

## Geometric and potential driving formation and evolution of biomolecular surfaces

P. W. Bates · Zhan Chen · Yuhui Sun ·  
Guo-Wei Wei · Shan Zhao

Received: 20 March 2008 / Revised: 4 September 2008 / Published online: 22 October 2008  
© Springer-Verlag 2008

**Abstract** This paper presents new geometrical flow equations for the theoretical modeling of biomolecular surfaces in the context of multiscale implicit solvent models. To account for the local variations near the biomolecular surfaces due to interactions between solvent molecules, and between solvent and solute molecules, we propose potential driven geometric flows, which balance the intrinsic geometric forces that would occur for a surface separating two homogeneous materials with the potential forces induced by the atomic interactions. Stochastic geometric flows are introduced to account for the random fluctuation and dissipation in density and pressure near the solvent–solute interface. Physical properties, such as free energy minimization (area decreasing) and incompressibility (volume preserving), are realized by some of our geometric flow equations. The proposed approach for geometric and potential forces driving the formation and evolution of biological surfaces is illustrated by extensive

---

P. W. Bates · Z. Chen · Y. Sun · G.-W. Wei (✉)  
Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA  
e-mail: wei@math.msu.edu

P. W. Bates  
e-mail: bates@math.msu.edu

Z. Chen  
e-mail: chenzha1@msu.edu

Y. Sun  
e-mail: sunyuhui@msu.edu

G.-W. Wei  
Department of Electrical and Computer Engineering, Michigan State University,  
East Lansing, MI 48824, USA

S. Zhao  
Department of Mathematics, University of Alabama, Tuscaloosa, AL 35487, USA  
e-mail: szhao@bama.ua.edu

numerical experiments and compared with established minimal molecular surfaces and molecular surfaces. Local modification of biomolecular surfaces is demonstrated with potential driven geometric flows. High order geometric flows are also considered and tested in the present work for surface generation. Biomolecular surfaces generated by these approaches are typically free of geometric singularities. As the speed of surface generation is crucial to implicit solvent model based molecular dynamics, four numerical algorithms, a semi-implicit scheme, a Crank–Nicolson scheme, and two alternating direction implicit (ADI) schemes, are constructed and tested. Being either stable or conditionally stable but admitting a large critical time step size, these schemes overcome the stability constraint of the earlier forward Euler scheme. Aided with the Thomas algorithm, one of the ADI schemes is found to be very efficient as it balances the speed and accuracy.

**Keywords** Biomolecular surface formation and evolution · Mean curvature flow · Potential driven geometric flows · High order geometric flows · Stochastic geometric flows · Computational algorithm

**Mathematics Subject Classification (2000)** 92E10 · 53A10 · 49Q10 · 65M06

## 1 Introduction

Rigorous, quantitative, and atomic scale description of complex biological systems is a challenge for both biological science and mathematical modeling. Under physiological conditions, most biological processes occur in water, which constitutes of 65–90% of human cell weight. However, explicit descriptions of biomolecules and their aqueous environment, including solvent, co-solutes, and mobile ions, are prohibitively expensive. Even so, a variety of methods, including Ewald summations, Euler summations, periodic images and reaction field theory, have been developed in the past few decades in attempts to produce accurate simulations. In many situations, our central concern is the solute, i.e., proteins, DNAs and RNAs, and their interactions, instead of water molecules. Therefore, multiscale analysis, which treats the solvent as a macroscopic continuum while admitting a microscopic atomic description for the biomolecule, is an attractive and sometimes indispensable approach. Implicit solvent models [2, 24] are efficient multiscale approaches to simulate complex, large-scale biological systems [19, 29, 38, 57, 67, 75, 79, 90]. These approaches have become very popular since the pioneering work by Warwicker and Watson in the early 1980s [79], and Honig in the 1990s [38]. One of the implicit solvent models employs the Poisson–Boltzmann (PB) equation, or Poisson equation (PE) if no salt is present for the electrostatic potential [2]. Another implicit solvent model is based on the generalized Born (GB) theory [25].

In both models, a solvent–molecule interface is required to separate the continuum domain from the biomolecule, which is described in atomistic detail. Currently, commonly used solvent–solute interface models include the van der Waals surface, the solvent accessible surface [42], and the molecular surface (MS) [56]. A partial differential equation (PDE) approach was proposed for the MS generation [82]. However, the MS definition suffers from geometric singularities, such as cusps and self-intersecting surfaces [18, 22, 32, 59], which are not only unphysical, but also

destabilize numerical simulations. Smoothly varying functions were proposed to avoid this problem [34,40,45,91]. However, it has been shown that some atomic centered dielectric functions may lead to unphysically high dielectric interstitial regions in implicit solvent models [73].

