Applied Mathematics and Computation 380 (2020) 125267

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Pseudo-transient ghost fluid methods for the Poisson-Boltzmann equation with a two-component regularization

Sheik Ahmed Ullah, Shan Zhao*

Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA

ARTICLE INFO

Article history:
Received 26 July 2019
Revised 2 February 2020
Accepted 22 March 2020

Keywords:
Poisson-Boltzmann Equation
Regularization
Pseudo-transient continuation approach
Ghost Fluid Method
Alternating direction implicit (ADI)

ABSTRACT

The Poisson Boltzmann equation (PBE) is a well-established implicit solvent continuum model for the electrostatic analysis of solvated biomolecules. The numerical solution of the nonlinear PBE is still a challenge due to its exponential nonlinear term, strong singularity by the source terms, and distinct dielectric regions. In this paper, a new pseudo-transient approach is proposed, which combines an analytical treatment of singular charges in a two-component regularization, with an analytical integration of nonlinear term in pseudo-time solution. To ensure efficiency, both fully implicit alternating direction implicit (ADI) and unconditionally stable locally one-dimensional (LOD) methods have been constructed to decompose three-dimensional linear systems into one-dimensional (1D) ones in each pseudo-time step. Moreover, to accommodate the nonzero function and flux jumps across the dielectric interface, a modified ghost fluid method (GFM) has been introduced as a first order accurate sharp interface method in 1D style, which minimizes the information needed for the molecular surface. The 1D finite-difference matrix generated by the GFM is symmetric and diagonally dominant, so that the stability of ADI and LOD methods is boosted. The proposed pseudo-transient GFM schemes have been numerically validated by calculating solvation free energy, binding energy, and salt effect of various proteins. It has been found that with the augmentation of regularization and GFM interface treatment, the ADI method not only enhances the accuracy dramatically, but also improves the stability significantly. By using a large time increment, an efficient protein simulation can be realized in steady-state solutions. Therefore, the proposed GFM-ADI and GFM-LOD methods provide accurate, stable, and efficient tools for biomolecular simulations.

1. Introduction

Solvated biomolecules and their electrostatic interaction with the surrounding solvent are critical to the studies of various important biological processes such as protein-drug binding site analysis, DNA recognition, protein folding, and protein-ligand binding. In the past few decades with the development of numerical methods and computational powers, the electrostatic analysis of functions and dynamics of biomolecular solvation has become more practical and effective. However, imitating these interactions is still computationally expensive for biologically significant systems. Here we are considering

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

https://doi.org/10.1016/j.amc.2020.125267
0096-3003/

the Poisson-Boltzmann Equation (PBE) to describe the electrostatic potential generated by a low dielectric medium inside a protein molecule with embedded atomic charges solvated in a high dielectric medium with dissolved ions. The analytical solution of the PBE is only available for some simple geometry such as a sphere or cylinder. Efficiency and accuracy are still critical issues in the numerical solution of the PBE for biophysical models with complex geometry, especially for macromolecules containing tens of thousands to millions of atoms.

As a commonly used mathematical model for the electrostatic analysis, the PBE is a nonlinear elliptic equation on multiple domains with discontinuous dielectric coefficients separated by the solute-solvent interface or molecular surface. The numerical difficulties for solving the PBE arise from nonlinearity, discontinuous dielectric coefficients, non-smoothness of the solution, and singularities in the source term due to the atomic charges. Effects of nonlinearity become significant with strong ionic presence [43].

In treating the PBE nonlinearity, two different approaches have been developed in the literature. The usual approach is to discretize the nonlinear PBE into an algebraic system using finite difference or finite element methods and then solve it by a nonlinear algebraic approach, such as nonlinear relaxation method [21,33], nonlinear conjugate gradient method [31] or inexact Newton method [20]. However, a blow-up or overflow problem [44] could be encountered in nonlinear algebraic methods, because the hyperbolic Sine function could take exponentially large values. The other approach [35,37,46] has been introduced recently based on the pseudo-transient continuation idea [1,22,32]. This approach converts the nonlinear PBE into a time-dependent form by introducing a pseudo-time derivative. The solution to the original problem is then retrieved from the steady-state solution of the time-dependent PBE. The main advantage of introducing pseudo-time derivative is to be able to split the time-dependent PBE into linear and nonlinear subsystems, so that the blow-up problem can be circumvented by analytically integrating the nonlinear subsystem [14,47]. For this reason, the pseudo-transient approach will be adopted in the present study.

In pseudo-time methods, the time-dependent PBE has to be solved until steady-state. To maintain efficiency, a large pseudo-time increment for Δt is required. This is why the existing pseudo-time methods usually adopted an implicit scheme in time stepping. Moreover, to convert the three dimensional (3D) PBE into a set of independent one-dimensional (1D) systems, both alternating direction implicit (ADI) [14,47] and locally one dimensional (LOD) [43] methods have been introduced in the literature. Especially in [14] the Douglas-Rachford ADI scheme has been used to split the linear subsystem with the 3D Laplacian operator into three subsystems with one-dimensional second order derivatives. Altogether this method has the first-order accuracy in time but is subject to a quite severe stability condition. Later in [43], the LOD method was introduced as an unconditionally stable method with reduced accuracy compared to ADI methods in [14]. Even though 1D subsystems produced by these ADI and LOD methods are tridiagonal and can be efficiently solved by using the Thomas Algorithm [39], they lack the treatment of the jump conditions at the interface which reduces the spatial accuracy near the interface. Also, the numerical error for these pseudo-transient methods has been observed often to be dominated by the singularity at the center of the atoms.

Besides strong nonlinearity, the numerical treatment of charge singularity is another challenge for the PBE. At atom centers, both the charge source and the potential solution blow up, and the conventional discretization is doomed to be inaccurate. This motivated many authors to develop different regularization methods [3,5,13,15,19,44,51] to avoid the loss of accuracy due to the singularity. In these methods, the potential function is decomposed into a singular component plus one or two other components to break down the PBE into a system of partial differential equations (PDEs) containing a Poisson equation with the singular term plus other equations. Thus the singular component can be handled separately using the analytical solution for the Poisson equation in terms of Coulomb potentials or Green's functions. So far, these type of regularization methods have never been used with the ADI or LOD type pseudo-transient methods.

The dielectric interface is also crucial in the numerical discretization of the PBE as it defines the boundary for the solute and solvent regions. Across a geometrically complex dielectric interface, or molecular surface, the potential solution is continuous, but its normal derivative is discontinuous. For the un-regularized PBE, the standard finite difference method is still convergent but degenerates to first-order convergence in space. However, the situation becomes worse for the regularized PBE [15] when both the potential and its flux are discontinuous across the interface for the regularized potential. The standard finite difference solution will diverge in this case, if no interface treatment is imposed. For this reason, the regularized PBE is usually solved by some special interface schemes, such as the matched interface and boundary (MIB) method [4,13,45,49,50], and finite element methods (FEMs) [9,23,44]. In particular, based on rigorous analysis and computation of error bounds, a reliable adaptive FEM has recently been developed for calculating electrostatic potentials in high precision near a Gaussian molecular surface [23]. We note that other advanced interface algorithms, such as boundary integral method [38], high order ghost fluid methods [2,8,17], and immersed finite volume method [10], can also be employed for solving the regularized PBE.

Nevertheless, when we are concerned with the pseudo-transient approach for the nonlinear PBE, the choice of interface algorithms becomes limited. For instance, pseudo time FEMs have been developed in [9] for solving both un-regularized and regularized PBEs. However, such pseudo time methods are inefficient by solving a large linear system at each pseudo-time step. Aiming to maintain the efficiency and stability of the ADI methods while restoring the accuracy to the second-order near the interface, several interface schemes have been developed in [28,29] for solving the diffusion equation. Then as a continuation of this approach recently matched ADI (mADI) method was developed in [27,42,48] to combine the MIB method (for interface treatment) with the ADI. But these ADI methods were mainly focused on the parabolic equations and

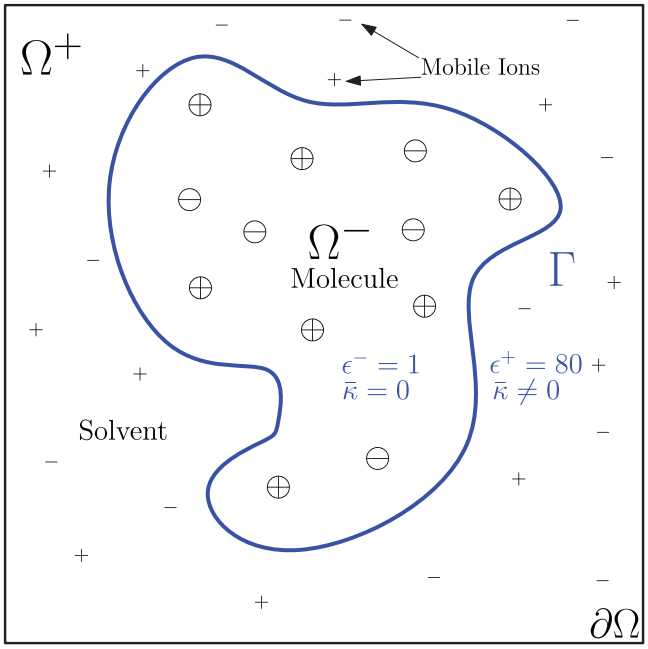

Fig. 1. Poisson-Boltzmann Model.

have never been applied to the PBE. In fact, the mADI [27,42,48] could become cumbersome in treating a very complicated interface, like the molecular surface in protein studies.

In this paper, our goal was to develop a new approach to solve the PBE combining the regularization, the pseudo-transient continuation, and the interface treatment so that both nonlinearity and singularity are properly treated. For the regularization, we have chosen the two-component regularization developed in [15], which is an efficient and accurate regularization method. But it changes the jump condition to be nonzero, which introduces the necessity of interface treatments. Otherwise, the standard central finite difference becomes divergent. Then motivated by the mADI method, we have modified the ghost fluid method (GFM) developed in [11] as the interface treatment for the present study. Like the original GFM [11], the modified GFM to be constructed is just first order accurate in space. We note that the GFM family has been dramatically improved in recent years to achieve very high order of accuracy [2,8,17]. However, our numerical setting requires the interface treatment to be carried out only in a 1D manner, in order to fit to the ADI framework. In this regard, the original GFM [11] becomes an ideal choice, because it can correct finite difference approximation, while maintaining the ADI efficiency. Moreover, the simplicity of the GFM allows it to be applied for large proteins with complicated geometry.

The modified GFM will be combined with the Douglas-Rachford ADI to generate a new PBE solver: GFM-ADIDR, which will significantly improve the accuracy and efficiency in comparing with the original ADI method [14]. But, it still has a time stability constraint when the pseudo-time step size is too big. To achieve a better stability, two LOD schemes will also be considered to replace the ADIDR scheme.

The rest of this work is organized as follows. Section 2 will start with the electrostatic of with the PBE. Then the two-component regularization and its incorporation will be discussed with the pseudo-transient approaches. Three types of operator splitting methods will be proposed in this section to solve the PBE. A modified GFM and its detailed derivation will be provided in this section. Section 3 will examine the numerical validation for the three proposed methods for the Kirkwood Sphere problem and other biological problems. Section 4 summarizes the findings made in the previous section and proposes some opportunities for future work.

2. Theory and Algorithm

2.1. Poisson-Boltzmann Equation (PBE)

The Poisson-Boltzmann Equation (PBE) is the governing equation of electrostatics for a solute macromolecule immersed in an aqueous solvent environment illustrated in Fig. 1. The computational domain $\Omega \in \mathbb{R}^3$ is separated into two regions, Ω^- and Ω^+ by the molecular surface Γ, which is an arbitrarily shaped dielectric interface. Here Ω^- is the molecule domain with dielectric constant ϵ^- and Ω^+ is the solvent domain with dielectric constant ϵ^+. The cubic shape boundary of $\Omega = \Omega^- \cup \Omega^+$ is denoted by $\partial\Omega$. The charges outside in Ω^+ are mobile ions which are described by the Boltzmann distribution. Then the

electrostatic interaction of this solute-solvent system for $\mathbf{r} \in \Omega$ is governed by the nonlinear PBE as

$$-\nabla \cdot (\epsilon(\mathbf{r})\nabla\phi(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r})\sinh(\phi(\mathbf{r})) = \rho(\mathbf{r}), \tag{1}$$

where $\phi(\mathbf{r})$ is the electrostatic potential and the singular source $\rho(\mathbf{r})$ term is defined as

$$\rho(\mathbf{r}) = 4\pi \frac{e_c^2}{k_B T} \sum_{i=1}^{N_c} q_i \delta(\mathbf{r} - \mathbf{r}_i). \tag{2}$$

A Dirichlet boundary condition is used in biomolecular simulations [20] as

$$\phi(\mathbf{r}) = \phi_b(\mathbf{r}) := \frac{e_c^2}{k_B T} \sum_{i=1}^{N_c} \frac{q_i e^{-|\mathbf{r}-\mathbf{r}_i|\sqrt{\frac{\bar{\kappa}^2}{\epsilon^+}}}}{\epsilon^+ |\mathbf{r} - \mathbf{r}_i|}, \quad \text{on } \partial\Omega.$$

Here N_c is the total number of atoms in the solute molecule, k_B is the Boltzmann constant, e_c is the fundamental charge and q_i, in the same unit as e_c is the partial charge on the ith atom of the solute molecule located at position $\mathbf{r}_i$. The Debye-Huckel parameter $\bar{\kappa}^2 = \left(\frac{2N_A e_c^2}{100 k_b T}\right) I_s = 8.486902807 \text{Å}^{-2} I_s$ [18] for $\mathbf{r} \in \Omega^+$ and $\bar{\kappa} = 0$ for $\mathbf{r} \in \Omega^-$. Here N_A is the Avogadro's Number and I_s is the molar ionic strength. We have converted the dimensionless electrostatic potential ϕ to the units kcal/mol/e_c at the room temperature T by multiplying it by 0.592183 [18]. The dielectric constant ϵ is a piecewise constant such that, $\epsilon(\mathbf{r}) = \epsilon^-$ for $\mathbf{r} \in \Omega^-$ and $\epsilon(\mathbf{r}) = \epsilon^+$ for $\mathbf{r} \in \Omega^+$. There are two conditions on Γ needed to be satisfied from the dielectric theory for the potential ϕ and flux density $\epsilon\phi_n$,

$$[\phi]_\Gamma = 0, \quad \text{and} \quad [\epsilon\phi_n]_\Gamma = 0. \tag{3}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the outer normal direction on the interface Γ and $\phi_n = \frac{\partial\phi}{\partial n}$ is the directional derivative in $\mathbf{n}$. The notation $[f]_\Gamma = f^+ - f^-$ represents the difference of the functional value across the interface Γ.

