UC Irvine

UC Irvine Electronic Theses and Dissertations

Title

An Efficient Second-Order Poisson-Boltzmann Method

Permalink

https://escholarship.org/uc/item/7dx3938q

Author

Wei, Haixin

Publication Date

2019

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA,
IRVINE

An Efficient Second-Order Poisson-Boltzmann Method

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF MATERIALS SCIENCE AND ENGINEERING

by

Haixin Wei

Thesis Committee:
Professor Ray Luo, Chair
Professor Long Chen
Assistant Professor Han Li

2019

© 2019 Haixin Wei

DEDICATION

To

my parents and friends

in recognition of their worth

thanks to my advisor, Prof. Ray Luo

, who is always supportive.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express the deepest appreciation to my committee chair, Professor Ray Luo, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and scholarship, and an excitement in regard to teaching. Without his guidance and persistent help this dissertation would not have been possible.

I would like to thank my committee members, Professor Long Chen and Professor Han Li, whose work demonstrated to me that our work of molecular modeling has an extensive influence on various scientific practices.

In addition, a thank you to Professor Zejun Ding of University of Science and Technology of China, who introduced me to computational physics, and whose enthusiasm had lasting effect.

Thanks for the financial support that was provided by the University of California, Irvine, National Institute of Health/NIGMS (GM093040 & GM079383).

ABSTRACT OF THE THESIS

An Efficient Second-Order Poisson-Boltzmann Method

By

Haixin Wei

Master of Materials Science and Engineering

University of California, Irvine, 2019

Professor Ray, Chair

Immersed interface method (IIM) is a promising high-accuracy numerical scheme for the Poisson-Boltzmann model that has been widely used to study electrostatic interactions in biomolecules. However, the IIM suffers from instability and slow convergence for typical applications. In this study, we introduced both analytical interface and surface regulation into IIM to address these issues. The analytical interface setup leads to better accuracy and its convergence closely follows a quadratic manner as predicted by theory. The surface regulation further speeds up the convergence for nontrivial biomolecules. In addition, uncertainties of the numerical energies for tested systems are also reduced by about half. More interestingly, the analytical setup significantly improves the linear solver efficiency and stability by generating more precise and better-conditioned linear systems. Finally, we implemented the bottleneck linear system solver on GPUs to further improve the efficiency of the method, so it can be widely used for practical biomolecular applications.

INTRODUCTION

Electrostatic interactions play crucial roles in biophysical processes such as protein-protein and protein-ligand interactions. Accurate and efficient treatment of electrostatics is thus vital in computational analyses of biomolecular structures and dynamics. A closely related issue is the modeling of water molecules and their electrostatic interactions with biomolecules that must be considered for any realistic representation of biomolecules at physiological conditions. Since most particles in explicit molecular model are water molecules that solvate the target biomolecules, treating these water molecules implicitly would allow higher computational efficiency without losing any atomic-level resolution of the biomolecules. The main idea of implicit solvation treatment is to model water molecules as continuous medium while still treating the biomolecular solutes in atomic detail. In this model, the solute molecule is treated as a low dielectric constant region with a number of point charges located at atomic centers, and the solvent is treated as a high dielectric constant region. Poisson-Boltzmann equation (PBE)-based implicit solvent model has been one such attempt and been widely used in biomolecular applications.

To solve PBE, numerical solutions are almost always needed for biomolecular applications, since analytic solution of the PBE can be achieved only in a few specific cases with simple solute geometry. Among the numerical solution methods, finite-difference methods (FDM),[1-16] finite-element methods[17-26] and boundary-element methods[27-44] are mostly used. Or alternatively, semi-analytical generalized Born approaches, particularly in MD, were also explored.[7, 9, 11, 12, 45-56] Within numerical methods, FDM has the advantage of being straightforward and physically transparent in its discretization. However, if a direct

discretization is used without considering the discontinuity in the dielectric constant, the numerical solutions tend to have large errors, and the errors are particularly obvious near the solute-solvent interface. To overcome this problem, Davis and McCammon proposed a harmonic average (HA) method to approach the approximate dielectric constant near surface in 1991.[57] Efforts have also been reported recently by Wei and co-workers and Li and co-workers to develop higher accuracy interface schemes, the immersed interface method (IIM) and the matched interface and boundary (MIB) method, to improve numerical accuracy of the PBE solution[58-66] Additional higher-order schemes are also developed,[67, 68] some of which are specifically for nonlinear PBE. The idea of IIM is to enforce the interface conditions into the finite-difference schemes at grid points near the interface. On the other hand, the scheme of MIB is enforcing the lowest-order jump condition repeatedly to achieve the high-order jump condition. Alternatively, instead of treating the solute-solvent interface explicitly, Alexov and co-workers proposed an approach that uses Gaussian-based smooth dielectric functions to model the implicit solvation environment in an interface-free manner.[69, 70] Some other interesting approaches have also been proposed to improve implicit solvent models, such as coupling electrostatic and nonelectrostatic interactions within the implicit solvation treatment,[71-75] explicit simulating implicit solvent as a fluid for the purpose of more physical modeling of solvation interactions,[76-78] and using the level set function to better define the solvent and solute interfaces for dielectric assignment.[79, 80] Among those methods, IIM has been a promising high-accuracy numerical scheme, and is able to achieve energy conservation in PB molecular dynamics (MD).[81] This can be attributed, in part, to the fact that a uniform higher accuracy of $O(h^2)$ can be achieved even near the interface.[82]

However, the original IIM has also suffered from numerical instability in complex biomolecular environments.[59] This in part is due to the use of the level set scheme to track the molecular interface and compute associated jump conditions. Here the molecular interface is defined with an analytical density function recently developed by our group.[83] It has been shown that the density function method performs better than geometry-based methods such as van der Waals surface and solvent exclude surface in terms of errors, transferability, conformation dependence, and convergence.[83] The extensive use of the level set scheme in the original IIM, however, was found to introduce noises in interface properties that leads to numerical instability. For example, it uses a least square interpolation approach to obtain all interface parameters and jump conditions.[82] Though this is a general scheme for partial differential equations with interface, it may not always work on the molecular interface formed by densely packed atoms. In addition to the instability issue, the original IIM is also very slow and does not work well even when coarse grids are used in biomolecular applications.[59] This can be attributed to the extensive use of finite-difference interface approximations that may also fail to capture the interface characteristics at coarse grid spacings and lead to inconsistent linear systems that are poorly conditioned and take longer time to solve.

In this study, we present a new implementation of the second-order accuracy IIM to address those issues. Specifically, we show how to calculate geometric properties and jump conditions analytically at the interface. In addition, the necessity and rationality of surface regulation are discussed, along with a simple and efficient regulation scheme. We also present a GPU implementation of the linear solver needed in the IIM, a bottom neck of the IIM. Our analysis shows that the new strategy improves the convergence and stability of

IIM, and also improves the efficiency of the method. The GPU implementation further removes the bottle neck of numerical procedure. These new developments have been implemented into the Amber molecular modeling suite and are freely available for the biomedical community.[84]

CHAPTER 1

METHODS

1.1 Immersed Interface Method

IIM is a more accurate method for interface treatment than most traditional ones, such as the harmonic average method.[59] In IIM, the interface can be used to separate the problem domain into inside (Ω^-), outside (Ω^+), and interface (Γ). Next finite-difference grid points are classified as regular grid points that are with all their neighboring grid points (connected with grid edges) in the same region or as irregular points that are not.

Next the jump conditions can be predefined as follows so the PDE is well-posted,

$$[\phi]_\Gamma = w,$$

$$[\epsilon\phi_n]_\Gamma = v \qquad (1)$$

where w and v are given by users for specific problems at hand. In this study w and v are defined for the singularity-free PBE in section 1.2.

For regular points, IIM uses the standard 7-point central finite-difference scheme, since those points do not have a neighbor located in a different region. This stencil has an accuracy of ($O(h^2)$). For irregular points, IIM proposes a new finite-difference stencil that contains at least 10 points instead of 7 (typically 27 points) are used to minimize the magnitude of the local truncation error while satisfying the jump conditions (Eqn 1). So the finite-difference scheme becomes,

$$\sum_m^{n_s} \gamma_m \phi(i+i_m, j+j_m, k+k_m) = f(i,j,k) + C(i,j,k) \qquad (2)$$

where n_s is the number of grid points (27 typically), γ_m are unknown coefficients, C(i,j,k) is the unknown correction term, and f(i,j,k) is the discretized right hand term of the

PBE. In this study, it is always zero because the singularity-free PBE is used as discussed in section 1.2.

The basic idea of IIM is to determine γ_m in Eqn (2) at the irregular points so that the second-order global accuracy is obtained as in an interface-free problem with the finite-difference/finite-volume discretization scheme. Specifically, IIM uses the interface relations (see section 1.5) to translate all the outside points into inside (or vice versa) and then forces the truncation error of the inside (or outside) to O(h^3).[82]

The local truncation error T(i,j,k) at grid point (i,j,k) is,

$$T(i,j,k) = \sum_m^{n_s} \gamma_m \phi(i+i_m, j+j_m, k+k_m) - C(i,j,k). \qquad (3)$$

after setting f(i,j,k)= 0. In the local coordinate frame, the Taylor expansion can be used to expand each ϕ in the neighborhood of the projection point (X^*) of the grid point (i,j,k) from each side of the interface to the second order. Thus there are 20 terms, corresponding to $\phi(X^*)^{\pm}$, $\phi_\xi(X^*)^{\pm}$, $\phi_\eta(X^*)^{\pm}$, $\phi_\tau(X^*)^{\pm}$, $\phi_{\xi\xi}(X^*)^{\pm}$, $\phi_{\xi\eta}(X^*)^{\pm}$, $\phi_{\xi\tau}(X^*)^{\pm}$, $\phi_{\eta\eta}(X^*)^{\pm}$, $\phi_{\eta\tau}(X^*)^{\pm}$, $\phi_{\tau\tau}(X^*)^{\pm}$. Here ξ, η and τ are local coordinates, and "+", "–" denote the outside and inside region, respectively. By using the interface relations as shown in the section 1.5,[82] those 20 terms are reduced to 10, all in the inside (or outside) region as shown below.

$$T(i,j,k) = a_1\phi^- + a_2\phi_\xi^- + a_3\phi_\eta^- + a_4\phi_\tau^- + a_5\phi_{\xi\xi}^- + a_6\phi_{\xi\eta}^- + a_7\phi_{\xi\tau}^- + a_8\phi_{\eta\eta}^- + a_9\phi_{\eta\tau}^- + a_{10}\phi_{\tau\tau}^- + T'(i,j,k) - C(i,j,k) + O(\max\{|\gamma_m|h^3\}), \qquad (4)$$

where $T'(i,j,k)$ is a constant term from imposing the interface relations.

