of 0.5 Å. The average energies and standard deviations were obtained from 30 runs of different grid orientations/offsets of the molecules.

Methods	HA	2 nd order HA	χ factor HA	IIM
1aci	-1030.38(0.73)	-1025.26(0.69)	-1026.63(0.76)	-1025.37(0.88)
1ah9	-1015.32(0.94)	-1011.52(0.89)	-1010.66(0.93)	-1010.84(1.1)
1b22	-749.49(0.61)	-745.64(0.64)	-746.93(0.65)	-746.32(0.71)
1aw0	-1064.12(0.59)	-1058.73(0.60)	-1060.93(0.59)	-1059.01(0.55)
1bbi	-1109.95(0.90)	-1103.83(0.85)	-1106.20(0.98)	-1101.10(1.2)
1bdc	-797.71(0.71)	-793.18(0.67)	-795.81(0.70)	-796.21(0.60)
1bw6	-1509.01(0.83)	-1503.28(0.81)	-1503.83(0.78)	-1504.29(0.90)
1c75	-903.08(0.70)	-898.59(0.72)	-899.27(0.69)	-899.27(0.73)