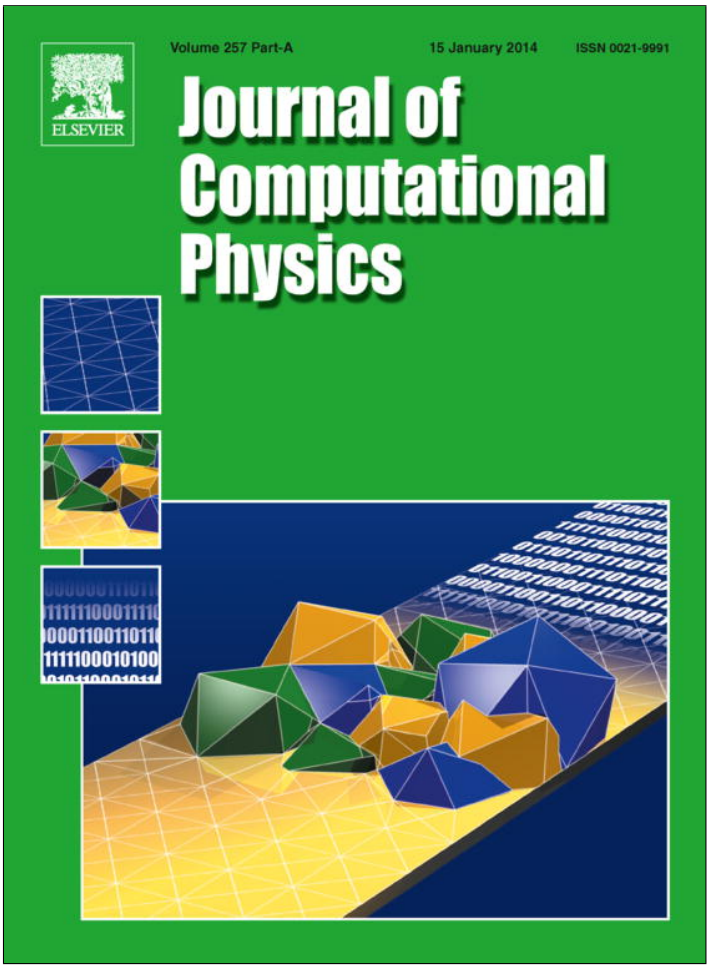

Journal of Computational Physics 257 (2014) 1000–1021

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Operator splitting ADI schemes for pseudo-time coupled nonlinear solvation simulations

Shan Zhao *

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

ARTICLE INFO

Article history:
Received 26 April 2012
Received in revised form 25 July 2013
Accepted 25 September 2013
Available online 1 October 2013

Keywords:
Solvation free energy
Molecular surface
Electrostatic potential
Nonlinear Poisson–Boltzmann equation
Operator splitting
Alternating direction implicit scheme

ABSTRACT

This work introduces novel operator splitting alternating direction implicit (ADI) schemes to overcome numerical difficulties in solving pseudo-time coupled nonlinear partial differential equations (PDEs) for biomolecular solvation analysis. Based on the variational formulation, a pseudo-transient continuation model has been previously formulated to couple a nonlinear Poisson–Boltzmann (NPB) equation for the electrostatic potential with a generalized Laplace–Beltrami equation defining the biomolecular surface. However, the standard numerical integration of the time dependent NPB equation is known to be very inefficient. Moreover, it encounters instability issues for smoothly varying solute–solvent interfaces so that a filtering process has to be conducted. In the present work, we propose to solve the unsteady NPB equation in an operator splitting framework so that the nonlinear instability can be completely avoided through an analytical integration. Central finite differences are employed to discretize the nonhomogeneous diffusion term of the generalized NPB equation to form tridiagonal matrices in the Douglas and Douglas–Rachford type ADI schemes. The proposed time splitting ADI schemes are found to be unconditionally stable for solving the NPB equation in benchmark examples with analytical solutions. For the solvation analysis involving two pseudo-time coupled nonlinear PDEs, the time stability of the NPB equation can be maintained by using very large time increments, so that without sacrificing the accuracy, the present biomolecular simulation becomes over ten times faster.

1. Introduction

In order to carry out quantitative description and analysis of various important biological processes at the atomic level, including signal transduction, DNA recognition, transcription, translation, protein folding and protein ligand binding, the analysis of the underlying biomolecular solvation is indispensable. This is because these important processes occur naturally in water, which comprises 65–90% of cellular mass. Biologically, the solvation analysis concerns with interactions between solute macromolecules and the surrounding solvent molecules or ions. Mathematically, solute–solvent interactions can be represented via solvation energies which have both polar and nonpolar contributions. The polar portion of solvation accounts for electrostatic interactions, which are ubiquitous for any system with charged molecules. The nonpolar portion consists of the contributions due to the surface tension, mechanical work, and attractive solvent–solute dispersion interactions. The implicit solvent theory, in which the solvent is treated as a continuous dielectric medium, is one of the most commonly used approaches for studying the biomolecular solvation in the literature [25,23,1].

* Tel.: +1 205 3485155; fax: +1 205 3487067.
E-mail address: szhao@bama.ua.edu.

0021-9991/$ – see front matter
http://dx.doi.org/10.1016/j.jcp.2013.09.043

Recently, a family of differential geometry based multiscale implicit solvent models has been developed for analyzing the equilibrium properties of solvation [29,7,8]. Computational realizations of such solvation models in both the Eulerian and Lagrangian formulations have been carried out [7,8]. Based on the fundamental laws of physics, a free energy minimization or optimization process is conducted in these models. The total free energy functional for the solvation analysis typically consists of the electrostatic potential, the geometrical effect of the solvent–solute interface, the mechanical work of the system and the dispersive solvent–solute interaction [29,7,8]. Mathematically, by using the Euler–Lagrange variation, two coupled nonlinear partial differential equations (PDEs) are derived as governing equations [7,8] – one generalized nonlinear Poisson–Boltzmann (NPB) equation for the electrostatic potential and one generalized Laplace–Beltrami equation defining the solvent–solute interface or the biomolecular surface. These differential geometry based multiscale models can be generalized to study more complex chemical and biological systems, such as molecular motors, ion channels, DNA packing, nanofluidic system, and virus evolution, by incorporating not only structure and solvent–solute interactions, but also molecular dynamics and fluid transports, into the total free energy functional [29]. More coupled PDEs are usually resulted [29].

More recently, we have proposed a pseudo-transient continuation model for the theoretical modeling of the biomolecular surface and solvation process [32]. The major improvement of this differential geometry based multiscale model in comparison with the previous ones [29,7,8] is a more efficient coupling of underlying nonlinear PDEs through the introduction of a pseudo-time in each process, giving rise to a time dependent NPB equation and a time dependent Laplace–Beltrami or geometric flow equation. By treating the solution of the nonlinear boundary value system as the steady state solution of a time dependent process, the overall model coupling is accomplished by the explicit Euler time integration and controlled by time increments. This coupling is simpler than the relaxation based iterative procedure used in the literature [7,8], and involves less controlling parameters. Moreover, the NPB equation can be treated in the same manner as the linearized Poisson–Boltzmann (LPB) equation, which is impossible in a conventional coupling. Based on the Eulerian formulation [7], the pseudo-time coupled solvation model [32] makes use of a smooth biomolecular surface, i.e., an overlapping transition region as the solute–solvent boundary. Computationally, the smooth solvent–solute interface possesses good differentiability in geometric flow evolution and generates a smooth dielectric profile for the electrostatic potential so that the central finite difference scheme can be simply employed for the spatial discretization of the generalized NPB equation [32].

However, there are considerable numerical difficulties associated with the temporal discretization of the pseudo-time coupled solvation model [32]. Such difficulties are essentially due to the nonlinear reaction term of the generalized NPB equation, which involves a hyperbolic sine function of the electrostatic potential in the univalent mobile ions setting. By using the explicit Euler scheme [32], a very small time increment Δt has to be used for updating the electrostatic potential, in order to balance this nonlinear term. For large protein systems, the Δt value for the NPB equation is more than 100 times smaller than that for the geometric flow equation in a synchronized pseudo-time coupling process. Moreover, even with such a small Δt, instability issues could be encountered at the smooth transition region of the solute–solvent boundary [32]. Basically, the nonlinear term of the generalized NPB equation is not completely shut off in regions that were originally belonged to the solute subdomain. On the other hand, according to Coulomb's law, the electrostatic potential could be very large in such regions, if it is close to some singular charges. When being exponentially amplified by the nonlinear term, such a potential value becomes numerically blowing-up. In our previous study, an artificial filtering process has to be employed to stabilize the computation [32].

In short, the time integration of the generalized NPB equation represents the major bottleneck in the numerical solution of the pseudo-transient solvation model. We will investigate novel temporal schemes in the present study for solving the NPB equation, while keep using the explicit Euler scheme for the time dependent Laplace–Beltrami equation.

A similar nonlinear term is also presented in the classical NPB equation, which is a widely used model in electrochemistry, solution chemistry, colloid chemistry, and biophysics, to describe the electrostatic interaction and ionic density distributions of a solvation process at the equilibrium state [13,1,18]. The dominant numerical approach for solving the NPB equation employs a finite difference or finite element discretization to form a nonlinear algebraic system

$$\mathbf{AU} + \mathbf{N}(\mathbf{U}) = \mathbf{b} \tag{1}$$

where $\mathbf{U}$ is the potential vector, $\mathbf{b}$ is the free charge vector, and $\mathbf{A}$ and $\mathbf{N}$ are, respectively, discrete operators for the linear and nonlinear parts of the NPB equation. Various algebraic methods have been proposed in the literature to solve Eq. (1) iteratively, such as nonlinear relaxation methods [15,22], nonlinear conjugate gradient method [19], and inexact Newton method [14]. Based on linear iterative methods, such as Gauss–Seidal or Jacobi, the nonlinear relaxation methods employ lower triangular matrices to approximate the inverses in solving Eq. (1), and are built in widely used biomolecular simulation packages such as Delphi [22] and PBEQ [15]. The nonlinear conjugate gradient method approximately solves a minimization problem with a nonlinear integral functional, and is built in the UHBD package [19]. The inexact Newton method solves the Jacobian system inexactly to find a descent direction and conducts a linear search along the descent direction to ensure global convergence [14]. In the APBS package, a fast algebraic multigrid solver has been combined with the inexact Newton method for solving the NPB equation [13,14]. A comparison of various nonlinear algebraic NPB solvers has been conducted in [6].

The first pseudo-transient continuation approach for solving the classical NPB equation via a long time solution of a convection–diffusion process is introduced in [26]. By employing the diffusion module of an existing finite element software, the pseudo-time dependent NPB equation is discretized in time by the implicit Euler scheme in [26]. A linearization

technique based on first order Taylor's expansion is then utilized to generate the coefficient matrix of the linear system at each time step by evaluating the nonlinear terms at the previous time instant. In other words, the nonlinear term of the NPB equation is treated explicitly, so that the overall time integration is semi-implicit. A more efficient semi-implicit approach is introduced in [24], in which the convection and nonlinear terms of the time dependent NPB equation are treated explicitly, while the diffusion term is integrated implicitly by a regularized alternating direction implicit (ADI) method [10]. In both semi-implicit approaches [26,24], a large time increment is allowed so that the steady state can be reached efficiently. Moreover, the semi-implicit ADI method [24] is usually faster because a tridiagonal finite difference system is solved in each alternating direction by the Thomas algorithm [27] and the regularization technique allows an even larger time increment [10].

We note that there is a subtle difference between the nonlinear term in the classical NPB equation [13,1,18] and that in the generalized NPB equation of the differential geometry based solvation models [29,7,32]. In particular, the nonlinear term is defined only on the solvent domain through a Heaviside step function in the classical NPB equation, while it is defined on both solvent domain and the smooth solvent–solute transition boundary in the generalized NPB equation. In other words, in the latter case, the electrostatic potential may even take nonzero values in the solute domain which is inside the so-called ion-exclusion layer [13,32]. By Coulomb's law, the potential could be large if it is quite close to a singular charge so that an extremely large value is resulted after evaluating the nonlinear term. Thus, the instability issue caused by the nonlinear term is more severe in the generalized NPB case than that in the classical NPB case. Nevertheless, it is noted that the spatial discretization of the generalized NPB equation is simpler than that of the classical NPB equation. The central finite difference can be simply employed for the former because of a smooth dielectric profile, whereas sophisticated interface treatments [34,12,31] are required for the latter due to a discontinuous dielectric profile.

The objective of this paper is to introduce novel operator splitting or time splitting ADI methods for solving the generalized NPB equation in the pseudo-transient solvation model [32]. The time splitting and ADI schemes [3,2,30,33,28,9,11,17] are powerful tools for solving time dependent PDEs due to their capability in enlarging the Courant–Friedrich–Lewy (CFL) number or even completely bypassing the CFL constraint if all split subsystems are analytically integrable. In the present study, an operator splitting framework is constructed so that the nonlinear subsystem of the NPB equation can be analytically integrated. Then, both Douglas and Douglas–Rachford ADI schemes [27] are formulated for solving the linear differentiation subsystem of the NPB equation. Comparing with the ADI scheme for the classical NPB equation [24], the proposed ADI schemes have two distinct features. First, the time dependent NPB equation will not be rewritten into a convection–diffusion process. Instead, central finite difference is employed to directly discretize the nonhomogeneous diffusion term in the present study so that the resulting linear system is more compact. Second, the nonlinear term is integrated analytically in the present operator splitting ADI schemes, while it is evaluated at the previous time step in the semi-implicit ADI scheme [24]. Due to these two features, the proposed ADI schemes are fully implicit. Therefore, they remain to be unconditionally stable when applied to the generalized NPB equation alone, and admit large time increments when applied to coupled PDEs of the pseudo-transient solvation system.

The rest of this paper is organized as the follows. In Section 2, we first introduce the pseudo-transient solvation model in the Eulerian formulation. The proposed time splitting ADI schemes will then be formulated for solving the generalized NPB equation. Numerical validations of the proposed ADI schemes through benchmark examples with analytical solutions are considered in Section 3. Both time dependent and time independent NPB equations will be studied. The applications of the proposed algorithms to various benchmark biological systems will be explored in Section 4. Finally, this paper ends with a conclusion.

2. Mathematical models and numerical algorithms

2.1. Physical background and mathematical models

Consider a solute macromolecule, such as protein, DNA or RNA, being immersed in an aqueous solvent environment. The differential geometry based solvation models [29,7,8,32] to be studied in this paper are all based on the multiscale implicit solvent theory [25,23,1], in which the aqueous solvent is treated as a continuum, while the macromolecule is described in discrete atomic detail. Mathematically, the entire domain $\Omega \subset \mathbb{R}^3$ of this solute–solvent system naturally consists of two regions, i.e., a macromolecular domain Ω_m and an aqueous solvent domain Ω_s with $\Omega = \Omega_s \cup \Omega_m$.

In biomolecular simulations, the boundary between the solvent and solute domains can be modeled as either a sharp interface [16,21,8] or a smooth interface [7,32]. A smooth solvent–solute interface model will be used in the present study, which is free of geometric singularities. In particular, we define an overlapping transition region Ω_b as the smooth solute–solvent boundary. Here $\Omega_b = \Omega_s \cap \Omega_m \neq \emptyset$. This multi-domain setting can be represented by a characteristic function for the solute domain $S : \mathbb{R}^3 \to \mathbb{R}$. Inside the biomolecule, S takes value one, while it becomes zero in the aquatic solvent. A continuous transition with values from one to zero is assumed by S at the solute–solvent boundary region Ω_b. On the other hand, $(1 - S)$ is the characteristic function for the solvent domain. Sample profiles of S and $(1 - S)$ along one straight line are depicted in Fig. 1(a). We note that both S and $(1 - S)$ take nonzero values at the solute–solvent boundary where Ω_s and Ω_m overlap. A sharp solute–solvent interface can be regarded as the limiting situation of the present smooth interface.

