Journal of Computational Physics 426 (2021) 109958

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Regularization methods for the Poisson-Boltzmann equation: Comparison and accuracy recovery

Arum Lee [a], Weihua Geng [b], Shan Zhao [a,*]

[a] Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA
[b] Department of Mathematics, Southern Methodist University, Dallas, TX 75275, USA

ARTICLE INFO

Article history:
Received 31 March 2020
Received in revised form 23 October 2020
Accepted 24 October 2020
Available online 28 October 2020

Keywords:
Poisson-Boltzmann equation
Regularization method
Charge singularity
Matched Interface and Boundary (MIB) method
Elliptic interface problem

ABSTRACT

A significant challenge in numerical solution to the Poisson-Boltzmann equation is due to singular charge sources in terms of Dirac delta functions. To overcome this difficulty, several regularization methods have been developed, in which the potential function is decomposed into two or three parts so that the singular component can be analytically solved using the Green's function, while other components become bounded. However, it was observed in the literature that some regularization methods are significantly less accurate than the others for unclear reasons, even though they are analytically equivalent. To understand this discrepancy, the numerical performance of four popular regularization methods is investigated in this work by implementing them with the Matched Interface and Boundary (MIB) approach, which is a sophisticated finite difference method for treating elliptic interface problems with discontinuous coefficients. With all four methods showing second order convergence, accuracy reduction is numerically observed in two schemes. This paper provides numerical analysis and experiment to trace the source of such reduction, and links the error to the fact that the Laplacian of Green's function is dropped outside the protein domain. While this term is analytically vanishing, its numerical negligence introduces a discretization error. Formulating via a proper elliptic interface problem, an effective accuracy recovery technique is proposed so that all four methods yield the same high precision. With this study, all involved regularization schemes are better understood and well connected into a unified framework.

1. Introduction

The Poisson-Boltzmann equation (PBE) is an elliptic equation with singular source terms and discontinuous coefficients across an interface. This equation is widely used as the governing equation of electrostatic of solvated biomolecules in a solvent environment with dissolved electrolytes [2,14,29,21]. In practice, the PBE is solved numerically as its analytical solution is only available for simple shapes such as a sphere. However, there are several numerical difficulties in solving the PBE, such as complex molecular surface, discontinuous coefficients across the surface, singular sources, unbounded domain, etc. In this study, we focus on methods treating the singular sources, which come from the presence of the Dirac delta distributions in modeling of the partial charges located at the atomic centers of the biomolecules.

* Corresponding author.
E-mail address: szhao@ua.edu (S. Zhao).

https://doi.org/10.1016/j.jcp.2020.109958
0021-9991/

Conventionally, the singular charges are distributed to the surrounding grid points [19,25,45,40] or approximated by the hat functions [35]. Recently, a second order accurate geometric discretization of the point charge sources has been introduced in [12] for solving Poisson's equation. A central difference discretization of the Dirac delta function representing a singular source across an interface has been developed in [36] for treating boundary conditions for Poisson problems on irregular domains. Alternatively an effective approach is to remove the delta functions from the PBE by using the Green's function based decomposition. In this approach the solution of the PBE is decomposed into several parts with one part as the Green's function, which is the fundamental solution to the Poisson equation with Dirac delta distributions as the source term. The remaining terms in the decomposition thus solve the regularized PBE without the singular sources. There are several regularization methods of this kind being proposed such as [6,8,17,23,9,38,47]. These methods differ from one another mainly in the way the potential solution is decomposed. In another regularization approach [4,27], the singular sources or delta functions are directly decomposed into local and global components, and are accurately approximated by range-separated tensor discretization. This is coupled with a splitting of the electrostatic potential, and the resulting long range component satisfies the PBE with a regularized source term. The range-separated regularization [4,27] can maintain the same interface and boundary conditions as the original PBE model at the expense of delta function tensor approximations, while the Green's function based regularizations [6,8,17,23,9,38,47] usually lead to different interface and boundary conditions. Recently, regularization in a setting of smoothly varying dielectric functions over diffuse interfaces and complex shapes has been studied in [37].

In this paper, we compare, analyze, and simulate four popular Green's function based regularization methods using finite difference and rigorous interface treatment. These methods, whose numerical algorithms are presented in the next section, are briefly reviewed below in the chronological order.

The first method was proposed and explored in [19,47]. This method decomposes the PBE solution into two components on the entire computational domain: the singular component which is the Green's function and the remaining part called regular component. This scheme is a natural way to decompose the potential since each component has a physical meaning [23]. By analyzing the analytical solution of the PBE for a spherical cavity with a unit charge at its center, Holst et al. [23] found that this scheme might suffer from numerical instability. More specifically, a small error in the numerical solution of the regular component can lead to a large relative error in the entire solution. Despite this accuracy reduction, this scheme was one of the pioneering works in the field and many other sub-sequential works are inspired by it.

The second method was proposed by Chern et al. [9]. In this method, the solution to the PBE is split into the Green's function and two other regular components. One of the regular components handles the effect of the Green's function on the molecular surface (the domain interface) and the other term is the remaining correction term, which solves a regularized PBE with a modified interface jump conditions. In the PBE model, the computational domain Ω is divided into the solvent region Ω^+ and the molecular region Ω^-. This solution decomposition takes place only in the molecular region Ω^-, instead of on the entire domain Ω. This decomposition resolves the numerical instability that Holst discussed in [23]. This method was implemented with finite difference method (FDM) for the two-dimensional PBE in [9] and later implemented with another FDM named Matched Interface and Boundary method (MIB) in three-dimensions by Geng et al. [17].

The third method was proposed by Xie [38]. In this scheme, the solution is split into three parts on the entire domain Ω, with the first component being the Green's function. The second component carries the nonhomogeneous data of interface and boundary conditions, leaving the third component carrying the nonlinearity of the PBE. This solution decomposition works for a solvent containing any number of ionic species. Thus, this scheme not only works for the symmetric 1:1 ionic model as in Eq. (1), but also for a more general setting. Furthermore, the components of the PBE solution are defined in both the weak sense and strong sense, therefore the scheme can be implemented with both FDM and Finite Element Method (FEM). Owing to FEM's convenience in treating the interfaces, the FEM implementation of this scheme is more accurate than a standard FDM. In this paper, we implemented this scheme with the MIB method [46], which is discretized using finite difference but has a rigorous treatment of interface conditions.

The fourth method was proposed by Geng and Zhao [18] using the MIB discretization. Inspired by [6], Geng and Zhao split the PBE solution into two components. Similar to the method of Chern el al. [9], this decomposition takes place only in the molecular region Ω^-. In comparison with the three-component method in [17], this method produces the same accuracy, but is easier to implement since it could avoid solving a boundary value Poisson equation inside the molecular region Ω^- and some computation related to the interface jump conditions.

The first goal of this work is to compare the above mentioned four schemes under the framework of the Matched Interface and Boundary (MIB) method [46]. Being a second order accurate PBE solver, the MIB method is an effective approach for this comparison, because in terms of the discretized linear algebraic system $Ax = b$, different regularization schemes are only different on the source term b, that is associated with the boundary and interface conditions.

Our numerical experiments show that two regularization schemes [17,18] are significantly more accurate than the other two schemes [47,38] when implemented with the MIB, while all four schemes achieve second order convergence. This motivates us to investigate the errors numerically and theoretically and track down to the source of this accuracy reduction for the two schemes with lower accuracy. By appropriately incorporating the error source into the MIB interface algorithm, we are able to propose novel recovery techniques for these two schemes so that their accuracy is completely recovered.

The rest of the paper is organized as follows. Section 2 provides some background information about the Poisson-Boltzmann model and a review of existing numerical solvers. Section 3 summarizes the MIB method and four regularization schemes, followed by numerical comparison of a benchmark test using these four schemes under the framework of MIB

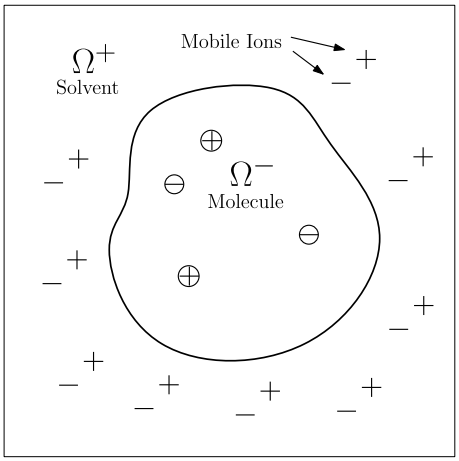

Fig. 1. A 2-dimensional illustration of the 3-dimensional domain. The molecular surface Γ divides the molecule domain Ω^+ and the solvent domain Ω^-.

method. Section 4 discusses the source of accuracy reduction and proposes a recovery technique, followed by its numerical validation. Section 5 ends the paper with concluding remarks.

2. Background

2.1. The Poisson-Boltzmann model

The Poisson-Boltzmann (PB) model is illustrated in Fig. 1, where Ω^- and Ω^+ are the molecule region and solvent region, separated by the molecular surface $\Gamma = \Omega^- \cap \Omega^+$. Note Ω^+ should be an infinite domain in the physical model, but a bounded computational domain such as the box $\Omega = \Omega^- \cup \Omega^+$ in the figure has to be applied for grid-based methods.

This model governs the electrostatic potential $\phi(r)$ on Ω by Gauss's law as

$$\begin{cases} -\epsilon^- \Delta\phi(r) = \rho(r), & r \in \Omega^- \\ -\epsilon^+ \Delta\phi(r) + \bar{\kappa}^2 \sinh\phi(r) = 0, & r \in \Omega^+ \\ [\phi]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon \frac{\partial\phi}{\partial n}\right]_\Gamma = 0, & r \in \Gamma \\ \phi(r) = \phi_b(r), & r \in \partial\Omega, \end{cases} \tag{1}$$

where

$$\rho(r) = 4\pi \sum_{j=1}^{N_c} q_i \delta(r - r_i), \tag{2}$$

is the source term representing the distribution of the N_c partial charges carried by the molecule. This term is given as the superposition of the Dirac delta function centered at the atomic centers r_i with charge quantity q_i.

In Eq. (1), the dielectric coefficient $\epsilon(r)$ is a piecewise constant function

$$\epsilon(r) = \begin{cases} \epsilon^+, & r \in \Omega^+ \\ \epsilon^-, & r \in \Omega^-, \end{cases} \tag{3}$$

with ϵ^+ and ϵ^- as the dielectric constants in the solvent region and molecule region. In our computational configuration, we choose $\epsilon^- = 1$ for molecule and $\epsilon^+ = 80$ for the solvent. The ionic screening coefficient $\bar{\kappa}^2(r)$ is also a piecewise constant function.

$$\bar{\kappa}^2(r) = \begin{cases} \epsilon^+ \kappa^2 & r \in \Omega^+ \\ 0 & r \in \Omega^-, \end{cases} \tag{4}$$

where κ is the inverse Debye screening length measuring the ionic strength I in mol/L as related by $\kappa^2 = 8.430325455 I/\epsilon^+$. In our computation, the partial charge uses the unit e_c, the fundamental charge and the length unit is Å. With these units, the unit of ϕ is e_c/Å. As in [18], to convert the unit of ϕ to the commonly used unit of kcal/mol/e_c, one needs to multiply ϕ

by the constant 332.0716 for room temperature $T = 298K$. Readers can find more details about the units of these physical quantities in [21,15].

