Treatment of geometric singularities in implicit solvent models

Sining Yu and Weihua Geng
Department of Mathematics, Michigan State University, East Lansing, Michigan 48824

G. W. Wei[a]
Department of Mathematics, Michigan State University, East Lansing, Michigan 48824
and Department of Electrical and Computer Engineering, Michigan State University,
East Lansing, Michigan 48824

Geometric singularities, such as cusps and self-intersecting surfaces, are major obstacles to the accuracy, convergence, and stability of the numerical solution of the Poisson-Boltzmann (PB) equation. In earlier work, an interface technique based PB solver was developed using the matched interface and boundary (MIB) method, which explicitly enforces the flux jump condition at the solvent-solute interfaces and leads to highly accurate biomolecular electrostatics in continuum electric environments. However, such a PB solver, denoted as MIBPB-I, cannot maintain the designed second order convergence whenever there are geometric singularities, such as cusps and self-intersecting surfaces. Moreover, the matrix of the MIBPB-I is not optimally symmetrical, resulting in the convergence difficulty. The present work presents a new interface method based PB solver, denoted as MIBPB-II, to address the aforementioned problems. The present MIBPB-II solver is systematical and robust in treating geometric singularities and delivers second order convergence for arbitrarily complex molecular surfaces of proteins. A new procedure is introduced to make the MIBPB-II matrix optimally symmetrical and diagonally dominant. The MIBPB-II solver is extensively validated by the molecular surfaces of few-atom systems and a set of 24 proteins. Converged electrostatic potentials and solvation free energies are obtained at a coarse grid spacing of 0.5 Å and are considerably more accurate than those obtained by the PBEQ and the APBS at finer grid spacings. © *2007 American Institute of Physics.* [DOI: 10.1063/1.2743020]

I. INTRODUCTION

The electrostatics of biomolecules plays an important role in their structure, function, stability, and dynamics.[1,2] Accurate evaluation of electrostatics has therefore always been a major task in molecular/structural biology. The natural form of biomolecules mostly involves solvent. Explicit electrostatic calculations of biomolecules in the solvent remain extremely expensive, despite the development of efficient Ewald summation methods, and alternative approaches based on reaction field theory, periodic images, and Euler summations. The continuum dielectric implicit solvent models,[3,4] in which the explicit interactions between molecules and solvent are represented by a mean-field formulation, have been very popular in structural biology and biochemistry[5–11] since the pioneering work by Warwicker and Watson in the early 1980s (Ref. 12) and Honig and Nichols in the 1990s.[7] Recent efforts focus on free energy estimation,[13–15] pK_α analysis,[16–18] and applications in molecular dynamics.[19,20] The implicit solvent theory retains a microscopic treatment of biomolecules, while adopting a macroscopic description of the solvent. In such an approach, the Poisson-Boltzmann equation (PBE), or Poisson equation if no salt is present, needs to be rigorously solved. An alternative while much fast approach is the generalized Born (GB) formalism. Nevertheless, the GB approach relies on the PBE solution as a calibration or reference standard[21] and has an unknown validity when it is applied to new problems.

Implicit solvent models require a biomolecular surface definition and a prescription of dielectric functions both inside and outside the biomolecule. The solution of the PBE is very sensitive to the discontinuous dielectric function. Earlier biomolecular surfaces, such as van der Waals surfaces and solvent accessible surfaces,[22] are not smooth and could cause instability in numerical simulations. The molecular surface (MS) as proposed by Lee and Richards[22] is designed to provide smoother interfaces. In fact, most PB calculations have used the MS. However, in certain geometric situations, the MS definition admits cusp and self-intersecting surface singularities,[23–26] which still cause numerical instability. Smoothly varying functions were proposed to avoid this problem.[19,27,28] Nonetheless, Swanson *et al.*[29] have shown that some atomic centered dielectric functions may lead to unphysical interstitial high dielectric regions in implicit solvent models. The solvent exclusion property of the MS is crucial to implicit solvent models.

Few analytical solutions of the PBE or Poisson equation exist for realistic biomolecular geometries and charge distributions. Therefore, these equations are usually solved numerically by a variety of computational methods. The computational tools available can be broadly categorized into (1) finite difference (FD) methods,[6,7,9,12,19,30,31] (2) finite element methods,[5,32–34] and (3) boundary integral methods.[10,11,35–39]

a)Author to whom correspondence should be addressed. Fax: (517)432-1562. Electronic mail: wei@math.msu.edu

All methods are subject to certain inherent advantages and limitations, which are closely tied to the associated underlying formulations. Finite element methods are optimal for applications that require the rapid adaptation of grid points to account for structural variation of biomolecules. However, generating unstructured grids for complex biomolecular interfaces is time consuming, especially for large biomolecules. Boundary integral methods enjoy several intrinsic advantages such as fewer unknowns, exact far-field treatment, and accurate representation of surface geometry and charge singularity. However, boundary integral methods are not very efficient in dealing with the nonlinear term in the PB model. Finite difference methods are the main workhorse for solving the PBE in computational structural biology and computational genetic engineering for the following reasons. (1) Using three-dimensional (3D) Cartesian grids, finite difference methods avoid the time-consuming grid generation step. (2) 3D Cartesian grids are a standard option in the most widely used software packages in computational biology, such as DELPHI,[40,41] UHBD,[42] MEAD,[41] APBS,[34,43,44] and CHARMM (Ref. 45) codes; therefore, finite difference based PB solvers naturally fit into these simulation packages. (3) Finite difference methods are relatively simple, particularly in conjunction with multigrid linear algebraic solvers, and can offer the best combination of speed, accuracy, and efficiency, making these the most popular approaches.[3] In the past decade, the main developments with respect to the PB methodology have been focused on acceleration of these numerical methods for the PBE.

Molecular surfaces and discontinuous dielectric functions are commonly employed in the solution of the PBE. Mathematically, the electrostatic potential field becomes nondifferentiable whenever the dielectric constant is discontinuous, which causes numerical instability. In general, the lack of smoothness leads to slow convergence in solving the PBE. In the worst-case scenario, the standard numerical methods do not converge at all for complex irregular solvent-protein interfaces. Computationally, to improve the convergence, the continuities of both electrostatic potential $\phi^- = \phi^+$ and its flux $\epsilon^-(\partial \phi^-/\partial n) = \epsilon^+(\partial \phi^+/\partial n)$ should be explicitly enforced across the dielectric interface, where superscript $-$ indicates that the potential is inside the biomolecules (protein) and the $+$ means that the potential is in the solvent. The partial derivative is defined in the normal direction of the interface as n is the unit outer normal vector. Among many existing PB solvers, errors induced by discontinuous dielectric functions at the solvent-molecular interface are alleviated in the adaptive discretization of some finite element methods[5,46,47] and controlled in the *posteriori* error estimates of other finite element methods.[43] However, explicit enforcement of the flux continuity condition in the context of geometric singularities has not been considered in the literature. Consequently, no PB solver of second order convergence, which means that the error will be reduced by a factor of 4 when the grid is halved, has ever been reported in the context of molecular surfaces of biomolecules. Irregular solvent-solute interface and geometric singularities reduce the accuracy and slow down the convergence of most existing PB solvers. Therefore, stability enhancement and convergence acceleration are pressing practical issues in developing the next generation PB solvers.

In the mathematical context, Peskin[48] pioneered the treatment of elliptic equations with discontinuous coefficients and singular sources. Recently, a number of other elegant methods have been proposed. Among them, the immersed interface method, proposed by LeVeque and Li,[49] is a remarkable second order sharp interface scheme. The ghost fluid method[50] was proposed as a relatively simple and easy to use approach. For irregular interfaces, it is natural to construct a solution in the finite element formulation.[51] A relevant while quite distinct approach is the integral equation method for complex geometry.[52] Recently, we have proposed the matched interface and boundary (MIB) method[53] as a systematic higher-order method for electromagnetic wave propagation and scattering in dielectric media. More recently, we have generalized the MIB for solving elliptic equations with curved interfaces with fourth and sixth order convergences.[54,55] The MIB approach makes use of fictitious domains so that the standard high-order central FD method can be applied across the interface without the loss of accuracy. The fictitious values on fictitious domains are determined from enforcing the flux continuity conditions at the exact position of the interface. Nevertheless, none of the aforementioned interface techniques is able to maintain its designed order of convergence and accuracy in the presence of geometric singularities. Indeed, technically, achieving higher-order convergence at geometric singularities is extremely challenging, despite great desire for doing so in practical applications. The best result in the literature is of 0.8th order convergence reported by Hou and Liu,[56] achieved with a finite element formulation in two dimensional (2D). Very recently, we have developed the first second order convergent MIB scheme in 2D for solving elliptic equations with geometric singularities.[57].

Most recently, we have addressed the interface problem in the PBE by proposing a MIB based Poisson-Boltzmann (MIBPB-I) solver to explicitly consider the flux continuity condition in the finite difference framework.[58] The MIBPB-I solver is of second order convergence in conjunction with smooth molecular surface and discontinuous dielectric functions. It has been validated with the exact Kirkwood solution[59] and tested by solvation energies of 24 proteins. Comparison with other existing methods indicates the potential of the MIBPB-I method. However, our earlier MIBPB-I solver does not maintain the second order convergence when the molecular surface includes cusps, sharp edges, sharp wedges, or self-intersecting surfaces. Moreover, due to the asymmetric matrix of the interface method, the MIBPB-I requires a large number of iterations in solving linear equations. In fact, its matrix does not converge for large proteins or small proteins with dense grids. The objective of the present work is to overcome these difficulties. The present approach, called MIBPB-II, is a generalization of our new 2D MIB method developed for handling sharp-edged interfaces.[57,60] Apart from its ability to maintain the second order convergence under the presence of geometric singularities, the MIBPB-II has an optimally symmetrical matrix, which dramatically reduces the number of iterations.