Finally, the expansion is matched against the differential equation to the leading terms to obtain a system of equations for the finite-difference coefficients. For the Poisson equation, we have $a_1 = 0, a_2 = 0, a_3 = 0, a_4 = 0, a_5 = 1, a_6 = 0, a_7 = 0, a_8 = 1, a_9 = 0,$

$a_{10} = 1$. Then the constant term $C(i,j,k)$ can be determined with the accuracy of $O(h^2)$. With 10 equations and 27 unknowns (γ_m), the problem is in principle solvable.[82]

Furthermore, it appears that with 17 extra degrees of freedom, special γ_m can be chosen to fulfill some optimization requirement such as the maximum principle preserving scheme, to achieve even better performance.[82] However, this may not always be the case, which will be discussed latter.

1.2 Removal of Charge Singularity

Point charge models are widely used in molecular simulations of biomolecules. However, the representation of point charges by delta functions introduces singularity to the PDE. Several strategies are available to remove the charge singularity,[21, 63, 85, 86] and here we adopt the recently developed reaction field potential method.[59]

Briefly this method solves the PDE in two different regions with different potentials, i.e. the reaction field potential inside and the total field potential outside. Here the reaction field potential is the potential caused only by induced charges.[87]

Thus, the unified equation describing the whole region is,

$$\nabla \cdot \varepsilon \nabla \phi = 0 \qquad (5)$$

but with the modified jump conditions as follows,

$$[\phi] = \sum_{charges} \frac{q}{r} = w$$

$$[\varepsilon \phi_n] = \varepsilon_{inside} \left(\sum_{charges} \frac{q}{r}\right)_n = v \qquad (6)$$

Here, the subscript n means the gradient is along the interface normal direction n.[59]

2.3 Density Function Strategy

To obtain the interface relations and the jump conditions, it is necessary to know the geometry of the interface and the jump condition values on the interface, in addition to the tangential derivatives of the jump conditions. These quantities can be obtained analytically if the density function strategy is used to define the interface.[82]

The main idea of the density function method is to use a smooth function to approximate the solute exclusive surface (SES) of a molecule, and in the meantime adjust the parameters of the density function accordingly so that the reaction field energies obtained from these two models agree the best.[83] Specifically, given the n^{th} atom centered at $\boldsymbol{r}_n$, its density function ρ_n is defined as,

$$\rho_n(x) = \rho_n(\frac{d-r_c}{2r_p}). \qquad (7)$$

Here, $d=|\boldsymbol{r}-\boldsymbol{r}_n|$ is the distance to the atom center, r_c is the VDW radius of the atom, and r_p is the solvent probe radius.

In addition, the density function satisfies the following constraints to be physical and reasonable,[83]

$$\rho_n(x) > 1, when\ x < 0$$

$$\rho_n(x) = 1, when\ x = 0\ .$$

$$\rho_n(x) < 1, when\ x > 0 \qquad (8)$$

Here points with x=0 are on the surface of an atom. To guarantee smoothness and good numerical behaviors, a cubic-spline interpolation function that fulfills the above constraints is used. Next, under the requirement of achieving the best agreement of reaction field energies with the given benchmark for a diversified set of training molecules, all the coefficients in the interpolation are optimized and the following formula is obtained.[83]

$$\rho_n(x) = \begin{cases} 1.00 - 4.527143x - 3.640532x^2 + 32.631235x^3 & 0.0 < \mathrm{x} < 0.2 \\ 0.21 - 2.067608x + 15.938209x^2 - 35.500854x^3 & 0.2 < \mathrm{x} < 0.4 \\ 0.15 + 0.047573x - 5.362303x^2 + 13.12218x^3 & 0.4 < \mathrm{x} < 0.6 \\ 0.05 - 0.522686x + 2.511005x^2 - 4.487867x^3 & 0.6 < \mathrm{x} < 0.8 \\ 0.01 - 0.056828x - 0.181716x^2 + 1.079289x^3 & 0.8 < \mathrm{x} < 1.0 \end{cases} \quad (9)$$

Given the atomic density function, the molecular density function is a summation of all the atomic density functions as

$$\rho(x) = \sum_n^{atom} \rho_n(x_n) \quad (10)$$

More details can be found in the literature.[83]

Note that in the original IIM, all the geometric and physical interface properties are obtained via the level-set numerical toolkit.[82] However, with the use of the density function, we can obtain all these quantities analytically, including the interface normal direction $\boldsymbol{\xi}$, interface curvatures $\xi_{\eta\eta}$, $\xi_{\tau\tau}$ and $\xi_{\eta\tau}$, jump condition values w and v, and their tangential derivatives on the interface. More importantly, by using this method, the numerical instability arising from applying hard spheres in classical molecular surface representation can also be minimized due to the use of a smoothly varying function.[83]

2.4 Analytical Interface Relations and Jump Conditions

A key improvement in the new analytical IIM is the complete elimination of the level set functions used to interpolate all interface properties. This is a more natural strategy because the underlining density function to represent the interface is a continuous cubic-spline function. In addition, the jump conditions, i.e. Coulomb potential and field (Eqn (6)), are also analytically available.

There are roughly three steps to determine all the analytical interface conditions on the interface.[82] The first step is to find the projection point of an irregular grid point. The projection point here means a point right on the interface that is close to the irregular point. Recall Eqns (3) and (4), this is necessary for the Taylor expansion used to compute the local truncation error. The best choice of projection point is the interface point that is closest to the irregular point. However, it is almost impossible to locate it, and even if we can, it is definitely not worth all the effort.[82] Instead, we used its normal projection point on the interface. The normal projection point can be easily obtained by finding the intersection point of the interface and the steepest descent vector starting from the irregular point (i.e. along the gradient of the density function). Because we are looking along the steepest descent direction, the projection point may also approximate the closest projection point reasonably well. Specifically, the Newton's method was used to search along the normal direction for the projection point. One of the advantages of this method is that we can locate the projection point as precise as possible and also relatively efficiently. Specifically, the precision was set to 10^{-6} in our method so that its location would never be an issue in most finite-difference setups, which means,

$$|\rho(projection\ point) - 1| \leq 10^{-6}. \qquad (11)$$

The second step is to calculate the local coordinate transformation matrix and the interface curvature. This is straightforward given the analytical density function. These quantities will be used in Eqns (4) and (6). To calculate the transformation matrix, we need the normal direction of the interface, which is the gradient direction of the density function of the projection point. Next, the other two tangential directions can be determined as follows,[82]

$$\begin{cases} \boldsymbol{\xi} = \left(\rho_x, \rho_y, \rho_z\right)^T \\ \boldsymbol{\eta} = (\rho_y, -\rho_x, 0)^T \\ \boldsymbol{\tau} = \left(\rho_x\rho_z, \rho_y\rho_z, -\rho_x^2 - \rho_y^2\right)^T \quad , \quad if\ \rho_y^2 \geq \rho_z^2 \\ \boldsymbol{\xi} = \left(\rho_x, \rho_y, \rho_z\right)^T \\ \boldsymbol{\eta} = (\rho_z, 0, -\rho_x)^T \\ \boldsymbol{\tau} = (-\rho_x\rho_y, \rho_x^2 + \rho_y^2, -\rho_y\rho_z)^T \quad , \quad otherwise \end{cases} \tag{12}$$

The three principal curvatures can then be obtained as follows. Under the local coordinates of the interface, the following relations hold,

$$\rho_{\boldsymbol{\eta}}(0,0,0) = \rho_{\boldsymbol{\tau}}(0,0,0) = 0 \tag{13}$$

Based on Eqn (13), the three curvatures are,

$$\xi_{\eta\eta} = -\frac{\rho_{\boldsymbol{\eta\eta}}}{\rho_{\boldsymbol{\xi}}}$$

$$\xi_{\tau\tau} = -\frac{\rho_{\boldsymbol{\tau\tau}}}{\rho_{\boldsymbol{\xi}}}$$

$$\xi_{\eta\tau} = -\frac{\rho_{\boldsymbol{\eta\tau}}}{\rho_{\boldsymbol{\xi}}} \tag{14}$$

The last step is to calculate all those jump conditions. Given Eqn (14), we obtained the following equations of derivatives of Eqn (6).