When a less polar macromolecule is immersed in a polar environment, surface free energy minimization occurs naturally to stabilize the system. Since the surface free energy is proportional to the surface area, the minimization of the surface free energy must lead to surface area minimization. Mathematically, such a minimization can be achieved with the Euler–Lagrange variation and has been extensively studied in the context of differential geometry, image processing, and computer-aided surface design. In particular, the surface area minimization naturally leads to the minimization of mean curvature as demonstrated by Chan and coworkers, and others [7,9,44,51,52,58,60,61,68,78,92]. Mumford and Shah proposed an energy variation functional to balance the fidelity and smoothness, and minimize the discontinuous domain [49]. The motion of surfaces by mean curvature was first studied by Brakke [5]. Geometric flows [84], particularly mean curvature flows, have been of interest in applied mathematics over many years with an emphasis on image processing and computer vision [26,33,48,54,62,63,66,71]. Mathematical techniques using the level set theory were devised by Osher and Sethian [37,50,53,58,66,92] and have been developed and applied by many others, e.g., [10,14,70]. Wei proposed a series of arbitrarily high order gradient controlled PDEs for image analysis [80], including the fourth order anisotropic diffusion equation and the Perona–Malik equation [55] as special cases. A coupled PDE system was also proposed to extract image edges by Wei and Jia [81]. Geometric flows have also been used for image surface evolution [17,39,72,85]. High order geometric flows such as the surface diffusion flow [64,65], the generalized Perona–Malik equation [80], variation-based PDE [12], curvature diminishing diffusion equation [88], and Willmore flow [17,20,84,87] are also proposed for image processing and/or surface analysis. A diffusion approach was used to model lipid membrane [77]. Motivated by the difficulty of surface analysis associated with implicit solvent models, we have recently proposed a new concept, the minimal molecular surface (MMS), for modeling the solvent–biomolecule interface [3,4]. The intrinsic curvature force is used to drive the surface formation and evolution. The resulting MMS minimizes the surface free energy. The MMSs were employed in implicit solvent models to evaluate electrostatic potentials and solvation free energies of 26 proteins [4] via the PB method.

Another challenge in current implicit solvent models is their inability to address ion correlation, polar–nonpolar coupling and solvent–solute interaction [1,8,11,15,21,27,28]. The formation and evolution of our MMS are driven purely by intrinsic geometric forces and thus neglect potential interactions near the interface. Recently, Dzubiella et al. [21] proposed an interesting free energy minimization procedure for coupling the polar–nonpolar interaction at the solvent–solute interface. They arrive at a PDE involving contributions from pressure, Gauss and mean curvatures, short-range repulsion, dispersion and electrostatic effects. More recently, Cheng et al. [13] has employed the level set approach to minimize the free energy functional proposed in [21] for complex molecules. Surfaces with various contributions are produced from this approach. To address current difficulties in implicit solvent models, the PDE-based

biomolecular surface models are promising, because the potential forces induced by the interactions can be accounted for by adding driving terms into the purely geometric PDE models. Moreover, the resulting PDE approach is easier to implement than the direct minimization procedure.

The objective of the present work is threefold. First, we propose a class of new bio-surfaces whose formation and evolution are driven not only by the intrinsic geometric forces, but also by forces induced by potential [80,83]. Via the potential effect, these surfaces are expected to account for the polar–nonpolar coupling and solvent–solute interaction in implicit solvent models. We discuss how physical properties, such as area decreasing and volume preserving, are realized in our PDE approach. We note that area decreasing geometric flows were studied by Lawson [41], and Gage and Hamilton [30] among others. In normal cases, decreasing surface area leads to surface energy minimization. However, the introduction of interaction potentials in the geometric PDEs balances the geometric force to some extent and so the minimum energy configuration is not necessarily that with least surface area. Volume preserving geometric flows were discussed by Huiskens [39] and others. In molecular dynamics, volume preservation is required for the canonical ensemble and grand canonical ensemble, and is equivalent to incompressibility. Under physiological conditions, most biological systems are incompressible.

Second, we explore the use of high order geometric PDEs for biomolecular surface modeling. The advantages of these high order geometric PDEs are that one may specify more boundary conditions and thus exercise better control of the biomolecular surface generation. It is also possible to ensure the balance between intrinsic geometric forces and potential forces in the surface generation via high order geometric PDEs. Some of the proposed high order geometric PDEs are modified versions of the earlier high order gradient controlled PDEs [80]. Numerical experiments are carried out to illustrate the unique feature and usefulness of these proposed high order geometric equations for biological surface formation and evolution.

