

Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation

Abstract

The Poisson-Boltzmann (PB) model is an effective approach for the electrostatics analysis of solvated biomolecules. The nonlinearity associated with the PB equation is critical when the underlying electrostatic potential is strong, but is extremely difficult to solve numerically. In this paper, we construct two operator splitting alternating direction implicit (ADI) schemes to efficiently and stably solve the nonlinear PB equation in a pseudo-transient continuation approach. The operator splitting framework enables an analytical integration of the nonlinear term that suppresses the nonlinear instability. A standard finite difference scheme weighted by piecewise dielectric constants varying across the molecular surface is employed to discretize the nonhomogeneous diffusion term of the nonlinear PB equation, and yields tridiagonal matrices in the Douglas and Douglas-Rachford type ADI schemes. The proposed time splitting ADI schemes are different from all existing pseudo-transient continuation approaches for solving the classical nonlinear PB equation in the sense that they are fully implicit. In a numerical benchmark example, the steady state solutions of the fully-implicit ADI schemes based on different initial values all converge to the time invariant analytical solution, while those of the explicit Euler and semi-implicit ADI schemes blow up when the magnitude of the initial solution is large. For the solvation analysis in applications to real biomolecules with various sizes, the time stability of the proposed ADI schemes can be maintained even using very large time increments, demonstrating the efficiency and stability of the present methods for biomolecular simulation.

Weihua Geng*, Shan Zhao†

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

Received 2012-09-28

Accepted 2013-03-19

Keywords

Nonlinear Poisson-Boltzmann equation • solvation free energy • molecular surface • electrostatic potential • operator splitting • alternating direction implicit scheme

MSC: 65N06, 65M06, 92-08, 92C05, 92C40

© Versita sp. z o.o.

1. Introduction

With the development of theoretical methods and computational techniques in the past few decades, molecular modeling has become a more and more effective and practical approach to mimic the behavior of molecules with biological significance. However, the atomic level description of molecular systems involving both a specific molecule and its aquatic and ionic surrounding is prohibitively expensive when it comes to investigating large conformational reorganization of biomolecules, such as in protein folding [11]. Alternatively, with the implicit Poisson-Boltzmann (PB) model one may

* E-mail: wgeng@as.ua.edu

† E-mail: szhao@as.ua.edu (Corresponding author)

explicitly include atomic details of the solute while implicitly treat the solvent by its mean influence and the mobile ions with a statistical mechanics distribution [2]. Therefore, the PB model substantially reduces the degree of freedom of the system.

In the PB model, the governing PB equation of electrostatic potential is a nonlinear elliptic equation defined on multiple domains with discontinuous coefficients across the solute-solvent interface or molecular surface [5, 6]. The PB equation admits analytical solutions only for simple geometries such as spheres [22] or rods [18]. For molecules with complex geometry, the PB equation can only be solved through numerical approximations. Solving the PB equation numerically is challenging due to the consideration of non-smooth solution, discontinuous coefficients (dielectric constants and ionic constants), irregular interface, singular source terms, infinite domain, as well as nonlinearity when ionic effects are strong.

There exist various numerical PB solvers and these solvers can be roughly but not completely classified into two categories: 1) The mesh-based finite difference/finite element methods [1, 10, 20, 25, 27, 29]; and 2) The boundary element method [7, 15–17, 21, 24, 35]. These PB solvers are designed based on different numerical algorithms and are aimed to applications from different perspectives. For example, the PB solvers embedded in molecular modeling packages such as Delphi [29], CHARMM [20], AMBER [25], and APBS [1] all employ finite difference discretizations and smoothened molecular surfaces. Although arguably these solvers have reduced accuracy, particularly on or near the molecular surface, the efficient, robust and user-friendly features of these PB solvers brought them popularity among the bio-oriented community. Mesh-based interface methods such as the Immersed Interface Method (IIM) [27] and Matched Interface and Boundary Poisson-Boltzmann (MIBPB) solver [10] can significantly improve the accuracy by rigorously treating the numerical difficulties such as discontinuities and singularities. However, the complexity of the algorithms to a certain extent reduces the efficiency. The boundary element methods have the potential to circumvent many of the numerical difficulties, and accelerate the solver with fast algorithms such as the fast multipole method (FMM) [7, 24, 35] and treecode [15]. However, the nonlinearity of the PB equation is cumbersome to address under the boundary element formulation. A hybrid boundary element approach which solves the nonlinear term by the finite difference is reported in [8].

In a usual approach for solving the nonlinear PB (NPB) equation, a nonlinear algebraic system is typically formed, after a spatial discretization by means of finite difference or finite element

$$\mathbf{A}\mathbf{U} + \mathbf{N}(\mathbf{U}) = \mathbf{b} \quad (1)$$

where \mathbf{U} is the vector of electrostatic potentials, \mathbf{b} is the vector of source terms, and \mathbf{A} and \mathbf{N} are, respectively, discrete operators for the linear and nonlinear parts of the NPB equation. Various algebraic methods can then be employed to solve Eq. (1) efficiently. A straightforward way is using the nonlinear relaxation method, in which the nonlinear term will be evaluated according to the previous approximations. One then needs to solve a linear system in each relaxation step. The commonly used linear iterative methods, such as Gauss-Seidal or Jacobi, can be employed as the internal iterative solver. The widely used biomolecular simulation packages such as Delphi [29] and CHARMM [20] employ the nonlinear relaxation methods. The nonlinear conjugate gradient method solves Eq. (1) approximately by calculating a minimization problem with a nonlinear integral functional, and is built in the UHBD package [26]. In the inexact Newton method [19], the Jacobian system of Eq. (1) is solved inexactly to find a descent direction first. A line search is then conducted along the descent direction to ensure global convergence. The performance of the inexact Newton method is greatly accelerated in the APBS package, where a fast algebraic multigrid solver is combined with the inexact Newton method [18, 19]. A comprehensive assessment of various algebra-based NPB solvers is carried out in [9].

The nonlinearity treatments related to the present study are pseudo-transient continuation approaches [31, 32] for solving the NPB equation. In such approaches, a time dependent NPB equation is constructed by introducing a pseudo-time. The solution of the original nonlinear boundary value system is essentially recovered by the steady state solution of the pseudo-time dependent process. The first pseudo-transient continuation approach for solving the NPB equation is introduced in [32], in which the pseudo-time convection diffusion process is solved by means of a diffusion module of an existing finite element software. To guarantee a large time step so that the long time integration can be computed quickly, the implicit Euler scheme is employed together with a linearization technique based on the first order Taylor expansion. The linearization gives rise to a coefficient matrix of the linear system at each time step by evaluating the nonlinear term at the previous time instant [32]. Since the nonlinear term of the NPB equation is treated explicitly, the overall time integration is semi-implicit. A more efficient semi-implicit approach is introduced in [31], in which

the time dependent NPB process is decomposed into diffusion, convection, and nonlinear parts. The convection and nonlinear parts are also treated explicitly, while the diffusion term is integrated implicitly by a regularized alternating direction implicit (ADI) method [13]. The regularization technique allows an even larger time increment [13]. Moreover, the Thomas algorithm [33] can be employed to solve the tridiagonal finite difference systems efficiently in each ADI step. Consequently, the semi-implicit approach [31] becomes even faster. However, a very large time increment is still prohibited in semi-implicit time integrations of pseudo-transient continuation approaches, because the nonlinear term of the NPB equation is typically a hyperbolic sine function which could be exponentially large.

The objective of this paper is to introduce fully implicit ADI methods for solving the NPB equation in a pseudo-transient continuation process, so that a very large time increment can be employed. The proposed ADI methods are built based on an operator splitting or time splitting framework, in which the nonlinear subsystem of the NPB equation can be analytically integrated. Such a time splitting procedure is often found to be a powerful tool in solving various time dependent partial differential equations (PDEs) [3, 4, 36, 39] due to its capability in enlarging the Courant-Friedrich-Lewy (CFL) number. The proposed fully implicit ADI schemes are different from the semi-implicit ADI scheme of [31] in terms of two features. First, the time-dependent NPB equation is rewritten into a convection-diffusion process in [31], while it is treated as a nonhomogeneous diffusion process in the present study. In particular, standard finite difference discretization is conducted for the entire nonhomogeneous diffusion term in the present implicit ADI integration. The resulting linear algebraic system is still tridiagonal. Second, the nonlinear term is evaluated at the previous time step in [31], while it is integrated analytically in the present operator splitting ADI schemes. Because of these two distinct features, the proposed ADI schemes become fully implicit.

