UNCONDITIONALLY STABLE TIME SPLITTING METHODS FOR THE ELECTROSTATIC ANALYSIS OF SOLVATED BIOMOLECULES

LEIGHTON WILSON AND SHAN ZHAO

(Communicated by Xiaoming He)

Abstract. This work introduces novel unconditionally stable operator splitting methods for solving the time dependent nonlinear Poisson-Boltzmann (NPB) equation for the electrostatic analysis of solvated biomolecules. In a pseudo-transient continuation solution of the NPB equation, the nonlinear term is analytically integrated, so that the difficulties in direct treatment of the strong nonlinearity can be bypassed. However, in a pseudo-time NPB computation, the use of large time increments is necessary to reach the steady state efficiently. The existing alternating direction implicit (ADI) methods for the transient NPB equation are known to be conditionally stable, although being fully implicit. To overcome this difficulty, we propose several new operator splitting schemes, in both multiplicative and additive styles, including locally one-dimensional (LOD) schemes and additive operator splitting (AOS) schemes. The proposed schemes become much more stable than the ADI methods, and some of them are indeed unconditionally stable in dealing with solvated proteins with source singularities and non-smooth solutions. By using finite differences in space and implicit integrations in time, the numerical orders of the proposed schemes are found to be one in both space and time. Nevertheless, the precision in calculating the electrostatic free energy is low, unless a small time increment is used. Further accuracy improvements are thus considered, through constructing a Richardson extrapolation procedure and a tailored recovery scheme in treating the vacuum case. After acceleration, the optimized LOD method can produce a reliable energy estimate by integrating for a small and fixed number of time steps. Since one only needs to solve a tridiagonal matrix in each one dimensional subsystem, the overall computation is very efficient. The unconditionally stable LOD method scales linearly with respect to the number of atoms in the protein studies, and is over 20 times faster than the conditionally stable ADI methods.

Key words. Nonlinear Poisson-Boltzmann equation, pseudo-transient continuation approach, time splitting, alternating direction implicit (ADI), locally one dimensional (LOD), additive operator splitting (AOS), electrostatic free energy.

1. Introduction

Analysis of the underlying biomolecular solvation is critical when carrying out quantitative descriptions of various important biological processes at the atomic level, such as protein folding and protein ligand bonding, DNA recognition, transcription, and translation. From a biological perspective, solvation analysis is concerned with interactions between a solute macromolecule and surrounding solvent ions. From a mathematical perspective, these solute-solvent interactions may be represented via solvation energies with contributions from polar and nonpolar sources. The polar portion arises from electrostatic interactions, which may be represented with the Poisson-Boltzmann (PB) model [6, 14]. The PB model provides a framework by which to model the distribution of electrostatic potential along the surface of a solute macromolecule within a surrounding solvent with a particular ionic concentration.

TABLE 1. Acronyms used in this paper.

ADI	alternating direction implicit	LOD	locally one-dimensional
AOS	additive operator splitting	IE	implicit Euler
CFL	Courant-Friedrich-Lewy	NPB	nonlinear Poisson Boltzmann
CN	Crank-Nicolson	PB	Poisson Boltzmann
FFT	fast Fourier transform	RE	Richardson extrapolation
MAOS	multiplicative-additive operator splitting		

In the PB model, the PB equation governing electrostatic potentials takes the form of a nonlinear elliptic equation on multiple domains with discontinuous dielectric coefficients across the molecular surface or solute-solvent interface [3, 4]. The PB equation cannot be solved analytically for molecules with complex geometries, only admitting analytical solutions for shapes such as spheres or rods [12, 16]. However, solving the PB equation numerically also presents significant difficulties because of the discontinuous dielectric coefficients, singularities in the source-term, non-smoothness of the solution, and significant nonlinearity when strong ionic effects are present.

Recently, a pseudo-transient continuation approach has been proposed for solving the nonlinear PB (NPB) equation [27, 28, 34], which creates a different way to tackle the nonlinear term of the NPB equation. In classical finite difference and finite element solutions of the NPB equation, a nonlinear algebraic system is typically formed through the discretization of the boundary value problem. A nonlinear relaxation method [15, 24] or inexact Newton method [13] can be employed to solve such a nonlinear system. In the pseudo-transient continuation approach [27, 28, 34], a pseudo-time derivative is added to the NPB equation so that one solves an initial-boundary value problem now. The steady state solution of this problem gives rise to the solution to the original boundary value problem. Numerically, it is desired that a large time step can be used so that the steady state can be computed quickly. Thus, the efficiency of a pseudo-time NPB solver is directly related to its stability, which critically depends on the nonlinear term of the NPB equation – a hyperbolic sine function that could be exponentially large.

Several time stepping schemes have been considered for solving the time dependent NPB equation. The explicit Euler solution is straightforward, but invokes a severe stability constraint [34]. Implicit time integrations have also been studied [27, 28], for which care has to be exercised in handling the nonlinear term. In [27], a linearization technique based on the first order Taylor expansion is proposed so that a linear system is formed at each step of the implicit Euler integration. This linearization essentially evaluates the nonlinear term at the previous time instant. Similarly, by treating the nonlinear term explicitly, a modified alternating direction implicit (ADI) method has been introduced in [28]. Since the Thomas algorithm [21] can be employed to solve the tridiagonal finite difference systems in this time splitting method, the efficiency is greatly improved. However, a very large time increment is still prohibited in these methods, because these implicit schemes are of semi-implicit nature.

More recently, we have successfully developed two fully-implicit ADI schemes [11, 35] for solving the time dependent NPB equation. The success lies in an analytical integration of the nonlinear term, and the use of a time splitting framework. This completely suppresses the nonlinear instability, so that these fully-implicit ADI

schemes are unconditionally stable in solving benchmark problems with smooth solutions. However, for the solvation analysis in applications to real biomolecules, these ADI schemes are still conditionally stable, probably because of the singularities in the charges and discontinuous dielectric coefficients. Nevertheless, these fully-implicit ADI schemes are still found to be quite efficient, due to their high temporal accuracy and much relaxed Courant-Friedrich-Lewy (CFL) constraints [11, 35]. Moreover, the analytical treatment completely avoids the difficulties in treating the strong nonlinearity of the time independent NPB equation so that the pseudo-time NPB solver really becomes a viable approach for electrostatic analysis.

The objective of this paper is to introduce various new operator splitting methods for solving the time dependent NPB equation. The present investigation will be conducted in the same pseudo-transient continuation framework as in [11, 35]. In particular, in spatial approximation, central finite difference will be employed to discretize the nonhomogeneous diffusion operator, which yields tridiagonal matrices in all dimension-splitting schemes. In temporal approximation, the analytical integration will be carried out for the split nonlinear subsystem. The ultimate goal of this work is to eventually construct unconditionally stable operator splitting methods for solving the NPB equation in practical biomolecular applications.

In the existing fully-implicit operator splitting NPB solvers [11, 35], the ADI schemes are constructed based on the classical ADI schemes [8, 7, 22]. In the literature, there exists another family of multiplicative operator splitting methods, i.e., the fractional step methods or locally one-dimensional (LOD) methods, originally introduced by Russian mathematicians [9, 31, 32]. In the present setup, the ADI discretizations can be written as some perturbations of multidimensional discretizations of the implicit methods, such as the Crank-Nicolson and backward Euler. In the proposed LOD methods, the similar splitting in alternating directions will be considered, but before the time discretization. In other words, the Crank-Nicolson and backward Euler discretizations will be conducted after the differential equations are split. Thus, the discretizations of the LOD methods are truly one-dimensional. This is different from the ADI methods, in which the right-hand side of the discretized equations contains derivatives from the other directions. Comparing with the ADI methods, the LOD methods tend to be more stable, but have a larger splitting error [21]. Moreover, a previous work on the application of ADI and LOD schemes to Maxwell's equations in isotropic and lossless media found that the LOD schemes were approximately 20% computationally less expensive than the ADI schemes [1]. Other comparisons of LOD schemes and traditional ADI schemes as applied to Maxwell's equations have also found lower computational costs for the LOD methods over ADI methods [10, 33]. This motivates us to develop several LOD schemes for solving the time dependent NPB equation in this work.

Besides the ADI and LOD methods, there exists another family of operator splitting methods, i.e., the additive operator splitting (AOS) methods [18, 19, 29]. In the AOS methods, the time integration of each split subsystem can be independently carried out, while in the multiplicative operator splitting methods, the integration of the present fractional step needs the solution from the previous fractional steps. Thus, the AOS methods are well suited to be implemented in parallel processors. The stability and accuracy of the AOS methods have been analyzed in [5]. The application of AOS schemes has proved popular for the purposes of nonlinear diffusion filtering for image processing [2, 30]. The AOS schemes are usually efficient and stable for relatively large Δt values, although potentially less accurate than the LOD schemes. However, their overall efficiency and simplicity merit further

exploration. In the present paper, different formulations and discretizations of the AOS methods for solving time dependent NPB equation will be considered. Previous work in image processing has also explored the properties of schemes combining multiplicative and additive operator splitting, finding higher levels of accuracy with such schemes than with simpler AOS methods [2]. We thus will also explore the construction and application of a multiplicative-additive operator splitting (MAOS) scheme for our current problem.

The rest of this paper is organized as follows. In Section 2, we first introduce the physical background of the problem. The pseudo-transient PB model will then be presented. The existing ADI schemes will be described and the remaining difficulties are discussed. The proposed operator splitting schemes will then be formulated for solving the time dependent NPB equation in Section 3. Numerical validations of the proposed schemes through benchmark examples with analytical solutions are considered in Section 4, while various benchmark biological systems and subsequent free energy of solvation calculations are presented in Section 5. Finally, this paper ends with a conclusion.

2. Mathematical models and existing algorithms

2.1. Governing equation. Consider a solute macromolecule in space surrounded by a solvent aqueous solution. As defined in [23], the molecular surface, or solute-solvent boundary Γ, of the macromolecule divides $\mathbb{R}^3$ into the closed domain of the solute molecule Ω_m and solvent domain Ω_s. The electrostatic interaction of such a solute-solvent system is governed by the nonlinear Poisson-Boltzmann (NPB) equation

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

where u is the electrostatic potential and the source term ρ_m is defined as

$$(2) \qquad \rho_m(\mathbf{r}) = \frac{4\pi e_c^2}{k_b T}\sum_{j=1}^{N_m} q_j \delta(\mathbf{r} - \mathbf{r}_j).$$

The dielectric constant ϵ is piecewise such that $\epsilon(\mathbf{r}) = \epsilon_m$ for $\mathbf{r} \in \Omega_m$ and $\epsilon(\mathbf{r}) = \epsilon_s$ for $\mathbf{r} \in \Omega_s$. Here k_b is the Boltzmann constant, e_c is the fundamental charge, and q_j, in the same units as e_c, is the partial charge on the jth atom of the solute macromolecule locate at position $\mathbf{r}_j$. The modified Debye-Huckel parameter $\bar{\kappa}$ is defined as $\bar{\kappa}^2(\mathbf{r}) = \left(\frac{2N_A e_c^2}{1000 k_b T}\right) I_s$, for $\mathbf{r} \in \Omega_s$ and $\bar{\kappa} = 0$ for $\mathbf{r} \in \Omega_m$, where N_A is Avogadro's Number and I_s is the molar ionic strength. At room temperature ($T = 298K$), $\bar{\kappa}^2 = 8.486902807\text{Å}^{-2}I_s$ from [12]. The NPB equation varies on the choice of units, and in the present setting we utilize the dimensionless form presented in [13]. We may convert the dimensionless electrostatic potential u to units kcal/mol/e_c at room temperature ($T = 298K$) through simply multiplying the potentials by 0.592183 [12]. Our numerical computations must take place on a finite domain Ω. We may assign values along the boundary $\partial\Omega$ according to the approximate analytical condition

$$(3) \qquad u(\mathbf{r}) = \frac{e_c^2}{k_b T}\sum_{i=1}^{N_m}\frac{q_i}{\epsilon_s|\mathbf{r} - \mathbf{r}_i|}e^{-\bar{\kappa}\frac{|\mathbf{r} - \mathbf{r}_i|}{\sqrt{\epsilon_s}}}.$$

When $\partial\Omega$ is of sufficient distance from the macromolecule subdomain Ω_m, Eq. (3) can be utilized to approximate the results for potentials found from Eq. (1). We note here that Eq. (3), for a collection of N_m partial charges q_i at positions $\mathbf{r}_i$, is

simply a linear superposition of Coulomb's Law. For simplicity, the boundary $\partial\Omega$ is assumed to be of a cubic shape.