In the differential geometry based solvation models [29,7,8,32], one first defines the total solvation free energy as a combination of polar and nonpolar components. The polar component, i.e., the polar solvation energy, is due to the electrostatic

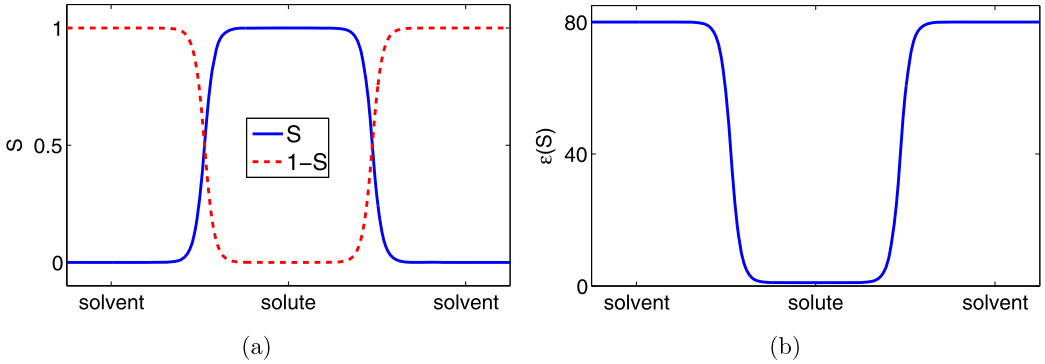

Fig. 1. (a) The characteristic functions S and $(1-S)$ of the solute and solvent regions along one straight line. (b) The dielectric function $\epsilon(S)$ along one straight line with $\epsilon_s = 80$ and $\epsilon_m = 1$.

interaction – a process which is ubiquitous for any system of charged or polar molecules. The nonpolar component typically involves the mechanical work of creating the vacuum of a biomolecular size in the solvent, the attractive dispersion effects near the solute–solvent interface, and the surface energy, which measures the disruption of intermolecular and/or intramolecular bonds that occurs when a surface is created.

The total solvation free energy can be regarded as a functional of two independent functions: the characteristic or hypersurface function $S(\mathbf{r})$ and the electrostatic potential $\phi(\mathbf{r})$ defined on the entire domain $\mathbf{r} \in \Omega$. Thus, the total free energy can be minimized via an optimal function $S(\mathbf{r})$ and an optimal potential $\phi(\mathbf{r})$ at the equilibrium state. Mathematically, such an optimal state can be achieved via the variational analysis, which leads to two coupled nonlinear PDEs [29,7,32]

$$-\nabla\cdot\left(\gamma\frac{\nabla S}{\|\nabla S\|}\right)+p-\rho_0 U^{\text{att}}+\rho_m\phi-\frac{\epsilon_m}{2}\|\nabla\phi\|^2+\frac{\epsilon_s}{2}\|\nabla\phi\|^2+k_BT\sum_{j=1}^{N_c}c_j\left(e^{-q_j\phi/k_BT}-1\right)=0, \tag{2}$$

$$\nabla\cdot\left(\epsilon(S)\nabla\phi\right)+(1-S)\sum_{j=1}^{N_c}c_jq_je^{-q_j\phi/k_BT}+S\rho_m=0, \tag{3}$$

with certain Dirichlet conditions prescribed for both $S(\mathbf{r})$ and $\phi(\mathbf{r})$ on $\partial\Omega$ – the boundary of the computational domain Ω.

Eq. (2) is a generalized Laplace–Beltrami equation defining the solute–solvent interface [29,7,32]. The first term $\nabla\cdot(\gamma\frac{\nabla S}{\|\nabla S\|})$ is a generalized Laplace–Beltrami operator, which is a generalization of the usual Laplacian operator to a smooth manifold. Here γ is the surface tension, which could be a function of the position $\gamma=\gamma(\mathbf{r})$ to reflect surface hydrophobicity at different locations. For simplicity, it will be treated as a constant in the present study. The next two terms are also of nonpolar nature. Here p is the hydrodynamic pressure representing volume mechanical work, ρ_0 is the solvent bulk density, and U^{att} is the attractive portion of the Van der Waals potential at a point $\mathbf{r}$. The last four terms are due to the electrostatic free energy, and among them, the first two terms and last two terms are originated from the solute and solvent, respectively. In the singular charge term, $\rho_m=\sum_j Q_j\delta(\mathbf{r}-\mathbf{r}_j)$ is the canonical density of molecular free charges, with Q_j being the partial charge on an atom located at $\mathbf{r}_j$. In the quadratic terms of the potential gradient, ϵ_m and ϵ_s are the electric permittivities of the macromolecule and solvent, respectively. In the Boltzmann term, the charge density of mobile ions is assumed to follow the Boltzmann distribution. Here k_B is the Boltzmann constant, T is the temperature, c_j is the bulk concentration of jth ionic species, q_j is the charge of the jth ionic species, and N_c is the number of ionic species.

Eq. (3) is a generalized nonlinear Poisson–Boltzmann (NPB) equation for the electrostatic potential ϕ [29,7,32]. Here a overall dielectric function is defined to be

$$\epsilon(S)=(1-S)\epsilon_s+S\epsilon_m. \tag{4}$$

One smooth profile of the dielectric function $\epsilon(S)$ is shown in Fig. 1(b). A smooth transition of dielectric values from ϵ_s to ϵ_m is clearly seen. From the numerical point of view, such a smooth dielectric profile prevents many computational difficulties in solving elliptic equations with discontinuous coefficients [34,12,31]. Nevertheless, the nonlinear Boltzmann term will introduce considerable difficulties in the present numerical study. We also note that by taking S to be a proper Heaviside function, the smooth transition region disappears and the dielectric function becomes discontinuous. Consequently, the present generalized NPB equation reduces to the classical NPB equation [13,1,18].

2.2. Pseudo-time coupled governing equations

Mathematically, the generalized Laplace–Beltrami equation (2) and the generalized NPB equation (3) constitute a coupled nonlinear boundary value system. In the previous variational solvation models [7,8], a simplified system based on the linearized Poisson–Boltzmann (LPB) process is usually considered, and a relaxation based iterative procedure is employed

to couple the reduced system. Theoretical modeling and computational simulation based on the full nonlinear system are highly nontrivial.

This motivates the development of a pseudo-time coupled solvation model [32]. Essentially, a pseudo-transient variation is introduced to convert both (2) and (3) from time independent forms to time dependent ones. For simplicity, we only present the pseudo-time coupled system under the assumption that all mobile ions are univalent in the solvent, because in this setting, the NPB equation takes its most commonly used nonlinear form. In particular, the pseudo-transient continuation for the electrostatic potential is attained via a time dependent NPB equation [32]

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \big(\epsilon(S)\nabla\phi\big) + S\rho_m - (1-S)\bar{\kappa}^2\left(\frac{k_B T}{e_c}\right)\sinh\left(\frac{e_c\phi}{k_B T}\right), \tag{5}$$

where e_c is the charge of an electron and $\bar{\kappa}$ is the modified Debye–Huckel parameter. This parameter is defined as

$$\bar{\kappa}^2 = \left(\frac{2N_A e_c^2}{1000 k_B T}\right) I_s, \tag{6}$$

where N_A is the Avogadro's number and I_s is the ion strength in the unit of mole. Numerically, when $T = 298$ K, we have $\bar{\kappa}^2 = (8.486902807\ \text{Å}^{-2})I_s$.

For the surface equation (2), the pseudo-transient continuation is attained via a potential driven geometric flow PDE [5,4,32],

$$\frac{\partial S}{\partial t} = \|\nabla S\|\left[\nabla \cdot \left(\gamma \frac{\nabla S}{\|\nabla S\|}\right) + V\right], \tag{7}$$

where V is a generalized potential, and will be simply calculated as

$$V = -p + \rho_0 U^{\text{att}} - \rho_m\phi + \frac{\epsilon_m}{2}\|\nabla\phi\|^2 - \frac{\epsilon_s}{2}\|\nabla\phi\|^2 - \frac{1}{2}\bar{\kappa}^2\phi^2 \tag{8}$$

in the present study. The attractive dispersion interaction U^{att} in (8) is calculated according to the Lennard-Jones potential [32].

Dirichlet boundary conditions are defined on $\partial\Omega$. The characteristic function S becomes zero away from the biomolecule. Thus, a Dirichlet zero condition is simply imposed for $S(\mathbf{r})$ on $\partial\Omega$. For the electrostatic potential ϕ, a far-field condition holds at the infinity, i.e., $\lim_{|\mathbf{r}|\to\infty}\phi(\mathbf{r}) = 0$. For a finite domain Ω, an approximate analytical condition can be employed

$$\phi(\mathbf{r}) = \sum_{i=1}^{N_m} \frac{Q_i}{\epsilon_s|\mathbf{r}-\mathbf{r}_i|} e^{-\bar{\kappa}|\mathbf{r}-\mathbf{r}_i|/\sqrt{\epsilon_s}}. \tag{9}$$

This condition (9) is actually a linear superposition of Coulomb's law for a series of N_m partial charges Q_i at positions $\mathbf{r}_i$ and is very accurate when $\partial\Omega$ is far away from the macromolecule subdomain.

The initial profiles of $S(\mathbf{r})$ and $\phi(\mathbf{r})$ are generated as the follows. The initial values of $S(\mathbf{r})$ is first defined as in [5,4]. Consider a macromolecule with total N_a number of atoms. Denote the center and radius of the ith atom to be $\mathbf{r}_i = (x_i, y_i, z_i)$ and r_i, respectively, for $i = 1, 2, \ldots, N_a$. We then define the domain enclosed by the solvent accessible surface to be $D = \bigcup_{i=1}^{N_a}\{\mathbf{r}\colon\ |\mathbf{r}-\mathbf{r}_i| < r_i + r_p\}$, where r_p is the probe radius. At $t = 0$, we define S to be

$$S(x, y, z, 0) = \begin{cases} 1, & (x, y, z) \in D, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

To enable a stable computation, we first evolve the hypersurface function S according to the geometric flow equation (7) in the absence of the driving potential V to a stopping time T_S. Typically, we choose $T_S = 3.5$ to generate a smooth hypersurface function $S(\mathbf{r})$. The initial profile for the electrostatic potential either can be chosen as $\phi = 0$ or can be obtained by solving a LPB equation [32]

$$-\nabla \cdot \big(\epsilon(S)\nabla\phi\big) + (1-S)\bar{\kappa}^2\phi = S\rho_m. \tag{11}$$

When t is sufficiently large, the steady state solutions under these two initial profiles converge to the same place.

With these initial and boundary values, the simultaneous time coupling of two nonlinear processes (5) and (7) will be conducted. The solution of the original boundary value system (2) and (3) in the univalent ions setting can be approximated by the resulted steady state solution. The overall coupling can be accomplished via standard time integration techniques, and controlled simply by time increments Δt. Typically, different Δt values shall be used for these two nonlinear parabolic processes, due to the different intrinsic time scale. Computationally, a long enough coupling time $T_c = 10$ is usually employed. An early convergence criterion is also allowed, i.e., the computation stops if the correction in the free energy of solvation is less than 10^{-6} [32]. To save the CPU time, only nontrivial values are updated in solving the geometric flow equation (7). In particular, we only solve the values of $S(x, y, z, t)$ at the nodes in between the Van der Waals surface and solvent accessible surface, i.e., $(x, y, z) \in \bigcup_{i=1}^{N_a}\{\mathbf{r}\colon\ r_i < |\mathbf{r}-\mathbf{r}_i| < (r_i + r_p)\}$. This implementation also protects the Van der Waals surface. In general, the present pseudo-time coupling is more efficient than the previous boundary value coupling and requires less controlling parameters [32].

2.3. Numerical discretization and difficulties

However, there exist grand difficulties in the numerical solution of the pseudo-time coupled solvation model. To gain an in-depth understanding, we first discuss a standard numerical discretization to the NPB equation and potential driven geometric flow equation. The central finite difference and explicit Euler scheme were used in [32] for spatial and temporal discretization, respectively.

To simplify the following discussions of numerical algorithms, the time dependent NPB equation (5) will be first rewritten in terms as the dimensionless electrostatic potential [13]

$$u(x, y, z, t) = \frac{e_c \phi(x, y, z, t)}{k_B T}.$$

By appropriately scaling Eq. (5), we arrive at the following time dependent NPB equation for u

$$\frac{\partial u}{\partial t} = \nabla \cdot \big(\epsilon(S)\nabla u\big) + cS\rho_m - (1-S)\bar{\kappa}^2 \sinh(u), \tag{12}$$

where $c = e_c/(k_B T)$. The Dirichlet boundary condition (9) can be similarly scaled:

$$u(\mathbf{r}) = c\sum_{i=1}^{N_m} \frac{Q_i}{\epsilon_s |\mathbf{r} - \mathbf{r}_i|} \mathrm{e}^{-\bar{\kappa}|\mathbf{r}-\mathbf{r}_i|/\sqrt{\epsilon_s}}. \tag{13}$$

We then rewrite the potential driven geometric flow equation (7) into the Cartesian components form [32]

$$\begin{aligned}\frac{\partial S}{\partial t} = {} & \gamma \frac{(S_x^2 + S_y^2)S_{zz} + (S_x^2 + S_z^2)S_{yy} + (S_y^2 + S_z^2)S_{xx}}{S_x^2 + S_y^2 + S_z^2} \\ & - \gamma \frac{2S_x S_y S_{xy} + 2S_x S_z S_{xz} + 2S_z S_y S_{yz}}{S_x^2 + S_y^2 + S_z^2} + \sqrt{S_x^2 + S_y^2 + S_z^2}\, V,\end{aligned} \tag{14}$$

where the subscripts of S denote partial derivatives, e.g. $S_x = \frac{\partial S}{\partial x}$. Numerically, to avoid a vanishing value in the denominator of Eq. (14), we add a very small number, say 10^{-7}, into the denominator, which does not affect the accuracy of our biomolecular simulation.

Consider a uniform mesh partition of the computational domain Ω. Without the loss of generality, we assume the grid spacing h in all x, y and z directions to be the same. Moreover, the same uniform mesh will be used for solving both the electrostatic potential u and the hypersurface function S. However, due to the different intrinsic time scale, different time increments have to be used for u and S, which will be denoted as Δt_u and Δt_S, respectively. To facilitate the following discussions, we adopt the following notation at a node (x_i, y_j, z_k, t_n): $f^n_{i,j,k} = \{f\}^n_{i,j,k} = f(x_i, y_j, z_k, t_n)$, where f could be the hypersurface function S, electrostatic potential u, generalized potential V, or their derivatives.