In a related study, we have also developed operator splitting ADI schemes [38] for solving a generalized NPB equation [37]. Even though the schemes presented in [38] are similar to the present ones, the underlying generalized NPB equation [37] differs significantly from the classical NPB equation. The pursuit of the operator splitting ADI schemes on the classical NPB equation distinguishes the present schemes from the previous work on generalized NPB in [38]. The most distinguishing feature of the generalized NPB equation [37] is the use of a smoothly defined solute-solvent interface via some characteristic functions in the Eulerian formulation [12, 34]. This smooth definition, on the one hand, makes the stability issue of the nonlinear term more severe, because the nonlinear term now takes nonzero values within the so-called ion-exclusion layer [18, 37, 38]. Note that in the classical NPB equation, the nonlinear term is defined outside the ion-exclusion layer through a Heaviside step function. Within the ion-exclusion layer, the electrostatic potential could be large by Coulomb's law, so that an extremely large value results after evaluating the nonlinear term. Consequently, the explicit Euler scheme is found to be unconditionally unstable in solving the time dependent generalized NPB equation [37, 38], whereas the explicit Euler scheme remains to be conditionally stable in the present study for the time dependent classical nonlinear PB equation. On the other hand, the smooth definition of solute-solvent interface induces a smooth dielectric profile; thus the central finite difference can be simply employed for the spatial discretization of the generalized NPB equation with good stability. Nevertheless, the dielectric profile is discontinuous in the classical NPB equation and the potential function loses its regularity across the material interface. In the present study, a standard finite difference discretization is adopted for simplicity and the non-smooth feature of the potential function is believed to introduce additional instability in our computation. Sophisticated interface treatments [14, 40] to account for the discontinuous dielectric profile might reduce the instability associated with the spatial discretization. However, such studies are beyond the scope of the present paper.

The rest of this paper is organized as follows. In Section 2, we first introduce a pseudo-transient Poisson-Boltzmann model. The proposed time splitting ADI schemes will then be formulated for solving the time dependent NPB equation. Numerical validations of the proposed ADI schemes through benchmark examples with analytical solutions and through various benchmark biological systems are provided in Section 3. Finally, this paper ends with a conclusion.

2. Mathematical models and numerical algorithms

2.1. Physical background and mathematical models

The Poisson-Boltzmann (PB) equation is the governing equation for electrostatic potential u such that $\{u : \mathbb{R}^3 \rightarrow \mathbb{R}\}$. Consider a solute macromolecule such as a protein immersed in an aqueous solvent environment. The molecular surface Γ as defined in [23, 28] divides \mathbb{R}^3 into the inside closed solute domain Ω_m and the outside open solvent domain Ω_s such that $\mathbb{R}^3 = \Omega_m \cup \Omega_s$, $\Omega_m \cap \Omega_s = \emptyset$ and $\Gamma = \partial\Omega_m$. Defined on such multiple domains, the nonlinear PB (NPB) equation

has the form of:

$$-\nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r})) + \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r})) = \rho_m(\mathbf{r}) \quad (2)$$

where ρ_m is a singular source

$$\rho_m(\mathbf{r}) = \frac{4\pi e_c^2}{k_B T} \sum_{j=1}^{N_c} q_j \delta(\mathbf{r} - \mathbf{r}_j). \quad (3)$$

The dielectric constant ϵ is assumed to be piecewise values such that $\epsilon(\mathbf{r}) = \epsilon_m$ for $\mathbf{r} \in \Omega_m$ and $\epsilon(\mathbf{r}) = \epsilon_s$ for $\mathbf{r} \in \Omega_s$, q_j in the units of fundamental charge e_c is the partial charge on the j th atom of the solute located at \mathbf{r}_j , and k_B is the Boltzmann constant. Here $\bar{\kappa}$ is the modified Debye-Huckel parameter, which is defined as

$$\bar{\kappa}^2(\mathbf{r}) = \left(\frac{2N_A e_c^2}{1000 k_B T} \right) I_s, \quad (4)$$

for $\mathbf{r} \in \Omega_s$ where N_A is Avogadro's number and I_s is the ionic strength in the unit of mole. Numerically, when $T = 298K$, we have $\bar{\kappa}^2 = 8.486902807 \text{ \AA}^{-2} I_s$ and $\frac{e_c^2}{k_B T} = 332.06364/0.592183 \text{ \AA}$ [18]. It is noted the form of the nonlinear PB equation (2) varies on different units. The present form (2) is the dimensionless form as explained in [19]. The dimensionless electrostatic potential u can be conveniently converted to the electrostatic potential of units $kcal/mol/e_c$ by multiplying the constant 0.592183 subject to room temperature ($T = 298K$) [18].

Physically, the boundary condition for the dimensionless electrostatic potential u is defined at infinity:

$$\lim_{|\mathbf{r}| \rightarrow \infty} u(\mathbf{r}) = 0. \quad (5)$$

For mesh-based numerical computations, a finite domain Ω is required. The approximated analytical condition (6) can be employed to assign the boundary condition on $\partial\Omega$

$$u(\mathbf{r}) = \frac{e_c^2}{k_B T} \sum_{i=1}^{N_m} \frac{q_i}{\epsilon_s |\mathbf{r} - \mathbf{r}_i|} e^{-\bar{\kappa} |\mathbf{r} - \mathbf{r}_i| / \sqrt{\epsilon_s}}. \quad (6)$$

Equation (6) 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 can be used to approximate the potential solved from Eq. (2) when $\partial\Omega$ is sufficiently distanced from the macromolecule subdomain. On the other hand, we have the interface jump condition across Γ ,

$$[u]_{\Gamma} = 0 \quad \text{and} \quad \left[\epsilon \frac{\partial u}{\partial \mathbf{n}} \right]_{\Gamma} = 0 \quad (7)$$

where $[u]_{\Gamma}$ is defined to be the difference between two limiting u values approaching from outside and inside, and \mathbf{n} is the outward normal direction.

2.2. Pseudo-time governing equation and numerical difficulties

Theoretical modeling and computational simulation based on the full nonlinear PB system are highly nontrivial. This motivates the development of pseudo-transient continuation approaches for solving the nonlinear PB equation [31, 32, 37, 38]. Essentially, a pseudo-transient variation is introduced to convert Eq. (2) from the time-independent form to a time-dependent form

$$\frac{\partial u}{\partial t}(\mathbf{r}, t) = \nabla \cdot (\epsilon(\mathbf{r}) \nabla u(\mathbf{r}, t)) - \bar{\kappa}^2(\mathbf{r}) \sinh(u(\mathbf{r}, t)) + \rho_m(\mathbf{r}). \quad (8)$$

Typically, one needs to first specify an initial solution, which could be the electrostatic potential solved from a linearized PB equation [37] or trivially $u = 0$. One then numerically integrates Eq. (8) for a sufficiently long time period. The solution of the original nonlinear PB equation (2) is essentially recovered by the steady state solution of the pseudo-time dependent process (8).

However, there exist great difficulties in the numerical integration of the time dependent NPB equation (8). Generally speaking, because a long time integration is required for (8), explicit time stepping methods are usually not efficient for pseudo-transient continuation approaches [31, 32, 37, 38]. In the literature, semi-implicit time stepping methods [31, 32] are commonly employed to solve time dependent NPB equation (8) so that a large time step could be used for a stable simulation. Nevertheless, a fully implicit time integration method has never been constructed for solving the classical nonlinear PB equation (2). Such a development is mainly hindered by the implicit treatment of the hyperbolic sine nonlinear term.

