A fast alternating direction implicit algorithm for geometric flow equations in biomolecular surface generation

Wufeng Tian and Shan Zhao*

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

SUMMARY

In this paper, a new alternating direction implicit (ADI) method is introduced to solve potential driven geometric flow PDEs for biomolecular surface generation. For such PDEs, an extra factor is usually added to stabilize the explicit time integration. However, two existing implicit ADI schemes are also based on the scaled form, which involves nonlinear cross derivative terms that have to be evaluated explicitly. This affects the stability and accuracy of these ADI schemes. To overcome these difficulties, we propose a new ADI algorithm based on the unscaled form so that cross derivatives are not involved. Central finite differences are employed to discretize the nonhomogenous diffusion process of the geometric flow. The proposed ADI algorithm is validated through benchmark examples with analytical solutions, reference solutions, or literature results. Moreover, quantitative indicators of a biomolecular surface, including surface area, surface-enclosed volume, and solvation free energy, are analyzed for various proteins. The proposed ADI method is found to be unconditionally stable and more accurate than the existing ADI schemes in all tests. This enables the use of a large time increment in the steady state simulation so that the proposed ADI algorithm is very efficient for biomolecular surface generation. Copyright © 2013 John Wiley & Sons, Ltd.

Received 11 September 2013; Revised 15 October 2013; Accepted 15 October 2013

KEY WORDS: biomolecular surface generation; geometric flow equation; multiscale nonlinear solvation model; alternating direction implicit (ADI) scheme; solvation free energy

1. INTRODUCTION

The modeling of molecular surfaces is of great importance in quantitative studies of macromolecules, including proteins, DNAs, molecular motors, and viruses, because the structure, function, dynamics, and transport of macromolecules depend on the features of molecular surfaces. In the implicit solvent biomolecular simulations [1–3], a biomolecular surface is indispensable as the solute–solvent boundary separating a macromolecule of interest from its surrounding aqueous environment. Such a boundary is used to define the dielectric constants and Debye–Huckel length parameters for calculating the electrostatic potential. The visualization of the calculated electrostatic potential and the underlying molecular surface play an important role in the analysis of biomolecular structure and interaction [4–6], such as protein folding, ligand-receptor binding, and molecular docking. Geometrically, the molecule surface also determines the area and volume of the macromolecule, which can be used in nonpolar analysis of solvation free energies [7–9].

Numerous molecular surface models have been developed in the molecular biology literature. The most commonly used ones are defined geometrically, including the Van der Waals (VdW) surface, the solvent-accessible surface (SAS) [10], and the solvent-excluded surface (SES) [11, 12]. These models are also known as ‘hard sphere’ models, in which atoms of a biomolecule are represented
as solid spheres using their VdW radii. The VdW surface can then be defined as the smallest envelope enclosing the collection of such spheres. The other two surfaces are defined by rolling a probe sphere with a given radius, which mimics the water molecule, around the VdW surface. The SAS is traced by the center of the probe, while the SES is traced by the inward-facing surface of the probe. These hard sphere molecular surface models are known to admit geometric singularities, such as cusps and self-intersecting surfaces [13–16].

In contrast, there exist a family of ‘soft sphere’ models, in which each atom is described by a Gaussian density distribution function with the dimension determined by its VdW radius [17–19]. The summation of these Gaussian soft clouds forms an electron density map and gives rise to Gaussian molecular surfaces at appropriate isosurfaces or level sets to approximate the rigid VdW surface, SAS, or SES. The volumetric density map can also be generated based on other smoothly decaying functions [20] or the maximum of Gaussian functions postprocessed by a low-pass filtering [21]. The smoothness of Gaussian surface models enables fast and robust molecular surface mesh generations [20, 22, 23] that are of high mesh quality for subsequent biomolecular visualizations or boundary/finite element simulations.

What are related to the present work are the PDE-based molecular surface models [15, 16, 24], in which smooth level set or hypersurface functions are produced by solving various PDEs for isosurface generation. We note that the PDE molecular surface models to be discussed here are different from the PDE-based surface smoothing techniques [22, 25] that start with an existing molecular surface. Instead, in PDE-based molecular surface models [15, 16, 24], atomic coordinates and VdW radii are directly employed to generate a hypersurface function that consequently determines the molecular surface. To our knowledge, the first PDE-based molecular surface model was constructed by Wei and his coworkers in 2005 [24], which employed a curvature-controlled diffusion equation to construct molecular multiresolution surfaces. In 2006, the Euler–Lagrange variation of the free energy minimization was applied for the first time to derive a variational PDE model for molecular surface generation [15, 16]. Neglecting other solute–solvent interactions, the surface free energy minimization of a macromolecule in the aquatic environment is simplified to be the surface area minimization, and the corresponding variational analysis leads to the mean curvature flow—a PDE whose solution under an appropriate constraint gives rise to the minimal molecular surface (MMS) [15, 16]. More general potential driven geometric flow PDEs have been developed in [26] by considering more solute–solvent interactions in biomolecular surface formation and evolution. The resulting variational PDE model is inherently multiscale in nature and enables the incorporation of microscopic interactions, such as VdW potentials, into the macroscopic curvature evolution. The biomolecular surface construction through other PDE models, such as the level set approach [27], PDE transform [28], and fractional PDEs [29], have also been proposed in the literature.

Besides single PDE models, there also exist multiple PDE models, in which the biomolecular surface extraction depends on other physical quantities of interests while also determines such quantities through feedback coupling. Such a flexible framework was first introduced by Wei [30] in developing a family of differential geometry based multiscale models for studying various chemical and biological systems. By considering a very general total free energy functional including not only polar and nonpolar solute–solvent interactions but also the dynamics and transports, the Euler–Lagrange variation yields a coupled PDE system, including a geometric flow equation for the molecular surface, the Poisson–Boltzmann equation for the electrostatic potential, and other equations for the fluid and molecular dynamics [30]. The particular formulation and computational realization of such models for analyzing the equilibrium properties of solvation have been studied in [8, 9, 31].

One common difficulty underlying all PDE models discussed previously is the fast numerical solution of three-dimensional (3D) PDEs in demanding biomolecular simulations involving large spatial and/or temporal scales. In particular, the steady state solution of the potential driven mean curvature flow equation, which is underlying many PDE molecular surface models [8, 9, 15, 16, 26, 30, 31], deserves a further study. Finite differences are usually employed for the spatial discretization of such an equation, while the time integration represents the major bottleneck in numerical simulations. The explicit Euler scheme was employed to solve the mean curvature equation in the original MMS model [15, 16], which is very inefficient. Subject to a severe stability
constraint, a large number of time steps has to be used in explicit schemes to reach the steady state. To relieve the stability issue, four implicit time-stepping schemes were proposed in [26], including two non-splitting schemes that solve 3D algebraic systems by iterative solvers and two splitting alternating direction implicit (ADI) schemes. Larger time increments are usually allowed in these implicit approaches, and two ADI schemes seem to be more efficient in solving large proteins, thanks to the Thomas algorithm in solving one-dimensional (1D) tridiagonal systems [26]. Nevertheless, these two ADI schemes are semi-implicit in the sense that there are nonlinear terms being evaluated at the previous time instant at each time step. This limits the temporal order of these ADI schemes to be at most one [26]. Moreover, it has been shown in solving another parabolic equation that the absolute stability cannot be always guaranteed for semi-implicit ADI schemes when the magnitude of the initial solution becomes larger [32].

The goal of this work is to develop a fast and accurate ADI algorithm for solving the mean curvature flow equation in molecular surface generation. The most distinct feature of the new ADI scheme comparing with the existing ADI schemes [26] is that the mean curvature equation will not be rewritten into two processes, containing a diffusion term and a nonlinear term with cross-derivatives. Instead, in the divergence form, it is treated as one nonhomogeneous diffusion process in the present study. By evaluating the diffusion coefficients at the previous time step, the central finite difference can be used to discretize the nonhomogeneous diffusion term so that the resulting linear systems are still tridiagonal. Because of the compactness, the new ADI scheme yields a higher temporal order and is unconditionally stable in our biomolecular simulations. We note that the present ADI scheme is constructed based on the same principle we recently advocated for developing the fully-implicit ADI schemes for solving the nonlinear Poisson–Boltzmann equation [32, 33]. Nevertheless, the nonlinear term is a reaction term in the Poisson–Boltzmann equation so that fully implicit schemes can be realized through operator splitting techniques [32, 33], whereas the present mean curvature equation involves nonlinearity in the diffusion term, which rules out a fully implicit linear time integration. On the other hand, we note that the mean curvature flow and other geometric flow equations have also been widely used for image processing [34–38]. Unconditionally, stable time integration schemes have also been developed in imaging studies [39–41] to solve geometric flow equations, and these approaches can potentially be applied to the biomolecular surface generation.

The rest of paper is organized as follows. Section 2 is devoted to the theory and formulation of the ADI time integration. We first briefly present the potential driven mean curvature flow equation and existing numerical works. Then, the new ADI scheme is constructed. Numerical validations based on various benchmark biological systems are considered in Section 3. Both stability and accuracy will be examined for the proposed ADI algorithm and two existing ADI schemes in solving the geometric flow equations. Physical quantities, such as surface areas and surface enclosed volumes, will be evaluated for atomic systems. The application of the proposed ADI algorithm to various atomic systems, amino acids, and protein molecules is conducted in Section 4. The solutions to geometric flow equations with and without driving potentials will be carried out. Solvation analysis via solving the Poisson–Boltzmann equation of the electrostatic potential will be considered based on the molecular surface generated through the new ADI scheme. Finally, this paper ends with concluding remarks.

2. MATHEMATICAL MODELS AND NUMERICAL ALGORITHMS

In this section, the MMS model will be introduced in the framework of the differential geometry-based multiscale solvation analysis [8, 9, 30, 31]. A subtle issue in the governing equations of the potential driven mean curvature flow will be discussed. Two existing ADI schemes will be briefly reviewed. Finally, the details of the proposed ADI algorithm will be offered.

2.1. Physical background and mathematical models

Consider a solute macromolecule such as a protein immersed in an aqueous solvent environment. The solute–solvent boundary or the molecular surface can be modeled as either a sharp interface [9, 15, 16, 32] or a smooth interface [8, 31, 33]. We will concern ourselves to the smooth interface
case in this work, even though the proposed algorithm can also be applied to the sharp interface case. In particular, we define a molecular domain \(\Omega_m\), a solvent domain \(\Omega_s\), and a boundary domain \(\Omega_b\). The entire domain \(\Omega \subset \mathbb{R}^3\) of this solute–solvent system consists of solute and solvent regions: \(\Omega = \Omega_s \cup \Omega_m\), while \(\Omega_b\) is an overlapping transition region \(\Omega_b = \Omega_s \cap \Omega_m \neq \emptyset\). Following the previous works [8, 31, 33], we introduce a characteristic function \(S(\mathbf{r})\) for the solute domain \(\{S : \Omega \rightarrow \mathbb{R}\}\). Inside the biomolecule, \(S(\mathbf{r})\) takes value one, while it becomes zero in the aquatic solvent. A continuous transition with values from one to zero is assumed by \(S(\mathbf{r})\) at the solute–solvent boundary region \(\Omega_b\). On the other hand, \((1-S(\mathbf{r}))\) is the characteristic function for the solvent domain. The isosurface \(S(\mathbf{r}) = C\) at an appropriate isovalue \(C\) defines the desired molecular surface, and \(S(\mathbf{r})\) is also known as a hypersurface function or a level set function.