2.2. Two-component regularization for singular sources

To avoid the numerical difficulty due to the singular source term, we consider a two-component regularization proposed in [15]. For this regularization the electrostatic potential ϕ has been considered as the addition of the coulomb component ϕ_c and the reaction field component ϕ_{RF} as $\phi = \phi_c + \phi_{RF}$. Here ϕ_c satisfies the following Poisson's equation,

$$\begin{cases} -\epsilon^- \Delta\phi_C(\mathbf{r}) = \rho(\mathbf{r}) \text{in } \mathbb{R}^3; \\ \phi_C(\mathbf{r}) = 0. \text{as } |\mathbf{r}| \to \infty. \end{cases}$$

which has the analytical solution as the Green's function G for ϕ_C as

$$G(\mathbf{r}) = \frac{e_c^2}{k_B T} \sum_{i=1}^{N_c} \frac{q_i}{\epsilon^- |\mathbf{r} - \mathbf{r}_i|}. \tag{4}$$

In this approach [15], one is solving the original solution ϕ in Ω^+ instead of the reaction component ϕ_{RF}. So a regularized potential is defined as

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

The jump conditions for $\tilde{\phi}$ can be shown as [15],

$$\phi^+ = \phi_{RF}^- + \phi_C^-, \text{ and } \epsilon^+ \frac{\partial\phi^+}{\partial n} = \epsilon^- \frac{\partial\phi_{RF}^-}{\partial n} + \epsilon^- \frac{\partial\phi_C^-}{\partial n} \text{ on } \Gamma.$$

Thus, the regularized PBE (RPBE) with corresponding interface and boundary conditions takes the following form:

$$-\nabla \cdot (\epsilon^- \nabla\tilde{\phi}) = 0 \text{ in } \Omega^-, \tag{5}$$

$$-\nabla \cdot (\epsilon^+ \nabla\tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0 \text{ in } \Omega^+, \tag{6}$$

$$\left[\tilde{\phi}\right] = G \text{ on } \Gamma, \tag{7}$$

$$\left[\epsilon \frac{\partial\tilde{\phi}}{\partial n}\right] = \epsilon^- \frac{\partial G}{\partial n} \text{ on } \Gamma, \tag{8}$$

$$\tilde{\phi} = \phi_b \text{ on } \partial\Omega. \tag{9}$$

Now from (5) and (6) we can summarize that,

$$-\nabla \cdot (\epsilon \nabla \tilde{\phi}) + \bar{\kappa}^2 \sinh(\tilde{\phi}) = 0 \quad \text{in } \Omega^- \cup \Omega^+. \tag{10}$$

Numerically, one needs to solve the RPBE interface problem given in (5)-(9) for $\tilde{\phi}$ and to recover the original solution as $\phi = \tilde{\phi}$ in Ω^+ and $\phi = \tilde{\phi} + G$ in Ω^-. Here the Green's function G can be calculated analytically from (4). We note that in jump conditions (7) and (8) both the solution and its flux are discontinuous. This means that interface algorithms are required for preventing accuracy reduction. Moreover, we note that no pseudo-transient approach has been developed for the current RPBE.

2.3. Pseudo-transient continuation approach for the RPBE

Similar to un-regularized cases [14,47], we propose a pseudo-transient continuation approach as an indirect method to solve the RPBE as

$$\frac{\partial u(\mathbf{r}, t)}{\partial t} = \nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r}, t)) - \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r}, t)), \quad \text{in } \Omega^- \cup \Omega^+, \tag{11}$$

$$[u] = G, \quad \text{on } \Gamma, \tag{12}$$

$$\left[\epsilon \frac{\partial u}{\partial n}\right] = \epsilon^- \frac{\partial G}{\partial n}, \quad \text{on } \Gamma, \tag{13}$$

$$u = \phi_b, \quad \text{on } \partial\Omega. \tag{14}$$

Here the time-independent RPBE in (10) has been converted into a time-dependent regularized PBE (TRPBE) in (11) which has no singular term $\rho(\mathbf{r})$. The steady-state solution $u(\mathbf{r}, t)$ as t being sufficiently large will produce the original solution $\tilde{\phi}(\mathbf{r})$. As the initial condition, we used the electrostatic potential solved from a linearized PBE [46] or trivially $u(\mathbf{r}, 0) = 0$. Then we numerically integrate (11) for a sufficiently long period to get the steady-state solution. Here the sign on the right-hand side of Eq. (11) has been considered as the reverse of the Eq. (10) to ensure numerical stability.

However, there are still some challenges left for the numerical integration of the TRPBE (11) because of the requirement of long-time integration, for which explicit time-stepping methods are usually not efficient [35,37,46]. Hence we employ fully-implicit time splitting methods [14,47], which have been used to solve the time-dependent PBE in the literature.

2.4. Time discretization: operator splitting schemes

Let us consider a uniform mesh with a grid spacing h in all x, y and z directions having N_x, N_y and N_z as the number of the grid points in each direction. We assume the vector $U^n = \left\{u^n_{ijk}\right\}$ for $i = 1, \ldots, N_x$, $j = 1, \ldots N_y$, and $k = 1, \ldots N_z$ having all the nodal values of u at the time level t_n as its elements. We use two stages to update U^n at time level t_n to U^{n+1} at time level $t_{n+1} = t_n + \Delta t$. In these two stages at each pseudo-time step, we develop several types of operator splitting schemes for updating U^n.

2.4.1. Alternating direction implicit (ADI) scheme

We first propose an alternating direction implicit (ADI) method for solving the TRPBE. In this scheme at each pseudo-time step from t_n to t_{n+1}, the time-dependent equation TRPBE in (11) will be solved by a first-order time splitting into the following two equations,

$$\frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w) \text{ with } W^n = U^n \text{ and } t \in [t_n, t_{n+1}], \tag{15}$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\epsilon \nabla v) \text{ with } V^n = W^{n+1} \text{ and } t \in [t_n, t_{n+1}]. \tag{16}$$

Then, we take $U^{n+1} = V^{n+1}$. For Eq. (15) we have the analytical solution as [43]

$$W^{n+1} = \ln\left(\frac{\cosh(\frac{1}{2}\bar{\kappa}^2 \Delta t) + \exp(-W^n)\sinh(\frac{1}{2}\bar{\kappa}^2 \Delta t)}{\exp(-W^n)\cosh(\frac{1}{2}\bar{\kappa}^2 \Delta t) + \sinh(\frac{1}{2}\bar{\kappa}^2 \Delta t)}\right), \tag{17}$$

which will help us to avoid to the difficulty due to the nonlinear term with $\sinh(\cdot)$ in (1). Here the right hand side of Eq. (17) is just a function of W and Δt which allows us to rewrite the Eq. (17) as $W^{n+1} = F(W^n, \Delta t)$ to facilitate the future discussion. For the temporal discretization of the Eq. (16) we consider Backward-Euler integration in time,

$$v^{n+1}_{i,j,k} = v^n_{i,j,k} + \Delta t\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right) v^{n+1}_{i,j,k}, \tag{18}$$

where δ_x^2, δ_y^2 and δ_z^2 are the central finite difference operators for the x, y, and z directions, respectively. For example, $\delta_x^2(v_{i,j,k}^n) = \frac{\epsilon_{i,j,k}}{h^2}(v_{i-1,j,k}^n - 2v_{i,j,k}^n + v_{i+1,j,k}^n)$. Numerically, these central difference operators will be applied away from the interface Γ, where corrections are needed near Γ.

For the second stage of the operator splitting, a first-order Douglas-Rachford ADI (ADIDR) scheme is used to decompose the discretized diffusion process (18), in x, y and z directions as

$$\left(1 - \Delta t\delta_x^2\right)v_{i,j,k}^* = \left[1 + \Delta t\left(\delta_y^2 + \delta_z^2\right)\right]v_{i,j,k}^n, \tag{19}$$

$$\left(1 - \Delta t\delta_y^2\right)v_{i,j,k}^{**} = v_{i,j,k}^* - \Delta t\delta_y^2\left(v_{i,j,k}^n\right), \tag{20}$$

$$\left(1 - \Delta t\delta_z^2\right)v_{i,j,k}^{n+1} = v_{i,j,k}^{**} - \Delta t\delta_z^2\left(v_{i,j,k}^n\right). \tag{21}$$

where v^* and v^{**} are two intermediate values. Here, the three-dimensional linear algebraic system in Eq. (18) has been decomposed into one dimensional linear algebraic systems in (19), (20) and (21). Each one of these equations has a tridiagonal structure, and can be efficiently solved by the Thomas algorithm [39] with a complexity like $O(N_x \log N_x)$. Thus, the complexity for solving all tridiagonal systems in each pseudo-time step of the ADIDR is on the order of $O(N \log N)$, where $N = N_x \times N_y \times N_z$ is the spatial degree of freedom.

On the other hand, by eliminating $v_{i,j,k}^*$ and $v_{i,j,k}^{**}$ and solving for $v_{i,j,k}^{n+1}$ in (18) we can obtain,

$$v_{i,j,k}^{n+1} = v_{i,j,k}^n + \Delta t\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right)v_{i,j,k}^{n+1} - \Delta t^2\left(\delta_x^2\delta_y^2 + \delta_x^2\delta_y^2 + \delta_z^2\delta_y^2\right)\left(v_{i,j,k}^{n+1} - v_{i,j,k}^n\right) + \Delta t^3\delta_x^2\delta_y^2\delta_z^2\left(v_{i,j,k}^{n+1} - v_{i,j,k}^n\right). \tag{22}$$

Hence the Douglas-Rachford scheme is a higher-order perturbation of the Implicit-Euler method (18). Since both (15) and (16) are first order in time this proposed ADIDR scheme is also first-order accurate in time. For the boundary conditions, the same Dirichlet boundary values of u are assumed for v, v^* and v^{**}. The entire time integration here is fully implicit. Previously, when a similar ADI scheme was constructed for the un-regularized PBE, a severe stability constraint was encountered [14].

2.4.2. Crank-Nicolson method with Locally One Dimensional Scheme (LODCN)

We next construct a locally one dimensional (LOD) scheme and discretize it by the Crank-Nicolson method. To this end, the TRPBE will be solved by a second-order time splitting method in three stages [14] as

$$\begin{aligned}
&\frac{\partial w}{\partial t} = -\frac{1}{2}\bar{\kappa}^2\sinh(w) \text{ with } W^n = U^n \text{ and } t \in [t_n, t_{n+1}],\\
&\frac{\partial v}{\partial t} = \nabla\cdot(\epsilon\nabla v) \text{ with } V^n = W^{n+1} \text{ and } t \in [t_n, t_{n+1}],\\
&\frac{\partial \tilde{w}}{\partial t} = -\frac{1}{2}\bar{\kappa}^2\sinh(\tilde{w}) \text{ with } \tilde{W}^n = V^{n+1} \text{ and } t \in [t_n, t_{n+1}].
\end{aligned} \tag{23}$$

Then we have $U^{n+1} = \tilde{W}^{n+1}$. As above, we use an analytical integration in the first and the last stage of this scheme. Symbolically, we have $W^{n+1} = F(W^n, \frac{\Delta t}{2})$ and $\tilde{W}^{n+1} = F(\tilde{W}^n, \frac{\Delta t}{2})$, where F is defined as in Eq. (17). The diffusion Eq. (23) is then solved by a LOD scheme. In particular, we first decompose the diffusion Eq. (23), in x, y, and z direction as three equations. Then, we discretize each equation using the Crank-Nicolson method as

$$\begin{aligned}
&\left(1 - \frac{\Delta t}{2}\delta_x^2\right)v_{i,j,k}^* = \left(1 + \frac{\Delta t}{2}\delta_x^2\right)v_{i,j,k}^n,\\
&\left(1 - \frac{\Delta t}{2}\delta_y^2\right)v_{i,j,k}^{**} = \left(1 + \frac{\Delta t}{2}\delta_y^2\right)v_{i,j,k}^*,\\
&\left(1 - \frac{\Delta t}{2}\delta_z^2\right)v_{i,j,k}^{n+1} = \left(1 + \frac{\Delta t}{2}\delta_z^2\right)v_{i,j,k}^{**}.
\end{aligned} \tag{24}$$

2.4.3. Implicit-Euler method with Locally One Dimensional Scheme(LODIE)

By changing the temporal discretization to the implicit Euler method, we propose another LOD method. For this scheme, the TRPBE (11) will be solved by the first-order time splitting in two stages as in the ADIDR scheme. In the first stage, Eq. (15) will be solved analytically by the function $W^{n+1} = F(W^n, \Delta t)$. For the diffusion Eq. (16), we first decompose it into x, y and z directions, and then discretize each equation by the implicit Euler method.

$$\begin{aligned}
&(1 - \Delta t\delta_x^2)v_{i,j,k}^* = v_{i,j,k}^n,\\
&(1 - \Delta t\delta_y^2)v_{i,j,k}^{**} = v_{i,j,k}^*,\\
&(1 - \Delta t\delta_z^2)v_{i,j,k}^{n+1} = v_{i,j,k}^{**}.
\end{aligned} \tag{25}$$

We note that LOD methods similar to (24) and (25) are found to be unconditionally stable in solving the un-regularized PBE [43]. Nevertheless, they are less accurate than the ADI method.

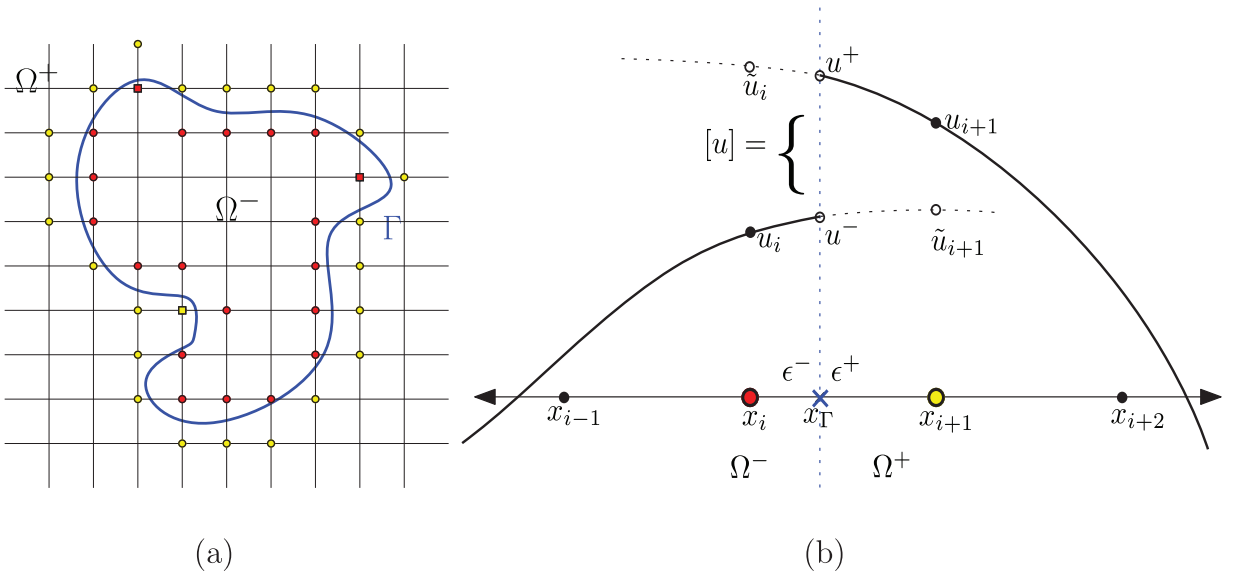

Fig. 2. (a) Domain and grid for the PBE. Irregular grid points are marked as circles, while corner points are shown as squares. (b) GFM in 1D. Here x_i and x_{i+1} are irregular points.

2.5. Spatial discretization: a modified ghost fluid method

For spatial discretization, the central difference approximations will be carried out for points away from the interface Γ. At the points adjacent to the interface Γ, the central difference operators δ_x^2, δ_y^2, and δ_z^2 have to be modified. In particular, an irregular point is defined as a grid node where at least one of its adjacent points is on the other side of the interface Γ. To recover the accuracy of central differences at irregular points, potential and flux jump conditions (12) and (13) should be enforced in the discretization. This gives rise to embedded finite difference formula of δ_x^2, δ_y^2 and δ_z^2 at irregular points. In this paper, a new ghost fluid method (GFM) will be formulated for the TRPBE using fictitious points (or ghost points).