2.2. Pseudo-time dependent NPB equation. The computational simulation of the NPB equation is of great importance for biomolecular modeling, but is quite challenging. Numerous approaches have been developed in the literature; see for example recent reviews [20, 17] and references therein. What are related to the present study is the pseudo-transient continuation approach for solving the NPB equation [27, 28, 34]. In this approach, the NPB equation (1) will be converted from its time independent form to a time dependent form by introducing a pseudo-transient variation, thus producing the time dependent NPB:

$$(4) \qquad \alpha\frac{\partial u}{\partial t}(\mathbf{r}, t) = \nabla \cdot (\epsilon(\mathbf{r})\nabla u(\mathbf{r}, t)) - \bar{\kappa}^2(\mathbf{r})\sinh(u(\mathbf{r}, t)) + \rho_m(\mathbf{r}).$$

where α is a scaling parameter. To solve this initial boundary value problem, we must first specify an initial solution for our potential values, which may be either the trivial solution $u = 0$, or the electrostatic potential solved from the linear Poisson-Boltzmann equation [34]. Equation (4) is then numerically integrated over a sufficiently large time period to reach a steady state solution, thus essentially recovering the solution to the original NPB equation (1).

2.3. Previous schemes and remaining difficulties. Several time integration methods have been developed for solving the time dependent NPB equation (4). A very small time increment Δt is usually required in the explicit Euler solution [34], which is inefficient for practical use. However, the construction of a fully implicit integration of the NPB equation is hindered by the presence of the nonlinear hyperbolic sine term in the NPB equation. By evaluating the nonlinear term at the previous time step, the regular implementation of the implicit schemes usually gives rise to a semi-implicit integration [27, 28].

A breakthrough has been made in our recent study [35], in which fully implicit alternating direction implicit (ADI) methods were developed for the first time in the literature. A generalized NPB equation was considered in [35], while the similar ADI schemes for the classical NPB equation (1) were introduced later in [11]. The success of these ADI schemes lies in an analytical integration of the nonlinear term in a time splitting framework. With this analytical treatment of the nonlinear term, unconditionally stable results are obtained in benchmark problems with smooth solutions. Unfortunately, these operator splitting ADI schemes are still conditionally stable when they are applied to real protein systems, because of various complex features of the biomolecular solvation analysis, including complicated molecular surfaces, singular source charges, discontinuous dielectric coefficients, and nonsmooth electrostatic potentials [11, 35]. We thus are motivated to develop novel numerical schemes which may be both computationally less expensive as well as more stable for significantly larger Δt values than the previously presented ADI schemes.

For a comparison, two ADI schemes presented [11] will also be examined in this work. We thus briefly review these two methods here. In the first order ADI scheme (*ADI1*) [11], at each time step from t_n to t_{n+1}, the time dependent NPB equation (4) is solved by a first order time splitting method in two stages. An analytical integration of the nonlinear term is conducted in the first stage. The nonhomogeneous diffusion equation of the second stage is first discretized by the implicit Euler scheme in time and the finite difference method in space. The multidimensional system is then solved by a classical Douglas-Rachford ADI scheme, by splitting the system into many one-dimensional (1D) systems of a tridiagonal

structure. Similarly, in the second order ADI scheme (*ADI2*) [11], a second order time splitting with three stages is considered with analytical treatments in the first and last stages. The Crank-Nicolson scheme is considered in the second stage, and is realized via a Douglas ADI implementation. The *ADI2* scheme is generally more accurate than the *ADI1* scheme, though it is computationally more expensive [11].

We finally note that the direct treatment of the nonlinear term is one of the major difficulties in numerical solution of the NPB equation, especially when the underlying electrostatic potential is strong [20, 17]. By using an analytical treatment, the pseudo-transient continuation approach avoids such a difficulty so that the ADI NPB solvers become viable methods for electrostatic analysis.

3. Towards the development of unconditionally stable schemes

The time splitting or operator splitting methods are powerful tools for solving time dependent partial differential equations. Besides the ADI schemes, there exist many other operator splitting methods in the literature that can reduce a multidimensional problem to sets of independent 1D problems. With the central difference approximation in space, these 1D systems will have tridiagonal structures, so that they can be efficiently solved using the Thomas algorithm [21]. Several commonly used operator splitting methods will be explored in this paper, for the purpose of eventually developing unconditionally stable schemes.

We adopt the following notations in this work. Following the tradition of the biomolecular simulation, we consider a uniform mesh with grid spacing h in the x, y, and z directions, with N_x, N_y, N_z being the number of grip points in each direction, respectively. The time increment is denoted as Δt. The notation $u_{i,j,k}^n = u(x_i, y_j, z_k, t_n)$ denotes the electrostatic potential at a node (x_i, y_j, z_k) and a time level $t = t_n$. Thus, the vector $\mathbf{U}^n = \{u_{i,j,k}^n\}$ for $i = 1, ..., N_x$, $j = 1, ..., N_y$, and $k = 1, ..., N_z$ denotes all nodal values of u at t_n. All proposed schemes will be focused on updating $\mathbf{U}^n$ at a time level t_n to $\mathbf{U}^{n+1}$ at next time level $t_{n+1} = t_n + \Delta t$.

3.1. Locally-one-dimensional (LOD) schemes.
The fractional step methods or locally one-dimensional (LOD) methods were first developed by Russian mathematicians [9, 31, 32]. Being multiplicative operator splitting methods too, the LOD methods adopt the similar splitting in alternating directions as in the ADI methods, but before the numerical discretization.

In the present study, we propose several LOD schemes to solve the time dependent NPB equation (4). The spatial and temporal discretization will be detailed for one LOD scheme, while the discretization of other schemes can be similarly constructed. In particular, in the first LOD scheme, at each time step from t_n to t_{n+1}, the time dependent NPB equation will be solved by a multiplicative operator splitting procedure consisting of five stages:

$$(5) \qquad \alpha \frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(6) \qquad \alpha \frac{\partial v}{\partial t} = \frac{\partial}{\partial x}\left(\epsilon \frac{\partial v}{\partial x}\right), \quad \text{with } \mathbf{V}^n = \mathbf{W}^{n+1}, \quad t \in [t_n, t_{n+1}],$$

$$(7) \qquad \alpha \frac{\partial p}{\partial t} = \frac{\partial}{\partial y}\left(\epsilon \frac{\partial p}{\partial y}\right), \quad \text{with } \mathbf{P}^n = \mathbf{V}^{n+1}, \quad t \in [t_n, t_{n+1}],$$

$$(8) \qquad \alpha \frac{\partial q}{\partial t} = \frac{\partial}{\partial z}\left(\epsilon \frac{\partial q}{\partial z}\right), \quad \text{with } \mathbf{Q}^n = \mathbf{P}^{n+1}, \quad t \in [t_n, t_{n+1}],$$

$$(9) \qquad \alpha \frac{\partial r}{\partial t} = \rho_m, \quad \text{with } \mathbf{R}^n = \mathbf{Q}^{n+1}, \quad t \in [t_n, t_{n+1}],$$

We finally have $\mathbf{U}^{n+1} = \mathbf{R}^{n+1}$. The global splitting error of the present LOD scheme is of order one, i.e., $O(\Delta t)$. Obviously, no discretization is involved in the present splitting, which is different from the previous ADI methods [11, 35].

In the LOD schemes, the numerical approximation is introduced after the splitting. Here an analytical integration can be conducted for the nonlinear equation (5), as in our previous studies [11, 35]. In particular, Eq. (5) is simply a separable ordinary differential equation in this context:

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

Thus, the integration can be carried out analytically:

$$(10) \qquad\qquad -2\tanh^{-1}(e^w) = -\bar{\kappa}^2 \frac{t}{\alpha} + C$$

Evaluating Eq. (10) at both t_n and t_{n+1}, we have

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

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

$$(11) \qquad\qquad \exp(\mathbf{W}^{n+1}) = \tanh(\tanh^{-1}(\exp(\mathbf{W}^n)) + \frac{1}{2\alpha}\bar{\kappa}^2 \Delta t)$$

Equation (11) can then be simplified to be

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

Therefore, given $\mathbf{W}^n$ at t_n, $\mathbf{W}^{n+1}$ can be calculated analytically according to (12) so that the difficulties associated with the $\sinh(\cdot)$ nonlinear term of the NPB equation may be bypassed.

Different LOD schemes can be obtained if we consider different time stepping procedures. We first propose the use of implicit Euler integration in time and central differencing in space to discretize Equations (6) through (8). Combining with our analytical solution for the $\sinh(\cdot)$ term (12), we formulate our first scheme, *LODIE1*:

$$(13) \qquad w_{i,j,k} = \ln\left(\frac{\cosh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t) + \exp(-u^n_{i,j,k})\sinh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t)}{\exp(-u^n_{i,j,k})\cosh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t) + \sinh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t)}\right),$$

$$(14)\ (1 - \frac{\Delta t}{\alpha}\delta_x^2)v_{i,j,k} = w_{i,j,k},$$

$$(15)\ (1 - \frac{\Delta t}{\alpha}\delta_y^2)p_{i,j,k} = v_{i,j,k},$$

$$(16)\ (1 - \frac{\Delta t}{\alpha}\delta_z^2)q_{i,j,k} = p_{i,j,k},$$

$$(17) \qquad u^{n+1}_{i,j,k} = q_{i,j,k} + \frac{\Delta t}{\alpha}Q(x_i, y_j, z_k),$$

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

$$\delta_x^2 v^n_{i,j,k} = \frac{1}{h^2}\left(\epsilon(x_{i+\frac{1}{2}}, y_j, z_k)(v^n_{i+1,j,k} - v^n_{i,j,k}) + \epsilon(x_{i-\frac{1}{2}}, y_j, z_k)(v^n_{i-1,j,k} - v^n_{i,j,k})\right),$$

$$\delta_y^2 v^n_{i,j,k} = \frac{1}{h^2}\left(\epsilon(x_i, y_{j+\frac{1}{2}}, z_k)(v^n_{i,j+1,k} - v^n_{i,j,k}) + \epsilon(x_i, y_{j-\frac{1}{2}}, z_k)(v^n_{i,j-1,k} - v^n_{i,j,k})\right),$$

$$\delta_z^2 v^n_{i,j,k} = \frac{1}{h^2}\left(\epsilon(x_i, y_j, z_{k+\frac{1}{2}})(v^n_{i,j,k+1} - v^n_{i,j,k}) + \epsilon(x_i, y_j, z_{k-\frac{1}{2}})(v^n_{i,j,k-1} - v^n_{i,j,k})\right).$$

Furthermore, $Q(x_i, y_j, z_k)$ is the distribution of all source charges in the source term ρ_m from Equation (2), distributed by a trilinear interpolation. The value of ϵ is determined by its location which is either on/inside or outside the molecular surface Γ. Specifically, $\epsilon(x, y, z) = \epsilon_m$ if $(x, y, z) \in \Omega_m$, and $\epsilon(x, y, z) = \epsilon_s$ if $(x, y, z) \in \Omega_s$, where the molecular surface Γ is determined by a commonly used software package: MSMS [25]. Both $Q(x_i, y_j, z_k)$ and $\epsilon(x_i, y_j, z_k)$ are time independent and need to be calculated only once at the beginning. Since the boundary condition of the steady state solution u, i.e., Eq. (3), is time independent, the split components w, v, p, and q are assumed to attain the same Dirichlet boundary values as u. The initial solution will be chosen as either a trivial solution $u = 0$, or the electrostatic potential solved from the linear Poisson-Boltzmann equation [34]. We utilize similar assumptions for our numerical simulations with all following methods.