Finally, the generation of the biomolecular surfaces which separate the biomolecule and the solvent is known to be a bottleneck for long-time molecular dynamics simulations of implicit solvent models. Typically, biomolecular surfaces are to be mapped out up to billions of times in the course of such a simulation. Therefore, any nontrivial improvement in computational efficiency implies a significant reduction in computational time in molecular dynamics simulations. One major goal of this paper is to construct efficient numerical schemes for the generation of the MMS [4] for large proteins using the mean curvature flow. The standard finite difference and the forward Euler time stepping are commonly utilized to solve these types of equations [4, 10, 14]. However, explicit algorithms have strict restriction on the time step for stability. Various implicit methods are explored in this paper and results from this study may be useful for the surface generation using other PDEs.

This paper is organized as follows. Theoretical modeling of biomolecular surfaces is presented in Sect. 2, where geometric and potential forces driving biomolecular surface flows are introduced. The properties of area decreasing and volume preserving evolutions are discussed. Stochastic geometric flows are proposed to appropriately account for some physical interaction and for the realization of the canonical and grand canonical ensembles. High order gradient flows are also introduced for surface

modeling. Section 3 is devoted to the applications of some of the proposal theoretical approaches. Computer experiments are designed to illustrate geometric and potential driven flows and high order biomolecular surface flows. Comparison is made with the MSs and our previous MMSs. Numerical algorithms for the fast generation of MMSs are proposed and illustrated in Sect. 4. Two different alternative direction implicit (ADI) schemes, a Crank–Nicolson scheme and a semi-implicit scheme are constructed and examined in our studies.

## 2 Theoretical formulation

### 2.1 Hypersurface and curvatures

Consider a  $C^2$  immersion  $\mathbf{f} : U \rightarrow \mathbb{R}^{n+1}$ , where  $U \subset \mathbb{R}^n$  is an open set. Here  $\mathbf{f}(\mathbf{u}) = (\mathbf{f}_1(\mathbf{u}), \mathbf{f}_2(\mathbf{u}), \dots, \mathbf{f}_{n+1}(\mathbf{u}))$  is a position vector labeled by  $\mathbf{p} = \mathbf{f}(\mathbf{u})$  for a point on a hypersurface, and  $\mathbf{u} = (u_1, u_2, \dots, u_n) \in U$ .

Tangent vectors (or directional vectors) are  $X_i = \frac{\partial \mathbf{f}}{\partial u_i} \in T_{\mathbf{p}}\mathbb{R}^{n+1}$ . The Jacobi matrix of the mapping  $\mathbf{f}$  is given by  $D\mathbf{f} = (X_1, X_2, \dots, X_n)$ .

Denote  $\langle, \rangle$  as the Euclidean inner product in  $\mathbb{R}^{n+1}$ ,  $i, j = 1, 2, \dots, n$ . The first fundamental form  $I$  of a surface element is  $I(X_i, X_j) := \langle X_i, X_j \rangle$  for every two tangent vectors  $X_i, X_j \in T_{\mathbf{u}}\mathbf{f}$ , the tangent hyperplane at  $\mathbf{f}(\mathbf{u})$ . In the coordinate  $\mathbf{f}(\mathbf{u}) = (\mathbf{f}_1(\mathbf{u}), \mathbf{f}_2(\mathbf{u}), \dots, \mathbf{f}_{n+1}(\mathbf{u}))$ , the first fundamental form is a symmetric, positive definite matrix  $(g_{ij}) = (I(X_i, X_j))$ .

Let  $\mathbf{N}(\mathbf{u})$  be the unit normal vector given by the Gauss map  $\mathbf{N} : U \rightarrow S^n \subset \mathbb{R}^{n+1}$ ,

$$\mathbf{N}(u_1, u_2, \dots, u_n) := \pm(X_1 \times X_2 \cdots \times X_n) / \|X_1 \times X_2 \cdots \times X_n\| \in \perp_{\mathbf{u}}\mathbf{f}, \quad (1)$$

where  $\times$  is the cross product in  $\mathbb{R}^{n+1}$  and  $\perp_{\mathbf{u}}\mathbf{f}$  is the normal space of  $\mathbf{f}$  at point  $\mathbf{p} = \mathbf{f}(\mathbf{u})$ . Note that  $T_{\mathbf{u}}\mathbf{f} \oplus \perp_{\mathbf{u}}\mathbf{f} = T_{\mathbf{f}(\mathbf{u})}\mathbb{R}^{n+1}$ , the tangent space at  $\mathbf{p}$ . By means of the normal vector  $\mathbf{N}$  and tangent vectors  $X_i$ , the second fundamental form is given by

$$II(X_i, X_j) = (h_{ij})_{i,j=1,\dots,n} = \left\langle \left\langle -\frac{\partial \mathbf{N}}{\partial u_i}, X_j \right\rangle \right\rangle_{ij}. \quad (2)$$

The mean curvature can be calculated from

$$H = h_{ij}g^{ji}, \quad (3)$$

where we use the Einstein summation convention, and  $(g^{ij}) = (g_{ij})^{-1}$ .