Additionally, $[\phi]_\Gamma = \phi^+ - \phi^-$ and $[\epsilon\phi_n]_\Gamma = \epsilon^+ \frac{\partial \phi^+}{\partial n} - \epsilon^- \frac{\partial \phi^-}{\partial n}$ are the interface conditions across the molecular surface with n as the normal direction. For the infinite domain with the PB model, we have the radiation condition $\lim_{|r|\to\infty} \phi(r) = 0$. However, for grid-based method, we will have to apply a Dirichlet boundary condition [21] imposed on $\partial\Omega$ as

$$\phi(r) = \phi_b(r) = \sum_{i=1}^{N_c} \frac{q_i e^{-\kappa|r-r_i|}}{\epsilon^+|r-r_i|}. \tag{5}$$

2.2. Numerical solutions

Numerical methods for the Poisson-Boltzmann equation (PBE) fall into two classes, (1) grid-based methods that discretize the entire domain, e.g. [24,10,25,3,31,6,7,39,11,32,33,13], and (2) boundary integral methods that discretize the molecular surface, e.g. [16,26,28,5,30,20,1,43,44,34]. These methods have various advantages and disadvantages in solving the PBE. Some of them focus on a particular aspect in solving the PBE or its biological applications.

The numerical solution to the PBE suffers many challenges such as 1) the molecular surface is geometrically complex; 2) the biomolecule is represented by singular point charges; 3) the dielectric function and electric field are discontinuous across the molecular surface; 4) the nonlinearity appears when ionic strength is strong; 5) the domain is unbounded.

In treating these numerical challenges, particularly 1)–3), a second order accurate finite-difference based interface method named matched interface and boundary (MIB) method has been developed [46,40,17,7]. In this work, the MIB scheme will be adopted as the PBE solver for comparing four regularization schemes. We note that there are other accurate finite difference interface algorithms available for dealing with the above discussed challenges, such as the adaptive finite difference method [32,33,13]. In principal, the present numerical comparison can be conducted by using a different finite difference algorithm, but the conclusion should remain the same.

3. Numerical algorithms

In this section, we first introduce the MIB scheme, which lays the basic framework for solving the elliptic interface problems. The four regularization methods are introduced next. These methods are then implemented under the MIB scheme so that interface conditions and geometric singularities are uniformly addressed for a better comparison of these methods numerically.

3.1. Matched Interface and Boundary method (MIB)

We first introduce the second order MIB scheme using a 1-d example with the equation, interface condition, and boundary condition given as in Eq. (6).

$$\begin{cases} -\epsilon^- \dfrac{d^2\phi}{dx^2}(x) = g_1(x), & x \in (0, \alpha) \\ -\epsilon^+ \dfrac{d^2\phi}{dx^2}(x) = g_2(x), & x \in (\alpha, 1) \\ [\phi]_\Gamma = \beta \\ \left[\epsilon \frac{d\phi}{dx}\right]_\Gamma = \gamma \\ \phi(x) = \phi_b(x), & x \in \partial\Omega. \end{cases} \tag{6}$$

Here, the domain $\Omega = [0, 1]$, the interface $\Gamma = \{\alpha\}$ divides the domain into $[0, \alpha)$ and $(\alpha, 1]$. The dielectric coefficient $\epsilon(x)$ is a piecewise constant function given as

$$\epsilon(x) = \begin{cases} \epsilon^+ \text{ if } x \in [0, \alpha), \\ \epsilon^- \text{ if } x \in (\alpha, 1]. \end{cases} \tag{7}$$

Suppose we approximate the second derivative by the central difference

$$\frac{d^2\phi}{dx^2}(x) \approx \frac{\phi(x-\Delta x) - 2\phi(x) + \phi(x+\Delta x)}{\Delta x^2}. \tag{8}$$

If all points on the finite difference stencil, i.e. $x - \Delta x, x$, and $x + \Delta x$, lie on the same side of the interface $x = \alpha$, the central finite difference in Eq. (8) can be applied normally. Otherwise, one should not directly apply function values on the

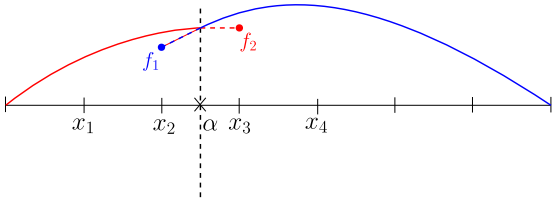

Fig. 2. A one-dimensional example.

opposite side due to the jumps in ϕ or $\frac{d\phi}{dx}$ at $x=\alpha$. For example, in Fig. 2, one should not use the value $\phi(x+\Delta x)=\phi(x_3)$ to approximate $\frac{d^2\phi}{dx^2}(x_2)$ as in Eq. (8). Instead, a fictitious value f_2 can be used to replace $\phi(x_3)$ in Eq. (8), i.e.

$$\frac{d^2\phi}{dx^2}(x_2) \approx \frac{\phi(x_1)-2\phi(x_2)+f_2}{\Delta x^2}. \tag{9}$$

Similarly, a fictitious value f_1 can be used to replace $\phi(x_2)$ in the discretization of $\frac{d^2\phi}{dx^2}(x_3)$. The fictitious values f_1 and f_2 are the smooth extensions of ϕ from one side of $x=\alpha$ to the other side in the way such that the jump conditions are satisfied. To calculate the coupled fictitious values f_1 and f_2, a polynomial interpolation is applied to discretize the jump conditions

$$\begin{cases} [\phi]_\Gamma = \phi^+(\alpha)-\phi^-(\alpha)=\beta \\ \left[\epsilon\frac{d\phi}{dx}\right]_\Gamma = \epsilon^+\frac{d\phi^+}{dx}(\alpha)-\epsilon^-\frac{d\phi^-}{dx}(\alpha)=\gamma, \end{cases} \tag{10}$$

in terms of $\{\phi(x_j)\}_{j=1,4}$ and the fictitious values f_1, f_2. For example, $\phi^+(\alpha)$ can be approximated by

$$\phi^+(\alpha) \approx w_1\phi(x_1)+w_2\phi(x_2)+w_3 f_2, \tag{11}$$

where $\{w_i\}_{i=1,3}$ are the weights obtained from polynomial interpolation. One can find similar approximations for $\phi^-(\alpha), \epsilon^+\frac{d\phi^+}{dx}(\alpha), \epsilon^-\frac{d\phi^-}{dx}(\alpha)$. It then follows from Eq. (10) such that

$$\begin{cases} (v_1 f_1+v_2\phi(x_3)+v_3\phi(x_4))-(w_1\phi(x_1)+w_2\phi(x_2)+w_3 f_2) & =[\phi]_\Gamma \\ \epsilon^+(\bar v_1 f_1+\bar v_2\phi(x_2)+\bar v_3\phi(x_3))-\epsilon^-(\bar w_1\phi(x_1)+\bar w_2\phi(x_2)+\bar w_3 f_2) & =\left[\epsilon\frac{d\phi}{dx}\right]_\Gamma. \end{cases} \tag{12}$$

Notice that the weights $\{w_i, v_i, \bar w_i, \bar v_i\}_{i=1,3}$ depend only on the interface position and mesh size. From the linear system (12), one can solve for fictitious values f_1 and f_2 as a linear combination of the solution $\{\phi(x_i)\}_{i=1,4}$, and the jump conditions $[\phi]_\Gamma$, and $\left[\epsilon\frac{d\phi}{dx}\right]_\Gamma$:

$$f_i = \sum_{j=1}^{4} c_j^i\phi(x_j)+c_5^i[\phi]_\Gamma+c_6^i\left[\epsilon\frac{d\phi}{dx}\right]_\Gamma,\ i=1,2. \tag{13}$$

Here $\left\{c_j^i\right\}$ are the weights that depend on the interface position and the mesh size. Replacing $\phi(x_3)$ by f_2 in (9), we have

$$\frac{d^2\phi}{dx^2}(x_2) \approx \frac{\phi(x_1)-2\phi(x_2)+\sum_{j=1}^{4} c_j^2\phi(x_j)+c_5^2[\phi]_\Gamma+c_6^2\left[\epsilon\frac{d\phi}{dx}\right]_\Gamma}{\Delta x^2}, \tag{14}$$

or in a simplified form using the modified weights $\bar c_j, j=1,\cdots,6$

$$\frac{d^2\phi}{dx^2}(x_2) \approx \sum_{j=1}^{4}\bar c_j\phi(x_j)+\bar c_5[\phi]_\Gamma+\bar c_6\left[\epsilon\frac{d\phi}{dx}\right]_\Gamma. \tag{15}$$

Similarly, f_1 is required to approximate $\frac{d^2\phi}{dx^2}(x_3)$. By using the discretization as in Eq. (15), the discretized Eq. (6) at x_2 has the form

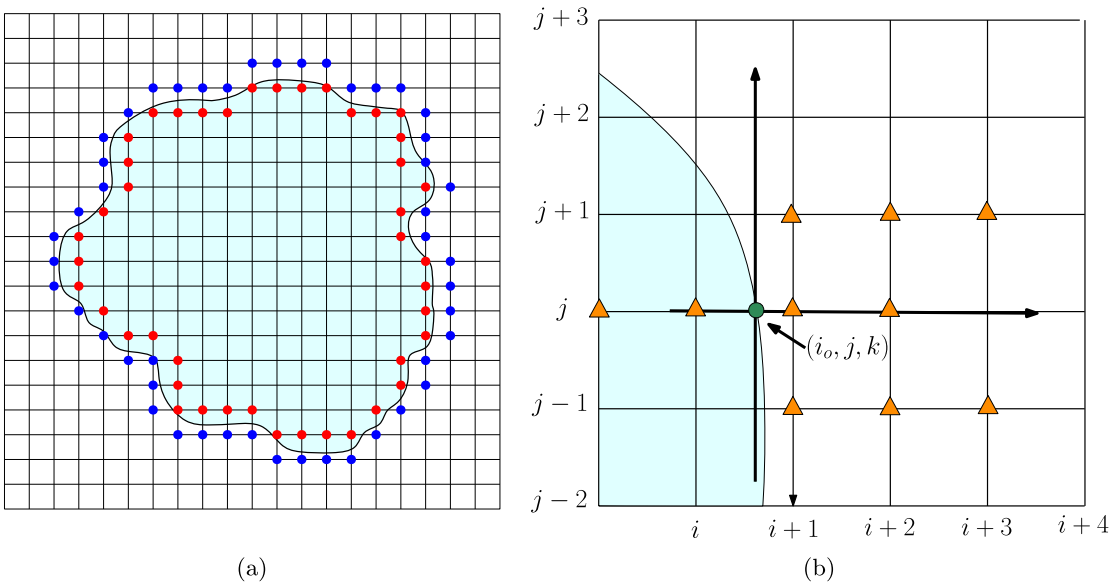

Fig. 3. A 2-d illustration of 3-d MIB scheme: (a) In the $z = z_k$ cross section, the irregular points (the blue and red grid points near the interface) have both function values and fictitious values; (b) Fictitious values at node (i, j, k) and $(i+1, j, k)$ are the linear combinations of the jump conditions at (i_o, j, k) and the values of ϕ at the neighboring nodes in x–y plane (shown as orange grids) and in x–z or y–z plane (not shown). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\sum_{j=1}^{4} \bar{c}_j \phi(x_j) \approx g_1(x_2) - \bar{c}_5[\phi]_\Gamma - \bar{c}_6 \left[\epsilon \frac{d\phi}{dx}\right]_\Gamma. \tag{16}$$

Combining this special discretization for irregular points involving jump conditions with the standard finite difference discretization for regular points, we form the linear algebraic system for $\phi(x_j)$, $j = 1, \cdots, n$ with n mesh points.