2.3. Numerical discretization

Consider a uniform mesh partition of the computational domain Ω . Without the loss of generality, we assume the equal grid spacing h in all x , y and z directions. We denote the time increment to be Δt . To facilitate the following discussions, we apply the notation $u_{i,j,k}^n = u(x_i, y_j, z_k, t_n)$ to denote the electrostatic potential at node (x_i, y_j, z_k, t_n) . To gain an in-depth understanding, we first discuss a standard numerical discretization to the time dependent NPB equation (8). In particular, the central finite difference and explicit Euler scheme are used for spatial and temporal discretization, respectively. The detailed discretization is given as

$$\begin{aligned}
 u_{i,j,k}^{n+1} = & u_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k) - \Delta t \tilde{\kappa}^2 \sinh(u_{i,j,k}^n) \\
 & + \frac{\Delta t}{h^2} \left[\epsilon(x_{i+\frac{1}{2}}, y_j, z_k) (u_{i+1,j,k}^n - u_{i,j,k}^n) + \epsilon(x_{i-\frac{1}{2}}, y_j, z_k) (u_{i-1,j,k}^n - u_{i,j,k}^n) \right. \\
 & + \epsilon(x_i, y_{j+\frac{1}{2}}, z_k) (u_{i,j+1,k}^n - u_{i,j,k}^n) + \epsilon(x_i, y_{j-\frac{1}{2}}, z_k) (u_{i,j-1,k}^n - u_{i,j,k}^n) \\
 & \left. + \epsilon(x_i, y_j, z_{k+\frac{1}{2}}) (u_{i,j,k+1}^n - u_{i,j,k}^n) + \epsilon(x_i, y_j, z_{k-\frac{1}{2}}) (u_{i,j,k-1}^n - u_{i,j,k}^n) \right]
 \end{aligned} \quad (9)$$

where $Q(x_i, y_j, z_k) = \frac{4\pi e^2}{k_B T} q(x_i, y_j, z_k)$ and $q(x_i, y_j, z_k)$ 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 source term ρ_m of Eq. (2).

It is noted that the spatial discretization presented in (9) is a standard central difference scheme for the nonhomogeneous Laplacian term $\nabla \cdot (\epsilon(\mathbf{r}) \nabla u)$. This central scheme is of second order of accuracy when the dielectric profile $\epsilon(\mathbf{r})$ is smooth. Accuracy reduction is inevitable for a discontinuous dielectric profile as in the present study. To restore the second order of accuracy, sophisticated interface treatments are required to enforce interface conditions (7) into the spatial discretization [14, 40]. Nevertheless, since the main objective of the present study is to develop fully implicit time integration schemes, we will continue using the simple finite difference scheme shown in (9). This ensures that our current development can be conveniently adopted in other finite difference schemes originated from different physical backgrounds. In the present study, the value of ϵ on half grid nodes, such as $\epsilon(x_{i+\frac{1}{2}}, y_j, z_k)$, is determined by its location $(x_{i+\frac{1}{2}}, y_j, z_k)$, which is either inside/on or outside the molecular surface Γ . In particular, $\epsilon(x_{i+\frac{1}{2}}, y_j, z_k) = \epsilon_m$ if $(x_{i+\frac{1}{2}}, y_j, z_k) \in \Omega_m$, and $\epsilon(x_{i+\frac{1}{2}}, y_j, z_k) = \epsilon_s$ if $(x_{i+\frac{1}{2}}, y_j, z_k) \in \Omega_s$.

A small Δt is usually required in the discretization (9) so that the explicit Euler scheme is very inefficient. Essentially, in order to maintain the time stability, the difference between $u_{i,j,k}^{n+1}$ and $u_{i,j,k}^n$ shall be small. Thus, near the solute-solvent interface Γ where the dimensionless electrostatic potential u could take a large value [37, 38], the nonlinear term $\sinh(u_{i,j,k}^n)$ is exponentially large. Consequently, a very small Δt has to be used in the explicit Euler scheme for a stable simulation. Therefore, for a long time integration to reach the steady state, the explicit Euler scheme must be computationally very expensive. This difficulty can be partially relieved when using semi-implicit time integration schemes [31, 32], but is not completely resolved because the nonlinear term is still treated explicitly in [31, 32], i.e., evaluated at the present time level $t = t_n$. The fully implicit schemes constructed by simply evaluating the nonlinear term at $t = t_{n+1}$ will generate a nonlinear algebraic system to be solved at each time step, which is obviously time-consuming, too.

2.4. Operator splitting alternating direction implicit (ADI) schemes

To overcome the aforementioned difficulties, we will develop novel operator splitting alternating direction implicit (ADI) schemes for solving the time dependent NPB equation (8), so that the nonlinear term can be handled fully implicitly. We note that the proposed ADI schemes are similar to those developed in the pseudo-transient continuation solution of the *generalized* NPB equation [38]. Nevertheless, it is the first time in the literature that such schemes are constructed for solving the *classical* nonlinear PB equation (2), to the best of our knowledge.

We assume a uniform mesh with a grid spacing h in all x , y , and z directions, taking N_x , N_y , and N_z as the number of grid points in each direction. Let the vector $\mathbf{U}^n = \{u_{ijk}^n\}$ for $i = 1, \dots, N_x$, $j = 1, \dots, N_y$, and $k = 1, \dots, N_z$ denote all nodal values of u at the time level t_n . We will develop two ADI schemes for updating \mathbf{U}^n at the time level t_n to \mathbf{U}^{n+1} at the time level $t_{n+1} = t_n + \Delta t$. In the time-dependent nonlinear PB equation (8), we will assume time-dependence only for u . We note that all other functions in (8), i.e., ϵ , $\bar{\kappa}^2$, and ρ_m are time independent.

2.4.1. ADI scheme 1

In the ADI scheme 1 or ADI1, at each time step from t_n to t_{n+1} , the time dependent NPB equation (8) will be solved by a first order time splitting method in two stages [36]

$$\frac{\partial \mathbf{W}}{\partial t} = -\bar{\kappa}^2 \sinh(\mathbf{W}), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}], \quad (10)$$

$$\frac{\partial \mathbf{V}}{\partial t} = \nabla \cdot (\epsilon \nabla \mathbf{V}) + \rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{W}^{n+1}, \quad t \in [t_n, t_{n+1}]. \quad (11)$$

We then have $\mathbf{U}^{n+1} = \mathbf{V}^{n+1}$.

For the first stage, Eq. (10) can be analytically solved,

$$\mathbf{W}^{n+1} = \ln \left(\frac{\cosh(\frac{1}{2}\bar{\kappa}^2\Delta t) + \exp(-\mathbf{W}^n) \sinh(\frac{1}{2}\bar{\kappa}^2\Delta t)}{\exp(-\mathbf{W}^n) \cosh(\frac{1}{2}\bar{\kappa}^2\Delta t) + \sinh(\frac{1}{2}\bar{\kappa}^2\Delta t)} \right). \quad (12)$$

In other words, with \mathbf{W}^n at t_n , \mathbf{W}^{n+1} can be calculated analytically. Thus the difficulties caused by the nonlinear term $\sinh(\cdot)$ in Eq. (2) are circumvented. The right-hand side of (12) is essentially a function of \mathbf{W} and Δt . We thus rewrite Eq. (12) as $\mathbf{W}^{n+1} = F(\mathbf{W}^n; \Delta t)$ to facilitate the following discussions.

A Douglas-Rachford type ADI scheme is then proposed to solve the nonhomogeneous diffusion equation (11). The discretization of Eq. (11) using backward-Euler integration in time and central differencing in space results in

$$v_{i,j,k}^{n+1} = v_{i,j,k}^n + \Delta t(\delta_x^2 + \delta_y^2 + \delta_z^2)v_{i,j,k}^{n+1} + \Delta t Q(x_i, y_j, z_k), \quad (13)$$

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

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

The Douglas-Rachford ADI scheme for Eq. (11) is formulated as [33]

$$\begin{aligned} (1 - \Delta t \delta_x^2) v_{i,j,k}^* &= [1 + \Delta t(\delta_y^2 + \delta_z^2)] v_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k), \\ (1 - \Delta t \delta_y^2) v_{i,j,k}^{**} &= v_{i,j,k}^* - \Delta t \delta_y^2 v_{i,j,k}^*, \\ (1 - \Delta t \delta_z^2) v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \Delta t \delta_z^2 v_{i,j,k}^{**}. \end{aligned} \quad (14)$$

In other words, the three dimensional linear algebraic system in the implicit scheme (13) is decomposed into one dimensional linear algebraic systems in (14). Moreover, each of these linear systems has a tridiagonal structure and thus can be efficiently solved by the Thomas algorithm [5, 33].