The shape operator  $\mathcal{L}$  of  $\mathbf{f}$  is the Weingarten map:  $\mathcal{L} := -DN \circ (D\mathbf{f})^{-1}$  defined pointwise by

$$\mathcal{L}_{\mathbf{u}} = -(DN|_{\mathbf{u}}) \circ (D\mathbf{f}|_{\mathbf{u}})^{-1} : T_{\mathbf{u}}\mathbf{f} \rightarrow T_{\mathbf{u}}\mathbf{f}, \quad (4)$$

where

$$DN|_{\mathbf{u}} : T_{\mathbf{u}}U \rightarrow T_{\mathbf{u}}\mathbf{f}, \quad (5)$$

and the map

$$D\mathbf{f}|_{\mathbf{u}} : T_{\mathbf{u}}U \rightarrow T_{\mathbf{u}}\mathbf{f} \tag{6}$$

is a linear isomorphism whose inverse mapping  $(D\mathbf{f}|_{\mathbf{u}})^{-1}$  is well defined. Note that  $\mathcal{L}$  is self-adjoint in the basis  $\frac{\partial \mathbf{f}}{\partial u_i}$  and  $\mathcal{L} \frac{\partial \mathbf{f}}{\partial u_i} = -\frac{\partial \mathbf{N}}{\partial u_i}$ . With this notation, it is easy to make the connection between the first and second fundamental forms

$$\begin{aligned} I(\mathcal{L}X_i, X_j) &= \left( \left\langle -\frac{\partial \mathbf{N}}{\partial u_i}, X_j \right\rangle \right)_{ij} = \left( \left\langle \mathbf{N}, \frac{\partial^2 \mathbf{f}}{\partial u_i \partial u_j} \right\rangle \right)_{ij} \\ &= II(X_i, X_j) = (h_{ij})_{i,j=1,\dots,n}, \end{aligned} \tag{7}$$

where  $II$  is a symmetric bilinear form on  $T_{\mathbf{u}}\mathbf{f}$  for every  $\mathbf{u} \in U$ . Similarly, the third and fourth fundamental forms are conveniently given in terms of the shape operator

$$III(X_i, X_j) = I(\mathcal{L}^2 X_i, X_j) = \left( \left\langle \frac{\partial \mathbf{N}}{\partial u_i}, \frac{\partial \mathbf{N}}{\partial u_j} \right\rangle \right)_{ij} \tag{8}$$

$$IV(X_i, X_j) = I(\mathcal{L}^3 X_i, X_j). \tag{9}$$

Principal curvatures are defined as the eigenvalues of  $\mathcal{L}$  with eigenvectors being unit tangent vectors. For  $n = 2$ , appropriate organization of the principal curvatures gives rise to the mean curvature  $H = \text{Tr}(\mathcal{L})$  and Gauss curvature  $K = \text{Det}(\mathcal{L})$

$$H = \kappa_1 + \kappa_2 \tag{10}$$

$$K = \kappa_1 \kappa_2. \tag{11}$$

The mean curvature defined in Eq. (10) differs from the conventional definition by a factor of 2. However, the present definition is convenient for the derivation of evolution equations. A point  $\mathbf{p} = \mathbf{f}(\mathbf{u})$  on the surface is called an elliptic, hyperbolic, parabolic, umbilic or level point if  $K(\mathbf{p}) > 0$ ,  $K(\mathbf{p}) < 0$ ,  $K(\mathbf{p}) = 0$ ,  $\kappa_1(\mathbf{p}) = \kappa_2(\mathbf{p})$ , and  $\kappa_1(\mathbf{p}) = \kappa_2(\mathbf{p}) = 0$ , respectively. For  $n = 3$ , we have

$$K_1 = \kappa_1 + \kappa_2 + \kappa_3 \tag{12}$$

$$K_2 = \kappa_1 \kappa_2 + \kappa_1 \kappa_3 + \kappa_2 \kappa_3 \tag{13}$$

$$K_3 = \kappa_1 \kappa_2 \kappa_3 \tag{14}$$

where  $K_1 = H = \text{Tr}(\mathcal{L})$  and  $K_3 = K = \text{Det}(\mathcal{L})$  are the mean curvature and the Gauss curvature, respectively.  $K_1$  and  $K_2$  differ from the standard definition by a factor of 3. The local property of the Gauss curvature can be used to classify the point as elliptic, hyperbolic, parabolic, etc. It follows from the Cayley–Hamilton theorem that in three dimensional case, the first four fundamental forms satisfy  $IV - HIII + K_2II - KI = 0$ .