In the next two sections, the theoretical formulation and the computational algorithm are given for the MIBPB-II solver. The treatment of molecular singularities and the optimization of the MIB matrix structure are developed. Then, the validation and application of the proposed MIBPB-II solver are presented. The accuracy and the order of convergence of the proposed MIB II method are validated by cusp singularity in a diatomic system and by the cusp and self-intersecting surface singularities in a four-atom system. The molecular surfaces of 24 proteins generated by the MSMS (Ref. 26) are employed to further test the convergence of the proposed Poisson solver. Detailed comparison on the accuracy and speed of convergence of the present method is given to our previous MIBPB-I and two other established methods. This article ends with a brief conclusion summarizing the main points.

II. THEORY AND ALGORITHM

A. Poisson-Boltzmann equation

Consider an open bounder domain $\Omega \in \mathbb{R}^3$. Let Γ be the interface which divides Ω into disjoint open subdomains, the biomolecular domain Ω^-, and the solvent domain Ω^+, i.e., $\Omega = \Omega^- \cup \Omega^+ \cup \Gamma$. The Poisson-Boltzmann equation arises under the assumption of the Boltzmann distribution for the solvent ions and leads to a hyperbolic sine term ($\sinh(u)$) for salt effect. Such a nonlinear term can be approximated by u under the weak potential approximation, and the linearized PBE can be given as

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \kappa^2(\mathbf{r}) u(\mathbf{r}) = f(\mathbf{r}), \qquad (1)$$

where ϵ is the dielectric coefficient, $u = e_c \phi / K_B T$ is the dimensionless electrostatic potential, ϕ is the electrostatic potential, and $\kappa(\mathbf{r})$ is the Debye-Hückel screening function describing ion strength. The source term represents the charge contribution $f = 4\pi e^2/k_B T \sum_{i=1}^{N} z_i \delta(\mathbf{r} - \mathbf{r}_i)$, with k_B the Boltzmann constant, e_c the electron charge, and z_i the charge fraction at position $\mathbf{r}_i$. The PBE satisfies the far-field boundary condition $\lim_{|\mathbf{r}| \to \infty} u(\mathbf{r}) = 0$, although the Dirichlet boundary condition is often used in a finite domain. With discontinuous dielectric coefficient at the solute-solvent interface, the PBE should be solved with the following interface jump conditions:

$$[u] = u^+(\mathbf{r}) - u^-(\mathbf{r}), \qquad (2)$$

$$[\epsilon v_n] = \epsilon^+(\mathbf{r}) \nabla u^+(\mathbf{r}) \cdot \mathbf{n} - \epsilon^-(\mathbf{r}) \nabla u^-(\mathbf{r}) \cdot \mathbf{n}, \qquad (3)$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the normal direction of the molecular surface, and $[u] = [\epsilon u_n] = 0$. These conditions are to be rigorously enforced at each intersecting point of the molecular surface and the Cartesian mesh line.

B. Matched interface and boundary method

To solve the linearized PBE with appropriate jump conditions in the finite difference framework, we first classify all the grid points into the regular ones and irregular ones. An irregular grid point is the one where the standard finite difference scheme involves grid points from both inside and outside the interface. To determine fictitious values at irregu-

lar grid points, it is convenient to define a local coordinate (ξ, η, ζ) at a specific intersecting point of the interface and a grid mesh line such that ξ is along the normal direction and η is in the x-y plane. The coordinate transformation can be given as

$$\begin{bmatrix} \xi \\ \eta \\ \zeta \end{bmatrix} = \mathbf{p} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad (4)$$

where $\mathbf{p}$ is the transformation matrix

$$\mathbf{p} = \begin{bmatrix} \sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi \\ -\sin\theta & \cos\theta & 0 \\ -\cos\phi\cos\theta & -\cos\phi\sin\theta & \sin\phi \end{bmatrix}. \qquad (5)$$

Here θ and ϕ are the azimuth and zenith angles with respect to the normal direction $\mathbf{n}$, respectively. In the new coordinates, the jump condition (3) can be written as

$$[\epsilon u_\xi] = \epsilon^+(\sin\phi\cos\theta u_x^+ + \sin\phi\sin\theta u_y^+ + \cos\phi u_z^+)$$
$$- \epsilon^-(\sin\phi\cos\theta u_x^- + \sin\phi\sin\theta u_y^- + \cos\phi u_z^-). \qquad (6)$$

Obviously, two other jump conditions can be generated by differentiating Eq. (2) with respect to the tangential direction η and the binormal direction ζ,

$$[u_\eta] = (-\sin\theta u_x^+ + \cos\theta u_y^+) - (-\sin\theta u_x^- + \cos\theta u_y^-), \qquad (7)$$

$$[u_\zeta] = (-\cos\phi\cos\theta u_x^+ - \cos\phi\sin\theta u_y^+ + \sin\phi u_z^+)$$
$$- (-\cos\phi\cos\theta u_x^- - \cos\phi\sin\theta u_y^- + \sin\phi u_z^-). \qquad (8)$$

In principle, it is possible to generate more jump conditions by further differentiating these low-order jump conditions. However, a by-product of such a procedure is the creation of high-order derivatives, whose evaluation often involves more grid points and is unfavorable. The MIB method makes use of only lower-order jump conditions. Therefore, we end up with four interface equations, i.e., Eqs. (2) and (6)–(8), which can be used to determine four desired unknown quantities related to the intersecting point of the molecular surface and the mesh line. In the MIB method, we extend the computational domains with fictitious values on both sides of the interface so that the standard finite difference scheme can be applied on a smooth domain near the interface without the loss of designed convergence. Fictitious values can be determined by the aforementioned jump conditions. However, in association with four jump conditions there are six derivatives, u_x^+, u_x^-, u_y^+, u_y^-, u_z^+, and u_z^-, which need to be calculated in appropriate subdomains near the interface. For complex geometric constraints, it is often very difficult to compute all of these derivatives. Therefore, in the MIB method, we eliminate two most difficult derivatives by using three relevant jump conditions.

From Eq. (4), we have

$$\begin{bmatrix} u_\xi \\ u_\eta \\ u_\zeta \end{bmatrix} = \mathbf{p} \cdot \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}. \tag{9}$$

Therefore, Eqs. (6)–(8) can be rewritten as follows:

$$\begin{bmatrix} [\beta u_\xi] \\ [u_\eta] \\ [u_\zeta] \end{bmatrix} = \mathbf{C} \cdot \begin{bmatrix} u_x^+ \\ u_x^- \\ u_y^+ \\ u_y^- \\ u_z^+ \\ u_z^- \end{bmatrix}, \tag{10}$$

where

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{C}_2 \\ \mathbf{C}_3 \end{bmatrix}$$

$$= \begin{bmatrix} p_{11}\beta^+ & -p_{11}\beta^- & p_{12}\beta^+ & -p_{12}\beta^- & p_{13}\beta^+ & -p_{13}\beta^- \\ p_{21} & -p_{21} & p_{22} & -p_{22} & p_{23} & -p_{23} \\ p_{31} & -p_{31} & p_{32} & -p_{32} & p_{33} & -p_{33} \end{bmatrix}, \tag{11}$$

where $p_{i,j}$ is the i, jth component of the transformation matrix $\mathbf{p}$ and $\mathbf{C}_i$ represents the ith row of $\mathbf{C}$. After the elimination of the lth and mth elements of $(u_x^+, u_x^-, u_y^+, u_y^-, u_z^+, u_z^-)$, Eq. (10) becomes

$$a[\beta u_\xi] + b[u_\eta] + c[u_\zeta] = (a\mathbf{C}_1 + b\mathbf{C}_2 + c\mathbf{C}_3) \cdot \begin{bmatrix} u_x^+ \\ u_x^- \\ u_y^+ \\ u_y^- \\ u_z^+ \\ u_z^- \end{bmatrix}, \tag{12}$$

where

$$a = C_{2l}C_{3m} - C_{3l}C_{2m},$$

$$b = C_{3l}C_{1m} - C_{1l}C_{3m},$$

$$c = C_{1l}C_{2m} - C_{2l}C_{1m}. \tag{13}$$

We therefore use Eqs. (2) and (12) to determine two fictitious values near the interface along a specific mesh line at a time. This procedure is systematically repeated to determine fictitious values along other two mesh lines and at other interface locations. In this manner, we have effectively reduced a 3D interface problem into a one-dimensional-like one.

Figure 1(a) shows a situation where fictitious values at two irregular points (i,j,k) and $(i,j+1,k)$ are to be determined by jump conditions (2) and (12). Assume that u_x^- and u_z^- are difficult to determine and are eliminated from two jump conditions. Four remaining derivatives, u_y^+, u_y^-, u_x^+, and u_z^+, will be calculated at the intersecting point (x_0, y_0, z_0). Unfortunately, there is no mesh line in the z direction passing
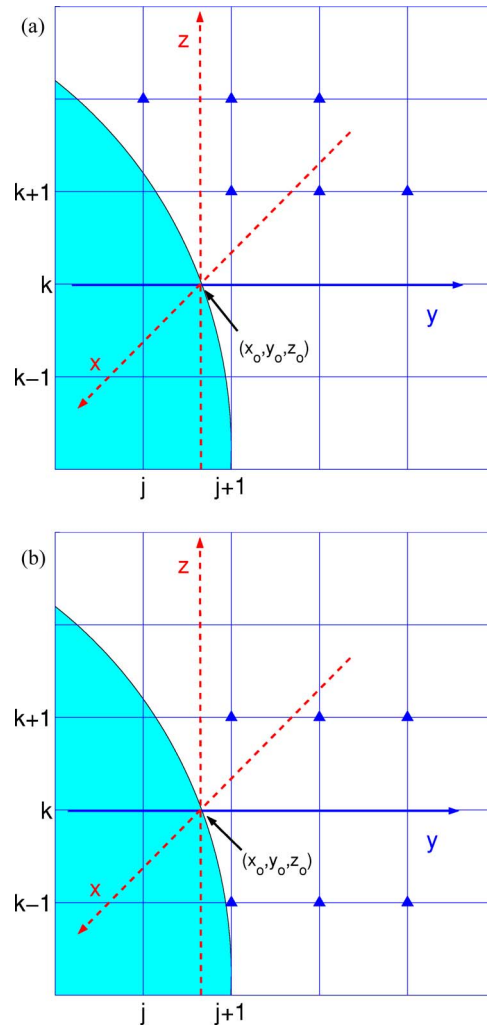