By eliminating $v_{i,j,k}^*$ and $v_{i,j,k}^{**}$ and solving for $v_{i,j,k}^{n+1}$ in (14), we obtain

$$\begin{aligned} v_{i,j,k}^{n+1} &= v_{i,j,k}^n + \Delta t(\delta_x^2 + \delta_y^2 + \delta_z^2)v_{i,j,k}^{n+1} + \Delta t Q(x_i, y_j, z_k) \\ &\quad - \Delta t^2(\delta_x^2 \delta_y^2 + \delta_y^2 \delta_z^2 + \delta_x^2 \delta_z^2)(v_{i,j,k}^{n+1} - v_{i,j,k}^n) + \Delta t^3 \delta_x^2 \delta_y^2 \delta_z^2 (v_{i,j,k}^{n+1} - v_{i,j,k}^n). \end{aligned} \quad (15)$$

Thus, the Douglas-Rachford scheme (14) is a higher order perturbation of the backward Euler scheme (13). Since both the backward Euler scheme and the time spitting scheme (10) - (11) are first order in time, the proposed ADI1 scheme is of first order accuracy in time. In the numerical simulations, the same Dirichlet boundary values are assumed for v , v^* and v^{**} as for u . The entire ADI1 time integration is fully implicit.

2.4.2. ADI scheme 2

In the ADI scheme 2 or ADI2, at each time step from t_n to t_{n+1} , the time dependent NPB equation (8) will be solved by a second order time splitting method in three stages [36]

$$\frac{\partial w}{\partial t} = -\frac{1}{2}\bar{\kappa}^2 \sinh(w), \quad \text{with } W^n = U^n, \quad t \in [t_n, t_{n+1}], \quad (16)$$

$$\frac{\partial v}{\partial t} = \nabla \cdot (\epsilon \nabla v) + \rho_m, \quad \text{with } V^n = W^{n+1}, \quad t \in [t_n, t_{n+1}], \quad (17)$$

$$\frac{\partial \tilde{w}}{\partial t} = -\frac{1}{2}\bar{\kappa}^2 \sinh(\tilde{w}), \quad \text{with } \tilde{W}^n = V^{n+1}, \quad t \in [t_n, t_{n+1}]. \quad (18)$$

We then have $U^{n+1} = \tilde{W}^{n+1}$. As in the ADI1 scheme, in the first and last stage of the ADI2 scheme, an analytical integration can be conducted. Symbolically, we have $W^{n+1} = F(W^n; \frac{\Delta t}{2})$ and $\tilde{W}^{n+1} = F(\tilde{W}^n; \frac{\Delta t}{2})$, where F is defined as in Eq. (12).

A Douglas type ADI scheme is proposed to solve the nonhomogeneous diffusion equation (17) in the second stage. The discretization of (17) using Crank-Nicolson integration in time and central differencing in space results in

$$[1 - \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)]v_{i,j,k}^{n+1} = [1 + \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)]v_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k). \quad (19)$$

This can be decomposed into x , y , and z directions to give a Douglas ADI scheme for (17) [33]

$$\begin{aligned} (1 - \frac{\Delta t}{2}\delta_x^2)v_{i,j,k}^* &= [1 + \frac{\Delta t}{2}(\delta_x^2 + 2\delta_y^2 + 2\delta_z^2)]v_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k), \\ (1 - \frac{\Delta t}{2}\delta_y^2)v_{i,j,k}^{**} &= v_{i,j,k}^* - \frac{\Delta t}{2}\delta_y^2 v_{i,j,k}^n, \\ (1 - \frac{\Delta t}{2}\delta_z^2)v_{i,j,k}^{n+1} &= v_{i,j,k}^{**} - \frac{\Delta t}{2}\delta_z^2 v_{i,j,k}^n. \end{aligned} \quad (20)$$

Similarly, the tridiagonal systems in (20) can be efficiently solved by the Thomas algorithm.

The elimination of $v_{i,j,k}^*$ and $v_{i,j,k}^{**}$ in (20) leads to

$$\begin{aligned} [1 - \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)]v_{i,j,k}^{n+1} &= [1 + \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)]v_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k) \\ &\quad - \frac{\Delta t^2}{4}(\delta_x^2 \delta_y^2 + \delta_y^2 \delta_z^2 + \delta_x^2 \delta_z^2)(v_{i,j,k}^{n+1} - v_{i,j,k}^n) + \frac{\Delta t^3}{8}\delta_x^2 \delta_y^2 \delta_z^2 (v_{i,j,k}^{n+1} - v_{i,j,k}^n). \end{aligned} \quad (21)$$

It can be shown that the resulting equation is a higher order perturbation of the Crank-Nicolson scheme (19) in three space variables. Thus, the temporal order of accuracy of the Douglas scheme (20) is two, which is consistent with the present time splitting (16) - (18). Therefore, the proposed fully implicit ADI2 scheme is of second order of accuracy in time. Again, the boundary values of v , v^* and v^{**} are the same as those for u . In terms of the computational complexity of one cycle, the ADI2 scheme involves extra flops compared with the ADI1 scheme. Nevertheless, the ADI2 scheme is usually more accurate than the ADI1 scheme.

2.4.3. Semi-implicit ADI1 scheme

For a comparison, we also consider the counterparts of the proposed two ADI schemes with semi-implicit time integration. The semi-implicit ADI1 (semi-ADI1) scheme is constructed based on the backward-Euler integration of the time dependent NPB equation (8), but with the nonlinear term evaluated at t_n

$$u_{i,j,k}^{n+1} = u_{i,j,k}^n + \Delta t(\delta_x^2 + \delta_y^2 + \delta_z^2)u_{i,j,k}^{n+1} + \Delta t Q(x_i, y_j, z_k) - \Delta t \bar{\kappa}^2 \sinh(u_{i,j,k}^n). \quad (22)$$

By treating the nonlinear term in (22) as an additional source term, the same Douglas-Rachford ADI discretization used in the ADI1 scheme can be utilized to form the semi-ADI1 scheme

$$\begin{aligned} (1 - \Delta t \delta_x^2)u_{i,j,k}^* &= [1 + \Delta t(\delta_y^2 + \delta_z^2)]u_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k) - \Delta t \bar{\kappa}^2 \sinh(u_{i,j,k}^n), \\ (1 - \Delta t \delta_y^2)u_{i,j,k}^{**} &= u_{i,j,k}^* - \Delta t \delta_y^2 u_{i,j,k}^n, \\ (1 - \Delta t \delta_z^2)u_{i,j,k}^{n+1} &= u_{i,j,k}^{**} - \Delta t \delta_z^2 u_{i,j,k}^n. \end{aligned} \quad (23)$$

2.4.4. Semi-implicit ADI2 scheme

Similarly, the semi-implicit ADI2 (semi-ADI2) scheme is constructed based on the Crank-Nicolson integration of the time dependent NPB equation (8),

$$\left[1 - \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right]u_{i,j,k}^{n+1} = \left[1 + \frac{\Delta t}{2}(\delta_x^2 + \delta_y^2 + \delta_z^2)\right]u_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k) - \Delta t \bar{\kappa}^2 \sinh(u_{i,j,k}^n). \quad (24)$$