In the differential geometry-based multiscale solvation models [8, 9, 30, 31], the governing PDEs of the hypersurface function \(S\) and other physical quantities are derived through a total free energy minimization. In particular, one total free energy functional of solvation for biomolecules at equilibrium can be given as [8, 30, 31]

\[
G_{\text{total}} = \int_{\Omega} \left\{ \gamma \| \nabla S \| + pS + \rho_0 (1-S) U^{\text{vdW}} + S \left[ \rho_m \phi - \frac{\epsilon_m}{2} \| \nabla \phi \|^2 \right] \right. \\
\left. + (1-S) \left[ -\frac{\epsilon_s}{2} \| \nabla \phi \|^2 - k_B T \sum_{j=1}^{N_c} c_j (e^{-q_j \phi / k_B T} - 1) \right] \right\} \, d\mathbf{r},
\]

where \(\gamma\) is the surface tension, \(p\) is the hydrodynamic pressure, \(\rho_0\) is the solvent bulk density, \(U^{\text{vdW}}(\mathbf{r})\) is the VdW potential, \(\rho_m(\mathbf{r})\) is the canonical density of molecular free charges, \(\phi(\mathbf{r})\) is the electrostatic potential, \(\epsilon_m\) is the electric permittivity of the macromolecule, \(\epsilon_s\) is the electric permittivity of the solvent, \(k_B\) is the Boltzmann constant, \(T\) is the temperature, \(c_j\) is the bulk concentration of \(j\)th ionic species, \(q_j\) is the charge of the \(j\)th ionic species, and \(N_c\) is the number of ionic species. We refer to the original references [8, 30, 31] for the detail formulation and physical interpretation of each term in Eq. (1).

The total free energy in Eq. (1) is a functional of two independent functions: the hypersurface function \(S(\mathbf{r})\) and the electrostatic potential \(\phi(\mathbf{r})\). The free energy minimization can be achieved via the Euler–Lagrange variation analysis

\[
\frac{\delta G_{\text{total}}}{\delta S} = 0, \quad \text{and} \quad \frac{\delta G_{\text{total}}}{\delta \phi} = 0.
\]

This leads two coupled nonlinear PDEs [8, 30, 31], that is, a generalized Laplace–Beltrami equation for the characteristic function \(S(\mathbf{r})\),

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

and a generalized Poisson–Boltzmann equation for the electrostatic potential \(\phi(\mathbf{r})\),

\[
\nabla \cdot \left( \epsilon(S) \nabla \phi \right) + (1-S) \sum_{j=1}^{N_c} c_j q_j e^{-q_j \phi / k_B T} = -S \rho_m,
\]

where \(\epsilon(S) = (1-S) \epsilon_s + S \epsilon_m\).

2.2. Potential driven geometric flow equations

A pseudo-transient continuation approach [8, 30–33] is commonly employed to solve the non-linear elliptic PDEs (3) and (4). We refer to [31–33] for more details about the time-dependent Poisson–Boltzmann equation and will focus only on the time-dependent Laplace–Beltrami equations or geometric flow equations [8, 15, 16, 26, 30] in this work. By rearranging the equation
and directly introducing a pseudo time, Eq. (3) can be converted from the time-independent form to a time-dependent form

\[
\frac{\partial S}{\partial t}(\mathbf{r}, t) = \nabla \cdot \left( \frac{\nabla S(\mathbf{r}, t)}{||\nabla S(\mathbf{r}, t)||} \right) + V(\phi; \mathbf{r}),
\]

(5)

where the generalized potential \(V(\phi; \mathbf{r})\) is given as

\[
V(\phi; \mathbf{r}) = \frac{1}{\gamma} \left[ -p + \rho_0 U \nabla \omega - \rho_m \phi + \frac{e_m}{2} \| \nabla \phi \|^2 - \frac{\varepsilon_S}{2} \| \nabla \phi \|^2 - k_B T \sum_{j=1}^{N_c} c_j \left( e^{-q_j \phi/k_B T} - 1 \right) \right].
\]

(6)

We note that in a different application or content, \(V\) will take a different form. Integrating Eq. (5) with respect to the pseudo time \(t\), the initial profile of \(S(\mathbf{r}, t=0)\) evolves into a steady state solution, which satisfies the original Eq. (3). This final characteristic function \(S(\mathbf{r})\) determines the biomolecular surface.

However, the geometric flow equation considered in the previous studies [8, 30, 31] is slightly different from Eq. (5) by introducing an extra scaling term

\[
\frac{\partial S}{\partial t}(\mathbf{r}, t) = \| \nabla S(\mathbf{r}, t) \| \left[ \nabla \cdot \left( \frac{\nabla S(\mathbf{r}, t)}{||\nabla S(\mathbf{r}, t)||} \right) + V(\phi; \mathbf{r}) \right].
\]

(7)

There are two justifications underlying the multiplication of such a scaling factor. On the one hand, as to be shown in our numerical studies, the steady state solutions of Eqs (5) and (7), and the corresponding molecular surfaces, essentially converge to the same place, as \(t\) goes to infinity. On the other hand, after dropping the potential term \(V\), the geometric flow equation (7) reduces to the original mean curvature flow equation

\[
\frac{\partial S}{\partial t}(\mathbf{r}, t) = \| \nabla S(\mathbf{r}, t) \| \nabla \cdot \left( \frac{\nabla S(\mathbf{r}, t)}{||\nabla S(\mathbf{r}, t)||} \right),
\]

(8)

proposed in [15, 16] for the MMS construction. So, the multiplication of the scaling term is consistent with the original molecular surface PDE models. Tracking back to the original MMS model, the introduction of factor \(\| \nabla S(\mathbf{r}, t) \|\) is to stabilize the time integration process [15, 16].

In fact, it is known in the imaging processing community [38, 40] that a severe stability condition is needed when an explicit time integration scheme is used to solve the mean curvature flow equation without the balancing factor \(\| \nabla S(\mathbf{r}, t) \|\). For simplicity, we consider a uniform space partition with the same step size \(h\) in all \(x, y, \) and \(z\) directions. The time increment \(\Delta t\) of an explicit time integration of Eq. (5) is limited by [38, 40]

\[
\Delta t = O(\| \nabla S(\mathbf{r}, t) \| h^2),
\]

(9)

which often runs into an instability, because \(S\) is flat in many regions in the present biomolecular surface application [15, 16]. Alternatively, after multiplying by the extra factor \(\| \nabla S(\mathbf{r}, t) \|\), the stability condition of Eqs (7) and (8) becomes a regular one [38, 40]

\[
\Delta t = O(h^2),
\]

(10)

for the explicit time integration, so that the explicit Euler scheme yields satisfactory results in the original MMS calculations [15, 16].

### 2.3. Numerical algorithms

Two ADI schemes have been previously constructed in [26] for solving Eqs (7) and (8) to generate molecular surfaces. In the present study, we will propose a more accurate and stable ADI scheme for solving the unscaled equation (5) for molecular surface extraction. This is motivated by a few considerations. First, Eq. (5) is the direct pseudo-transient form of the Laplace–Beltrami equation (3) derived from the Euler–Lagrange analysis. Second, due to the use of implicit time integration, there is no need to introduce the extra factor \(\| \nabla S(\mathbf{r}, t) \|\) for a stability relaxation. Finally,
Without the extra factor, Eq. (5) is much simpler so that we can design a much improved ADI discretization.

Without the loss of generality, the potential $V$ can be assumed to be independent from the hypersurface function $S$ and will be regarded as a general potential in the present discussion, other than the one defined in Eq. (6). The partial derivatives will be denoted using subscripts, for example, $S_x := \frac{\partial S}{\partial x}$, $S_{xx} := \frac{\partial^2 S}{\partial x^2}$, and so on. To avoid singularities numerically, a small constant is included in the norm of the gradient [38, 40]

$$
\|\nabla S(\mathbf{r}, t)\| := \sqrt{S_x^2 + S_y^2 + S_z^2 + \eta},
$$

where $\eta = 10^{-7}$ is usually used, whose impact on the accuracy is negligible.

2.3.1. Alternating direction implicit scheme 1. The first ADI scheme constructed in [26] will be referred as the ADI scheme 1 or ADI1 in the present paper. In this scheme, the scaled mean curvature flow Eq. (7) is rewritten as

$$
\frac{\partial S}{\partial t} = S_{xx} + S_{yy} + S_{zz} + \Phi(S)
$$

where the reaction term $\Phi$ is highly nonlinear

$$
\Phi = -S_x^2 S_{xx} + S_y^2 S_{yy} + S_z^2 S_{zz} + 2S_x S_y S_{xy} + 2S_x S_z S_{xz} + 2S_y S_z S_{yz} + V \sqrt{S_x^2 + S_y^2 + S_z^2 + \eta}
$$

(13)

By always evaluating $\Phi$ at the previous time instant $t_n$, the Laplacian $\Delta S$ can be discretized by the standard Douglas ADI scheme [42]. This yields the scheme ADI1 [26].

2.3.2. Alternating direction implicit scheme 2. The second ADI scheme constructed in [26] will be referred as the ADI scheme 2 or ADI2 in the present paper. In this scheme, Eq. (7) is rewritten into a different form

$$
\frac{\partial S}{\partial t} = D^x(S)S_{xx} + D^y(S)S_{yy} + D^z(S)S_{zz} + \Psi(S)
$$

(14)

where

$$
D^x = \frac{S_x^2 + S_y^2 + \eta}{S_x^2 + S_y^2 + S_z^2 + \eta}, \quad D^y = \frac{S_x^2 + S_y^2 + \eta}{S_x^2 + S_y^2 + S_z^2 + \eta}, \quad D^z = \frac{S_x^2 + S_y^2 + \eta}{S_x^2 + S_y^2 + S_z^2 + \eta},
$$

(15)

and

$$
\Psi = -2S_x S_y S_{xy} + S_x S_z S_{xz} + S_y S_z S_{yz} + V \sqrt{S_x^2 + S_y^2 + S_z^2 + \eta}
$$

(16)

Again, the reaction term $\Psi$ is calculated at $t_n$. By evaluating the diffusion coefficients $D^x$, $D^y$, and $D^z$ also at $t_n$, one actually solves a variable–coefficient linear diffusion equation within each time interval $[t_n, t_{n+1}]$ so that the Douglas ADI discretization [42] can be formed. This gives rise to the scheme ADI2 [26].

The ADI1 is known to be more stable than the ADI2, while the ADI2 is slightly more accurate than the ADI1 [26]. Nevertheless, both schemes only have first order of accuracy in time, due to the evaluation of nonlinear reaction terms at $t_n$ time. The further improvement of two ADI schemes seems to be difficult, because $\Phi$ and $\Psi$ involve non-splitting cross derivatives.
2.3.3. Alternating direction implicit scheme 3. To overcome the difficulties associated with the ADI1 and ADI2 schemes, we propose a more accurate and stable ADI scheme, which is referred as the ADI scheme 3 or ADI3 in the present paper. As discussed previously, the unscaled mean curvature flow Eq. (5) will be discretized. Moreover, we will not rewrite the governing equation into a fully Cartesian form, as in the ADI1 and ADI2 schemes. Instead, the mean curvature flow will be treated as a nonhomogeneous diffusion process in the divergence form

\[
\frac{\partial S}{\partial t} = \frac{\partial}{\partial x} \left( \beta(S) \frac{\partial S}{\partial x} \right) + \frac{\partial}{\partial y} \left( \beta(S) \frac{\partial S}{\partial y} \right) + \frac{\partial}{\partial z} \left( \beta(S) \frac{\partial S}{\partial z} \right) + V, \tag{17}
\]

where the conductivity coefficient \( \beta \) is given as

\[
\beta = \frac{1}{\sqrt{S_x^2 + S_y^2 + S_z^2 + \eta}}. \tag{18}
\]

Consider a uniform mesh partition of the computational domain \( \Omega \). Without the loss of generality, we assume the grid spacing \( h \) in all \( x, y, \) and \( z \) directions to be the same. Denote the time increment to be \( \Delta t \) and take \( N_x, N_y, \) and \( N_z \) as the number of grid points in each direction. To facilitate the following discussions, we adopt the following notation at node \((x_i, y_j, z_k, t_n)\):

\[
S_{i,j,k}^n = S(x_i, y_j, z_k, t_n). \tag{19}
\]

The proposed discretization involves half-grid values, such as \( \beta_{i+\frac{1}{2},j,k}^n = \beta(x_{i+\frac{1}{2}}, y_j, z_k, t_n) \), where \( x_{i+\frac{1}{2}} = x_i + \frac{1}{2}h \).