FIG. 1. (Color online) (a) Typical selection of auxiliary points ▲ in MIB I. (b) Typical selection of auxiliary points ▲ in MIB II.

through the interface point (x_0, y_0, z_0). Thus, interpolation schemes are required to calculate u_x^+ and u_z^+. For example, to calculate u_z^+, interpolation is carried out with auxiliary points. There are different ways to choose auxiliary points. The selection of auxiliary points is optimized in the present MIB II scheme as follows. Because of the presence of other singularities, such as singular charge distribution, it is important to avoid possible accuracy reduction due to the interference of two kinds of singular terms at the same set of grid points. Therefore, auxiliary points are selected away from those grid points that carry part singular charges in an interpolation scheme. Nevertheless, this problem can be completely avoided by using Green's formulation to convert the charge singularity into an interface problem.[61] Another important criterion is to select auxiliary points as close to the intersecting point (x_0, y_0, z_0) as possible. For instance, under the same geometric situation depicted in Fig. 1, the MIB I normally selects auxiliary points as shown in Fig. 1(a), while MIB II would select those as shown in Fig. 1(b). As such, the MIB II matrix is optimally symmetrical and much more diagonally dominant than that of the MIB I. Consequently, the MIB II matrix requires much fewer number of iterations than that of the MIB I and leads to a significant reduction in CPU time.

TABLE I. Electrostatic solvation energy ΔG in kcal/mol and the error in the surface potential for a sphere with a centered unit charge. The exact solvation energy is -81.98 kcal/mol.

| Mesh (Å) | ΔG | | | Errors in surface potential | | | | | |
| | | | | MIBPB-II | | PBEQ | | APBS | |
	MIBPB-II	PBEQ	APBS	E_1	E_2	E_1	E_2	E_1	E_2
0.50	−81.99	−85.78	−85.85	2.60	6.68	17.05	84.26	17.06	84.26
0.20	−81.97	−82.84	−82.58	0.30	1.57	7.51	74.44	7.50	74.43
0.10	−81.98	−82.49	−82.27	0.04	0.38	3.84	62.30	3.83	62.30
0.05	−81.98	−82.20	−82.03	0.01	0.09	1.94	46.95	1.89	46.18

The general ideas of the MIB method are as follows. First, simple Cartesian grids are used even if the problem is defined on irregular domains and/or irregular interfaces. Second, the standard central FD schemes are utilized to discretize differential equation with the help of fictitious values on the extended subdomains. Third, physical jump conditions, including the flux jump condition, are explicitly enforced at the intersecting points between the mesh lines and the interface, which, in turn, determines the fictitious values and guarantees the convergence of the FD discretization. Fourth, only lower-order jump conditions are used to avoid the possible involvement of cross derivatives in the higher-order jump conditions and higher dimensional polynomials. Finally, to achieve desired convergence, the lower-order jump conditions are utilized repeatedly. Some key ideas for the treatment of 2D geometric singularities can be found in Ref. 57. Nonetheless, technical details of the MIB II treatment of 3D geometric singularities are significantly more intricate due to vast topological variations of molecular surfaces of proteins. Their description is beyond the scope of the present paper and will be presented elsewhere.[50]

III. RESULTS AND DISCUSSION

In this section, we examine the accuracy and test the convergence order of the present MIB II or MIBPB-II when electrostatic potentials are computed in the presence of geometric singularities. The speed of convergence in terms of the number of iterations N_{BiGG} is also studied. Comparison is made to our earlier MIB I technique[58] and the standard second order FD scheme. However, specific MIB I results presented in this article are generated from a new MIB I code. Moreover, to test the matrix properties of the MIB I and the MIB II, a common linear algebraic solver, the biconjugate gradient (BiGG), was used, and the number of iterations N_{BiGG} required by both methods is compared. Electrostatic potential was solved with the MIBPB-II, and for comparison also with MIBPB-I, PBEQ,[28] a representative finite difference PB solver from CHARMM,[45] and APBS,[34,43,44] a multigrid finite element and finite difference PB solver recently developed primarily for massively parallel computing. The finite difference function of the APBS is utilized in our calculations. Molecular surfaces are generated by using the MSMS program[26] at density 10. In all test cases, the dielectric constant is taken as $\epsilon^- = 1$ and $\epsilon^+ = 80$. The probe radius is set to 1.4 Å unless specified. From the electrostatic potentials in vacuum ϕ_{vac} and in the presence of the dielectric environ-

ment ϕ_{dielec}, the electrostatic free energy of solvation $\Delta G_{\text{solv,elec}}$ is calculated from the explicit charges q_i at positions r_i

$$\Delta G_{\text{solv,elec}} = \frac{1}{2} \sum_i q_i [\phi_{\text{dielec}}(r_i) - \phi_{\text{vac}}(r_i)]. \tag{14}$$

To compare the computational performance, we use three error measurements, the maximum absolute error L_∞, the surface maximum absolute error E_1, and the surface maximum percentage error E_2,

$$L_\infty = \max_\Omega |u(x,y,z) - u_{\text{ex}}(x,y,z)|,$$

$$E_1 = \max_\Gamma |u(x,y,z) - u_{\text{ex}}(x,y,z)|,$$

$$E_2 = 100 \times \max_\Gamma \left| \frac{u(x,y,z) - u_{\text{ex}}(x,y,z)}{u_{\text{ex}}(x,y,z)} \right|, \tag{15}$$

where u and u_{ex} are numerical and exact solutions, respectively. Here E_1 and E_2 are computed over all irregular points near the interface Γ where the modified difference schemes are used. The tolerance of BiCG iterations is set to be 1.0×10^{-6} in all the cases throughout the paper. The order of convergence is calculated for the L_∞ error.

A. Validation

1. Dielectric sphere with a unit charge

To establish the validity and performance of the present MIBPB-II method, we consider a unit charge at the center of a dielectric sphere of 2 Å radius, for which the PBE admits analytical solutions due to Kirkwood.[59] This case has been used to examine the MIBPB-I in our previous studies.[58] In terms of accuracy, the MIBPB-I and MIBPB-II give essentially identical results since there is no geometric singularity. Table I lists the solvation energies and errors in the electrostatic potential of the MIBPB-II in comparison with those of the PBEQ and the APBS. It is observed that in terms of the solvation energy and E_2 error, the MIBPB-II results at a coarse mesh of 0.5 Å are more accurate than those of other two methods at a fine mesh of 0.05 Å. Due to the explicit enforcement of flux jump conditions at the interface, the MIBPB-II is of second order convergence, which is indicated by about fourfold decrease in all three errors as the grid spacing is halved. In contrast, convergence orders of PBEQ and APBS are about 0.3.
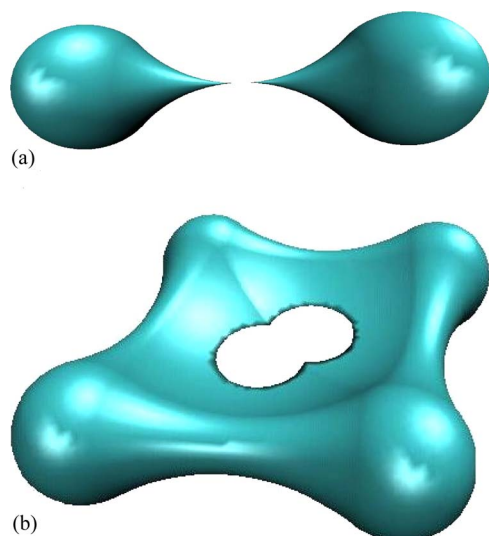
FIG. 2. (Color online) Molecular surface singularities. The radius of carbon atoms is 1.5 Å in both cases. (a) Centers of atoms are $(-3.62, 0, 0)$ and $(3.62, 0, 0)$, and the probe radius is 5.1 Å. (b) Centers of atoms are $(0, 4.2, 0)$, $(0, -4.2, 0)$, $(5, 0, 0)$, and $(-5, 0, 0)$ and the probe radius is 4.9 Å.

2. Molecular surface singularities of few-atom systems

Having established the MIB II method for solving PBE with the molecular surface interface, we test its performance on molecular surface singularities. In the diatomic system, cusps occur in the molecular surface when the atomic distance is enlarged, as shown in Fig. 2(a). Both cusps and self-intersecting surfaces occur in a four-atom system, see Fig. 2(b). These singularities pose challenges to numerical methods and break down the designed second order convergence of our previous MIB I. Note that for the geometries shown in Fig. 2, there is no analytical solution that satisfies PBE (1) with the interface conditions (2) and (3). In order to test the convergence of the present MIB II method, we consider the Poisson equation given by Eq. (1) with $\kappa = 0$. A

standard method to construct an analytically solvable system of the Poisson equation is to allow the source term to vary with the prescribed exact solution. Thus, we set the exact solution to be

$$u^- = 10 \cos x \cos y \cos z + 20, \quad u^+ = 10(x + y + z) + 1.$$
(16)

The source term $f(x, y)$ and interface conditions $[u]$, $[\epsilon u_n]$ can be easily derived from the exact solution. Note that the solutions admit a jump at the interface. To illustrate the performance of the optimized matrix in MIB II, we include a spherical geometry, case I, for which both MIB I and MIB II have similar accuracy but different speeds of convergence.