$$w_\eta = \sum_i \frac{q_i}{{r_i}^3}\eta_i$$

$$w_\tau = \sum_i \frac{q_i}{{r_i}^3}\tau_i$$

$$w_{\eta\eta} = \sum_i 3\frac{q_i}{{r_i}^5}{\eta_i}^2 - \frac{q_i}{{r_i}^3} + \frac{q_i}{{r_i}^3}\xi_i \cdot (\xi_{\eta\eta})_i$$

$$w_{\tau\tau} = \sum_i 3\frac{q_i}{{r_i}^5}{\tau_i}^2 - \frac{q_i}{{r_i}^3} + \frac{q_i}{{r_i}^3}\xi_i \cdot (\xi_{\tau\tau})_i$$

$$w_{\eta\tau} = \sum_i 3\frac{q_i}{{r_i}^5}\eta_i\tau_i + \frac{q_i}{{r_i}^3}\xi_i \cdot (\xi_{\eta\tau})_i$$

$$v_\eta = \sum_i 3\frac{q_i}{{r_i}^5}\xi_i\eta_i$$

$$v_\tau = \sum_i 3\frac{q_i}{r_i^5}\xi_i\tau_i \qquad (15)$$

The summation here is over all the charges or atoms. Now all the quantities needed are obtained, the interface relations can be constructed as,

$$\phi^+ = \phi^- + w$$

$$\phi_\xi^+ = \frac{\varepsilon^-}{\varepsilon^+}\phi_\xi^- + \frac{v}{\varepsilon^+}$$

$$\phi_\eta^+ = \phi_\eta^- + w_\eta$$

$$\phi_\tau^+ = \phi_\tau^- + w_\tau$$

$$\phi_{\xi\xi}^+ = \frac{\varepsilon^-}{\varepsilon^+}\phi_{\xi\xi}^- + \left(\frac{\varepsilon^-}{\varepsilon^+} - 1\right)\phi_{\eta\eta}^- + \left(\frac{\varepsilon^-}{\varepsilon^+} - 1\right)\phi_{\tau\tau}^- + \left(\phi_\xi^+ - \phi_\xi^-\right)\left(\xi_{\eta\eta} + \xi_{\tau\tau}\right) - w_{\eta\eta} - w_{\tau\tau}$$

$$\phi_{\xi\eta}^+ = \frac{\varepsilon^-}{\varepsilon^+}\phi_{\xi\eta}^- + \left(\phi_\eta^+ - \frac{\varepsilon^-}{\varepsilon^+}\phi_\eta^-\right)\xi_{\eta\eta} + \left(\phi_\tau^+ - \frac{\varepsilon^-}{\varepsilon^+}\phi_\tau^-\right)\xi_{\eta\tau} + \frac{v_\eta}{\varepsilon^+}$$

$$\phi_{\xi\tau}^+ = \frac{\varepsilon^-}{\varepsilon^+}\phi_{\xi\tau}^- + \left(\phi_\eta^+ - \frac{\varepsilon^-}{\varepsilon^+}\phi_\eta^-\right)\xi_{\eta\tau} + \left(\phi_\tau^+ - \frac{\varepsilon^-}{\varepsilon^+}\phi_\tau^-\right)\xi_{\tau\tau} + \frac{v_\tau}{\varepsilon^+}$$

$$\phi_{\eta\eta}^+ = \phi_{\eta\eta}^- + \left(\phi_\xi^- - \phi_\xi^+\right)\xi_{\eta\eta} + w_{\eta\eta}$$

$$\phi_{\tau\tau}^+ = \phi_{\tau\tau}^- + \left(\phi_\xi^- - \phi_\xi^+\right)\xi_{\tau\tau} + w_{\tau\tau}$$

$$\phi_{\eta\tau}^+ = \phi_{\eta\tau}^- + \left(\phi_\xi^- - \phi_\xi^+\right)\xi_{\eta\tau} + w_{\eta\tau} \qquad (16)$$

2.5 Surface Regulation

Even if the density function surface was designed to handle the deeply buried atoms as the standard SES approach, it is still limited in approximating the solvent reentry interface formed among solvent accessible atoms.[83] For example, the reentry interface is often flatter than what the density function predicts, as shown in Figure 1. In addition, it is almost impossible to model the reentry surface as being formed by a spherical probe as in

SES. We thus resort to a surface regulation scheme to adjust the interface geometric parameters to partially alleviate the limitations.

Our regulation scheme follows two principles. First, we exploited the original SES idea that the molecular interface is simply a surface formed by probing the molecules of hard spheres of different radii. Thus, the interface is a union of solvent-exposed spheres of different radii. Second, the inward-facing portion (aka solvent reentry portion) of the interface cannot have a surface curvature higher than that of solvent probe. Thus, a regulation scheme was developed with the following two steps for the solvent reentry portion, i.e. when the surface curvature is negative. First, if the curvature of the surface is larger than $1/r_{\mathrm{p}}$, it is reset to $1/r_{\mathrm{p}}$, where r_{p} is the solvent probe radius (Figure 1). Second, the mixing curvature $\xi_{\eta\tau}$ is set to be zero, and the other two curvatures is set to equal to whichever has a smaller absolute value. This step implies the use of a sphere to represent the reentry surface. These changes were found to improve the numerical stability and convergence of the IIM calculations, especially for coarse grid situations, as discussed in later chapter, Results and Discussion.

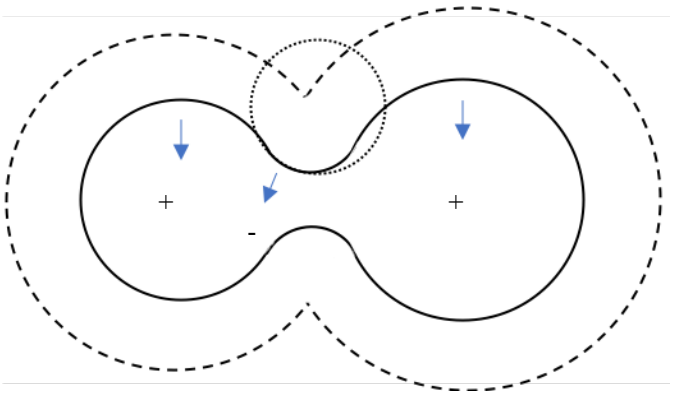


Figure 1. Treatment of the reentry surface. The solid line represents the solvent excluded surface. The dash line represents the solvent accessible surface. The dot circle

represents the solvent probe. “+” signs represent surface elements with positive curvature and “-” signs represent surface elements with negative curvature.

2.6 Overview of the Numerical Procedure

Apparently, the whole process of solving PBE with IIM involves many procedures. Here we outline the main numerical procedures step by step.

1) Use the molecular density function to classify grid points into regular and irregular points;

2) For regular points, set up the discretized equation with the standard seven-point stencil;

3) For irregular points, calculate all the needed interface geometry parameters (e.g. curvatures) using the density function;

4) Apply the regulation scheme to revise the obtained interface geometry parameters;

5) For irregular points, continue calculating the needed jump conditions, i.e. w and v, analytically using the density function and the geometry parameters just obtained;

6) Set up the discretized equation for irregular points using the obtained geometry parameters and jump conditions;

7) Combine the equations of both regular and irregular points and solve the full linear system to obtain the potentials;

In the above procedures, steps 3), 4), and 5) were completely rewritten in our analytical implementation of IIM. Step 7 was also rewritten for both CPU and GPU calculations.

CHAPTER 2

ANALYTICAL TEST CASES AND COMPUTATION DETAILS

To validate the accuracy and precision of the proposed method, we first used a well-studied analytical model, i.e. a single dielectric sphere imbedded with point charges. The analytical potential in the inside region is,

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

$$Q_{lm} = \sum_{k=1}^{N_q} q_k r_k^{l} Y_{lm}^{*}(\theta_k,\varphi_k) \qquad (17)$$

where R is the radius of the sphere, and the center is set to the origin. N_q is the number of charges, Y_{lm} are the spherical harmonics as spherical coordinates are used.

After solving the finite-difference equations, only potentials at grid points are known. To obtain potential at any position (x,y,z), we utilized the one-side least-square interpolation method.[59] Next the reaction field energy is the summation of the products of the reaction field potential and the charges.

$$\Delta G = \frac{1}{2}\sum_{charges} q_i \phi_{RF} \qquad (18)$$

For the analytical test, four off-centered charged models were used, which are monopole, dipole, quadrupole, and octupole. The radius of sphere was set to 2 Å for convenience, which is about the size of a united carbon atom. The Cartesian coordinates of each charges are listed in Table S1. The inside dielectric constant was set to 1.0, with that of outside set to 80.0. The truncation order l in Eqn (18) was chosen to be 120, leading to a precision of 10^{-6}.

A total of 573 biomolecular structures from the Amber PBSA benchmark suite were used to study the accuracy and efficiency of the new method.[13] We further chose eight small proteins about 1000 atoms from the benchmark suite to analyze the convergence of the new method, so that the jobs can be handled on our local compute nodes at the finest grid spacing tested. These biomolecules were assigned charges of Cornell *et al*[88] and the modified bond radii. The probe radius was set to 1.4 Å. All testing runs were performed with the following conditions unless specified otherwise. The convergence criterion of 10^{-4} was used for the BiCG linear system solver. The default grid spacing was 0.5 Å for most calculations, except that they were set as 1.0 Å, 0.5 Å, 0.33 Å, 0.25 Å, and 0.125 Å in the convergence analysis. The ratio of the grid dimension over the solute dimension (the *fillratio* keyword in Amber) was set to 1.5 and 4.0 for biomolecular tests and analytical tests, respectively. No electrostatic focusing was applied. The potential values on all grid points were initialized to zero. The conductor boundary, i.e. zero potential, was used in all non-periodic PBSA calculations. The dielectric constants were set to 1 and 80 for solute and solvent, respectively. All other parameters were set as default in the PBSA module in Amber 18 package.[84]

A BiCG linear system solver for GPUs was also implemented using the Nvidia CUDA Sparse Matrix (cuSPARSE) library, which provides basic linear algebra procedures for sparse matrix operations.[89] The CSR matrix format was used for the non-symmetric coefficient matrix as in our previous publication.[90] All GPU and CPU tests were conducted on a dedicated compute node with two NVIDIA TITAN Xp GPU cards, one Intel Xeon E5-1620 v3 CPU, and 16GB main memory. Our time measurements for both solvers include all