For PDEs in 2-d and 3-d, the MIB scheme utilizes the similar ideas but requires much more complicated treatment in jump conditions [45,46], geometric singularities [40–42], and charge singularities [17,18] and we only summarize the key ingredients of the algorithms using Figs. 3a-3b, which are 2-d illustration of the 3-d MIB scheme.

In 2-d and 3-d MIB schemes, the jump conditions are imposed as the normal derivatives and tangential derivatives transferred to partial derivatives in directions of the Cartesian coordinates. Each pair of irregular points as shown in Fig. 3a have both solution values and fictitious values. The fictitious values are linear combination of solution values and jump conditions expressed as

$$f_{i,j,k} = \sum_{(x_I, y_J, z_K) \in \mathbb{S}_{i,j,k}} W_{I,J,K} \phi\left(x_I, y_J, z_K\right) + W_0[\phi]_\Gamma + W_1 \left[\epsilon \frac{\partial \phi}{\partial n}\right]_\Gamma, \tag{17}$$

where $\mathbb{S}_{i,j,k}$ represents the set of neighboring points of $\left(x_i, y_j, z_k\right)$ with index (i, j, k) involved in the polynomial interpolation and $W_{I,J,K}$, W_0, W_1 are the weights from interpolation and transformation. The size of $\mathbb{S}_{i,j,k}$ is normally 10 for 2-d as seen in Fig. 3b and 16 for 3-d for the interpolation in multiple dimension [45,46]. However, when geometric singularities are present, larger number of neighboring points will be involved [40–42].

3.2. Singular source regularization

For treating the charge singularities involved in the PBE, Green's function based regularization methods decompose the potential solution into several components, with one of them being Green's function,

$$\mathcal{G}(r) = \sum_{i=1}^{N_c} \frac{q_i}{\epsilon^- |r - r_i|},$$

which solves the free space Poisson's equation

$$\begin{cases} -\epsilon^- \Delta \phi_c = \rho(r) \\ \lim\limits_{|r| \to \infty} \phi_c(r) = 0. \end{cases} \tag{18}$$

The remaining terms in ϕ thus solve a regularized PBE without the singular delta distributions. We next introduce several schemes from the literature which use this essential decomposition idea. An interesting phenomenon we have found is that although these schemes are analytically equivalent, their numerical results show different accuracy. We will demonstrate the differences, identify the reason behind it, and design numerical schemes to recover the accuracy. We start with a brief summary of the four above-mentioned schemes.

Scheme 1 [19,47]: In this method, the PBE solution is decomposed into the singular and regular component throughout the computational domain, i.e.

$$\phi = \mathcal{G} + \phi_{RF}, \text{ on } \Omega, \tag{19}$$

where ϕ_{RF} is the reaction field component.

The substitution of this ϕ into Eq. (1) leads to

$$\begin{cases} -\epsilon^- \Delta \phi_{RF}(r) - \epsilon^- \Delta \mathcal{G}(r) = \rho(r), & r \in \Omega^- \\ -\epsilon^+ \Delta \phi_{RF}(r) - \epsilon^+ \Delta \mathcal{G}(r) + \bar{\kappa}^2 \sinh(\mathcal{G} + \phi_{RF}) = 0, & r \in \Omega^+ \\ [\phi_{RF}]_\Gamma = -[\mathcal{G}]_\Gamma, & r \in \Gamma \\ \left[\epsilon \frac{\partial \phi_{RF}}{\partial n}\right]_\Gamma = -\left[\epsilon \frac{\partial \mathcal{G}}{\partial n}\right]_\Gamma, & r \in \Gamma \\ \phi_{RF}(r) + \mathcal{G}(r) = \phi_b(r), & r \in \partial\Omega. \end{cases} \tag{20}$$

Since $\epsilon^- \Delta \mathcal{G} = \rho(r)$ in Ω^-, $\Delta \mathcal{G} = 0$ in Ω^+, and $\mathcal{G}$ and $\frac{\partial \mathcal{G}}{\partial n}$ are continuous across the interface Γ, Eq. (20) with this decomposition is reduced to a regularized problem for reaction potential ϕ_{RF}

$$\begin{cases} -\epsilon^- \Delta \phi_{RF} = 0, & r \in \Omega^- \\ -\epsilon^+ \Delta \phi_{RF} + \bar{\kappa}^2 \sinh(\mathcal{G} + \phi_{RF}) = 0, & r \in \Omega^+ \\ [\phi_{RF}]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon \frac{\partial \phi_{RF}}{\partial n}\right]_\Gamma = \left(\epsilon^- - \epsilon^+\right) \frac{\partial \mathcal{G}}{\partial n}, & r \in \Gamma \\ \phi_{RF} = \phi_b - \mathcal{G}, & r \in \partial\Omega. \end{cases} \tag{21}$$

Scheme 2 [6,18]: The second scheme is similar to the first scheme with the difference that the PBE solution is decomposed in the solute region only, i.e.

$$\phi = \phi_{RF} + \mathcal{G}, \text{ in } \Omega^-.$$

The substitution of this ϕ into Eq. (1) leads to

$$\begin{cases} -\epsilon^- \Delta \phi_{RF}(r) - \epsilon^- \Delta \mathcal{G}(r) = \rho(r), & r \in \Omega^- \\ -\epsilon^+ \Delta \phi(r) + \bar{\kappa}^2 \sinh(\phi(r)) = 0, & r \in \Omega^+ \\ \phi(r^+) - \phi_{RF}(r^-) - \mathcal{G}(r^-) = 0, & r \in \Gamma \\ \epsilon^+ \frac{\partial \phi}{\partial n}(r^+) - \epsilon^- \frac{\partial \phi_{RF}}{\partial n}(r^-) - \epsilon^- \frac{\partial \mathcal{G}}{\partial n}(r^-) = 0, & r \in \Gamma \\ \phi(r) = \phi_b(r), & r \in \partial\Omega. \end{cases} \tag{22}$$

Using the fact that $\epsilon^- \Delta \mathcal{G} = \rho(r)$, we have

$$\begin{cases} -\epsilon^- \Delta \tilde{\phi} = 0, & r \in \Omega^- \\ -\epsilon^+ \Delta \tilde{\phi} + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0, & r \in \Omega^+ \\ [\tilde{\phi}]_\Gamma = \phi(r^+) - \phi_{RF}(r^-) = \mathcal{G}, & r \in \Gamma \\ \left[\epsilon \frac{\partial \tilde{\phi}}{\partial n}\right]_\Gamma = \epsilon^+ \frac{\partial \phi}{\partial n}(r^+) - \epsilon^- \frac{\partial \phi_{RF}}{\partial n}(r^-) = \epsilon^- \frac{\partial \mathcal{G}}{\partial n}, & r \in \Gamma \\ \tilde{\phi} = \phi_b, & r \in \partial\Omega, \end{cases} \tag{23}$$

which is a regularized equation with unknown $\tilde{\phi}$ defined as

$$\tilde{\phi} = \begin{cases} \phi_{RF} & \text{in } \Omega^- \\ \phi & \text{in } \Omega^+. \end{cases} \tag{24}$$

Note: $\tilde{\phi}$ in Scheme 2 is discontinuous across the interface Γ while ϕ_{RF} in Scheme 1 is continuous. For the MIB method, the discontinuity of the solution does not introduce additional numerical difficulties. For the finite element method, a discontinuous Galerkin FEM [11] shall be used.

Scheme 3 [38]: In this method, the PBE solution is split into three components on the whole domain i.e.

$$\phi(r) = \mathcal{G}(r) + \Psi(r) + \bar{\Phi}(r),$$

for $r \in \Omega$. In this decomposition, Ψ satisfies

$$\begin{cases} \Delta\Psi(r) = 0, & r \in \Omega^+ \cup \Omega^- \\ \Psi(r^+) = \Psi(r^-), & r \in \Gamma \\ \epsilon^+ \frac{\partial \Psi(r^+)}{\partial n} = \epsilon^- \frac{\partial \Psi(r^-)}{\partial n} + (\epsilon^- - \epsilon^+) \frac{\partial \mathcal{G}(r^+)}{\partial n} & r \in \Gamma \\ \Psi(r) = \phi_b(r) - \mathcal{G}(r), & r \in \partial\Omega, \end{cases} \tag{25}$$

and $\bar{\Phi}$ is the remaining correction term.

The substitution of this ϕ into Eq. (1) leads to

$$\begin{cases} -\epsilon^- \left(\Delta\bar{\Phi}(r) + \Delta\Psi(r) + \Delta\mathcal{G}(r)\right) = \rho(r), & r \in \Omega^- \\ -\epsilon^+ \left(\Delta\bar{\Phi}(r) + \Delta\Psi(r) + \Delta\mathcal{G}(r)\right) + \bar{\kappa}^2 \sinh\left(\bar{\Phi}(r) + \Psi(r) + \mathcal{G}(r)\right) = 0, & r \in \Omega^+ \\ [\bar{\Phi}]_\Gamma + [\Psi]_\Gamma + [\mathcal{G}]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon \frac{\partial \bar{\Phi}}{\partial n}\right]_\Gamma + \left[\epsilon \frac{\partial \Psi}{\partial n}\right]_\Gamma + \left[\epsilon \frac{\partial \mathcal{G}}{\partial n}\right]_\Gamma = 0, & r \in \Gamma \\ \bar{\Phi}(r) + \Psi(r) + \mathcal{G}(r) = \phi_b(r), & r \in \partial\Omega. \end{cases} \tag{26}$$

By using the facts that $\epsilon^+\Delta\mathcal{G} = 0$ in Ω^+, $\epsilon^-\Delta\mathcal{G} = \rho(r)$ in Ω^- and $\Delta\Psi(r) = 0$ on $\Omega^+ \cup \Omega^-$, we have

$$\begin{cases} -\epsilon^-\Delta\bar{\Phi} = 0, & r \in \Omega^- \\ -\epsilon^+\Delta\bar{\Phi}(r) + \bar{\kappa}^2 \sinh\left(\bar{\Phi}(r) + \Psi(r) + \mathcal{G}(r)\right) = 0, & r \in \Omega^+ \\ \bar{\Phi}(r^+) = \bar{\Phi}(r^-), & r \in \Gamma \\ \epsilon^+ \frac{\partial \bar{\Phi}(r^+)}{\partial n} = \epsilon^- \frac{\partial \bar{\Phi}(r^-)}{\partial n}, & r \in \Gamma \\ \bar{\Phi}(r) = 0, & r \in \partial\Omega. \end{cases} \tag{27}$$

This scheme requires solving Eqs. (25) and (27) for Ψ and $\bar{\Phi}$ respectively.