The semi-discretization of Eq. (17) using the Crank–Nicolson time integration at a general spacial node \((x_i, y_j, z_k)\) results in

\[
\frac{S_{i,j,k}^{n+1} - S_{i,j,k}^n}{\Delta t} = \left[ \frac{\partial}{\partial x} \left( \beta \frac{\partial S}{\partial x} \right) \right]_{i,j,k}^{n+\frac{1}{2}} + \left[ \frac{\partial}{\partial y} \left( \beta \frac{\partial S}{\partial y} \right) \right]_{i,j,k}^{n+\frac{1}{2}} + \left[ \frac{\partial}{\partial z} \left( \beta \frac{\partial S}{\partial z} \right) \right]_{i,j,k}^{n+\frac{1}{2}} + V^{n+\frac{1}{2}}, \tag{20}
\]

which is second-order accurate in time. The spatial derivatives in Eq. (19) can be approximated through second-order central finite differences, for example,

\[
\left[ \frac{\partial}{\partial x} \left( \beta \frac{\partial S}{\partial x} \right) \right]_{i,j,k}^{n+\frac{1}{2}} \approx \frac{1}{h^2} \left( \beta_{i+\frac{1}{2},j,k}^n S_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - S_{i,j,k}^{n+\frac{1}{2}} \right), \tag{21}
\]

We thus introduce three finite difference operators \( \delta_{xx}, \delta_{yy}, \) and \( \delta_{zz} \) for the spatial discretization of Eq. (19),

\[
\left[ \frac{\partial}{\partial x} \left( \beta \frac{\partial S}{\partial x} \right) \right]_{i,j,k}^{n+\frac{1}{2}} \approx \delta_{xx} S_{i,j,k}^{n+\frac{1}{2}} := \frac{1}{h^2} \left( \beta_{i+\frac{1}{2},j,k}^n \left( S_{i+\frac{1}{2},j,k}^{n+\frac{1}{2}} - S_{i,j,k}^{n+\frac{1}{2}} \right) + \beta_{i-\frac{1}{2},j,k}^n \left( S_{i-\frac{1}{2},j,k}^{n+\frac{1}{2}} - S_{i,j,k}^{n+\frac{1}{2}} \right) \right), \tag{22}
\]

\[
\left[ \frac{\partial}{\partial y} \left( \beta \frac{\partial S}{\partial y} \right) \right]_{i,j,k}^{n+\frac{1}{2}} \approx \delta_{yy} S_{i,j,k}^{n+\frac{1}{2}} := \frac{1}{h^2} \left( \beta_{i,j+\frac{1}{2},k}^n \left( S_{i,j+\frac{1}{2},k}^{n+\frac{1}{2}} - S_{i,j,k}^{n+\frac{1}{2}} \right) + \beta_{i,j-\frac{1}{2},k}^n \left( S_{i,j-\frac{1}{2},k}^{n+\frac{1}{2}} - S_{i,j,k}^{n+\frac{1}{2}} \right) \right), \tag{23}
\]

We note that in Eqs (21)–(23), \( \beta \) is evaluated at the previous time instant \( t_n \), instead of \( t_{n+\frac{1}{2}} \). This is because nonlinear algebraic systems will be resulted if \( \beta^{n+\frac{1}{2}} \) was used, which introduce considerable difficulties. The approximation of \( \beta^{n+\frac{1}{2}} \) by \( \beta^n \) reduces the temporal accuracy by about a half order.
The calculation of $\beta$ deserves a further illustration. For instance, we consider

$$\beta_{i+1/2,j,k}^n = \left[ \frac{1}{\sqrt{S_x^2 + S_y^2 + S_z^2 + \eta}} \right]_{i+1/2,j,k}^n. \quad (24)$$

The first-order derivatives in Eq. (24) will be approximated by central differences, but the discretization forms will appear to be different, due to a half grid shifting in one Cartesian direction. In particular, we have

$$S_x^n_{i+1/2,j,k} \approx \left( \frac{S^n_{i+1,j,k} - S^n_{i,j,k}}{h} \right)^2, \quad \text{(25)}$$

$$S_y^n_{i+1/2,j,k} \approx \left( \frac{S^n_{i,j+1,k} - S^n_{i,j-1,k}}{4h} + \frac{S^n_{i+1,j+1,k} - S^n_{i+1,j-1,k}}{4h} \right)^2, \quad \text{(26)}$$

$$S_z^n_{i+1/2,j,k} \approx \left( \frac{S^n_{i,j,k+1} - S^n_{i,j,k-1}}{4h} + \frac{S^n_{i+1,j,k+1} - S^n_{i+1,j,k-1}}{4h} \right)^2. \quad \text{(27)}$$

A second order of accuracy in space is maintained in these approximations throughout the domain. Other half grid values of $\beta$ in $y$ and $z$ directions can be similarly approximated.

Based on the aforementioned spatial and temporal approximations, the full discretization form of Eq. (17) can be obtained by taking an average $S^{n+1/2} \approx (S^{n+1} + S^n)/2,$

$$\left[ 1 - \frac{\Delta t}{2} (\delta_{xx} + \delta_{yy} + \delta_{zz}) \right] S^{n+1}_{i,j,k} = \left[ 1 + \frac{\Delta t}{2} (\delta_{xx} + 2\delta_{yy} + 2\delta_{zz}) \right] S^n_{i,j,k} + \Delta t V^n_{i,j,k}^{n+1/2}, \quad \text{(28)}$$

where the external force term $V$ can be evaluated at $t^n_{n+1/2}$ if it is time dependent. If it is time independent, $V^n_{n+1/2}$ will be calculated according to its initial value. The overall accuracy order of Eq. (28) is two in space and about 1.5 in time.

We propose a new Douglas ADI scheme for the mean curvature flow Eq. (17)

$$\left( 1 - \frac{\Delta t}{2} \delta_{xx} \right) S^*_{i,j,k} = \left[ 1 + \frac{\Delta t}{2} (\delta_{xx} + 2\delta_{yy} + 2\delta_{zz}) \right] S^n_{i,j,k} + \Delta t V^n_{i,j,k}^{n+1/2}, \quad \text{(29)}$$

To see the connection between Eqs (28) and (29), one can eliminate $S^*_{i,j,k}$ and $S^{**}_{i,j,k}$ in Eq. (29)

$$\left( 1 - \frac{\Delta t}{2} \delta_{xx} \right) \left( 1 - \frac{\Delta t}{2} \delta_{yy} \right) \left( 1 - \frac{\Delta t}{2} \delta_{zz} \right) S^{n+1}_{i,j,k}$$

$$= \left[ 1 + \frac{\Delta t}{2} (\delta_{xx} + \delta_{yy} + \delta_{zz}) + \frac{\Delta t^2}{4} (\delta_{xx} \delta_{yy} + \delta_{yy} \delta_{zz} + \delta_{zz} \delta_{xx}) - \frac{\Delta t^3}{8} \delta_{xx} \delta_{yy} \delta_{zz} \right] S^n_{i,j,k} + \Delta t V^n_{i,j,k}^{n+1/2}. \quad \text{(30)}$$

This can be further written into the form

$$\left[ 1 - \frac{\Delta t}{2} (\delta_{xx} + \delta_{yy} + \delta_{zz}) \right] S^{n+1}_{i,j,k} = \left[ 1 + \frac{\Delta t}{2} (\delta_{xx} + \delta_{yy} + \delta_{zz}) \right] S^n_{i,j,k} + \Delta t V^n_{i,j,k}^{n+1/2}$$

$$- \frac{\Delta t^2}{4} (\delta_{xx} \delta_{yy} + \delta_{yy} \delta_{zz} + \delta_{zz} \delta_{xx}) \left( S^{n+1}_{i,j,k} - S^n_{i,j,k} \right) + \frac{\Delta t^3}{8} \delta_{xx} \delta_{yy} \delta_{zz} \left( S^{n+1}_{i,j,k} - S^n_{i,j,k} \right) \cdot \quad \text{(31)}$$
In other words, the proposed ADI3 scheme given in Eq. (29) is a higher order perturbation in time of the Crank–Nicolson scheme given in Eq. (28). Thus, similar to Eq. (28), the proposed ADI3 scheme is of second-order accuracy in space and about (1.5)th order accuracy in time.

The proposed ADI3 scheme is unconditionally stable, because in each alternating direction, we have basically a 1D Crank–Nicolson scheme [42]. Great efficiency can be achieved in the biomolecular surface generation. On the one hand, a large time increment Δt is permitted for steady state simulations. On the other hand, the 3D linear algebraic system in the implicit scheme of Eq. (28) is decomposed into one 1D linear algebraic systems in Eq. (29). Moreover, each of these linear systems has a tridiagonal structure and thus can be efficiently solved by the Thomas algorithm [42].

2.4. Initial and boundary conditions

The same initial and boundary conditions will be used in the ADI1, ADI2, and ADI3 schemes to solve the unscaled and scaled geometric flow equations, that is, Eqs (5) and (7). This ensures the uniqueness of the steady state solution. For a sufficiently large domain Ω, the boundary of Ω, that is, ∂Ω, is away from the molecular domain Ωm, so that the characteristic function S can be assumed to be zero on ∂Ω. Following [16,26], the initial values of S will be assumed to be piecewise constants. Consider a macromolecule with total Na number of atoms. Denote the center and radius of the ith atom to be \( \mathbf{r}_i = (x_i, y_i, z_i) \) and \( r_i \), respectively, for \( i = 1, 2, \ldots, N_a \). We then define the domain enclosed by the SAS to be \( D = \bigcup_{i=1}^{N_a} \{ \mathbf{r} : |\mathbf{r} - \mathbf{r}_i| < r_i + r_p \} \), where \( r_p \) is a probe radius. At \( t = 0 \), let S to be

\[
S(x, y, z, 0) = \begin{cases} 
1, & (x, y, z) \in D \\
0, & \text{otherwise.} 
\end{cases}
\] (32)

With initial and boundary values, the hypersurface function \( S(x, y, z, t) \) will be evolved according to the geometric flow equation (5) or (7). To obtain the desired molecular surface of biomolecules, the VdW spheres shall be protected in the computation through defining a control function at the preprocessing stage. Then, this control function will be simply imposed in every time step. We refer to the original studies [16,26] for more details. After a sufficient large time \( t = T \), a steady state solution \( S(x, y, z, T) \) is typically achieved. This hypersurface function \( S(x, y, z, T) \) includes a family of level surfaces. The desired molecular surface can be extracted as an isosurface, \( S(x, y, z, T) = C \). In our computation, the molecular surface is often generated by using \( C \) values ranging from 0.97 to 0.99. However, molecular surfaces generated based on the numerical solution \( S(x, y, z, T) \) are usually not very smooth, due to some numerical artifacts introduced by the finite difference approximations based on Cartesian grids. To compensate these numerical artifacts, we will further integrate the geometric flow equations for a very short time duration, such as 0.01, without the VdW constraints. This extra time integration is short enough so that the final molecular surface essentially keeps the same shape but becomes quite smooth. Finally, standard software packages, such as the MATLAB or VMD, are employed to extract the selected isosurfaces and visualize the resulting molecular surfaces.