By treating the nonlinear term in (24) as a source term, the same Douglas ADI discretization used in the ADI2 scheme leads to the semi-ADI2 scheme

$$\begin{aligned} \left(1 - \frac{\Delta t}{2}\delta_x^2\right)u_{i,j,k}^* &= \left[1 + \frac{\Delta t}{2}(\delta_x^2 + 2\delta_y^2 + 2\delta_z^2)\right]u_{i,j,k}^n + \Delta t Q(x_i, y_j, z_k) - \Delta t \bar{\kappa}^2 \sinh(u_{i,j,k}^n), \\ \left(1 - \frac{\Delta t}{2}\delta_y^2\right)u_{i,j,k}^{**} &= u_{i,j,k}^* - \frac{\Delta t}{2}\delta_y^2 u_{i,j,k}^n, \\ \left(1 - \frac{\Delta t}{2}\delta_z^2\right)u_{i,j,k}^{n+1} &= u_{i,j,k}^{**} - \frac{\Delta t}{2}\delta_z^2 u_{i,j,k}^n. \end{aligned} \quad (25)$$

We note that the present semi-ADI1 and semi-ADI2 schemes are different from the existing semi-implicit approaches [31, 32]. The primary difference is in spatial discretization. The simplest central difference discretization is used in the present semi-implicit ADI schemes, while more complex finite difference and finite element formulas are used in [31, 32]. However, the explicit treatment of the nonlinear term is the same in the existing and present semi-implicit ADI schemes. We will illustrate how this explicit treatment will affect the stability in the next section.

2.4.5. Convergence criteria

We propose to approximate the solution to the nonlinear PB equation (2) by using the steady state solution to the time-dependent nonlinear PB equation (8), which can be achieved in theory when $t \rightarrow \infty$. Numerically we will need a stop criterion when t is sufficiently large. Two convergence criteria will be adopted simultaneously in this paper to maintain the numerical efficiency.

Criterion 1

Time integration stops when the change in the solvation energy of two consecutive time steps is less than a given tolerance. This criterion is particularly useful when calibrating the proposed schemes in constructed examples with analytical solutions.

Criterion 2

Time integration stops when $t \geq T_u$, where T_u is a user specified time. After many numerical trials, we find $T_u = 10$ is a reasonable choice for terminating the time integration in the solution of the time-dependent NPB equation (8). This usually produces adequately converged solutions for real protein systems.

3. Numerical validations

In this section, we validate our proposed ADI schemes numerically. We first solve the nonlinear PB equation on a sphere and compare the numerical results with the available analytical solution for tests of stability as well as both spatial and temporal accuracy and convergence. We next apply our new algorithms to compute the solvation energy for a series of proteins with various sizes and geometric structures. All simulations are compiled with Intel visual Fortran compiler and run on a Dell 6600 with i7-2760QM 2.4GHz CPU and 16GB memory.

3.1. A spherical cavity with analytical solution

For the nonlinear PB equation, the analytical solutions are available only for simple and special geometries. We here design a case for a spherical cavity based on the reference [30]

$$u(r) = \begin{cases} \frac{1}{\epsilon R} - \frac{1}{R} + \frac{1}{||r||} & ||r|| < R, \\ \frac{1}{\epsilon ||r||} & ||r|| > R. \end{cases} \quad (26)$$

$$\rho_m(r) = \begin{cases} 4\pi\delta(r) & ||r|| < R, \\ \bar{\kappa}^2 \sinh\left(\frac{1}{\epsilon ||r||}\right) & ||r|| > R, \end{cases} \quad (27)$$

Table 1. The critical m values for explicit Euler, semi-ADI1 (s-ADI1), semi-ADI2 (s-ADI2), ADI1, and ADI2 for solving the time-dependent nonlinear PB equation on a sphere subject to initial solutions with different magnitude H . The scheme is stable if $\Delta t \leq \frac{h^2}{m}$.

H	$h=0.5$					$h=0.25$				
	Euler	s-ADI1	s-ADI2	ADI1	ADI2	Euler	s-ADI1	s-ADI2	ADI1	ADI2
1	462	1	12	4	11	476	10	15	10	15
10	462	23	26	15	11	476	10	15	16	15
12	463	144	165	16	11	476	37	39	17	15
15	2142	2053	2092	18	11	540	464	497	18	14
20	$\sim 10^6$	∞	∞	19	11	∞	∞	∞	19	15

where $\varepsilon = \varepsilon_s/\varepsilon_m$ and R is the radius of the sphere. By defining the source term ρ_m of Eq. (2) according to Eq. (27), it can be verified that the solution in Eq. (26) satisfies Eq. (2) as well as the interface jump conditions in Eq. (7). Note that there are both singularity and non-smoothness involved in the present analytical solution. The singularity is from the source term as seen in Eq. (27) and the non-smoothness is due to the interface jump conditions in Eq. (7). Both features will significantly reduce the accuracy of the spatial discretization and introduces additional instability in temporal discretization. In our study of the spherical case, the chosen sphere has radius $R = 1 \text{ \AA}$ with one centered charge of $1e_c$. The dielectric constants are set as $\varepsilon_s = 80$ and $\varepsilon_m = 1$. The nonlinear constant $\bar{\kappa}$ is set as 1.

3.1.1. Convergence and stability

We first examine the convergence of the pseudo-transient continuation approach. For this purpose, we solve the time dependent nonlinear PB equation (8) with different initial solutions $u(\mathbf{r}, 0)$ until the stopping criteria are satisfied. We then compare our steady state solutions with the time independent analytical solution (26) to verify if the convergence is achieved. Besides the trivial initial solution $u(\mathbf{r}, 0) = 0$, we specifically construct a family of initial solutions

$$u(\mathbf{r}, 0) = H \cos\left(\frac{\pi}{6}x\right) \cos\left(\frac{\pi}{6}y\right) \cos\left(\frac{\pi}{6}z\right) + \frac{1}{\varepsilon \sqrt{x^2 + y^2 + z^2}}, \quad (28)$$

with different magnitude H for the study of the sensitivities of the schemes subject to the nonlinearity (the $\sinh(u)$ term). It can be seen that the initial solution (28) satisfies the boundary conditions for the computational domain $(x, y, z) \in \{[-3, 3] \times [-3, 3] \times [-3, 3]\}$. We considered various H values ranging from 1 to 20 for explicit Euler, semi-ADI1, semi-ADI2, ADI1, and ADI2 simulations for a comparison. It is found that under different initial solutions, all tested schemes converge to the analytical solution when t is sufficiently large, except for a few extreme cases where the convergence is ruined by the instability. This validates the present pseudo-transient continuation approach.

We next quantitatively analyze the stability of the explicit Euler, semi-ADI1, semi-ADI2, ADI1, and ADI2 schemes. Because of the singularity and non-smoothness features of the solution, all time integration schemes here are conditionally stable, i.e., there is a maximum stable time increment $\Delta t = \frac{h^2}{m}$ at different spatial mesh size h . We report the critical m values of these schemes in Table 1. From this table, we see that with the increase of H in the initial solutions (28), and thus the increase of the nonlinearity, the critical Δt values of the explicit Euler scheme and semi-implicit ADI schemes become exponentially smaller and approach zero eventually, while those of ADI schemes are essentially unchanged. Therefore, the fully-implicit schemes demonstrated much improved stability compared with the explicit and semi-implicit schemes.