Scheme 4 [9,17]: Instead of splitting the PBE solution on the whole domain, in this method, ϕ is only split in the solute region Ω^-, i.e.

$$\phi(r) = \hat{\phi}(r) + \mathcal{G}(r) + \phi^0(r), \quad \text{for } r \in \Omega^-. \tag{28}$$

Here, ϕ^0 is the harmonic component and is defined only on $\Omega^- \cup \Gamma$ and satisfies the equation

$$\begin{cases} \Delta\phi^0(r) = 0, & \text{in } \Omega^- \\ \phi^0(r) = -\mathcal{G}(r), & \text{on } \Gamma, \end{cases} \tag{29}$$

and $\hat{\phi}$ is thus the remaining correction term.

The substitution of this ϕ into Eq. (1) leads to

$$\begin{cases} -\epsilon^-\Delta\hat{\phi}(r) - \epsilon^-\Delta(\mathcal{G}(r) + \phi^0(r)) = \rho(r), & r \in \Omega^- \\ -\epsilon^+\Delta\hat{\phi}(r) + \bar{\kappa}^2 \sinh\hat{\phi}(r) = 0, & r \in \Omega^+ \\ [\hat{\phi}]_\Gamma + [\mathcal{G} + \phi^0]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon \frac{\partial \hat{\phi}}{\partial n}\right]_\Gamma + \left[\epsilon \frac{\partial (\mathcal{G} + \phi^0)}{\partial n}\right]_\Gamma = 0, & r \in \Gamma \\ \hat{\phi}(r) = \phi_b(r), & r \in \partial\Omega. \end{cases} \tag{30}$$

By using the facts that $\epsilon^-\Delta\mathcal{G} = \rho(r)$ and $\Delta\phi^0(r) = 0$ in Ω^-, $\mathcal{G}$ and ϕ^0 are only defined in $\Omega^- \cup \Gamma$, and $\mathcal{G} + \phi^0 = 0$ on Γ, we have

$$\begin{cases} -\epsilon^-\Delta\hat{\phi}(r) = 0, & r \in \Omega^- \\ -\epsilon^+\Delta\hat{\phi}(r) + \bar{\kappa}^2 \sinh\hat{\phi}(r) = 0, & r \in \Omega^+ \\ [\hat{\phi}]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon \frac{\partial \hat{\phi}}{\partial n}\right]_\Gamma = \epsilon^- \frac{\partial \mathcal{G}}{\partial n}(r^-) + \epsilon^- \frac{\partial \phi^0}{\partial n}(r^-), & r \in \Gamma \\ \hat{\phi}(r) = \phi_b(r), & r \in \partial\Omega. \end{cases} \tag{31}$$

In this method, Eqs. (29) and (31) need to be solved for ϕ^0 and $\hat{\phi}$ respectively.

Table 1
Solving the Poisson equation and the PBE on a spherical cavity of radius 2 with a unit charge e_c centered at (0,0,0), $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 0$ for the Poisson equation and $\kappa = 0.1257$ for the PBE.

h	The Poisson equation						The PBE					
	Scheme 1 & 3			Scheme 2 & 4			Scheme 1 & 3			Scheme 2 & 4		
	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}
1	6.6E-02		-88.08	8.4E-04		-81.94	5.3E-02		-86.21	6.7E-04		-81.96
1/2	1.2E-02	2.45	-83.35	1.5E-04	2.45	-81.96	9.1E-03	2.54	-82.86	1.2E-04	2.54	-81.97
1/4	3.3E-03	1.88	-82.21	4.2E-05	1.88	-81.98	2.5E-03	1.84	-82.12	3.2E-05	1.84	-81.98
1/8	6.7E-04	2.29	-82.02	8.5E-06	2.29	-81.98	5.3E-04	2.27	-82.01	6.7E-06	2.27	-81.98
1/16	1.2E-04	2.54	-81.99	1.5E-06	2.54	-81.98	9.4E-05	2.48	-81.98	1.5E-06	2.14	-81.98

3.3. Numerical comparison

In this section, we provide the numerical results of solving the PBE on some benchmark problems and on a set of 24 different proteins using the four regularization schemes discussed above. In addition to Scheme 2 and Scheme 4 implemented by Geng et al. [17,18] under the MIB framework, we newly implemented Scheme 1 and Scheme 3 with the MIB method for comparison purpose.

The benchmark problems are defined with Ω^- as a ball with radius $R = 2$Å and a positive unit charge at the center/origin. The Poisson equation is given as

$$-\nabla \cdot (\epsilon(r)\nabla\phi(r)) = 4\pi\delta(r), \tag{32}$$

and the PBE is given as

$$-\nabla \cdot (\epsilon(r)\nabla\phi) + \bar{\kappa}^2(r)\sinh(\phi(r)) = f(r), \tag{33}$$

where

$$f(r) = \begin{cases} 4\pi\delta(r), & \text{if } \|r\| < R \\ \bar{\kappa}^2 \sinh\left(\dfrac{1}{\epsilon^+\|r\|}\right) & \text{if } \|r\| > R. \end{cases}$$

Here the ionic strength $I = 0.15$ mol/L, $\kappa = 8.430325455I/\epsilon^+ = 0.1257$, and $\bar{\kappa}^2 = \epsilon^+\kappa^2$.

With these configurations, the analytical solution to both Eqs. (32) and (33) is

$$\phi(r) = \begin{cases} \dfrac{1}{\epsilon^+ R} - \dfrac{1}{\epsilon^- R} + \dfrac{1}{\epsilon^-\|r\|}, & \text{if } \|r\| < R \\ \dfrac{1}{\epsilon^+\|r\|}, & \text{if } \|r\| > R. \end{cases} \tag{34}$$

We use the analytical solution to calibrate the numerical solutions and impose the boundary conditions. For treating nonlinearity of the PBE, we applied inexact Newton's method as implemented in [18,7,22]. In all of our numerical tests, we set the relative tolerance for the biconjugate iterative solver as 10^{-6}. The electrostatic potential has the unit kcal/mol/e_c. The solvation free energy with the unit kcal/mol is evaluated as

$$E_{sol} = \sum_{i=1}^{N_c} q_i \phi_{RF}(r_i),$$

where the reaction potential ϕ_{RF} is defined in Eq. (19).

The errors are reported in the form

$$e_\phi = \max_{i=1}^{N_i} \left|\phi_{\text{true}}(x_i) - \phi_{\text{num}}(x_i)\right|,$$

where x_i are the irregular grid points near the interface in which at least one point in the seven-point finite difference stencil is on the opposite side of the interface. Here N_i is the total number of irregular grid points, and $\phi_{\text{true}}(x_i)$ and $\phi_{\text{num}}(x_i)$ are values of $\phi(x_i)$ obtained from the analytical solution and the numerical solution respectively.

Table 1 reports the solvation free energy E_{sol}, the electrostatic potential error e_ϕ and its convergence order with respect to the mesh refinement by solving the Poisson equation (32) and the PBE (33). For this spherical example, the numerical results of Scheme 1 and 3 are identical, and those of Scheme 2 and 4 are also the same. Thus, only half of original data is reported in Table 1 to save the space. From the results, we can see all four methods demonstrate the 2nd order convergence pattern. However, Schemes 1 and 3 present larger relative surface potential errors as compared with Schemes 2 and 4. We will investigate these numerical artifacts the next.

Table 2
Selected ratios between the errors e_ϕ from the four schemes reported in Table 1. An explanation of such error ratios will be given in Section 4.

h	The Poisson equation					The Poisson-Boltzmann equation				
	1	1/2	1/4	1/8	1/16	1	1/2	1/4	1/8	1/16
Sch.1/Sch.2	78.57	80.00	78.57	78.82	80.00	79.00	79.00	79.00	79.00	79.00
Sch.3/Sch.4	78.57	80.00	78.57	78.82	66.67	79.00	79.00	79.06	78.99	62.45

Table 3
Solving the nonlinear PBE (1) on 24 proteins using the four schemes under the MIB frame at different mesh size h. The differences between some energies listed in this table are shown in Fig. 4.

Protein	Schemes 1 & 3			Scheme 2			Scheme 4		
	$h=1$	$h=0.5$	$h=0.25$	$h=1$	$h=0.5$	$h=0.25$	$h=1$	$h=0.5$	$h=0.25$
1ajj	-1215.31	-1158.57	-1144.51	-1137.74	-1139.74	-1140.84	-1145.27	-1140.81	-1141.09
2erl	-1037.81	-968.68	-956.27	-966.07	-952.54	-953.24	-968.11	-952.98	-953.41
1cbn	-343.29	-312.67	-306.06	-307.77	-303.38	-304.05	-310.00	-303.94	-304.15
1vii	-1010.19	-923.74	-908.34	-910.99	-902.58	-903.88	-915.87	-903.38	-904.76
1fca	-1300.95	-1224.75	-1209.19	-1217.07	-1204.66	-1205.72	-1220.11	-1205.25	-1205.90
1bbl	-1089.69	-1013.57	-994.72	-994.44	-988.64	-989.35	-998.59	-988.81	-989.57
2pde	-869.97	-834.73	-825.39	-818.30	-821.21	-823.10	-826.87	-823.14	-823.42
1sh1	-835.22	-774.34	-759.13	-749.89	-754.20	-754.72	-757.60	-755.31	-755.01
1vjw	-1361.52	-1267.03	-1246.77	-1251.24	-1241.28	-1242.26	-1254.98	-1241.88	-1242.47
1uxc	-1291.49	-1170.23	-1147.38	-1154.48	-1139.50	-1141.66	-1159.26	-1140.79	-1141.99
1ptq	-1006.64	-897.39	-879.44	-894.04	-873.50	-874.88	-898.17	-874.55	-875.14
1bor	-940.52	-873.05	-858.85	-854.34	-853.65	-855.19	-859.66	-854.59	-855.41
1fxd	-3458.41	-3354.86	-3329.36	-3327.40	-3321.94	-3322.97	-3332.34	-3322.73	-3323.22
1r69	-1210.97	-1113.31	-1095.15	-1107.15	-1088.84	-1090.65	-1111.97	-1089.74	-1090.92
1mbg	-1491.37	-1379.41	-1360.05	-1370.45	-1353.60	-1354.95	-1373.12	-1354.31	-1355.18
1bpi	-1446.91	-1336.96	-1313.61	-1322.61	-1304.61	-1306.54	-1326.67	-1305.37	-1306.79
1hpt	-918.19	-837.44	-819.49	-825.13	-811.71	-814.52	-827.33	-812.64	-814.77
451c	-1141.25	-1055.74	-1034.04	-1024.48	-1026.06	-1027.86	-1031.33	-1027.59	-1028.20
1svr	-1932.94	-1752.30	-1721.70	-1734.47	-1711.50	-1713.67	-1743.57	-1712.93	-1714.10
1frd	-3081.05	-2905.01	-2874.81	-2887.75	-2863.26	-2867.03	-2892.96	-2864.99	-2867.50
1a2s	-2072.11	-1954.00	-1929.64	-1930.10	-1921.65	-1923.66	-1934.38	-1922.68	-1923.56
1neq	-1984.20	-1778.67	-1744.72	-1759.59	-1732.17	-1735.20	-1767.50	-1733.34	-1735.63
1a63	-2694.42	-2436.70	-2391.65	-2407.65	-2375.10	-2378.94	-2419.82	-2378.04	-2379.62
1a7m	-2403.14	-2214.41	-2174.58	-2182.78	-2160.92	-2164.17	-2197.34	-2163.26	-2164.75

Table 2 reports some selected ratios between the errors e_ϕ from the four schemes as reported in Table 1. Due to the similarities of the schemes, we compare Scheme 1 with Scheme 2 and Scheme 3 with Scheme 4. The ratios are approximately 78-80 except when h is too small (e.g. $h = 1/16$). Similar ratios are observed in [23] for the Poisson equation.