In general, given an  $n$ -dimensional manifold  $\Xi$  embedded in  $\mathbb{R}^{n+1}$  and a vector field  $\mathbf{b}$  defined in  $\Xi$  the divergence of  $\mathbf{b}$  relative to  $\Xi$  is given by

$$\nabla_{\Xi} \cdot \mathbf{b} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u_i} \left( \sqrt{g} \mathbf{b}^i \right), \tag{15}$$

where  $g = \text{Det}(g_{ij})$  is the Gram determinant of  $\mathbf{f}$ , the surface being described locally by a mapping  $\mathbf{f} : U \subset \mathbb{R}^n$  to  $\mathbb{R}^{n+1}$ . The gradient of a scalar field  $b$  on the manifold is given by

$$(\nabla_{\Xi} b)^i = g^{ij} \frac{\partial}{\partial u_j} b. \tag{16}$$

The Laplace–Beltrami operator is given by

$$\Delta_{\Xi} b = \nabla_{\Xi} \cdot \nabla_{\Xi} b = \frac{1}{\sqrt{g}} \frac{\partial}{\partial u_i} \left( \sqrt{g} g^{ij} \frac{\partial}{\partial u_j} b \right). \tag{17}$$

Applying this to the position vector  $\mathbf{p}$  gives  $\Delta_{\Xi} \mathbf{p} = H\mathbf{N}$ , where  $\mathbf{N}$  is the normal vector at  $\mathbf{p}$ . The integral of a function  $b$  on the manifold is given by

$$\int_{\Xi} b d\sigma = \int_U b \sqrt{g} du_1 du_2 \cdots du_n.$$

### 2.2 Mean curvature flow for minimal molecular surfaces

The surface deformation or evolution of macromolecules can be postulated as a time-dependent process for a family of smooth surfaces, which are defined as an immersion of two-dimensional manifolds  $\Xi(t)$  in  $\mathbb{R}^3$ . Here  $\Xi(t)$  are parameterized by  $t$ ,  $\{\Xi(t) : t \geq 0, \Xi(0) = \Xi_0\}$ , where  $\Xi_0$  is the initial surface determined by the input configuration of the molecule with appropriate selection of atomic radii, such as expanded van der Waals radii. Assuming a homogeneous biomolecule in a homogeneous solvent, the equation of motion for a surface position vector  $\mathbf{p}(t) \in \Xi(t)$  in the separating surface will be in the normal direction and given by

$$\frac{\partial \mathbf{p}}{\partial t} = \Delta_{\Xi} \mathbf{p} = H\mathbf{N}. \tag{18}$$

This is the mean curvature flow and has been used for the generation of the minimal molecular surface (MMS) [4]. To generate the desired MMS, appropriate geometric constraints originating from molecular boundaries are required as explained in detail in [4]. In particular, the van der Waals surface of the molecule is protected during the evolution.

### 2.3 Generalized geometric flows for biomolecular surfaces

More general geometric flows can be cast in the form of

$$\frac{\partial \mathbf{p}}{\partial t} = V_g \mathbf{N}, \quad \Xi(0) = \Xi_0, \tag{19}$$

where  $V_g$  is the amplitude of the velocity  $\mathbf{V}$  induced by geometric forces. The surface diffusion flow is obtained by choosing  $V_g = -\Delta_{\Xi} H$  [23,64,65]

$$\frac{\partial \mathbf{p}}{\partial t} = -\Delta_{\Xi} H \mathbf{N}, \quad \Xi(0) = \Xi_0. \quad (20)$$

By selecting  $V_g = -\Delta_{\Xi} H - 2H(H^2 - K)$ , one has the Willmore flow [20,69,84,87]

$$\frac{\partial \mathbf{p}}{\partial t} = [-\Delta_{\Xi} H - 2H(H^2 - K)] \mathbf{N}, \quad \Xi(0) = \Xi_0, \quad (21)$$

where  $K$  is the Gaussian curvature. The Willmore flow is designed to minimize the Willmore energy

$$E_W = \int_{\Xi(t)} (H^2 - K) d\sigma = \frac{1}{4} \int_{\Xi(t)} (\kappa_1 - \kappa_2)^2 d\sigma,$$

where  $\kappa_1$  and  $\kappa_2$  are two principle curvatures. As a result, the Willmore flow minimizes the difference of two principle curvatures. Generalized surface diffusion flows ( $V_g = (-1)^k \Delta_{\Xi}^k H$ ) can also be considered [85]

$$\frac{\partial \mathbf{p}}{\partial t} = (-1)^k \Delta_{\Xi}^k H \mathbf{N}, \quad \Xi(0) = \Xi_0. \quad (22)$$

These generalized geometric flows can also be used for our biomolecular surface modeling. A good reason for using high order equations is that one can specify more boundary conditions; for the second order equations one can only specify a single condition (such as Dirichlet or Neumann boundary conditions, or their combinations). With more boundary conditions one can reasonably hope to preserve desirable features in the solvent density function. Apart from possible isolated singularities appearing at discrete rare instances in time, these flows produce smooth surfaces. The smoothness in surfaces is useful for biological applications, such as in implicit solvent models.