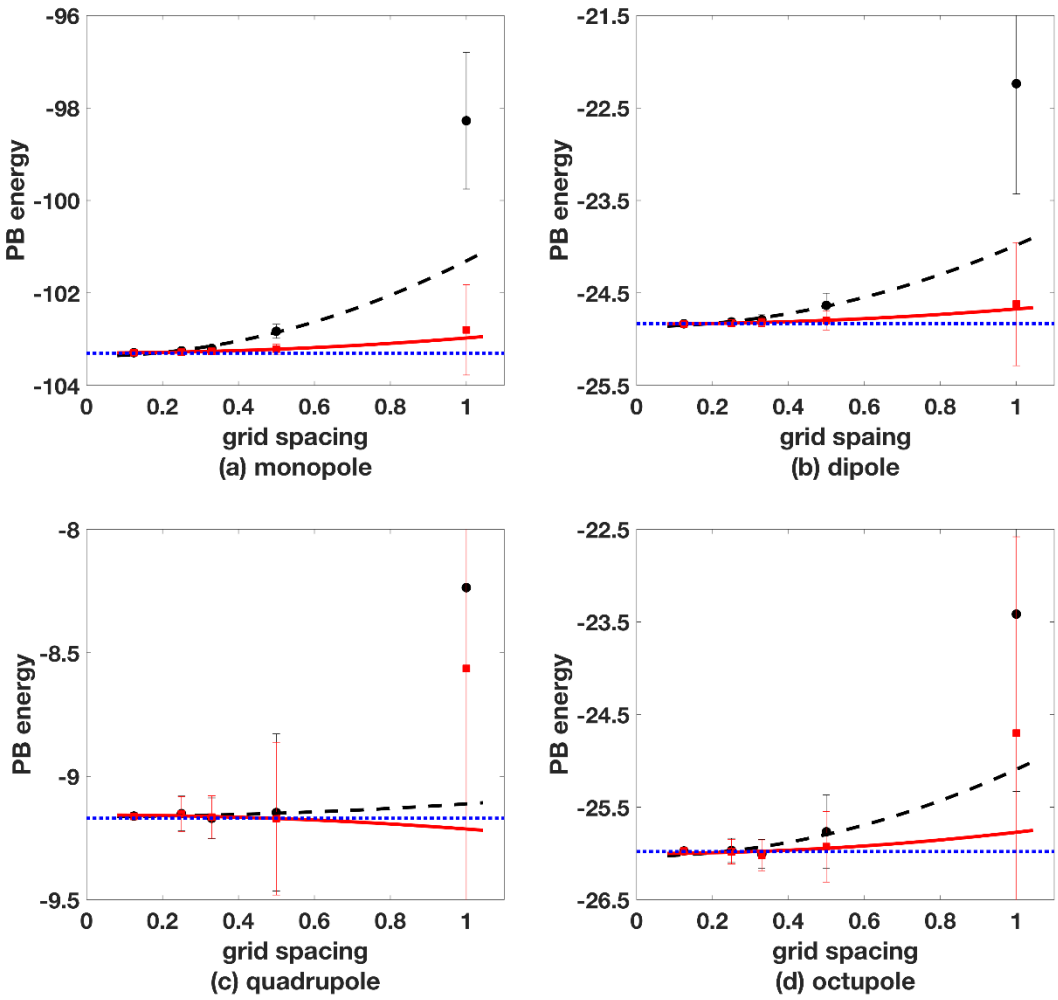
execution time of the solver routines, i.e. time elapsed on device (GPU) and on host (CPU) and also for transferring data between the device and the host.

CHAPTER 3

RESULTS AND DISCUSSION

4.1 Performance on Analytical Test Cases

4.1.1 Reaction Field Energy The accuracy and precision of reaction field energies were investigated with five different grid spacings, 1 Å, 0.5 Å, 0.33 Å, 0.25 Å and 0.125 Å. The detailed results of different programs with and without analytical setup are shown in Figure 2, Table 1, and Table S2.



— analytical setup --- numerical setup ⋯ analytical value

Figure 2. Convergence of reaction field energies (kcal/mol) versus grid spacing (Å) for analytical sphere models with both numerical and analytical setups for IIM. The energy results are obtained by averaging 30 grid orientations/offsets for each test case. All the curves are obtained by fitting data to parabolas. Note that the data points at grid spacing 1 Å are not included in the fitting.

Clearly, the analytical setup obtains better accuracy than the numerical setup in all the cases regardless of the grid spacing. Interestingly, the analytical setup version can obtain a good result even with coarse grid, for example in 1 Å spacing. In Figure 2 (a) and (b), though not included in the fitting process, data points at grid spacing of 1 Å are close to the fitting curves in the analytical setup, showing that those data have already entered the convergence region. However, for numerical setup, there is no sign of entering convergence region for data at 1 Å for any test cases. Table 1 further shows that the relative errors are already less than 0.1% for all the cases at the grid spacing of 0.5 Å when the analytical setup was used, much less than those of the numerical setup with the medium value of about 0.7%, (Table S2). This indicates that the analytical setup leads to faster converged results than the numerical setup.

Table 1. Reaction field energies (kcal/mol) for the analytical sphere models with the analytical setup. Each energy is an average of 30 random placements of the finite difference grid over a tested model. Analytical reaction field energies were calculated using Mathematica 7.0.

Spacing	1.000Å	0.500Å	0.330Å	0.250Å	0.125Å	Analytical
Monopole	-102.80	-103.22	-103.26	-103.28	-103.29	-103.30
Dipole	-24.62	-24.80	-24.81	-24.83	-24.83	-24.83
Quadrupole	-8.56	-9.17	-9.17	-9.15	-9.16	-9.17

Octupole	-24.70	-25.93	-26.02	-25.98	-25.98	-25.98

4.1.2 Surface Field Surface properties are often important for calculation of electrostatic forces, so we used the electric field at interface to show how well the interface properties can be reproduced in both setups. The results from different setups are presented in Figure 3. The root mean square deviation (RMSD) values were calculated by comparing values from different interpolation schemes and the theoretical values obtained from Eqn (17), where the three interpolation methods used are those studied in detail in the literature.[91]

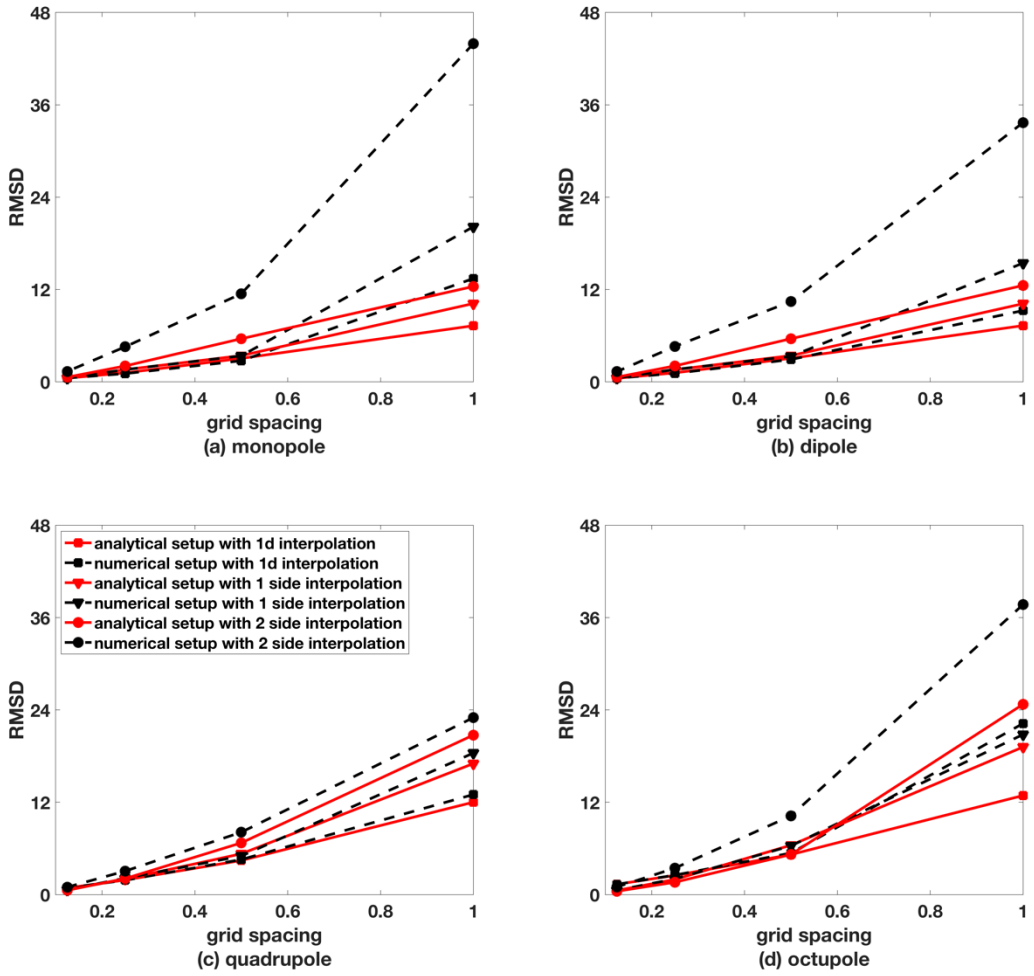


Figure 3. RMSDs between computed normal component of the surface electric field with analytical values (kcal/mol-e-Å) versus grid spacing (Å) for analytical sphere models with both numerical and analytical setups for IIM.