By using the explicit Euler scheme, the temporal derivative of Eq. (14) at a node (x_i, y_j, z_k, t_n) is approximated to be

$$\left\{\frac{\partial S}{\partial t}\right\}^n_{i,j,k} \approx \frac{S^{n+1}_{i,j,k} - S^n_{i,j,k}}{\Delta t_S},$$

while all terms on the right-hand side part of Eq. (14) are discretized based on the central differences at time level $t = t_n$. For instance, we have

$$\{S_x\}^n_{i,j,k} \approx \frac{S^n_{i+1,j,k} - S^n_{i,j,k}}{h} \quad \text{and} \quad \{S_{xx}\}^n_{i,j,k} \approx \frac{S^n_{i+1,j,k} - 2S^n_{i,j,k} + S^n_{i-1,j,k}}{h^2},$$

and similar approximations for $\{S_y\}^n_{i,j,k}$, $\{S_{yy}\}^n_{i,j,k}$, $\{S_z\}^n_{i,j,k}$ and $\{S_{zz}\}^n_{i,j,k}$. The mix derivatives $\{S_{xy}\}^n_{i,j,k}$, $\{S_{xz}\}^n_{i,j,k}$, and $\{S_{yz}\}^n_{i,j,k}$ are similarly discretized, e.g.

$$\{S_{yz}\}^n_{i,j,k} \approx \frac{S^n_{i,j+1,k+1} + S^n_{i,j-1,k-1} - S^n_{i,j+1,k-1} - S^n_{i,j-1,k+1}}{4h^2}.$$

With these approximations, the final discretization form of (14) can be given symbolically

$$\begin{aligned}\frac{S^{n+1}_{i,j,k} - S^n_{i,j,k}}{\Delta t_S} = {} & \frac{\gamma[(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2]\{S_{zz}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2} + \frac{\gamma[(\{S_x\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2]\{S_{yy}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2} \\ & + \frac{\gamma[(\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2]\{S_{xx}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2} - \frac{2\gamma\{S_x\}^n_{i,j,k}\{S_y\}^n_{i,j,k}\{S_{xy}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2}\end{aligned}$$

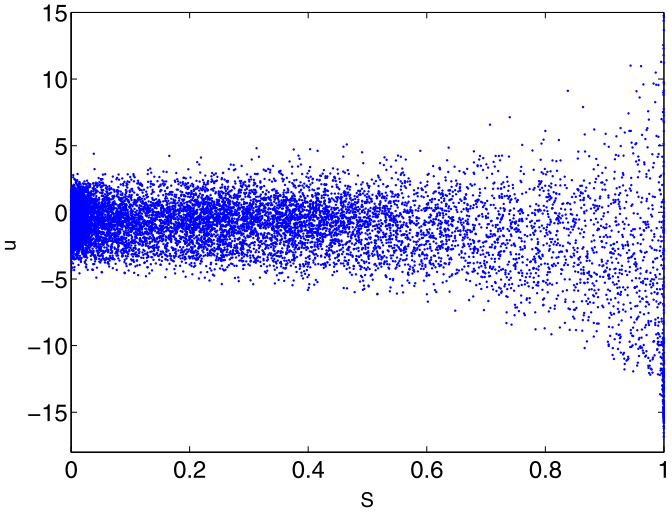

Fig. 2. The plot of the electrostatic potential u against the corresponding hypersurface function S at all nodes where $0 < S < 1$ (PDB ID: 1ajj). Here, u is the filtered explicit Euler solution of the time dependent NPB equation at time $T_c = 10$.

$$- \frac{2\gamma\{S_x\}^n_{i,j,k}\{S_z\}^n_{i,j,k}\{S_{xz}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2} - \frac{2\gamma\{S_y\}^n_{i,j,k}\{S_z\}^n_{i,j,k}\{S_{yz}\}^n_{i,j,k}}{(\{S_x\}^n_{i,j,k})^2 + (\{S_y\}^n_{i,j,k})^2 + (\{S_z\}^n_{i,j,k})^2}$$
$$+ \sqrt{\left(\{S_x\}^n_{i,j,k}\right)^2 + \left(\{S_y\}^n_{i,j,k}\right)^2 + \left(\{S_z\}^n_{i,j,k}\right)^2} V^n_{i,j,k}. \tag{15}$$

Likewise, the explicit Euler discretization of the time dependent NPB equation (12) is given as

$$\begin{aligned} u^{n+1}_{i,j,k} = {} & u^n_{i,j,k} + \Delta t_u c S^n_{i,j,k} Q(x_i, y_j, z_k) - \Delta t_u \left(1 - S^n_{i,j,k}\right)\bar{\kappa}^2 \sinh\left(u^n_{i,j,k}\right) \\ & + \frac{\Delta t_u}{h^2}\left(\epsilon(x_{i+\frac{1}{2}}, y_j, z_k)\left(u^n_{i+1,j,k} - u^n_{i,j,k}\right) + \epsilon(x_{i-\frac{1}{2}}, y_j, z_k)\left(u^n_{i-1,j,k} - u^n_{i,j,k}\right)\right. \\ & + \epsilon(x_i, y_{j+\frac{1}{2}}, z_k)\left(u^n_{i,j+1,k} - u^n_{i,j,k}\right) + \epsilon(x_i, y_{j-\frac{1}{2}}, z_k)\left(u^n_{i,j-1,k} - u^n_{i,j,k}\right) \\ & \left. + \epsilon(x_i, y_j, z_{k+\frac{1}{2}})\left(u^n_{i,j,k+1} - u^n_{i,j,k}\right) + \epsilon(x_i, y_j, z_{k-\frac{1}{2}})\left(u^n_{i,j,k-1} - u^n_{i,j,k}\right)\right) \end{aligned} \tag{16}$$

where $Q(x_i, y_j, z_k)$ is the fractional charge at grid point (x_i, y_j, z_k), which is obtained by using the trilinear interpolation to distribute all charges in the charge density ρ_m. The evaluation of ϵ value on half grid nodes, such as $\epsilon(x_{i+\frac{1}{2}}, y_j, z_k)$, is also obtained via linear interpolation. Numerically, Eq. (16) will be used to update $u^{n+1}_{i,j,k}$ only when $S^n_{i,j,k} < 1$, i.e., when $(x_i, y_j, z_k) \in \Omega_s$. If $S^n_{i,j,k} \geqslant 1$, $u^{n+1}_{i,j,k}$ will be solved by Eq. (16) without imposing the nonlinear term $-\Delta t_u(1 - S^n_{i,j,k})\bar{\kappa}^2 \sinh(u^n_{i,j,k})$.

Two major difficulties are found in the explicit time integration of the pseudo-time coupled nonlinear biomolecular system [32]. The first difficulty is an inefficiency induced by a very small time increment Δt_u in solving u for the coupled solvation system of S and u. In particular, for the potential driven geometric flow equation (14), a large $\Delta t_S = h^2/4.5$ is enough to ensure the stability in solving S. Nevertheless, a very small $\Delta t_u = h^2/585$ has to be used for the time dependent NPB equation (12) for a stable result of u. In other words, by advancing S for one time step, we will update u for 130 time steps. The inefficiency is essentially due to the nonlinear term $\sinh(u)$ in the NPB equation. When the magnitude of u is large, this nonlinear term takes an extremely large value so that Δt_u has to be very small in the explicit time stepping.

Moreover, even with such a small Δt_u, instability issue is experienced in solving the NPB equation [32]. As discussed in [32], such an instability issue is associated with the smooth definition of the solute–solvent interface in the nonlinear solvation model. In particular, the characteristic function for the solvent subdomain is $(1 - S)$, which, unfortunately, is not completely zero in outer region of the macromolecule subdomain. Thus, there could exist certain places which are very close to some partial charges of the macromolecule, but in which $(1 - S)$ values are not zero. If really close to partial charges, the electrostatic potential u will be very large according to Coulomb's law. This is then exponentially amplified by the $\sinh(\cdot)$ function, so that blowing up values are encountered in the biomolecular simulations.

To illustrate this difficulty numerically, we depict the electrostatic potential obtained by the explicit Euler solution of the NPB equation for a protein system (PDB ID: 1ajj) in Fig. 2. In this figure, the electrostatic potential u is plotted against the characteristic function S at all nodes where $0 < S < 1$. It can be seen that for those blowing up nonlinear values, S values are indeed very close to one. A filtering process has to be used to limit the nonlinear contribution of the large u values [32]. Such a stabilizing technique actually alters the physical behavior of the electrostatic potential u near the solute–solvent interface. Thus it introduces some numerical artifacts in the subsequent electrostatic analysis. It is highly desired to design novel numerical schemes which could withstand such a strong nonlinear effect for the biomolecular solvation analysis.

2.4. Operator splitting alternating direction implicit (ADI) schemes

Both aforementioned numerical difficulties are associated with the NPB equation (12). In fact, the bottleneck of the present biomolecular simulation essentially lies in the explicit time integration of the NPB equation. For this reason, we will keep using the standard discretization (15) for the potential driven geometric flow equation (7), while novel operator splitting alternating direction implicit (ADI) schemes will be developed in the present subsection for the NPB equation. In particular, to overcome the strong instability effect of the time dependent NPB equation, we will construct fully implicit time stepping schemes. Nevertheless, at each time step of an implicit time stepping, we need to solve a large linear algebraic system. Thus, we develop some novel ADI schemes for the NPB equation to improve the computational efficiency. To treat the nonlinear term effectively, we will adopt an operator splitting framework so that the nonlinear term can be handled fully implicitly.

For simplicity, we will denote the time increment to be Δt by dropping the subscript u here, because we will concern ourselves to the solution of the electrostatic potential u only in this subsection. A uniform space mesh with a grid spacing h in all x, y, and z directions is also considered. Denote the mesh size in x, y, and z direction to be, respectively, N_x, N_y, and N_z. Denote all nodal values of a function u at time level t_n to be one vector $\mathbf{U}^n = [u^n_{ijk}]^T$, for $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, and $k = 1, \ldots, N_z$. We will develop two ADI schemes for updating $\mathbf{U}^n$ at a time level t_n to $\mathbf{U}^{n+1}$ at the next time level $t_{n+1} = t_n + \Delta t$. In the NPB equation (12), ρ_m is determined by the molecular structure and thus is time independent. Without the loss of generality, we will assume the hypersurface function S and the dielectric function ϵ are independent of time in the course of the time integration of u.

2.4.1. ADI scheme 1

In the ADI scheme 1 or ADI1, at each time step from t_n to t_{n+1} and when $S(x, y, z) < 1$, the time dependent NPB equation (12) will be solved by a first order time splitting method in two stages [30]

$$\frac{\partial w}{\partial t} = -(1 - S)\bar{\kappa}^2 \sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n,\ t \in [t_n, t_{n+1}], \tag{17}$$

$$\frac{\partial v}{\partial t} = \nabla \cdot \big(\epsilon(S)\nabla v\big) + cS\rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{W}^{n+1},\ t \in [t_n, t_{n+1}]. \tag{18}$$

We then have $\mathbf{U}^{n+1} = \mathbf{V}^{n+1}$. When $S(x, y, z) \geqslant 1$, one only needs to solve (18).

For the first stage, we note that Eq. (17) is actually a separable ordinary differential equation in the present setting

$$\frac{dw}{\sinh(w)} = -(1 - S)\bar{\kappa}^2\, dt.$$

Thus, the integration can be carried out analytically

$$\int \frac{1}{\sinh(w)}\, dw = \int -(1 - S)\bar{\kappa}^2\, dt,$$

$$-2\tanh^{-1}\big(e^w\big) = -(1 - S)\bar{\kappa}^2 t + C. \tag{19}$$

Evaluating Eq. (19) at both t_n and t_{n+1}, we have

$$\tanh^{-1}\big(\exp(\mathbf{W}^{n+1})\big) - \tanh^{-1}\big(\exp(\mathbf{W}^n)\big) = \frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t.$$

Taking $\tanh(\cdot)$ in both-hand sides and re-arranging the terms, we have

$$\exp(\mathbf{W}^{n+1}) = \tanh\left(\tanh^{-1}\big(\exp(\mathbf{W}^n)\big) + \frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t\right). \tag{20}$$

Eq. (20) can be greatly simplified to be

$$\mathbf{W}^{n+1} = \ln\left(\frac{\cosh(\frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t) + \exp(-\mathbf{W}^n)\sinh(\frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t)}{\exp(-\mathbf{W}^n)\cosh(\frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t) + \sinh(\frac{1}{2}(1 - S)\bar{\kappa}^2\Delta t)}\right) := F\big(\mathbf{W}^n; S, \Delta t\big). \tag{21}$$

In other words, with $\mathbf{W}^n$ at t_n, $\mathbf{W}^{n+1}$ can be calculated analytically according to (21) so that various difficulties associated with the $\sinh(\cdot)$ nonlinear term of the NPB equation are simply bypassed. Here, to facilitate the following discussions, we have introduced an updating function $F(\mathbf{W}; S, \Delta t)$, which is a function of $\mathbf{W}$ and depends on the parameters S and Δt as well.

A Douglas–Rachford type ADI scheme is proposed to solve the nonhomogeneous diffusion equation (18) in the second stage. The discretization of (18) using the backward-Euler integration in time and central differencing in space results in

$$v^{n+1}_{i,j,k} = v^n_{i,j,k} + \Delta t\big(\delta_x^2 + \delta_y^2 + \delta_z^2\big)v^{n+1}_{i,j,k} + \Delta t c S\rho_m, \tag{22}$$

where δ_x^2, δ_y^2, and δ_z^2 are the central difference operators in the x, y, and z directions, respectively,

$$\delta_x^2 v_{i,j,k}^n = \frac{1}{h^2}\left(\epsilon(x_{i+\frac{1}{2}}, y_j, z_k)\left(v_{i+1,j,k}^n - v_{i,j,k}^n\right) + \epsilon(x_{i-\frac{1}{2}}, y_j, z_k)\left(v_{i-1,j,k}^n - v_{i,j,k}^n\right)\right),$$
$$\delta_y^2 v_{i,j,k}^n = \frac{1}{h^2}\left(\epsilon(x_i, y_{j+\frac{1}{2}}, z_k)\left(v_{i,j+1,k}^n - v_{i,j,k}^n\right) + \epsilon(x_i, y_{j-\frac{1}{2}}, z_k)\left(v_{i,j-1,k}^n - v_{i,j,k}^n\right)\right),$$
$$\delta_z^2 v_{i,j,k}^n = \frac{1}{h^2}\left(\epsilon(x_i, y_j, z_{k+\frac{1}{2}})\left(v_{i,j,k+1}^n - v_{i,j,k}^n\right) + \epsilon(x_i, y_j, z_{k-\frac{1}{2}})\left(v_{i,j,k-1}^n - v_{i,j,k}^n\right)\right).$$

The Douglas–Rachford ADI scheme for (18) is formulated as [27]

$$\begin{aligned}
\left(1 - \Delta t\delta_x^2\right)v_{i,j,k}^* &= \left[1 + \Delta t\left(\delta_y^2 + \delta_z^2\right)\right]v_{i,j,k}^n + \Delta tcS\rho_m,\\
\left(1 - \Delta t\delta_y^2\right)v_{i,j,k}^{**} &= v_{i,j,k}^* - \Delta t\delta_y^2 v_{i,j,k}^n,\\
\left(1 - \Delta t\delta_z^2\right)v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \Delta t\delta_z^2 v_{i,j,k}^n.
\end{aligned} \tag{23}$$

By eliminating $v_{i,j,k}^*$ and $v_{i,j,k}^{**}$, we have

$$\left(1 - \Delta t\delta_x^2\right)\left(1 - \Delta t\delta_y^2\right)\left(1 - \Delta t\delta_z^2\right)v_{i,j,k}^{n+1} = \left[1 + \Delta t^2\left(\delta_x^2\delta_y^2 + \delta_y^2\delta_z^2 + \delta_x^2\delta_z^2\right) - \Delta t^3\delta_x^2\delta_y^2\delta_z^2\right]v_{i,j,k}^n + \Delta tcS\rho_m. \tag{24}$$

After fully expanding terms, Eq. (24) can be written in the form

$$\begin{aligned}
v_{i,j,k}^{n+1} = {} & v_{i,j,k}^n + \Delta t\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right)v_{i,j,k}^{n+1} + \Delta tcS\rho_m\\
& - \Delta t^2\left(\delta_x^2\delta_y^2 + \delta_y^2\delta_z^2 + \delta_x^2\delta_z^2\right)\left(v_{i,j,k}^{n+1} - v_{i,j,k}^n\right) + \Delta t^3\delta_x^2\delta_y^2\delta_z^2\left(v_{i,j,k}^{n+1} - v_{i,j,k}^n\right).
\end{aligned} \tag{25}$$

In comparison with Eq. (22), it is clear that the Douglas–Rachford ADI scheme is a higher order perturbation of the backward Euler scheme in three space variables. Thus, the Douglas–Rachford scheme (23) is of first order of accuracy in time, which is consistent with the overall first order operator splitting formulation. Therefore, the temporal order of the proposed ADI1 scheme is one. Since the Dirichlet boundary condition (13) is time independent, the boundary values for all intermediate variables v, v^*, and v^{**} can be simply taken according to (13). On the other hand, because the dielectric profile $\epsilon(S)$ is smooth owing to the smooth definition of solute–solvent interface, the proposed ADI1 scheme is of second order of accuracy in space. Since each of the steps in Eq. (23) is basically a one-dimensional backward Euler scheme, the Douglas–Rachford scheme is unconditionally stable [27]. Combining this with the analytical integration in the first stage, the proposed ADI1 scheme is unconditionally stable in solving the time dependent NPB equation. We also note that equations in (23) correspond to tridiagonal systems, and thus can be solved efficiently by the Thomas algorithm [27,4].