3. NUMERICAL VALIDATIONS

In this section, we carry out several numerical experiments to test the stability and accuracy of the proposed ADI3 scheme. For a comparison, the ADI1 and ADI2 schemes are also studied in these benchmark examples. A uniform mesh size \( h \) along with three dimensions will be used in all studies.

3.1. Benchmark test 1 – with analytical solutions

We first solve geometric flow equations with a smooth analytical solution. Consider a cubic domain \( \Omega = [0, 2\pi] \times [0, 2\pi] \times [0, 2\pi] \). The analytical solution is assumed to be [26]

\[
S(x, y, z, t) = \sin(x) \sin(y) \sin(z) \cos(t)
\] (33)
For the ADI3 scheme, the characteristic function \( S(x, y, z, t) \) is evolved according to the unscaled Eq. (5) with the potential term \( V \) being specially defined as

\[
V(x, y, z, t) = -\frac{A_1 - A_2}{A_3} - A_4
\]  

(34)

where

\[
A_1 = -\cos(t)^3 \sin(x) \sin(y) \sin(z)(3\eta(\cos(t))^{-2} + 2(\cos(x) \sin(y) \sin(z))^2 + 2(\sin(x) \cos(y) \cos(z))^2)
\]

(35)

\[
A_2 = 2(\cos(t))^3 \sin(x) \sin(y) \sin(z)((\cos(x) \cos(y) \sin(z))^2 + (\sin(x) \cos(y) \sin(z))^2)
\]

(36)

\[
A_3 = (\eta + (\cos(t))^2(\cos(x) \sin(y) \sin(z))^2 + (\sin(x) \cos(y) \sin(z))^2)^\frac{1}{2}
\]

(37)

\[
A_4 = (\sin(x) \sin(y) \sin(z)) \sin(t)
\]

(38)

For the ADI1 and ADI2 schemes, the characteristic function \( S(x, y, z, t) \) is evolved according to the scaled Eq. (7) subject to another potential term \( V \). The detailed formulation of such a potential \( V \) can be similarly constructed as in [26].

In our computations, the initial value of \( S \) is chosen as the analytical solution at time \( t = 0 \). Because the solution \( S \) is smooth, all three ADI schemes are found to be unconditionally stable. With analytical solutions, the numerical errors of three ADI schemes can be exactly evaluated. Here, the standard \( L_2 \) norm error will be reported, and the stopping time is set to be \( T = 10 \). We first test the spatial order of accuracy of three schemes. By using a sufficiently small \( \Delta t = 0.0005 \), the numerical errors of three ADI schemes are reported in Table I. By examining the errors in successive mesh refinements, the numerically calculated orders of convergence are also listed in Table I. We note that the accuracies and orders of the ADI1 and ADI2 schemes are almost identical, while the orders of the proposed ADI3 scheme are slightly higher. The central finite difference discretizations are used in all three ADI schemes. Thus, the spatial orders of three ADI schemes are all around two in Table I.

In Table II, by using a sufficiently small spacing value \( h = \frac{\pi}{16} \), numerical errors and temporal orders of three ADI schemes are reported. The results of the ADI1 and ADI2 are again close, while the ADI3 scheme performs much better. As we previously discussed, due to the evaluation of nonlinear reaction terms in the previous time instant \( t_n \), both ADI1 and ADI2 schemes are first order of accurate in time. By avoiding to do so, the temporal order of the ADI3 scheme is improved to be about 1.5th order. The numerical results listed in Table II verify these conclusions numerically.

To further illustrate the convergences, the numerical results of Tables I and II are also plotted in Figure 1. In these plots, we performed a least square fitting (the solid lines) of the errors against \( h \) or \( \Delta t \). The slopes \( r \) of the solid straight lines are calculated and listed in the legends of both charts. The magnitude of \( r \) reflects the average convergence rate. It can be seen that the errors of the ADI1 and ADI2 are visually indistinguishable, while the ADI3 scheme clearly performs better.

| Table I. Numerical convergence in space of the benchmark test 1. |
|-----------------|----------------|------------------|------------------|
|                  | ADI1     | ADI2     | ADI3     |
| \( h \)         | \( L_2 \)  | Order    | \( L_2 \)  | Order    | \( L_2 \)  | Order    |
| \( \frac{\pi}{4} \) | 1.65E-02 | 1.65E-02 | 1.31E-02 |
| \( \frac{\pi}{8} \) | 4.53E-03 | 1.87     | 4.53E-03 | 1.87     | 3.37E-03 | 1.96     |
| \( \frac{\pi}{16} \) | 1.22E-03 | 1.89     | 1.22E-03 | 1.89     | 8.89E-04 | 1.92     |
| \( \frac{\pi}{32} \) | 3.48E-04 | 1.81     | 3.47E-04 | 1.81     | 2.32E-04 | 1.94     |

Copyright © 2013 John Wiley & Sons, Ltd.


DOI: 10.1002/cnm
We next consider a diatomic system with two atoms of the same radius \( r_1 = r_2 = 2 \) Å. The centers of these two atoms are located at \( r_1 = (-2, 0, 0) \) and \( r_2 = (2, 0, 0) \), and the computational domain is chosen as \((x, y, z) \in [-6, 6] \times [-4, 4] \times [-4, 4] \). Based on such a domain, the initial value of \( S \) is defined as piecewise constants, as discussed in Section 2.4, with a probe radius \( r_p = 1.5 \) Å. The mean curvature flow equations with \( V = 0 \) are considered, and the integration will be carried out until \( T = 2 \).

Because no exact solution is available, to benchmark our computations, we construct two reference solutions by using the explicit Euler scheme to solve the scaled Eq. (7) and unscaled Eq. (5), respectively. For this purpose, we first study the stability of the explicit Euler scheme. For the scaled Eq. (7), by testing several \( h \) and \( \eta \) values, the stability condition is numerically detected as

\[
\Delta t \leq 0.5h^2.  
\]

For the unscaled Eq. (5), the stability condition is approximately found to be

\[
\Delta t \leq 1.16 \times 10^{-4} h^2,  
\]

for \( \eta = 10^{-7} \). We note that these two conditions numerically verified the theoretical results presented in Eqs (9) and (10), because at the flat region of \( S, \| \nabla S \| \) is proportional to \( \sqrt{\eta} \). Here, by taking \( \eta = 10^{-7} \), the ratio between two stability constants involved in Eqs (39) and (40) is actually on the order of \( \sqrt{\eta} \). If a smaller \( \eta \) value, such as \( \eta = 10^{-10} \), is used, the stability condition given in Eq. (40) for the unscaled equation becomes more severe.

We next discuss the role of the parameter \( \eta \). By taking \( \eta \) as a vanishing number such as \( 10^{-7} \), one avoids the numerical singularities in the MMS computation, while the impact of \( \eta \) on the steady state solution \( S \) is negligible. For instance, for the present diatomic system with \( h = 0.1 \), we have found that the difference in the \( L_2 \) norm between two ADI3 solutions with \( \eta = 10^{-7} \) and \( \eta = 10^{-10} \) is as small as \( 1.05 \times 10^{-4} \). Thus, \( \eta = 10^{-7} \) is employed in all studies of this paper except for the present subsection. For the present study, the stability condition given in Eq. (40) is still very expensive for integrating to \( T = 2 \). We thus choose \( \eta = 1 \) in this subsection only, so that the explicit Euler scheme can be employed to generate the reference solution in the unscaled equation case. In particular, we
generate two reference solutions by the Euler scheme based on a fine mesh with $\Delta t = 2 \times 10^{-5}$ and $h = 0.05 \, \text{Å}$. Even though there are some modeling differences between the present reference solutions and the ideal hypersurface functions with a vanishing $\eta$, these reference solutions enable us to quantitatively examine the numerical errors of the ADI schemes in solving this diatomic system.

We first study the spatial accuracy of the ADI schemes. By using $\Delta t = 0.005$, the $L_2$ errors of three ADI solutions comparing with the corresponding reference solutions are shown in Table III. It is interesting to see that the spatial errors and orders of three ADI schemes are almost identical, even though different PDEs and reference solutions are underlying these approximations. Moreover, the spatial orders of three ADI schemes become much worse than those of the benchmark test 1. This is because the solution $S$ is defined to be discontinuous at the beginning in this example and is in general non-smooth during the time integration. The low regularity of $S$ will degrade central finite differences to about first order of accuracy in space. To demonstrate this point, the $L_2$ errors of three ADI schemes with respective to different $h$ values are plotted in Figure 2(a). The least squares fitted convergence rates of three ADI schemes are obviously just first order.

The temporal integration of the ADI schemes will also be affected by the non-smoothness of the solution. By using $h = 0.05$, the numerical errors of three ADI schemes based on different $\Delta t$ values are listed in Table IV. It can be observed that the error of the ADI2 scheme is unstable under the reported $\Delta t$ values. In fact, the ADI2 scheme becomes stable only when $\Delta t \leq 0.03$ for $h = 0.05$. The similar conditional stability has also been reported in the literature [26], when the ADI2 scheme is applied

| Table III. Numerical convergence in space of the benchmark test 2. |
|---------------------|---------------------|---------------------|
| $h$ | ADI1 | ADI2 | ADI3 |
| | $L_2$ | Order | $L_2$ | Order | $L_2$ | Order |
| 0.8 | $9.97E-02$ | | $9.97E-02$ | | $1.01E-01$ | |
| 0.4 | $6.41E-02$ | 0.64 | $6.41E-02$ | 0.64 | $6.44E-02$ | 0.65 |
| 0.2 | $2.62E-02$ | 1.29 | $2.62E-02$ | 1.29 | $2.66E-02$ | 1.28 |
| 0.1 | $9.50E-03$ | 1.46 | $9.50E-03$ | 1.46 | $1.02E-02$ | 1.40 |

![Figure 2. Numerical results of the benchmark test 2. (a) Numerical convergence in space and (b) numerical convergence in time.](image_url)

| Table IV. Numerical convergence in time of the benchmark test 2. |
|---------------------|---------------------|---------------------|
| $\Delta t$ | ADI1 | ADI2 | ADI3 |
| | $L_2$ | Order | $L_2$ | Order | $L_2$ | Order |
| 1.0 | $4.55E-01$ | | $3.64E-01$ | | $3.32E-01$ | |
| 0.5 | $4.21E-01$ | 0.11 | $1.94E+00$ | $N.A.$ | $1.48E-01$ | 0.65 |
| 0.25 | $3.75E-01$ | 0.17 | $1.15E+02$ | $N.A.$ | $7.78E-02$ | 0.92 |
| 0.125 | $2.76E-01$ | 0.44 | $2.40E+03$ | $N.A.$ | $2.20E-02$ | 1.82 |
| 0.0625 | $1.14E-01$ | 1.27 | $3.01E+05$ | $N.A.$ | $3.41E-03$ | 2.69 |
to real biomolecular simulations, in which the initial value of $S$ is always defined discontinuously. On the other hand, the unconditional stability of the ADI1 and ADI3 schemes is not impacted by the non-smoothness of $S$. Moreover, the ADI3 scheme yields a much higher accuracy than the ADI1 scheme. In Figure 2(b), the temporal convergence rates of the ADI1 and ADI3 schemes are analyzed. It can be seen that the ADI3 scheme is still able to achieve about (1.5)$^{th}$ order of accuracy for a problem with low regularity solutions, while the order of the ADI2 scheme is as low as 0.5. Thus, the proposed ADI3 scheme will be more accurate and stable than the existing ADI schemes in real biomolecular simulations.