The GFM is a sharp interface technique introduced in [11] to treat the two-face contact discontinuities in the Euler equations. It extends values across the interface into an artificial fluid (ghost fluid), inducing the jump conditions at the interface. The GFM scheme was later extended in [30] to solve elliptic equations with discontinuous coefficients. In the present study, a modification is further introduced in handling tangential jumps, which results in an even simpler sharp interface algorithm for elliptic interface problems. In particular, our modified GFM allows a simple implementation in handling geometrically complicated molecule surfaces of 3D proteins. Comparing with other finite difference interface algorithms for elliptic PDEs, one distinct advantage of the GFM including the present modified GFM is that the jump corrected finite difference discretization produces a symmetry structure in matrix form. This turns out to be crucial in maintaining the stability in the operator splitting solution of the TRPBE. The trade-off is that the GFM method formally has a first order of accuracy in space. But the numerical convergent rate is typically a little bit higher, and hence is sufficiently accurate for most biomolecular simulations.

2.5.1. The standard GFM in one dimension

In the proposed operator splitting methods including ADIDR, LODIE, and LODCN, the TRPBE in 3D is decomposed into 1D along *x, y* and *z* direction alternatively. To keep the computational efficiency of these methods, we need to correct the finite difference operators δ_x^2, δ_y^2 and δ_z^2 individually by enforcing jump conditions. Without the loss of generality, let us consider the correction of δ_x^2 in this subsection. In 1D, the proposed GFM is the same as the standard GFM [30]. Moreover, following ref. [26], we will present the GFM in 1D with the aid of fictitious points.

We consider the domain $\Omega = [a, b]$ to be a finite closed interval on the x-axis with the subdomains $\Omega^- = [a, x_\Gamma)$ and $\Omega^+ = (x_\Gamma, b]$. This interval is then discretized by a sequence of equally spaced grids, $a = x_0 < x_1 < \ldots < x_n = b$ with uniform mesh size h. Point interface Γ is denoted by x_Γ and positions between x_i and x_{i+1} for some i as shown in Fig. 2(b). We also consider u as a 1D function, which satisfies the PBE as $u = u_{in}$ in Ω^- with $u = u_{out}$ in Ω^+. Moreover, let us assume the 1D jump conditions are given as

$$[u]_\Gamma = A, \quad \left[\epsilon \frac{\partial u}{\partial x}\right]_\Gamma = B. \tag{26}$$

At each node, the function value is denoted as $u_i = u(x_i)$. Referring to the Fig. 2(b), two fictitious values are also assumed at two irregular points x_i and x_{i+1}, i.e., $\tilde{u}_i$ and $\tilde{u}_{i+1}$. These fictitious values can be regarded as smooth extension of function values from the other side of interface. Using finite difference, the one-sided limiting values in (26) can be approximated

[26] by

$$u^+ = \tilde{u}_i(1-\lambda) + u_{i+1}\lambda + O(h^2), \qquad \frac{\partial u^+}{\partial x} = \frac{u_{i+1} - \tilde{u}_i}{h} + O(h),$$
$$u^- = u_i(1-\lambda) + \tilde{u}_{i+1}\lambda + O(h^2), \qquad \frac{\partial u^-}{\partial x} = \frac{\tilde{u}_{i+1} - u_i}{h} + O(h),$$

where $h = x_{i+1} - x_i$ and $\lambda = \frac{x_\Gamma - x_i}{h}$. So the jump conditions can be discretized as:

$$[u] = (\tilde{u}_i - u_i)(1-\lambda) + (u_{i+1} - \tilde{u}_{i+1})\lambda + O(h^2),$$
$$\left[\epsilon \frac{\partial u}{\partial x}\right] = \frac{1}{h}\left(\epsilon^+ u_{i+1} - \epsilon^+ \tilde{u}_i - \epsilon^- \tilde{u}_{i+1} + \epsilon^- u_i\right) + O(h).$$

These two discretized jump conditions can be solved to express the fictitious values $\tilde{u}_i$ and $\tilde{u}_{i+1}$ as a linear combination of two function values u_i and u_{i+1}, as well as two known jump values A and B.

Next, the fictitious values can be substituted in the central difference approximation of $u_{xx} = \frac{\partial^2 u}{\partial x^2}$ at x_i and x_{i+1}:

$$u_{xx}(x_i) = \frac{\epsilon^-}{h^2}(u_{i-1} - 2u_i + \tilde{u}_{i+1}) + O(h^2),$$
$$u_{xx}(x_{i+1}) = \frac{\epsilon^+}{h^2}(\tilde{u}_i - 2u_{i+1} + u_{i+2}) + O(h^2).$$

This results in the corrected finite difference formula

$$u_{xx}(x_i) = \frac{1}{h^2}(a_1 u_{i-1} + b_1 u_i + c_1 u_{i+1}) + \frac{\epsilon^-}{h^2}(e_1 A + f_1 B) + O(1),$$
$$u_{xx}(x_{i+1}) = \frac{1}{h^2}(a_2 u_i - b_2 u_{i+1} + c_2 u_{i+2}) + \frac{\epsilon^+}{h^2}(e_2 A + f_2 B) + O(1),$$

where $d = \epsilon^+\lambda + \epsilon^-(1-\lambda)$, $a_1 = \epsilon^-$, $b_1 = -\epsilon^-\left(1 + \frac{\epsilon^+}{d}\right)$, $c_1 = \frac{\epsilon^-\epsilon^+}{d}$, $e_1 = \frac{\epsilon^-}{d}$, $f_1 = \frac{h\lambda}{d}$, $a_2 = \frac{\epsilon^+\epsilon^-}{d}$, $b_2 = -\epsilon^+\left(1 + \frac{\epsilon^-}{d}\right)$, $c_2 = \epsilon^+$, $e_2 = -\frac{\epsilon^+}{d}$, and $f_2 = \frac{h(1-\lambda)}{d}$.

By correcting the discrete operator $\delta_x^2(u_i)$ and $\delta_x^2(u_{i+1})$ correspondingly, at irregular points x_i and x_{i+1}, the GFM produces an $O(1)$ truncation error locally. At all other points, the second order central difference will be applied and the GFM method will yield the global error of $O(h)$ [7]. We note that the present derivation based on fictitious values gives the same formulas as in the standard GFM [30] in 1D. We also observe that the first term in corrected discrete operators is in terms of the original points and the second term is known. Only the coefficients in the first terms contribute to the finite difference operator matrix. Therefore, the 1D GFM finite difference matrix is tridiagonal, diagonally dominant (as $|b_1| - a_1 - c_1 = 0$ and $|b_2| - a_2 - c_2 = 0$), and symmetric (as $a_2 = c_1$).

2.5.2. A modified GFM in two dimensions

We next propose a new GFM in 2D. Because the PBE under our concern is a 3D model, the proposed 2D GFM will not be numerically examined in this paper. Nevertheless, it is natural to present the modified GFM first in 2D first - one can easily see the difference between the standard and modified GFM methods in 2D.

For a 2D elliptic interface problem, function and flux jumps are assumed to be

$$[u]_\Gamma = g, \quad \left[\epsilon \frac{\partial u}{\partial n}\right]_\Gamma = \epsilon^- \frac{\partial g}{\partial n}, \tag{27}$$

for a given function $g(x, y)$, which is smooth near the interface Γ. These conditions mimic the TRPBE jump conditions (12) and (13) in 2D. We note that with the jump conditions (27), $\left[\epsilon \frac{\partial u}{\partial x}\right]$ and $\left[\epsilon \frac{\partial u}{\partial y}\right]$ are still unknown, so that the 1D GFM method discussed above cannot be directly applied. Simple approximations were suggested in the GFM [30] to generate Cartesian jump conditions. To show the details, we express $\left[\epsilon \frac{\partial u}{\partial x}\right]$ and $\left[\epsilon \frac{\partial u}{\partial y}\right]$ in terms of $\left[\epsilon \frac{\partial u}{\partial n}\right]$ and $\left[\epsilon \frac{\partial u}{\partial \tau}\right]$

$$\left[\epsilon \frac{\partial u}{\partial x}\right] = \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left[\epsilon \frac{\partial u}{\partial \tau}\right],$$
$$\left[\epsilon \frac{\partial u}{\partial y}\right] = \sin\theta \left[\epsilon \frac{\partial u}{\partial n}\right] - \cos\theta \left[\epsilon \frac{\partial u}{\partial \tau}\right].$$

where θ is the angle between the normal direction and positive x direction, as shown in Figure 3 (a). Here τ is the tangential direction.

To simplify the jump-conditions, Liu, Fedkiw and Kang [30] smeared out the tangential derivative by considering $\left[\epsilon \frac{\partial u}{\partial \tau}\right] = 0$ to obtain the following approximations,

$$\left[\epsilon \frac{\partial u}{\partial x}\right] \approx \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right], \quad \left[\epsilon \frac{\partial u}{\partial y}\right] \approx \sin\theta \left[\epsilon \frac{\partial u}{\partial n}\right],$$

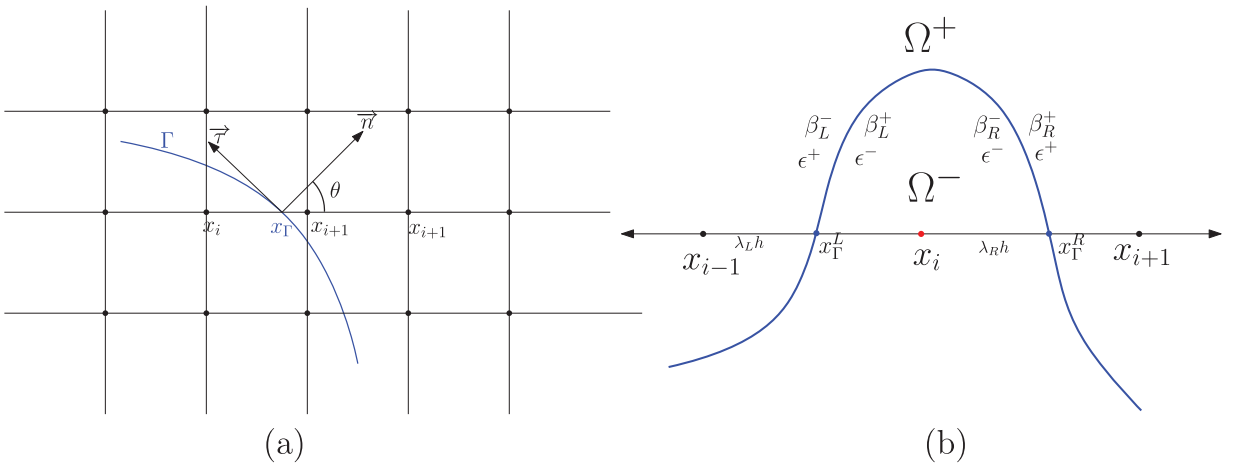

Fig. 3. (a): 2D GFM with the interface Γ and the normal direction $\vec{n}$; (b): 2D representation for a corner point treatment for the point x_i.

which involve an $O(1)$ approximation error. Using such approximations, the 1D GFM discretization can be applied to correct finite difference operators δ_x^2 and δ_y^2.

In this work, taking advantage of the form of jump conditions (27), we propose another GFM method. The standard GFM [30] still requires the determination of the normal direction or θ in 2D. However, for the PBE, the molecular surface is complicated for real proteins. It is desired to have a numerical method that requires less surface information. To this end, let us consider the following identities for $\left[\epsilon \frac{\partial u}{\partial x}\right]$

$$\begin{aligned}
\left[\epsilon \frac{\partial u}{\partial x}\right] &= \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left[\epsilon \frac{\partial u}{\partial \tau}\right] \\
&= \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left(\epsilon^+ \frac{\partial u^+}{\partial \tau} - \epsilon^- \frac{\partial u^-}{\partial \tau}\right) \\
&= \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left(\epsilon^+ \frac{\partial u^+}{\partial \tau} - \epsilon^- \frac{\partial u^+}{\partial \tau} + \epsilon^- \frac{\partial u^+}{\partial \tau} - \epsilon^- \frac{\partial u^-}{\partial \tau}\right) \\
&= \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left(\epsilon^- \left(\frac{\partial u^+}{\partial \tau} - \frac{\partial u^-}{\partial \tau}\right) + (\epsilon^+ - \epsilon^-) \frac{\partial u^+}{\partial \tau}\right) \\
&= \cos\theta \left[\epsilon \frac{\partial u}{\partial n}\right] + \sin\theta \left(\epsilon^- \left[\frac{\partial u}{\partial \tau}\right] + (\epsilon^+ - \epsilon^-) \frac{\partial u^+}{\partial \tau}\right).
\end{aligned}$$

The flux jump is given in (27) as $\left[\epsilon \frac{\partial u}{\partial n}\right] = \epsilon^- \frac{\partial g}{\partial n}$. Moreover, since $[u] = g$, the tangential jump can be calculated as $\left[\frac{\partial u}{\partial \tau}\right] = \frac{\partial g}{\partial \tau}$. By plugging these jump values to the preceding equation, we have

$$\left[\epsilon \frac{\partial u}{\partial x}\right] = \epsilon^- \frac{\partial g}{\partial n} \cos\theta + \epsilon^- \frac{\partial g}{\partial \tau} \sin\theta + (\epsilon^+ - \epsilon^-) \frac{\partial u^+}{\partial \tau} \sin\theta. \tag{28}$$

By dropping the third term in (28), which is essentially equivalent to assume $\frac{\partial u^+}{\partial \tau} = 0$, one derives another approximation

$$\left[\epsilon \frac{\partial u}{\partial x}\right] \approx \epsilon^- \left(\frac{\partial g}{\partial n} \cos\theta + \frac{\partial g}{\partial \tau} \sin\theta\right) = \epsilon^- \frac{\partial g}{\partial x}.$$

Similarly, it can be shown that

$$\left[\epsilon \frac{\partial u}{\partial y}\right] \approx \epsilon^- \frac{\partial g}{\partial y}.$$

Like the original GFM, our new GFM also omits some tangential information. Moreover, for both GFM methods, the approximated Cartesian jumps actually satisfy the normal flux jump exactly, although different approximations are employed. For both methods, the local truncation error is $O(1)$, but the global truncation error will be $O(h)$ or first order. Nevertheless, in our new method, the normal direction of the complicated molecular surface is not required, so that the implementation of the modified GFM is much simpler than the standard GFM.

2.5.3. A modified GFM in three dimensions

In this subsection, we will only present the modified GFM in 3D. Consider that the interface Γ intersects the grid line in the x direction at a point (x_Γ, y_j, z_k) which is located between (x_i, y_j, z_k) and (x_{i+1}, y_j, z_k). We therefore have two irregular grid points, (i, j, k) and $(i+1, j, k)$. The fictitious values $\tilde{u}_{i,j,k}$ and $\tilde{u}_{i+1,j,k}$ are to be determined by the 1D GFM. To this end, an approximated Cartesian jump $\left[\epsilon \frac{\partial u}{\partial x}\right]$ is needed. A 3D tensor production decomposition will be carried to rewrite the 3D jump conditions (12) and (13) into 1D ones.