Table 3 reports the results in terms of electrostatic solvation free energy after solving the nonlinear PBE in Eq. (1) on a collection of 24 proteins using the same parameter configuration. It can be seen from this table that for all 24 proteins, the solvation energies in Scheme 2 and Scheme 4 converge faster than in Scheme 1 and Scheme 3 when the mesh size h is reduced. Note results from Schemes 1 and 3 are different only after 10^{-6} thus we listed them together for saving space.

To better present the information embedded in Table 3, we plot the data in a reorganized way in Fig. 4. Here we choose the solvation energies produced by Scheme 2 with mesh size $h = 0.25$ as the reference and calculate the difference ΔE_{sol} compared with different schemes at different mesh sizes. Since we find that Schemes 1 and 3 are almost numerically equivalent, we only present results from Scheme 1. Fig. 4(a) compares results from Scheme 2 at $h = 1.0$ and $h = 0.5$ and from Scheme 1 at $h = 1.0, 0.5, 0.25$ using results from Scheme 2 at $h = 0.25$ as the reference. Fig. 4(b) compare results from Scheme 2 at $h = 1.0$ and $h = 0.5$ (shown at different scale) and from Scheme 4 at $h = 1.0, 0.5, 0.25$ using results from Scheme 2 at $h = 0.25$ as the reference. These plots clearly showed when the meshes are refined all schemes converge and Scheme 4 converges much faster than Scheme 1.

The final linear algebraic systems $Ax = b$ to be solved for all four schemes have the identical matrix A but different vectors b. The differences in b, which are caused by the different source terms and interface conditions as seen from these four schemes in Eqs (21), (23), (27), (31), lead to different iteration numbers in solving the linear PBE and different numbers in which the linear solver is called by the inexact Newton's method. As an example, we show in Fig. 5, compared with Scheme 1, Scheme 2 calls the linear solver LINBCG fewer times and uses less CPU time.

4. Numerical analysis and accuracy recovery

In this section, we first point out the source of accuracy reduction from numerical analysis on Schemes 1 and 2. Based on this source, we provide numerical techniques to recover the accuracy, followed by an additional note about the numerical equivalence between Schemes 1 and 3. Finally we numerically validate the accuracy recovery techniques.

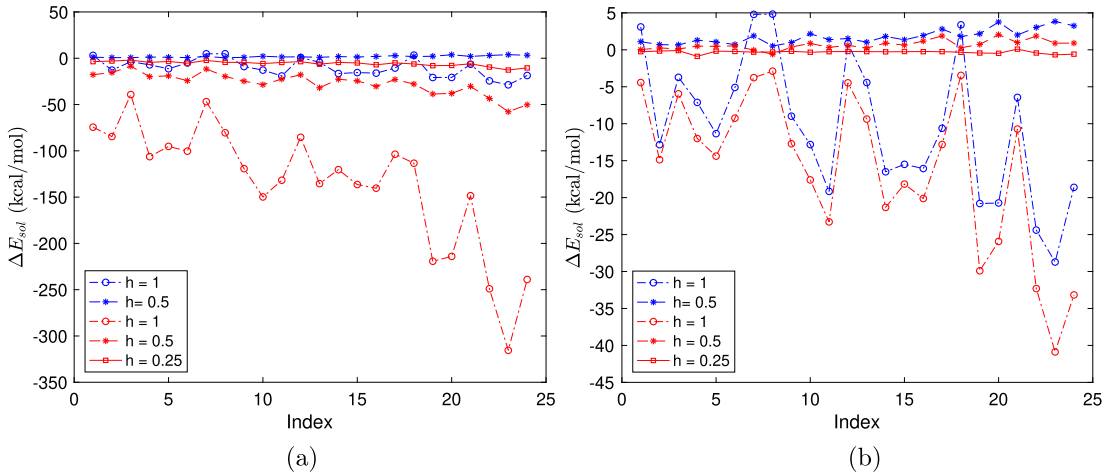

Fig. 4. Graphic presentation of selected data from Table 3: (a) ΔE_{sol} from Scheme 1 (red) at $h = 1, 0.5, 0.25$ and Scheme 2 (blue) at $h = 1, 0.5$ compared with results from Scheme 2 at $h = 0.25$; (b) ΔE_{sol} from Scheme 4 (red) at $h = 1, 0.5, 0.25$ and Scheme 2 (blue) at $h = 1, 0.5$ compared with results from Scheme 2 at $h = 0.25$.

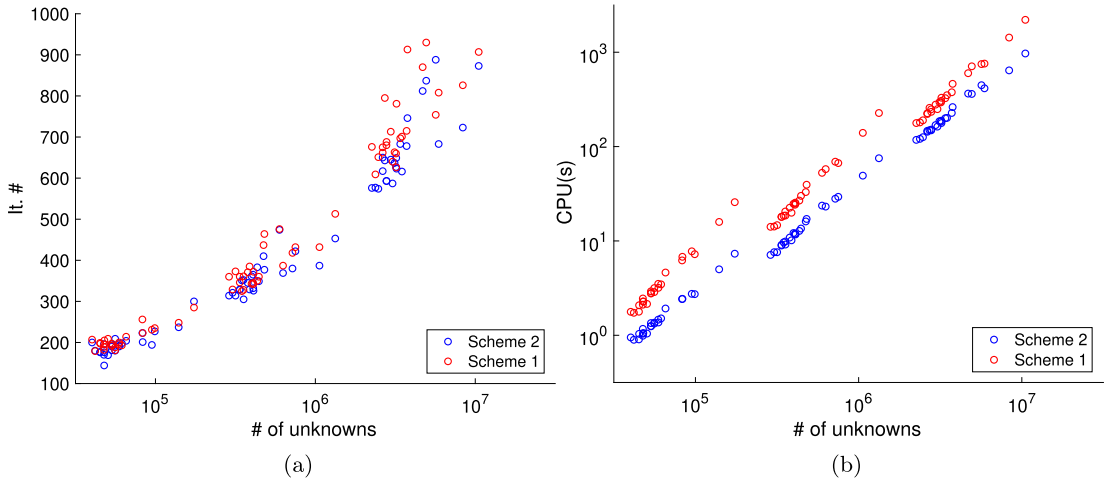

Fig. 5. (a) Number of iterations and (b) CPU time of Scheme 1 (red) and Scheme 2 (blue) for 24 proteins.

4.1. Source of accuracy reduction

The essential source of accuracy reduction is due to omitting $\epsilon^+\Delta\mathcal{G}$ in Ω^+. While this term is analytically vanishing, its numerical negligence introduces an error in estimating the reaction field component ϕ_{RF}. Unfortunately, a solution upscaling is associated with the decomposition in singular source regularization. We observed that the solution ϕ to the PBE is in a much smaller scale compared to ϕ_{RF} or the Green's function $\mathcal{G}$. Consequently, the error in ϕ_{RF} is greatly amplified into a big error in ϕ. Here we use Schemes 1 and 2 to illustrate the idea, which is further supported by rigorous proofs.

In discretizing ϕ_{RF} in Scheme 1, $\epsilon^+\Delta\mathcal{G}$ is omitted in

$$-\epsilon^+\Delta\phi_{RF} + \bar{\kappa}^2 \sinh(\mathcal{G} + \phi_{RF}) = 0, \tag{35}$$

while in Scheme 2, $\epsilon^+\Delta\mathcal{G}$ is actually contained in $\epsilon^+\Delta\tilde{\phi}$

$$-\epsilon^+\Delta\tilde{\phi} + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0. \tag{36}$$

Let x_0 be a regular grid point in Ω^+ and $Df(x_0)$ be the 7-point centered difference stencil for Δf at x_0. Let e_1 and e_2 denote the discretization errors in Scheme 1 and Scheme 2 at x_0 as

$$e_1 = \epsilon^+\Delta\phi_{RF}(x_0) - \epsilon^+ D\phi_{RF}(x_0), \tag{37}$$

$$e_2 = \epsilon^+\Delta\phi(x_0) - \epsilon^+ D\phi(x_0). \tag{38}$$

Using the fact that for Scheme 1

$$\phi = \phi_{RF} + \mathcal{G},$$

in Ω^+ and the linearity of Df and Δf, we have

$$e_2 = e_1 + \epsilon^+ \Delta\mathcal{G}(x_0) - \epsilon^+ D\mathcal{G}(x_0). \tag{39}$$

Thus, e_1 and e_2 differ by $\epsilon^+\Delta\mathcal{G}(x_0) - \epsilon^+ D\mathcal{G}(x_0)$, which is the discretization error of $\epsilon^+\Delta\mathcal{G}(x_0)$ using centered difference. This difference is in the same scale as e_1, but is much larger than e_2.

Quantitatively we use the Poisson equation (32) and its analytical solution (34) as an example to illustrate the idea. The singular component in this case is $\mathcal{G}(r) = \dfrac{1}{\epsilon^- \|r\|}$ and the regular component is

$$\phi_{RF}(r) = \phi(r) - \mathcal{G}(r) = \begin{cases} \left(\dfrac{1}{\epsilon^+} - \dfrac{1}{\epsilon^-}\right)\dfrac{1}{R} & \text{if } r \in \Omega^- \\ \dfrac{1}{\epsilon^+\|r\|} - \dfrac{1}{\epsilon^-\|r\|} & \text{if } r \in \Omega^+. \end{cases} \tag{40}$$

Using $\phi(r) = \dfrac{1}{\epsilon^+ ||r||}$ in the solvent region Ω^+, we have in Ω^+

$$\phi_{RF}(r) = \left(\frac{1}{\epsilon^+} - \frac{1}{\epsilon^-}\right)\epsilon^+\phi(r) = \left(\frac{1}{\epsilon^+} - \frac{1}{\epsilon^-}\right)\epsilon^-\mathcal{G}(r). \tag{41}$$

Let $\epsilon^- = 1$ in Ω^- and $\epsilon^+ = 80$ in Ω^+, then on Ω^+

$$\phi_{RF} = -79\phi = -\frac{79}{80}\mathcal{G}. \tag{42}$$

Furthermore, since in Ω^-, ϕ_{RF} and $\tilde{\phi}$ equal to the same constant, we have on $\Omega^+ \cup \Omega^-$,

$$D\phi_{RF} = -79D\tilde{\phi} = -\frac{79}{80}D\mathcal{G}. \tag{43}$$

From the linearity of the operators, the discretization error e_1, e_2 and $e_3 = \epsilon^+(\Delta\mathcal{G}(x_0) - D\mathcal{G}(x_0))$ have the same relationship as $e_1 = -79e_2 = \dfrac{-79}{80}e_3$. A similar error analysis can be done for grid points that lie in Ω^- and near the interface Γ. The ratios between the discretization errors of Eqs. (21) and (23) are all about 79 from Eq. (42) for the same reason. This explains why we observed that e_1 is about 79 times of e_2 in magnitude in Table 2.