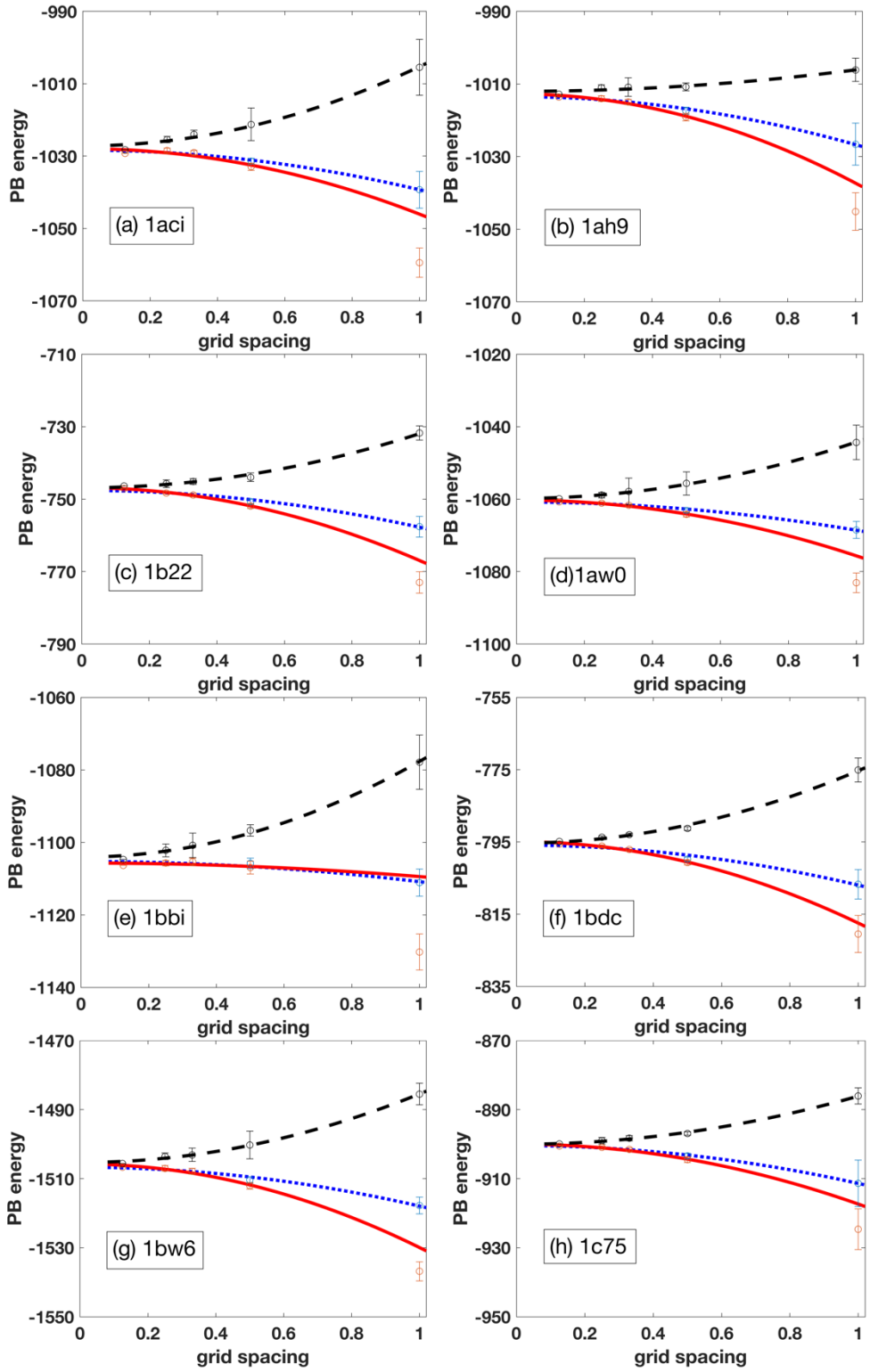
It is clear that for all the three interpolation schemes, the analytical setup always gives better agreement with theory. Of all the interpolation schemes used, the 1-d interpolation shows the best performance, which is consistent with the literature observation.[91] Interestingly for some cases, the worst results for the analytical setup can even beat the best results for the numerical setup.

4.2 Reaction Field Energies of Nontrivial Biomolecules

4.2.1 Convergence Analysis We conducted convergence analysis of the PB reaction field energy versus grid spacing for the IIM with different setups. The results for the eight chosen proteins are shown in Figure 4. IIM with both analytical and numerical setups converge at very fine grids in a unified quadratic manner, as predicted by the theory. The converged values are consistent with our previously implemented solvers, and also consistent with other existing solvers as in DelPhi and APBS, and higher order solver such as MIBPB.[92] However, the convergence curves do not show a significant improvement after we analytically computed all interface parameters and jump conditions. Indeed, some of the test cases were found to lead to even worse results (as in case (b) of Figure 4).

The reason is that we do not have proper interface parameters. When numerically solving PDEs for systems with interfaces, interface parameters must converge in a quadratic form, so we can guarantee the whole numerical scheme converges in a quadratic manner. However, this does not hold for the analytical setup. Even if the analytical setup

can obtain extremely accurate interface parameters (with relative error of 10^{-6}) on each projection point, the projection point may not necessarily be a good representation of its neighborhood on the interface. Therefore, when we resort to a finer grid spacing, a new projection point in this neighborhood may give a set of totally different interface parameters (even the sign may be different for curvatures due to the crowded atomic packing in biomolecules), causing the method fail to converge in a quadratic manner. Indeed, if we include the data at the grid spacing of 1.0 Å in the fitting curves, the asymptotic energy would converge to a wrong value. This phenomenon shows that the data at the space 1.0 Å have not entered the convergence region, thus leading to poor performance with the analytical setup. We believe this is because the interface parameters do not converge in a quadratic form, which motivated us to develop a surface regulation scheme.



— analytical setup --- numerical setup ⋯ analytical setup with surface regulation

Figure 4. Convergence trends for reaction field energies of nontrivial biomolecules (kcal/mol) versus grid spacing (Å). The energy results are obtained by averaging of systematic 30 rotations/offsets of the tested molecules. All the curves are obtained by fitting data at discrete grid point to a parabola ($y=a+bx^2$). (Note that for the analytical setup without surface regulation, data at grid spacing equal to 1 Å are not included in the fitting.)

That better interface parameters are required to unleash the full potential of the analytical setup can also be illustrated in the following rationale. Nontrivial biomolecules in our test cases contain atoms with radii as small as 0.6 Å. Thus, a sampling grid spacing of 1.0 Å or 0.5 Å is inadequate to capture the surface curvatures ($\xi_{\eta\eta}$, etc). In the numerical IIM scheme, the interface properties (such as curvatures) are not accurate as they would be in the analytical IIM, because certain fuzziness from the use of the coarse grid is introduced within the neighborhood of the projection point.[82] Coincidently, however, these inaccurate properties lead to a better representation of the surface for the numerical IIM when the coarse grid is used (Figure 4). Therefore, for a coarse grid such as of 1.0 Å, the accuracy of the IIM scheme cannot achieve what the theory best predicts and thus the data deviate from the fitting curves. Thus, a better way to deal with coarse grids is to adopt some fuzziness for the surface description. We implemented the surface regulation to introduce some "inaccuracy" but more "representativeness" into the analytical IIM for coarse grid spacings such as 1.0 Å.

As shown in Figure 4, the convergence performance of the analytical setup with surface regulation is much better than the results without regulation and the data follow the quadratic pattern more closely. Worth noting is that the PB energies at the grid spacing of 1.0 Å enter the convergence region when the surface regulation is used. In addition, the

analytical setup makes it converge faster (flatter convergence curve) than the numerical setup for every test case. The detailed discussion is shown below.

4.2.2 Accuracy and Uncertainty For the widely used coarse grid spacing of 1.0 Å, the analytical setup without regulation does not show an improvement. The maximum, medium and minimum relative errors of reaction field energy at 1.0 Å for the eight tested proteins are 3.51%, 2.90% and 2.05%, respectively (Table 2). Those for the numerical setup are 2.54%, 1.79%, and 0.58%, respectively (Table S3). It can be seen that the analytical setup alone does not improve the convergence for complex interfaces. The reason is that those projection points do not have enough representativeness as we discussed above. However, for analytical setup with regulation, those three errors are 1.22%, 1.03%, and 0.49%, respectively (Table 3), with the accuracy almost doubled for all chosen cases. This shows that by introducing surface regulation, we successfully made the interface more representative and thus improved the convergence properties. As for the widely used coarse grid spacing of 0.5 Å, the convergence improvement is similar to that of 1.0 Å, with the convergence of the analytical setup with regulations is also roughly twice better than the other two schemes.

Since our method is developed to balance accuracy and efficiency as it is mostly used to process tens of thousands of structures to post-process MD trajectories, such as in MMPBSA calculations, it is essential to make sure the PB energy error is negligible compared to the intrinsic error of MMPBSA, which is usually about 10% as discussed in one of our recent papers.[93] For the recommended grid spacing of 0.5 Å, the relative error of PB energies with respect to the limiting values at 0 Å is about 0.5% (Table 3), which is twenty times smaller than the intrinsic error and clearly negligible. It is also interesting to

compare with some existing higher-order solvers, such as MIBPB, which showed a relative error of about 0.2% at their recommended grid spacing of 1 Å with respect to the finest tested grid of 0.2 Å.[92] This is basically the same if we also compare the values with those at 0.2 Å instead of 0 Å for the analytical IIM. In summary the accuracy of both our second-order solver and higher-order solvers is sufficient for post-processing methods that are based on intensive PBSA calculations. While on the other hand, the trade-off in accuracy and efficiency implies that our second-order solver shall benefit more in CPU intensive applications.