To use the jump condition which is defined in the normal direction of the interface point, it is convenient to introduce a local coordinates (ξ, η, ζ) such that ξ is along the normal direction and η is in the xy plane. Denote θ and ψ as the azimuth and zenith angles with respect to the normal direction ξ. We then have

$$\left[\epsilon \frac{\partial u}{\partial x}\right] = \sin\psi\cos\theta\left[\epsilon \frac{\partial u}{\partial \xi}\right] - \sin\theta\left[\epsilon \frac{\partial u}{\partial \eta}\right] - \cos\psi\cos\theta\left[\epsilon \frac{\partial u}{\partial \zeta}\right]. \tag{29}$$

Note that $\left[\epsilon \frac{\partial u}{\partial \xi}\right] = \epsilon^- \frac{\partial G}{\partial n}$ by (13), since ξ is along the normal direction. Now for the η direction flux jump, we have the following identities

$$\begin{aligned}
\left[\epsilon \frac{\partial u}{\partial \eta}\right] &= \epsilon^+ \frac{\partial u^+}{\partial \eta} - \epsilon^- \frac{\partial u^-}{\partial \eta} \\
&= \epsilon^+ \frac{\partial u^+}{\partial \eta} - \epsilon^- \frac{\partial u^+}{\partial \eta} + \epsilon^- \frac{\partial u^+}{\partial \eta} - \epsilon^- \frac{\partial u^-}{\partial \eta} \\
&= \epsilon^- \left(\frac{\partial u^+}{\partial \eta} - \frac{\partial u^-}{\partial \eta}\right) + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \eta} \\
&= \epsilon^- \frac{\partial G}{\partial \eta} + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \eta},
\end{aligned} \tag{30}$$

where we have applied the tangential jump

$$\left(\frac{\partial u^+}{\partial \eta} - \frac{\partial u^-}{\partial \eta}\right) = \left[\frac{\partial u}{\partial \eta}\right] = \frac{\partial}{\partial \eta}[u] = \frac{\partial G}{\partial \eta}.$$

Similarly along the ζ axis, one can derive

$$\left[\epsilon \frac{\partial u}{\partial \zeta}\right] = \epsilon^- \frac{\partial G}{\partial \zeta} + (\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \zeta}. \tag{31}$$

Then from equations (29), (30) and (31) we have,

$$\begin{aligned}
\left[\epsilon \frac{\partial u}{\partial x}\right] &= \epsilon^- \left(\sin\psi\cos\theta \frac{\partial G}{\partial \xi} - \sin\theta \frac{\partial G}{\partial \eta} - \cos\psi\cos\theta \frac{\partial G}{\partial \zeta}\right) \\
&\quad - \sin\theta(\epsilon^+ - \epsilon^-)\frac{\partial u}{\partial \eta}^+ - \cos\psi\cos\theta(\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \zeta} \\
&= \epsilon^- \frac{\partial G}{\partial x} - \sin\theta(\epsilon^+ - \epsilon^-)\frac{\partial u}{\partial \eta}^+ - \cos\psi\cos\theta(\epsilon^+ - \epsilon^-)\frac{\partial u^+}{\partial \zeta}.
\end{aligned}$$

By dropping the last two terms of the preceding equation or assuming the vanishing of $\frac{\partial u}{\partial \eta}^+$ and $\frac{\partial u^+}{\partial \zeta}$, we end up with a simple approximation

$$\left[\epsilon \frac{\partial u}{\partial x}\right] \approx \epsilon^- \frac{\partial G}{\partial x} \tag{32}$$

Similar derivations along y axis and z axis give rises to

$$\left[\epsilon \frac{\partial u}{\partial y}\right] \approx \epsilon^- \frac{\partial G}{\partial y}, \quad \text{and} \quad \left[\epsilon \frac{\partial u}{\partial z}\right] \approx \epsilon^- \frac{\partial G}{\partial z}. \tag{33}$$

We note that the approximated Cartesian jumps (32) and (33) will satisfy the normal flux jump condition $\left[\epsilon \frac{\partial u}{\partial n}\right] = \epsilon^- \frac{\partial G}{\partial n}$ exactly. So the proposed modified GFM will preserve normal jumps in computations.

2.5.4. Corner point treatment for the modified GFM method

Because the molecular surface is geometrically complex in 3D, corner points are frequently encountered in protein studies. Fortunately, the corner point treatment is easy in the proposed GFM method, because it is formulated in 1D and involves only the nearest grid nodes.

Consider a corner point when the interface crosses the grid line for the same axis twice around x_i where x_{i-1} and x_{i+1} are on the other side of the interface as shown in Figure 3 (b). In our modified GFM, the jump conditions are reduced to

Cartesian directions. Hence, the numerical approximation in one Cartesian direction does not interfere with other Cartesian directions. So, the corner points can be treated in 1D.

There are two types of corner point situations. In one type the point x_i is in Ω^- and the other two adjacent points are in Ω^+ (see Figure 3 (b)). For this case we define $G_L = -G$ and $G_R = G$ to derive the following equations for the fictitious points $\tilde{u}_{i-1}$ and $\tilde{u}_{i+1}$:

$$\tilde{u}_{i-1} = A_1 u_{i-1} + B_1 u_i + C_1 G_L - D_1 \epsilon^- \frac{\partial G_L}{\partial x} + O(h^2),$$

$$\tilde{u}_{i+1} = A_2 u_{i+1} + B_2 u_i - C_2 G_R - D_2 \epsilon^- \frac{\partial G_R}{\partial x} + O(h^2),$$

where $A_1 = \frac{\beta_L^-}{F_1}, B_1 = \frac{\lambda_L(\beta_L^+ - \beta_L^-)}{F_1}, C_1 = \frac{\beta_L^-}{F_1}, D_1 = \frac{h\lambda_L}{F_1}$,
$A_2 = \frac{\beta_R^+}{F_2}, B_2 = \frac{(1-\lambda_R)(\beta_R^- - \beta_R^+)}{F_2}, C_2 = \frac{\beta_R^+}{F_2}, D_2 = \frac{h(1-\lambda_R)}{F_2}$,
$\beta_L^- = \epsilon^+, \beta_L^+ = \epsilon^-, \beta_R^- = \epsilon^-, \beta_R^+ = \epsilon^+$,
$\lambda_L = \frac{x_\Gamma^L - x_{i-1}}{h}, \lambda_R = \frac{x_\Gamma^R - x_i}{h}$,
$F_1 = \beta_L^+ \lambda_L + \beta_L^- (1 - \lambda_L)$, and $F_2 = \beta_R^+ \lambda_R + \beta_R^- (1 - \lambda_R)$.

Correcting central differences by using these fictitious values, we obtain the finite difference operator δ_x^2 at corner points. The other type of corner points can be similarly treated. We observe that the symmetric and tridiagonal matrix structure is still preserved, while the local approximation error is still $O(1)$ at corner points.

2.5.5. Comparison remarks

It is interesting to compare the new GFM method with the existing finite difference interface algorithms, such as high order GFM [2,8,17] and MIB [49,50] methods. The MIB method is the first second-order-accurate scheme for solving the PBE in the literature [4,13,45], and it can handle complicated interface geometry, including sharp corners and edges. However, these high order interface treatments couple grid nodes in all Cartesian directions simultaneously, so that they cannot be applied in the ADI/LOD framework. The mADI method developed for the heat equation [27,42,48] also produces second-order interface treatments. Nevertheless, the mADI method is semi-implicit in time integration and may have a stability constraint for complex interfaces in 3D. Albeit being first order in space, the proposed GFM is much easier to program compared to the MIB and mADI methods. By neglecting tangential jumps, the GFM allows a simple 1D interface treatment to correct central difference operators in the ADI/LOD framework and maintain the fully-implicit nature in time integration. Moreover, since the Green's function $G(x, y, z)$ is analytically available, the calculations of approximated Cartesian jumps (32) and (33) are very simple. One does not need to compute auxiliary points in other directions as in the MIB method [4,13,45,49,50], nor the non-orthogonal tangential jumps as in the mADI method [27,42,48].

The modified GFM becomes even simpler than the standard GFM [30], because one does not need to compute the normal direction for the complicated molecular surface. This has significantly simplified the implementation because the handling of molecular surface in finite difference method [4] or the mesh generation of finite element method [9] is known to be the bottleneck in the numerical solution of the PBE. The modified GFM reduces the required amount of information about the molecular surface, but preserves the normal flux jumps and attains a first-order convergence in space like the original GFM.

In the present study, the molecular surface is generated by the MSMS software [34], which is the most widely used surface definition in the literature. In the MSMS output, the molecular surface is represented through a triangular mesh, including the coordinate and normal direction of each vertex. All that we need in the modified GFM are locations of the points where the triangularized surface is intersecting with the Cartesian grid lines. A Lagrangian-to-Eulerian algorithm similar to the one presented in [45] is employed to calculate where one Cartesian grid line is cut through a particular triangle. Numerically, this is a highly nontrivial job, because the molecular surface is known to have geometrical singularities [45]. Without calculating the normal directions, the present Lagrangian-to-Eulerian algorithm becomes simpler than that of [45].

2.6. Solvation free energy from Electrostatic potential

The solvation free energy can be defined as the energy released when the solute in free space is dissolved in solvent and it can be expressed in terms of the electrostatic free energy ΔG_{ele} and the coulomb energy E_{cou} as

$$\Delta G_{\text{ele}} = E_{\text{cou}} + E_{\text{sol}} = \sum_{i=1}^{N_c} \sum_{j=1, j\neq i}^{N_c} \frac{q_i q_j}{\epsilon^- d_{i,j}} + E_{\text{sol}} \tag{34}$$

where q_i and q_j are the charges at the center of the atoms and $d_{i,j}$'s are the distance between the i-th and j-the atom. Sharp and Honig in [36] described the calculation electrostatic free energy using

$$\Delta G_{\text{ele}} = \int_{\mathbb{R}^3} \left(\phi \rho + \Delta\Pi - \frac{1}{2} \epsilon |\mathbf{E}|^2 \right) d\mathbf{r},$$

where ϕ is the electrostatic potential, ρ is the fixed charge density represented as a smeared surface charge or as a collection of point charges, $\Delta\Pi$ is the excess osmotic pressure of the mobile ion cloud, and $\frac{1}{2}\epsilon|\mathbf{E}|^2$ is the electrostatic stress.

Now for the simplification of the numerical validations of our proposed schemes in the next section we omitted the energy components related to the mobile ion pressure and the electrostatic stress to report the solvation free energy E_{sol} as

$$E_{\text{sol}} = \Delta G_{\text{ele}} - E_{\text{cou}} \approx \sum_{i=1}^{N_c} q_i \phi_{RF}(\mathbf{r}_i), \tag{35}$$

where the react field potential ϕ_{RF} can be calculated as the difference between the regularized potential $\tilde{\phi}$ and Green's function G. Here the contribution of the mobile ion pressure and the electrostatic stress to the whole calculation is really small and computationally more challenging. The readers can refer to [12,16,36] for more details.

3. Numerical Validation

To validate our proposed algorithm we are providing numerical simulations in this section. First we solve the nonlinear PBE for the Kirkwood Sphere problem and compare the numerical results with the analytical solution. Then, we focus on a particular protein to examine the convergence, accuracy, and efficiency of the proposed pseudo-time methods. A set of proteins is next studied to compare with literature results. Finally, we also explore the use of new methods for calculating binding energy and salt effect.

Based on the two-component regularization and pseudo-transient operator splitting, three new algorithms are developed in this work, i.e., GFM-ADIDR, GFM-LODCN, and GFM-LODIE methods. For a comparison, the ADI algorithm introduced in [14] will also be considered, which deals with the original PBE without regularization and has no interface treatment. One can expect that such an ADI method is very inaccurate, due to the poor approximation involved in distributing delta functions at atomic centers to surrounding grid nodes and the lack of correction in finite difference operators. Through numerical experiments, we will show that the proposed interface treatment together with the two-component regularization not only recovers the accuracy, but also improves the stability and efficiency.

3.1. Kirkwood Sphere with analytical solution

We first consider a benchmark problem known as the Kirkwood Sphere, where a charge is located at its center. This problem has the following analytical solution and source term [14]

$$\phi(\mathbf{r}) = \begin{cases} \dfrac{1}{\varepsilon R} - \dfrac{1}{R} + \dfrac{1}{||\mathbf{r}||} & ||\mathbf{r}|| < R, \\ \dfrac{1}{\varepsilon ||\mathbf{r}||} & ||\mathbf{r}|| > R. \end{cases} \quad \rho(\mathbf{r}) = \begin{cases} 4\pi\delta(\mathbf{r}) & ||\mathbf{r}|| < R, \\ \bar{\kappa}^2 \sinh\left(\dfrac{1}{\varepsilon ||\mathbf{r}||}\right) & ||\mathbf{r}|| > R, \end{cases} \tag{36}$$

where $\varepsilon = \epsilon^+/\epsilon^-$, $R = 2$ is the radius of the sphere and $\bar{\kappa} = 1$. We have a unit charge $1e_c$ located at the center of the sphere (0,0,0). The computational domain is set by the boundaries from -7 to 7 in x, y, and z directions. Here the dielectric constants are set as $\epsilon^+ = 80$ outside the sphere and $\epsilon^- = 1$ inside the sphere. The ionic strength is $I_s = 0.01$ for all the tests discussed in this section for the Kirkwood Sphere problem. It can be shown that the analytical solution with the source term defined in (36) will satisfy the nonlinear PBE in (1) together with the jump condition defined in (3). Both of the major challenges like the singularity and the non-smoothness are present in this benchmark problem. To use this benchmark problem to test the stability and the convergence in time and space, we computed the L_2 error and L_∞ error using the following measures:

$$L_\infty = \max_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|, \quad L_2 = \sqrt{\frac{\sum_{i,j,k} |\phi_{\text{exact}} - \phi_{\text{num}}|^2}{N}},$$

where ϕ_{true} is the analytical solution and ϕ_{num} is the numerical solution representing the electrostatic potential for the nonlinear PBE. For the L_2 error we have used $N = N_x \times N_y \times N_z$ as the total number of unknowns on the grid points.

Stability Test: To numerically test the stability, one can solve the time-dependent PBE by running the time loops for a lot of iterations to check if the computation blows-up or not. In this study, we chose to run the whole process for 10^4 iterations for different choices of pseudo-time step size Δt for each one of our proposed methods along with the previous ADI method [14]. Our goal here is to check if these numerical methods are unconditionally stable or not, and if they are conditionally stable, then what is the condition on the critical choice for Δt. We focused on the variation of Δt and used two grids with $h = 0.5$ and $h = 0.25$ to avoid the difficulty due to the larger grid spacing. We have found all three of our methods to be stable for all tested $\Delta t \in [0.001, 5]$ for this benchmark problem. To illustrate this we considered the sampling for Δt such as $\Delta t \in t_{set} = \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$ and the stopping time $T_{end} = 10^4 \cdot \Delta t$, so that enough accumulations of numerical errors are experienced.