4.2. Recovery techniques

Since removing $\mathcal{G}$ away from ϕ_{RF} is the source of accuracy reduction, we propose to modify equation (21) in Scheme 1 by keeping the term $-\epsilon^+\Delta\mathcal{G}$ in Ω^+:

$$\begin{cases} -\epsilon^-\Delta\phi_{RF} = 0 & r \in \Omega^+ \\ -\epsilon^+\Delta\phi_{RF} - \epsilon^+\Delta\mathcal{G} + \sinh(\phi_{RF} + \mathcal{G}) = 0 & r \in \Omega^- \\ [\phi_{RF}]_\Gamma = 0, & r \in \Gamma \\ \left[\epsilon\frac{\partial\phi_{RF}}{\partial n}\right]_\Gamma = \left(\epsilon^- - \epsilon^+\right)\frac{\partial\mathcal{G}}{\partial n}, & r \in \Gamma \\ \phi_{RF} = \phi_b - \mathcal{G} & r \in \partial\Omega, \end{cases} \tag{44}$$

and approximate the term $-\epsilon^+\Delta\mathcal{G}$ using the same finite difference discretization applied to $-\epsilon^+\Delta\phi_{RF}$ and $-\epsilon^+\Delta\tilde{\phi}$. In other words, we would like to add a finite difference approximation of

$$-\nabla\cdot(\epsilon(r)\nabla\tilde{\mathcal{G}}(r)) = \begin{cases} 0, & r \in \Omega^- \\ -\epsilon^+\Delta\mathcal{G}(r), & r \in \Omega^+, \end{cases} \tag{45}$$

where

$$\tilde{\mathcal{G}}(r) = \begin{cases} 0, & r \in \Omega^- \\ \mathcal{G}(r), & r \in \Omega^+, \end{cases} \tag{46}$$

solves

$$\begin{cases} -\epsilon^- \Delta \tilde{\mathcal{G}}(r) = 0, & r \in \Omega^- \\ -\epsilon^+ \Delta \tilde{\mathcal{G}}(r) = 0, & r \in \Omega^+ \\ [\tilde{\mathcal{G}}]_\Gamma = \mathcal{G}, & r \in \Gamma \\ \left[\epsilon \frac{\partial \tilde{\mathcal{G}}}{\partial n}\right]_\Gamma = \epsilon^+ \frac{\partial \mathcal{G}}{\partial n}, & r \in \Gamma \\ \tilde{\mathcal{G}} = \mathcal{G}, & r \in \partial\Omega. \end{cases} \tag{47}$$

Equation (47) is another interface problem with interface jump conditions, thus it must be solved using the MIB scheme for discretization in the same fashion as discretizing Eqs. (21) and (23).

In order to prove the recovery scheme, we use MIB(u) to denote the MIB discretization of the following operator with the interface conditions:

$$\begin{cases} -\epsilon^- \Delta u, & \text{if } r \in \Omega^- \\ -\epsilon^+ \Delta u, & \text{if } r \in \Omega^+ \\ [u]_\Gamma = \alpha, & r \in \Gamma \\ \left[\epsilon \frac{\partial u}{\partial n}\right]_\Gamma = \beta, & r \in \Gamma. \end{cases} \tag{48}$$

Consider a uniform mesh with spacing size h, and denote the evaluation of function u on this mesh as $u|_h$, which has been relabeled as a vector. We also use (A_u, S_u) to denote the pair of matrix-vector in the MIB discretization of u, i.e.,

$$\text{MIB}(u) = A_u u|_h + S_u.$$

Note the S_u here is not the discretization of the right hand side of the PDE but the linear combination of the interface conditions involved in MIB discretization. The MIB operator is linear thus for $u = av + bw$, we have

$$\text{MIB}(u) = a\text{MIB}(v) + b\text{MIB}(w).$$

Under this configuration, adding the correction term $\text{MIB}(\tilde{\mathcal{G}}) = A_{\tilde{\mathcal{G}}}\tilde{\mathcal{G}}|_h + S_{\tilde{\mathcal{G}}}$ to the discretization of PDE (21) to recover the accuracy is in fact solving

$$\text{MIB}(\phi_{RF}) + \text{MIB}(\tilde{\mathcal{G}}) + \bar{\kappa}^2(r)\sinh(\phi_{RF} + \mathcal{G}) = 0, \tag{49}$$

instead of solving

$$\text{MIB}(\phi_{RF}) + \bar{\kappa}^2 \sinh(\phi_{RF} + \mathcal{G}) = 0. \tag{50}$$

Below is a proposition for the accuracy recovery of Scheme 1, followed by its proof.

Proposition 1: Let the MIB discretization errors of Eqs. (21) and (23) be

$$E_1 = [-\nabla \cdot (\epsilon \nabla \phi_{RF})]_h + \bar{\kappa}^2 [\sinh(\phi_{RF} + \mathcal{G})]_h - \text{MIB}(\phi_{RF}) - \bar{\kappa}^2 \sinh\left([\phi_{RF}]_h + \tilde{\mathcal{G}}|_h\right), \tag{51}$$

$$E_2 = \left[-\nabla \cdot (\epsilon \nabla \tilde{\phi})\right]_h + \bar{\kappa}^2 \left[\sinh(\tilde{\phi})\right]_h - \text{MIB}(\tilde{\phi}) - \bar{\kappa}^2 \sinh(\tilde{\phi}|_h). \tag{52}$$

The difference between the MIB discretization errors in Scheme 1 and Scheme 2 is the MIB discretization of Eq. (47): $\text{MIB}(\tilde{\mathcal{G}}) = A_{\tilde{\mathcal{G}}}\tilde{\mathcal{G}}|_h + S_{\tilde{\mathcal{G}}}$, that is

$$E_1 - E_2 = \text{MIB}(\tilde{\mathcal{G}}).$$

Proof. Since the difference between $[\sinh(\tilde{\phi})]_h$ and $\sinh(\tilde{\phi}|_h)$ is on the order of the double precision limit, the MIB discretization errors are essentially determined by the approximations for elliptic operators

$$E_1 = [-\nabla \cdot (\epsilon \nabla \phi_{RF})]_h - \text{MIB}(\phi_{RF}), \tag{53}$$

$$E_2 = \left[-\nabla \cdot (\epsilon \nabla \tilde{\phi})\right]_h - \text{MIB}(\tilde{\phi}). \tag{54}$$

From Eq. (24), we know

$$\tilde{\phi} = \begin{cases} \phi_{RF} & \text{in } \Omega^- \\ \phi = \phi_{RF} + \mathcal{G} & \text{in } \Omega^+. \end{cases} \tag{55}$$

By Eq. (46), we thus have the identity

$$\tilde{\phi} = \phi_{RF} + \tilde{\mathcal{G}}. \tag{56}$$

By the linearity of elliptic operators and MIB operators, we have

$$\begin{aligned} E_2 &= \left[-\nabla\cdot(\epsilon\nabla\tilde{\phi})\right]_h - \mathrm{MIB}(\tilde{\phi}) \\ &= [-\nabla\cdot(\epsilon\nabla\phi_{RF})]_h + \left[-\nabla\cdot(\epsilon\nabla\tilde{\mathcal{G}})\right]_h - \mathrm{MIB}(\phi_{RF}) - \mathrm{MIB}(\tilde{\mathcal{G}}) \\ &= E_1 + \left[-\nabla\cdot(\epsilon\nabla\tilde{\mathcal{G}})\right]_h - \mathrm{MIB}(\tilde{\mathcal{G}}). \end{aligned}$$

Recall that $-\epsilon^-\Delta\tilde{\mathcal{G}}$ and $-\epsilon^+\Delta\tilde{\mathcal{G}}$ are vanishing in Ω^- and Ω^+, respectively. Thus, the grid values $\left[-\nabla\cdot(\epsilon\nabla\tilde{\mathcal{G}})\right]_h$ are all zeros. This gives rise to

$$E_2 = E_1 - \mathrm{MIB}(\tilde{\mathcal{G}}). \quad \square \tag{57}$$

This shows that the difference in discretization error of Scheme 1 and Scheme 2 is $\mathrm{MIB}(\tilde{\mathcal{G}})$. Therefore, adding the correction $\mathrm{MIB}(\tilde{\mathcal{G}})$ as in Eq. (49) reduces the discretization error of Scheme 1 from E_1 down to E_2.

4.3. An additional note on numerical equivalence

We notice that in the MIB framework, Schemes 1 and 3 have almost the same accuracy. This is because $\epsilon^+\Delta\mathcal{G}$ has also been dropped in Scheme 3. For a more detailed analysis, let us use the same notation above. We will prove the following claim:

Proposition 2: Using the MIB, the discretization error of Scheme 1 given by (51) equals the total discretization errors of equation (25) and (27)

$$\begin{aligned} E_3 = &\left[-\nabla\cdot(\epsilon\nabla\bar{\Phi})\right]_h + \bar{\kappa}^2\left[\sinh(\mathcal{G}+\bar{\Phi}+\Psi)\right]_h - \mathrm{MIB}(\bar{\Phi}) - \bar{\kappa}^2\sinh\left(\tilde{\mathcal{G}}|_h + \bar{\Phi}|_h + \Psi|_h\right) \\ &+ [-\nabla\cdot(\epsilon\nabla\Psi)]_h - \mathrm{MIB}(\Psi). \end{aligned} \tag{58}$$

Proof. We first note that equation (25) can be rewritten as

$$\begin{cases} -\epsilon^-\Delta\Psi = 0, & r\in\Omega^- \\ -\epsilon^+\Delta\Psi = 0, & r\in\Omega^+ \\ [\Psi]_\Gamma = 0, & r\in\Gamma \\ \left[\epsilon\frac{\partial\Psi}{\partial n}\right]_\Gamma = (\epsilon^- - \epsilon^+)\frac{\partial\mathcal{G}}{\partial n} & r\in\Gamma \\ \Psi(r) = \phi_b(r) - \mathcal{G}(r), & r\in\partial\Omega. \end{cases} \tag{59}$$

That is why the discretization error of Ψ in (58) is expressed as $[-\nabla\cdot(\epsilon\nabla\Psi)]_h - \mathrm{MIB}(\Psi)$. Similarly, the error E_3 is primarily due to the difference between elliptic operators and their MIB discretizations,

$$E_3 = [-\nabla\cdot(\epsilon\nabla\Psi)]_h - \mathrm{MIB}(\Psi) + \left[-\nabla\cdot(\epsilon\nabla\bar{\Phi})\right]_h - \mathrm{MIB}(\bar{\Phi}).$$

Using the fact that $\phi_{RF} = \Psi + \bar{\Phi}$ and the linearity, we have

$$\begin{aligned} E_3 - E_1 =& \left([-\nabla\cdot(\epsilon\nabla\Psi)]_h - \mathrm{MIB}(\Psi) + \left[-\nabla\cdot(\epsilon\nabla\bar{\Phi})\right]_h - \mathrm{MIB}(\bar{\Phi})\right) \\ &- \left([-\nabla\cdot(\epsilon\nabla\phi_{RF})]_h - \mathrm{MIB}(\phi_{RF})\right) \\ =& \left[-\nabla\cdot(\epsilon\nabla\Psi) - \nabla\cdot(\epsilon\nabla\bar{\Phi}) + \nabla\cdot(\epsilon\nabla\phi_{RF})\right]_h + \left(\mathrm{MIB}(\phi_{RF}) - \mathrm{MIB}(\Psi) - \mathrm{MIB}(\bar{\Phi})\right) \\ =& 0. \quad \square \end{aligned}$$

This shows E_3 and E_1 are equivalent. Therefore, similar to Scheme 1, to improve the accuracy of Scheme 3, we propose adding $\mathrm{MIB}(\tilde{\mathcal{G}})$ to the discretization of Eq. (25) or (27), that is for keeping $\epsilon^+\Delta\mathcal{G}$ in the equation (27) instead of dropping it. Notice that doing this might not improve the accuracy in finding Ψ or $\bar{\Phi}$ separately but it reduces the total error in discretizing Eqs. (25) and (27) down to E_2 (see Eq. (54)) and thus receives a higher accuracy in calculating the sum $\Psi + \bar{\Phi}$.