3.1.2. Spatial convergence rate

We then study the spatial convergence rate of ADI schemes with the same example. The simulation results for the spherical case can be found in Table 2. Denoting u_h as the numerical solution, we use the following measures to estimate errors in a relative manner:

$$L_\infty = \frac{\max |u - u_h|}{\max |u|}, \quad L_2 = \sqrt{\frac{\sum_{i,j,k} |u - u_h|^2}{\sum_{i,j,k} |u|^2}}. \quad (29)$$

Table 2. Spatial order of convergence in solving the time dependent nonlinear PB equation on a sphere: $\Delta t = h^2/20$.

h	ADI1 scheme				ADI2 scheme			
	L_∞	rate	L_2	rate	L_∞	rate	L_2	rate
1	4.28E-01	1.18	5.41E-01	0.48	5.49E-01	1.46	8.85E-01	1.13
0.5	1.89E-01	2.83	3.88E-01	1.50	1.99E-01	2.84	4.04E-01	1.55
0.25	2.66E-02	0.12	1.37E-01	0.80	2.79E-02	0.17	1.38E-01	0.79
0.125	2.44E-02	-0.07	7.89E-02	0.75	2.47E-02	-0.06	7.99E-02	0.76
0.0625	2.56E-02	-	4.68E-02	-	2.57E-02	-	4.73E-02	-

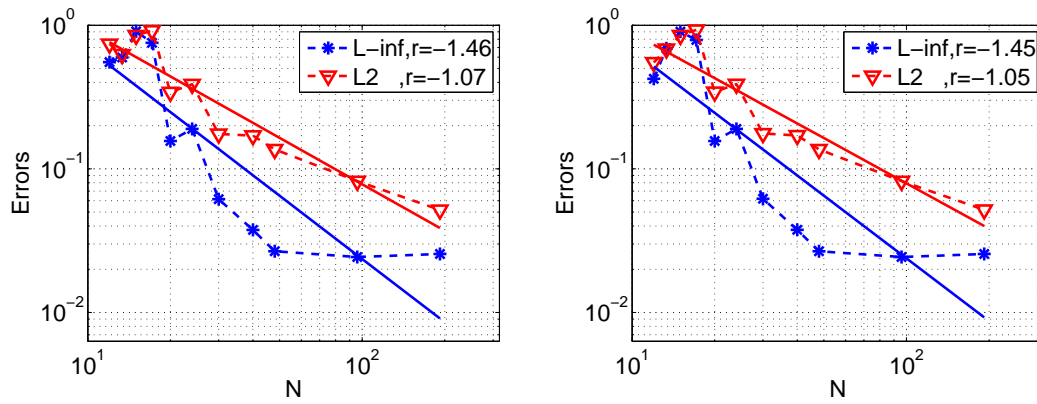


Fig 1. Spatial order of convergence in solving the time dependent nonlinear PB equation on a sphere. Left: ADI1, Right: ADI2; $\Delta t = h^2/20$; $h=12/N$; the error-grid pairs are connected with dashed lines; the linear least square fittings are plotted with solid lines; L_∞ error (blue star) and L_2 error (red triangles).

In producing results in Table 2, we vary spatial increment h and set $\Delta t = h^2/20$ to fulfill the stability requirement. The convergence rate is nonuniform because the charge singularities (the delta function) and the non-smoothness of the solution across the interface Γ are treated in an approximated sense for the present standard finite difference discretization. However, we could still see in average approximately the first-order convergence in L_∞ and L_2 norms of both ADI1 and ADI2 schemes. This approximately first order convergence rate can also be found in reference [14] when the linearized PB equation on a sphere is solved using a similar spatial discretization.

To further demonstrate this convergence behavior, we provide detailed error-mesh plots in Fig. 1. In these plots, we perform a least square fitting (the solid lines) of the errors against the number of mesh points N , which is inversely proportional to h . The slopes r of the solid straight lines are calculated and listed in the legends of both plots. The magnitude of r reflects the average convergence rate. It can be seen that both ADI1 and ADI2 schemes deliver the approximately first-order convergence in both L_∞ and L_2 norms. Visually, the errors of both ADI1 and ADI2 schemes are almost the same when N is large. Nevertheless, there are some minor differences when N is small.

3.1.3. Temporal convergence rate

We finally focus on the temporal order of convergence, which is reported in Table 3. Since the spatial discretization error is large due to the singularity and non-smoothness as described previously, a direct comparison between numerical solutions at different h and the analytical solution will not be able to reveal the temporal order. Instead, we fix $h = 0.125$ and use the electrostatic potentials solved at the finest time increment $\Delta t = 2.5E-05$ as a reference solution. The potentials computed based on other Δt values are benchmarked with this reference solution. This enables us to compute the temporal discretization error through essentially the cancellation of the spatial discretization error. We observe second order temporal convergence from the table for both ADI1 and ADI2 schemes. For a fixed Δt , the ADI2 is more accurate than the ADI1. We note some better-than-expected performance for the ADI1 scheme, i.e., the numerical order

Table 3. Temporal order of convergence in solving the time dependent nonlinear PB equation on a sphere: $h=0.125$.

Δt	ADI1 scheme				ADI2 scheme			
	L_∞	rate	L_2		rate	L_∞	rate	L_2
8.0E-04	1.79E-04	2.81	4.39E-04	2.62	1.32E+00	18.16	1.02E+00	16.22
4.0E-04	2.55E-05	2.52	7.14E-05	2.43	4.49E-06	2.29	1.34E-05	2.25
2.0E-04	4.45E-06	2.33	1.33E-05	2.30	9.19E-07	2.22	2.82E-06	2.19
1.0E-04	8.83E-07	2.46	2.70E-06	2.44	1.97E-07	2.38	6.17E-07	2.34
5.0E-05	1.60E-07	-	4.98E-07	-	3.78E-08	-	1.22E-07	-
2.5E-05	0.00E+00	-	0.00E+00	-	0.00E+00	-	0.00E+00	-

Table 4. Solvation energies (*kcal/mol*) for 24 proteins; stopping criterion: $T_u = 10$; $h = 0.5$; $\Delta t = h^2/50$ for ADI1/ADI2, and $\Delta t = h^2/480$ for explicit Euler.

Index	PDB-ID	# of atoms	LPBE	Euler	ADI1	ADI2
1	1ajj	519	-1213.78	-1333.04	-1316.25	-1328.30
2	1bbl	576	-1163.73	-1286.00	-1269.24	-1280.71
3	1bor	832	-987.09	-1075.39	-1065.07	-1072.48
4	1bpi	898	-1465.92	-1585.65	-1565.25	-1578.63
5	1cbn	648	-409.01	-452.06	-446.20	-450.10
6	1fca	729	-1234.59	-1323.47	-1314.47	-1320.99
7	1frd	1478	-2936.78	-3120.12	-3092.71	-3111.72
8	1fxd	824	-3039.65	-3162.14	-3149.43	-3158.63
9	1hpt	858	-1029.24	-1134.34	-1116.94	-1128.82
10	1mbg	903	-1466.03	-1571.46	-1555.40	-1566.29
11	1neq	1187	-2007.86	-2166.15	-2142.26	-2158.52
12	1ptq	795	-1005.64	-1130.84	-1114.09	-1125.14
13	1r69	997	-1268.61	-1370.37	-1355.16	-1365.36
14	1sh1	702	-890.88	-992.49	-982.82	-989.50
15	1svr	1435	-2056.34	-2244.25	-2219.44	-2236.25
16	1uxc	809	-1319.45	-1432.75	-1419.11	-1427.98
17	1vii	596	-1066.64	-1130.47	-1120.59	-1127.31
18	2erl	573	-1020.60	-1085.42	-1077.59	-1082.90
19	2pde	667	-914.54	-1014.30	-1004.96	-1011.56
20	451c	1216	-1256.01	-1359.58	-1346.82	-1354.48
21	1a2s	1272	-1995.11	-2115.75	-2099.94	-2111.06
22	1a63	2065	-2868.92	-3125.45	-3088.77	-3113.34
23	1a7m	2809	-2486.44	-2687.16	-2664.38	-2677.98
24	1vjw	828	-1307.41	-1417.15	-1403.47	-1412.87

is higher than the theoretical one. In fact, the same finding has been observed for the ADI1 scheme when it is applied to solve the generalized PB equation defined on a smooth solute-solvent boundary and a conventional analysis of the temporal convergence is considered [38].

3.2. Solvation energy of proteins

In this subsection, we validate the proposed time splitting ADI schemes by solving the nonlinear PB equation for real protein systems. In particular, we will compute solvation energies on a series of proteins with different size and geometric structures. For a comparison, we also report the results produced by the explicit Euler method.