We observe in Figure 4 (b), (c) and (d) that both L_2 and L_∞ errors to be bounded for all $\Delta t \in t_{set}$. For larger values of Δt such as $\Delta t = 5$, the numerical errors might feel meaningless, but as long as these errors remain bounded, this demonstrates the stability of the underlying time integration. In comparison the ADI method [14] in Figure 4 (a), blows-up for $\Delta t > 0.2$ in both cases. Also, the error lines are almost flat (constant) for $\Delta t < 0.02$ in Figure 4 (a), indicating that the L_2 and L_∞ error

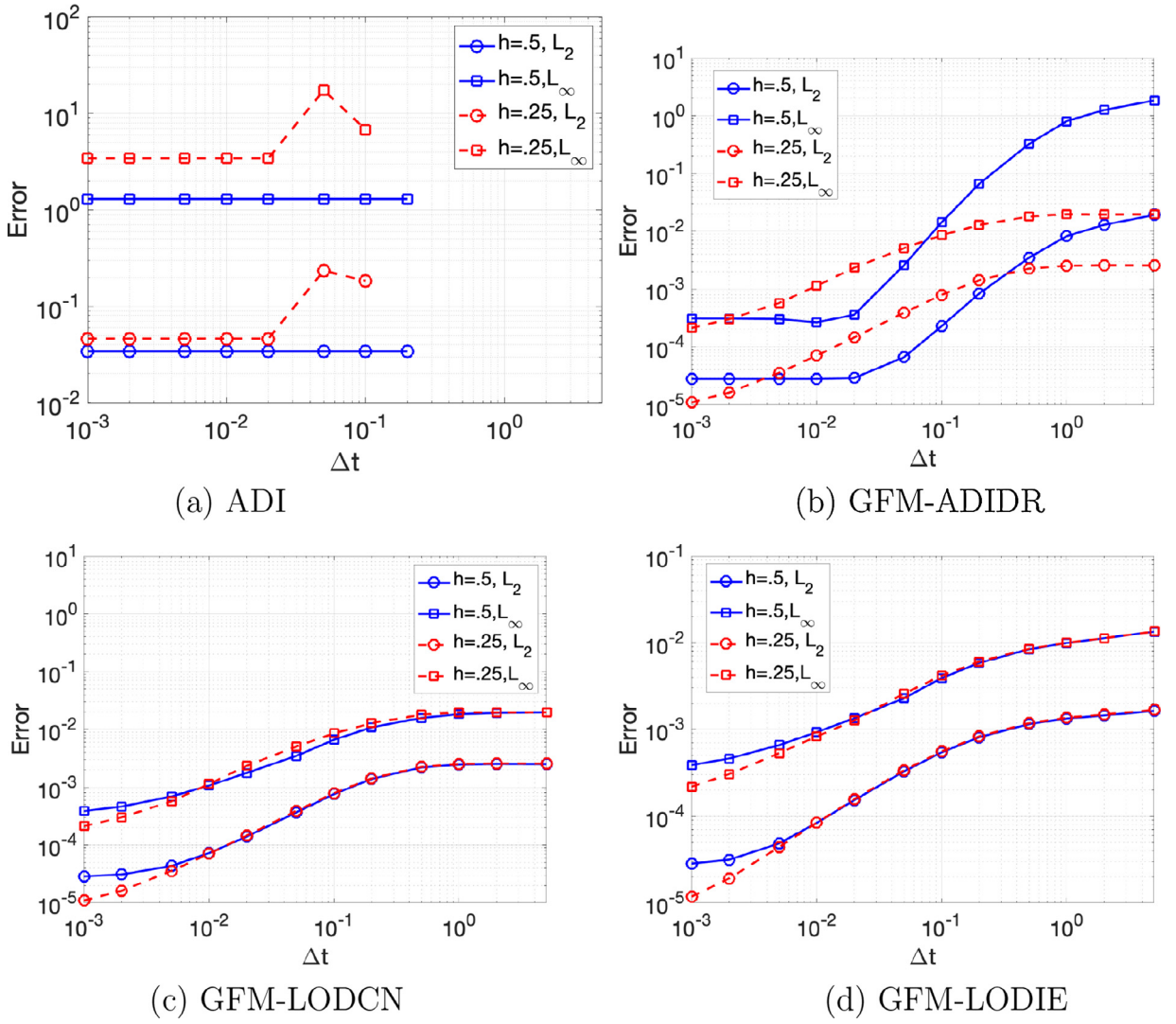

Fig. 4. Stability test for the Kirkwood Sphere with $T_{end} = 10^4 \cdot \Delta t$. In subplot (a), the ADI method is unstable for large Δt, so that only part of curves are depicted. The other three methods are unconditionally stable for one atom model.

are polluted and do not decrease when Δt becomes smaller like the other methods do. This observation again emphasizes the importance of the regularization and GFM interface treatment.

Spatial Convergence: In this study, we investigated the order of accuracy for the spatial convergence in terms of L_2 and L_∞ errors in Table 1 for the Kirkwood Sphere. A sufficiently small pseudo-time step size $\Delta t = 0.001$ is used, while the grid spacing h is reduced from 2 to 1/8. Within this range of h, we have noticed the accuracy of the GFM-ADIDR to be nearly second-order while it gradually decreases for the GFM-LODCN and GFM-LODIE. In this experiment, the stopping time has been set to $T_{end} = 10$ and found to be long enough to produce acceptable results. We have also computed the solvation free energy E_{sol} for the nonlinear PBE in (1) with the source term defined in (2). The solvation free energy for this setup can also be computed analytically as -81.97820845. For all three of our proposed methods, the solvation free energies E_{sol} are found to be very close to this analytical value as reported in Table 1. In comparison, the accuracy of the ADI method decreases instead of increasing as the grid spacing becomes finer. The major reason behind this is that the atom center or the origin is one of the grid points. Due to the singularity at this point, the actual difference between the numerical solution and the analytical solution goes to infinity, since the term $\frac{1}{||\mathbf{r}||}$ in the analytical solution becomes infinity at the center of the sphere. So we excluded this point to calculate our L_2 and L_∞ errors showed in Table 1. However, still, it affects the error of neighboring points and ultimately dominates the errors at all other points.

Temporal Convergence: Finally, we investigate the temporal convergence of all three of our proposed methods for the Kirkwood Sphere in Figure 5 along with the ADI method. A small $h = 0.125$ is employed in all cases. We have chosen the

Table 1
Spatial convergence test for the Kirkwood Sphere with $\Delta t = 0.001$ and $T_{end} = 10$. The analytical solvation free energy is $E_{sol} = -81.97820845$.

h	L_2	Order	L_∞	Order	E_{sol}
			ADI		
2	6.45E-03	N/A	3.82E-02	N/A	-92.699927
1	4.88E-03	0.40	7.61E-02	-0.99	-83.683388
1/2	3.43E-02	-2.81	1.30E+00	-4.09	-85.921222
1/4	4.63E-02	-0.43	3.44E+00	-1.41	-83.279725
1/8	5.11E-02	-0.14	7.49E+00	-1.12	-82.680633
			GFM-ADIDR		
2	3.14E-04	N/A	1.82E-03	N/A	-81.742795
1	1.18E-04	1.41	8.28E-04	1.13	-82.132181
1/2	2.79E-05	2.08	3.10E-04	1.42	-82.063724
1/4	8.51E-06	1.71	1.23E-04	1.34	-82.051117
1/8	1.49E-06	2.51	4.47E-05	1.46	-82.046462
			GFM-LODCN		
2	3.14E-04	N/A	1.82E-03	N/A	-81.742788
1	1.18E-04	1.41	8.89E-04	1.03	-82.132148
1/2	2.87E-05	2.04	3.87E-04	1.20	-82.063684
1/4	1.10E-05	1.39	2.13E-04	0.86	-82.051064
1/8	7.16E-06	0.62	1.41E-04	0.60	-82.046402
			GFM-LODIE		
2	3.16E-04	N/A	1.84E-03	N/A	-81.738399
1	1.17E-04	1.43	8.85E-04	1.06	-82.123319
1/2	2.84E-05	2.05	3.89E-04	1.19	-82.055388
1/4	1.18E-05	1.27	2.18E-04	0.83	-82.043011
1/8	9.40E-06	0.33	1.48E-04	0.56	-82.046402

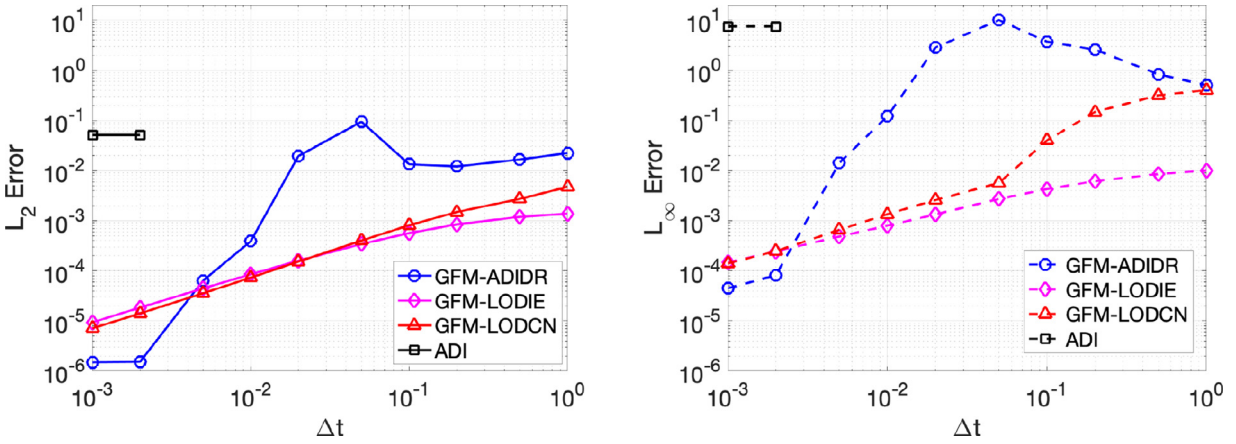

Fig. 5. Temporal convergence test for the Kirkwood Sphere with $h = 0.125$ and $T_{end} = 10$. Left: L_2 error; Right: L_∞ error.

fixed stopping time to be $T_{end} = 10$ and $\Delta t \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1\}$. As we observe in Figure 5, the GFM-ADIDR is the most accurate method for the smaller Δt but as Δt increases the GFM-LODCN and GFM-LODIE perform better than the GFM-ADIDR in terms of both the L_2 and L_∞ errors. For a large Δt, the GFM-LODIE method remains the most accurate. On the other hand, the original ADI method [14] is represented by the black lines in Figure 5, which blow-up just after $\Delta t = 0.002$.

In terms of convergence, the GFM-LODCN and GFM-LODIE reduce the error monotonically. However, the GFM-ADIDR is non-convergent initially but produces a rapid convergence when Δt is sufficiently small. The abnormal convergence pattern of the GFM-ADIDR is believed to relate to perturbation errors and instability. As shown in Eq. (22), the ADI time integration is a higher-order perturbation of 3D implicit Euler solution of the heat equation. For large Δt values, such perturbation terms involving mixed derivatives produces a quite large error. In the present study, the perturbation error grows as the number of pseudo-time steps becomes larger. The growth rate is not significant so that the computation is still stable, but can ruin the convergence. Note that the numerical error appears to be small for $\Delta t = 1$, simply because T_{end} is fixed as 10 so that there are only a few steps. On the other hand, when Δt is sufficiently small, the perturbation errors can be effectively damp out by high order terms of Δt, so that convergence shows up. The instability issue of the GFM-ADIDR method becomes evident in protein studies.

Table 2
Stability test for the calculation of the Solvation Energy (*kcal/mol*) of 1cbn for $h = 0.5$, $T_{end} = 10^4 \cdot \Delta t$.

Δt	ADI	GFM-ADIDR	GFM-LODCN	GFM-LODIE
0.001	-459.5742719854	-303.00657886	-303.00154088	-302.80556740
0.002	-458.1104685049	-302.99808443	-302.98279503	-302.61375402
0.005	-452.1116748705	-302.90138868	-302.83729044	-302.03341621
0.01	NaN	-302.85075766	-302.68957418	-301.31096348
0.02	NaN	-302.69227062	-302.33466174	-300.06953533
0.05	NaN	-302.04538862	-301.08386563	-297.04896758
0.1	NaN	-300.84648128	-298.86787256	-293.05472908
0.2	NaN	-298.63624081	-294.81110682	-286.88648904
0.5	NaN	-293.21633440	-284.83429816	-274.64462149
0.7	NaN	NaN	-279.04495504	-269.01031327
1	NaN	NaN	-271.29161057	-262.50478312
2	NaN	NaN	-251.54655749	-248.89416804
5	NaN	NaN	-218.29937773	-230.91124478

3.2. Biological applications

In this section, we focus on exploring the stability and accuracy of GFM-ADIDR, GFM-LODCN, and GFM-LODIE schemes by considering the solvation analysis of proteins. Even though all of our proposed schemes were found to be stable for the one atom case (Kirkwood Sphere) with $\Delta t = [0.001, 5]$, it is of great interest to see if these schemes are stable for real protein systems. We will compare all three methods in details for a particular protein system and try to identify a good choice for the pseudo-time increment Δt and the grid spacing h to get the most accurate result within a reasonable amount of computational time. The selected Δt and h are used later to calculate the solvation free energy of 24 proteins.

3.2.1. Protein Crambin

To validate our proposed schemes with a real protein, we consider a hydrophobic protein Crambin (PDB ID : 1cbn). It is a 46 residue protein homologous to a membrane-active plant toxins [40]. It is found in the seeds of *Crambe abyssinicia* and has local anesthetic activity in a lobster leg axon (J. Marquis [40]). We used the MSMS package to generate the molecular surface for this protein using the crystallographic data recorded at 130K as reported in [40]. For this step, we use the probe radius of 1.4Å and the density of 10 in the MSMS package to calculate the molecular surface.

Without an analytical solution, we will mainly focus on the solvation free energy in this study. After calculating the electrostatic potential ϕ, Eq. (35) is discretized further to calculate the solvation free energy where Q is the trilinear interpolation of charge centers to Cartesian grid nodes. The potential values are obtained by scaling our calculated dimensionless potentials with the constant 0.596163438 corresponding for the room temperature (300K). In all cases, a large stopping time T_{end} is used to ensure that the steady-state solution is reached. For the dielectric constant, we have used $\epsilon^+ = 80$ for water as the solvent and $\epsilon^- = 1$ for the region inside the protein. The ionic strength is set to $I_s = 0.15$.

Stability Analysis: A stability study is first carried out for four algorithms by taking $h = 0.5$ and $T_{end} = 10^4 \cdot \Delta t$. Like the previous study, different Δt values have been computed, and the corresponding solvation free energies are reported in Table 2. It can be seen that the ADI method is still subject to the most severe stability constraint, and fails to converge for $\Delta t \geq 0.01$. Unlike the one atom case, the GFM-ADIDR method becomes conditionally stable for protein studies, even though the critical $\Delta t = 0.5$ is large enough for most pseudo-time simulations. Remarkably, the GFM-LODCN and GFM-LODIE methods are unconditionally stable for real biomolecular simulations.

In order to recommend discretization parameter values for further studies, here we try to identify a value of Δt as large as possible, while without losing too much accuracy for a real protein like 1cbn. Note that in Table 2 for $\Delta t > 0.5$, the GFM-ADIDR diverges while the other two LOD schemes lose a significant amount of accuracy. If we consider the solvation free energy (≈ -302 *kcal/mol*) for $\Delta t = 0.005$ to be accurate enough and compare all other solvation free energies in Table 2, it can be observed that with the increase of Δt all three proposed schemes loose accuracy but at different rates. The GFM-ADIDR scheme is usually more accurate while GFM-LODIE scheme is more robust as it remains stable for larger values of Δt. The performance of the GFM-LODCN is roughly in between the other two schemes in terms of accuracy and stability. To be uniform among all three proposed methods and the other proteins in this paper, we recommend using $\Delta t = 0.05$ and $h = 0.5$ for large scale protein studies.

Even though both ADI algorithms are conditionally stable for the 1cbn study, the stability constraints are different, which implies a huge difference in real applications. For the GFM-ADIDR method, the critical value of Δt is about 0.5. This makes GFM-ADIDR a very practical method, because with such a large Δt, one will not concern about the efficiency for steady-state simulations. In comparison, for the ADI method, this critical value is $\Delta t = 0.005$, which results in unreasonably longer time to reach the steady-state to produce any meaningful results.