The numerical errors are computed at three mesh sizes, $h = 0.5$, 0.25, and 0.125 Å, by using MIB I, MIB II, and FD methods. It can be seen from Table II that the second order convergence is achieved by MIB II in all three cases, while MIB I has second order convergence only for case I. MIB I converges slowly and irregularly in cases II and III due to molecular surface singularities. Furthermore, the numerical errors of MIB II are very similar in all cases, which indicates that the geometric singularities have little effect on the accuracy and convergence order of the present MIB II scheme. In contrast, the accuracy of MIB I strongly depends on the location and shape of molecular surface singularities. In case I, MIB I and MIB II produce almost identical results because both methods have the same interface treatment for this case. In cases II and III, L_∞ and E_1 are always the same in MIB I, which indicates that the maximum errors originated from the irregular points near the interface due to the lack of the treatment of geometric singularities. In contrast, L_∞ and E_1 errors are quite different in MIB II in all cases because the largest errors of MIB II occur at the largest value of the solution. At a fine grid, the MIB I errors are about three orders larger than those of the MIB II. The standard FD scheme does not converge at all due to the discontinuous nature of the solution at the interface. Nevertheless, its L_∞ errors and E_1 errors are

TABLE II. Numerical errors of three methods for the Poisson equation. Case I: Sphere of radius 2.0 Å. Case II: Diatomic molecular surface with cusp singularities. Case III: Four-atom molecular surface with cusp and self-intersecting surface singularities.

h (Å)		Case I			Case II			Case III		
		MIB II	MIB I	FD	MIB II	MIB I	FD	MIB II	MIB I	FD
0.5	L_∞	2.1E−1	2.1E−1	6.5E+1	1.4E−1	2.7E+0	1.1E+2	1.4E−1	7.1E+0	1.1E+2
	E_1	3.6E−2	3.5E−2	6.5E+1	5.4E−2	2.7E+0	1.1E+2	8.3E−2	7.1E+0	1.1E+2
	E_2	1.7E−1	1.7E−1	3.2E+2	2.0E−1	3.5E+1	5.1E+2	3.4E+0	6.8E+2	1.2E+3
	N_{BiCG}	91	114	116	98	167	136	130	598	166
0.25	L_∞	5.5E−2	5.5E−2	6.8E+1	3.7E−2	1.7E+0	1.1E+2	3.6E−2	9.1E+2	1.1E+2
	E_1	3.3E−3	3.3E−3	6.8E+1	9.1E−3	1.7E+0	1.1E+2	1.4E−2	9.1E+0	1.1E+2
	E_2	2.4E−2	2.4E−2	3.3E+2	5.5E−2	2.5E+1	5.1E+2	5.7E−1	4.9E+2	1.2E+3
	Order	1.94	1.94	−0.05	1.93	0.69	0.02	1.97	−0.36	−0.07
	N_{BiCG}	120	145	156	128	165	171	153	1125	268
0.125	L_∞	1.4E−2	1.4E−2	6.7E+1	9.4E−3	1.1E+0	1.1E+2	9.1E−3	3.9E+0	1.1E+2
	E_1	4.8E−4	4.8E−4	6.7E+1	1.2E−3	1.1E+0	1.1E+2	1.8E−3	3.9E+0	1.1E+2
	E_2	1.6E−2	1.4E−2	3.2E+2	8.2E−3	1.9E+1	5.1E+2	2.7E−1	1.2E+3	4.7E+3
	Order	1.97	1.97	0.02	1.96	0.63	0.01	1.98	1.23	0.02
	N_{BiCG}	151	475	239	214	313	378	224	6421	323

identical and its errors in cases II and III are larger than its errors in case I. Indeed, the interface and geometric singularities are the main source of errors in conventional Poisson equation solvers.

The number of iterations, N_{BiCG}, increases in all three methods both as the mesh is refined and as interfaces get more complex. Case III requires the largest number of iterations because it has the largest computational domain among these three cases. When the grid spacing is repeatedly halved, which enlarges the matrix size by a factor of 8 each time, N_{BiGG} of MIB II increases slightly from 130 to 153 and 224, while N_{BiGG} of MIB I increases from 598 to 1125 and 6421. The CPU time required for a single iteration of BiGG scheme is roughly the same for the matrices generated by MIB I, MIB II, and FD methods in a given case. It is seen that the N_{BiGG} and thus the CPU time are significantly reduced in the MIB II compared to those of the MIB I, attributing to the new matrix optimization procedure. The FD method requires slightly larger N_{BiGG} than the MIB II method, partially because of the bad condition number of the FD scheme for the discontinuous solution and because of optimally symmetrical matrix of the MIB II method. It is worth mentioning that a common BiGG linear equation solver is used here to test the matrix properties of three methods. In the original MIB I method, the incomplete LU decomposition scheme is utilized to accelerate the speed of solving the linear system,[58] which gives competitive CPU time for matrices of size up to $10^6 \times 10^6$, but is infeasible for larger matrices generated by either a larger computational domain or a smaller mesh spacing for a given domain.

Figure 3 plots the difference between the exact solutions and the numerical ones computed by MIB I and MIB II methods at $h=0.5$ Å. As expected, the MIB I errors are distributed mainly around the geometric singularities. Whereas, the MIB II errors are distributed around the local maximum of the solution. The different scales in the plot indicate the different error magnitudes in two methods. In fact, even larger differences can be observed when the mesh is refined, as shown in Table II.

3. Molecular surface singularities of proteins

The geometric singularities of few-atom systems illustrated in the last subsection are relatively simple. The ultimate computational challenges come from large proteins, which exhibit sophisticated topologies and geometric variations. Their molecular surfaces possess a variety of unnamed geometric singularities, particularly in low resolution protein structures. Therefore, it is important to examine the convergent properties of the present MIB II method for solving the Poisson equation with the molecular surfaces of proteins. To this end, we employ a set of 24 proteins used in previous studies.[21,58] For all structures, extra water molecules are excluded and hydrogen atoms are added to obtain full all-atom models. Their molecular surfaces are computed by using the MSMS program[26] and used as the dielectric interface. We again set the exact solution to that prescribed by Eq. (16). The source terms and the jump conditions of the Poisson equation are derived accordingly for each molecular surface of the 24 proteins.
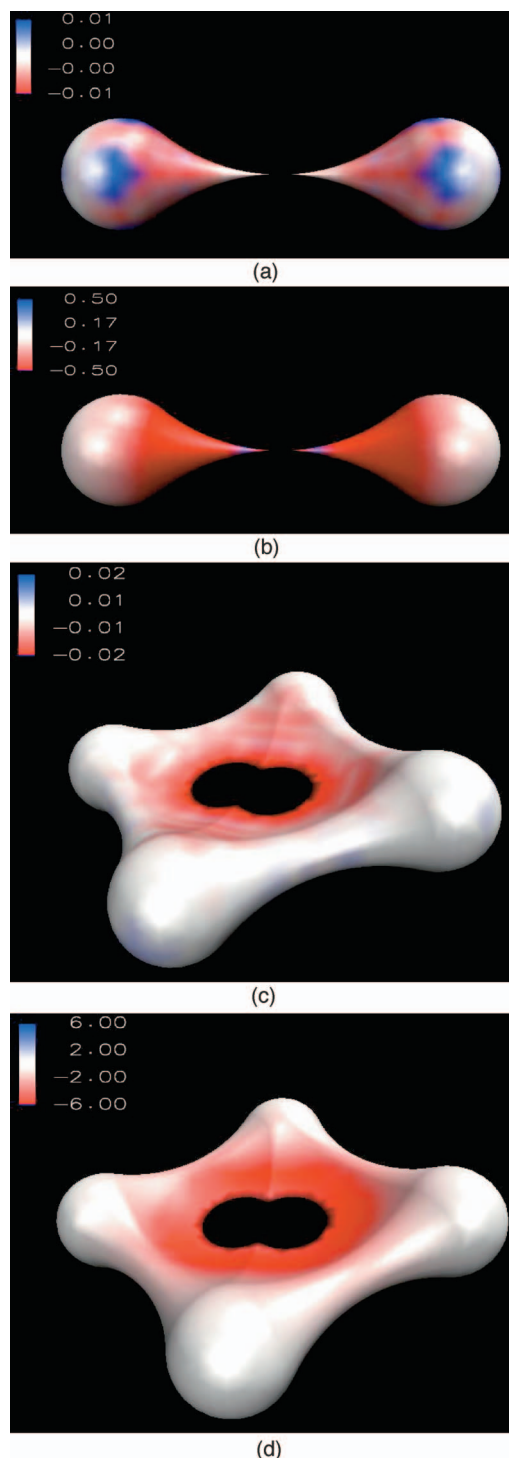

FIG. 3. (Color) The surface projections of numerical errors in MIB I and MIB II. (a) Numerical errors obtained by MIB II for the diatomic system. (b) Numerical errors obtained by MIB I for the diatomic system. (c) Numerical errors obtained by MIB II for the four-atom system. (d) Numerical errors obtained by MIB I for the four-atom system.

The numerical errors and the number of iterations N_{BiGG} at $h=0.5$ Å and $h=0.25$ Å are listed in Table III for both the MIB I and MIB II methods. Although proteins are ordered according to their gyration radii, their numbers of atoms N_a are also listed in the table. In order to compare the performance of these methods, the numerical results of MIB I at $h=0.25$ Å are needed. However, the original MIB I method

TABLE III. Accuracy and convergent tests on the molecular surfaces of 24 proteins.