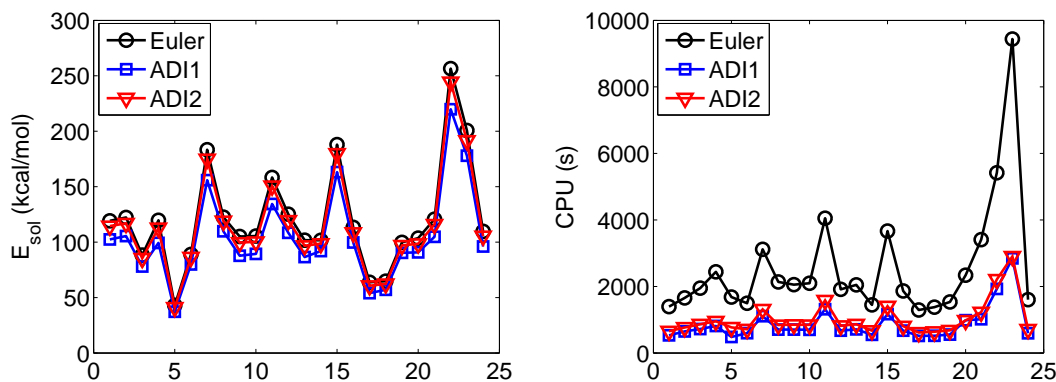


Fig 2. Energy difference (left) and CPU time (right) in solving nonlinear PB equation on 24 proteins

The solvation energy or free energy of solvation is the energy released when the solute in free space is dissolved in solvent. In molecular simulation, it can be computed by evaluating the difference between total free energy of the solute in the vacuum and in the solvent. In the context that only the electrostatic effects are considered, the solvation energy can be defined as

$$\Delta G = G_s - G_0 = \frac{1}{2} \int_{\Omega} \rho_m(\phi(r) - \phi_0(r)) dr \quad (30)$$

where ϕ and ϕ_0 in the units of kcal/mol/ e_c are electrostatic potentials in the presence of the solvent and the vacuum, respectively. They are obtained by scaling the dimensionless potentials with the constant 0.592183 subject to room temperature ($T = 298K$) [18]. Based on the trilinear interpolation of the singular charges in ρ_m , the solvation energy can be calculated as

$$\Delta G = \frac{1}{2} \sum_i \sum_j \sum_k Q(x_i, y_j, z_k) (\phi(x_i, y_j, z_k) - \phi_0(x_i, y_j, z_k)), \quad (31)$$

We solve the nonlinear PB equation (2) on a set of 24 proteins by using explicit Euler, ADI1 and ADI2 schemes. For simulation on proteins, the dielectric constants are set as $\epsilon_s = 80$ and $\epsilon_m = 1$. The ionic strengths I is set as 0.15M, which makes the nonlinear constant $\bar{\kappa} = 0.1257$.

To quantitatively justify the accuracy, efficiency and stability, we compute the solvation energy based on electrostatic potentials solved from the nonlinear PB equation. The numerical results of the ADI1, ADI2, and Euler methods are listed in Table 4. The corresponding parameters are listed in the caption of the table and the choice of the Δt is subject to the stability requirement. From this table, we can see that all three nonlinear methods produce consistent solvation energies. In particular, the explicit Euler scheme is expected to produce some of the most accurate results, because a much smaller Δt is applied and the same spatial discretization is used. The second order time discretization ADI2 is supposed to yield more accurate results than the first order ADI1. Indeed, the results in the table adequately showed this, i.e., the results of the ADI2 are closer to those of the Euler scheme. On the other hand, the solution of the Linearized PB equation (LPBE) is carried out by using the same finite difference discretization and a biconjugate gradient solver. The solvation free energies of LPBE results are also listed in Table 4. It can be seen that there are some large differences between linear and nonlinear PB models, reflecting the importance of the nonlinear term in Eq. (2).

We also depict the simulation results in Fig. 2. To avoid the scaling problem caused by the large variation of the solvation energy value, in Fig. 2(a) we plot the differences between nonlinear PB results generated by ADI1, ADI2, and Euler schemes and linearized PB results. This figure shows the differences between linear and nonlinear models. It also demonstrates the consistency among three nonlinear PB solvers. We then plot the CPU time used by three algorithms for solving the nonlinear PB equation in Fig. 2(b). It is seen that the explicit Euler method costs about 2-3 times more CPU time than both ADI schemes.

Finally, we plot the surface potentials of a protein (1mbg) in Fig. 3. Besides the nonlinear PB results generated by the ADI1 scheme, we also compare with the linear PB results obtained by the same finite difference discretization. From

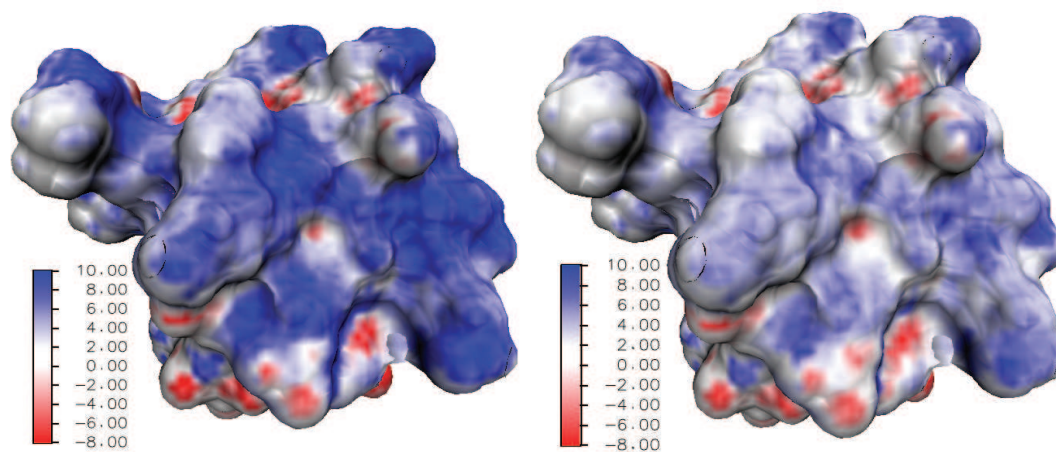


Fig 3. Surface potential mapping of protein 1mbg. Left: Linearized PB model; Right: Nonlinear PB model.

the figures, we can see the nonlinear PB results are generally in lighter colors, which stand for weaker electrostatic potentials. This phenomenon can be well explained by the nonlinear screening effect of the electrolytes when we have nonzero ionic strength $I = 0.15M$.

4. Conclusion

This paper presents the first fully implicit time integration scheme for solving the nonlinear Poisson-Boltzmann (NPB) equation in a pseudo-transient continuation framework. In such a framework, the boundary value solution of the NPB equation is essentially recovered by the steady state solution to the pseudo-time NPB process. The long time integration in this process necessitates the development of implicit schemes that allow a large time increment Δt . However, all existing implicit methods are semi-implicit, because of the difficulty in the implicit treatment of the nonlinear hyperbolic sine term. This bottleneck is overcome in the present paper by using a novel operator splitting procedure. After splitting, the nonlinear instability can be completely avoided through analytical integration, while both Douglas and Douglas-Rachford alternating direction implicit (ADI) schemes are developed for solving the nonhomogeneous diffusion subsystem. A standard finite difference scheme is employed for spatial discretization. Other more accurate finite difference methods can be applied, as long as the tridiagonal structure is preserved in each coordinate direction. The Thomas algorithm is utilized to solve linear algebraic systems efficiently in the proposed fully implicit ADI schemes.

The proposed operator splitting ADI schemes are first validated by solving the NPB equation on spherical cavities where time invariant analytical solutions are available subject to specially designed source terms. By considering different initial values, it is demonstrated that all steady state solutions of the fully-implicit ADI methods converge to the analytical solution, while those of the explicit Euler scheme and semi-implicit ADI methods blow up when the magnitude of the initial solution is large. The spatial convergence rate of the proposed ADI schemes is found to be nearly first order, because the accuracy of the standard finite difference is reduced by the singularities from the source term and non-smoothness of the solution across the molecular surface. Special cancellation is considered in order to reveal the true temporal order of accuracy. It is interesting to observe that both the proposed ADI schemes then yield a second order in time. The application of both ADI schemes to calculate solvation free energies for a series of 24 proteins is considered next. In all of our experiments, it is found that the fully implicit ADI schemes allow much larger Δt values than that of the explicit Euler scheme, so that the present biomolecular simulation becomes faster. Moreover, 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. In particular, we illustrate the theoretical difference between the linearized Poisson-Boltzmann model and the present NPB model in electrostatic analysis. The development of more robust spatial discretization associated with the proposed temporal schemes are under our consideration.