We further formulate a second scheme based on the LOD1 scheme that utilizes Crank-Nicolson integration in time and central differencing in space. This scheme, *LODCN1*, replaces Equations (14) through (16) from *LODIE1* with the following:

$$(18) \qquad (1 - \frac{1}{2\alpha}\Delta t \delta_x^2)v_{i,j,k} = (1 + \frac{1}{2\alpha}\Delta t \delta_x^2)w_{i,j,k},$$

$$(19) \qquad (1 - \frac{1}{2\alpha}\Delta t \delta_y^2)p_{i,j,k} = (1 + \frac{1}{2\alpha}\Delta t \delta_y^2)v_{i,j,k},$$

$$(20) \qquad (1 - \frac{1}{2\alpha}\Delta t \delta_z^2)q_{i,j,k} = (1 + \frac{1}{2\alpha}\Delta t \delta_z^2)p_{i,j,k}$$

Minor changes can be resulted if we alter the order of the subsystems in the LOD splitting. To illustrate this, we also propose a second set of LOD schemes that solve the five stages presented in (5) through (9) in a modified order: the source term equation in (9) is treated first, while the nonlinear equation in (5) is treated last. This LOD method, similar to the first one, has a form utilizing implicit Euler integration in time and another form utilizing Crank-Nicolson integration in time. *LODIE2*, then, simply consists of equations (17), (14), (15), (16), (13), respectively. *LODCN2* consists of (17), (18), (19), (20), (13), respectively.

We conclude this subsection by presenting some theoretical results of the proposed LOD schemes.

Remark 3.1. *The proposed LODIE1, LODIE2, LODCN1, and LODCN2 schemes are of first order accuracy in time, because the underlying LOD splittings are first order and the discretization error of the implicit Euler or Crank-Nicolson integration is at least first order.*

Remark 3.2. *If the solution u is sufficiently smooth, the proposed LODIE1, LODIE2, LODCN1, and LODCN2 schemes are unconditionally stable, because each individual implicit Euler or Crank-Nicolson time integration with central difference approximation is unconditionally stable.*

3.2. Additive operator splitting (AOS) schemes. We next propose a series of schemes utilizing an additive operator splitting (AOS) formulation [18, 19, 29]. Unlike the ADI and LOD schemes, in which the subsystems have to be solved sequentially, the split equations can be solved concurrently in the AOS schemes. Thus, the AOS methods are well suited to be implemented in parallel processors. Since the present work focuses on the stability investigation, a simple series implementation is still conducted for the AOS schemes.

In the first AOS scheme, at each time step from t_n to t_{n+1}, we solve the time dependent NPB equation in the following stages:

$$(21) \quad \alpha\frac{\partial w}{\partial t} = -4\bar{\kappa}^2\sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(22) \quad \alpha\frac{\partial v}{\partial t} = 4\frac{\partial}{\partial x}\left(\epsilon\frac{\partial v}{\partial x}\right) + 4\rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(23) \quad \alpha\frac{\partial p}{\partial t} = 4\frac{\partial}{\partial y}\left(\epsilon\frac{\partial p}{\partial y}\right), \quad \text{with } \mathbf{P}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(24) \quad \alpha\frac{\partial q}{\partial t} = 4\frac{\partial}{\partial z}\left(\epsilon\frac{\partial q}{\partial z}\right), \quad \text{with } \mathbf{Q}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(25) \quad \mathbf{U}^{n+1} = \frac{1}{4}\left(\mathbf{W}^{n+1} + \mathbf{V}^{n+1} + \mathbf{P}^{n+1} + \mathbf{Q}^{n+1}\right).$$

In other words, the NPB system is split into four parts, i.e., one nonlinear subsystem and three subsystems along Cartesian directions. The time independent source term is assigned to the x-direction subsystem. Then an arithmetic average is carried out to advance $\mathbf{U}^n$ to $\mathbf{U}^{n+1}$.

Similar to our LOD schemes, we will formulate two types of time discretizations. Using implicit Euler integration in time and central differencing in space to discretize Equations (22) through (24), we propose first the scheme *AOSIE1*:

$$(26) \quad w_{i,j,k} = 4\ln\left(\frac{\cosh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t) + \exp(-u_{i,j,k}^n)\sinh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t)}{\exp(-u_{i,j,k}^n)\cosh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t) + \sinh(\frac{1}{2\alpha}\bar{\kappa}^2\Delta t)}\right),$$

$$(27) \quad (1 - \frac{4}{\alpha}\Delta t\delta_x^2)v_{i,j,k} = u_{i,j,k}^n + \frac{4}{\alpha}\Delta t\rho_m,$$

$$(28) \quad (1 - \frac{4}{\alpha}\Delta t\delta_y^2)p_{i,j,k} = u_{i,j,k}^n,$$

$$(29) \quad (1 - \frac{4}{\alpha}\Delta t\delta_z^2)q_{i,j,k} = u_{i,j,k}^n,$$

$$(30) \quad u_{i,j,k}^{n+1} = \frac{1}{4}\left(w_{i,j,k} + v_{i,j,k} + p_{i,j,k} + q_{i,j,k}\right)$$

We also formulate a second scheme based on the AOS1 scheme that utilizes Crank-Nicolson integration in time and central differencing in space. This scheme, *AOSCN1*, replaces Equations (27) through (29) from *AOSIE1* with the following:

$$(31) \quad (1 - \frac{2}{\alpha}\Delta t\delta_x^2)v_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_x^2)u_{i,j,k}^n + \frac{4}{\alpha}\Delta t\rho_m,$$

$$(32) \quad (1 - \frac{2}{\alpha}\Delta t\delta_y^2)p_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_y^2)u_{i,j,k}^n,$$

$$(33) \quad (1 - \frac{2}{\alpha}\Delta t\delta_z^2)q_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_z^2)u_{i,j,k}^n$$

For the AOS schemes, since each subsystem is solved independently, any change in their order will not alter the numerical outcome. In the present study, we consider a different variation for the proposed AOS schemes. In the *AOSCN1* scheme, the source term is imposed only along the x direction. We are interested in a more symmetric splitting by introducing a second set of AOS schemes. At each time step

from t_n to t_{n+1}, we solve the time dependent NPB equation in the following stages:

$$(34) \qquad \alpha \frac{\partial w}{\partial t} = -4\bar{\kappa}^2 \sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(35) \qquad \alpha \frac{\partial v}{\partial t} = 4\frac{\partial}{\partial x}\left(\epsilon \frac{\partial v}{\partial x}\right) + \frac{4}{3}\rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(36) \qquad \alpha \frac{\partial p}{\partial t} = 4\frac{\partial}{\partial y}\left(\epsilon \frac{\partial p}{\partial y}\right) + \frac{4}{3}\rho_m, \quad \text{with } \mathbf{P}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(37) \qquad \alpha \frac{\partial q}{\partial t} = 4\frac{\partial}{\partial z}\left(\epsilon \frac{\partial q}{\partial z}\right) + \frac{4}{3}\rho_m, \quad \text{with } \mathbf{Q}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$(38) \qquad \mathbf{U}^{n+1} = \frac{1}{4}\left(\mathbf{W}^{n+1} + \mathbf{V}^{n+1} + \mathbf{P}^{n+1} + \mathbf{Q}^{n+1}\right)$$

It is obvious that we split the source term into three Cartesian subsystems so that certain symmetry is maintained in three directions. It is noted that we do not distribute the source term into the nonlinear subsystem, because the time integration of the resulting nonlinear process becomes quite involved.

We similarly formulate two schemes based on the AOS2 method, one utilizing implicit Euler integration in time and another utilizing Crank-Nicolson integration in time. Both of these schemes largely take a similar form as the *AOSIE1* scheme; for *AOSIE2*, Equations (27) through (29) are replaced with the following:

$$(39) \qquad (1 - \frac{4}{\alpha}\Delta t\delta_x^2)v_{i,j,k} = u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m,$$

$$(40) \qquad (1 - \frac{4}{\alpha}\Delta t\delta_y^2)p_{i,j,k} = u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m,$$

$$(41) \qquad (1 - \frac{4}{\alpha}\Delta t\delta_z^2)q_{i,j,k} = u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m,$$

and similarly for the *AOSCN2* scheme, we utilize the following series of equations:

$$(42) \qquad (1 - \frac{2}{\alpha}\Delta t\delta_x^2)v_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_x^2)u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m,$$

$$(43) \qquad (1 - \frac{2}{\alpha}\Delta t\delta_y^2)p_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_y^2)u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m,$$

$$(44) \qquad (1 - \frac{2}{\alpha}\Delta t\delta_z^2)q_{i,j,k} = (1 + \frac{2}{\alpha}\Delta t\delta_z^2)u_{i,j,k}^n + \frac{4}{3\alpha}\Delta t\rho_m.$$

The error and stability analysis of the general AOS schemes can be found in [5]. For the present AOS schemes, we have the following results.

Remark 3.3. *The proposed AOSIE1, AOSIE2, AOSCN1, and AOSCN2 schemes are of first order accuracy in time, because the underlying AOS splittings are first order and the discretization error of the implicit Euler or Crank-Nicolson integration is at least first order.*

Remark 3.4. *If the solution u is sufficiently smooth, the proposed AOSIE1, AOSIE2, AOSCN1, and AOSCN2 schemes are unconditionally stable, because each individual implicit Euler or Crank-Nicolson time integration with central difference approximation is unconditionally stable.*

3.3. Multiplicative-additive operator splitting (MAOS) schemes. Finally, we propose two hybrid schemes combining multiplicative and additive operator splitting stages. Our intention is to treat the nonlinear subsystem separately from the three linear Cartesian subsystems. So, at each time step from t_n to t_{n+1}, the time dependent NPB equation will be solved by a two-stage multiplicative operator

splitting scheme, the second of which is solved with an additive operator splitting scheme:

$$\tag{45} \alpha\frac{\partial w}{\partial t} = -\bar{\kappa}^2 \sinh(w), \quad \text{with } \mathbf{W}^n = \mathbf{U}^n, \quad t \in [t_n, t_{n+1}],$$

$$\tag{46} \alpha\frac{\partial v}{\partial t} = 3\frac{\partial}{\partial x}\left(\epsilon\frac{\partial v}{\partial x}\right) + \rho_m, \quad \text{with } \mathbf{V}^n = \mathbf{W}^n, \quad t \in [t_n, t_{n+1}],$$

$$\tag{47} \alpha\frac{\partial p}{\partial t} = 3\frac{\partial}{\partial y}\left(\epsilon\frac{\partial p}{\partial y}\right) + \rho_m, \quad \text{with } \mathbf{P}^n = \mathbf{W}^n, \quad t \in [t_n, t_{n+1}],$$

$$\tag{48} \alpha\frac{\partial q}{\partial t} = 3\frac{\partial}{\partial z}\left(\epsilon\frac{\partial q}{\partial z}\right) + \rho_m, \quad \text{with } \mathbf{Q}^n = \mathbf{W}^n, \quad t \in [t_n, t_{n+1}],$$

$$\tag{49} \mathbf{U}^{n+1} = \frac{1}{3}\left(\mathbf{V}^{n+1} + \mathbf{P}^{n+1} + \mathbf{Q}^{n+1}\right)$$

Again, just as with the previous schemes, this MAOS method is discretized using implicit Euler integration in time, forming the *MAOSIE* scheme, and also discretized using Crank-Nicolson integration in time, forming the *MAOSCN* scheme.