#### 2.4 Potential driving geometric flows for biomolecular surfaces

It is noted that the association of “surface area decreasing” and “surface free energy minimizing” during the time evolution are equivalent under the assumption of homogeneous solvent–solute interfaces, i.e., the energy density is a constant over  $\Xi$ . However, this will generally not be the case due to the polar–nonpolar coupling and surface response to solvent density variations and ion binding. Therefore, the biomolecular surface motion should be driven not only by intrinsic geometric forces, but also by forces induced by interaction potentials. To this end, we propose the following potential driving geometric flows (PDGF)

$$\frac{\partial \mathbf{p}}{\partial t} = (V_g + V_p) \mathbf{N}, \quad \Xi(0) = \Xi_0 \quad (23)$$



where  $V_p$  are the forces from equilibrium pressure contribution of the solvent, contributions due to solvent–solvent and solvent–solute interactions, and van der Waals interactions of solute atoms near the interface. In the spirit of implicit solvent theory, the microscopic detail of individual solvent molecules will not be included. Their impact on the surface formation should be expressed in terms of macroscopic variables, such as density, mean velocity, pressure and temperature, and transport coefficients such as dielectric constant, viscosity, etc. Therefore, local force can be expressed as the product of the particle density function and the gradient of the particle interaction potential. However, van der Waals interactions of solute atoms can be treated explicitly.

### 2.5 Incompressible geometric flows for biomolecular surfaces

In biological processes, such as protein folding, it is very important to keep track of certain physical quantities, such as area and volume. At the steady state, the surface area of a biological molecule or complex might not be necessarily the minimum because it is the total free energy that is minimized during the time evolution and even the surface free energy is not necessarily minimized by minimizing surface area, as mentioned above. However, incompressibility is an important feature of most biological systems at normal temperature. Let  $A(t)$  and  $V(t)$  denote, respectively, the area of  $\Xi(t)$  and the volume of the region enclosed by  $\Xi(t)$ . Then the rate of change of the area and the volume during the deformation of  $\Xi(t)$  can be computed respectively as

$$\frac{dA(t)}{dt} = - \int_{\Xi(t)} (V_g + V_p) H d\sigma \quad \text{and} \quad \frac{dV(t)}{dt} = \int_{\Xi(t)} (V_g + V_p) d\sigma, \quad (24)$$

where  $d\sigma$  is an area element. It is easy to show that the mean curvature flow, Eq. (19), is area decreasing ( $\frac{dA(t)}{dt} = - \int_{\Xi(t)} H^2 d\sigma \leq 0$ ) and not volume preserving. In the context of biomolecular surface modeling, area decreasing is associated with the surface free energy minimization, while volume preserving is associated with the incompressibility during a biological process. We therefore propose incompressible geometric flows, such as

$$\frac{\partial \mathbf{p}}{\partial t} = (V_g + V_p - \bar{V}) \mathbf{N}, \quad \Xi(0) = \Xi_0 \quad (25)$$

where  $\bar{V} = \int_{\Xi(t)} (V_g + V_p) d\sigma / \int_{\Xi(t)} d\sigma$ . Note that this equation conserves the volume during the surface evolution since  $\frac{dV(t)}{dt} = \int_{\Xi(t)} (V_g + V_p - \bar{V}) d\sigma = 0$ .

It is also easy to show that the surface diffusion flow Eq. (20) is area decreasing and volume preserving, and thus can be used as a higher order equation for the surface formation. The higher order geometric flows given by Eq. (22) are volume preserving for  $k \geq 1$

$$\int_{\Xi(t)} \Delta_{\Xi}^k H d\sigma = \int_{\Xi(t)} \Delta_{\Xi} (\Delta_{\Xi}^{k-1} H) d\sigma = \int_{\Xi(t)} \nabla_{\Xi}(1) \cdot \nabla_{\Xi} (\Delta_{\Xi}^{k-1} H) d\sigma = 0.$$

## 2.6 Stochastic surface dynamics

In the implicit solvent models, the solvent is considered as a dielectric continuum so that each individual solvent molecule is not explicitly described. The solvent–solute electrostatic interaction is accounted for by the Poisson–Boltzmann equation or the generalized Born approximation. However, many other important solvent–solute interactions, such as the polar–nonpolar interaction, the solvent collision or scattering on the biomolecular surface, solvent density fluctuation, energy dissipation on the biomolecular surface, are not sufficiently addressed. We therefore introduce a stochastic velocity to account for some of these effects

$$\frac{\partial \mathbf{p}}{\partial t} = (V_g + V_p + V_r - \bar{V})\mathbf{N}, \quad \Xi(0) = \Xi_0, \quad (26)$$

where  $V_r$  is the amplitude of the random velocity  $\mathbf{V}_r$  in the surface normal direction. We take the random velocity to have mean value zero and to be independently distributed in time, i.e.,  $\langle \mathbf{V}_r \rangle_e = 0$  and  $\langle \mathbf{V}_r(t)\mathbf{V}_r^T(t') \rangle_e = \eta\delta(t - t')$ , where  $\langle \cdot \rangle_e$  denotes average over an appropriate ensemble distribution,  $T$  the transpose and  $\eta$  the strength of the random velocity. Here the average of  $\bar{V}$  includes the effect of the three other terms.