In the GFM family, the ADI and LOD methods have different ranges of the application domain. Being unconditionally stable does not mean that the LOD methods are always better. For example, when high precision is needed in some applications, the GFM-ADIDR method could be a better choice. For GFM-ADIDR, the critical Δt value for stability tells us a safe

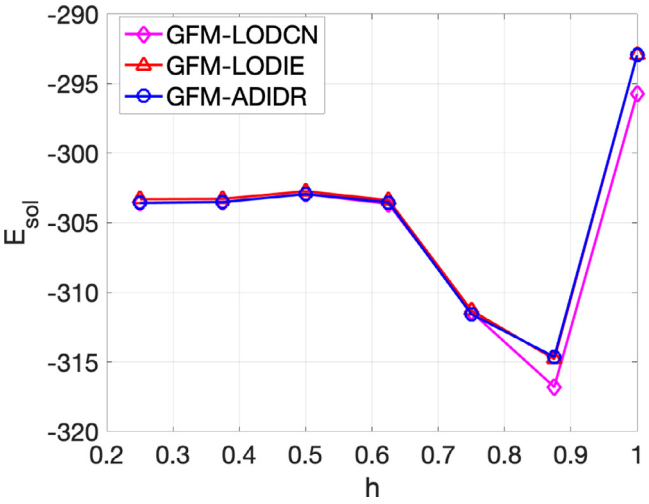

Fig. 6. Spatial convergence for *1cbn* with $\Delta t = 0.001$ and $T_{end} = 10$.

Table 3
Impact of domain size to the solvation free energy of 1*cbn* calculated by the GFM-ADIDR method.

X	Y	Z	E_{sol}
[-10,32]	[-9,29]	[-15,28]	-302.0594
[-13,35]	[-12,32]	[-18,31]	-302.0594
[-16,38]	[-14,35]	[-20,34]	-302.0626
[-19,41]	[-17,38]	[-23,36]	-302.0656

zone. Within this safe zone, the GFM-ADIDR results are reliable. However, the GFM-LOD methods do not provide such protection to prevent inaccurate estimates. On the other hand, in some other applications where efficiency is more important than accuracy, the GFM-LODIE or GFM-LODCN with a large Δt should be employed.

Spatial Convergence: In Figure 6 we investigate how the change of grid spacing h affects the solvation free energy of real proteins like 1*cbn*. A small time increment $\Delta t = 0.001$ is used with the stopping time $T_{end} = 10$. Because the CPU time increases when h becomes smaller, we need to find the value of h as large as possible to save computational efforts. On the other hand, a small enough h shall be used to guarantee accuracy. We can see from Figure 6 that the spatial convergence with respect to h is not monotonic for the GFM method. This is because Cartesian grids are used in our computations so that the molecular surface will cut the Cartesian grid lines in a random manner. Nevertheless, for $h < 0.6$, the solvation free energy does not change that much. Following the convention of this field, we will choose $h = 0.5$ for calculating the solvation free energies. Such a h value has been found to be sufficiently accurate for other proteins.

Steady-state solution: In our simulations, the pseudo-transient solution always approaches a steady-state as $t \to \infty$, although the theoretical justification about the existence of the steady-state solution for the present nonlinear operator is not a trivial job. Here, we illustrate the steady-state convergence of the TRPBE model by numerically checking two different types of initial conditions. For the first type, we just set $\phi^0 = 0$. Then for the second type, we solve a linearized PBE (LPBE) [46] and use such a solution as initial condition $\phi^0 = \phi_{LPBE}$. We take $h = 0.5$ and $\Delta t = 0.05$, and run the iterations until $T_{end} = 10$. As we can see in Figure 7 (a), (b) and (c), all of our proposed methods reach the same steady-state solution for both types of initial conditions. This numerically demonstrated the uniqueness of a steady-state solution. It is also noticeable in Figure 7 (a), (b) and (c), that initializing ϕ^0 from LPBE is better than initializing ϕ^0 as zero since it reaches the steady-state earlier.

In Figure 7(d) we report the calculated solvation free energy difference ΔE^n_{sol} at the n-th pseudo-time step defined as

$$\Delta E^n_{sol}(\phi^n) = |E^n_{sol}(\phi^n) - E^{n-1}_{sol}(\phi^{n-1})|.$$

As we observe in Figure 7(b), this energy difference ΔE^n_{sol} decreases over time for all three new methods, indicating the steady-state convergence. Because of this, the energy difference is frequently used as another criterion for steady-state convergence. For example, one can design the pseudo-time integration to be stopped when $\Delta E^n_{sol}(\phi^n) < 10^{-4}$.

The computational domain is another factor that could influence the steady-state. We have investigated the effect of the size of the computational domain on the calculation of the solvation free energy by the GFM-ADIDR scheme. As we observe in Table 3, changing the length of the cubic domain in the *x, y* and *z* directions does not change the solvation free energy for 1*cbn* significantly. So we set the computational domain as tight as possible, by keeping the whole molecule inside plus a fixed distance from the furthest atom from the coordinate origin.

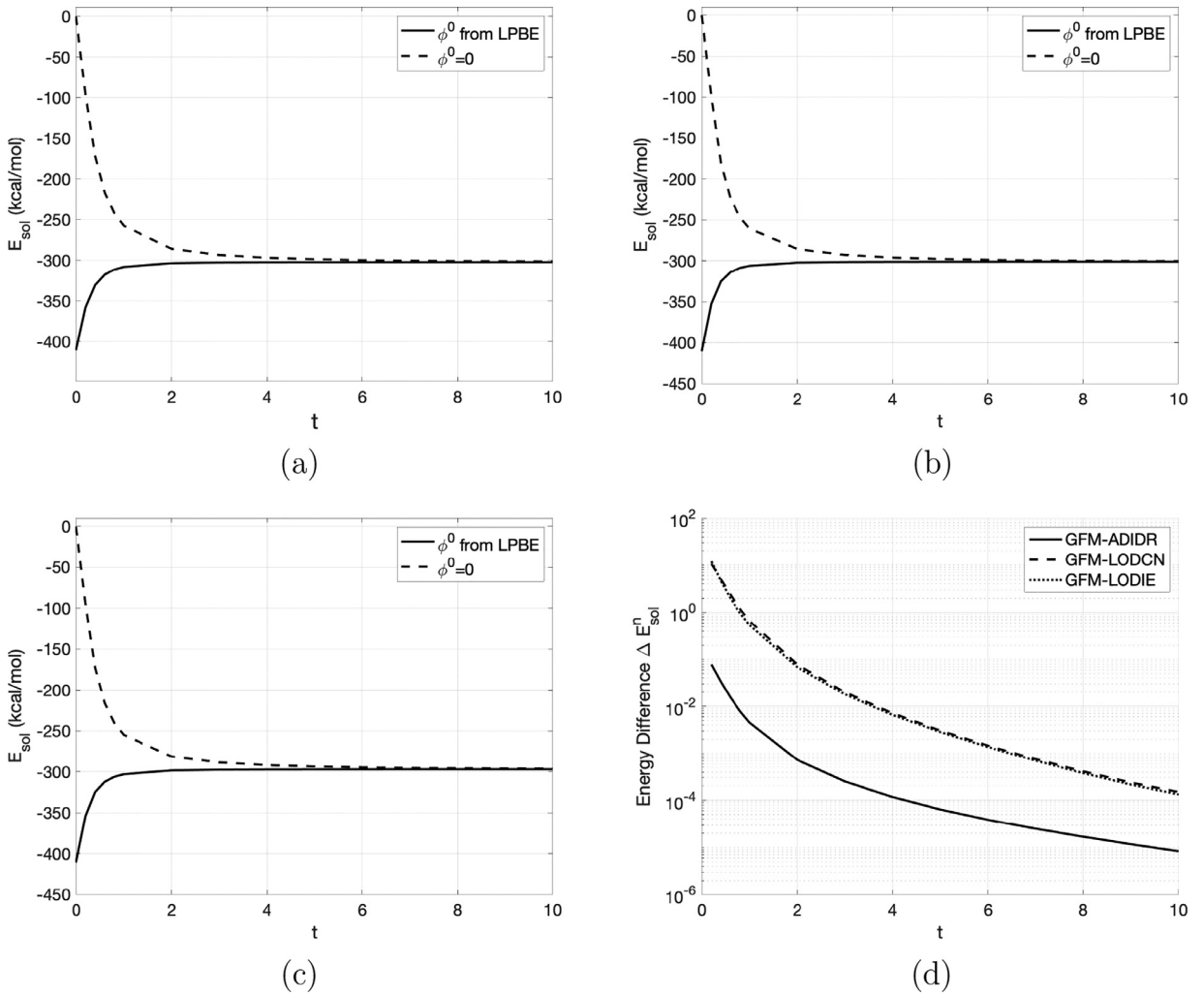

Fig. 7. Steady-state Analysis: The Solvation Energy (*kcal/mol*) of 1*cbn* with $h = 0.5$, $\Delta t = 0.05$ as $t \to \infty$. (a) GFM-ADIDR, (b) GFM-LODCN, (c) GFM-LODIE, (d) Solvation Energy difference $\Delta E^n_{sol}(\phi^n)$.

Electrostatic potential on the molecular surface: In Figure 8, we focus on the electrostatic potential on the surface of the protein 1cbn for all three of our proposed schemes. The difference is not that much noticeable unless we focus in the black squared region of the front side. This shows that even though there are some differences in the solvation free energies calculated by the proposed three methods, there is no significant difference for the electrostatic potentials for our choice for parameter values.

3.2.2. Solvation free energy of 24 proteins

Next we solve the nonlinear PBE and compute the solvation free energy for a collection of 24 proteins as in [13,15]. The dielectric constants and ionic strength are same as our study for the protein Crambin(1cbn), i.e., $\epsilon^- = 1, \epsilon^+ = 80$ and $I_s = 0.15$.

The solvation free energies of the proposed schemes are compared with the rMIB scheme [15] and ADI scheme [14] in Table 4 and Figure 9(a). In our simulations, the steady-state integration will be stopped when $\Delta E^n_{sol} < 10^{-4}$ or $T_{end} = 50$. For the GFM-ADIDR, GFM-LODCN, and GFM-LODIE methods, we take $\Delta t = 0.05$. Based on the same stopping criteria, we have also computed solvation free energies by using the ADI method [14], which does not involve regularization and interface treatment. To ensure the stability, a small enough $\Delta t = 0.001$ has to be used in the ADI scheme. We also note that the energies of the rMIB method [15] are computed from solving the linearized PBE, while all other energies in Table 4 are calculated from the nonlinear PBE. Nevertheless, this does not matter too much for a small ionic strength, such as $I_s = 0.15$.

It can be observed from Table 4 that the results from our proposed schemes are very close to those of the rMIB scheme while solving the nonlinear PBE instead of the linear PBE. As we have identified in the previous sections, GFM-ADIDR method appeared to be more accurate than the other two of our proposed schemes and achieves the same level of accuracy

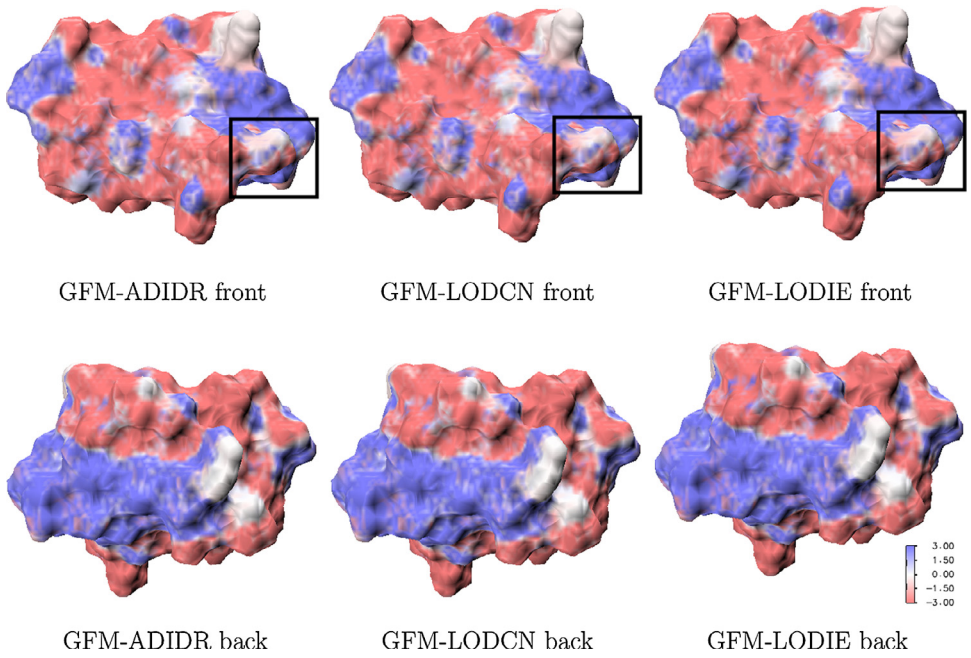

Fig. 8. Electrostatic potential for 1cbn using $\Delta t = 0.05$ and $h = 0.5$.

Table 4
Solvation free energies (*kcal/mol*) of 24 Proteins considering $\Delta t = 0.001$ for ADI and $\Delta t = 0.05$ for GFM-ADIDR, GFM-LODCN, GFM-LODIE. Stopping condition is set to be either $T_{end} = 50$ or $\Delta E^n_{sol} < 10^{-4}$.

PDB	No. of Atoms	rMIB [15]	ADI [14]	GFM-ADIDR	GFM-LODCN	GFM-LODIE
1ajj	519	-1139.48	-1371.10	-1139.45	-1139.07	-1133.06
2erl	573	-952.36	-1165.28	-952.13	-951.03	-945.96
1cbn	648	-303.33	-459.51	-302.06	-301.09	-297.06
1vii	596	-902.31	-1154.67	-901.78	-901.36	-895.15
1fca	729	-1204.44	-1458.16	-1205.53	-1205.09	-1199.33
1bbl	576	-988.40	-1302.49	-986.15	-985.72	-979.60
2pde	667	-820.97	-1018.66	-819.17	-819.83	-813.45
1sh1	702	-753.99	-999.92	-751.69	-751.84	-745.15
1vjw	826	-1241.07	-1513.17	-1244.74	-1243.52	-1236.67
1uxc	809	-1139.25	-1478.20	-1138.50	-1135.22	-1128.39
1ptq	795	-873.32	-1170.00	-867.98	-867.40	-859.71
1bor	832	-853.47	-1102.40	-852.49	-851.24	-844.68
1fxd	824	-3321.39	-3653.81	-3321.68	-3321.34	-3313.83
1r69	997	-1088.62	-1419.35	-1085.45	-1084.66	-1076.32
1mbg	903	-1353.31	-1685.70	-1352.63	-1351.30	-1343.68
1bpi	898	-1304.37	-1672.02	-1301.61	-1299.86	-1291.40
1hpt	858	-812.49	-1147.42	-809.02	-808.09	-799.24
451c	1216	-1027.21	-1379.27	-1023.71	-1022.61	-1012.70
1svr	1435	-1711.11	-2257.80	-1707.87	-1706.38	-1693.11
1frd	1478	-2862.50	-3376.35	-2863.69	-2863.03	-2850.42
1a2s	1272	-1921.20	-2292.15	-1919.28	-1917.70	-1907.96
1neq	1187	-1731.71	-2223.08	-1729.87	-1728.61	-1716.47
1a63	2065	-2374.41	-3149.69	-2370.80	-2369.26	-2350.42
1a7m	2809	-2160.34	-2771.41	-2155.05	-2152.48	-2135.73

as rMIB schemes. Table 4 also confirms that if the GFM-ADIDR method fails to converge for any protein then GFM-LODCN or GFM-LODIE can be used due to the fact that they are more stable and the results are not that much different than GFM-ADIDR.