2.4.2. ADI scheme 2

In the ADI scheme 2 or ADI2, at each time step from t_n to t_{n+1} and when $S(x, y, z) < 1$, the time dependent NPB equation (12) will be solved by a second order time splitting method in three stages [30]

$$\frac{\partial w}{\partial t} = -\frac{1}{2}(1 - S)\bar{\kappa}^2\sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n,\ t \in [t_n, t_{n+1}], \tag{26}$$
$$\frac{\partial v}{\partial t} = \nabla\cdot\left(\epsilon(S)\nabla v\right) + cS\rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{W}^{n+1},\ t \in [t_n, t_{n+1}], \tag{27}$$
$$\frac{\partial \tilde{w}}{\partial t} = -\frac{1}{2}(1 - S)\bar{\kappa}^2\sinh(\tilde{w}), \quad \text{with } \tilde{\mathbf{W}}^n = \mathbf{V}^{n+1},\ t \in [t_n, t_{n+1}]. \tag{28}$$

We then have $\mathbf{U}^{n+1} = \tilde{\mathbf{W}}^{n+1}$. When $S(x, y, z) \geqslant 1$, one only needs to solve (27).

The present time splitting method is also known as the Strang splitting [30]. Other high order time splitting methods can be generated in a systematic way [30], but will not be considered in this paper. As in the ADI1 scheme, in the first and last stage of the ADI2 scheme, an analytical integration can be conducted. Symbolically, we have $\mathbf{W}^{n+1} = F(\mathbf{W}^n; S, \frac{\Delta t}{2})$ and $\tilde{\mathbf{W}}^{n+1} = F(\tilde{\mathbf{W}}^n; S, \frac{\Delta t}{2})$, where F is defined in Eq. (21).

A Douglas type ADI scheme is proposed to solve the nonhomogeneous diffusion equation (27) in the second stage. The discretization of (27) using the Crank–Nicolson integration in time and central differencing in space results in

$$\left[1 - \frac{\Delta t}{2}\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right)\right]v_{i,j,k}^{n+1} = \left[1 + \frac{\Delta t}{2}\left(\delta_x^2 + \delta_y^2 + \delta_z^2\right)\right]v_{i,j,k}^n + \Delta tcS\rho_m. \tag{29}$$

The Douglas ADI scheme for (27) is formulated as [27]

$$\left(1-\frac{\Delta t}{2}\delta_x^2\right)v_{i,j,k}^{*}=\left[1+\frac{\Delta t}{2}\left(\delta_x^2+2\delta_y^2+2\delta_z^2\right)\right]v_{i,j,k}^{n}+\Delta tcS\rho_m,$$
$$\left(1-\frac{\Delta t}{2}\delta_y^2\right)v_{i,j,k}^{**}=v_{i,j,k}^{*}-\frac{\Delta t}{2}\delta_y^2 v_{i,j,k}^{n},$$
$$\left(1-\frac{\Delta t}{2}\delta_z^2\right)v_{i,j,k}^{n+1}=v_{i,j,k}^{**}-\frac{\Delta t}{2}\delta_z^2 v_{i,j,k}^{n}. \tag{30}$$

After the elimination of $v_{i,j,k}^{*}$ and $v_{i,j,k}^{**}$, this leads to

$$\left(1-\frac{\Delta t}{2}\delta_x^2\right)\left(1-\frac{\Delta t}{2}\delta_y^2\right)\left(1-\frac{\Delta t}{2}\delta_z^2\right)v_{i,j,k}^{n+1}$$
$$=\left[1+\frac{\Delta t}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)+\frac{\Delta t^2}{4}\left(\delta_x^2\delta_y^2+\delta_y^2\delta_z^2+\delta_x^2\delta_z^2\right)-\frac{\Delta t^3}{8}\delta_x^2\delta_y^2\delta_z^2\right]v_{i,j,k}^{n}+\Delta tcS\rho_m. \tag{31}$$

This can be further written in the form

$$\left[1-\frac{\Delta t}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)\right]v_{i,j,k}^{n+1}=\left[1+\frac{\Delta t}{2}\left(\delta_x^2+\delta_y^2+\delta_z^2\right)\right]v_{i,j,k}^{n}+\Delta tcS\rho_m$$
$$-\frac{\Delta t^2}{4}\left(\delta_x^2\delta_y^2+\delta_y^2\delta_z^2+\delta_x^2\delta_z^2\right)\left(v_{i,j,k}^{n+1}-v_{i,j,k}^{n}\right)+\frac{\Delta t^3}{8}\delta_x^2\delta_y^2\delta_z^2\left(v_{i,j,k}^{n+1}-v_{i,j,k}^{n}\right). \tag{32}$$

It is easy to show that the resulting equation is a higher order perturbation of the Crank–Nicolson scheme (29) in three space variables. Thus, the temporal order of accuracy of the Douglas scheme (30) is two, which is the same as the Strang time splitting method. Consequently, the proposed ADI2 scheme is of second order of accuracy in both time and space. Here, the Dirichlet boundary condition (13) is also used for intermediate variables v, v^{*}, and v^{**}. The Douglas scheme is also unconditionally stable, because in each alternating direction, we have basically a one-dimensional Crank–Nicolson scheme [27]. Thus, the overall ADI2 scheme is also unconditionally stable in solving the time dependent NPB equation. By using the Thomas algorithm, the tridiagonal systems are also efficiently solved in the ADI2 scheme. In terms of the computational complexity of one cycle, the ADI2 scheme involves a larger flops count than that of the ADI1 scheme. Nevertheless, the ADI2 scheme is usually more accurate than the ADI1 scheme.

3. Numerical validations

In this section, we validate the proposed time splitting ADI schemes by considering constructed examples with analytical solutions. Both time dependent and time independent NPB equations will be studied for investigating the proposed numerical algorithms and pseudo-transient approximations. All simulations are compiled with an Intel Linux Fortran compiler and run on a Dell precision 690n with four dual-core Intel Xeon processors.

3.1. Time dependent nonlinear Poisson–Boltzmann equation

We first study a time dependent NPB equation with a smooth solute–solvent interface

$$\frac{\partial u}{\partial t}=\nabla\cdot(\epsilon\nabla u)-(1-S)\bar{\kappa}^2\sinh(u)+f. \tag{33}$$

A benchmark example with an analytical solution will be constructed for a simple one-ball system [8]. Using the analytical solution, we can numerically examine the stability, accuracy, and convergence of the proposed operator splitting ADI schemes. The molecular surface is assumed to be independent of the time and determined by the following hypersurface function [8]:

$$S(x,y,z)=\begin{cases}1 & \text{if } r<a,\\ -2(\frac{b-r}{b-a})^3+3(\frac{b-r}{b-a})^2 & \text{if } a\leqslant r\leqslant b,\\ 0 & \text{if } b<r,\end{cases} \tag{34}$$

where $r=\sqrt{x^2+y^2+z^2}$ and a and b are two constants with $a<b$. The time invariant dielectric profile is given as

$$\epsilon(x,y,z)=\epsilon_1 S(x,y,z)+\epsilon_2\big(1-S(x,y,z)\big), \tag{35}$$

with ϵ_1 and ϵ_2 being two dielectric coefficients. We note that the present definition of S takes a shape which is similar to that of a particular solution of our potential driven geometric flow equation. Mathematically, both $S(x,y,z)$ and $\epsilon(x,y,z)$ are C^1 but not C^2 continuous across the spheres $r=a$ and $r=b$.

In the present study, the analytical solution is time dependent

$$u(x,y,z,t)=\sin(Cx)\sin(Cy)\sin(Cz)\big(1+\exp(-Dt)\big), \tag{36}$$

where C and D are two constants. Based on the analytical solution (36), the source term $f(x,y,z,t)$ can be determined according to (33)

$$\begin{aligned} f=&(1-S)\bar{\kappa}^2\sinh\big(\sin(Cx)\sin(Cy)\sin(Cz)\big(1+\exp(-Dt)\big)\big)-D\sin(Cx)\sin(Cy)\sin(Cz)\exp(-Dt)\\ &-C\epsilon_x\cos(Cx)\sin(Cy)\sin(Cz)\big(1+\exp(-Dt)\big)-C\epsilon_y\sin(Cx)\cos(Cy)\sin(Cz)\big(1+\exp(-Dt)\big)\\ &-C\epsilon_z\sin(Cx)\sin(Cy)\cos(Cz)\big(1+\exp(-Dt)\big)+3C^2\epsilon\sin(Cx)\sin(Cy)\sin(Cz)\big(1+\exp(-Dt)\big). \end{aligned}$$

Here, the derivatives of $\epsilon(x,y,z)$ are given as

$$\begin{aligned} \epsilon_x&=6(\epsilon_2-\epsilon_1)\left(\frac{b-r}{b-a}\right)\left(\frac{r-a}{b-a}\right)\left(\frac{x}{r(b-a)}\right),\quad \text{for } a\leqslant r\leqslant b;\ \epsilon_x=0,\ \text{otherwise},\\ \epsilon_y&=6(\epsilon_2-\epsilon_1)\left(\frac{b-r}{b-a}\right)\left(\frac{r-a}{b-a}\right)\left(\frac{y}{r(b-a)}\right),\quad \text{for } a\leqslant r\leqslant b;\ \epsilon_y=0,\ \text{otherwise},\\ \epsilon_z&=6(\epsilon_2-\epsilon_1)\left(\frac{b-r}{b-a}\right)\left(\frac{r-a}{b-a}\right)\left(\frac{z}{r(b-a)}\right),\quad \text{for } a\leqslant r\leqslant b;\ \epsilon_z=0,\ \text{otherwise}. \end{aligned} \tag{37}$$

In the present study, we choose $a=1$ and $b=3$ in the hypersurface function $S(x,y,z)$ and $C=\frac{\pi}{4}$ and $D=\frac{1}{10}$ in the analytical solution $u(x,y,z,t)$. A cubic domain $[-4,4]\times[-4,4]\times[-4,4]$ is chosen such that a Dirichlet boundary condition $u=0$ is assumed on all boundaries. The model parameters are chosen as $\bar{\kappa}=1$, $\epsilon_1=1$ and $\epsilon_2=80$. By taking the initial solution $u(x,y,z,0)$ to be the analytical solution (36) at time $t=0$, the time dependent NLB equation (33) will be integrated by both ADI schemes until $t=10$.

We first examine the temporal accuracy and stability of both ADI schemes. By choosing $h=1$, $h=0.5$, $h=0.25$, and $h=0.125$, numerical errors of two ADI schemes are depicted in Fig. 3 against the time increment Δt. A similar pattern can be observed in all four cases with different spatial resolutions, i.e., the errors approach to constant levels when Δt is small or large enough. On the one end, when Δt becomes larger and larger, both L_∞ and L_2 errors of two ADI schemes approach to constant levels. In particular, the constant levels of L_∞ errors are almost the same in all four cases, while the constant levels of L_2 errors become slightly larger when h is smaller. This means that, by using an extremely large Δt, the ADI simulation produces inaccurate results without surprise. However, it is interesting to note that the computation remains to be unconditionally stable for all h values. On the other end, when Δt becomes smaller and smaller, both L_∞ and L_2 errors of two ADI schemes are limited by the accuracies of the spatial discretization. The same central finite difference discretization is employed in both ADI schemes. Thus, both ADI schemes approach to the same constant level as $\Delta t\to 0$ in all cases. We also note that the limiting precisions in L_∞ and L_2 become smaller whenever the spatial mesh is refined.

When a moderate Δt is used, it can be seen from Fig. 3 that the errors of the ADI2 are usually smaller than those of the ADI1 based on the same mesh. This is because theoretically the temporal order of the ADI2 is two, whereas that of the ADI1 is one. Nevertheless, this accuracy gain comes with a price. By using the same Δt and h, the CPU time of the ADI2 scheme is typically longer than that of the ADI1 scheme. Moreover, the numerically detected temporal order of convergence of the ADI1 scheme seems to be quite high in Fig. 3. In all cases, in comparison with the ADI2 scheme, the ADI1 scheme begins to converge at a smaller Δt and approaches the limiting precision at a smaller Δt. However, in the converging range of Δt, the slopes of error curves of the ADI1 scheme are comparable to those of the ADI2 scheme.

We next quantitatively investigate the spatial order of accuracy of both ADI schemes. By taking $\Delta t=0.00125$, the L_∞ and L_2 errors of two ADI schemes are reported in Table 1. As shown in Fig. 3, the present temporal error is small enough such that the approximation error is mainly produced by the spatial discretization. By examining the errors in successive mesh refinements, the numerically calculated rates of convergence are also listed in Table 1. We note that the accuracies and orders displayed in both ADI schemes are almost identical, because the same finite difference discretization is utilized in both schemes. In particular, the spatial order in the L_2 norm is clearly second order, while that in the L_∞ norm is about 1.5th order. The minor order reduction of L_∞ errors is believed to be due to the C^1 continuous feature of dielectric profile. Actually, the maximum errors are usually present around either $r=a$ or $r=b$.

We then quantitatively investigate the temporal order of accuracy of both ADI schemes. By taking $h=0.125$, L_∞ and L_2 errors of two ADI schemes are reported in Table 2. We note that all errors presented in Table 2, except those of the ADI2 with $\Delta t=0.005$, are larger than the limiting precisions of $h=0.125$ shown in Table 1. In other words, in such cases, the approximation error is dominated by the temporal discretization. This enable us to check the temporal orders. It can be seen from Table 2 that both ADI schemes deliver a second order of convergence in time for both L_∞ and L_2 errors. The ADI2 scheme starts to converge earlier. Thus, the numerical orders detected at $\Delta t=0.04$ are close to two for the ADI2, while those of the ADI1 are still away from two. Meanwhile, the ADI2 scheme reaches the limiting precisions earlier. Hence, the numerical orders of the ADI2 for $\Delta t=0.005$ are polluted, while those of the ADI1 remain to be two.