3.3. Molecular surfaces of atomic systems

After studying the stability and accuracy of the ADI schemes in solving geometric flow PDEs, we next investigate the molecular surfaces that are constructed based on the ADI solutions. The steady state hypersurface function $S(x, y, z, T)$ defines a family of isosurfaces. In the present study, the molecular surface will be extracted as an isosurface $S = C$ and is denoted as $\Gamma$. To quantitatively characterize such a molecular surface, we will study the area of this surface and the volume enclosed by this surface.

The surface area and volume of the sharp surface $\Gamma$ can be calculated through integrals of the original hypersurface function $S$, because $S$ is the so-called Eulerian representation of the molecular surface [8,30,31]. Consider a density function $f$ over the domain $\Omega$ with a uniform mesh. The surface integral of $f$ over $\Gamma$ can be approximated by [28,29,43]

$$\int_{\Gamma} f(x, y, z) dS \approx \sum_{(i,j,k) \in I} \left( f(x_o, y_j, z_k) \frac{|n_x|}{h} + f(x_i, y_o, z_k) \frac{|n_y|}{h} + f(x_i, y_j, z_o) \frac{|n_z|}{h} \right) h^3$$

(41)

where $(x_o, y_j, z_k)$ is the intersecting point of the interface $\Gamma$ and the $x$ mesh line that passes through the point $(x_i, y_j, z_k)$, and $n_x$ is the $x$ component of the unit normal vector at the point $(x_o, y_j, z_k)$. The variables in $y$ and $z$ directions are similarly defined. The summation in Eq. (41) needs to be conducted only in a small set of grid points, that is, $I$. In particular, we define irregular grid points to be points with neighbor from the other side of the interface $\Gamma$. The set $J$ is the set of irregular grid points inside or on the interface $\Gamma$ [43]. The volume integral of a density function can be simply approximated by

$$\int_{\Omega_\Gamma} f dV \approx \sum_{(i,j,k) \in J} f(x_i, y_j, z_k) h^3$$

(42)

where $\Omega_\Gamma$ is the domain enclosed by the $\Gamma$, and $J$ is the set of the grid points inside $\Omega_\Gamma$. The surface area and volume of the molecular surface $\Gamma$ can be computed by using Eqs (41) and (42) with $f = 1$.

Before we apply Eqs (41) and (42) to molecular surface PDE models, we first validate the area and volume formulas given in Eqs (41) and (42) with $f = 1$.
Table V. Surface area of the Van der Waals surface (unit: Å²).

<table>
<thead>
<tr>
<th>h</th>
<th>System</th>
<th>0.5</th>
<th>0.25</th>
<th>0.125</th>
<th>0.0625</th>
<th>Exact value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>Monatomic</td>
<td>48.36</td>
<td>49.70</td>
<td>50.15</td>
<td>50.23</td>
<td>50.27</td>
</tr>
<tr>
<td>0.25</td>
<td>Diatomic</td>
<td>96.72</td>
<td>99.40</td>
<td>100.31</td>
<td>100.46</td>
<td>100.53</td>
</tr>
</tbody>
</table>

Table VI. Volume enclosed by the Van der Waals surface (unit: Å³).

<table>
<thead>
<tr>
<th>h</th>
<th>System</th>
<th>0.5</th>
<th>0.25</th>
<th>0.125</th>
<th>0.0625</th>
<th>Exact value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>Monatomic</td>
<td>32.13</td>
<td>32.95</td>
<td>33.35</td>
<td>33.46</td>
<td>33.51</td>
</tr>
<tr>
<td>0.25</td>
<td>Diatomic</td>
<td>64.25</td>
<td>65.91</td>
<td>66.70</td>
<td>66.93</td>
<td>67.02</td>
</tr>
</tbody>
</table>

Figure 3. Time history of the area and enclosed volume of the molecular surfaces generated by three alternating direction implicit schemes. (a) Area and (b) volume.

area and volume converge to the analytical values of the VdW surfaces. This validates the area and volume formulas given in Eqs (41) and (42) and our numerical implementations.

With the established area and volume formulas, we can quantitatively compare the molecular surfaces generated by the three ADI schemes. As we discussed previously, the governing PDEs for the ADI1/ADI2 and ADI3 are different by a scaling term. Theoretically, the steady state molecular surfaces of two PDE models should be the same. To numerically verify this point, we take the previously studied diatomic system as an example. The same piecewise constant initial values for $S$ are employed in all ADI approximations of the mean curvature flow equations. The grid is also chosen as the same for three ADI schemes with $\Delta t = 0.01$ and $h = 0.1$. The time histories of the calculated area and volume of three ADI schemes are shown in Figure 3. It can be seen that the area and volume of three ADI schemes quickly converge to the steady state as $t$ increases. The steady state values of the ADI1 and ADI2 schemes are identical and are very close to those of the ADI3 scheme. The relative difference between the steady state values of the ADI1/ADI2 and ADI3 results is less than the 0.5% for the present example. Such a non-vanishing difference might be due to the approximation errors accumulated in numerical solutions of the PDEs and calculations of the area and volume.

In our last validation, we consider the generation of molecular surfaces for a moving diatomic system. Consider two atoms of the same radius $r = 2$ Å. Let the atomic centers be $\mathbf{r}_1 = (-\frac{L}{2}, 0, 0)$ and $\mathbf{r}_2 = (\frac{L}{2}, 0, 0)$, where $L \geq 4$ is the separating distance between the atomic centers. We examine the change in the molecular surface with $L$ increases from $L = 4$ to $L = 4.8$. In all cases, we take $\Delta t = 0.01$ and $h = 0.1$, and a sufficiently large computational domain $\Omega$ is used. The isosurfaces generated at $S = 0.98$ based on the steady state solutions of the ADI3 scheme is shown in Figure 4.
For a small $L$ value, the molecular surface consists of two parts, that is, contact surfaces of two spheres and a catenoid, which is a reentrant surface that connects the two atoms. As the separation length $L$ is gradually increased, the neck of the catenoid becomes thinner and thinner. Eventually, the molecular surface breaks into two disjoint pieces, without generating any geometrical singularities [16].

Although the topological change of the molecular surface seems to be abrupt at a critical distance, for which the catenoid is broken, the changes in the surface area and enclosed volume with respect to $L$ are actually quite smooth. To see this, the area and volume of the molecular surfaces generated by the three ADI schemes at different $L$ values are listed in Table VII. We first note that the area and volume formulas given in Eqs (41) and (42) involve much larger approximation errors in the present study than those in Tables V and VI. For a comparison, we add one more $L$ value, $L = 5$, in Table VII. For both $L = 4.8$ and $L = 5$, the resulting molecular surfaces are simply the VdW surface of two isolated spheres. It can be seen from Table VII that the volumes are the same for $L = 4.8$ and $L = 5$, while the areas are slightly different. Such minor differences are due to the discretization errors in solving the PDEs and can be neglected in our analysis. However, if one compares the areas and volumes at $L = 5$ with the exact values listed in Tables V and VI, one can see large deviations, especially in areas. The area and volume formulas yield excellent estimates in the previous study based on smooth Gaussian functions. Nevertheless, the hypersurface function $S$ solved from geometric flow PDEs is visually a step function [16]. A very steep change from $S = 1$ to $S = 0$ occurs around the molecular surface, so that the gradient approximations underlying Eq. (41) are very inaccurate. Thus, Eqs (41) and (42) have much larger approximation errors in the present study.

However, the approximation errors in the area and volume calculations are systematically presented in all data listed in Table VII. If we just focus on the changes of the area and volume with respect to the separation length $L$, such errors will not interfere with our analysis. Indeed, there are certain patterns that can be observed in the changes of area and volume. In particular, when $L$
increases, the area becomes larger and larger, until a constant value is reached. This actually means that for each position, the generated molecular surface is the one of the smallest area and wrapping the VdW spheres. In this sense, such a molecular surface is referred as the MMS [16]. On the other hand, the volume is not a monotonic function of $L$. However, when $L$ is large enough so that the catenoid is broken, the volume drops to a constant level. Finally, we note that there are some minor differences between the ADI3 estimates and those of the ADI1 and ADI2, due to the use of slightly different models and numerical discretizations. However, in general, the results of the ADI3 scheme are consistent with those of the ADI1 and ADI2, while the ADI3 scheme is usually more stable and accurate.

4. BIOLOGICAL APPLICATIONS

In this section, we validate the proposed ADI3 algorithm for realistic biomolecular surface generation. For simplicity, the other two ADI schemes will not be examined. The geometric flow equation (5) with and without the potential term $V$ will be studied. The solvation analysis of two types of biological systems including small compounds and large proteins will be studied in details. Again, in all cases, a uniform mesh size $h$ is used along all three dimensions, and a large enough computational domain is chosen.

4.1. Mean curvature flow model

In this subsection, the ADI3 algorithm is applied to solve the mean curvature flow Eq. (5) with $V = 0$, for various biological systems. The resulting surface is known as the MMS in the literature [15, 16]. For a comparison, the SES generated by the MSMS package [14] will be considered in several examples.

We first generate the MMS of a B-DNA double helix segment with 494 atoms (PDB ID: 425D). Here, the mean curvature flow equation is solved with $h = 0.5$ Å and $\Delta t = 0.1$, until $T = 10$. The corresponding MMS extracted at $S = 0.98$ is shown in Figure 5(a). The SES generated by the MSMS package with the probe radius being 1.5 Å is depicted in Figure 5(b). In comparing these two biomolecular surfaces, it can be seen that the MMS is smoother, especially at the groove.

Figure 5. The minimal molecular surface (a) and the solvent-excluded surface (b) of a B-DNA double helix segment (PDB ID: 425D).
regions. This is because the total surface area is minimized in the MMS model following the mean curvature flow equation. Here, the area of the MMS is found to be 3468.42 Å², which is smaller than the numerical area of the SES (3500.36 Å²). On the other hand, the enclosed volume of the MMS (7351.38 Å³) is larger than that of the SES (7143.96 Å³), also owing to the surface minimization.

We next study the hemoglobin, an important metalloprotein in red blood cells (PDB ID: 1hga). What is interesting in this biological system with 4649 atoms is the presence of a small pinhole near the center of four globular protein subunits. It is known in the MMS generation [15, 16] that if only the VdW regions are protected in the numerical solution of the mean curvature flow Eq. (5), the inaccessible internal cavities and open cavities of macromolecules will not be identified, because creating a cavity will lead to a larger total surface area. To capture such cavities, additional geometric constraints have to be introduced into the control function at the preprocessing stage. In particular, following [15, 16], regions outside the SAS [10] will be protected with $S_{D0}$ through properly re-defining the control function, which usually takes a very little extra CPU time. Here, the probe radius is taken as 1.5 Å in the SAS and SES definitions. Then, the mean curvature flow equation with the new control function is solved with $h = 0.5$ Å and $\Delta t = 0.1$, until $T = 10$. Now, an open cavity is clearly seen in the generated MMS at $S = 0.98$ (Figure 6). In comparing with the SES of the hemoglobin, the size of the pinhole of the MMS is slightly smaller than that of the SES. Again, the surface area of the MMS (17485.47 Å²) is smaller than that of the SES (19751.11 Å²), while the enclosed volume of the MMS (80722.25 Å³) is larger than that of the SES (77156.07 Å³). These studies clearly demonstrate the main property of the MMS model, that is, the minimization of total surface area subject to the prescribed geometrical constraints.