By combining the previous studies, we have the following results for the MAOS schemes.

Remark 3.5. *The proposed MAOSIE and MAOSCN schemes are of first order accuracy in time, because the underlying MAOS splitting is first order and the discretization error of the implicit Euler or Crank-Nicolson integration is at least first order.*

Remark 3.6. *If the solution u is sufficiently smooth, MAOSIE and MAOSCN schemes are unconditionally stable, because each individual implicit Euler or Crank-Nicolson time integration with central difference approximation is unconditionally stable.*

4. Numerical validation

In this section, we validate the proposed schemes numerically by solving the NPB equation on a sphere – a case with an analytical solution. We will explore the stability as well as spatial and temporal convergence of the proposed time splitting schemes and compare these results to our previous ADI methods. Such studies will help us to identify well-performed NPB solvers to be used in real biomolecular simulations. All simulations are compiled with the Intel Fortran Compiler and run on an early-2011 MacBook Pro with an i7-2820QM 2.3GHz GPU and 8GB memory.

4.1. Benchmark problem. The NPB equation only possesses analytical solutions for certain simple geometries, such as a sphere. For verification of our schemes, we introduce the following solution for the case of a spherical cavity, based on a case from [26]:

$$\tag{50} u(\mathbf{r}) = \begin{cases} \frac{1}{\epsilon R} - \frac{1}{R} + \frac{1}{\|\mathbf{r}\|}, & \|\mathbf{r}\| < R \\ \frac{1}{\epsilon\|\mathbf{r}\|}, & \|\mathbf{r}\| > R \end{cases} \qquad \rho_m(\mathbf{r}) = \begin{cases} 4\pi\epsilon_m\delta(\mathbf{r}), & \|\mathbf{r}\| < R \\ \bar{\kappa}^2 \sinh\left(\frac{1}{\epsilon\|\mathbf{r}\|}\right), & \|\mathbf{r}\| > R \end{cases}$$

where $\epsilon = \epsilon_s/\epsilon_m$ and R is the radius of the spherical cavity. We note here that because of the singularity in the source term defined by (50) and the non-smoothness due to the interface jump conditions, the accuracy of the finite difference spatial discretization is normally reduced and additional instability may be introduced in the time-stepping. In our numerical validations, we choose $R = 1$Å with a single centered charge of 1 e_c. Our dielectric constants are $\epsilon_s = 80$ and $\epsilon_m = 1$, and we

set our nonlinear constant $\bar{\kappa} = 1$. Furthermore, for our scaling parameter, we select the standard value $\alpha = 1$.

A cubic domain $[-3, 3] \times [-3, 3] \times [-3, 3]$ with the same spacing in all three directions $h = \Delta x = \Delta y = \Delta z$ is used in our computations. A Dirichlet boundary condition is assumed on all boundaries with the boundary data being given by the analytical solution. For the purpose of investigating the convergence and stability of our new schemes, we will examine the sensitivity of the methods to different initial conditions. To this end, we construct a family of nontrivial initial solutions, which satisfy the Dirichlet boundary condition

$$(51) \qquad u(\mathbf{r}, 0) = H \cos(\frac{\pi}{6}x) \cos(\frac{\pi}{6}y) \cos(\frac{\pi}{6}z) + \frac{1}{\epsilon \sqrt{x^2 + y^2 + z^2}}.$$

Different magnitudes of H will be employed. Note when H is large, like $H = 20$, the value of the nonlinear term $\sinh(u)$ is on the order of 10^8. This imposes a great difficulty to explicit and semi-implicit time integration schemes. With such an initial solution at $t = 0$, the time-stepping will be carried out until a stopping time $t = T$ with a time increment Δt. Denoting u_h as the numerical solution, the following measures are used to estimate relative errors:

$$(52) \qquad L_\infty = \frac{\max |u - u_h|}{\max |u|}, \ \ L_2 = \sqrt{\frac{\Sigma_{i,j,k} |u - u_h|^2}{\Sigma_{i,j,k} |u|^2}}.$$

4.2. Stability. We first investigate the stability. For smooth solutions, the proposed time splitting methods are unconditionally stable. Nevertheless, the present analytical solution u is just C^0 continuous across the circular interface. It is thus of interest to test the stability of our newly constructed schemes for nonsmooth solutions.

Previously, the stability of several explicit and implicit schemes has been examined for solving the time dependent NPB equation, by using this benchmark example [11]. All tested schemes in that paper were found to be stable with a constraint $\Delta t \le \frac{h^2}{m}$ for some m. However, when the parameter H changes from 1 to 20, which greatly amplifies the nonlinear effect, all explicit and semi-implicit time schemes become unconditionally unstable, while the fully implicit ADI schemes still remain to be stable with a similar m value bounded by $m < 20$ [11].

In the present study, it is found numerically that the proposed time splitting methods are all unconditionally stable for all tested H values in the range of $H \in [1, 20]$. To illustrate this, we consider the following Δt samplings: $\Delta t \in \{0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$. In our tests, the stopping time T is chosen as $T = 10^4 \Delta t$ so that enough accumulations are experienced, and the L_2 errors will be reported. Of course, when Δt is as large as $\Delta t = 5$, the numerical error is meaningless. But as long as this error remains to be finite, this demonstrates the stability of the underlying time integration.

By considering two mesh sizes ($h = 0.5$ or $h = 0.25$), and two H values ($H = 1$ or $H = 20$), the numerical results of the proposed LOD, AOS, and MAOS schemes are depicted in Fig. 1 and Fig. 2. By producing finite error values for all tested Δt and h values, all new schemes are demonstrated to be unconditionally stable for the present example. Moreover, such an unconditional stability is not affected by the strong nonlinearity with $H = 20$. For a comparison, the results of the ADI schemes [11] are also shown in Fig. 1 and Fig. 2. The missing points in both figures denote Δt values for which the ADI methods were unstable. The present stability results of the ADI schemes are consistent with those reported in [11].
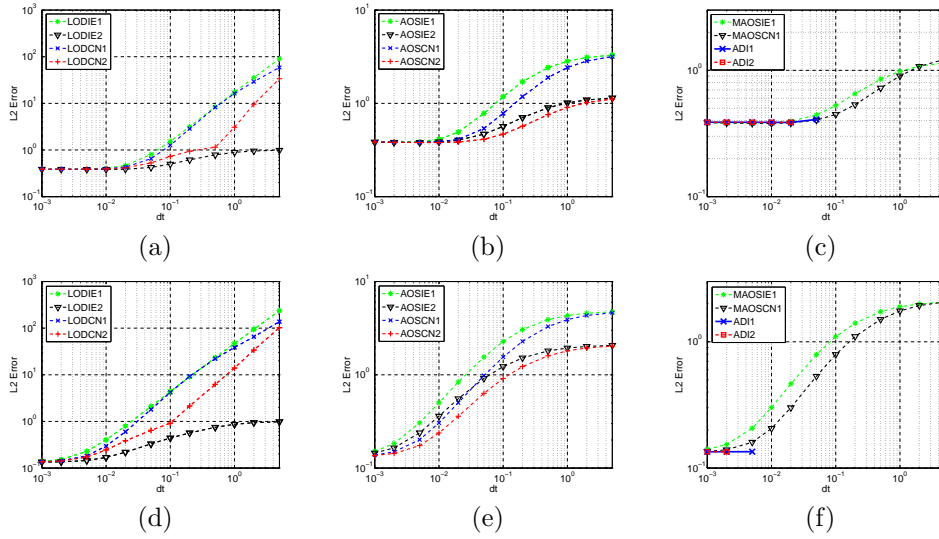
FIGURE 1. Stability verification with $H = 1$. (a) The LOD schemes with $h = 0.5$; (b) The AOS schemes with $h = 0.5$; (c) The MAOS and ADI schemes with $h = 0.5$; (d) The LOD schemes with $h = 0.25$; (e) The AOS schemes with $h = 0.25$; (f) The MAOS and ADI schemes with $h = 0.25$.

FIGURE 2. Stability verification with $H = 20$. (a) The LOD schemes with $h = 0.5$; (b) The AOS schemes with $h = 0.5$; (c) The MAOS and ADI schemes with $h = 0.5$; (d) The LOD schemes with $h = 0.25$; (e) The AOS schemes with $h = 0.25$; (f) The MAOS and ADI schemes with $h = 0.25$.

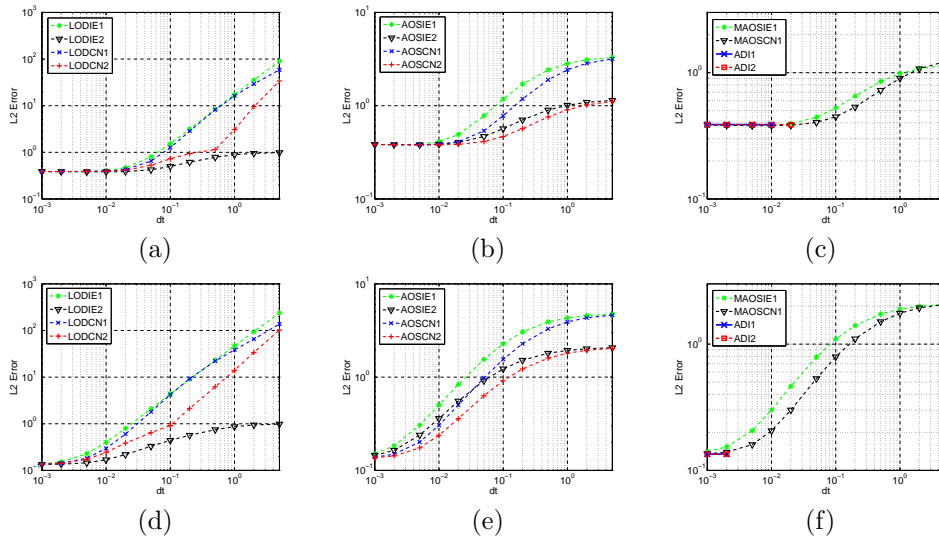We summarize the stability results shown in Fig. 1 and Fig. 2 by reporting the stability ranges of all tested methods in Table 2. For the LOD, AOS, and MAOS schemes, the range is always $[0.001, 5]$, because they are stable for all sampled Δt

TABLE 2. Stability range of all methods for the sampled Δt values.

Method	$H = 1$		$H = 20$	
	$h = 0.5$	$h = 0.25$	$h = 0.5$	$h = 0.25$
LOD	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$
AOS	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$
MAOS	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$	$[0.001, 5]$
ADI1	$[0.001, 0.05]$	$[0.001, 0.005]$	$[0.001, 0.01]$	$[0.001, 0.002]$
ADI2	$[0.001, 0.02]$	$[0.001, 0.002]$	$[0.001, 0.02]$	$[0.001, 0.002]$

values. The upbounds of stability ranges of the ADI schemes are all significantly less than 5, indicating that the ADI schemes are unstable for large Δt values. We note that the reported stability range in Table 2 is just a subset of the actual stability interval of each scheme, because such a range is concluded based on selected Δt values. For example, all reported schemes are stable for $\Delta t < 0.001$. But the interval $(0, 0.001)$ is not included in our ranges. Similarly, the proposed LOD, AOS, and MAOS schemes could be stable for some $\Delta t > 5$. But such large Δt values are of little numerical interest.