In Figure 9, the solvation free energies of five methods are depicted in part (a). It is noticeable that the solvation free energies for GFM-ADIDR, GFM-LODCN, and GFM-LODIE are very close to each other. However, the solvation free energies from the ADI method are always 200 to 700 *kcal*/*mol* lower than the other methods. This justifies the development of regularization and GFM algorithm in the present work. With these new components, the GFM-ADIDR method becomes much more accurate than the ADI method. In Figure 9 (b), we take the rMIB energy as a reference, and the differences

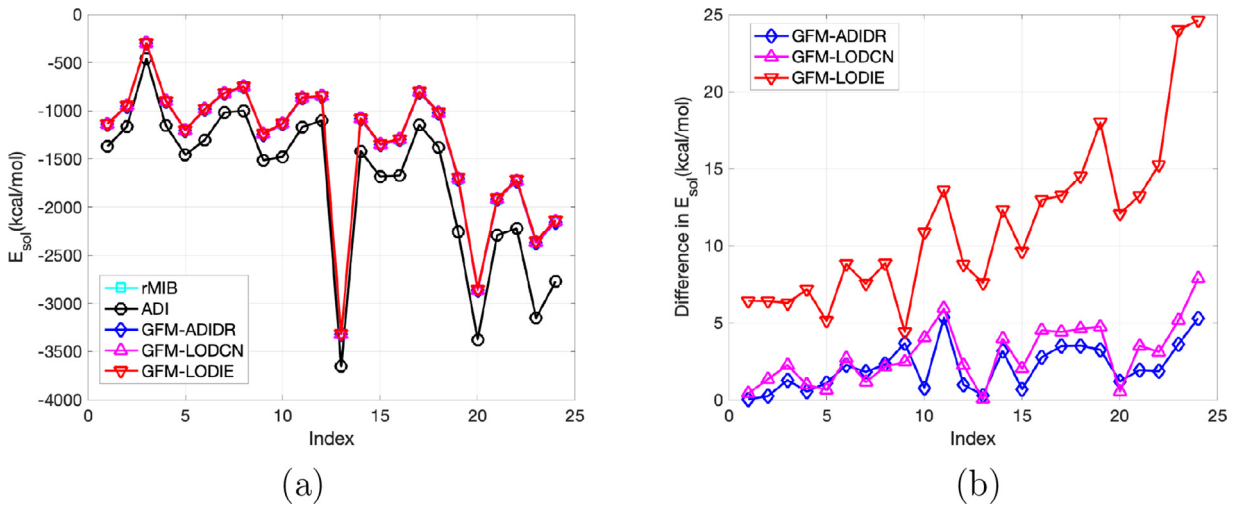

Fig. 9. (a): Solvation free energy of 24 proteins; (b): Difference in solvation free energy compared to rMIB method. Here "Index" is representing the set of all 24 proteins individually.

between rMIB and these new methods are reported. In this scale, one can see that the GFM-ADIDR and GFM-LODCN are more accurate than the GFM-LODIE.

The CPU time of four pseudo-transient approaches is reported in Figure 10 and is shown against the total number of unknowns for 24 proteins. In the first four subplots, a linear least-squares fitting is conducted on the log scale. In particular, denote N as the degree of freedom. Recall that for each pseudo-time step, the complexity of the proposed algorithms all scales as $O(N\log N)$. In this study, the CPU time for all steps is assumed to follow the pattern $\text{CPU} = C \cdot N^r$ for some constant C. Here the ratio r characterizes the complexity of the algorithm and can be estimated by the least-squares method in the log scale. It can be seen in Figure 10 that all four methods have a ratio of around 1.4, i.e., CPU is on the order of $O(N^{1.4})$. This demonstrates that the pseudo-transient approaches are actually quite efficient for biomolecular simulations. For the three new algorithms, the GFM-LODIE is most efficient with $r = 1.30$, while the GFM-LODCN is slowest with $r = 1.51$. The GFM-ADIDR method achieves the medium speed with $r = 1.41$. A direct comparison of four pseudo-transient methods is shown in part (e). It is seen that the ADI method takes around 10 times more CPU time compared to the newly proposed methods. A major reason behind that is the necessity of Δt to be as small as 0.001 for the ADI method to keep it stable and accurate enough for all 24 proteins tested here. For the newly proposed methods, we are able to use $\Delta t = 0.05$ requiring less time to compute the solvation free energies which are more accurate than the ADI method.

3.2.3. Binding energy of 2a9x

Binding energies play an important role in viral transcription and antiviral drug design [41]. In particular, a better accuracy of the binding energy of the BIV Tat Protein and BIV TAR RNA in the HIV viral replication can significantly help in search of new antiviral drugs that repress the replication by blocking transactivation of viral RNA transcription [24]. In this section, we demonstrate the ability of the GFM-ADIDR method to compute the binding energy of the BIV Tat Protein and BIV TAR RNA.

The electrostatic binding free energy can be calculated by the following formula based on the free energy cycle,

$$E^{\text{AB}}_{\text{bind}} = \Delta G^{\text{AB}}_{\text{ele}} - \Delta G^{\text{A}}_{\text{ele}} - \Delta G^{\text{B}}_{\text{ele}} = [E^{\text{AB}}_{\text{sol}} + E^{\text{AB}}_{\text{cou}}] - [E^{\text{A}}_{\text{sol}} + E^{\text{A}}_{\text{cou}}] - [E^{\text{B}}_{\text{sol}} + E^{\text{B}}_{\text{cou}}],$$

where the free energies of the complex AB and its monomers A and B on the right-hand side can be calculated from the solvation free energies in Eq. (34).

In Table 5, we find that the binding energies calculated by the GFM-ADIDR are very close to the results from rMIB method reported in [15]. This shows the ability of the GFM-ADIDR method to calculate other types of energies for large protein-ligand complexes like 2a9x. In Figure 11 the electrostatic potential of 2a9x and its monomers are visualized on their surface in a color scale. These type of visualizations help to provide optimized parameters for the molecular mechanics calculations of the drug candidate that will influence binding affinity [25].

3.2.4. Salt effect on the binding affinity

The nonlinear PBE is often used to describe the salt effects on the binding of ligands, peptides, and proteins to nucleic acids, membranes, and proteins. In this investigation, we have tested the performance of the proposed GFM-ADIDR scheme for the evaluation of the salt effect in the protein-protein binding of the complex Lactoglobulin dime(A-B) (PDB ID 1beb). Physically, the binding affinity can be quantitatively represented based on the binding-free energies, which reflect the non-specific salt dependence of the formation of macromolecular complexes. The binding affinity is then calculated as the slope

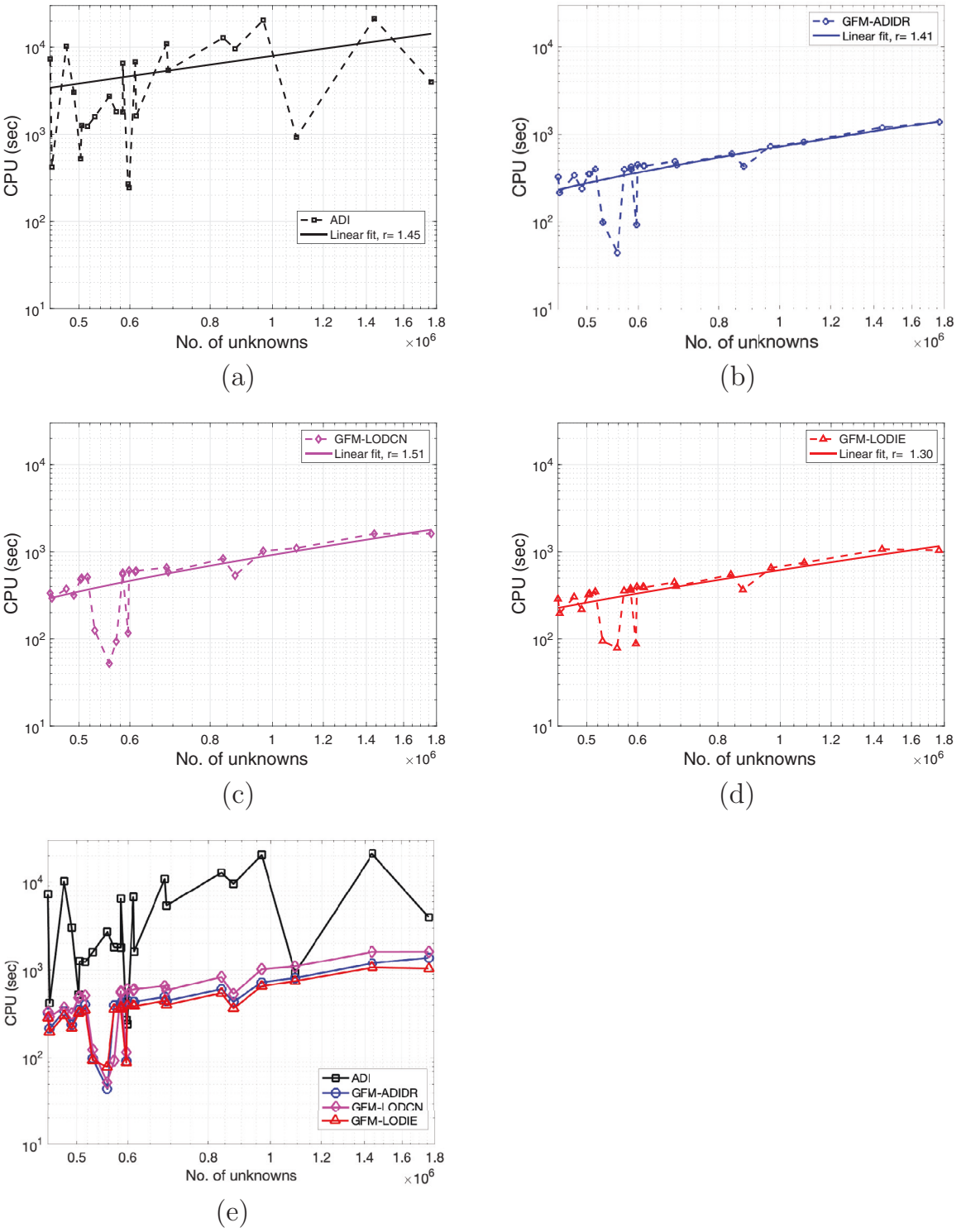

Fig. 10. The CPU time of four pseudo-transient method. In (a) - (d), a linear fitting is conducted and the corresponding ratio r is reported. The comparison of four methods is given in part (e).

Table 5
Binding energy of 2a9x

h	$E_{sol}^{complex}$	$E_{sol}^{protein}$	E_{sol}^{RNA}	$E_{bind}^{complex}$
		rMIB [15]		
1	-5816.38	-1021.94	-8893.39	383.60
1/2	-5821.22	-1025.86	-8898.54	387.84
1/4	-5823.39	-1026.27	-8900.52	388.05
		GFM-ADIDR		
1	-5834.10	-1027.14	-8915.17	392.84
1/2	-5824.82	-1025.98	-8905.59	391.38
1/4	-5841.62	-1026.40	-8916.69	386.12

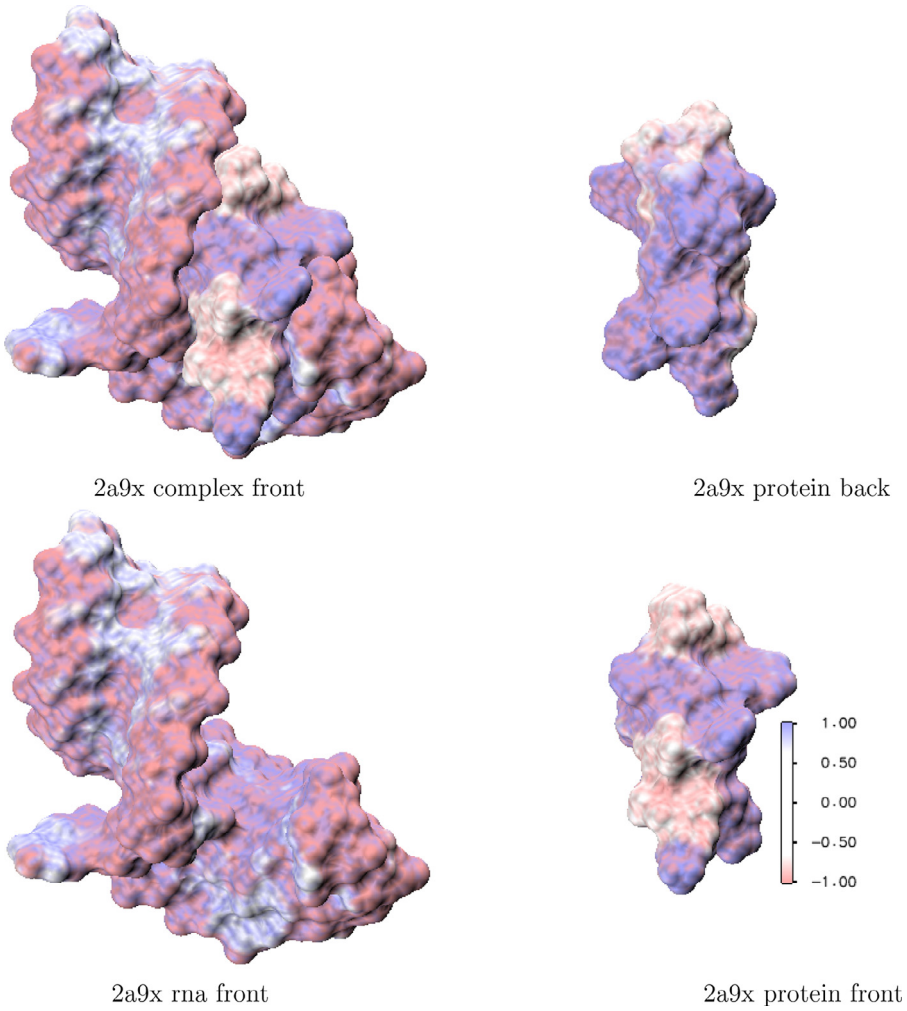

Fig. 11. Electrostatic potential for 2a9x using the GFM-ADIDR for $\Delta t = 0.05$ and $h = 0.5$

ratio of the salt-dependent binding energy at certain salt strength I_s against the natural logarithm of I_s. The electrostatic binding-free energy can be further split into $E_{cou}(I_s)$'s as the salt-independent parts and $E_{sol}(I_s)$'s as the salt-dependent parts. The variation of the salt-dependent part of the binding-free energy $\Delta E_{bind}(I_s)$ can thus be calculated as the difference in $E_{bind}(I_s)$ for some nonzero salt strength and the zero salt concentration because the salt independent parts cancel out. Altogether we have the following formula:

$$\begin{aligned}\Delta E_{bind}(I_s) &= E_{bind}(I_s) - E_{bind}(0)\\ &= [E_{sol}^{AB}(I_s) - E_{sol}^{AB}(0)] - [E_{sol}^{A}(I_s) - E_{sol}^{A}(0)] - [E_{sol}^{B}(I_s) - E_{sol}^{A}(0)].\end{aligned}$$

For this study, we use the same model parameters as earlier for 24 proteins. In Figure 12, we report the calculated binding free energy with the experimental result. The slope ratio or the binding affinity is calculated and reported in Table 6

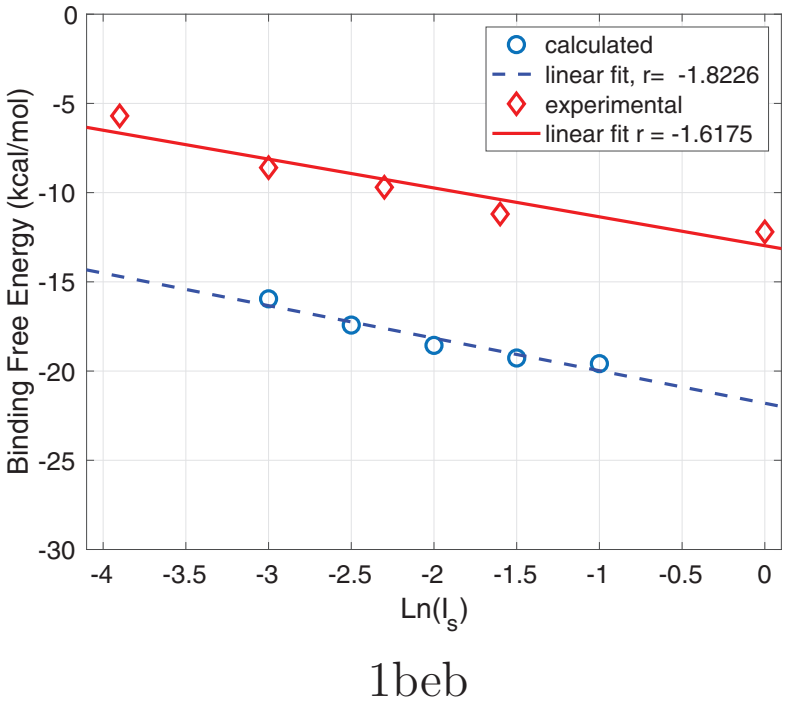

Fig. 12. The salt dependence of the binding affinities.