In Tables 1 and 2, the computation time consumed by the ADI simulations is also reported. It can be observed that the ADI2 usually spends about 25%–35% more CPU time than the ADI1, for a fixed h and Δt. When h is fixed, the CPU time

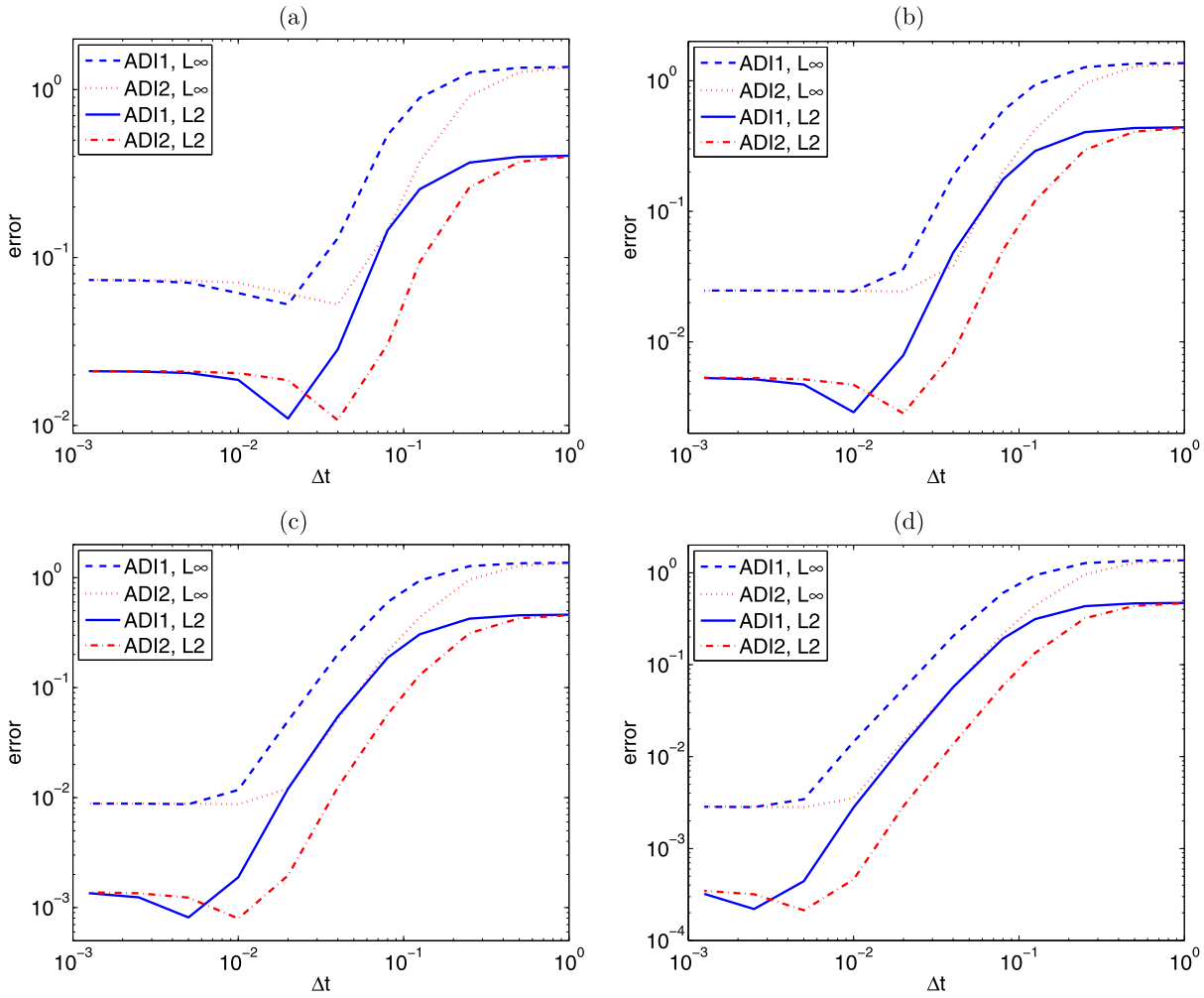

Fig. 3. Numerical errors against Δt in solving the time dependent nonlinear Poisson–Boltzmann equation. (a) $h = 1$; (b) $h = 0.5$; (c) $h = 0.25$; (d) $h = 0.125$.

Table 1
Spatial order of convergence in solving the time dependent nonlinear Poisson–Boltzmann equation. Here Δt is fixed to be 0.00125. CPU time in second is reported.

h	ADI1 scheme					ADI2 scheme				
	L_∞	Order	L_2	Order	CPU	L_∞	Order	L_2	Order	CPU
1	7.35E−2		2.10E−2		0.73	7.36E−2		2.11E−2		0.97
0.5	2.48E−2	1.57	5.29E−3	1.99	6.84	2.48E−2	1.57	5.31E−3	1.99	9.18
0.25	8.83E−3	1.49	1.35E−3	1.97	59.85	8.84E−3	1.49	1.38E−3	1.95	79.72
0.125	2.85E−3	1.63	3.22E−4	2.07	646.87	2.86E−3	1.63	3.47E−4	1.99	849.22

Table 2
Temporal order of convergence in solving the time dependent nonlinear Poisson–Boltzmann equation. Here h is fixed to be 0.125. CPU time in second is reported.

Δt	ADI1 scheme					ADI2 scheme				
	L_∞	Order	L_2	Order	CPU	L_∞	Order	L_2	Order	CPU
0.08	6.04E−1		1.93E−1		10.15	2.17E−1		5.95E−2		12.70
0.04	2.05E−1	1.56	5.70E−2	1.76	20.16	5.63E−2	1.95	1.37E−2	2.12	27.13
0.02	5.45E−2	1.91	1.33E−2	2.10	40.79	1.49E−2	1.92	2.89E−3	2.24	54.59
0.01	1.45E−2	1.91	2.80E−3	2.25	84.92	3.53E−3	2.08	4.63E−4	2.64	107.95
0.005	3.43E−3	2.08	4.42E−4	2.67	168.38	2.82E−3	0.32	2.13E−4	1.12	212.98

needed by both ADI1 and ADI2 is almost doubled when Δt is halved. This means that the complexity of the ADI schemes scales linearly with respect to the total time steps. When Δt is fixed, the CPU time of both ADI schemes becomes about 10 times larger when h is halved. Thus, the complexity of the ADI scheme with respect to the total spatial unknowns is slightly

higher than that of the explicit schemes, but is very favorable for three-dimensional implicit time integrations, thanks to the Thomas algorithm.

3.2. Nonlinear Poisson–Boltzmann equation

We next study a time independent or regular NPB equation

$$-\nabla\cdot(\epsilon\nabla u)+(1-S)\bar{\kappa}^2\sinh(u)=f. \tag{38}$$

This study enables us to design a realistic stopping criterion for real biomolecular solvation simulations. The same smooth solute–solvent hypersurface function $S(x,y,z)$ and dielectric profile $\epsilon(x,y,z)$ as in (34) and (35) are employed again. The analytical solution to (38) is time independent

$$u(x,y,z)=\sin(Cx)\sin(Cy)\sin(Cz) \tag{39}$$

with only one parameter C. The source term $f(x,y,z)$ is also constructed by solving the governing equation (38)

$$\begin{aligned} f=&(1-S)\bar{\kappa}^2\sinh\big(\sin(Cx)\sin(Cy)\sin(Cz)\big)+3C^2\epsilon\sin(Cx)\sin(Cy)\sin(Cz)\\ &-C\epsilon_x\cos(Cx)\sin(Cy)\sin(Cz)-C\epsilon_y\sin(Cx)\cos(Cy)\sin(Cz)-C\epsilon_z\sin(Cx)\sin(Cy)\cos(Cz), \end{aligned}$$

where the derivatives of ϵ are given by Eq. (37). The same parameter values are employed, i.e., $a=1$, $b=3$, $C=\frac{\pi}{4}$, $\bar{\kappa}=1$, $\epsilon_1=1$, and $\epsilon_2=80$. The domain is still $[-4,4]\times[-4,4]\times[-4,4]$ so that the zero Dirichlet boundary condition still holds on all boundaries.

In the present study, the time independent NPB equation (38) is solved by considering a pseudo-transient continuation approximation, i.e., by solving time dependent NPB equation (33) until the steady state. Here, a trivial initial solution $u=0$ is employed. Then, the time dependent NPB equation (33) will be iteratively solved by the time splitting ADI schemes until certain convergence criterion is satisfied.

Since the analytical solution is available, we first consider a convergence criterion based on the actual error. In particular, the computation will stop when the change in the L_∞ errors of two consecutive time steps is less than a given tolerance. This stopping scheme is referred to as *Convergence Criterion 1* in the following. The tolerance is chosen to be $\mathrm{TOL}=10^{-7}$, which is guaranteed to be smaller than all L_∞ errors involved in the present ADI studies.

Numerical errors of two ADI schemes for solving the time independent NPB equation based on the Convergence Criterion 1 are shown as solid and broken lines in Fig. 4. Similar patterns as in the time dependent NPB equation studies can be observed. In particular, the ADI errors based on different mesh size h are almost the same when $\Delta t=1$. When Δt becomes sufficiently small, both ADI schemes produce the same limiting precisions and such precisions are different when h is different. For moderate Δt values, two ADI schemes again generate parallel convergence curves, even though the ADI2 is more accurate based on a fixed Δt.

Quantitative studies can be similarly conducted to analyze the order of convergence of two ADI schemes using the Convergence Criterion 1. By using $\Delta t=0.00125$, the spatial orders of the finite difference discretization are examined in Table 3. It is clear that the spatial order in the L_2 norm is of second order, while that in the L_∞ norm is about 1.5th order. By using $h=0.125$, the temporal orders of both ADI schemes are investigated in Table 4. It is clear that both ADI schemes attain a second order of accuracy in time for both error norms.

The present numerical orders based on the Convergence Criterion 1 are very similar to those of the previous example, even though a regular NPB equation is solved. To explain this phenomenon, we need to distinguish two types of approximation errors in solving the time independent NPB equation by the present pseudo-transient approach. The first one is the difference between the solution of NPB equation (38) and the converged solution of time dependent NPB equation (33), which depends on the convergence criterion. Such an error can be regarded as a modeling error in the context of biomolecular solvation analysis. The other one is due to the temporal and spatial discretizations of the ADI schemes and finite difference methods in solving (33). In the present study, the Convergence Criterion 1 is checked based on the actual L_∞ error between the analytical solution of the time independent NPB equation and the numerical solution to the time dependent NPB equation. The fact that the Convergence Criterion 1 is satisfied with a small enough tolerance in all cases demonstrates the effectiveness and robustness of the present pseudo-transient continuation approximation. Moreover, the Convergence Criterion 1 also guarantees that the modeling error is sufficiently small so that the present results are primarily determined by the discretization errors.

CPU times are also reported in Tables 3 and 4. Comparing with the CPU results in solving the time dependent NPB equation, it is interesting to note two differences. First, the CPU time of the ADI2 is no longer always larger than that of the ADI1. Second, when Δt becomes smaller with a fixed h, the computation could become faster. Both interesting features are due to the earlier satisfaction of the Convergence Criterion 1 in some cases.

The new features of the CPU results motivate us to further examine the number of iterations in all ADI computations of the NPB equation (38). Denote the number of iterations to be N_i in the present study. The plots of N_i against Δt for four h values are shown in Fig. 5. It can be observed from the figure that both ADI1 and ADI2 schemes yield a zigzag curve when Δt changes. Such a shape is better to be interpreted in connection with the numerical convergence patterns shown

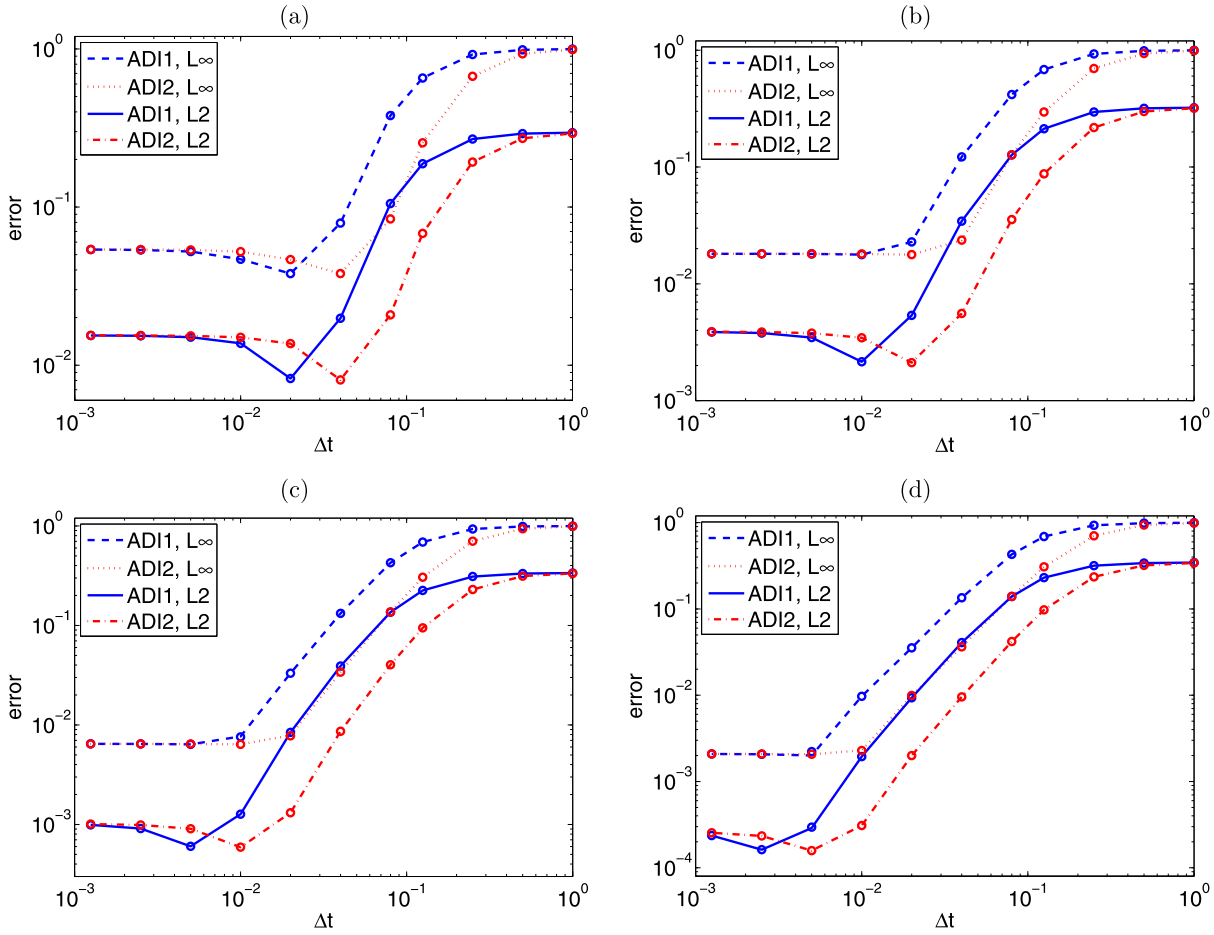

Fig. 4. Numerical errors against Δt in solving the time independent or regular nonlinear Poisson–Boltzmann equation. (a) $h = 1$; (b) $h = 0.5$; (c) $h = 0.25$; (d) $h = 0.125$. In all cases, the errors of the Convergence Criterion 1 are shown as solid or broken lines, while the errors of the Convergence Criterion 2 are plotted as open circles.