PDB ID	N_a	MIB	$h=0.5$ Å				$h=0.25$ Å			
			L_∞	E_1	E_2	N_{BiCG}	L_∞ (Order)	E_1	E_2	N_{BiCG}
1ajj	519	II	$2.4E-1$	$1.1E-1$	$9.3E-1$	278	$6.0E-2(1.98)$	$1.7E-2$	$1.6E-1$	486
		I	$2.5E-1$	$2.5E-1$	$1.4E+0$	3 809	$8.6E-2(153)$	$8.6E-2$	$3.3E-1$	
2pde	667	II	$2.3E-1$	$1.2E-1$	$8.9E-1$	306	$5.9E-2(1.97)$	$1.9E-2$	$2.3E-1$	498
		I	$4.7E+0$	$4.7E+0$	$2.3E+1$	4 444	$1.7E+0(1.47)$	$1.7E+0$	$9.4E+0$	
1vii	596	II	$2.4E-1$	$1.4E-1$	$1.4E+0$	273	$6.0E-2(199)$	$1.9E-2$	$2.0E-1$	488
		I	$3.2E+0$	$3.2E+0$	$6.3E+1$	4 560	$1.5E+0(1.14)$	$1.5E+0$	$7.4E+0$	
2erl	573	II	$2.3E-1$	$1.1E-1$	$4.2E+0$	265	$5.8E-2(1.98)$	$1.8E-2$	$2.0E-1$	491
		I	$2.3E-1$	$1.1E-1$	$4.2E+0$	1 888	$6.2E-2(1.89)$	$6.2E-2$	$2.3E-1$	
1cbn	648	II	$2.3E-1$	$1.4E-1$	$9.0E-1$	295	$6.0E-2(1.95)$	$1.8E-2$	$1.5E-1$	497
		I	$2.3E-1$	$1.4E-1$	$9.0E-1$	5 682	$8.6E-2(1.43)$	$8.6E-2$	$3.7E-1$	
1bor	832	II	$2.3E-1$	$1.3E-1$	$6.3E+0$	319	$5.9E-2(1.98)$	$1.9E-2$	$1.8E-1$	538
		I	$2.3E-1$	$2.2E-1$	$6.3E+0$	17 307	$1.1E+0(-2.27)$	$1.1E+0$	$4.1E+0$	
1bbl	576	II	$2.4E-1$	$1.1E-1$	$9.3E-1$	322	$6.0E-2(2.01)$	$1.8E-2$	$2.4E-1$	492
		I	$3.4E+0$	$3.4E+0$	$8.1E+1$	17 307	$2.3E-1(3.89)$	$2.3E-1$	$1.2E-0$	
1fca	729	II	$2.4E-1$	$1.2E-1$	$1.2E+0$	249	$6.0E-2(1.99)$	$2.2E-2$	$2.0E-1$	423
		I	$2.4E-1$	$1.2E-1$	$1.1E+0$	4 145	$6.2E-2(1.96)$	$4.1E-2$	$2.1E-1$	
1uxc	809	II	$2.4E-1$	$1.3E-1$	$1.5E+0$	281	$6.0E-2(2.00)$	$1.9E-2$	$1.7E-1$	516
		I	$2.4E-1$	$1.3E-1$	$1.5E+0$	4 031	$6.0E-2(2.00)$	$1.9E-2$	$1.7E-1$	
1sh1	702	II	$2.4E-1$	$1.2E-1$	$9.0E-1$	289	$6.0E-2(2.00)$	$1.7E-2$	$1.5E-1$	525
		I	$3.6E+0$	$3.6E+0$	$1.1E+2$	10 366	$3.9E+0(-0.13)$	$3.9E+0$	$2.5E+1$	
1mbg	903	II	$2.3E-1$	$1.1E-1$	$1.8E+0$	290	$5.8E-2(1.97)$	$1.7E-2$	$1.7E-1$	487
		I	$6.7E+0$	$6.7E+0$	$2.3E+0$	4 604	$7.7E-1(3.12)$	$7.7E-1$	$3.3E+0$	
1ptq	795	II	$2.4E-1$	$1.2E-1$	$1.1E+0$	289	$5.9E-2(2.00)$	$1.8E-2$	$1.7E-1$	525
		I	$2.4E-1$	$1.2E-1$	$1.1E+0$	6 291	$6.9E-2(1.77)$	$6.9E-2$	$3.7E-1$	
1vjw	828	II	$2.3E-1$	$1.2E-1$	$1.1E+0$	283	$5.9E-2(1.95)$	$1.9E-2$	$1.8E-1$	443
		I	$2.3E-1$	$1.2E-1$	$1.1E+0$	4 490	$5.9E-2(1.95)$	$3.2E-2$	$1.8E-1$	
1fxd	824	II	$2.3E-1$	$1.2E-1$	$1.7E+0$	286	$5.9E-2(1.99)$	$1.9E-2$	$1.7E-1$	510
		I	$2.3E-1$	$2.3E-1$	$1.7E+0$	3 852	$1.5E-1(0.63)$	$1.5E-1$	$9.1E-1$	
1r69	997	II	$2.3E-1$	$1.2E-1$	$2.9E+0$	303	$5.9E-2(1.99)$	$1.9E-2$	$1.8E-1$	514
		I	$2.3E-1$	$1.2E-1$	$2.9E+0$	4 912	$5.9E-2(1.99)$	$2.8E-2$	$2.2E-1$	
1hpt	858	II	$2.4E-1$	$1.1E-1$	$1.1E+0$	292	$6.1E-2(1.97)$	$1.9E-2$	$1.6E-1$	502
		I	$2.3E-1$	$1.7E-1$	$1.1E+0$	11 346	$1.0E+0(-5.47)$	$1.0E+0$	$4.4E+0$	
1bpi	898	II	$2.4E-1$	$1.3E-1$	$1.1E+0$	302	$6.1E-2(1.97)$	$1.9E-2$	$1.6E-1$	517
		I	$6.5E+0$	$6.5E+0$	$3.0E+1$	10 396	$4.1E+0(0.66)$	$4.1E+0$	$2.1E+1$	
451c	1216	II	$2.4E-1$	$1.5E-1$	$6.0E+0$	342	$6.0E-2(2.02)$	$1.9E-2$	$2.3E-1$	563
		I	$6.6E+0$	$6.6E+0$	$3.3E+1$	10 049	$3.9E+0(0.75)$	$3.9E+0$	$2.1E+1$	
1a2s	1272	II	$2.3E-1$	$1.3E-1$	$2.4E+0$	379	$5.8E-2(1.98)$	$1.9E-2$	$4.0E-1$	651
		I	$2.3E-1$	$2.1E-1$	$2.5E+0$	41 290	$2.7E-1(-0.25)$	$2.7E-1$	$1.6E+0$	
1frd	1478	II	$2.4E-1$	$1.2E-1$	$3.1E+0$	298	$6.0E-2(2.03)$	$1.9E-2$	$6.6E-1$	503
		I	$1.6E+0$	$1.6E+0$	$4.4E+0$	17 263	$1.5E-1(3.45)$	$1.5E-1$	$8.9E-1$	
1svr	1435	II	$2.4E-1$	$1.2E-1$	$1.5E+0$	391	$2.9E-2(2.01)$	$2.0E-2$	$2.0E-1$	678
		I	$2.7E+0$	$2.7E+0$	$9.4E+0$	77 080	$1.2E+0(1.12)$	$1.2E+0$	$5.4E+0$	
1neq	1187	II	$2.4E-1$	$1.3E-1$	$1.2E+0$	369	$5.9E-2(2.00)$	$1.9E-2$	$1.6E-1$	688
		I	$1.4E+0$	$1.4E+0$	$3.2E+0$	17 894	$1.5E+0(-0.13)$	$1.5E+0$	$6.4E+0$	

TABLE III. (*Continued.*)

PDB ID	N_a	MIB	$h=0.5$ Å				$h=0.25$ Å			
			L_∞	E_1	E_2	N_{BiCG}	L_∞ (Order)	E_1	E_2	N_{BiCG}
1a63	2065	II	2.4E−1	1.3E−1	2.2E+0	434	6.0E−2(1.99)	2.0E−2	2.4E−1	665
		I	4.6E+0	4.6E+0	4.0E+1	83 477	1.1E−1(5.33)	1.1E−1	5.9E−1	
1a7m	2890	II	2.5E−1	1.3E−1	4.0E+0	419	6.7E−2(1.90)	2.2E−2	6.3E−1	726
		I	8.4E+0	8.4E+0	4.1E+2	414 908	5.0E+0(0.73)	5.0E+0	2.3E+2	

does not converge at such a grid resolution.[58] We therefore have applied the matrix optimization procedure to the MIB I and refer to such a scheme as "accelerated MIB I." The difference between the MIB II and the accelerated MIB I is that the latter does not have the present treatment of geometric singularities. The MIB II results show great consistency on all 24 molecular surfaces. With the same mesh size, the numerical errors are almost the same, which indicates the robustness of MIB II. Comparing the L_∞ and the E_1 in the table, one may notice that the L_∞ error is always greater than the E_1 error for MIB II in all cases, indicating that the molecular surface singularities are not the source of the maximal error. In contrast, the L_∞ error is the same as the E_1 error for MIB I in most cases. At the grid resolution $h=0.5$ Å, the MIB I has the largest E_2 error for 1svr, indicating some special geometric singularity in the molecular surface of 1svr. Furthermore, the N_{BiCG} required by the original MIB I at $h=0.5$ Å is also provided in Table III for a comparison of convergence speed. As it can be seen, the N_{BiCG} of MIB I can be orders of magnitude larger than that of MIB II. As the number of atoms N_a increases, the N_{BiCG} of MIB II increases slightly. For example, the number of interactions N_{BiCG} only increases less than twice when the matrix size is eight times larger due to halving the mesh. This confirms the excellent matrix properties of the present MIB II method.

Figure 4 shows the convergence patterns of MIB I and MIB II over molecular surfaces of 24 proteins, which are sorted in the same order as that in Table III. The second order convergence of the MIB II on the molecular surfaces of 24 proteins is depicted in Fig. 4(a). The numerical errors of the MIB II method obtained at $h=0.25$ Å and $h=0.5$ Å are denoted by dots and stars, respectively. The figure indicates the uniform second order convergence of MIB II. Figure 4(b) shows the numerical errors of the MIB I obtained at $h=0.5$ Å and accelerated MIB I obtained at $h=0.25$ Å. The convergence pattern of the MIB I is irregular and depends on the occurrence of molecular surface singularities. It is worth noting that at the coarse grid, $h=0.5$ Å, the MIB I has similar accuracy as MIB II for 12 protein molecular surfaces. When the mesh is refined to $h=0.25$ Å, the accelerated MIB I has similar accuracy as MIB II has only in five cases (1fca, 1r69, 1uxr, 1vjw, and 2erl). It implies that there are very few molecular surface singularities in the molecular surfaces of these five proteins. However, geometric singularities commonly occur in the molecular surfaces of most proteins.