Table 6
Comparison of the binding affinity of 1beb.

	Charges			Slope ratios		
PDB	AB	A	B	Experimental	GFM-ADIDR	LFLPB [6]
1beb	+26	+13	+13	−1.62	−1.82	−2.02

as in [46]. The result attained by the Lagrangian formulation linearized PB (LFLPB) model [6] is also given in Table 6 for a comparison. We observe the binding affinity calculated by the GFM-ADIDR method to be sufficiently close to experimental data and is better than the LFLPB.

4. Conclusion

The pseudo-transient and regularization methods are recently developed approaches to solve the nonlinear Poisson-Boltzmann equation (PBE). Even though these two methods were successful in circumventing different challenges of the PBE, each one of them was lacking the advantages of the other one. In this study, when we attempted to combine them, we faced a new challenge due to the nonzero jump conditions, namely discontinuous solution and flux across the molecular surface. A delicate interface treatment is thus called. The matched interface and boundary (MIB) [4,13,45,49,50] and matched alternating direction implicit (mADI) [27,42,48] methods could be good choices for this type of interface problems, but the MIB scheme would have ruined the tridiagonal structure of the finite difference operator matrix, while the mADI method could have stability issue due to its semi-implicit nature. This motivates us to consider the GFM method [30], which produces a symmetric and tridiagonal structure. Moreover, the GFM method has been further modified in this work to reduce the interface information, so that it becomes simpler in implementation for geometrically complicated molecular surfaces. Three pseudo-time methods have been developed in this work, including one ADIDR scheme and two locally-one-dimensional methods using either Crank-Nicolson (LODCN) or implicit-Euler (LODIE) time stepping.

In numerical experiments, the accuracy, efficiency, and stability of three proposed methods, as well as the existing ADI scheme [14], are investigated. Without regularization and interface treatment, the ADI method is much less accurate than the present algorithms. Moreover, we have found that with two-component regularization and modified GFM algorithm, which are primarily designed for accuracy, the GFM-ADIDR method is much more stable than the ADI method. This makes the GFM-ADIDR method a very practical approach for electrostatic analysis, while the ADI method is impractical by requiring too small pseudo-time step size.

A detailed comparison of three new algorithms has also been conducted. The GFM-LODIE and GFM-LODCN are found to be unconditionally stable for all cases, while the GFM-ADIDR is conditionally stable with a mild stability constraint. The

computational complexities of three methods are all scaled like $O(N^{1.4})$ in protein studies, where N is the degree of freedom. This demonstrates the efficiency of the proposed pseudo-transient methods. In particular, the GFM-LODIE is the fastest one, while the GFM-LODCN is the slowest. In terms of spatial order, by using a small Δt, the convergence rates of three methods are all greater than one. This justifies the effectiveness of the proposed GFM scheme for interface treatment. By taking into account both temporal and spatial accuracies, the GFM-ADIDR and GFM-LOD methods could have different ranges of the application domain. In general, the GFM-ADIDR method is recommended when a high precision is required, while the GFM-LOD methods shall be employed when the efficiency matters.

In the future, we plan to explore an adaptive pseudo-time stepping in the new PBE algorithms. The adaptive selection of Δt in pseudo-transient approaches has been studied in [1,22,32]. The general goal will be improving the computational efficiency by reaching the steady state faster with an adaptive Δt, while maintaining the same accuracy.

Acknowledgment

This research is partially supported by the Simons Foundation award 524151 and the National Science Foundation (NSF) of USA under grant DMS-1812930.

References

[1] R.E. Bank, D.J. Rose, Parameter selection for newton-like methods applicable to nonlinear partial differential equations, SIAM Journal on Numerical Analysis 17 (6) (1980) 806–822, doi:10.1137/0717068.
[2] D. Bochkov, F. Gibou, Solving poisson-type equations with robin boundary conditions on piecewise smooth interfaces, Journal of Computational Physics 376 (2019) 1156–1198, doi:10.1016/j.jcp.2018.10.020.
[3] Q. Cai, J. Wang, H.K. Zhao, R. Luo, On removal of charge singularity in Poisson-Boltzmann equation, Journal of Chemical Physics 130 (14) (2009).
[4] D. Chen, Z. Chen, C. Chen, W. Geng, G.W. Wei, MIBPB: A software package for electrostatic analysis, Journal of Computational Chemistry 32 (4) (2011) 756–770, doi:10.1002/jcc.21646.
[5] L. Chen, M. Holst, J. Xu, The Finite Element Approximation of the Nonlinear Poisson-Boltzmann equation, SIAM Journal on Numerical Analysis 45 (6) (2007) 2298–2320.
[6] Z. Chen, N. Baker, G. Wei, Differential geometry based solvation model ii: Lagrangian formulation., Journal of mathematical biology 63 (6) (2011) 1139–1200.
[7] I.-L. Chern, Y.-C. Shu, A coupling interface method for elliptic interface problems, Journal of Computational Physics 225 (2) (2007) 2138–2174, doi:10.1016/j.jcp.2007.03.012.
[8] A. Coco, G. Russo, Second order finite-difference ghost-point multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, Journal of Computational Physics 361 (2018) 299–330, doi:10.1016/j.jcp.2018.01.016.
[9] W. Deng, J. Xu, S. Zhao, On developing stable finite element methods for pseudo-time simulation of biomolecular electrostatics, Journal of Computational and Applied Mathematics 330 (2018) 456–474, doi:10.1016/j.cam.2017.09.004.
[10] R.E. Ewing, Z. Li, T. Lin, Y. Lin, The immersed finite volume element methods for the elliptic interface problems, Mathematics and Computers in Simulation 50 (1) (1999) 63–76, doi:10.1016/S0378-4754(99)00061-0.
[11] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method), Journal of Computational Physics 152 (2) (1999) 457–492.
[12] W. Geng, G. Wei, Multiscale molecular dynamics using the matched interface and boundary method, Journal of Computational Physics 230 (2) (2011) 435–457.
[13] W. Geng, S. Yu, G. Wei, Treatment of charge singularities in implicit solvent models, The Journal of Chemical Physics 127 (11) (2007) 114106.
[14] W. Geng, S. Zhao, Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation, Molecular Based Mathematical Biology 1 (2013) 109–123.
[15] W. Geng, S. Zhao, A two-component Matched Interface and Boundary (MIB) regularization for charge singularity in implicit solvation, Journal of Computational Physics 351 (2017) 25–39.
[16] M.K. Gilson, M.E. Davis, B.A. Luty, J.A. McCammon, Computation of electrostatic forces on solvated molecules using the Poisson-Boltzmann equation, The Journal of Physical Chemistry 97 (14) (1993) 3591–3600.
[17] A. Helgadóttir, Y.T. Ng, C. Min, F. Gibou, Imposing mixed dirichlet-neumann-robin boundary conditions in a level-set framework, Computers & Fluids 121 (2015) 68–80, doi:10.1016/j.compfluid.2015.08.007.
[18] M.J. Holst, Multilevel Methods for the Poisson-Boltzmann Equation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1993 Ph.D. thesis. UMI Order No. GAX94-11653
[19] M.J. Holst, J.A. McCammon, Z. Yu, Y. Zhou, Y. Zhu, Adaptive Finite Element Modeling Techniques for the Poisson-Boltzmann equation, Communications in computational physics m (1) (2010) 28.
[20] M.J. Holst, F. Saied, Numerical solution of the nonlinear Poisson–Boltzmann equation: Developing more robust and efficient methods, Journal of Computational Chemistry 16 (3) (1995) 337–364.
[21] W. Im, D. Beglov, B. Roux, Continuum solvation model: Computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation, Computer Physics Communication 11 (1-3) (1998) 59–75.
[22] C.T. Kelley, D.E. Keyes, Convergence analysis of pseudo-transient continuation, SIAM Journal on Numerical Analysis 35 (2) (1998) 508–523, doi:10.1137/S0036142996304796.
[23] J. Kraus, S. Nakov, S. Repin, Reliable computer simulation methods for electrostatic biomolecular models based on the poisson-boltzmann equation, (2018), preprint.
[24] T.C. Leeper, Z. Athanassiou, R.L. Dias, J.A. Robinson, G. Varani, TAR RNA recognition by a cyclic peptidomimetic of Tat protein, Biochemistry 44 (37) (2005) 12362–12372.
[25] R.A. Lewis, Chapter 4 The Development of Molecular Modelling Programs: The Use and Limitations of Physical Models, The Royal Society of Chemistry, 2012, doi:10.1039/9781849733410-00088.
[26] C. Li, Z. Wei, C. Campbell, S. Ashlyn, G. Long, S. Zhao, Alternating direction ghost-fluid methods for solving the heat equation with interfaces, Preprint (2020).
[27] C. Li, S. Zhao, A matched Peaceman–Rachford ADI method for solving parabolic interface problems, Applied Mathematics and Computation 299 (2017) 28–44.
[28] Z. Li, Y.-Q. Shen, A numerical method for solving heat equations involving interfaces, in: Electronic Journal of Differential Equations, Conf, 3, 1999, pp. 100–108.
[29] J. Liu, Z. Zheng, IIM-based ADI finite difference scheme for nonlinear convection-diffusion equations with interfaces, Applied Mathematical Modelling 37 (3) (2013) 1196–1207.

[30] X.-D. Liu, R.P. Fedkiw, M. Kang, A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains, Journal of Computational Physics 160 (1) (2000) 151–178.
[31] B.A. Luty, M.E. Davis, J.A. McCammon, Solving the finite-difference non-linear Poisson–Boltzmann equation, Journal of Computational Chemistry 13 (9) (1992) 1114–1118.
[32] S. Pollock, A regularized newton-like method for nonlinear pde, Numerical Functional Analysis and Optimization 36 (11) (2015) 1493–1511, doi:10.1080/01630563.2015.1069328.
[33] W. Rocchia, E. Alexov, B. Honig, Extending the applicability of the nonlinear Poisson-Boltzmann equation: Multiple dielectric constants and multivalent ions, Journal of Physical Chemistry B 105 (28) (2001) 6507–6514.
[34] M.F. Sanner, A.J. Olson, J.-C. Spehner, Reduced surface: An efficient way to compute molecular surfaces, Biopolymers 38 (3) (1996) 305–320, doi:10.1002/(SICI)1097-0282(199603)38:3<305::AID-BIP4>3.0.CO;2-Y.
[35] A. Sayyed-Ahmad, K. Tuncay, P.J. Ortoleva, Efficient solution technique for solving the Poisson-Boltzmann equation, Journal of Computational Chemistry 25 (8) (2004) 1068–1074.
[36] K.A. Sharp, B. Honig, Calculating total electrostatic energies with the nonlinear poisson-boltzmann equation, The Journal of Physical Chemistry 94 (19) (1990) 7684–7692.
[37] A. Shestakov, J. Milovich, A. Noy, Solution of the nonlinear Poisson-Boltzmann equation using pseudo-transient continuation and the finite element method, Journal of Colloid and Interface Science 247 (1) (2002) 62–79.
[38] O. Steinbach, Springer-Verlag New York, 2008, doi:10.1007/978-0-387-68805-3.
[39] J. Strikwerda, Finite Difference Schemes and Partial Differential Equations, SIAM, 2004.
[40] M.M. Teeter, S. Roe, N.H. Heo, Atomic resolution (0.83 A) crystal structure of the hydrophobic protein crambin at 130 k, Journal of Molecular Biology 230 (1) (1993) 292–311.
[41] D.-P. Wang, R.C. Rizzo, J. Tirado-Rives, W.L. Jorgensen, Antiviral drug design: computational analyses of the effects of the l100i mutation for hiv-rt on the binding of nnrtis, Bio-organic and Medicinal Chemistry Letters 11 (21) (2001) 2799–2802.
[42] Z. Wei, C. Li, S. Zhao, A spatially second order alternating direction implicit (ADI) method for solving three dimensional parabolic interface problems, Computers and Mathematics with Applications (2017).
[43] L. Wilson, S. Zhao, Unconditionally stable time splitting methods for the electrostatic analysis of solvated biomolecules, International Journal of Numerical Analysis and Modeling 13 (6) (2016) 852–878.
[44] D. Xie, New solution decomposition and minimization schemes for Poisson–Boltzmann equation in calculation of biomolecular electrostatics, Journal of Computational Physics 275 (2014) 294–309.
[45] S. Yu, W. Geng, G.W. Wei, Treatment of geometric singularities in implicit solvent models, The Journal of Chemical Physics 126 (24) (2007) 244108, doi:10.1063/1.2743020.
[46] S. Zhao, Pseudo-time-coupled nonlinear models for biomolecular surface representation and solvation analysis, International Journal for Numerical Methods in Biomedical Engineering 27 (12) (2011) 1964–1981.
[47] S. Zhao, Operator splitting adi schemes for pseudo-time coupled nonlinear solvation simulations, Journal of Computational Physics 257 (2014) 1000–1021.
[48] S. Zhao, A Matched Alternating Direction Implicit (ADI) Method for Solving the Heat Equation with Interfaces, Journal of Scientific Computing (2015).
[49] S. Zhao, G. Wei, High-order FDTD methods via derivative matching for Maxwell's equations with material interfaces, Journal of Computational Physics 200 (1) (2004) 60–103, doi:10.1016/j.jcp.2004.03.008.
[50] Y. Zhou, S. Zhao, M. Feig, G. Wei, High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources, Journal of Computational Physics 213 (1) (2006) 1–30, doi:10.1016/j.jcp.2005.07.022.
[51] Z. Zhou, P. Payne, M. Vasquez, N. Kuhn, M. Levitt, Finite-difference solution of the Poisson–Boltzmann equation: Complete elimination of self-energy, Journal of Computational Chemistry 17 (11) (1996) 1344–1351.