Table 3
Spatial order of convergence in solving the time independent or regular nonlinear Poisson–Boltzmann equation. Here Δt is fixed to be 0.00125 and the Convergence Criterion 1 is used. CPU time in second is reported.

h	ADI1 scheme					ADI2 scheme				
	L_∞	Order	L_2	Order	CPU	L_∞	Order	L_2	Order	CPU
1	5.39E−2		1.54E−2		0.007	5.39E−2		1.54E−2		0.008
0.5	1.81E−2	1.57	3.87E−3	1.99	0.20	1.81E−2	1.58	3.89E−3	1.99	0.26
0.25	6.45E−3	1.49	9.90E−4	1.97	1.60	6.46E−3	1.49	1.01E−3	1.95	2.17
0.125	2.08E−3	1.63	2.36E−4	2.07	18.14	2.09E−3	1.63	2.54E−4	1.99	24.13

Table 4
Temporal order of convergence in solving the time independent or regular nonlinear Poisson–Boltzmann equation. Here h is fixed to be 0.125 and the Convergence Criterion 1 is used. CPU time in second is reported.

Δt	ADI1 scheme					ADI2 scheme				
	L_∞	Order	L_2	Order	CPU	L_∞	Order	L_2	Order	CPU
0.08	4.29E−1		1.40E−1		8.50	1.39E−1		4.20E−2		8.83
0.04	1.35E−1	1.67	4.08E−2	1.77	12.28	3.64E−2	1.94	9.56E−3	2.13	10.66
0.02	3.54E−2	1.93	9.34E−3	2.13	15.37	9.91E−3	1.84	2.00E−3	2.23	18.91
0.01	9.71E−3	1.87	1.95E−3	2.26	24.59	2.29E−3	2.11	3.09E−4	2.66	21.80
0.005	2.00E−3	2.28	2.89E−4	2.75	5.53	2.07E−3	0.15	1.58E−4	0.97	6.85

in Fig. 4. In particular, when Δt is quite large, the numerical errors are almost constant for both ADI schemes. In such a case, we found that N_i is almost doubled when Δt is halved. Thus, a quasi-linear pattern is presented in the plot of the iteration number against Δt. Such a pattern can be roughly seen in Fig. 5, albeit additional attention has to be paid because a logarithmic scale is used for Δt. If the absolute stopping time, i.e., $N_i \Delta t$, was reported, we could see that the convergence is reached almost at a constant time level for sufficiently large Δt values. When Δt becomes smaller, the numerical errors

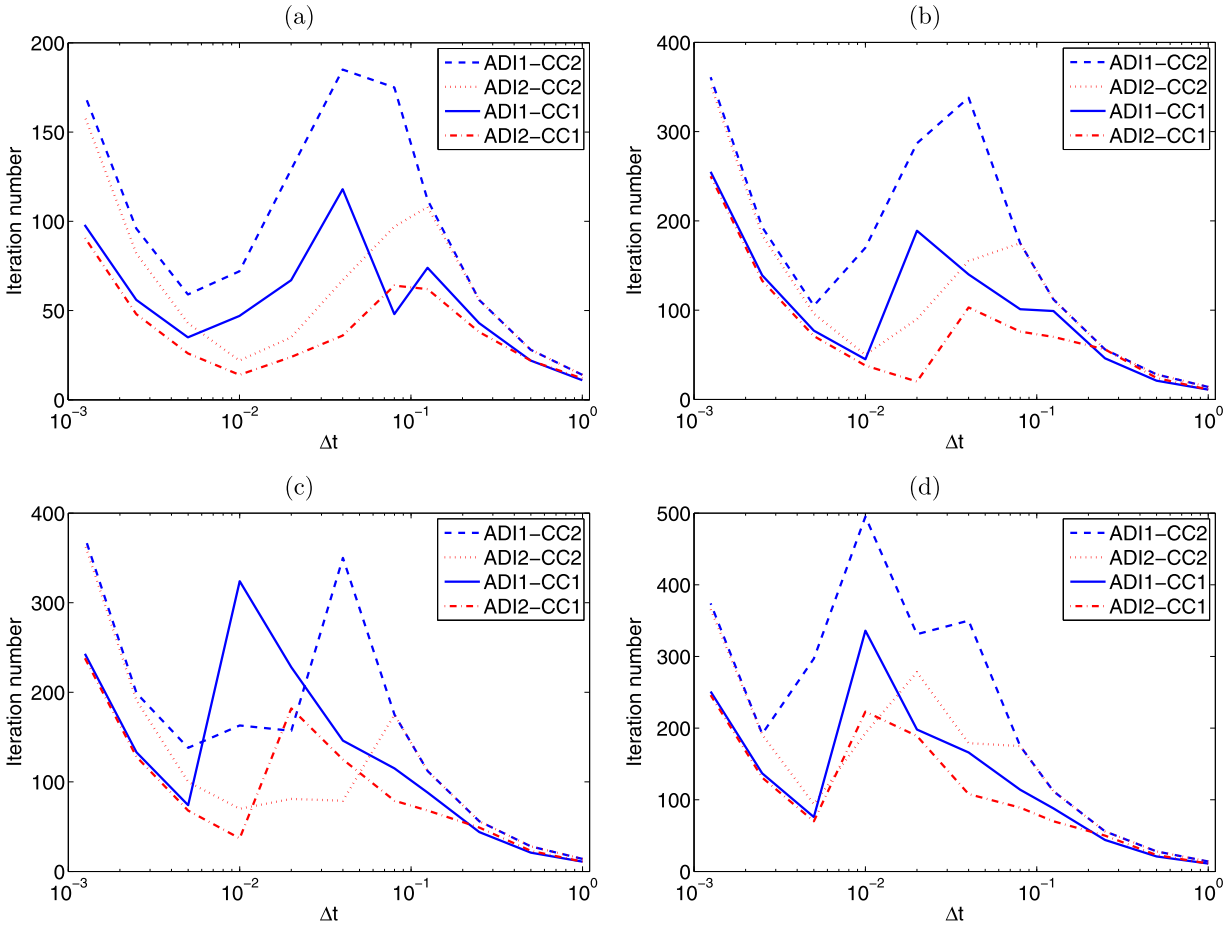

Fig. 5. Iteration numbers against Δt in solving the time independent or regular nonlinear Poisson–Boltzmann equation. (a) $h = 1$; (b) $h = 0.5$; (c) $h = 0.25$; (d) $h = 0.125$. In all charts, CC1 and CC2 denotes Convergence Criterion 1 and Convergence Criterion 2, respectively.

start to decline, as can be seen in Fig. 4. For such moderate Δt values, the number of iterations keeps to increase for a while and then suddenly drops, when Δt gets smaller and smaller. Another tuning point appears in all charts of Fig. 5 when the Δt is sufficiently small. When Δt is even smaller, a quasi-linear pattern occurs again with N_i almost doubled when one halves Δt values. We note that during this second period of the quasi-linear pattern, the ADI errors are constants again due to the reach of the limiting precisions.

Out of two tunning points in a plot of the number of iterations against Δt, we are more interested in the second one. We note that if one excludes large Δt values which are surely inaccurate, the number of iterations achieves the global minimum at this turning point. Computationally, an ADI scheme takes the shortest CPU time by using the corresponding Δt. Moreover, at this global minimum, the limiting precision of the ADI scheme is almost reached. In other words, when one further reduces Δt value, the accuracy almost remains unchanged. Therefore, the Δt value corresponding to this global minimum delivers the best numerical performance in terms of both speed and accuracy. The present study indicates that one may identify this desired Δt value by searching for the minimal number of iterations in a practical biomolecular simulation.

However, in real biomolecular simulations, the analytical solutions are not available, so that the Convergence Criterion 1 is impossible to check. A practical stopping scheme should depend on the numerical solution itself, instead of the corresponding numerical error. We thus consider next the *Convergence Criterion 2*, in which the computation will stop when the change in the L_2 norm of the numerical solutions of two consecutive time steps is less than a given tolerance. We note that in real biomolecular solvation computations, numerical iterations are usually terminated if the change in the solvation free energy is small enough. Defined based an integral over the entire domain, in some sense, the solvation free energy can be regarded as a weighted average value of the electrostatic potential. This actually motivates us to choose the L_2 norm of the numerical solution as the evaluation quantity, since it is also an averaged quantity. In the present study, a very small tolerance $\mathrm{TOL} = 10^{-12}$ is employed to compare with the difference in the L_2 norms. Such a tolerance is found to be able to reproduce almost the same results as in the Convergence Criterion 1 for small enough Δt values.

Nevertheless, when Δt is pretty large, the difference between the solutions of two consecutive time steps is always large. Thus, by no means, such a difference could be less than 10^{-12}. Therefore, one has to consider an additional stopping scheme for large Δt values in the Convergence Criterion 2. Based on our previous investigations, we know that when Δt is large, the absolute stopping time $N_i \Delta t$ in the Convergence Criterion 1 is almost a constant. This motivates us to choose the absolute stopping time as the another criterion. In particular, we define a stopping time $T_u = 14$, which is larger than

any absolute stopping time of the Convergence Criterion 1 and impose a stopping mechanism: the computation will stop if $t \geqslant T_u$. Therefore, in the Convergence Criterion 2, the simulation terminates when either $t \geqslant T_u$ or the difference in the L_2 norms less than 10^{-12}. It is interesting to note that based on trials and errors, the stopping scheme in the previous biomolecular solvation simulations also depends on an absolute stopping time and a tolerance for the solvation free energy [32,7,8]. The present numerical studies justify the previous empirical designs.

We next investigate the numerical performance of the proposed Convergence Criterion 2. The numerical errors of two ADI schemes based on the Convergence Criterion 2 are shown as open circles in Fig. 4. It can be seen that these errors are almost identical to those of the Convergence Criterion 1. The relative difference between them is usually less than 1%. The number of iterations N_i of the Convergence Criterion 2 is also depicted in Fig. 5. First, we found that the shapes of N_i for the Convergence Criterion 2 are similar to those for the Convergence Criterion 1. In particular, two quasi-linear patterns for small and large enough Δt values are also seen. Second, excluding large Δt values where a quasi-linear pattern is still followed, a global minimum in N_i can be identified again which ensures the shortest CPU time and yields the best accuracy. Third, in almost all cases, the number of iterations of the Convergence Criterion 2 is larger than that of the Convergence Criterion 1. In other words, based on analytical solutions, the Convergence Criterion 1 can sense the convergence earlier. In summary, the present study demonstrates that with a minor increment in CPU time, the Convergence Criterion 2 can reproduce the optimal results of the Convergence Criterion 1, without knowing the analytical solutions.

4. Applications

In this section, the numerical performance of the entire pseudo-time coupled solvation model will be examined. In the present pseudo-transient solvation model, the forward Euler scheme is still used in the integration of the potential driven geometric flow equation (7), while two time splitting ADI schemes will be tested for the NPB equation (12). We will first specify model parameters and general simulation setup. We then study the numerical convergence and stability of the entire solvation simulation. At last, the solvation analysis of two types of biological systems including small compounds and large proteins is studied in details.

4.1. Model parameters and simulation setup

The parameters involved in the pseudo-time coupled nonlinear solvation model are chosen as the follows. The Lennard-Jones potential U^{att} in (8) is calculated as in [32]. The dielectric constants are taken as $\epsilon_m = 1$ and $\epsilon_s = 80$. The ionic strength is chosen as $I_s = 0.1$ throughout. The atomic radii of each atom in all cases are enlarged according to [7]. In particular, the radii from the CHARMM force field need to be multiplied by a common factor of value 1.1. As in the previous studies [7,8,32], the surface tension γ is treated as a fitting parameter, with its initial value being $\gamma = 1/15$ to scale other parameters. We set the bulk density coefficient to be $\rho_s/\gamma = 2$ and choose the pressure coefficient as $p/\gamma = 0.2$. The final values of γ are different for various real systems [7,8,32]. We note that all parameter values used in the present study are identical to those in the previous pseudo-time coupled model [32].

With a uniform mesh size h along all three dimensions, a large enough computational domain is chosen in all cases so that the error introduced by the approximate boundary condition (13) is negligible. Again, the time increment for u and S is denoted as Δt_u and Δt_S, respectively. In all simulations, Δt_S is fixed to be $\Delta t_S = h^2/4.5$, while Δt_u is chosen according to the stability constrain. The initial profile of the hypersurface function S will be generated as in [32], while the initial value of the electrostatic potential u is simply chosen as zero. Then, a pseudo-time coupling will be conducted until the convergence criterion 2 is satisfied. In particular, for real biomolecular systems, the computation will stop if either the coupling time T_c is reached or the correction in the solvation free energy is less than 10^{-6}.

The solvation free energy is calculated as the follows. It is known that the total free energy functional of solvation does not directly provide the total solvation free energy. Actually, one needs to calculate the difference of the macromolecular system in the vacuum and in the solvent. Denote the converged solution to be $S(\mathbf{r})$ and $u(\mathbf{r})$, where $\mathbf{r} = (x, y, z)$. Recall that the unscaled electrostatic potential is $\phi = \frac{k_B T}{e_c} u$. The solvation free energy can be computed as

$$\Delta G = G_{\text{np}} + (G_{\text{p}} - G_0), \tag{40}$$

where G_{np} and G_{p} is, respectively, the nonpolar and polar solvation free energy of the solute solvent system with different ϵ_s and ϵ_m, while G_0 is the polar free energy calculated from the homogeneous (vacuum) environment with $\epsilon_s = \epsilon_m = 1$ [32]. The term $G_{\text{p}} - G_0$ can be regarded as the electrostatic solvation free energy. Following [32], the polar solvation free energy of the macromolecule is evaluated as

$$G_{\text{p}} = \frac{1}{2} \int_{\Omega} S(\mathbf{r}) \rho_m \phi(\mathbf{r})\, d\mathbf{r} = \frac{1}{2} \sum_{i=1}^{N_m} Q(\mathbf{r}_i) \phi(\mathbf{r}_i), \tag{41}$$

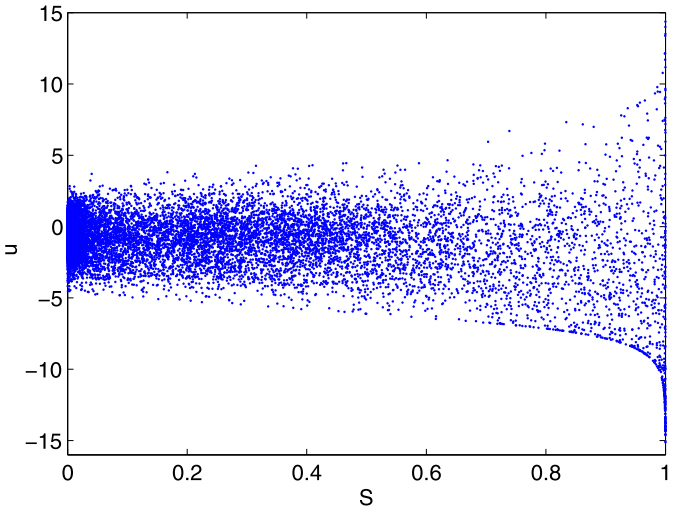

Fig. 6. The plot of the electrostatic potential u against the corresponding hypersurface function S at all nodes where $0 < S < 1$ (PDB ID: 1ajj). Here, u is the ADI2 solution of the time dependent NPB equation at time $T_c = 10$.

where $Q(\mathbf{r}_i)$ is the ith partial charge at a location $\mathbf{r}_i$ in the biomolecule, and N_m is the total number of partial charges. Similarly, the electrostatic solvation free energy can be calculated as

$$\Delta G_{\mathrm{p}} = G_{\mathrm{p}} - G_0 = \frac{1}{2}\sum_{i=1}^{N_m} Q(\mathbf{r}_i)\big(\phi(\mathbf{r}_i) - \phi_0(\mathbf{r}_i)\big), \tag{42}$$

where ϕ and ϕ_0 are electrostatic potentials in the presence of the solvent and the vacuum, respectively. The nonpolar solvation free energy of the macromolecule, G_{np}, is computed exactly according to its definition [32]. Finally, the total solvation free energy can be computed according to (40).