We finally examine the efficiency of the proposed ADI algorithm in solving the geometric flow equation for a series of proteins. A set of 23 proteins, which has been frequently used for testing the previous solvation models [8, 9, 31], will be considered. In addition, several ultra large biological systems studied in [20, 28] will also be examined. The discrete atomic structures of these proteins are prepared as in the literature. In particular, hydrogen atoms are added to obtain full all-atom models, and extra water molecules that are attached to proteins are excluded in all structures. The CHARMM22 force field is employed to derive partial charges at atomic sites and atomic VdW radii in angstroms.

For the MMS generation, the present procedure consists of two parts, that is, the solution of the geometric flow PDEs and the isosurface extraction or visualization. We will report the CPU time for the first part here, that is, generating the steady state solution by using the ADI algorithm. In the second part, the molecular surface visualization is conducted through standard software packages, such as the MATLAB and VMD. Thus, the CPU time required for the second part is independent of the proposed ADI algorithm. Because the proposed ADI algorithm is more stable and accurate than the existing ADI methods, this enables us to use a large time increment, such as $\Delta t = 1$, to improve

![Figure 6. The minimal molecular surface (a) and the solvent-excluded surface (b) of the hemoglobin (PDB ID: 1hga).](image-url)
Table VIII. The surface area, enclosed volume, and corresponding CPU time in seconds of the minimal molecular surfaces.

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>No. of atoms</th>
<th>Area</th>
<th>Volume</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ajj</td>
<td>519</td>
<td>2038.11</td>
<td>4972.75</td>
<td>0.43</td>
</tr>
<tr>
<td>2erl</td>
<td>573</td>
<td>2142.13</td>
<td>5581.13</td>
<td>0.41</td>
</tr>
<tr>
<td>1bbi</td>
<td>576</td>
<td>2434.84</td>
<td>5615.00</td>
<td>0.59</td>
</tr>
<tr>
<td>1vii</td>
<td>596</td>
<td>2309.51</td>
<td>5401.63</td>
<td>0.40</td>
</tr>
<tr>
<td>1cbn</td>
<td>648</td>
<td>2204.76</td>
<td>6088.25</td>
<td>0.51</td>
</tr>
<tr>
<td>2pde</td>
<td>667</td>
<td>2534.55</td>
<td>6411.00</td>
<td>0.55</td>
</tr>
<tr>
<td>1sh1</td>
<td>702</td>
<td>2493.44</td>
<td>6889.13</td>
<td>0.54</td>
</tr>
<tr>
<td>1fca</td>
<td>729</td>
<td>2505.68</td>
<td>7337.00</td>
<td>0.49</td>
</tr>
<tr>
<td>1ptq</td>
<td>795</td>
<td>2681.86</td>
<td>7590.00</td>
<td>0.62</td>
</tr>
<tr>
<td>1uxc</td>
<td>809</td>
<td>2637.54</td>
<td>7243.13</td>
<td>0.57</td>
</tr>
<tr>
<td>1fxd</td>
<td>824</td>
<td>2739.86</td>
<td>8245.13</td>
<td>0.62</td>
</tr>
<tr>
<td>1bor</td>
<td>832</td>
<td>2687.56</td>
<td>7607.50</td>
<td>0.66</td>
</tr>
<tr>
<td>1hpt</td>
<td>858</td>
<td>3069.32</td>
<td>8146.75</td>
<td>0.63</td>
</tr>
<tr>
<td>1bpi</td>
<td>896</td>
<td>2991.37</td>
<td>8553.75</td>
<td>0.72</td>
</tr>
<tr>
<td>1mbg</td>
<td>903</td>
<td>2837.37</td>
<td>8368.50</td>
<td>0.57</td>
</tr>
<tr>
<td>1r69</td>
<td>997</td>
<td>2835.00</td>
<td>9304.25</td>
<td>0.63</td>
</tr>
<tr>
<td>1neq</td>
<td>1187</td>
<td>4372.92</td>
<td>11467.13</td>
<td>1.24</td>
</tr>
<tr>
<td>451c</td>
<td>1216</td>
<td>3733.98</td>
<td>11864.13</td>
<td>0.74</td>
</tr>
<tr>
<td>1a2s</td>
<td>1272</td>
<td>3938.48</td>
<td>12583.88</td>
<td>0.92</td>
</tr>
<tr>
<td>1svr</td>
<td>1435</td>
<td>4226.87</td>
<td>12844.00</td>
<td>1.05</td>
</tr>
<tr>
<td>1frd</td>
<td>1478</td>
<td>4047.54</td>
<td>14235.88</td>
<td>0.91</td>
</tr>
<tr>
<td>1r63</td>
<td>2065</td>
<td>6430.95</td>
<td>19286.75</td>
<td>1.64</td>
</tr>
<tr>
<td>1a7m</td>
<td>2809</td>
<td>6952.82</td>
<td>25650.00</td>
<td>1.99</td>
</tr>
<tr>
<td>1beb</td>
<td>4972</td>
<td>11269.56</td>
<td>48149.63</td>
<td>3.65</td>
</tr>
<tr>
<td>1vng</td>
<td>8808</td>
<td>16869.10</td>
<td>86000.25</td>
<td>5.33</td>
</tr>
<tr>
<td>1tas</td>
<td>12636</td>
<td>22731.36</td>
<td>122470.88</td>
<td>7.43</td>
</tr>
<tr>
<td>1maa</td>
<td>33257</td>
<td>57205.56</td>
<td>322510.13</td>
<td>46.27</td>
</tr>
<tr>
<td>4bfl</td>
<td>46160</td>
<td>60314.91</td>
<td>461149.75</td>
<td>30.51</td>
</tr>
</tbody>
</table>

Figure 7. The CPU time in seconds consumed by the ADI3 algorithm for generating the minimal molecular surfaces.

The area and enclosed volume of the molecular surfaces, and the corresponding CPU time are listed in Table VIII. Here, all computations are conducted on an SGI Xeon E5-4640 CPU core operating at 2.4 GHz and 8 GB of memory. It is seen in Table VIII that for small protein systems, the generation of the MMS steady state solutions can be performed within 1 s. In general, the CPU time increases as the number of atoms becomes larger. To achieve a better understanding, we plot the CPU time against the number of atoms in Figure 7. It can be observed that the CPU time is roughly a linear function of the number of atoms. In particular, we denote the CPU time and number of atoms to be $t_S$ and $N_a$, respectively. A least squares linear fitting can be conducted to represent $t_S$ in seconds as a function of $N_a$: $t_S = 0.000893N_a - 0.169118$. This linear relation is a
ADI SCHEME FOR MOLECULAR SURFACE GENERATION

Valuable property in real biomolecular simulations, because it allows us to predict the applicability of the proposed ADI algorithm in treating extra large proteins and estimate the corresponding execution time.

4.2. Potential driven geometric flow

In this subsection, the ADI3 algorithm is applied to solve the geometric flow PDE (5) with a non-vanishing $V$. This gives rise to general molecular surfaces, other than the MMS. Various different forms of the potential $V$ will be examined. We note that the ADI3 algorithm is unconditionally stable in all computations.

We first consider a constant potential $V$ over the entire domain. A diatomic system with radii $r_1 = r_2 = 2$ Å and centers $r_1 = (-2.2, 0, 0)$ and $r_2 = (2.2, 0, 0)$ is studied. Here, we choose $\Delta t = 0.01$, and $T = 10$ in the ADI3 method. The isosurfaces extracted at $S = 0.98$ for three $V$ values are shown in Figure 8. It can be seen that without altering the structures of two VdW balls, the potential $V$ could deform the shape of the molecular surface. When the potential is positive, the potential is repulsive in the sense that the catenoid becomes fatter. The potential becomes attractive when it is negative, for which the catenoid becomes thinner. The contour plots of these three surfaces at the cross section $z = 0$ are shown in Figure 9. We can see that three surfaces have the same contact surfaces, which are parts of two spheres. The potential $V$ mainly affects the reentrant surface that connects the two atoms.

We next study the local modification of the molecular surface by defining $V$ as some spatial dependent functions. The same diatomic system and ADI parameters as in the previous example are employed. The general potential is assumed to take the form

$$V(x, y, z) = A + B \left( \frac{4}{\sqrt{x^2 + (y - 4.6)^2 + z^2}} \right)^7,$$

where $A$ and $B$ are some constants. This potential is centered at $(0, 4.6, 0)$ and decays rapidly away from the center. Thus, its impact on the molecular surface of the diatomic system is asymmetric, that is, it mainly affects the side above the $x$-$z$ plane. Two pairs of parameter values are considered. When $A = 0$ and $B = -0.427$, the corresponding molecular surface is shown in Figure 10(a). In part (b) of the same figure, the contour plots at the cross section $z = 0$ are shown for this molecular surface and the MMS with $V = 0$. It is seen that this potential is partially attractive. In Figure 11, the potential driven molecular surface and its contour for $A = -0.176$ and $B = 0.169$ are depicted. Now, the potential is partially repulsive.

![Figure 8](image)

Figure 8. The molecular surfaces generated by the potential driven geometric flow equation (5) with different $V$ values. (a) $V = 0.5$, (b) $V = 0$, and (c) $V = -0.5$.

![Figure 9](image)

Figure 9. The contour plot of molecular surfaces at the cross section $z = 0$. The red solid line is for $V = 0$, while green and blue broken lines are for $V = 0.5$ and $V = -0.5$, respectively.
W. TIAN AND S. ZHAO

Figure 10. (a) The potential driven molecular surface with \( A = 0 \) and \( B = -0.427 \) and (b) the corresponding contour plot at the cross section \( z = 0 \) (blue dash line). Here, the red solid line is for the minimal molecular surface with \( V = 0 \).

Figure 11. (a) The potential driven molecular surface with \( A = -0.176 \) and \( B = 0.169 \) and (b) the corresponding contour plot at the cross section \( z = 0 \) (blue dash line). Here, the red solid line is for the minimal molecular surface with \( V = 0 \).