Table 2. Reaction field energies (kcal/mol) of eight selected proteins with analytical setup.

spacing	1aci	1ah9	1b22	1aw0	1bbi	1bdc	1bw6	1c75
1.000Å	-1059.46	-1045.14	-772.98	-1083.10	-1130.23	-820.50	-1536.82	-924.65
0.500Å	-1032.95	-1019.10	-751.89	-1064.22	-1106.95	-800.71	-1512.05	-904.49
0.333Å	-1029.06	-1014.96	-748.88	-1061.70	-1105.35	-797.09	-1507.84	-901.55
0.250Å	-1028.47	-1014.02	-748.30	-1061.11	-1105.72	-796.15	-1507.01	-900.93
0.125Å	-1029.26	-1013.55	-747.08	-1060.72	-1106.34	-795.56	-1506.63	-900.68
lim values	-1028	-1013	-746.8	-1060	-1106	-794.9	-1506	-900.0

Table 3. Reaction field energies (kcal/mol) of eight selected proteins with analytical setup and the regulation treatment.

spacing	1aci	1ah9	1b22	1aw0	1bbi	1bdc	1bw6	1c75
1.000Å	-1035.44	-1023.20	-754.93	-1065.92	-1108.35	-804.33	-1514.27	-908.97
0.500Å	-1028.61	-1014.24	-748.51	-1061.08	-1103.45	-797.99	-1507.33	-901.28
0.333Å	-1025.94	-1011.44	-746.64	-1059.50	-1102.77	-795.28	-1504.58	-899.36
0.250Å	-1026.06	-1011.24	-746.53	-1059.39	-1103.41	-794.77	-1504.37	-899.20
0.125Å	-1027.52	-1011.6	-745.98	-1059.60	-1104.72	-794.70	-1504.93	-899.54
lim values	-1026	-1011	-745.9	-1059	-1103	-794.6	-1504	-898.7

Another point worth addressing is the numerical uncertainty of numerical reaction field energies. As stated previously, the final PB energy is an average over PB energies of 30 different grid orientations/offsets of our test molecule. The standard deviations of collected

data can be used to show the numerical uncertainty of the algorithm. Table 4 shows the standard deviations of the 30 energies computed at grid spacings of 0.5 Å and 1.0 Å. For the numerical setup at grid spacing of 1.0 Å, the maximum, medium and minimum fluctuations among the 8 tested proteins are 7.70, 3.27, and 1.93 kcal/mol, respectively. Those for the analytical setup at the same grid spacing are 5.87, 4.51 and 2.68 kcal/mol, respectively. And for the analytical setup with regulation, they are 3.25, 2.62 and 1.86 kcal/mol, respectively. In general, the energy of the numerical setup may fluctuate about 2~3 times larger than the analytical setup, which clearly shows the analytical setup is much more stable and reliable than the numerical setup. Similar behaviors were also observed with the grid spacing of 0.5 Å. The regulation scheme almost always shows a smaller fluctuation than the other two, and the performance is roughly 2~3 times better as shown in Table 4. Thus it is clear that both the convergence accuracy and the numerical fluctuation of our new analytical IIM solver are in the order of a few tenths of a percentage point at the recommended grid spacing of 0.5 Å, and also with higher efficiency.

Table 4. Reaction field energies standard deviation (kcal/mol) of eight selected proteins with three setups at grid spacing of 0.5 Å and 1.0 Å.

	1aci	1ah9	1b22	1aw0	1bbi	1bdc	1bw6	1c75
0.5 Å								
numerical	4.529	1.070	1.209	3.214	1.571	0.509	4.015	0.532
analytical	1.017	1.037	0.660	0.721	1.782	0.628	0.944	0.821
regulation	0.838	1.046	0.707	0.570	1.332	0.591	0.971	0.722
1.0 Å								
numerical	7.704	3.202	1.934	4.743	7.476	3.332	3.128	2.333
analytical	4.024	5.191	2.938	2.678	4.997	5.154	2.762	5.873
regulation	3.250	2.937	2.752	2.292	2.768	1.885	2.495	1.856

4.3 Timing and Robustness

Figure 5 shows the timing comparison between two different setups for IIM. It is clear that the analytical setup is faster in all the cases, and the improvement in the solver time is significant for grid spacings of widely used 1.0 Å and 0.5 Å, for which the performance of analytical setup is five-times/double of that of the numerical setup on average. We believe this is because the analytical setup gives much more precise coefficients for the final discretized equations, leading to better-conditioned linear systems. Thus, the numerical solver takes fewer iterations to reach the convergence. Because the solver phase is the most time-consuming part in these programs as shown in Table 5, this leads to a dramatic reduction in the overall running time. For finer grid spacing of 0.25 Å, the speedup is not that significant any more, which is understandable, as the difference between analytical setup and numerical setup becomes negligible.

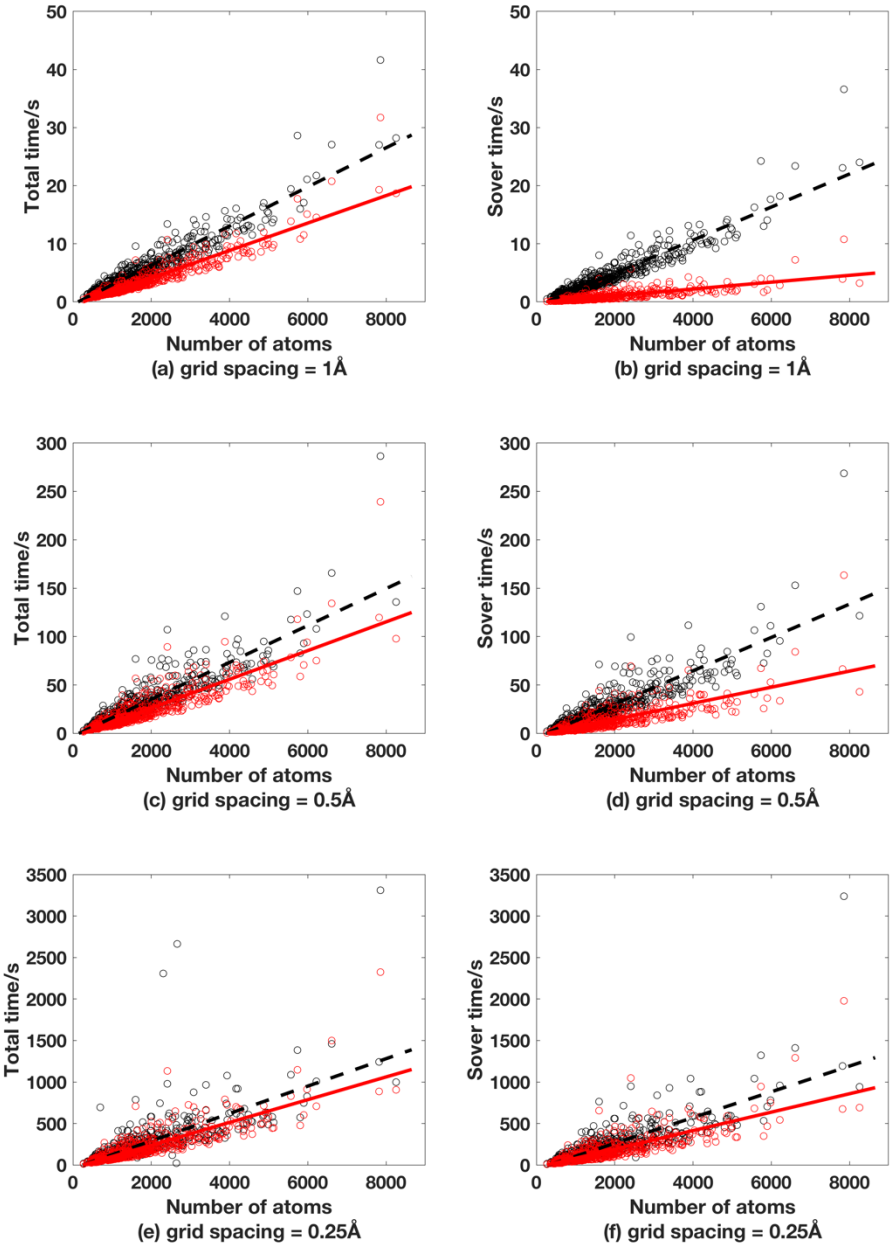
Table 5. Percentages of the average solver time in IIM runs for the protein test set.

grid spacing	1Å	0.5Å	0.25Å
numerical setup	74%	82%	90%
analytical setup	23%	51%	78%

Figure 5 also shows that the analytical setup can be more stable. For spacing of both 1Å and 0.5Å cases, the timing data are more concentrated near the fitting curves, which shows a more unified scaling behavior. Besides, for the protein test set, 3 (out of 573) cases failed for the numerical setup (bad data or NaN for PB energies), but none for the analytical setup, as showed in Table 6. We further confirmed this result with the analytical setup tests in which no case failed. The reaction field energies of the analytical setup tests are shown in Table S4 in Supplementary Materials. This is consistent with the observations in other analyses of the methods.

Table 6. Failed test cases for the IIM runs with numerical setup.

grid spacing	1Å	0.5Å	0.25Å
Failures	1lyp 1dip	1pih 1mut 1e6q	2jhb 1hyk 1gnc

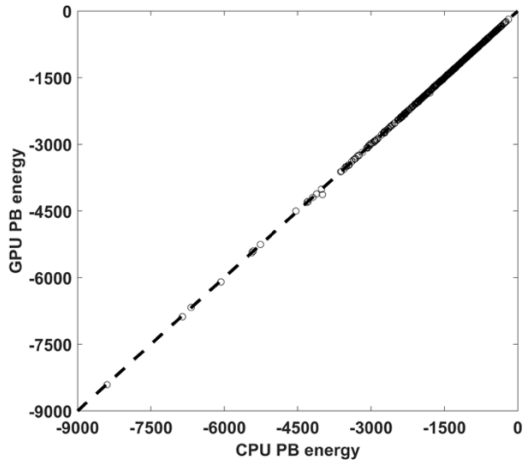


— analytical setup --- numerical setup

Figure 5. Timing comparison between two different setups (without regulation).

4.4 GPU Implementation

Finally, we implemented the BiCG linear system solver for the IIM, which is the most time-consuming portion of the algorithm as shown in Table 5. The results with the protein test set are showed in Figure 6. The PB energies agree excellently between GPU and CPU programs, with the medium relative error of 0.07% for the test set. The timing of the GPU solver also shows an impressive performance boost here, with the speedup ratio of about 20. The implementation on GPU makes the analytical setup IIM program more feasible for practical biomolecular applications.