4.2. Numerical convergence

We first test the stability of the proposed pseudo-transient solvation models. A protein system (PDB ID: 1ajj) is chosen for this purpose. We set the spacing as $h = 0.5$ Å and the coupling time as $T_c = 10$. It is found that both ADI schemes are no longer unconditionally stable for protein systems. In particular, the critical time increment is found to be $\Delta t_u = h^2/18$ for both ADI schemes. We note that under the same setting, the critical time increment for the explicit Euler scheme is $\Delta t_u = h^2/585$. Thus, a significant acceleration in biomolecular simulations can be achieved by using the proposed ADI schemes.

The electrostatic potential u generated by the ADI2 scheme is shown in Fig. 6, which can be studied together with the electrostatic potential u generated by the explicit Euler scheme shown in Fig. 2. It can be seen that both solutions have similar distribution and attain large $|u|$ values around $S = 1$. In particular, it can be observed from Fig. 6 that the largest $|u|$ value is around $u = -15$. By evaluating such a potential in $\sinh(u)$, an extremely large nonlinear term is attained. Any explicit integration is essentially unstable in dealing with such a large nonlinear term. Thus, a filtering procedure has to be adopted in the forward Euler scheme [32]. Denote the nonlinear term as $g(S,u) := (1-S)\bar{\kappa}^2\sinh(u)$. In the present study, whenever $|g(S,u)|$ is greater than 10^4, it will be forced to be $g(S,u) = \pm 10^4$ in the explicit Euler computation. Such a filtering process stabilizes the explicit Euler scheme, and is used to produce the results shown in Fig. 2. Nevertheless, no filtering is needed in the proposed ADI schemes, due to the analytical treatment of the nonlinear term. Basically, the proposed ADI schemes make sure that the biomolecular simulation becomes more tolerable to large potential values, without altering the underlying physics. We note that an edge of an exponential shape can be clearly observed in Fig. 6 for some negative u values. If one plots $g(S,u)$ against S, one will see that all electrostatic potential values on the edge will produce $g(S,u)$ values of the same magnitude. This indicates that the critical time increment for the ADI2 scheme is exactly determined by the nonlinear term.

When an even larger Δt_u is used in the proposed ADI schemes, the computation usually becomes unstable. In particular, it is found that right before the blowing-up of the potential values, some large and negative S values are produced. Such an instability shall be related to the coupling of the potential driven geometric flow equation (7) and its explicit Euler integration. Nevertheless, the exact cause of such an instability could be complicated and is beyond the scope of the present paper.

We next examine the steady state convergence of the ADI schemes. Using $\Delta t_u = h^2/18$, the time histories of the area, volume and solvation free energy generated by both ADI schemes are depicted in Fig. 7. For a comparison, we also include another two sets of time historical values in Fig. 7. One set is generated by using the filtered explicit Euler scheme for solving the NPB equation (5), while the other set is obtained by using explicit Euler scheme for solving the time dependent

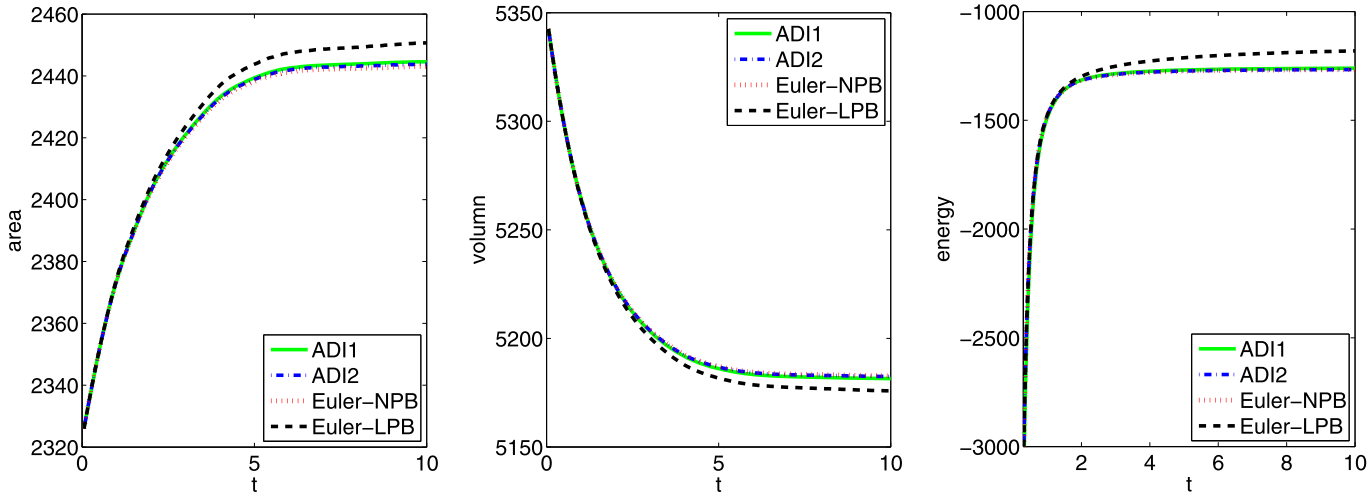

Fig. 7. The time evolution histories of area (left), volume (center), and solvation free energy (right) for the protein 1ajj.

LPB equation. These two sets will be referred to as the Euler–NPB and Euler–LPB results, respectively, in the present study. A small time increment $\Delta t_u = h^2/585$ has to be used in both Euler cases. It can be observed from Fig. 7 that the steady state solution is numerically achieved in all numerical experiments when t is large. There is a noticeable difference between the LPB result and other three NPB results, for all three quantities, whereas the differences between two ADI schemes and the filtered Euler scheme are normally indistinguishable. From numerical point of view, a very small Δt_u value is used in the explicit Euler scheme, while a much larger Δt_u value is employed in ADI computations. The agreement between the ADI and Euler results thus demonstrates the accuracies of the proposed ADI schemes.

4.3. Solvation energies of small compounds

We next apply the proposed pseudo-transient solvation model to compute the solvation free energies of a set of 17 small compounds. This test set was originally studied by Nicholls et al. [20] by using various approaches. This benchmark set has also been examined by using various differential geometry based solvation models [7,8,32]. Nontrivial biological features are involved in this set of compounds, such as the existence of polyfunctional or interacting polar groups, which lead to strong solute–solvent interactions [20]. Generally speaking, due to these features, this set is considered as a challenging test in biomolecular simulations [7,8]. On the other hand, this is an excellent benchmark set, because experimental measurements are available. However, it should be emphasized that these experimental results are not the limiting values to which our ADI numerical results should converge. Instead, the experiment data shall be treated as a benchmark for assessing the modeling errors of the pseudo-time coupled nonlinear solvation model [32], or more generally, the implicit solvent models [25,23,1].

For this set of 17 compounds, the OpenEye-AM1-BCC v1 parameters are used for the charges, while atomic coordinates and radii are based on a new parameterization introduced by Nicholls et al., i.e., the so-called ZAP-9 form [20]. Basically, in the ZAP-9 parameterization, certain types of radii are adjusted from Bonds radii so that a better agreement with experimental data is achieved. With these structures and charge parameters, both implicit solvent and explicit solvent approaches have been examined in [20]. In the explicit solvent model, the solvent is represented in molecular or atomic details. Thus, the explicit solvent approach is usually more accurate, but is also more expensive than the implicit solvent approaches [25,23,1]. In [20], the root mean square error (RMS) of the explicit solvent model is found to be 1.71 ± 0.05 kcal/mol, while the smallest RMS for implicit solvent approaches that they examined is 1.87 ± 0.03 kcal/mol.

Three NPB solvers, including two ADI schemes and the filtered Euler scheme, are considered here. Because these compounds are small biological systems, we employ a dense mesh with $h = 0.25$ Å and a short coupling time $T_c = 5$ in the present pseudo-transient solvation simulation. The critical Δt_u values are determined numerically. In particular, we have $\Delta t_u = h^2/4.5$ and $\Delta t_u = h^2/13.5$, respectively, for the ADI1 and ADI2 schemes, while that of the filtered Euler scheme is very severe: $\Delta t_u = h^2/630$. Consequently, the ADI1 and ADI2 scheme is, respectively, about 36 times and 12 times faster than the filtered Euler scheme. An even larger Δt_u might be possible in the ADI1 case, but is not tested. This is because the code has to be modified so that fractional time steps are allowed in solving the hypersurface function S. For the fitting parameter γ, an initial value of $\gamma = 1/15$ is used in all NPB simulations. The final value is chosen as $\gamma = 0.0095$ and $\gamma = 0.0102$, respectively, for two ADI schemes and the Euler scheme.

The numerical results of the Euler scheme and two ADI schemes are summarized in Table 5. In comparison with the experiment data, the errors of these three NPB solvers are also reported. It can be observed from Table 5 the results of both ADI schemes are identical. Moreover, the difference between the ADI results and those of the Euler scheme is very minor. The RMS for the Euler, ADI1, and ADI2, is 1.8359, 1.8294, and 1.8293 kcal/mol, respectively. This demonstrates the accuracy of the ADI schemes, because the filtered Euler results are solved by using a very small time step $\Delta t_u = h^2/630$. On the other hand, from the modeling error point of view, the present RMS errors are slightly better than those of the implicit solvent

Table 5
Electrostatic solvation free energies (kcal/mol) for 17 compounds.

Compound	Experimental	Euler–NPB		ADI1		ADI2	
		ΔG	Error	ΔG	Error	ΔG	Error
glycerol triacetate	−8.84	−10.10	−1.26	−10.22	−1.38	−10.22	−1.38
benzyl bromide	−2.38	−3.71	−1.33	−3.53	−1.15	−3.53	−1.15
benzyl chloride	−1.93	−3.91	−1.98	−3.73	−1.80	−3.73	−1.80
m-bis(trifluoromethyl)benzene	1.07	−1.95	−3.02	−1.90	−2.97	−1.90	−2.97
N,*N*-dimethyl-p-methoxybenzamide	−11.01	−7.36	3.65	−7.30	3.71	−7.30	3.71
N,*N*-4-trimethylbenzamide	−9.76	−5.99	3.77	−5.88	3.88	−5.89	3.87
bis-2-chloroethyl ether	−4.23	−2.61	1.62	−2.71	1.52	−2.71	1.52
1,1-diacetoxyethane	−4.97	−6.49	−1.52	−6.58	−1.61	−6.58	−1.61
1,1-diethoxyethane	−3.28	−2.88	0.40	−2.90	0.38	−2.90	0.38
1,4-dioxane	−5.05	−4.58	0.47	−4.62	0.43	−4.62	0.43
diethyl propanedioate	−6.00	−5.96	0.04	−6.04	−0.04	−6.04	−0.04
dimethoxymethane	−2.93	−3.41	−0.48	−3.46	−0.53	−3.46	−0.53
ethylene glycol diacetate	−6.34	−6.78	−0.44	−6.87	−0.53	−6.87	−0.53
1,2-diethoxyethane	−3.54	−2.63	0.91	−2.72	0.82	−2.72	0.82
diethyl sulfide	−1.43	−1.31	0.12	−1.19	0.24	−1.19	0.24
phenyl formate	−4.08	−6.62	−2.54	−6.52	−2.44	−6.52	−2.44
imidazole	−9.81	−10.40	−0.59	−10.42	−0.61	−10.42	−0.61
RMS		1.8359		1.8294		1.8293	

models studied in [20]. This means that the proposed pseudo-transient solvation model is a competitive implicit solvent model for solvation analysis.

4.4. Solvation energies of proteins

We finally consider some larger biological systems, i.e., a set of 23 proteins, which has been used for testing the previous solvation models [12,7,8,32]. The discrete atomic structures of these proteins are prepared as in the previous studies [12,7,8,32]. In particular, hydrogen atoms are added to obtain full all-atom models and extra water molecules that are attached to proteins are excluded in all structures. The CHARMM22 force field is employed to derive partial charges at atomic sites and atomic Van der Waals radii in angstroms.

Four numerical schemes, i.e., the Euler–LPB, Euler–NPB, ADI1, and ADI2, are tested for these 23 proteins. In all computations, the spacing is chosen as $h = 0.5$ Å and the coupling time is $T_c = 10$. The critical time increment is found to be $\Delta t_u = h^2/585$ and $\Delta t_u = h^2/18$, respectively, for two Euler schemes and two ADI schemes. Following [32], the surface tension parameter γ is fixed to be $\gamma = 1/15$ throughout, with all other model parameters being chosen as in the previous subsections.

The solvation free energies of these 23 proteins estimated by two Euler schemes and two ADI schemes are listed in Table 6. For a comparison, the results of two other models, i.e., MIBPB and LPB–BVP, are also given in Table 6. The LPB–BVP results are generated by using one of the original differential geometry based solvation models [8], in which, a boundary value problem (BVP) of the LPB equation is solved in each step of an iterative procedure. The MIBPB results are calculated by using a second order interface treatment, the matched interface and boundary (MIB) method [34,31], for solving the classical LPB equation with a sharp solute–solvent interface [12].

With various differences in the models or governing equations, the solvation free energies shown in Table 6 are of course different. Nevertheless, there is still a general agreement between the present results and the literature ones. In particular, the Euler–LPB results are quite close to the LPB–BVP and MIBPB results, since they are all based on the linearized Poisson–Boltzmann equation. On the other hand, there is a noticeable difference between the Euler–LPB and Euler–NPB results, even though all numerical settings except the reaction term are identical in these two Euler computations. This demonstrates the modeling difference between linear and nonlinear models, which could be utilized appropriately in practical biomolecular studies. An excellent agreement between the Euler–NPB results and those of two ADI schemes can be seen. In fact, the relative deviation of the ADI2 and ADI1 estimates from the Euler–NPB one is no larger than 0.2% and 0.6%, respectively. With a very small time stepsize $\Delta t_u = h^2/585$, the Euler–NPB result can be regarded as the reference solution here. In this sense, both ADI schemes provide decent accuracies with a large time stepsize $\Delta t_u = h^2/18$, and the ADI2 results are more accurate than the ADI1 results, as predicted by the numerical orders. The present findings indicate that the proposed ADI schemes can provide accurate and reliable estimates based on the pseudo-time coupled solvation model.

Besides the solvation free energy, the visual difference in the electrostatic potential u predicted by the linear and nonlinear models is also studied. To illustrate this difference, three solvers including the Euler–LPB and two ADI schemes are employed and a protein system with PDB ID: 1fac is considered. The result of the Euler–NPB scheme is omitted here, because it is indistinguishable from those of the ADI schemes. Here, we first generate isosurfaces $S(\mathbf{r}) = 0.5$ in each estimate. The corresponding potential u is then projected onto these isosurfaces, see Fig. 8. Biologically, these surface potential plots could reveal the fast/slow electrostatic potential changing region on the solute–solvent boundary. Such visual information may be used to detect charged active sites, which are helpful to the further studies on the protein–protein or protein–ligand

Table 6
Electrostatic solvation free energies (kcal/mol) for 23 proteins.