We finally consider a generalized VdW potential \( V \), which has been frequently used in the differential geometry based multiscale solvation models [8, 9, 31]. In particular, without considering the feedback from the electrostatic potential \( \phi \), the potential \( V \) of Eq. (6) can be rewritten as

\[
V(\mathbf{r}) = -\frac{P}{\gamma} + \frac{\phi_0}{\gamma} U^{\text{att}}(\mathbf{r}),
\]

where the attractive potential \( U^{\text{att}} \) is the modeled based on the 12-6 Lennard–Jones potential [8, 9, 31]. In particular, the Lennard–Jones potential with respect to the \( i \)th partial charge at location \( \mathbf{r}_i \) is given as

\[
U^{\text{LJ}}_i(\mathbf{r}) = \beta_i \left[ \left( \frac{\sigma_i + \sigma_s}{\lVert \mathbf{r} - \mathbf{r}_i \rVert} \right)^{12} - 2 \left( \frac{\sigma_i + \sigma_s}{\lVert \mathbf{r} - \mathbf{r}_i \rVert} \right)^6 \right],
\]

where \( \beta_i \) is the well-depth parameter, and \( \sigma_i \) and \( \sigma_s \) are solute atomic and solvent radii, respectively. The attractive portion \( U^{\text{att}} \) and the repulsive portion \( U^{\text{rep}} \) can then be obtained via a decomposition of the Lennard–Jones potential. Two types of decompositions are commonly used. The first one is the so-called '6-12' decomposition

\[
U^{\text{att,6/12}}_i(\mathbf{r}) = -2\beta_i \left( \frac{\sigma_i + \sigma_s}{\lVert \mathbf{r} - \mathbf{r}_i \rVert} \right)^6, \quad U^{\text{rep,6/12}}_i(\mathbf{r}) = \beta_i \left( \frac{\sigma_i + \sigma_s}{\lVert \mathbf{r} - \mathbf{r}_i \rVert} \right)^{12}.
\]

Based on the Weeks–Chandler–Anderson (WCA) theory [44], the WCA decomposition is the other one

\[
U^{\text{att,WCA}}_i(\mathbf{r}) = \begin{cases} 
-\beta_i(\mathbf{r}), & 0 < \lVert \mathbf{r} - \mathbf{r}_i \rVert < \sigma_i + \sigma_s \\
U^{\text{LJ}}_i(\mathbf{r}), & \lVert \mathbf{r} - \mathbf{r}_i \rVert \geq \sigma_i + \sigma_s
\end{cases},
\]

\[
U^{\text{rep,WCA}}_i(\mathbf{r}) = \begin{cases} 
U^{\text{LJ}}_i(\mathbf{r}) + \beta_i(\mathbf{r}), & 0 < \lVert \mathbf{r} - \mathbf{r}_i \rVert < \sigma_i + \sigma_s \\
0, & \lVert \mathbf{r} - \mathbf{r}_i \rVert \geq \sigma_i + \sigma_s
\end{cases}.
\]

Following [8, 9, 31], the WCA decomposition is employed in the present study. The Lennard–Jones parameters are chosen as follows: \( \sigma_i \) is taken from atomic radius, and \( \sigma_s \) is chosen to be 0.65Å [7].

Copyright © 2013 John Wiley & Sons, Ltd.


DOI: 10.1002/cnm
Given \( \sigma_i \) and \( \sigma_j \), the value of \( \beta_i \) can be solved from the equation
\[
\beta_i = \frac{1}{D_i} \left[ \left( \frac{\sigma_i + \sigma_j}{|r_i - r_j|} \right)^{12} - 2 \left( \frac{\sigma_i + \sigma_j}{|r_i - r_j|} \right)^6 \right] = D_i,
\]
which holds if \( r \) is on the VdW surface of the atom. Here, the constant \( D_i \) should take different values for various types of atoms. For simplicity, we use a uniform value \( D_i = 1 \). The surface tension \( \gamma \) is treated as a scaling parameter with the value \( \gamma = 1/15 \). We set the bulk density coefficient to be \( \rho_b \gamma = 2 \) and choose pressure coefficient as \( p/\gamma = 0.2 \).

By using the VdW potential \( V \) given in Eq. (45), the ADI3 algorithm is applied to generate molecular surfaces for the protein systems considered in the last subsection. Besides the potential \( V \), the other structural parameters remain to be the same. The ADI parameters are also chosen as \( h = 0.5\, \text{Å} \), \( \Delta t = 1 \), and \( T = 10 \) for efficiency. The calculated area and enclosed volume, and the consumed CPU time are reported in Table IX. In comparing with the results of the MMSs given in Table VIII, it can be seen that for every protein, the volume becomes smaller, while the area becomes larger, after applying the VdW potential \( V \). This is because the VdW potential \( V \) defined in (45) is attractive by taking negative values near the VdW surface. The resulted molecular surface is bended towards the center. Consequently, the volume becomes smaller. On the other hand, because the MMS attains the smallest surface area, any deformation of the molecular surface will result in a larger area. The visual differences between the MMSs and VdW potential driven molecular surfaces for two proteins (PDB IDs: 1blb and 1neq) are shown in Figure 12. It can be seen that the volume of the molecular surface becomes smaller at certain places after applying the VdW potential \( V \).

In Table IX, two CPU results are reported, that is, \( t_S \) is for the ADI algorithm, and \( t_V \) is for the VdW potential setup. It is seen that most execution time is consumed by the calculation of the VdW potential \( V \). If we just consider \( t_S \), the present CPU results are slightly longer than those of the MMSs, because the incorporation of \( V \) in the ADI algorithm needs just two more arithmetic

<table>
<thead>
<tr>
<th>PDB ID</th>
<th>No. of atoms</th>
<th>Area</th>
<th>Volume</th>
<th>( t_S )</th>
<th>( t_V )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1ajj</td>
<td>519</td>
<td>2099.00</td>
<td>4781.38</td>
<td>0.50</td>
<td>2.22</td>
</tr>
<tr>
<td>2erl</td>
<td>573</td>
<td>2199.44</td>
<td>5384.38</td>
<td>0.42</td>
<td>2.28</td>
</tr>
<tr>
<td>1blb</td>
<td>576</td>
<td>2529.93</td>
<td>5341.25</td>
<td>0.59</td>
<td>2.92</td>
</tr>
<tr>
<td>1vii</td>
<td>596</td>
<td>2432.53</td>
<td>5115.13</td>
<td>0.39</td>
<td>2.28</td>
</tr>
<tr>
<td>1ebn</td>
<td>648</td>
<td>2268.93</td>
<td>5860.25</td>
<td>0.52</td>
<td>3.05</td>
</tr>
<tr>
<td>2pde</td>
<td>667</td>
<td>2619.43</td>
<td>6170.38</td>
<td>0.54</td>
<td>3.10</td>
</tr>
<tr>
<td>1sh1</td>
<td>702</td>
<td>2597.68</td>
<td>6633.38</td>
<td>0.48</td>
<td>3.13</td>
</tr>
<tr>
<td>1fca</td>
<td>729</td>
<td>2612.16</td>
<td>7070.50</td>
<td>0.49</td>
<td>3.22</td>
</tr>
<tr>
<td>1ptq</td>
<td>795</td>
<td>2777.14</td>
<td>7315.75</td>
<td>0.58</td>
<td>4.11</td>
</tr>
<tr>
<td>1uxc</td>
<td>809</td>
<td>2741.14</td>
<td>6937.38</td>
<td>0.58</td>
<td>4.10</td>
</tr>
<tr>
<td>1fhd</td>
<td>824</td>
<td>2827.03</td>
<td>7946.00</td>
<td>0.61</td>
<td>4.59</td>
</tr>
<tr>
<td>1bor</td>
<td>832</td>
<td>2820.13</td>
<td>7281.63</td>
<td>0.64</td>
<td>4.80</td>
</tr>
<tr>
<td>1hpt</td>
<td>858</td>
<td>3178.68</td>
<td>7826.13</td>
<td>0.61</td>
<td>4.76</td>
</tr>
<tr>
<td>1bpi</td>
<td>898</td>
<td>3078.71</td>
<td>8257.75</td>
<td>0.69</td>
<td>5.40</td>
</tr>
<tr>
<td>1mbg</td>
<td>903</td>
<td>2976.07</td>
<td>8028.13</td>
<td>0.58</td>
<td>4.87</td>
</tr>
<tr>
<td>1r69</td>
<td>997</td>
<td>2958.37</td>
<td>8977.25</td>
<td>0.64</td>
<td>5.59</td>
</tr>
<tr>
<td>1neq</td>
<td>1187</td>
<td>4525.79</td>
<td>10969.38</td>
<td>1.42</td>
<td>12.00</td>
</tr>
<tr>
<td>451c</td>
<td>1216</td>
<td>3898.07</td>
<td>11420.50</td>
<td>0.73</td>
<td>7.80</td>
</tr>
<tr>
<td>1a2s</td>
<td>1272</td>
<td>4102.00</td>
<td>12127.75</td>
<td>0.92</td>
<td>10.06</td>
</tr>
<tr>
<td>1svr</td>
<td>1435</td>
<td>4369.94</td>
<td>12381.50</td>
<td>1.06</td>
<td>12.99</td>
</tr>
<tr>
<td>1frd</td>
<td>1478</td>
<td>4235.85</td>
<td>13751.88</td>
<td>0.95</td>
<td>11.96</td>
</tr>
<tr>
<td>1r63</td>
<td>2065</td>
<td>6675.93</td>
<td>18555.13</td>
<td>1.81</td>
<td>29.81</td>
</tr>
<tr>
<td>1a7m</td>
<td>2809</td>
<td>7288.65</td>
<td>24743.38</td>
<td>2.17</td>
<td>47.65</td>
</tr>
<tr>
<td>1beb</td>
<td>4972</td>
<td>11725.46</td>
<td>46820.63</td>
<td>4.11</td>
<td>149.19</td>
</tr>
<tr>
<td>1vng</td>
<td>8808</td>
<td>17642.75</td>
<td>83772.75</td>
<td>5.88</td>
<td>360.22</td>
</tr>
<tr>
<td>1tas</td>
<td>12636</td>
<td>23795.40</td>
<td>11963.75</td>
<td>7.66</td>
<td>492.71</td>
</tr>
<tr>
<td>1maa</td>
<td>33257</td>
<td>59988.03</td>
<td>314893.75</td>
<td>46.53</td>
<td>7139.90</td>
</tr>
<tr>
<td>4bfl</td>
<td>46160</td>
<td>63546.52</td>
<td>453094.5</td>
<td>31.81</td>
<td>6685.39</td>
</tr>
</tbody>
</table>
operations in (29). In Figure 13, the CPU time $t_S$ for the VdW potential case is plotted against the number of atoms $N_a$. A least squares linear fitting is also conducted to form a linear representation $t_S = 0.000915N_a - 0.153788$. The slope of this linear function is very close to that of the previous case, which demonstrates the efficiency of the proposed ADI3 algorithm.

4.3. Solvation free energy
At last, we consider the application of the proposed ADI algorithm to the electrostatic analysis. For this purpose, we first generate the characteristic function $S$ by solving the geometric flow equation (5) subject to the VdW potential $V$ defined in (45). The solvent–solute dielectric profile can then be defined as $\epsilon(S) = (1 - S)\epsilon_s + S\epsilon_m$, where $\epsilon_s$ and $\epsilon_m$ are the dielectric constants for the
solvent and molecule, respectively. The electrostatic potential $\phi$ of proteins can be attained via the numerical solution of the Poisson–Boltzmann equation (4). In the present study, a pseudo-transient continuation approach based a second order fully implicit ADI scheme [32,33] is employed to solve the nonlinear Poisson–Boltzmann equation.

With the steady state values of the hypersurface function $S$ and the electrostatic potential $\phi$, the solvation free energy is calculated as follows. It is known that the total free energy functional of solvation does not directly provide the total solvation free energy. Actually, one needs to calculate the difference of the macromolecular system in the vacuum and in the solvent. The solvation free energy is calculated as follows. It is known that the total free energy functional of the nonlinear Poisson–Boltzmann equation.

$$\Delta G = G_{np} + (G_p - G_0),$$

(50)

where $G_{np}$ and $G_p$ are, respectively, the nonpolar and polar solvation free energies of the solute–solvent system with different $\epsilon_s$ and $\epsilon_m$, while $G_0$ is the polar free energy calculated from the homogeneous (vacuum) environment with $\epsilon_s = \epsilon_m = 1$ [31]. The term $G_p - G_0$ can be regarded as the electrostatic solvation free energy. The nonpolar solvation free energy of the macromolecule, $G_{np}$, is computed exactly according to its definition [31]. In the present study, the polar part is evaluated as

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

(51)

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

$$\Delta G_p = G_p - G_0 = \frac{1}{2} \sum_{i=1}^{N_m} Q(\mathbf{r}_i) (\phi(\mathbf{r}_i) - \phi_0(\mathbf{r}_i)),$$

(52)

where $\phi$ and $\phi_0$ are electrostatic potentials in the presence of the solvent and the vacuum, respectively.