4.3. Spatial Convergence. We next study the spatial convergence of the proposed time splitting schemes. For this purpose, we vary the spatial mesh size N from 13 to 109, and take the time increment to be sufficiently small in all cases, i.e., $\Delta t = \frac{h^2}{20}$. We take the magnitude $H = 1$ in the initial solution and the stopping time to be $T = 10$ for the rest of the studies in this section.

The L_2 and L_∞ errors of the LOD, AOS, and MAOS schemes are reported in Fig. 3 in logarithmic scales. A similar pattern can be seen for all tested schemes. In particular, the numerical convergences become slower when N increases. The orders in L_2 norms are higher than those in L_∞ norms and the L_2 errors are also smaller than the L_∞ errors. To calculate an overall order of accuracy, a linear least-squares fitting is conducted in all cases. The fitted linear functions are also shown as the solid lines in Fig. 3. The slope r of these lines reflects the average convergence rate. It can be seen that L_∞ orders are all below one. As pointed out in [11], such accuracy reduction is due to the singular solution at the center of the sphere and the non-smoothness of the solution across the interface Γ, which is treated in an approximate sense in the present finite difference discretization. The non-uniform convergence in the L_∞ norm is particularly affected by the charge singularity at the center, because large errors exist in the calculation of potential near the center. However, the overall L_2 orders of all schemes are found to be within 0.7 to 1.2. This suggests that these schemes display approximately first-order convergence in the L_2 norm.

We are also interested in a comparison among these schemes. Interestingly, both LODIE schemes produce the exact same errors for this simple spherical case, as do the LODCN schemes. For other schemes, some minor changes will affect the orders slightly. In general, the orders of the AOS and MAOS schemes appear significantly worse than those exhibited by the LOD schemes.

4.4. Temporal Convergence. We finally investigate the temporal convergence of our schemes, displayed in Table 3. Because we should expect the spatial discretization error to be much larger than the temporal discretization error, due to the charge singularity and non-smooth interface previously mentioned, we may not be able to determine temporal order of convergence by comparing errors calculated
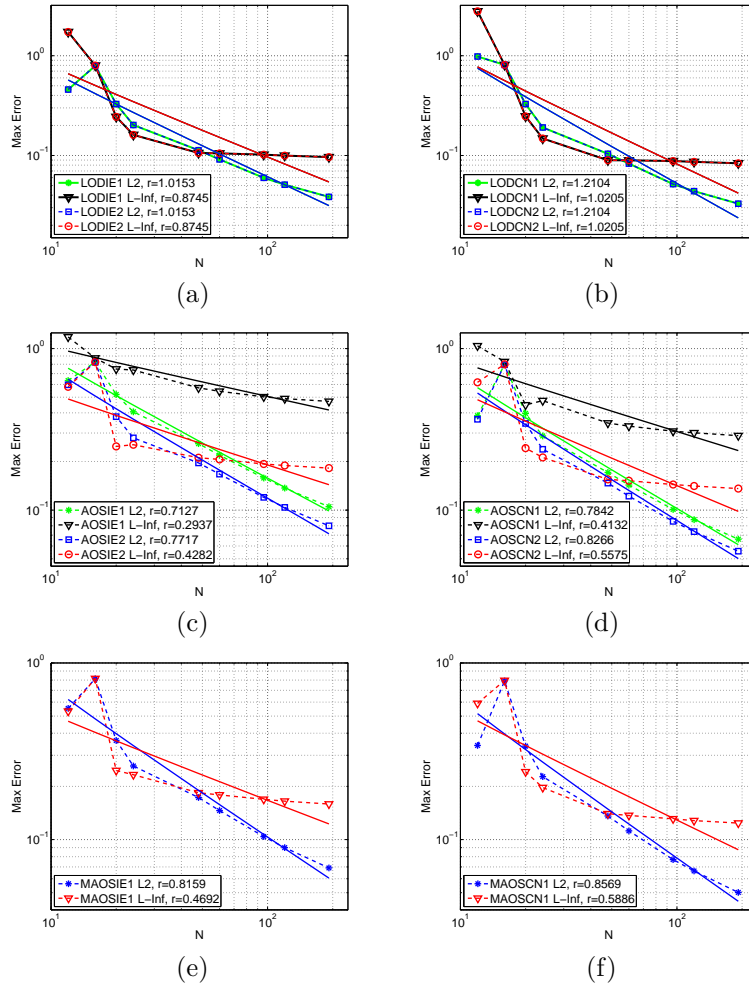
FIGURE 3. Spatial convergence in solving the sphere benchmark problem. (a) The LODIE schemes; (b) The LODCN schemes; (c) The AOSIE schemes; (d) The AOSCN schemes; (e) The MAOSIE schemes; (f) The MAOSCN schemes. In all charts, the dashed lines are error plots and the solid lines are linear least square fittings.

at various time increments Δt to the analytical solution. In fact, with a given h, the same error is yielded in our computations with different time increment Δt.

Thus, we choose a fixed spacing $h = 0.125$, and use the potentials calculated at the time increment $\Delta t = 2.5\text{E-}05$ as the reference solution. We then proceed to calculate the L_2 and L_∞ norms, and their respective orders of convergence, in the same manner as the previous subsection. The temporal order of convergence for all schemes in L_2 and L_∞ norms appears to be slightly higher than first-order, and the orders for the MAOS schemes in Table 3 appear to be the lowest. We also note that two versions of the LODIE and LODCN schemes yield exactly the same results.

TABLE 3. Temporal convergences in solving the benchmark problem.

dt	L_2	Order	L_∞	Order	L_2	Order	L_∞	Order
	LODIE1 and LODIE2				LODCN1 and LODCN2			
8E-4	8.04E-2	1.06	1.55E-1	1.06	5.46E-2	1.07	1.06E-1	1.07
4E-4	3.85E-2	1.11	7.44E-2	1.11	2.59E-2	1.11	5.03E-2	1.11
2E-4	1.79E-2	1.23	3.45E-2	1.23	1.20E-2	1.23	2.32E-2	1.23
1E-4	7.65E-3	1.59	1.48E-2	1.59	5.11E-3	1.59	9.91E-3	1.59
5E-5	2.55E-3		4.91E-3		1.70E-3		3.30E-3	
	AOSIE1				AOSIE2			
8.0E-4	1.20E-1	0.98	1.16E-1	0.97	8.36E-2	0.99	1.16E-1	0.97
4.0E-4	6.09E-2	1.06	5.94E-2	1.06	4.22E-2	1.06	5.94E-2	1.06
2.0E-4	2.92E-2	1.20	2.86E-2	1.20	2.02E-2	1.20	2.86E-2	1.20
1.0E-4	1.27E-2	1.57	1.25E-2	1.57	8.77E-3	1.57	1.25E-2	1.57
5.0E-5	4.27E-3		4.19E-3		2.95E-3		4.19E-3	
	AOSCN1				AOSCN2			
8.0E-4	6.31E-2	1.01	6.16E-2	1.00	4.36E-2	1.01	6.16E-2	1.00
4.0E-4	3.13E-2	1.08	3.07E-2	1.08	2.16E-2	1.08	3.07E-2	1.08
2.0E-4	1.49E-2	1.21	1.46E-2	1.21	1.02E-2	1.21	1.46E-2	1.21
1.0E-4	6.42E-3	1.58	6.31E-3	1.58	4.43E-3	1.58	6.31E-3	1.58
5.0E-5	2.15E-3		2.11E-3		1.48E-3		2.11E-3	
	MAOSIE1				MAOSCN1			
8.0E-4	6.40E-2	1.00	8.96E-2	0.99	3.31E-2	1.02	4.69E-2	1.01
4.0E-4	3.20E-2	1.07	4.53E-2	1.06	1.64E-2	1.08	2.33E-2	1.08
2.0E-4	1.53E-2	1.21	2.16E-2	1.20	7.73E-3	1.21	1.10E-2	1.21
1.0E-4	6.61E-3	1.43	9.40E-3	1.43	3.33E-3	1.43	4.75E-3	1.43
5.0E-5	2.45E-3		3.48E-3		1.24E-3		1.76E-3	

5. Biological applications

In this section, we further explore the stability and accuracy of the LOD, AOS, and MAOS schemes by considering the solvation analysis of real proteins. A detailed comparison among different time splitting schemes will be conducted for a particular protein system. Some corrections will be introduced to improve the accuracy of the time splitting. After identifying an optimum NPB solver, its usage on various protein systems will be considered. The CPU acceleration with respect to the existing operator splitting schemes will be examined.

5.1. Solvation energy and numerical setup. To quantitatively verify and compare the accuracy and efficiency of our new schemes, we compute solvation free energy based on the electrostatic potential calculated from the NPB equation. The energy released when the solute macromolecule is dissolved in solvent is known as the free energy of solvation, or solvation free energy. This solvation free energy may be calculated in our setting by computing the difference between total free energy of the macromolecule in a vacuum and in the solvent. Because we consider here only electrostatic effects, we may define the solvation energy as

$$(53) \qquad \Delta G = G_s - G_0 = \frac{1}{2} \int_\Omega \rho_m(\phi_m(\mathbf{r}) - \phi_0(\mathbf{r})) \, d\mathbf{r}$$

where ϕ_m is the electrostatic potential, in units of $kcal/mol/e_c$, in solvent, and ϕ_0 is the electrostatic potential in vacuum. These potential values are obtained by scaling
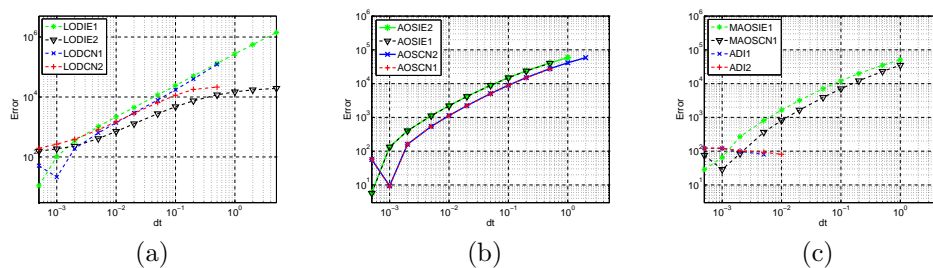
FIGURE 4. Stability verification for the protein 1ajj. (a) The LOD schemes; (b) The AOS schemes; (c) The MAOS and ADI schemes.

TABLE 4. Stability range of all methods on 1ajj for the sampled Δt values.

Method	Stability Range	Method	Stability Range
LODIE1	$[0.0005, 5]$	AOSIE1	$[0.0005, 0.5]$
LODIE2	$[0.0005, 5]$	AOSIE2	$[0.0005, 1]$
LODCN1	$[0.0005, 0.5]$	AOSCN1	$[0.0005, 0.5]$
LODCN2	$[0.0005, 0.5]$	AOSCN2	$[0.0005, 2]$
MAOSIE1	$[0.0005, 1]$	ADI1	$[0.0005, 0.005]$
MAOSCN1	$[0.0005, 1]$	ADI2	$[0.0005, 0.01]$

our calculated dimensionless potentials with the constant 0.592183, corresponding to room temperature $(298K)$. In a discrete setting, we may calculate the solvation energy as

$$(54) \qquad \Delta G = \frac{1}{2} \sum_i \sum_j \sum_k Q(x_i, y_j, z_k)(\phi_m(x_i, y_j, z_k) - \phi_0(x_i, y_j, z_k))$$

where Q is the trilinear interpolation of the singular charges in ρ_m. The scaled potential ϕ_m is calculated based on the time dependent NPB equation, while ϕ_0 is computed by solving a simple Poisson equation with a constant dielectric coefficient. As in the literature [20, 17], the Fast Fourier Transform (FFT) can be used to efficiently determine ϕ_0.