Technically, the introduction of stochastic processes enables the realization of canonical and grand canonical ensembles, in which the volume and temperature are kept as constant. Temperature is associated with the strength of the random term. The realization of other ensembles, such as the isothermal-isobaric ensemble is under investigation. The mathematical analysis of the proposed stochastic geometric flows is involved and interesting, but not the focus of the present discussion.

## 3 Applications of generalized geometric flows to biological surfaces

### 3.1 Eulerian and Lagrangian formalisms

The formation of biomolecular surfaces can be pursued via either the Lagrangian formalism or the Eulerian formalism. In the Lagrangian formalism, surface elements are evolved directly under various driving forces. In the Eulerian formalism, the surface is embedded in a (implicit) hypersurface and the latter is evolved under prescribed driving forces. The desired surface is obtained from an isosurface extraction procedure. The Lagrangian formalism is straightforward for force prescription and computationally efficient, but may have difficulties in handling geometric singularities. The Eulerian formalism can easily handle the geometric singularities, but is more time consuming. Moreover, it is not clear how to implement the incompressibility in the present Eulerian formalism.

The Eulerian formalism can be obtained via an appropriate choice of the immersion  $\mathbf{f}$ . Previously, we have chosen  $\mathbf{f} = (x, y, z, S)$  which maps  $U \subset \mathbb{R}^3$  to  $\mathbb{R}^4$  [4]. It is easy to show that

$$\Delta_{\Xi} S = \langle \mathbf{N}, \mathbf{S} \rangle H = \frac{H}{\sqrt{g}} = \frac{1}{\sqrt{g}} \nabla \cdot \left( \frac{\nabla S}{\sqrt{g}} \right) \quad (27)$$

where  $\mathbf{N} = \frac{(-S_x, -S_y, -S_z, 1)}{\sqrt{g}}$  is the normal vector in  $\mathbb{R}^4$ ,  $\mathbf{S} = (0, 0, 0, 1)$ , and  $\Xi$  is the graph of  $S$ . Note that  $\mathbf{N}$  is the normal vector for the surface immersion  $\mathbf{f}$ , rather than for the hypersurface function  $S$ . For a given surface, it is easy to show that its normal vector does not depend on whether its description is based on an implicit function or an explicit one. In our earlier work, the mean curvature flow for the MMS [4] was constructed as

$$\frac{\partial S}{\partial t} = \frac{\Delta_{\Xi} S}{\langle \mathbf{N}, \mathbf{S} \rangle^2} = \sqrt{g} \nabla \cdot \left( \frac{\nabla S}{\sqrt{g}} \right). \tag{28}$$

The computational procedure for the surface formation and evolution of molecules as dictated by (28) was described in detail in [4].

### 3.2 Potential driving geometric flows

#### 3.2.1 Formulation

In the Eulerian formulation, the potential driving geometric flow (PDGF) can be given as

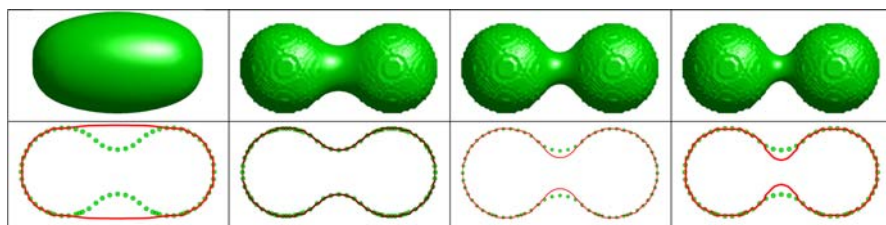
$$\frac{\partial S}{\partial t} = \sqrt{g} \nabla \cdot \left( \frac{\nabla S}{\sqrt{g}} \right) + P, \tag{29}$$

where  $g = 1 + |\nabla S|^2$  and  $P(\mathbf{r}, S, |\nabla S|)$  is a driving term due to the potential, whose physical origin has been discussed in an earlier section. A possible choice of the potential term is to set  $P = \sqrt{g} H_p(x, y, z)$ , where  $H_p(x, y, z)$  is an interaction potential term. Obviously, Eq. (29) includes our previous MMS formula [4] as a special case with  $P = 0$ . We note that for a general choice of  $P$ , the area-decreasing and volume-preserving properties are usually not satisfied.