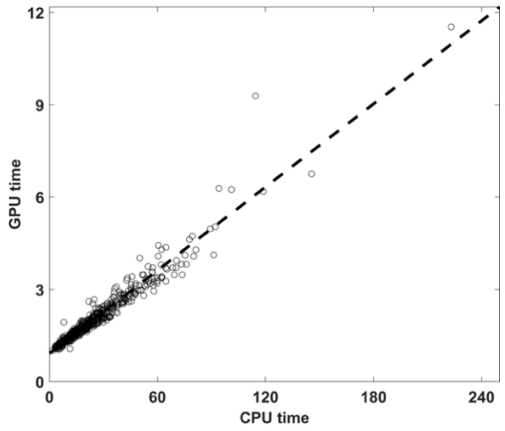


Figure 6. Comparison of PB energies (kcal/mol) and solver timing (seconds) between GPU and CPU runs. The energy trend line is 1.001x+1.088, with the medium relative deviation between the two sets 0.066%. The timing trend line is 0.045x+0.921, with the GPU program about 20 times faster than the CPU program.

CHAPTER 3

CONCLUSION

In this study, we introduced both analytical setup of interface conditions and surface regulation into the IIM to address its instability and slow convergence. The analytical setup was found to obtain more accurate solutions than the numerical setup in the analytical test cases. The method was found to be able to obtain a good result even with coarse grid spacings, like 1 Å. At the widely used grid spacing of 0.5 Å, the analytical setup can also lead to much smaller relative errors (less than 0.1%) than the numerical setup. Overall the analytical setup converges in a quadratic manner, as predicted by theory.

The surface regulation scheme further speeds up the convergence for nontrivial biomolecules. For the widely used coarse grid spacings of 0.5 and 1.0 Å, the analytical setup with the surface regulation was found to roughly halve the relative errors from the numerical setup. Specifically, the relative error of PB energies at 0.5 Å is about 0.5% with respect to the limiting values at 0 Å or about 0.2% with respect to the values computed at 0.2 Å. These are similar to those of the higher-order solvers such as MIBPB at their recommended grid spacing. In addition, the numerical uncertainty of the PB energies is reduced to a few tenth of a percentage point of the mean. In summary both numerical error and uncertainty of the new analytical IIM method are much smaller than the thermal noise of typical room-temperature MD simulations, for which the PB method is often used as a post-processing tool.

More interestingly the analytical setup significantly improves the solver efficiency at the tested coarser grid spacings. For grid spacings of 1 Å and 0.5 Å, the solver time was

decreased by 5x and 2x, respectively. This is because the analytical setup generates more precise coefficients for the final discretized equations, leading to better-conditioned linear systems. This also leads to a more stable algorithm. In our protein test set over 500 proteins, none of them failed with the analytical setup. Finally, we implemented the GPU BiCG solver for the IIM. The PB energies agree excellently between GPU and CPU programs, and the timing analysis shows that the GPU solver is about 20 faster than the CPU solver, which makes the analytical setup IIM program more feasible for practical biophysical studies.

REFERENCE

1. 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 Structure Function and Genetics* **1986,** *1* (1), 47-59.

2. Davis, M. E.; McCammon, J. A., Solving the Finite-Difference Linearized Poisson-Boltzmann Equation - a Comparison of Relaxation and Conjugate-Gradient Methods. *J. Comput. Chem.* **1989,** *10* (3), 386-391.

3. 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.

4. Luty, B. A.; Davis, M. E.; McCammon, J. A., Solving the Finite-Difference Nonlinear Poisson-Boltzmann Equation. *J. Comput. Chem.* **1992,** *13* (9), 1114-1118.

5. Holst, M.; Saied, F., Multigrid Solution of the Poisson-Boltzmann Equation. *J. Comput. Chem.* **1993,** *14* (1), 105-113.

6. Forsten, K. E.; Kozack, R. E.; Lauffenburger, D. A.; Subramaniam, S., Numerical-Solution of the Nonlinear Poisson-Boltzmann Equation for a Membrane-Electrolyte System. *J. Phys. Chem.* **1994,** *98* (21), 5580-5586.

7. 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.

8. 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.

9. Luo, R.; David, L.; Gilson, M. K., Accelerated Poisson-Boltzmann calculations for static and dynamic systems. *Journal of Computational Chemistry* **2002,** *23* (13), 1244-1253.

10. Bashford, D., An Object-Oriented Programming Suite for Electrostatic Effects in Biological Molecules. *Lecture Notes in Computer Science* **1997,** *1343*, 233-240.

11. Lu, Q.; Luo, R., A Poisson-Boltzmann dynamics method with nonperiodic boundary condition. *Journal of Chemical Physics* **2003,** *119* (21), 11035-11047.

12. Prabhu, N. V.; Zhu, P. J.; Sharp, K. A., Implementation and testing of stable, fast implicit solvation in molecular dynamics using the smooth-permittivity finite difference Poisson-Boltzmann method. *Journal of Computational Chemistry* **2004,** *25* (16), 2049-2064.

13. Wang, J.; Luo, R., Assessment of Linear Finite-Difference Poisson-Boltzmann Solvers. *Journal of Computational Chemistry* **2010,** *31* (8), 1689-1698.

14. Cai, Q.; Hsieh, M. J.; Wang, J.; Luo, R., Performance of Nonlinear Finite-Difference Poisson-Boltzmann Solvers. *J Chem. Theory Comput.* **2010,** *6* (1), 203-211.

15. 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.

16. Sharp, K.; Honig, B., Lattice models of electrostatic interactions–the finite-difference Poisson-Boltzmann method. *Chem. Scr. A* **1989,** *29*, 71-74.

17. Cortis, C. M.; Friesner, R. A., Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes. *J. Comput. Chem.* **1997,** *18* (13), 1591-1608.

18. 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.

19. 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.

20. Shestakov, A. I.; Milovich, J. L.; 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.

21. Chen, L.; Holst, M. J.; Xu, J. C., The finite element approximation of the nonlinear Poisson-Boltzmann equation. *SIAM Journal on Numerical Analysis* **2007,** *45*, 2298-2320.

22. Xie, D.; Zhou, S., A new minimization protocol for solving nonlinear Poisson–Boltzmann mortar finite element equation. *BIT Numerical Mathematics* **2007,** *47* (4), 853-871.

23. Friedrichs, M.; Zhou, R. H.; Edinger, S. R.; Friesner, R. A., Poisson-Boltzmann analytical gradients for molecular modeling calculations. *J Phys. Chem. B* **1999,** *103* (16), 3057-3061.

24. Bond, S. D.; Chaudhry, J. H.; Cyr, E. C.; Olson, L. N., A First-Order System Least-Squares Finite Element Method for the Poisson-Boltzmann Equation. *J Comput. Chem.* **2010,** *31* (8), 1625-1635.

25. Lu, B.; Holst, M. J.; McCammon, J. A.; Zhou, Y. C., Poisson-Nernst-Planck equations for simulating biomolecular diffusion-reaction processes I: Finite element solutions. *J Comput Phys* **2010,** *229* (19), 6979-6994.

26. Lu, B. Z.; Zhou, Y. C., 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.

27. 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.

28. 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.

29. Zauhar, R. J.; Morgan, R. S., The Rigorous Computation of the Molecular Electric-Potential. *J. Comput. Chem.* **1988,** *9* (2), 171-187.

30. Rashin, A. A., Hydration Phenomena, Classical Electrostatics, and the Boundary Element Method. *J. Phys. Chem.* **1990,** *94* (5), 1725-1733.

31. Yoon, B. J.; Lenhoff, A. M., A Boundary Element Method for Molecular Electrostatics with Electrolyte Effects. *J. Comput. Chem.* **1990,** *11* (9), 1080-1086.

32. Juffer, A. H.; Botta, E. F. F.; Vankeulen, B. A. M.; Vanderploeg, A.; Berendsen, H. J. C., The Electric-Potential of a Macromolecule in a Solvent - a Fundamental Approach. *J. Comput. Phys.* **1991,** *97* (1), 144-171.

33. Zhou, H. X., Boundary-Element Solution of Macromolecular Electrostatics - Interaction Energy between 2 Proteins. *Biophys. J.* **1993,** *65* (2), 955-963.

34. 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.

35. Purisima, E. O.; Nilar, S. H., A Simple yet Accurate Boundary-Element Method for Continuum Dielectric Calculations. *J. Comput. Chem.* **1995,** *16* (6), 681-689.

36. Liang, J.; Subramaniam, S., Computation of molecular electrostatics with boundary element methods. *Biophys. J.* **1997,** *73* (4), 1830-1841.

37. Vorobjev, Y. N.; Scheraga, H. A., A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *J. Comput. Chem.* **1997,** *18* (4), 569-583.

38. 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.

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

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

41. Lu, B. Z.; Cheng, X. L.; Hou, T. J.; McCammon, J. A., 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).

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

43. Lu, B. Z.; Cheng, X. L.; Huang, J. F.; McCammon, J. A., An Adaptive Fast Multipole Boundary Element Method for Poisson-Boltzmann Electrostatics. *J Chem. Theory Comput.* **2009,** *5* (6), 1692-1699.

44. Bajaj, C.; Chen, S. C.; 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.

45. Dominy, B. N.; Brooks, C. L., Development of a generalized born model parametrization for proteins and nucleic acids. *Journal Of Physical Chemistry B* **1999,** *103* (18), 3765-3773.

46. Onufriev, A.; Bashford, D.; Case, D. A., Modification of the generalized Born model suitable for macromolecules. *Journal Of Physical Chemistry B* **2000,** *104* (15), 3712-3720.

47. Srinivasan, J.; Trevathan, M. W.; Beroza, P.; Case, D. A., Application of a pairwise generalized Born model to proteins and nucleic acids: inclusion of salt effects. *Theoretical Chemistry Accounts* **1999,** *101* (6), 426-434.