Table 4
Solving the Poisson equation and the PBE using the original Schemes 1 and 3 and the improved Schemes 1 and 3 on a spherical cavity with a unit charge e_c centered at (0,0,0): spherical radius $R = 2$, $\epsilon^+ = 80$, $\epsilon^- = 1$, $\kappa = 0$ for the Poisson equation and $\kappa = 0.1257$ for the PBE.

h	The Poisson equation						The PBE					
	Schemes 1 & 3			Improved Schemes 1 & 3			Schemes 1 & 3			Improved Schemes 1 & 3		
	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}	e_ϕ	ord.	E_{sol}
1	6.6E-02		-88.08	8.4E-04		-81.94	5.3E-02		-86.21	6.7E-04		-81.96
1/2	1.2E-02	2.45	-83.35	1.5E-04	2.45	-81.96	9.1E-03	2.54	-82.86	1.2E-04	2.54	-81.97
1/4	3.3E-03	1.88	-82.21	4.2E-05	1.88	-81.98	2.5E-03	1.84	-82.12	3.2E-05	1.85	-81.98
1/8	6.7E-04	2.29	-82.02	8.5E-06	2.29	-81.98	5.3E-04	2.27	-82.01	6.7E-06	2.27	-81.98
1/16	1.2E-04	2.54	-81.99	1.5E-06	2.54	-81.98	9.4E-05	2.48	-81.98	1.5E-06	2.14	-81.98

Table 5
Comparison on solvation energies produced between improved Schemes 1 and 3 and the original Schemes 1 and 3 for solving the nonlinear PBE on a collection of 24 proteins. Results from Scheme 2 as in Table 3 are also listed for reference.

Protein	Schemes 1 & 3			Improved Schemes 1 & 3			Scheme 2		
	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 1$	$h = 0.5$	$h = 0.25$	$h = 1$	$h = 0.5$	$h = 0.25$
1ajj	-1215.31	-1158.57	-1144.51	-1137.74	-1139.74	-1140.84	-1137.74	-1139.74	-1140.84
2erl	-1037.81	-968.68	-956.27	-966.07	-952.54	-953.24	-966.07	-952.54	-953.24
1cbn	-343.29	-312.67	-306.06	-307.77	-303.38	-304.05	-307.77	-303.38	-304.05
1vii	-1010.19	-923.74	-908.34	-910.99	-902.58	-903.88	-910.99	-902.58	-903.88
1fca	-1300.95	-1224.75	-1209.19	-1217.07	-1204.66	-1205.72	-1217.07	-1204.66	-1205.72
1bbl	-1089.69	-1013.57	-994.72	-994.44	-988.64	-989.35	-994.44	-988.64	-989.35
2pde	-869.97	-834.73	-825.39	-818.30	-821.21	-823.10	-818.30	-821.21	-823.10
1sh1	-835.22	-774.34	-759.13	-749.89	-754.20	-754.72	-749.89	-754.20	-754.72
1vjw	-1361.52	-1267.03	-1246.77	-1251.24	-1241.28	-1242.26	-1251.24	-1241.28	-1242.26
1uxc	-1291.49	-1170.23	-1147.38	-1154.48	-1139.50	-1141.66	-1154.48	-1139.50	-1141.66
1ptq	-1006.64	-897.39	-879.44	-894.04	-873.50	-874.88	-894.04	-873.50	-874.88
1bor	-940.52	-873.05	-858.85	-854.34	-853.65	-855.19	-854.34	-853.65	-855.19
1fxd	-3458.41	-3354.86	-3329.36	-3327.40	-3321.94	-3322.97	-3327.40	-3321.94	-3322.97
1r69	-1210.97	-1113.31	-1095.15	-1107.15	-1088.84	-1090.65	-1107.15	-1088.84	-1090.65
1mbg	-1491.37	-1379.41	-1360.05	-1370.45	-1353.60	-1354.95	-1370.45	-1353.60	-1354.95
1bpi	-1446.91	-1336.96	-1313.61	-1322.61	-1304.61	-1306.54	-1322.61	-1304.61	-1306.54
1hpt	-918.19	-837.44	-819.49	-825.13	-811.71	-814.52	-825.13	-811.71	-814.52
451c	-1141.25	-1055.74	-1034.04	-1024.48	-1026.06	-1027.86	-1024.48	-1026.06	-1027.86
1svr	-1932.94	-1752.30	-1721.70	-1734.47	-1711.50	-1713.67	-1734.47	-1711.50	-1713.67
1frd	-3081.05	-2905.01	-2874.81	-2887.75	-2863.26	-2867.03	-2887.75	-2863.26	-2867.03
1a2s	-2072.11	-1954.00	-1929.64	-1930.10	-1921.65	-1923.66	-1930.10	-1921.65	-1923.66
1neq	-1984.20	-1778.67	-1744.72	-1759.59	-1732.17	-1735.20	-1759.59	-1732.17	-1735.20
1a63	-2694.42	-2436.70	-2391.65	-2407.65	-2375.10	-2378.94	-2407.65	-2375.10	-2378.94
1a7m	-2403.14	-2214.41	-2174.58	-2182.78	-2160.92	-2164.17	-2182.78	-2160.92	-2164.17

4.4. Numerical verification

In this section, we improve Schemes 1 and 3 using the recovery techniques as described above and test these improved schemes on the same benchmark problems studied in Section 3.3.

The improvements can be clearly observed in Table 4, which reports the results of solving the Poisson equation (32) and the nonlinear PBE (33) with the improved schemes as compared with the original results from Table 1. The errors produced by improved Schemes 1 and 3 now are as accurate as those from Schemes 2 and 4 in Table 1.

Following the same fashion, we also solve the nonlinear PBE using the improved Schemes 1 and 3 on the set of 24 proteins and report the solvation energies in Table 5 as compared with the results in Table 3 using the original Schemes 1 and 3.

To better present the information embedded in Table 5, we plot the data in a reorganized way in Fig. 6. Here we choose the solvation energies produced by Scheme 2 with different step size $h = 1, 0.5, 0.25$ as the reference and calculate the difference ΔE_{sol} between the reference and the solvation energies produced by the improved Schemes 1 and 3 with different step sizes. The difference between these energies is equal to $C \times 10^{-6}$ where C is a positive constant with magnitude less than 10 and 10^{-6} is the tolerance chosen for the inexact Newton method and the biconjugate solver. This shows that the results produced by the improved Schemes 1 and 3 are as accurate as that from Scheme 2 as opposed to large difference using the original Schemes 1 and 3 in Fig. 4(a). These results verify our analysis in section 4 and show that the recovery technique successfully restores the accuracy of Scheme 1 and Scheme 3.

Furthermore, we plot the number of iterations and CPU time using Scheme 1 and its improved version in Fig. 7 and we can see that the improvement does not significantly affect the efficiency of the scheme.

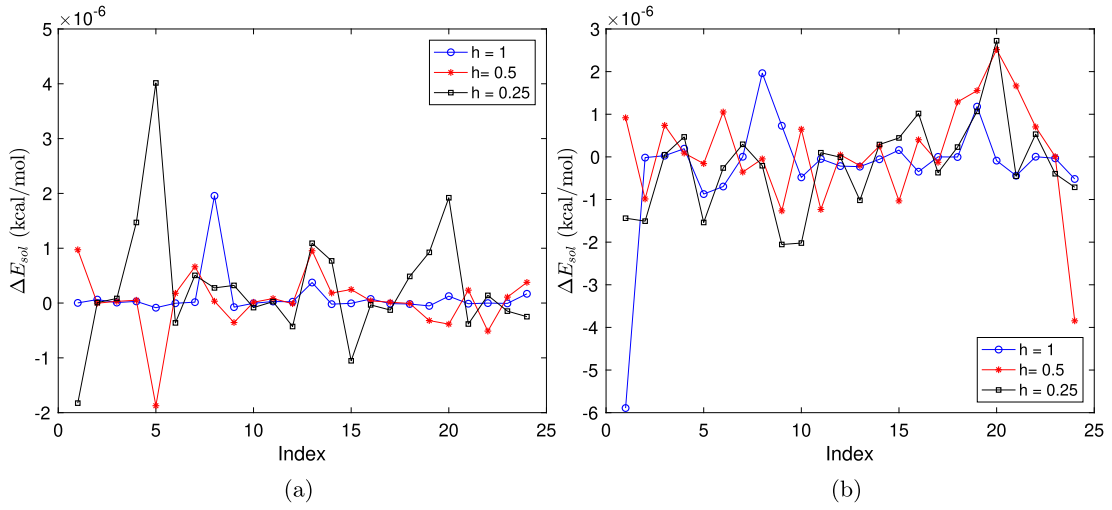

Fig. 6. Differences between solvation energies calculated with (a) Improved Scheme 1 and Scheme 2 and (b) Improved Scheme 3 and Scheme 2 for 24 proteins.

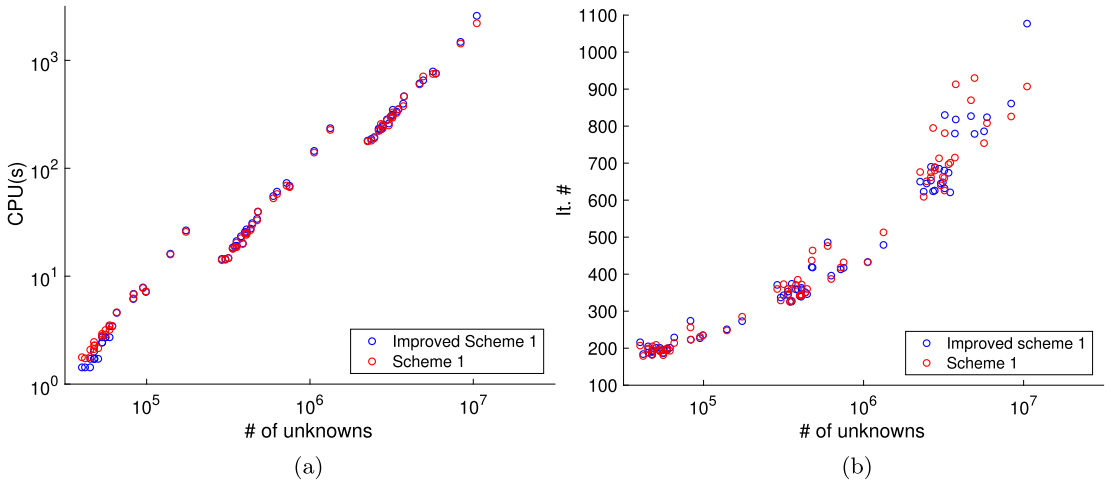

Fig. 7. (a) CPU time and (b) number of iterations of Scheme 1 (red) and Improved Scheme 1 (blue) for 24 proteins.