In all cases, a uniform mesh size $h = 0.5$ is used along all three dimensions and a large enough computational domain is chosen. In our calculations, we set the dielectric constants $\epsilon_s = 80$, and $\epsilon_m = 1$. The ionic strength I is set to 9.48955M, so the nonlinear constant $\bar{\kappa}$ is 8.9743. The scaling parameter is chosen as $\alpha = 1$, unless specified otherwise. A sufficiently large stopping time T will be used in each case to ensure that the steady state solution is reached. The molecular surface Γ underlying our computations is calculated based on the MSMS package [25]. In the MSMS surface generation, the probe radius is set to 1.4, and the density is chosen as 10. A Lagrangian to Eulerian conversion [36] is conducted to convert the triangular surface mesh of the MSMS package into 3D Cartesian grid values for determining $\epsilon(x_i, y_j, z_k)$.

5.2. Stability analysis. In the previous section, all of the proposed time splitting schemes are found to be unconditionally stable for a single atom case. It is of great interest to see if these schemes are still stable for real protein systems. We carry out this stability study by considering a protein 1ajj, a low-density lipoprotein receptor. As in the previous studies, we set $T = 10^4 \Delta t$ and consider a set of sampled Δt

values, $\Delta t \in \{0.0005, 0.001, 0.002, 0.005, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5\}$. For a reference, an Euler solution is produced from the explicit Euler method with a very small time increment $\Delta t = 1\text{E-}05$ and a large enough $T = 10$. The solvation energy of the Euler solution is calculated to be $\Delta G = -1209.7$, which will be taken as the reference energy value in studying pseudo-time NPB solvers.

The stability results in terms of the solvation energy errors of the proposed time splitting schemes are depicted in Fig. 4. We note that for the LODIE1 and LODIE2 methods, the finite error at each Δt implies stability of these two methods at every tested time step value. In fact, the LODIE methods are also found to be unconditionally stable for solving other protein systems, while all other time splitting schemes are conditionally stable for real proteins. In particular, as shown in Table 4, the LODCN, AOSIE1, and AOSCN1 methods were only unstable for $\Delta t = 1, 2, 5$. The AOSIE2 and MAOS methods were only unstable for $\Delta t = 2, 5$. The AOSCN2 method was only unstable for $\Delta t = 5$. The ADI methods, however, showed much reduced stability, with ADI1 only stable for $\Delta t = 0.0005, 0.001, 0.002, 0.005$, and ADI2 only stable for $\Delta t = 0.0005, 0.001, 0.002, 0.005, 0.01$. We note again that the stability range of Table 4 is simply based on a set of discrete Δt values being sampled. Such a range does not give a sharp estimate to the upper bound of the actual stability interval. For example, the LODCN methods are unstable at $\Delta t = 1$ and stable at $\Delta t = 0.5$. So only $\Delta t = 0.5$ is included in the stability range in Table 4. The real critical Δt is in between 0.5 and 1, and has to be detected separately. In this case, such a critical value is about 0.93. In summary, the newly created schemes, particularly the LODIE schemes, exhibit much improved stability over the existing ADI schemes.

5.3. Accuracy improvements. We next examine the accuracy by also considering the protein 1ajj. Table 5 shows calculated solvation energies of various time splitting schemes based on $T = 10$ and different Δt values. For each case, the CPU time and the percentage error with respect to the reference value generated by the Euler solution are also reported. For each scheme, it can be observed that the accuracy improves when Δt becomes smaller, while the CPU time increases linearly. Unfortunately, to produce a reasonable solvation energy value, a very small $\Delta t = 0.0005$ is typically required in these time splitting schemes, which implies a very slow computation. The superior stability of the proposed time splitting schemes cannot be taken advantage of, due to such a poor accuracy. On the other hand, we note that the LODIE2 scheme clearly produced the smallest error, while also requiring the least CPU time for all Δt. Based on these observations, and the unconditional stability of the LODIE2 method, we will focus only on the LODIE2 scheme in the following studies. Several remedies will be proposed to improve the accuracy of the LODIE2 scheme.

Our main goal here is to produce a reliable, albeit not highly accurate, estimate of the solvation energy based on some ultra large Δt values, so that the efficiency of the LODIE2 scheme can be significantly enhanced. Given the nearly first order convergence of the LODIE2 scheme shown in Table 5, we first investigate the use of the Richardson extrapolation technique to accelerate the convergence. With this process, we may cancel out high order error terms in our calculated electrostatic potentials by taking a linear combination of two results, calculated with different Δt time increments. In particular, we propose the following pointwise Richardson approximation

$$(55) \qquad \phi_m(x_i, y_j, z_k) = 2\phi_m^{\Delta t/2}(x_i, y_j, z_k) - \phi_m^{\Delta t}(x_i, y_j, z_k)$$

Table 5. Electrostatic solvation free energies (kcal/mol) and the corresponding CPU time in seconds for protein 1ajj by using operator splitting schemes.

Δt	ΔG	% Error	CPU	Δt	ΔG	% Error	CPU
	LODIE1				LODCN1		
0.0050	− 80.2	93.37%	150	0.0050	− 477.2	60.55%	165
0.0020	− 759.8	37.19%	369	0.0020	− 919.3	24.00%	383
0.0010	−1203.8	18.57%	706	0.0010	−1065.0	11.96%	772
0.0005	−1097.5	9.28%	1463	0.0005	−1137.5	5.97%	1535
	LODIE2				LODCN2		
0.0050	−1521.2	25.75%	142	0.0050	−1918.2	58.57%	159
0.0020	−1336.1	10.46%	335	0.0020	−1495.7	23.64%	372
0.0010	−1273.3	5.26%	648	0.0010	−1353.2	11.86%	739
0.0005	−1241.6	2.64%	1452	0.0005	−1281.6	5.94%	1476
	AOSIE1				AOSCN1		
0.005	1.1	100.09%	169	0.005	− 305.6	74.74%	179
0.0020	− 705.4	41.69%	399	0.0020	− 947.1	21.70%	481
0.0010	− 953.3	21.19%	802	0.0010	−1076.9	10.97%	901
0.0005	−1080.4	10.69%	1688	0.0005	−1143.0	5.51%	1883
	AOSIE2				AOSCN2		
0.005	4.6	100.38%	185	0.005	− 572.6	52.66%	205
0.0020	− 704.1	41.79%	410	0.0020	− 946.5	21.75%	451
0.0010	− 952.7	21.24%	819	0.0010	−1076.6	11.00%	872
0.0005	−1080.1	10.71%	1681	0.0005	−1142.8	5.52%	1792
	MAOSIE1				MAOSCN1		
0.005	− 571.0	52.80%	199	0.005	− 748.8	38.10%	197
0.0020	− 838.4	30.69%	394	0.0020	−1022.9	15.44%	418
0.0010	−1022.0	15.51%	781	0.0010	−1115.7	7.77%	887
0.0005	−1115.2	7.81%	1556	0.0005	−1162.5	3.90%	1751

Table 6. Electrostatic solvation free energies (kcal/mol) and the corresponding CPU time in seconds for protein 1ajj by using the improved LODIE2 and ADI methods.

Scheme	Δt	ΔG	% Error	CPU
LODIE2 RE	0.0050, 0.0025	−1212.8	0.25%	376
LODIE2 RE	0.0150, 0.0075	−1234.7	2.07%	133
LODIE2 RE	0.0500, 0.0250	−1435.5	18.67%	48
LODIE2 RE+V	0.4000, 0.2000	−1122.4	5.27%	27
ADI1	0.005	−1190.1	1.62%	165
ADI2	0.005	−1203.8	0.49%	202

where $\phi_m^{\Delta t}(x_i, y_j, z_k)$ represents the steady state electrostatic potential at a point (x_i, y_j, z_k) calculated with a time increment Δt. Essentially, a linear combination of our vectors of potentials calculated with time increment Δt and $\Delta t/2$ is taken.

The results of the LODIE2 scheme after applying the Richardson extrapolation are given in Table 6. Such a new LODIE2 method is labeled as the LODIE2 RE method. The reported CPU time of the LODIE2 RE scheme includes calculations of Δt, $\Delta t/2$, and Eq. (55). Typically, such a CPU time is about 3 times larger than

that of the LODIE method with the same Δt. Nevertheless, as can be seen from Table 6, the LODIE2 RE scheme becomes about 100 times more accurate. For a comparison, the results of the ADI1 and ADI2 methods [11] are also listed in Table 6. It can be seen that by using the same $\Delta t = 0.005$, the LODIE2 RE scheme is more accurate than the ADI methods, although it is also more expensive. We note that, being unconditionally stable, the LODIE2 RE method could be more efficient by using a larger Δt. On the other hand, the ADI methods will be unstable when $\Delta t > 0.01$.

To further improve the energy estimate of the LODIE2 scheme, we propose to apply the LODIE method to calculate both ϕ_m and ϕ_0 in Eq. (54). To our knowledge, such a treatment has never been explored in the existing pseudo-transient continuation approaches for solving the NPB equation [34, 11, 35]. In the literature, the pseudo-time methods are only applied to solve ϕ_m, while ϕ_0 is still obtained via the FFT fast Poisson solver. Symbolically, we can rewrite Eq. (54) as

$$(56) \qquad \Delta G = \frac{1}{2} \sum_i \sum_j \sum_k Q(x_i, y_j, z_k)(\phi_m^{\mathrm{LOD}}(x_i, y_j, z_k) - \phi_0^{\mathrm{FFT}}(x_i, y_j, z_k)),$$

in which $\phi_m^{\mathrm{LOD}}(x_i, y_j, z_k)$ is calculated by the LODIE2 method, while $\phi_0^{\mathrm{FFT}}(x_i, y_j, z_k)$ is calculated by the FFT method. In this work, we propose to solve the Poisson equation in vacuum by using the LODIE2 method and denote the corresponding potential as $\phi_0^{\mathrm{LOD}}(x_i, y_j, z_k)$. The electrostatic free energy is then calculated as

$$(57) \qquad \Delta G = \frac{1}{2} \sum_i \sum_j \sum_k Q(x_i, y_j, z_k)(\phi_m^{\mathrm{LOD}}(x_i, y_j, z_k) - \phi_0^{\mathrm{LOD}}(x_i, y_j, z_k)).$$

Even though the application of the LODIE2 scheme to the vacuum case would produce greater error in calculating ϕ_0 than the application of the FFT, we should expect the cancellation of some spatial-temporal discretization error between the vacuum and solvent cases. Thus, we should expect a potentially more accurate free energy of solvation value.

By applying the LODIE2 RE method to both solvent and vacuum cases, the new method is labeled as LODIE2 RE+V. Furthermore, to achieve the best balance between the accuracy and efficiency, we have studied various combinations of time increment Δt and the model scaling parameter α. In practical testing of the LODIE2 RE+V method implementation, we find temporal convergence to be slow for the vacuum case. Thus, we shall choose the scaling parameter α other than one. The optimal practical scaling parameter through empirical testing was determined to be $\alpha = \frac{1}{25}$. We find that the resulting method produces superior results with $T = 10$ and $\Delta t = 0.4$. The results of such a LODIE2 RE+V method are also given in Table 6. It can be seen that the relative error of the LODIE2 RE+V method with $\Delta t = 0.4$ is about 5%, which is acceptable in common biological simulations. However, the CPU time is roughly 15% of that of the ADI methods.