B. Applications

1. Solvation free energy

After the confirmation of the second order convergence of the proposed MIB II scheme, we now explore the impact of this new algorithm to the numerical solution of the PB equation for the electrostatic analysis of proteins. The MIB II based PB solver is denoted as MIBPB-II and its results are compared with those of our earlier MIBPB-I,[58] the PBEQ,[28] and the APBS.[34,43,44] The same set of proteins[21,58] used in the last subsection is employed for the present study. For all
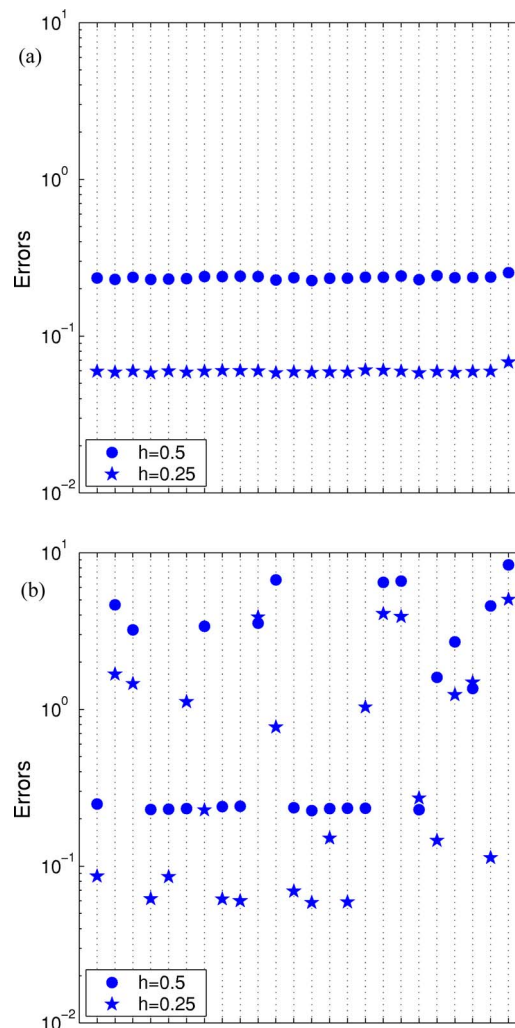

FIG. 4. (Color online) Accuracy test on molecular surface of 24 proteins. (a) L_∞ errors obtained by MIB II. (b) L_∞ errors obtained by the accelerated MIB I.

TABLE IV. Electrostatic solvation energies ΔG in kcal/mol calculated by using the MIBPB-I, MIBPB-II, APBS, and PBEQ. CPU times used by MIBPB-II at $h=0.5$ Å and by APBS at $h \sim 0.2$ Å are given.

h (Å)	MIBPB-II		MIBPB-I	APBS		PBEQ			CPU time (s)	
	0.5	0.25	0.5	0.4	0.2	0.5	0.25	0.15	MIBPB-II	APBS
1ajj	−1136.9	−1136.6	−1139.5	−1158.0	−1138.6	−1170.7	−1145.7	−1138.8	51	77
2pde	−819.5	−820.9	−820.6	−829.7	−819.6	−849.3	−832.2	−824.0	66	97
1vii	−902.5	−901.6	−901.5	−924.1	−904.4	−936.8	−914.1	−905.0	51	83
2erl	−950.2	−948.8	−949.6	−970.9	−951.8	−980.5	−959.1	−953.0	48	77
1cbn	−305.4	−303.8	−304.4	−315.4	−305.4	−327.6	−310.8	−307.0	63	97
1bor	−854.1	−854.6	−853.4	−870.3	−855.3	−890.7	−865.7	−857.0	86	119
1bbl	−994.2	−987.5	−986.6	−1007.1	−991.1	−1028.4	−1000.5	−993.0	70	108
1fca	−1201.3	−1199.9	−1202.5	−1222.1	−1208.6	−1244.3	−1220.4	−1213.0	59	102
1uxc	−1141.6	−1138.8	−1146.5	−1163.7	−1142.4	−1192.6	−1154.1	−1144.0	76	112
1shl	−754.6	−753.4	−753.4	−778.3	−763.3	−803.2	−772.2	−764.0	63	84
1mbg	−1350.9	−1349.9	−1353.5	−1372.3	−1353.2	−1398.3	−1362.5	−1355.0	86	109
1ptq	−871.7	−872.4	−871.8	−898.1	−875.1	−915.0	−886.4	−877.0	77	112
1vjw	−1237.9	−1237.9	−1243.9	−1265.4	−1246.9	−1291.5	−1256.9	−1247.0	75	98
1fxd	−3299.8	−3300.2	−3302.8	−3330.2	−3309.4	−3348.7	−3323.5	−3313.0	86	123
1r69	−1088.9	−1087.7	−1088.1	−1114.7	−1093.8	−1142.7	−1103.3	−1093.0	93	115
1hpt	−811.6	−813.1	−811.2	−837.4	−818.5	−867.0	−830.9	−820.0	104	112
1bpi	−1304.0	−1302.0	−1302.5	−1339.8	−1311.2	−1356.6	−1318.9	−1307.0	95	167
451c	−1025.0	−1024.7	−1026.3	−1058.8	−1030.5	−1084.2	−1045.0	−1033.0	128	163
1a2s	−1912.8	−1913.6	−1911.6	−1948.6	−1921.1	−1968.5	−1933.0	−1920.0	165	173
1frd	−2852.2	−2851.5	−2852.4	−2904.4	−2865.4	−2941.9	−2884.0	−2867.0	183	⋯
1svr	−1713.5	−1711.5	−1709.3	−1753.7	−1720.6	−1784.9	−1740.0	−1712.0	195	⋯
1neq	−1730.7	−1729.7	−1728.8	−1777.7	−1735.5	−1800.7	−1752.0	−1738.0	189	⋯
1a63	−2375.5	−2373.7	−2372.0	−2428.6	−2389.5	−2496.1	−2416.0	−2394.0	383	⋯
1a7m	−2158.0	−2157.3	−2156.5	−2211.5	−2166.4	−2260.1	−2193.0	−2169.0	541	⋯

structures hydrogen atoms were added to obtain full all-atom models. Partial charges at atomic sites and atomic van der Waals radii defining the dielectric boundary were taken from the CHARMM22 force field.[62] The solvation free energies are calculated by MIBPB-II at the grid spacings of 0.5 and 0.25 Å and by PBEQ at 0.5, 0.25, and 0.15 Å. The APBS results are reported at a coarse grid spacing of about 0.4 Å and a fine grid spacing of about 0.2 Å; however, it allows grid adjustments from 0.33 to 0.48 Å for the coarse grid and from 0.19 to 0.21 Å for the fine grid. The results of MIBPB-I were calculated at the grid spacing of 0.5 Å.[58] Due to its unfavorable matrix, no result at a refined grid could be produced. These results are listed in Table IV in the order of gyration radii for proteins. The same order is used in all figures in the following analysis.

Figure 5(a) gives a bird's-eye view of the performance of these methods. It is seen that at a coarse scale of thousands kcal/mol, the results of three methods are in good agreement with each other. Note that results from finer grids are selected for APBS and PBEQ to achieve this effect.

From Table IV, it is seen that MIBPB-II results at a coarse grid, $h=0.5$ Å, are very similar to those at the refined grid $h=0.25$ Å. Indeed, this is true as shown in Fig. 5(b), where differences of these two types of solutions are plotted. Except for one case, 1bbl, the MIBPI-II differences between two grids are all within 5 kcal/mol, disregarding the size of and the radius of gyration of the protein. This means that the MIBPB-II is well converged at a coarse grid of $h=0.5$ Å. However, this is not the case for APBS and PBEQ, as shown

in Fig. 5(b). Their differences between two grids vary dramatically, in the magnitude of 10–80 kcal/mol, implying that their results at the coarse grid are not converged yet. In particular, their differences are apparently larger for proteins of larger radii of gyration, indicating that their convergence and reliability depend on the radius of gyration or, loosely, on the size of the system. From Table IV, it is seen that APBS and PBEQ results are generally more negative than those of MIBPB-II. An interesting observation is that, as the grid is refined, the results of both APBS and PBEQ converge toward those of the MIBPB-II.