Acknowledgment

The research of Geng was supported in part by the NSF grant DMS-0915057 and the University Alabama New Faculty Startup Fund. The research of Zhao was supported in part by the NSF grant DMS-1016579.

References

- [1] N.A. Baker, D. Sept, S. Joseph, M.J. Holst, J.A. McCammon, Electrostatics of nanosystems: application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences* 98 (2001) 10037-10041.
- [2] N.A. Baker, Improving implicit solvent simulations: a Poisson-centric view. *Curr. Opin. Struct. Biol.* 15 (2005) 137-143.
- [3] W. Bao and Q. Du, Computing the ground state solution of Bose-Einstein condensates by a normalized gradient flow. *SIAM J. Sci. Comput.* 25 (2004) 1674-1697.
- [4] W. Bao, S. Jin and P.A. Markowich, On time-splitting spectral approximation for the Schrodinger equation in the semiclassical regime. *J. Comput. Phys.* 175 (2002) 487-524.
- [5] P.W. Bates, Z. Chen, Y.H. Sun, G.W. Wei, and S. Zhao, Geometric and potential driving formation and evolution of biomolecular surfaces. *J. Math. Biol.* 59 (2009) 193-231.
- [6] P.W. Bates, G.W. Wei and S. Zhao, Minimal molecular surfaces and their applications. *J. Comput. Chem.* 29 (2008) 380-391.
- [7] A.H. Boschitsch, M.O. Fenley, and H.-X. Zhou, Fast boundary element method for the linear Poisson-Boltzmann equation. *J. Phys. Chem. B* 106 (2002) 2741-2754.
- [8] A.H. Boschitsch and M.O. Fenley, Hybrid boundary element and finite difference method for solving the nonlinear Poisson-Boltzmann equation. *J. Comput. Chem.* 25 (2004) 935-955.
- [9] Q. Cai, M.J. Hsieh, J. Wang, and R. Luo, Performance of nonlinear finite-difference Poisson-Boltzmann solvers. *J. Chem. Theory Comput.* 6 (2010) 203-211.
- [10] D. Chen, Z. Chen, C.J. Chen, W.H. Geng, G.W. Wei, MIBPB: A software package for electrostatic analysis. *J. Comput. Chem.* 32 (2011) 756-770.
- [11] J.H. Chen, C.L. Brooks, J. Khandogin, Recent advances in implicit solvent based methods for biomolecular simulations. *Curr. Opin. Struct. Biol.* 18 (2008) 140-148.
- [12] Z. Chen, N. Baker, and G.W. Wei, Differential geometry based solvation model I: Eulerian formation, *J. Comput. Phys.* 229 (2010) 8231-8258.
- [13] W. Dai and R. Nassar, A generalized Douglas ADI method for solving three-dimensional parabolic differential equations on multilayers. *Int. J. Numer. Meth. Heat Fluid Flow* 7 (1997) 659-674.
- [14] W.H. Geng, S.N. Yu, and G.W. Wei, Treatment of charge singularities in implicit solvent models. *J. Phys. Chem.* 127 (2007) Art. ID 114106, 20pp.
- [15] W.H. Geng and R. Krasny, Treecode-accelerated boundary integral Poisson-Boltzmann solver. *J. Comput. Phys.* (2013) accepted.
- [16] W.H. Geng, Parallel higher-order boundary integral electrostatics computation on molecular surfaces with curved triangulation. *J. Comput. Phys.* 241 (2013) 253-265.
- [17] W.H. Geng and J. Ferosh, A GPU-accelerated direct-sum boundary integral Poisson-Boltzmann solver. *Comput. Phys. Commun.* 184 (2007) 1490-1496.
- [18] M.J. Holst, Multilevel methods for the Poisson-Boltzmann equation, Ph.D. thesis, University of Illinois at Urbana-Champaign, Illinois, USA, 1993.
- [19] M.J. Holst and F. Saied, Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods. *J. Comput. Chem.* 16 (1995) 337-364.
- [20] W. Im, D. Beglov and B. Roux, Continuum solvation model: Computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation. *Comput. Phys. Commun.* 111 (1998) 59-75.
- [21] A. Juffer, E. Botta, B. van Keulen, A. van der Ploeg, H. Berendsen, The electric potential of a macromolecule in a solvent: a fundamental approach. *J. Comput. Phys.* 97 (1991) 144-171.

- [22] J.G. Kirkwood, Theory of solution of molecules containing widely separated charges with special application to Zwitterions. *J. Chem. Phys.* 7 (1934) 351–361.
- [23] B. Lee and F.M. Richards, Interpretation of protein structures: estimation of static accessibility. *J. Mol. Biol.* 55 (1973) 379–400.
- [24] B.Z. Lu, X.L. Cheng, J.F. Huang, J.A. McCammon, Order N algorithm for computation of electrostatic interactions in biomolecular systems. *Proceedings of the National Academy of Sciences* 103 (2006) 19314–19319.
- [25] R. Luo, L. David, M.K. Gilson, Accelerated Poisson-Boltzmann calculations for static and dynamic systems. *J. Comput. Chem.* 23 (2002) 1244–1253.
- [26] B.A. Luty, M.E. Davis, and J.A. McCammon, Solving the finite-difference nonlinear Poisson-Boltzmann equation. *J. Comput. Chem.* 13 (1992) 1114–1118.
- [27] Z.H. Qiao, Z.L. Li, T. Tang, A finite difference scheme for solving the nonlinear Poisson-Boltzmann equation modeling charged spheres. *J. Comput. Math.* 24 (2006) 252–264.
- [28] F.M. Richards, Areas, volumes, packing and protein structure. *Annu. Rev. Biophys. Bioeng.* 6 (1977), 151–176.
- [29] W. Rocchia, E. Alexov, and 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.
- [30] S. A. Sauter and C. Schwab, *Boundary Element Methods*, Springer, 2010.
- [31] A. Sayyed-Ahmad, K. Tuncay, and P.J. Ortoleva, Efficient solution technique for solving the Poisson-Boltzmann equation. *J. Comput. Chem.* 25 (2004) 1068–1074.
- [32] A.I. Shestakov, J.L. Milovich, and 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.
- [33] J.C. Strikwerda, *Finite difference schemes and partial differential equations*, Second Edition, SIAM, 2004.
- [34] G.W. Wei, Differential geometry based multiscale models. *Bullet. Math. Bio.* 72 (2010) 1562–1622.
- [35] R. Yokota, J.P. Bardhan, M.G. Knepley, L.A. Barba, T. Hamada, Biomolecular electrostatics using a fast multipole BEM on up to 512 GPUS and a billion unknowns. *Comput. Phys. Commun.* 182 (2011) 1272–1283.
- [36] S. Yu, S. Zhao, and G.W. Wei, Local spectral time-splitting method for first and second order partial differential equations. *J. Comput. Phys.* 206 (2005) 727–780.
- [37] S. Zhao, Pseudo-time coupled nonlinear models for biomolecular surface representation and solvation analysis. *Int. J. Numer. Method Bio. Engrg.* 27 (2011) 1964–1981.
- [38] S. Zhao, Operator splitting ADI schemes for pseudo-time coupled nonlinear solvation simulations. *J. Comput. Phys.* (2013) submitted.
- [39] S. Zhao, J. Ovadia, X. Liu, Y.-T. Zhang, and Q. Nie, Operator splitting implicit integration factor methods for stiff reaction-diffusion-advection systems. *J. Comput. Phys.* 230 (2011) 5996–6009.
- [40] Y.C. Zhou, S. Zhao, M. Feig and 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.