PDB ID	Atom #	MIBPB [12]	LPB–BVP [7]	Euler–LPB	Euler–NPB	ADI1	ADI2
1ajj	519	−1137.2	−1178.5	−1180.9	−1267.4	−1260.6	−1266.9
1bbl	576	−986.8	−965.9	−970.5	−978.8	−977.2	−978.2
1bor	832	−853.7	−871.4	−866.8	−934.5	−928.8	−933.3
1bpi	898	−1301.9	−1281.2	−1276.3	−1285.8	−1283.4	−1284.7
1cbn	648	−303.8	−304.8	−251.4	−256.3	−255.5	−256.0
1fca	729	−1200.1	−1200.6	−1181.3	−1224.9	−1221.8	−1223.8
1frd	1478	−2852.2	−2844.8	−2842.8	−2885.6	−2881.3	−2884.1
1fxd	824	−3299.8	−3291.9	−3314.4	−3351.0	−3347.0	−3349.4
1hpt	858	−811.6	−808.2	−784.5	−792.5	−790.4	−791.6
1mbg	903	−1346.1	−1328.2	−1317.5	−1332.4	−1328.7	−1330.9
1neq	1187	−1730.1	−1713.9	−1699.8	−1713.4	−1710.3	−1712.1
1ptq	795	−873.1	−866.2	−852.7	−872.9	−869.6	−871.6
1r69	997	−1089.5	−1072.7	−1041.2	−1050.3	−1048.2	−1049.4
1sh1	702	−753.3	−771.8	−794.5	−823.0	−819.2	−821.9
1svr	1435	−1711.2	−1704.6	−1731.3	−1754.8	−1750.6	−1753.6
1uxc	809	−1138.7	−1125.7	−1143.5	−1154.1	−1151.7	−1153.1
1vii	596	−901.5	−892.0	−888.5	−895.4	−893.6	−894.7
2erl	573	−948.8	−935.8	−916.2	−921.2	−919.8	−920.6
2pde	667	−820.9	−843.0	−846.0	−887.0	−881.6	−885.5
451c	1216	−1024.6	−1020.6	−964.0	−981.5	−978.5	−980.3
1a2s	1272	−1913.5	−1900.3	−1827.8	−1845.2	−1842.5	−1844.2
1a63	2065	−2373.5	−2380.5	−2389.7	−2429.0	−2423.9	−2427.2
1a7m	2809	−2155.5	−2179.8	−2081.2	−2150.9	−2141.3	−2147.4

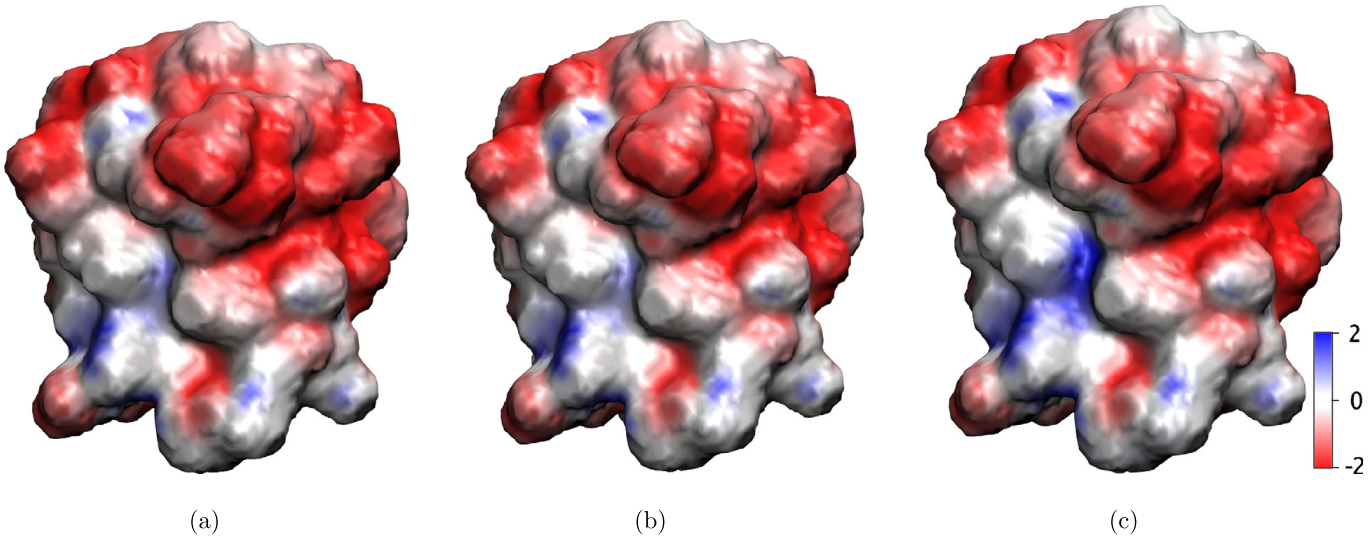

Fig. 8. Plots of the surface potential of the protein 1fca. (a) ADI1; (b) ADI2; (c) Euler–LPB.

interactions. A comparison among three isosurfaces in Fig. 8 shows that two ADI results are almost visually identical, while the LPB surface potential is slightly different from that of the NPB model. Specifically, the cavity regions tend to have a higher positive charge value in the linear model. Hence, for the purpose of guiding the protein–protein interaction analysis, the linear and nonlinear models could function differently.

We finally compare the computational speed of three NPB solvers. The CPU time in minutes consumed in the solvation analysis of 23 proteins is shown in Fig. 9. Due to the use of a large time increment Δt_u, the ADI schemes are much faster than the explicit Euler scheme in biomolecular simulations. On average, it is found that the ADI1 and ADI2 scheme is, respectively, 13.58 and 10.58 times faster than the Euler–NPB scheme in solving large protein systems. We note that per time step, the ADI1 scheme involves less algebraic operations than the ADI2 scheme so that the ADI1 scheme is slight faster than the ADI2 scheme when using the same time increment $\Delta t_u = h^2/18$.

5. Conclusion

This paper overcomes the numerical difficulties in solving the time dependent nonlinear Poisson–Boltzmann (NPB) equation for the biomolecular solvation analysis. The time integration of the NPB equation is known to be the bottleneck in numerical solution of entire pseudo-time coupled solvation system, because the usual explicit schemes are inefficient and unstable in dealing with the hyperbolic sine nonlinear term of the NPB equation [32]. A novel operator splitting procedure is

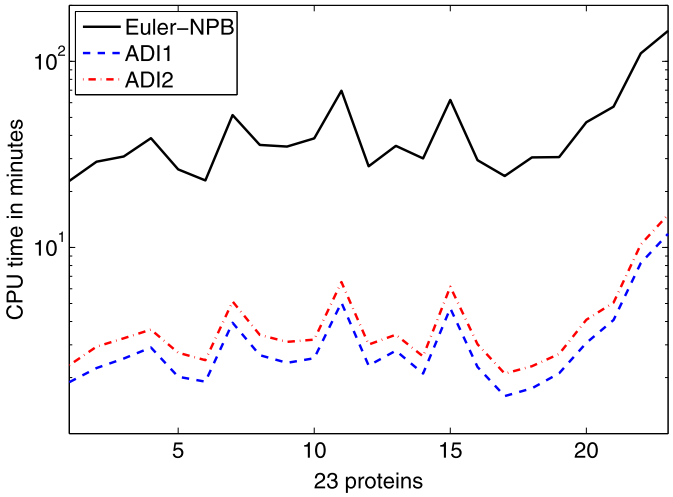

Fig. 9. CPU time in minutes consumed in solvation analysis of 23 proteins.

introduced in this paper so that the nonlinear instability can be completely avoided through an analytical integration. Based on that, both Douglas and Douglas–Rachford alternating direction implicit (ADI) schemes are developed for solving the non-homogeneous diffusion subsystem of the split NPB equation. Due to the definition of a smoothly varying solute–solvent interface, central finite differences are simply employed to secure a second order of accuracy in spatial discretization. The resulting linear systems in both ADI schemes are tridiagonal, and can be efficiently solved via the Thomas algorithm.

The proposed operator splitting ADI schemes are first validated by studying time dependent and time independent NPB equations with constructed analytical solutions. In both tests, the ADI schemes are found to be unconditionally stable in solving the NPB equation alone. Both spatial and temporal orders of accuracy are found to be two for the ADI scheme 2 (ADI2), which agrees with the theoretical estimate. The same orders hold for the ADI scheme 1 (ADI1), even though the temporal order of the ADI1 is designed to be one. This indicates a temporal super convergence of the ADI1. In studying the time independent NPB equation, the pseudo-transient approximation error between the steady state solution and the solution of the original NPB equation is carefully investigated. A realistic stopping criterion is developed, in which the computation will stop if a stopping time is reached or the average change in the solution is less than a given tolerance. By using such a stopping criterion, the pseudo-transient approximation error is found to be dominated by the discretization errors of the ADI schemes.

The application of both ADI schemes to calculate solvation free energies of small compounds and large protein systems is then considered. Due to the complex coupling in the real solvation system, the entire time integration is no longer unconditionally stable. Nevertheless, much larger time increments are allowed in both ADI schemes so that the present biomolecular simulation becomes over 10 times faster, while the same level of accuracy is still maintained. In general, the ADI1 scheme is slightly faster than the ADI2 scheme, while the ADI2 scheme is slightly more accurate than the ADI1 scheme. Since the proposed ADI schemes can withstand a very strong nonlinear effect, the full prediction power of the nonlinear solvation model can be numerically accomplished. The modeling difference between the NPB and linearized Poisson–Boltzmann (LPB) equations in the pseudo-time coupled solvation analysis is demonstrated.

Further stability analysis and computational acceleration of the entire coupled solvation system are under our consideration. The development of a robust adaptive procedure [17] in the present NPB steady state solution that enables an increasingly larger time increment as time becomes larger will be explored in a future work.

Acknowledgements

This work was supported in part by the NSF grants DMS-1016579 and DMS 1318898, and by a UA Research Grants Committee Award.

References

[1] N.A. Baker, Improving implicit solvent simulations: a Poisson-centric view, Curr. Opin. Struct. Biol. 15 (2005) 137–143.
[2] W. Bao, Q. Du, Computing the ground state solution of Bose–Einstein condensates by a normalized gradient flow, SIAM J. Sci. Comput. 25 (2004) 1674–1697.
[3] W. Bao, S. Jin, P.A. Markowich, On time-splitting spectral approximation for the Schroedinger equation in the semiclassical regime, J. Comput. Phys. 175 (2002) 487–524.
[4] P.W. Bates, Z. Chen, Y.H. Sun, G.W. Wei, S. Zhao, Geometric and potential driving formation and evolution of biomolecular surfaces, J. Math. Biol. 59 (2009) 193–231.
[5] P.W. Bates, G.W. Wei, S. Zhao, Minimal molecular surfaces and their applications, J. Comput. Chem. 29 (2008) 380–391.
[6] Q. Cai, M.J. Hsieh, J. Wang, R. Luo, Performance of nonlinear finite-difference Poisson–Boltzmann solvers, J. Chem. Theory Comput. 6 (2010) 203–211.
[7] Z. Chen, N. Baker, G.W. Wei, Differential geometry based solvation model I: Eulerian formation, J. Comput. Phys. 229 (2010) 8231–8258.

[8] Z. Chen, N. Baker, G.W. Wei, Differential geometry based solvation model II: Lagrangian formation, J. Math. Biol. 63 (2011) 1139–1200.
[9] M. Cui, Compact alternating direction implicit method for two-dimensional time fractional diffusion equation, J. Comput. Phys. 231 (2012) 2621–2633.
[10] W. Dai, R. Nassar, A generalized Douglas ADI method for solving three-dimensional parabolic differential equations on multilayers, Int. J. Numer. Methods Heat Fluid Flow 7 (1997) 659–674.
[11] R.I. Fernandes, G. Fairweather, An ADI extrapolated Crank–Nicolson orthogonal spline collocation method for nonlinear reaction–diffusion systems, J. Comput. Phys. 231 (2012) 6248–6267.
[12] W. Geng, S. Yu, G.W. Wei, Treatment of charge singularities in implicit solvent models, J. Phys. Chem. 127 (2007) 114106.
[13] M.J. Holst, Multilevel methods for the Poisson–Boltzmann equation, Ph.D. thesis, University of Illinois at Urbana-Champaign, 1993.
[14] M.J. Holst, F. Saied, Numerical solution of the nonlinear Poisson–Boltzmann equation: Developing more robust and efficient methods, J. Comput. Chem. 16 (1995) 337–364.
[15] W. Im, D. Beglov, B. Roux, Continuum solvation model: Computation of electrostatic forces from numerical solutions to the Poisson–Boltzmann equation, Comput. Phys. Commun. 111 (1998) 59–75.
[16] B. Lee, F.M. Richards, Interpretation of protein structures: estimation of static accessibility, J. Mol. Biol. 55 (1973) 379–400.
[17] W. Lo, L. Chen, M. Wang, Q. Nie, Efficient and robust methods for steady state patterns in reaction–diffusion systems, J. Comput. Phys. 232 (2012) 5062–5077.
[18] B.Z. Lu, Y.C. Zhou, M.J. Holst, J.A. McCammon, Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications, Commun. Comput. Phys. 3 (2008) 973–1009.
[19] B.A. Luty, M.E. Davis, J.A. McCammon, Solving the finite-difference nonlinear Poisson–Boltzmann equation, J. Comput. Chem. 13 (1992) 1114–1118.
[20] A. Nicholls, D.L. Mobley, P.J. Guthrie, J.D. Chodera, V.S. Pande, Predicting small-molecule solvation free energies: An informal blind test for computational chemistry, J. Med. Chem. 51 (2008) 769–779.
[21] F.M. Richards, Areas, volumes, packing and protein structure, Annu. Rev. Biophys. Bioeng. 6 (1977) 151–176.
[22] W. Rocchia, E. Alexov, B. Honig, Extending the applicability of the nonlinear Poisson–Boltzmann equation: Multiple dielectric constants and multivalent ions, J. Phys. Chem. B 105 (2001) 6507–6514.
[23] B. Roux, T. Simonson, Implicit solvent models, Biophys. Chem. 78 (1999) 1–20.
[24] A. Sayyed-Ahmad, K. Tuncay, P.J. Ortoleva, Efficient solution technique for solving the Poisson–Boltzmann equation, J. Comput. Chem. 25 (2004) 1068–1074.
[25] K.A. Sharp, B. Honig, Electrostatic interactions in Macromolecules: theory and applications, Annu. Rev. Biophys. Biophys. Chem. 19 (1990) 301–332.
[26] A.I. Shestakov, J.L. Milovich, A. Noy, Solution of the nonlinear Poisson–Boltzmann equation using pseudo-transient continuation and the finite element method, J. Colloid Interface Sci. 247 (2002) 62–79.
[27] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations, 2nd edition, SIAM, 2004.
[28] H. Wang, K.X. Wang, An $O(N \log^2 N)$ alternating-direction finite difference method for two-dimensional fractional diffusion equations, J. Comput. Phys. 230 (2011) 7830–7839.
[29] G.W. Wei, Differential geometry based multiscale models, Bull. Math. Biol. 72 (2010) 1562–1622.
[30] S. Yu, S. Zhao, G.W. Wei, Local spectral time-splitting method for first and second order partial differential equations, J. Comput. Phys. 206 (2005) 727–780.
[31] S. Zhao, High order matched interface and boundary methods for the Helmholtz equation in media with arbitrarily curved interfaces, J. Comput. Phys. 229 (2010) 3155–3170.
[32] S. Zhao, Pseudo-time coupled nonlinear models for biomolecular surface representation and solvation analysis, Int. J. Numer. Methods Biomed. Eng. 27 (2011) 1964–1981.
[33] S. Zhao, J. Ovadia, X. Liu, Y.-T. Zhang, Q. Nie, Operator splitting implicit integration factor methods for stiff reaction–diffusion–advection systems, J. Comput. Phys. 230 (2011) 5996–6009.
[34] Y.C. Zhou, S. Zhao, M. Feig, G.W. Wei, High order matched interface and boundary (MIB) schemes for elliptic equations with discontinuous coefficients and singular sources, J. Comput. Phys. 213 (2006) 1–30.