48. Tsui, V.; Case, D. A., Molecular dynamics simulations of nucleic acids with a generalized born solvation model. *J Am Chem Soc* **2000,** *122* (11), 2489-2498.

49. Gallicchio, E.; Zhang, L. Y.; Levy, R. M., 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.

50. Lee, M. S.; Salsbury, F. R.; Brooks, C. L., Novel generalized Born methods. *J Chem Phys* **2002,** *116* (24), 10606-10614.

51. Onufriev, A.; Case, D. A.; Bashford, D., Effective Born radii in the generalized Born approximation: The importance of being perfect. *Journal Of Computational Chemistry* **2002,** *23* (14), 1297-1304.

52. Gallicchio, E.; Levy, R. M., AGBNP: An analytic implicit solvent model suitable for molecular dynamics simulations and high-resolution modeling. *Journal of Computational Chemistry* **2004,** *25* (4), 479-499.

53. Onufriev, A.; Bashford, D.; Case, D. A., 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.

54. Li, X. F.; Hassan, S. A.; Mehler, E. L., 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.

55. Chen, J. H.; Im, W. P.; Brooks, C. L., Balancing solvation and intramolecular interactions: Toward a consistent generalized born force field. *J Am Chem Soc* **2006,** *128* (11), 3728-3736.

56. 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.

57. Davis, M. E.; Mccammon, J. A., 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.

58. LeVeque, R. J.; Li, Z., The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J. Numer. Anal.* **1994,** *31*, 1019-1044.

59. 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.

60. 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. *Physical Chemistry Chemical Physics* **2013**.

61. 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.

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

63. Geng, W. H.; Yu, S. N.; Wei, G. W., Treatment of charge singularities in implicit solvent models. *J Chem. Phys.* **2007,** *127* (11).

64. Yu, S. N.; Wei, G. W., Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities. *Journal of Computational Physics* **2007,** *227* (1), 602-632.

65. Zhou, Y. C.; Feig, M.; Wei, G. W., Highly accurate biomolecular electrostatics in continuum dielectric environments. *Journal of Computational Chemistry* **2008,** *29*, 87-97.

66. Zheng, Q.; Yang, S. Y.; Wei, G. W., Biomolecular surface construction by PDE transform. *Int J Numer Meth Bio* **2012,** *28* (3), 291-316.

67. Boschitsch, A. H.; Fenley, M. O., Hybrid boundary element and finite difference method for solving the nonlinear Poisson–Boltzmann equation. *Journal of computational chemistry* **2004,** *25* (7), 935-955.

68. Boschitsch, A. H.; Fenley, M. O., A fast and robust Poisson–Boltzmann solver based on adaptive Cartesian grids. *Journal of chemical theory and computation* **2011,** *7* (5), 1524-1540.

69. 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.

70. Chakravorty, A.; Jia, Z.; Peng, Y.; Tajielyato, 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.

71. Dzubiella, J.; Swanson, J. M. J.; McCammon, J. A., Coupling nonpolar and polar solvation free energies in implicit solvent models. *J Chem. Phys.* **2006,** *124* (8).

72. Dzubiella, J.; Swanson, J. M. J.; McCammon, J. A., Coupling hydrophobicity, dispersion, and electrostatics in continuum solvent models. *Phys. Rev. Lett.* **2006,** *96* (8).

73. Cheng, L. T.; Dzubiella, J.; McCammon, J. A.; Li, B., Application of the level-set method to the implicit solvation of nonpolar molecules. *J Chem. Phys.* **2007,** *127* (8).

74. Chen, Z.; Zhao, S.; Chun, J.; Thomas, D. G.; Baker, N. A.; Bates, P. W.; Wei, G. W., Variational approach for nonpolar solvation analysis. *J Chem. Phys.* **2012,** *137* (8).

75. Thomas, D. G.; Chun, J.; Chen, Z.; Wei, G. W.; Baker, N. A., Parameterization of a geometric flow implicit solvation model. *J Comput. Chem.* **2013,** *34* (8), 687-695.

76. 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.

77. 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.

78. 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.

79. 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.

80. Botello-Smith, W. M.; Liu, X. P.; Cai, Q.; Li, Z. L.; Zhao, H. K.; Luo, R., Numerical Poisson-Boltzmann model for continuum membrane systems. *Chem. Phys. Lett.* **2013,** *555*, 274-281.

81. 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.

82. Li, Z.; Ito, K., *The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains*. Siam: 2006; Vol. 33.

83. 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.

84. Case, D. A.; Brozell, S. R.; Cerutti, D. S.; Cheatham, T. E., III; Cruzeiro, V. W. D.; Darden, T. A.; Duke, R. E.; Ghoreishi, D.; Gohlke, H.; Goetz, A. W.; Greene, D.; Harris, R.; Homeyer, N.; Izadi, S.; Kovalenko, A.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Mermelstein, D. J.; Merz, K. M.; Miao, Y.; Monard, G.; Nguyen, H.; Omelyan, I.; Onufriev, A.; Pan, F.; Qi, R.; Roe, D. R.; Roitberg, A.; Sagui, C.; Schott-Verdugo, S.; Shen, J.; Simmerling, C. L.; Swails, J.; Walker, R. C.; Wang, J.; Wei, H.; Wolf, R. M.; Wu, X.; Xiao, L.; York, D. M.; Kollman, P. A., AMBER 2018. University of California, San Francisco., 2018.

85. Zhou, Z. X.; Payne, P.; Vasquez, M.; Kuhn, N.; Levitt, M., Finite-difference solution of the Poisson-Boltzmann equation: Complete elimination of self-energy. *J Comput. Chem.* **1996,** *17* (11), 1344-1351.

86. Chern, I.-L.; Liu, J.-G.; Wang, W.-C., Accurate Evaluation of Electrostatics for Macromolecules in Solution. *Methods and Applications of Analysis* **2003,** *10*, 309-328.

87. Rocchia, W.; Sridharan, S.; Nicholls, A.; Alexov, E.; Chiabrera, A.; Honig, B., Rapid grid-based construction of the molecular surface and the use of induced surface

charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *Journal of computational chemistry* **2002,** *23* (1), 128-137.

88. Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A., A 2ND GENERATION FORCE-FIELD FOR THE SIMULATION OF PROTEINS, NUCLEIC-ACIDS, AND ORGANIC-MOLECULES. *J. Am. Chem. Soc.* **1995,** *117* (19), 5179-5197.

89. Nvidia NVIDIA CUDA Sparse Matrix library. https://developer.nvidia.com/cusparse (accessed October 1st, 2016).

90. Qi, R.; Botello-Smith, W. M.; 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.

91. 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.

92. Nguyen, D. D.; Wang, B.; Wei, G. W., Accurate, robust, and reliable calculations of Poisson–Boltzmann binding energies. *Journal of computational chemistry* **2017,** *38* (13), 941-948.

93. Wang, C.; Nguyen, P. H.; Pham, K.; Huynh, D.; Le, T. B. N.; Wang, H.; Ren, P.; Luo, R., Calculating protein–ligand binding affinities with MMPBSA: Method and error analysis. *Journal of computational chemistry* **2016,** *37* (27), 2436-2446.

SUPPLEMENTARY MATERIALS

Table S1. Cartesian coordinates of solute charges for the analytical monopole, dipole, quadrupole and octapole models, respectively.

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, -0.10)
Quadrupole	(0.70, -0.10, -0.10) (-0.10, 0.70, -0.10)	(0.70, 0.70, -0.10) (-0.10, -0.10, -0.10)
Octupole	(0.57, -0.20, 0.57) (-0.20, 0.57, 0.57) (0.57, -0.20, -0.20) (-0.20, 0.57, -0.20)	(0.57, 0.57, 0.57) (-0.20, -0.20, 0.57) (0.57, 0.57, -0.20) (-0.20, -0.20, -0.20)

Table S2. Reaction field energies (kcal/mol) for analytical sphere models with the numerical setup. Each reported energy is an average of 30 energies computed with random placements of the finite difference grid over a tested model. Analytical reaction field energies were calculated using Mathematica 7.0.

Spacing	1.000Å	0.500Å	0.330Å	0.250Å	0.125Å	Analytical
Monopole	-98.28	-102.83	-103.20	-103.25	-103.29	-103.30
Dipole	-22.24	-24.63	-24.79	-24.81	-24.83	-24.83
Quadrupole	-8.24	-9.15	-9.17	-9.15	-9.16	-9.17
Octupole	-23.41	-25.77	-26.00	-25.97	-25.97	-25.98

Table S3. Reaction field energies (kcal/mol) of eight selected proteins with the numerical setup.

spacing	1aci	1ah9	1b22	1aw0	1bbi	1bdc	1bw6	1c75
1.000Å	-1005.46	-1006.12	-731.75	-1044.30	-1077.87	-775.09	-1485.50	-886.03
0.500Å	-1021.22	-1010.82	-743.94	-1055.65	-1096.69	-791.29	-1500.24	-896.94
0.333Å	-1023.97	-1010.85	-745.13	-1057.86	-1100.80	-792.96	-1503.08	-898.26
0.250Å	-1025.45	-1011.00	-745.72	-1058.89	-1102.20	-793.70	-1503.55	-898.97
0.125Å	-1028.24	-1012.80	-746.32	-1059.89	-1104.71	-794.85	-1505.51	-899.91
lim values	-1027	-1012	-746.9	-1060	-1104	-795.3	-1505	-900.0

Table S4. Reaction field energies (kcal/mol) of eight failed test cases with the analytical setup.

spacing	1lyp	1dip	1pih	1mut	1e6q	2jhb	1hyk	1gnc
1.000Å	-2442.64	-1627.51	-960.92	-1816.24	-2880.66	-1450.66	-629.21	-1664.35
0.500Å	-2431.12	-1595.02	-940.52	-1774.41	-2788.34	-1411.38	-615.45	-1619.93
0.250Å	-2427.93	-1589.63	-936.65	-1772.68	-2772.91	-1403.33	-612.19	-1615.88