5. Conclusion

In this paper, we investigate the regularization methods for treating the charge singularities involved in the Poisson-Boltzmann equation. We implement four regularization schemes under the matched interface and boundary (MIB) framework and discover that although these schemes are analytically equivalent, their numerical implementations demonstrate different accuracy. In particular, our numerical experiments on several benchmark problems show that Scheme 2 [6,18] and Scheme 4 [9] produce better accuracy than Scheme 1 [19,47] and Scheme 3 [38]. Inspired by the error analysis in [23], we found that in Schemes 1 and 3, the subtraction of the Green's function term from the decomposition of the solution ϕ scales up the regularized solution thus causes accuracy reduction. In terms of the regularized PDEs to solve, we found that schemes that drop $\Delta\mathcal{G}$ show lower accuracy. Based on these findings, we propose a recovery technique to compensate for the accuracy reduction. The numerical results show that this technique can perfectly recover the accuracy, thus verifying our proposed explanation to the problems and the effectiveness of the recovery technique. Based on these works, all regularization schemes are well understood and connected in a unified framework, which is an important contribution to the development of the Green's function based decomposition schemes for treating source singularities.

CRediT authorship contribution statement

Arum Lee: Formal analysis, Software, Validation, Visualization, Writing - original draft. **Weihua Geng:** Methodology, Software, Writing - original draft. **Shan Zhao:** Conceptualization, Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The research of Dr. Geng was supported in part by The National Science Foundation (NSF) grant DMS-1819193. The research of Dr. Zhao was supported in part by the NSF grant DMS-1812930.

References

[1] Chandrajit Bajaj, Shun-Chuan Chen, Alexander Rand, An efficient higher-order fast multipole boundary element solution for Poisson-Boltzmann-based molecular electrostatics, SIAM J. Sci. Comput. 33 (2) (2011) 826–848.
[2] N.A. Baker, Improving implicit solvent simulations: a Poisson-centric view, Curr. Opin. Struct. Biol. 15 (2) (2005) 137–143.
[3] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Electrostatics of nanosystems: application to microtubules and the ribosome, Proc. Natl. Acad. Sci. USA 98 (18) (2001) 10037–10041.
[4] Peter Benner, Venera Khoromskaia, Boris Khoromskij, Cleophas Kweyu, Matthias Stein, Computing electrostatic potentials using regularization based on the range-separated tensor format, arXiv:1901.09864v1, 2019.
[5] Alexander H. Boschitsch, Marcia O. Fenley, Huan-Xiang Zhou, Fast boundary element method for the linear Poisson-Boltzmann equation, J. Phys. Chem. B 106 (10) (2002) 2741–2754.
[6] Qin Cai, Jun Wang, Hong-Kai Zhao, Ray Luo, On removal of charge singularity in Poisson-Boltzmann equation, J. Chem. Phys. 130 (14) (2009).
[7] Duan Chen, Zhan Chen, Changjun Chen, Weihua Geng, G.W. Wei, MIBPB: a software package for electrostatic analysis, J. Comput. Chem. 32 (2011) 657–670.
[8] L. Chen, M.J. Holst, J. Xu, The finite element approximation of the nonlinear Poisson-Boltzmann equation, SIAM J. Numer. Anal. 45 (2007) 2298–2320.
[9] I.L. Chern, J.-G. Liu, W.-C. Weng, Accurate evaluation of electrostatics for macromolecules in solution, Methods Appl. Anal. 10 (2) (2003) 309–328.
[10] M.E. Davis, J.D. Madura, J. Sines, B.A. Luty, S.A. Allison, J.A. McCammon, Diffusion-controlled enzymatic reactions, Methods Enzymol. 202 (1991) 473–497.
[11] W. Deng, X. Zhufu, J. Xu, S. Zhao, A new discontinuous Galerkin method for the nonlinear Poisson-Boltzmann equation, Appl. Math. Lett. 257 (2015) 1000–1021.
[12] Raphael Egan, Frédéric Gibou, Geometric discretization of the multidimensional Dirac delta distribution – application to the Poisson equation with singular source terms, J. Comput. Phys. 346 (2017) 71–90.
[13] Raphael Egan, Frédéric Gibou, Fast and scalable algorithms for constructing solvent-excluded surfaces of large biomolecules, J. Comput. Phys. 374 (2018) 91–120.
[14] F. Fogolari, A. Brigo, H. Molinari, The Poisson-Boltzmann equation for biomolecular electrostatics: a tool for structural biology, J. Mol. Recognit. 15 (6) (2002) 377–392.
[15] Weihua Geng, A boundary integral Poisson–Boltzmann solvers package for solvated bimolecular simulations, Comput. Math. Biophys. 3 (2015) 43–58.
[16] Weihua Geng, Robert Krasny, A treecode-accelerated boundary integral Poisson-Boltzmann solver for electrostatics of solvated biomolecules, J. Comput. Phys. 247 (2013) 62–78.
[17] Weihua Geng, Sining Yu, G.W. Wei, Treatment of charge singularities in implicit solvent models, J. Chem. Phys. 127 (2007) 114106.
[18] Weihua Geng, Shan Zhao, A two-component matched interface and boundary (MIB) regularization for charge singularity in implicit solvation, J. Comput. Phys. 351 (2017) 25–39.
[19] M.K. Gilson, M.E. Davis, B.A. Luty, J.A. McCammon, Computation of electrostatic forces on solvated molecules using the Poisson-Boltzmann equation, J. Phys. Chem. 97 (14) (1993) 3591–3600.
[20] Leslie Greengard, Denis Gueyffier, Per-Gunnar Martinsson, Vladimir Rokhlin, Fast direct solvers for integral equations in complex three-dimensional domains, Acta Numer. 18 (2009) 243–275, 005.
[21] Michael J. Holst, The Poisson-Boltzmann Equation: Analysis and Multilevel Numerical Solution, PhD thesis, UIUC, 1994.
[22] Michael J. Holst, Faisal Saied, Numerical solution of the nonlinear Poisson-Boltzmann equation: developing more robust and efficient methods, J. Comput. Chem. 16 (3) (1995) 337–364.
[23] M.J. Holst, J.A. McCammon, Z. Yu, Y.C. Zhou, Y. Zhu, Adaptive finite element modeling techniques for the Poisson-Boltzmann equation, Commun. Comput. Phys. 11 (2012) 179–214.
[24] B. Honig, A. Nicholls, Classical electrostatics in biology and chemistry, Science 268 (5214) (1995) 1144–1149.
[25] W. Im, D. Beglov, B. Roux, Continuum solvation model: electrostatic forces from numerical solutions to the Poisson-Boltzmann equation, Comput. Phys. Commun. 111 (1–3) (1998) 59–75.
[26] A. Juffer, E. Botta, B. van Keulen, A. van der Ploeg, H. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach, J. Comput. Phys. 97 (1991) 144–171.
[27] Boris N. Khoromskij, Range-separated tensor decomposition of the discretized Dirac delta and elliptic operator inverse, J. Comput. Phys. 401 (2020) 108998.
[28] J. Liang, S. Subranmaniam, Computation of molecular electrostatics with boundary element methods, Biophys. J. 73 (1997) 1830–1841.
[29] B.Z. Lu, Y.C. Zhou, M.J. Holst, J.A. McCammon, Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications, Commun. Comput. Phys. 3 (5) (2008) 973–1009.
[30] Benzhuo Lu, Xiaolin Cheng, J. Andrew McCammon, A new-version-fast-multipole-method-accelerated electrostatic calculations in biomolecular systems, J. Comput. Phys. 226 (2) (2007) 1348–1366.
[31] R. Luo, L. David, M.K. Gilson, Accelerated Poisson-Boltzmann calculations for static and dynamic systems, J. Comput. Chem. 23 (13) (2002) 1244–1253.
[32] Mohammad Mirzadeh, Maxime Theillard, Frédéric Gibou, A second-order discretization of the nonlinear Poisson–Boltzmann equation over irregular geometries using non-graded adaptive Cartesian grids, J. Comput. Phys. 230 (5) (2011) 2125–2140.
[33] Mohammad Mirzadeh, Maxime Theillard, Asdis Helgadottir, David Boy, Frédéric Gibou, An adaptive, finite difference solver for the nonlinear Poisson-Boltzmann equation with applications to biomolecular computations, Commun. Comput. Phys. 13 (1) (2013) 150–173.
[34] C. Quan, B. Stamm, Y. Maday, A domain decomposition method for the Poisson–Boltzmann solvation models, SIAM J. Sci. Comput. 41 (2) (2019) B320–B350.
[35] Anna-Karin Tornberg, Bjorn Engquist, Numerical approximations of singular source terms in differential equations, J. Comput. Phys. 200 (2) (2004) 462–488.

[36] John D. Towers, A source term method for Poisson problems on irregular domains, J. Comput. Phys. 361 (2018) 424–441.
[37] Siwen Wang, Arum Lee, Emil Alexov, Shan Zhao, A regularization approach for solving Poisson's equation with singular charge sources and diffuse interfaces, Appl. Math. Lett. 102 (2020) 106144.
[38] D. Xie, New solution decomposition and minimization schemes for Poisson-Boltzmann equation in calculation of biomolecular electrostatics, J. Comput. Phys. 275 (2014) 294–309.
[39] Jinyong Ying, Dexuan Xie, A new finite element and finite difference hybrid method for computing electrostatics of ionic solvated biomolecule, J. Comput. Phys. 298 (2015) 636–651.
[40] Sining Yu, Weihua Geng, G.W. Wei, Treatment of geometric singularities in implicit solvent models, J. Chem. Phys. 126 (2007) 244108.
[41] Sining Yu, G.W. Wei, Three-dimensional matched interface and boundary (MIB) method for treating geometric singularities, J. Comput. Phys. 227 (2007) 602–632.
[42] Sining Yu, Yongcheng Zhou, G.W. Wei, Matched interface and boundary (MIB) method for elliptic problems with sharp-edged interfaces, J. Comput. Phys. 224 (2) (2007) 729–756.
[43] Bo Zhang, Benzhuo Lu, Xiaolin Cheng, Jingfang Huang, Nikos P. Pitsianis, Xiaobai Sun, J. Andrew McCammon, Mathematical and numerical aspects of the adaptive fast multipole Poisson-Boltzmann solver, Commun. Comput. Phys. 13 (1) (2013) 107–128, 001.
[44] Yimin Zhong, Kui Ren, Richard Tsai, An implicit boundary integral method for computing electric potential of macromolecules in solvent, J. Comput. Phys. 359 (2018) 199–215.
[45] Y.C. Zhou, G.W. Wei, On the fictitious-domain and interpolation formulations of the matched interface and boundary (MIB) method, J. Comput. Phys. 219 (1) (2006) 228–246.
[46] Y.C. Zhou, Shan Zhao, Michael Feig, G.W. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (1) (2006) 1–30.
[47] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn, M. Levitt, Finite-difference solution of the Poisson-Boltzmann equation: complete elimination of self-energy, J. Comput. Chem. 17 (1996) 1344–1351.