5.4. Solvation energies of proteins. We finally validate the proposed unconditionally stable LODIE2 RE and LODIE2 RE+V methods by considering a series of proteins with different size and geometric structures. The conditionally stable ADI methods [11] will also be tested for a comparison. In the following studies, the same stopping time $T = 10$ is used in all methods. For the ADI, LODIE2 RE and LODIE2 RE+V methods, the time increment is chosen as $\Delta t = 0.005$, $\Delta t = 0.05$, and $\Delta t = 0.4$, respectively. We note that the ADI results in the present study are different from those in [11], because a stronger nonlinearity is considered here with the ionic strength $I = 9.48955$M, while $I = 0.15$M in [11].

TABLE 7. Electrostatic solvation free energies (kcal/mol) for 23 proteins. All proteins use a density parameter of 10 except for 451c and 1a7m, which use 58 and 40, respectively.

PDB	#atoms	Euler	ADI1	ADI2	LODIE2 RE	LODIE2 RE+V
1ajj	519	−1209.7	−1190.1	−1203.8	−1435.5	−1122.4
1bbl	576	−1269.8	−1246.6	−1263.6	−1517.1	−1221.5
1bor	832	−1043.0	−1029.7	−1039.2	−1351.2	−1048.5
1bpi	898	−1440.6	−1415.9	−1432.4	−1799.9	−1300.3
1cbn	648	−458.3	−449.9	−455.7	−717.1	−489.7
1fca	729	−1109.9	−1098.4	−1106.7	−1407.3	−1157.0
1frd	1478	−2645.1	−2611.9	−2634.9	−3252.5	−2530.3
1fxd	824	−2235.8	−2220.6	−2231.5	−2599.0	−2342.6
1hpt	858	−1139.0	−1116.8	−1132.3	−1512.4	−1096.1
1mbg	903	−1401.7	−1383.2	−1395.8	−1763.6	−1347.0
1neq	1187	−2092.9	−2064.6	−2083.9	−2577.5	−1974.9
1ptq	795	−1098.5	−1079.2	−1092.0	−1389.9	−985.6
1r69	997	−1302.1	−1283.4	−1296.1	−1701.6	−1230.6
1sh1	702	−994.6	−981.1	−990.8	−1279.3	−970.1
1svr	1435	−2236.3	−2205.7	−2226.5	−2812.2	−2080.8
1uxc	809	−1363.8	−1345.7	−1357.9	−1671.3	−1275.4
1vii	596	−1109.7	−1097.0	−1105.7	−1307.4	−1046.5
2erl	573	−925.5	−961.6	−922.2	−1163.1	−942.7
2pde	667	−974.4	−962.5	−970.9	−1243.0	−897.2
451c	1216	−1326.7	−1845.8	−1341.8	−1800.2	−1295.0
1a2s	1272	−1814.0	−1794.1	−1808.1	−2320.2	−1898.3
1a63	2065	−3117.2	−3072.2	−3102.4	−3937.5	−2931.3
1a7m	2809	−2545.6	−1845.8	−1847.4	−2950.0	−1921.9

We first consider a set of 23 proteins, which have been used for testing the previous solvation models [34, 11, 35]. For a reference, the explicit Euler method with $T = 10$ and $\Delta t = 1.0$E-05 is also employed. The electrostatic solvation free energies calculated by the tested methods are given in Table 7. As in the previous studies, the Euler solution can be regarded as the reference energy value, to which all tested time splitting methods should converge. It can be observed from Table 7 that the LODIE2 RE method with $\Delta t = 0.05$ yields a poor accuracy, even though it is much faster than the ADI methods. On the other hand, being even faster, the LODIE2 RE+V method produces energy estimates which are in good agreement with those of the Euler method. We note that the results of two proteins, i.e., 451c and 1a7m, appear to be inconsistent. In generating the molecular surface of the tested proteins, a density value of 10 is used in the MSMS software [25]. However, the ADI1 method with $\Delta t = 0.005$ turns out to be unstable for these two proteins, while such an instability was not encountered in our previous study [11], because a strong nonlinearity with $I = 9.48955$M is considered here. To fix the problem while still using $\Delta t = 0.005$ for the ADI1, we have chosen the density parameters to be 58 and 40, respectively, for 451c and 1a7m in the MSMS package. Due to this change, the differences in energy results of the tested methods are quite large for these two proteins.

We next study a set of 26 proteins with larger atom numbers. The electrostatic solvation free energies produced by the ADI1, ADI2, and LODIE2 RE+V methods

TABLE 8. Electrostatic solvation free energies (kcal/mol) for additional proteins.

PDB	#atoms	ADI1	ADI2	LODIE2 RE+V
2lzj	1687	–	−1983.2	−1922.6
2l6n	2093	–	−2593.5	−2533.8
1hd2	2414	−2203.6	−2237.6	−2121.8
1urm	2425	–	−2261.0	−2155.7
1qah	3954	–	−3853.4	−3513.6
3lup	4363	−6090.9	−6147.5	−5730.8
1eri	4522	−5482.5	−5549.4	−5134.9
1beb	4972	–	−6701.7	−6208.2
2a33	5182	−5074.6	−5131.9	−4914.5
1wkd	5755	–	−5335.5	−5069.4
1e5m	6060	–	−4863.0	−4528.6
2w2o	6246	–	−6481.0	−5940.6
2w2q	6253	−6331.0	−6399.2	−5795.5
2w2p	6263	–	−6369.8	−5742.3
2w2m	6288	–	−6804.7	−6042.3
4dn3	6344	−3718.5	−3721.7	−3789.8
3r79	6737	−8065.4	−8146.4	−7525.4
1bif	6974	−7137.7	−7050.5	−6381.2
1gsu	7164	−5906.5	−5964.6	−5531.7
4dn4	7390	−5947.1	−5970.1	−5826.7
3loq	8136	−12060.4	−11494.7	−10466.5
3gcw	8505	−8594.8	−8568.3	−7514.3
3bps	8513	–	−8389.9	−7517.9
1rva	8726	–	−9007.5	−7892.4
1vng	8808	−7735.6	−7798.1	−6596.6
1vns	8815	–	−7908.3	−6976.1

are shown in Table 8. A density parameter of 10 is used in the MSMS package for all calculations. The proteins for which the ADI1 method was unstable for this parameter are marked with a dash in the ADI1 column. No reference solutions are computed for these proteins, because the computation of the Euler solution is too time consuming for large proteins. In comparing the LODIE2 RE+V results with those of ADI1 and ADI2, the present results suggest that the proposed time splitting method provides reliable and fairly accurate energy estimates.

To quantitatively analyze the efficiency of the LODIE2 RE+V method, we report the CPU time in seconds of the ADI1, ADI2, and LODIE2 RE+V methods for the proteins reported in Tables 7 and 8 in Fig. 5. Note that only the CPU time consumed in solving the NPB and Poisson equations is reported here. The CPU times used for the initial numerical setup, including the trilinear interpolation of the source term, the Lagrangian to Eulerian conversion for computing the dielectric coefficient, etc., are not considered here, because they are the same in all time splitting schemes. To achieve a better understanding, we plot the CPU time against the number of atoms N_a in Fig. 5. It is clear that the CPU time is roughly a linear function of N_a. A least square linear fitting can be conducted to represent CPU time in seconds as a function of N_a: CPU= $mN_a + b$. Here, we have $(m, b) = (0.253, 66.9)$, $(0.224, 16.0)$, and $(0.011, 6.8)$, respectively, for the ADI1, ADI2, and
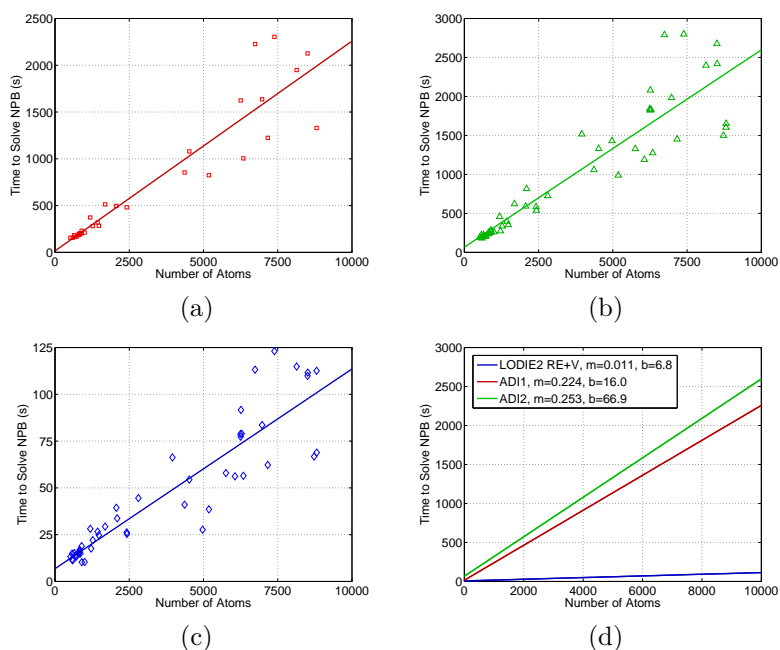
FIGURE 5. The CPU time in seconds consumed by the ADI1,
ADI2, and LODIE2 RE+V methods for solving the time depen-
dent NPB equation. (a) ADI1; (b) ADI2; (c) LODIE2 RE+V; (d)
A comparison of three methods.

LODIE2 RE+V methods. It is clear from Fig. 5 that the time required for the
modified LODIE2 method grows much less slowly as the number of atoms increases.
On average, the modified LODIE2 method is over 20 times more efficient than the
ADI methods for large proteins.

Finally, we plot the surface potentials of a protein (PDB ID: 4DN4) in Fig. 6.
Since no reference solution is offered in Table 8, it is of interest to further study
the reliability of the LODIE2 RE+V method for processing the proteins listed
in that table, such as the 4DN4. The electrostatic potentials generated by the
ADI1, ADI2, and LODIE2 RE+V methods are depicted in parts (a), (b), and (c),
respectively. Recall that the energies of the LODIE2 RE+V method in Table 8
are after recovery. But the original potential $\phi_m^{\text{LOD}}(x_i, y_j, z_k)$ is used to generate
the surface plot in (c). Hence, the part (c) is significantly different from those of
the ADI methods. The colors of this picture are much lighter, which stands for
much weaker projected potentials. Just like in calculating the free energy, we can
improve the result by considering the vacuum potentials generated by the FFT and
LOD methods. In part (d), the recovered solution defined as $\phi_m^{\text{LOD}}(x_i, y_j, z_k) +$
$\phi_0^{\text{LOD}}(x_i, y_j, z_k) - \phi_0^{\text{FFT}}(x_i, y_j, z_k)$ is utilized. Visually, this new result is close to
those of the ADI methods. Hence, the recovered LODIE2 RE+V potential can
also be similarly used in analyzing the fast/slow electrostatic potential changing
region on the solute-solvent boundary for biomolecular studies. The present study
demonstrates that besides providing a reliable free energy estimation, the proposed
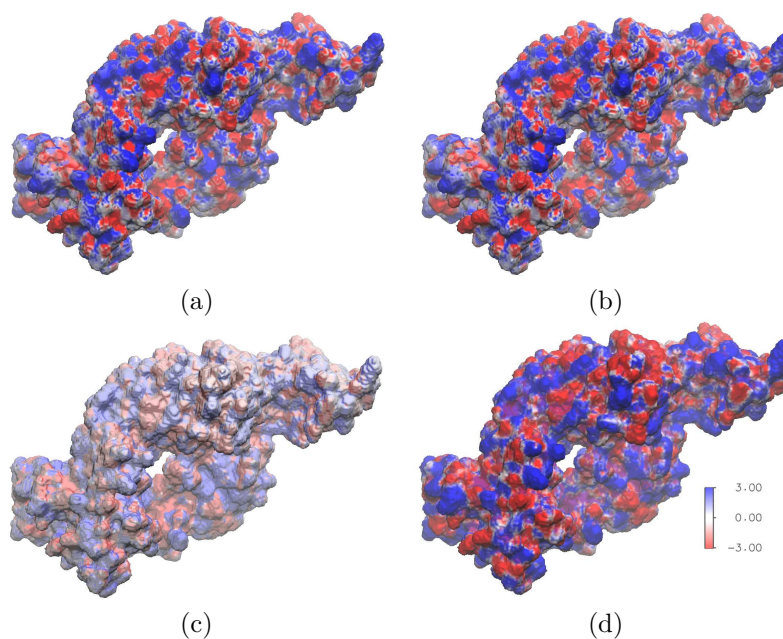LODIE method can also generate a usable pointwise potential.