Since the results of APBS and PBEQ converge to those of MIBPB-II, which are well converged at $h=0.25$ Å, we choose the solvation free energies of the MIBPB-II as references to show the convergence property of other methods. In Fig. 5(c), the differences in the solvation free energies are plotted between PBEQ and MIBPB-II, APBS and MIBPB-II, and MIBPB-I and MIBPB-II. The results of MIBPB-I calculated from a coarse grid of $h=0.5$ Å are essentially converged, confirming our earlier claim.[58] The PBEQ method at the grid resolution of $h=0.25$ Å produces the largest differences in solvation free energies, as shown in Fig. 5(c). The magnitudes of such differences range from a few kcal/mol to 43 kcal/mol and exhibit a growing trend in the increase of the radius of gyration. This indicates the lack of reliability of the PBEQ method at the grid resolution of $h=0.25$ Å in practical applications. Only when the grid is refined to $h=0.15$ Å could PBEQ further reduce its differences to less than 20 kcal/mol, as indicated in Table IV. However, such a
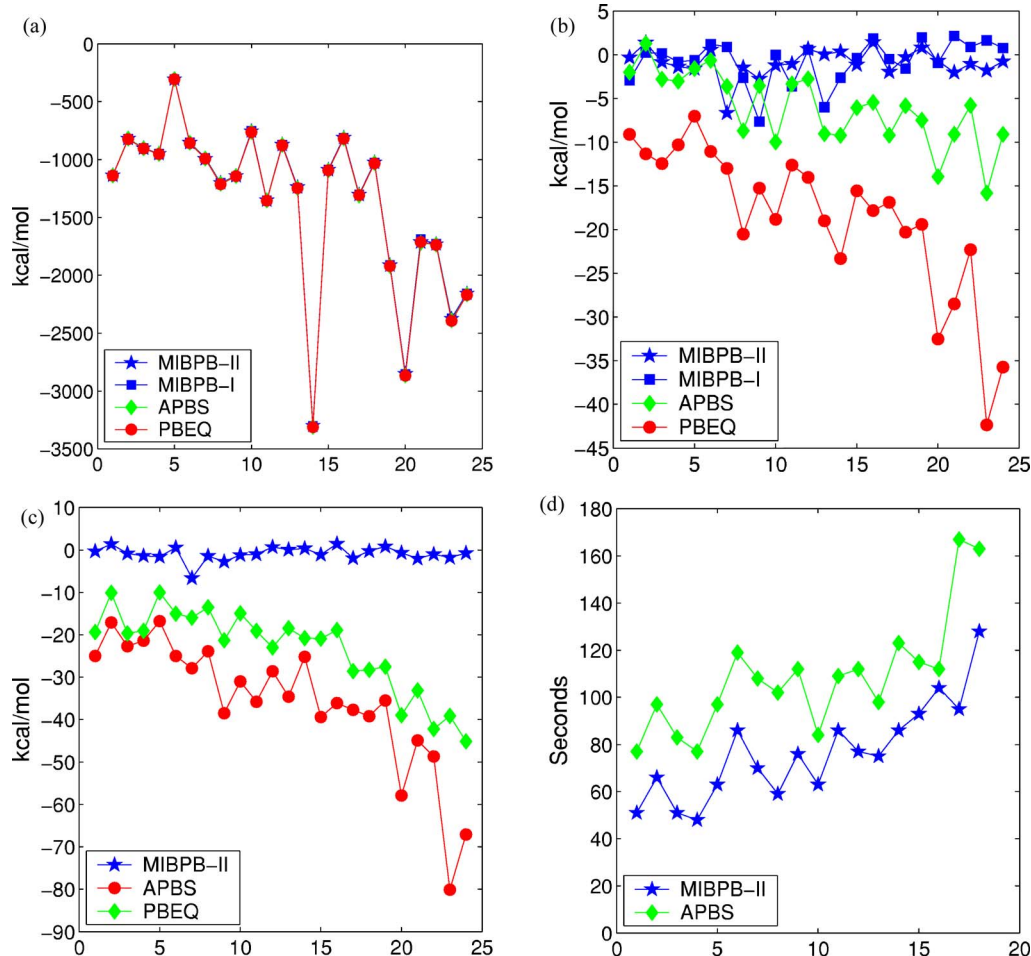
FIG. 5. (Color online) Comparison of solvation free energies of proteins, which are listed in the order of gyration radii, as shown in Table IV. (a) Solvation free energies of $\Delta G_{MIBPB-II}(h=0.5$ Å$)$, $\Delta G_{MIBPB-I}(h=0.5$ Å$)$, $\Delta G_{APBS}(h\sim0.2$ Å$)$, and $\Delta G_{PBEQ}(h=0.25$ Å$)$. (b) Differences of solvation free energies between coarse mesh and fine mesh, i.e., $\Delta G_{MIBPB-II}(h=0.5$ Å$)-\Delta G_{MIBPB-II}(h=0.25$ Å$)$, $\Delta G_{MIBPB-I}(h=0.5$ Å$)-\Delta G_{MIBPB-I}(h=0.25$ Å$)$, $\Delta G_{PBEQ}(h=0.5$ Å$)-\Delta G_{PBEQ}(h=0.25$ Å$)$, and $\Delta G_{APBS}(h\sim0.4$ Å$)-\Delta G_{APBS}(h\sim0.2$ Å$)$. (c) Differences of solvation free energies between MIBPB-II and other methods, i.e., $\Delta G_{MIBPB-II}(h=0.5$ Å$)-\Delta G_{MIBPB-II}(h=0.25$ Å$)$, $\Delta G_{MIBPB-I}(h=0.5$ Å$)-\Delta G_{MIBPB-II}(h=0.25$ Å$)$, $\Delta G_{PBEQ}(h=0.25$ Å$)-\Delta G_{MIBPB-II}(h=0.25$ Å$)$, and $\Delta G_{APBS}(h\sim0.2$ Å$)-\Delta G_{MIBPB-II}(h=0.25$ Å$)$. (d) CPU time used by APBS at about 0.2 Å and MIBPB-II at 0.5 Å for 18 proteins.

grid resolution may not be very practical in most electrostatic analysis due to large computational costs. The results of APBS obtained at $h\sim0.2$ Å generally show much smaller differences and a weak dependence on the radius of gyration, partly due to its ability of adaptation to the interface. However, MIBPB-II is more accurate at the grid resolution $h=0.5$ Å than the APBS at finer grid of $h\sim0.2$ Å.

It is interesting to compare the CPU cost of the MIBPB-II with that of APBS at a similar level of accuracy for computing the solvation free energy. Figure 5(d) depicts the CPU time of MIBPB-II and APBS at the grid resolutions of $h=0.5$ Å and $h\sim0.2$ Å, respectively. A dedicated personal computer of 2.8 GHz Pentium D CPU with 2 Gbyte RAM was used for the CPU time comparison. However, the CPU time of the APBS for the last six largest proteins, 1neq, 1a2s, 1svr, 1frd, 1a63, and 1a7m, is inaccessible because their memory requirements are either extremely close to or exceed the 2 Gbyte limit of the dedicated WIN32 system. The computation of these six proteins is therefore completed in a high performance computer where the precise CPU time cannot be recorded because of disturbance from other users. It is to be noted that at the same grid resolution, the present

MIBPB-II solver with a biconjugate gradient linear solver requires more CPU time than the APBS solver. However, at a given accuracy as shown in Fig. 5(c), the MIBPB-II requires less CPU time than APBS.

2. Electrostatic potentials

Finally, we consider the surface electrostatic potential of a heme-binding protein, Fe(II) cytochrome C551 from the organism Pseudomonas aeruginosa (PDB ID: 451c). The electrostatics distribution near the heme-bonding site and inside the binding cavity is very important for the electrostatic steering effect during the protein-heme docking process. The potential is computed with MIBPB-II, MIBPB-I, and PBEQ at the grid resolution of $h=0.5$ Å. Figure 6 illustrates the surface electrostatic potential obtained with MIBPB-II, and for a comparison, the differences between the surface electrostatic potentials of MIBPB-II and MIBPB-I, and between MIBPB-II and PBEQ. It is seen that the inner surface of the cavity is mostly positively charged except of middle section there the Fe(II) ion would locate, which is consistent with the fact that the heme surface is mostly negatively charged be-
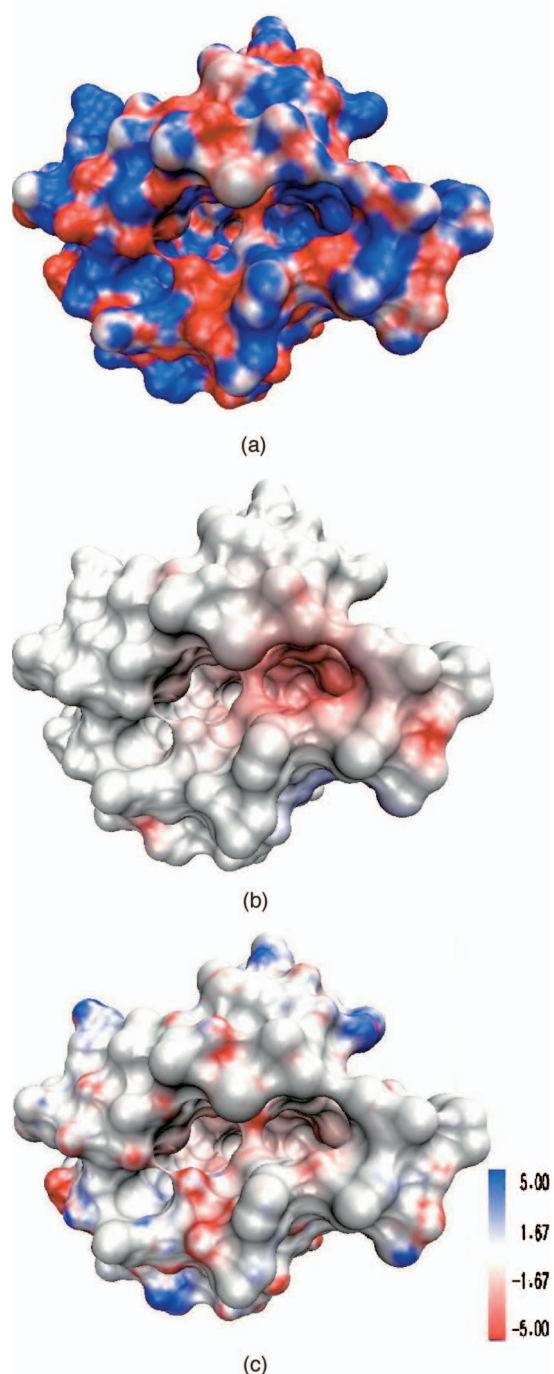
FIG. 6. (Color) Comparison of surface electrostatic potentials ϕ in kcal/mol e of cytochrome C551 at $h=0.5$ Å. (a) $\phi_{\text{MIBPB-II}}$, (b) $\phi_{\text{MIBPB-II}} - \phi_{\text{MIBPB-I}}$, and (c) $\phi_{\text{MIBPB-II}} - \phi_{\text{PBEQ}}$.

cause it has a positively charged Fe(II) core. From the difference plots, we found that MIBPB-I differs from MIBPB-II only for a few isolated spots due to the molecular surface singularities. The light color in these spots indicates small differences. However, the discrepancies between MIBPB-II and PBEQ are more intensive and mostly distributed around convex surfaces where there are more untreated irregular grid points. We expect that these discrepancies of about 5 kcal/mol e would have a significant consequence in a quantitative analysis of the interaction between cytochrome C and heme.