The parameters of nonlinear solvation model are chosen according to the literature works [8, 9, 31, 33]. In particular, the ionic strength is chosen as $I_s = 0.1$ throughout. The atomic radii of each atom in all cases are enlarged according to [8], that is, the radii from the CHARMM force field need to be multiplied by a common factor of value 1.1. The dielectric constants are taken as $\epsilon_m = 1$ and $\epsilon_s = 80$. The surface tension $\gamma$ is treated as a fitting parameter, with its initial value being $\gamma = 1/15$ to scale other parameters. We set the bulk density coefficient to be $\rho_0/\gamma = 2$ and choose the pressure coefficient as $p/\gamma = 0.2$. The final values of $\gamma$ are different for various real systems [8, 9, 31, 33]. We note that all parameter values used in the present study are identical to those in the previous pseudo-time solvation models [31, 33].

We first calculate the solvation free energies of a set of 17 small compounds. This test set was originally studied by Nicholls et al. [45] by using various approaches and was subsequently examined by using many different solvation models [8, 9, 31, 33]. Nontrivial biological features are involved in this set of compounds, such as the existence of polyfunctional or interacting polar groups, which lead to strong solute–solvent interactions [45]. Generally speaking, due to these features, this set is considered as a challenging test in biomolecular simulations [8, 9]. On the other hand, this is an excellent benchmark set, because experimental measurements are available. However, it should be emphasized that these experimental results are not the limiting values to which our ADI numerical results should converge. Instead, the experiment data shall be treated as a benchmark for assessing the modeling errors of the pseudo-time nonlinear solvation model [31, 33].

The structure and charge information of the 17 compounds are adopted from those of Nicholls et al. [45]. The OpenEye-AM1-BCC v1 parameters are used for the charges, while atomic coordinates and radii are based on a new parameterization introduced by Nicholls et al., that is, the so-called ZAP-9 form. Basically, in the ZAP-9 parameterization, certain types of radii are adjusted from Bonds radii so that a better agreement with experimental data is achieved. Both implicit solvent and explicit solvent approaches have been examined in [45] for this set of compounds. For the more
that there is a general agreement between the present results and the literature ones. In particular, the
sharp solute–solvent interface [46]. The BVP-LPB results are generated by using one of the original
accurate, while more expensive, explicit solvent model, the root mean square error (RMS) is found
to be 1.71 ± 0.05 kcal/mol. The best implicit solvent model gives a RMS 1.87 ± 0.03 kcal/mol.

In present study, a dense mesh with \( h = 0.25 \AA \) is employed to achieve a better spatial resolution
for these small chemical compounds. The ADI3 time parameters are set to be \( \Delta t = 0.1 \) and \( T = 10 \).
For the fitting parameter \( \gamma \), an initial value of \( \gamma = 1/15 \) is used, while its final value is \( \gamma = 0.0080 \).
The calculated solvation free energies are listed in Table X. For a comparison, literature results
generated by the pseudo-time coupled nonlinear Poisson–Boltzmann (PTC-NPB) model [33], are also
included. Both the PTC-NPB model of [33] and the present one are implicit solvent models. We
note that there are mainly two modeling differences between them. (i) The scaled geometric flow
equation (7) is integrated by the forward Euler scheme in the PTC-NPB, while the unscaled equa-
tion (5) is solved by the ADI scheme in the present work; (ii) The PTC-NPB model is a coupled
model, in which the hypersurface function \( S \) also depends on the electrostatic potential \( \phi \), while
the present model is a uncoupled one, in which \( S \) is calculated without using \( \phi \). Besides these two
aspects, the same nonlinear Poisson–Boltzmann (NPB) solver is utilized in both [33] and the present
study, and other procedures and parameters are also the same. It is interesting to see in Table X that a
much smaller modeling error is involved in the present study than that for the PTC-NPB model. The
RMS error of the present model is as low as 1.7280 kcal/mol, which is almost as good as that of the
explicit solvent model [45]. The exact cause behind this improvement of implicit solvent modeling
deserves further research studies, however, is beyond the scope of this work.

We finally calculate solvation free energies for the set of 23 proteins, which have been studied
in the previous subsections. To save the CPU time, a larger spacing \( h = 0.5 \AA \) is employed. The
ADI3 time parameters are set to be \( \Delta t = 0.1 \) and \( T = 10 \). The final value of the fitting param-
eter \( \gamma \) is chosen as \( \gamma = 1/20 \). The solvation free energies of these 23 proteins estimated by the
present ADI approach are listed in Table XI. For a comparison, literature results of the PTC-NPB
model and two other linearized Poisson–Boltzmann (LPB) models, that is, MIBPB and BVP-LPB,
are also reported. The MIBPB results are calculated by using a second-order interface treatment,
the matched interface and boundary (MIB) method, for solving the classical LPB equation with a
sharp solute–solvent interface [46]. The BVP-LPB results are generated by using one of the original
differential geometry-based solvation models [9], in which, a boundary value problem (BVP) of the
LPB equation is solved in each step of an iterative procedure. It can be observed from Table XI
that there is a general agreement between the present results and the literature ones. In particular,

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

<table>
<thead>
<tr>
<th>Compound</th>
<th>Exptl</th>
<th>( \Delta G )</th>
<th>Error</th>
<th>( \Delta G )</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glycerol triacetate</td>
<td>–8.84</td>
<td>–10.22</td>
<td>–1.38</td>
<td>–10.23</td>
<td>–1.39</td>
</tr>
<tr>
<td>Benzyl bromide</td>
<td>–2.38</td>
<td>–3.53</td>
<td>–1.15</td>
<td>–3.48</td>
<td>–1.08</td>
</tr>
<tr>
<td>Benzyl chloride</td>
<td>–1.93</td>
<td>–3.73</td>
<td>–1.80</td>
<td>–3.63</td>
<td>–1.70</td>
</tr>
<tr>
<td>m-bis(trifluoromethyl)benzene</td>
<td>1.07</td>
<td>–1.90</td>
<td>–2.97</td>
<td>–0.56</td>
<td>–1.63</td>
</tr>
<tr>
<td>N,N-dimethyl-p-methoxybenzamide</td>
<td>–11.01</td>
<td>–7.30</td>
<td>3.71</td>
<td>–7.25</td>
<td>3.76</td>
</tr>
<tr>
<td>N,N-4-trimethylbenzamide</td>
<td>–9.76</td>
<td>–5.89</td>
<td>3.87</td>
<td>–5.81</td>
<td>3.95</td>
</tr>
<tr>
<td>bis-2-chloroethyl ether</td>
<td>–4.23</td>
<td>–2.71</td>
<td>1.52</td>
<td>–2.77</td>
<td>1.46</td>
</tr>
<tr>
<td>1,1-diacetoxyethane</td>
<td>–4.97</td>
<td>–6.58</td>
<td>1.61</td>
<td>–6.60</td>
<td>1.63</td>
</tr>
<tr>
<td>1,1-diethoxyethane</td>
<td>–3.28</td>
<td>–2.90</td>
<td>0.38</td>
<td>–2.90</td>
<td>0.38</td>
</tr>
<tr>
<td>1,4-dioxane</td>
<td>–5.05</td>
<td>–6.42</td>
<td>0.43</td>
<td>–4.62</td>
<td>0.43</td>
</tr>
<tr>
<td>Diethyl propanedioate</td>
<td>–6.00</td>
<td>–6.04</td>
<td>0.04</td>
<td>–6.10</td>
<td>0.10</td>
</tr>
<tr>
<td>Dimethoxyethane</td>
<td>–2.93</td>
<td>–3.46</td>
<td>0.53</td>
<td>–3.51</td>
<td>0.58</td>
</tr>
<tr>
<td>Ethylene glycol diacetate</td>
<td>–6.34</td>
<td>–6.87</td>
<td>0.53</td>
<td>–6.88</td>
<td>0.54</td>
</tr>
<tr>
<td>1,2-diethoxyethane</td>
<td>–3.54</td>
<td>–2.72</td>
<td>0.82</td>
<td>–2.74</td>
<td>0.81</td>
</tr>
<tr>
<td>Diethyl sulfide</td>
<td>–1.43</td>
<td>–1.19</td>
<td>0.24</td>
<td>–1.13</td>
<td>0.30</td>
</tr>
<tr>
<td>Phenyl formate</td>
<td>–4.08</td>
<td>–6.52</td>
<td>2.44</td>
<td>–6.44</td>
<td>2.36</td>
</tr>
<tr>
<td>Imidazole</td>
<td>–9.81</td>
<td>–10.42</td>
<td>0.61</td>
<td>–10.42</td>
<td>0.61</td>
</tr>
</tbody>
</table>

RMS 1.8293 1.7280
nonlinear Poisson–Boltzmann models could be significant [32, 33]. We note that the difference between linear and nonlinear Poisson–Boltzmann equation. We consider the potential $u$ generated by the PTC-NPB model and present approach. Isosurfaces at $S(r) = 0.98$ are first constructed for both models. The corresponding potential $u$ is then projected onto these isosurfaces (Figure 14). Except for some minor visual differences, these two surface potentials are very similar, because both models are based on the same nonlinear Poisson–Boltzmann equation. We note that the difference between linear and nonlinear Poisson–Boltzmann models could be significant [32, 33].
5. CONCLUSION

This paper overcomes the numerical difficulties in solving the potential driven geometric flow PDEs for molecular surface generation. Two molecular surface PDE models exist in the literature and differ by a scaling term. The functionality and necessity of this scaling term are discussed. Based on the scaled PDE model, two existing ADI schemes involve nonlinear cross derivatives which have to be evaluated explicitly at the previous time instant. Consequently, the existing ADI schemes could be unstable in biomolecular simulations and have a low temporal order. A novel ADI method is introduced in this paper based on the divergence form of the unscaled geometric PDE so that cross derivatives are bypassed. In this compact form, the geometric flow equation is treated as one nonhomogeneous diffusion process, which can be discretized by the central finite differences to produce tridiagonal matrices. Due to the compactness, the new ADI scheme yields a higher temporal order and is unconditionally stable.

The proposed ADI algorithm is first validated by solving geometric flow PDEs with analytical solutions and reference solutions. The new ADI scheme is always stable and more accurate than the existing ADI schemes. In the molecular surface generation, quantitative studies based on surface area and surface-enclosed volume are carried out for several atomic systems. Two molecular surface PDE models are found to yield the same steady state, while the isosurfaces of three ADI methods are consistent. The application of the proposed ADI scheme to real biomolecular simulations is studied next. The MMS generated by the new ADI method without the potential term is demonstrated for several interesting macromolecules. Due to the use of a large time increment, the proposed ADI algorithm is very efficient in generating steady state solutions for molecular surface construction. A linear relationship between the execution time and the number of atoms is observed. The impact of different potential terms on the molecular surface is then illustrated. Finally, calculation of solvation free energies for small compounds and large protein systems is considered. The present ADI results are in good agreement with the literature.

The development of an efficient nonlinear solvation simulator based on the present ADI molecular surface construction and a fast nonlinear Poisson–Boltzmann solver is currently under our consideration. Stability analysis of the entire solvation system and adaptive steady state solution will be explored in a future work.

ACKNOWLEDGEMENT

This work was supported in part by NSF grants DMS-1016579 and DMS-1318898. Zhao thanks the Beijing Computational Science Research Center (CSRC) for its hospitality and support during his sabbatical in Beijing, where this manuscript is finalized.

REFERENCES

4. Spolar RS, Record Jr. MT. Coupling of local folding to site-specific binding of proteins to DNA. Science 1994; 263:777–784.