FIGURE 6. Plots of the surface potential (kcal/mol/e_c) of the protein 4DN4 using different potential solutions. (a) ADI1; (b) ADI2; (c) LODIE2 RE+V; (d) LODIE2 RE+V with recovery.

6. Conclusion

This paper presents the first unconditionally stable numerical scheme for solving the nonlinear Poisson-Boltzmann (NPB) equation in a pseudo-transient approach. Using this approach, the solution to the NPB equation is recovered from the steady state solution to the time dependent pseudo-transient form. However, due to the long time integration of the process, it is necessary to develop numerical schemes that are stable and accurate for large time increments. The alternating direction implicit (ADI) methods previously developed for solving the time dependent NPB equation [11, 35] are fully implicit, but are still conditionally stable. To construct unconditionally stable NPB solution methods, we consider various operator splitting procedures to all five terms of the NPB equation, in both multiplicative and additive styles. This gives rise to locally one-dimensional (LOD) schemes, additive operator splitting (AOS) schemes, and multiplicative-additive operator splitting (MAOS) scheme. In these schemes, the nonlinear term is analytically integrated, and both implicit Euler and Crank-Nicolson time integrations are formulated. A standard finite difference scheme is utilized for spatial discretization in all schemes. Extensive numerical experiments are conducted to verify the unconditional stability and accuracy of the proposed time splitting schemes. One LOD scheme is found to outperform other schemes in terms of both stability and accuracy, and is recommended for electrostatic free energy analysis of real proteins. Further improvements are introduced to enhance the accuracy and efficiency of this LOD scheme for a fast biomolecular simulation.

In the following list, we summarize the major numerical features of the proposed operator splitting schemes:

- **Stability.** For a smooth solution, all proposed time splitting methods should be unconditionally stable, because each individual implicit Euler or Crank-Nicolson time integration with central difference approximation is unconditionally stable. A major task in our numerical experiments is to verify the stability for non-smooth solutions. In the case of a simple spherical cavity test, these methods remain unconditionally stable. For real protein systems, only the LODIE methods are stable for any Δt. Some proposed methods, such as AOSIE1, AOSCN1, and LODCN methods, become conditionally stable, while other methods, such as AOSIE2, AOSCN2, MAOS methods, are stable for all $\Delta t < 1$. Overall, the stability constraints of the proposed time splitting methods are much improved in comparing with the existing ADI methods. The LODIE method is the most stable method.

- **Accuracy.** Because the charge singularities (the delta functions) and the non-smoothness of the solution across the interface Γ are treated in an approximate sense in the present central difference discretization, all numerical schemes achieve roughly the first-order convergence in space. All proposed time splitting methods are of first order accuracy in time, because the underlying time splitting is first order and the discretization error of the implicit Euler or Crank-Nicolson integration is at least first order. However, as shown in protein tests, a very small Δt has to be used in order to produce an accurate estimate of electrostatic free energy. A Richardson extrapolation technique and a recovery based on the replacement of the fast Fourier transform method by the operator splitting method are proposed to significantly improve the precision in energy calculations. The accelerated LODIE method becomes biologically useful, with a large $\Delta t = 0.4$ and a stopping time $T = 10$.

- **Efficiency.** By using central difference discretization, the resulting linear systems in each alternating direction are tridiagonal, and can be efficiently solved via the Thomas algorithm [21]. Without the loss of generality, we can assume that the numbers of grid nodes along x, y and z directions are all on the order N. Then, the linear system in each subsystem of the time splitting procedure has the dimensions $N \times N$. The Thomas algorithm solution of such a system has a complexity on the order of $O(N)$, so that the operations for calculating one time step in the proposed time splitting methods are on the order of $O(N^3)$. Since a fixed number of total time steps is used in the LODIE solution, the overall complexity of the proposed LODIE method is still $O(N^3)$, making this method very attractive to large protein systems. As demonstrated in our CPU tests, the LODIE method scales linearly with respect to the number of atoms of the protein. On average, the unconditionally stable LODIE method is over 20 times more efficient than the conditionally stable ADI methods for large proteins.

When the nonlinearity effect of the NPB equation is strong, the efficient algebraic solution of a large scale nonlinear system remains a challenge in biomolecular simulations [20, 17]. Such a difficulty is simply bypassed in a pseudo-transient continuation approach through an analytical integration. Moreover, the present study reveals that an unconditionally stable computation is feasible through the proposed schemes, so that the efficiency of the pseudo-time NPB solution is significantly improved. Consequently, the pseudo-time NPB solver becomes quite promising for solving strong nonlinear protein systems.

It is of great interest to further generalize the proposed time splitting methods by considering other numerical difficulties associated with the NPB equation. A regularization procedure to treat the source singularities is under our investigation. The use of a different molecular surface definition will also be examined. The stability proof of the proposed operator splitting methods remains an open question, mainly due to the nonlinear hyperbolic sine term.

Acknowledgments

References

[1] I. Ahmed, E.-K. Chua, E.-P. Li, and Z. Chen, Development of the three-dimensional unconditionally stable LOD-FDTD method, IEEE Trans. Antennas Propag., 56, 3596-3600, (2008).

[2] D. Barash and T. Schlick, Multiplicative operator splittings in nonlinear diffusion: from spatial splitting to multiple timesteps, J. Math. Imaging Vision, 19, 33-48, (2003).

[3] 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, 193-231, (2009).

[4] P.W. Bates, G.W. Wei and S. Zhao, Minimal molecular surfaces and their applications, J. Comput. Chem., 29 380-391, (2008).

[5] D.S. Daoud, On the numerical solution of multi-dimensional parabolic problem by the additive splitting up method, Appl. Math. Comput., 162, 197-210, (2005).

[6] M.E. Davis and J.A. McCammon, Electrostatics in biomolecular structure and dynamics, Chem. Rev., 90, 509-521, (1990).

[7] J. Douglas, Jr., On the numerical integration of $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{\partial u}{\partial t}$ by implicit methods, J. Soc. Indust. Appl. Math., 3, 42-65, (1955).

[8] J. Douglas, Jr. and D. Peaceman, Numerical solution of two-dimensional heat flow problems, Amer. Inst. Chem. Engrg. J., 1, 505-512, (1955).

[9] E. D'Yakonov, Difference schemes with split operators for multidimensional unsteady problems (English translation), USSR Comp. Math., 3, 518-607, (1963).

[10] T.H. Gan and E.L. Tan, Unconditionally stable fundamental LOD-FDTD method with second-order temporal accuracy and complying divergence, IEEE Trans. Antennas Propag., 61, 2630-2638, (2013).

[11] W. Geng and S. Zhao, Fully implicit ADI schemes for solving the nonlinear Poisson-Boltzmann equation, Molecular Based Math. Bio., 1, 109-123, (2013).

[12] M.J. Holst, Multilevel methods for the Poisson-Boltzmann equation, Ph.D. thesis, University of Illinois at Urbana-Champaign, (1993).

[13] M.J. Holst and F. Saied, Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods, J. Comput. Chem., 16, 337-364, (1995).

[14] B. Honig and A. Nicholls, Classical electrostatics in biology and chemistry, Science, 268, 1144-1149, (1995).

[15] 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.

[16] J.G. Kirkwood, Theory of solution of molecules containing widely separated charges with special application to Zwitterions, J. Chem. Phys., 7, 351-361, (1934).

[17] C. Li, L. Li, M. Petukh, and E. Alexov, Progress in developing Poisson-Boltzmann equation solvers, Molec. Based Math. Bio., 1, 42-62, (2013).

[18] T. Lu, P. Neittaanmaki, and X.-C. Tai, A parallel splitting up method and its application to Navier-Stokes equations, Appl. Math. Lett., 4, 25-29, (1991).

[19] T. Lu, P. Neittaanmaki, and X.-C. Tai, A parallel splitting up method for partial differential equations and its application to Navier-Stokes equations, RAIRO Math. Model Numer. Anal., 26, 673-708, (1992).

[20] B.Z. Lu, Y.C. Zhou, M.J. Holst, and J.A. McCammon, Recent progress in numerical methods for the Poisson-Boltzmann equation in biophysical applications, Commun. Comput. Phys., 3, 973-1009, (2008).

[21] A.R. Mitchell and D. F. Griffiths, The finite difference methods in partial differential equations, John Wiley & Sons, Inc., Chichester (1980).

[22] D. Peaceman and H. Rachford, The numerical solution of parabolic and elliptic equations, J. Soc. Indust. Appl. Math., 3, 28-41, (1955).

[23] F.M. Richards, Areas, volumes, packing and protein structure, Annu. Rev. Biophys. Bioeng., 6, 151-176, (1977).

[24] 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.

[25] M. Sanner, A. Olson, and J. Spehner, Reduced surface: an efficient way to compute molecular surfaces, Biopolymers, 38, 305-320, (1996).

[26] S. A. Sauter and C. Schwab, Boundary Element Methods, Springer, (2010).

[27] A. Sayyed-Ahmad, K. Tuncay, and P.J. Ortoleva, Efficient solution technique for solving the Poisson-Boltzmann equation, J. Comput. Chem., 25, 1068-1074, (2004).

[28] 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, 62-79, (2002).

[29] X.-C. Tai and P. Neittaanmaki, A parallel finite element splitting-up methods for parabolic problems, Numer. Methods Partial Different. Equat., 7, 209-225, (1991).

[30] J. Weickert, K.J. Zuiderveld, B.M. ter Haar Romeny, and W.J. Niessen, Parallel implementations of AOS schemes: a fast way of nonlinear diffusion filtering, IEEE Inter. Conf. Image Processing, 3, 396-399, (1997).

[31] N. Yanenko, Convergence of the method of splitting for the heat conduction equations with variable coefficients (English translation), USSR Comp. Math., 3, 1094-1100, (1963).

[32] N. Yanenko, The method of fractional steps, Springer-Verlag, Berlin, Heidelberg, and New York, 1971. (English translation; originally published in Russian, 1967).

[33] S.-C. Yang, Z. Chen, Y. Yu, and W.-Y. Yin, An unconditionally stable one-step arbitrary-order leapfrog ADI-FDTD method and its numerical properties, IEEE Trans. Antennas Propag., 60, 1995-2003, (2012).

[34] S. Zhao, Pseudo-time coupled nonlinear models for biomolecular surface representation and solvation analysis, Int. J. Numer. Method Bio. Engrg., 27, 1964-1981, (2011).

[35] S. Zhao, Operator splitting ADI schemes for pseudo-time coupled nonlinear solvation simulations. J. Comput. Phys., 257, 1000-1021, (2014).

[36] Y.C. Zhou, M. Feig, and G.W. Wei, Highly accurate biomolecular electrostatics in continuum dielectric environments, J. Comput. Chem., 29, 87-97, (2008).

Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA.
E-mail: lwwilson@umich.edu and szhao@ua.edu