IV. CONCLUSION

This article reports a new generation of interface based Poisson-Boltzmann (PB) equation solvers that take explicit care of geometric singularities, such as cusps and self-intersecting surfaces in the molecular surface definition.[22] Our previous matched interface and boundary (MIB) method based PB solver, denoted as MIBPB-I, was the first PB solver to explicitly enforce the flux jump conditions at solvent-solute interfaces and could consequently provide highly accurate biomolecular electrostatics in continuum dielectric environments.[58] However, the MIBPB-I cannot maintain its designed second order convergence whenever there are geometric singularities. Moreover, the MIBPB-I matrix is not optimally symmetrical and diagonally dominant, resulting in a severe convergence problem. The present method, denoted as MIBPB-II, is designed to overcome the aforementioned difficulties. A new MIB scheme is proposed to rigorously treat arbitrarily complex interfaces in the Cartesian representation.[57,60] The MIB matrix is made optimally symmetrical and diagonally dominant. The proposed MIBPB-II is extensively validated by the molecular surfaces of few-atom systems and a set of 24 proteins. Uniform second order convergence of MIBPB-II is confirmed for singular molecular surface interfaces of proteins, disregarding the radius of gyration. The MIBPB-II matrix is optimally symmetrical, which leads to a significant reduction in the number of interactions required by a linear equation solver.

The MIBPB-II is applied to electrostatic potential calculations of a dielectric sphere with a unit charge, which admits an analytical solution,[59] and a set of 24 proteins. As a comparison, the finite difference based PBEQ and APBS are employed for the same electrostatic calculations. For the analytical case, the MIBPB-II at a coarse grid of $h=0.5$ Å was found to be more accurate than the PBEQ and APBS at a fine grid of $h=0.05$ Å in terms of both surface electrostatic potential and solvation energy. For 24 proteins, we found that the solvation free energies computed by the MIBPB-II have essentially converged at the grid resolution of $h=0.5$ Å. The results of APBS and PBEQ converge toward those of the MIBPB-II when their grids are refined. The discrepancies between converged MIBPB-II solvation free energies, which are obtained at the grid of $h=0.5$ Å, and those of PBEQ obtained at $h=0.15$ Å are nearly 20 kcal/mol for proteins with a large radius of gyration. Slightly smaller discrepancies are found between converged MIBPB-II solvation free energies and those of APBS obtained at about $h \sim 0.2$ Å. The accuracy of both PBEQ and APBS shows a dependence on the radius of gyration. The present MIBPB-II requires more CPU time than APBS and PBEQ at a given grid resolution but less CPU time at a given accuracy.

As the interface flux jump conditions are rigorously enforced at complex solvent-solute interfaces and convergence order is systematically restored at geometric singularities, the proposed MIBPB-II solver ought to deliver more reliable surface electrostatic potentials. Indeed, converged surface electrostatic potentials are obtained at the resolution of $h=0.5$ Å. The discrepancies between converged MIBPB-II surface electrostatic potentials and those of PBEQ at the

same grid resolution are about 5 kcal/mol e. Such discrepancies would induce a consequence in a quantitative analysis. A rigorous Green's function treatment of singular charge distributions is under our consideration.[61]

[1] P. Koehl, Curr. Opin. Struct. Biol. **16**, 142 (2006).
[2] C. L. Vizcarra and S. L. Mayo, Curr. Opin. Struct. Biol. **9**, 622 (2005).
[3] N. A. Baker, Curr. Opin. Struct. Biol. **15**, 137 (2005).
[4] M. Feig and C. L. Brooks III, Curr. Opin. Struct. Biol. **14**, 217 (2004).
[5] C. M. Cortis and R. A. Friesner, J. Comput. Chem. **18**, 1591 (1997).
[6] K. E. Forsten, R. E. Kozack, D. A. Lauffenburger, and S. Subramaniam, J. Phys. Chem. **98**, 5580 (1994).
[7] B. Honig and A. Nicholls, Science **268**, 1144 (1995).
[8] B. Roux and T. Simonson, Biophys. Chem. **78**, 1 (1999).
[9] K. A. Sharp and B. Honig, Annu. Rev. Biophys. Biophys. Chem. **19**, 301 (1990).
[10] Y. N. Vorobjev and H. A. Scheraga, J. Comput. Chem. **18**, 569 (1997).
[11] R. J. Zauhar and R. S. Morgan, J. Mol. Biol. **186**, 815 (1985).
[12] J. Warwicker and H. C. Watson, J. Mol. Biol. **154**, 671 (1982).
[13] F. Fogolari, A. Brigo, and H. Molinari, Biophys. J. **85**, 159 (2003).
[14] P. A. Kollman I. Massova, C. Reyes *et al.*, Acc. Chem. Res. **33**, 889 (2000).
[15] J. M. J. Swanson, R. H. Henchman, and J. A. McCammon, Biophys. J. **86**, 67 (2004).
[16] R. E. Georgescu, E. G. Alexov, and M. R. Gunner, Biophys. J. **83**, 1731 (2002).
[17] J. E. Nielsen and J. A. McCammon, Protein Sci. **12**, 313 (2003).
[18] J. Warwicker, Protein Sci. **13**, 2793 (2004).
[19] Q. Lu and R. Luo, J. Chem. Phys. **119**, 11035 (2003).
[20] N. V. Prabhu, P. Zhu, and K. A. Sharp, J. Comput. Chem. **25**, 2049 (2004).
[21] M. Feig, A. Onufriev, M. S. Lee, W. Im, D. A. Case, and C. L. Brooks III, J. Comput. Chem. **25**, 265 (2004).
[22] B. Lee and F. M. Richards, J. Mol. Biol. **55**, 379 (1971).
[23] M. L. Connolly, J. Appl. Crystallogr. **18**, 499 (1985).
[24] F. Eisenhaber and P. Argos, J. Comput. Chem. **14**, 1272 (1993).
[25] V. Gogonea and E. Osawa, Supramol. Chem. **3**, 303 (1994).
[26] M. F. Sanner, A. J. Olson, and J. C. Spehner, Biopolymers **38**, 305 (1996).
[27] J. A. Grant, B. T. Pickup, and A. Nicholls, J. Comput. Chem. **22**, 608 (2001).
[28] W. Im, D. Beglov, and B. Roux, Comput. Phys. Commun. **111**, 59 (1998).
[29] J. M. J. Swanson, J. Mongan, and J. A. McCammon, J. Phys. Chem. **109**, 14769 (2005).
[30] B. Z. Lu, W. Z. Chen, C. X. Wang, and X. J. Xu, Proteins **48**, 497 (2002).
[31] R. Luo, L. David, and M. K. Gilson, J. Comput. Chem. **23**, 1244 (2002).
[32] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon, Proc. Natl. Acad. Sci. U.S.A. **98**, 10037 (2001).
[33] N. A. Baker, D. Sept, M. J. Holst, and J. A. McCammon, IBM J. Res. Dev. **45**, 427 (2001).
[34] M. Holst and F. Saied, J. Comput. Chem. **14**, 105 (1993).
[35] A. Bordner and G. Huber, J. Comput. Chem. **24**, 353 (2003).
[36] A. Boschitsch, M. Fenley, and H.-X. Zhou, J. Phys. Chem. B **106**, 2741 (2002).
[37] A. Boschitsch and M. Fenley, J. Comput. Chem. **25**, 935 (2004).
[38] A. Juffer, E. Botta, B. van Keulen, A. van der Ploeg, and H. Berendsen, J. Comput. Phys. **97**, 144 (1991).
[39] J. Liang and S. Subranmaniam, Biophys. J. **73**, 1830 (1997).
[40] I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig, Proteins **1**, 47 (1986).
[41] W. Rocchia, E. Alexov, and B. Honig, J. Phys. Chem. B **105**, 6507 (2001).
[42] M. E. Davis, J. D. Madura, B. A. Luty, and J. A. McCammon, Comput. Phys. Commun. **62**, 187 (1991).
[43] M. Holst, N. Baker, and F. Wang, J. Comput. Chem. **21**, 1319 (2000).
[44] N. Baker, M. Holst, and F. Wang, J. Comput. Chem. **21**, 1343 (2000).
[45] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, J. Comput. Chem. **4**, 187 (1983).
[46] C. M. Cortis and R. A. Friesner, J. Comput. Chem. **18**, 1570 (1997).
[47] P. E. Dyshlovenko, Comput. Phys. Commun. **147**, 335 (2002).
[48] C. S. Peskin, J. Comput. Phys. **25**, 220 (1977).
[49] R. J. LeVeque and Z. L. Li, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **31**, 1019 (1994).
[50] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, J. Comput. Phys. **152**, 457 (1999).
[51] I. Babuška, Computing **5**, 207 (1970).
[52] A. Mayo, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **21**, 285 (1984).
[53] S. Zhao and G. W. Wei, J. Comput. Phys. **200**, 60 (2004).
[54] Y. C. Zhou, S. Zhao, M. Feig, and G. W. Wei, J. Comput. Phys. **213**, 1 (2006).
[55] Y. C. Zhou and G. W. Wei, J. Comput. Phys. **219**, 228 (2006).
[56] S. Hou and X.-D. Liu, J. Comput. Phys. **202**, 411 (2005).
[57] S. N. Yu, Y. C. Zhou, and G. W. Wei, J. Comput. Phys. **224**, 729 (2007).
[58] Y. C. Zhou, M. Feig, and G. W. Wei, J. Comput. Chem. (to be published).
[59] J. G. Kirkwood, J. Chem. Phys. **7**, 351 (1934).
[60] S. N. Yu and G. W. Wei, J. Comput. Phys. (to be published).
[61] W. H. Geng, S. N. Yu, and G. W. Wei, J. Comput. Chem. (to be published).
[62] A. D. MacKerell, Jr., D. Bashford, M. Bellott *et al.*, J. Phys. Chem. **102**, 3586 (1998).