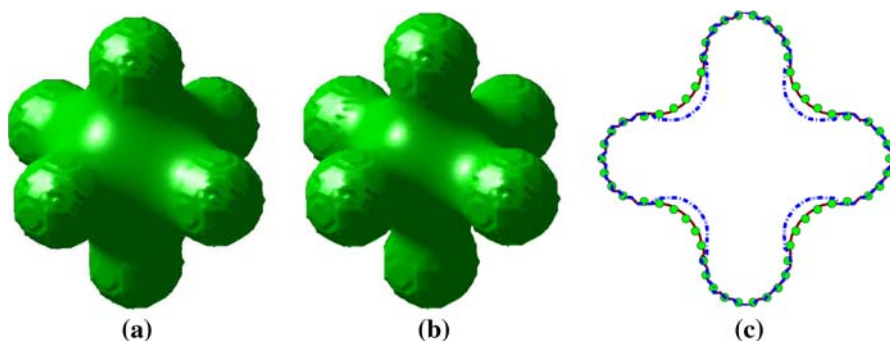
By using Eq. (29), the proposed potential driving biomolecular surface flow can be constructed in a procedure similar to what we used for the generation of the MMS [4]. The surface evolves according to Eq. (29) until  $\sqrt{g} \nabla \cdot \left( \frac{\nabla S}{\sqrt{g}} \right) + P = 0$  everywhere, except for the protected van der Waals sphere surfaces. Finally we extract the desirable biomolecular surface from the hypersurface function  $S$  by choosing a level surface  $S = C$ .

#### 3.2.2 Numerical experiments

We first consider cases where  $H_p$  is constant over the whole domain. Figure 1 illustrates the impact of  $H_p$ , taken as a series of constants, to the geometry of a diatomic system. By setting the van der Waals (VDW) radius  $r = 2.0 \text{ \AA}$ , the PDGF results are compared with the molecular surface (MS) generated with a probe radius  $r_p = 1.4 \text{ \AA}$ . It can be seen that at an appropriate  $H_p$  value, we can reproduce the MS of the diatomic molecule. In other words, by using such an  $H_p$  value, some effect of probe rolling can be well captured in the PDGF model, at least for diatomic systems. This motivates us



**Fig. 1** The PDGF surfaces of a diatomic molecule plotted at  $H_p = -1.200, -0.666, 0.000$  and  $+0.300$ , from left to right. The corresponding comparison with the molecular surface (circles) with the probe radius  $r_p = 1.4 \text{ \AA}$  is given at the cross section  $z = 0$  in the second row. The coordinates of the diatomic molecule are  $(x, y, z) = (-2.4, 0, 0)$  and  $(2.4, 0, 0)$ . The van der Waals radius is set to  $2.0 \text{ \AA}$

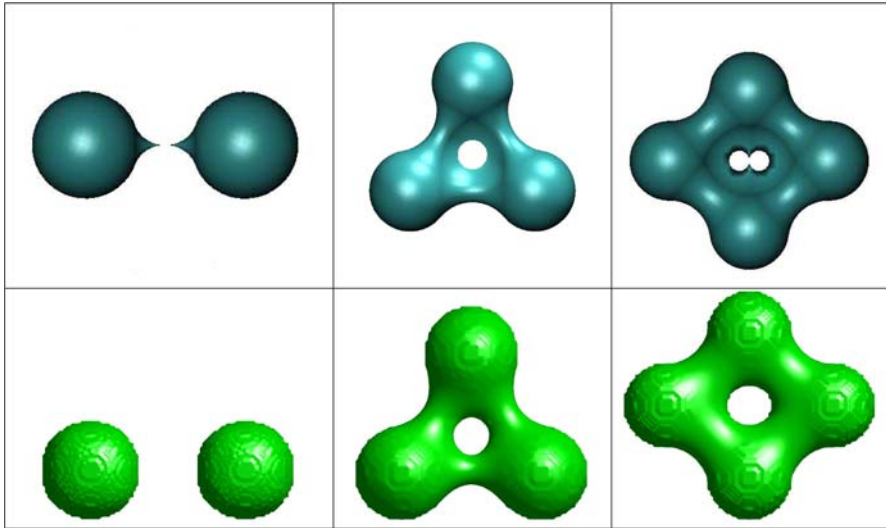


**Fig. 2** The PDGF surfaces of a seven-atom system with VDW radius  $r_v = 1.0 \text{ \AA}$ . The coordinates are generated with a pair of atoms with center distance  $4.8 \text{ \AA}$  and by  $\frac{\pi}{2}$  rotations. **a**  $H_p = -0.666$ ; **b**  $H_p = 0$ ; **c** the comparison of cross section  $z = 0$  (circles molecular surface with  $r_p = 1.4 \text{ \AA}$ ; solid line  $H_p = -0.666$ ; dashed line  $H_p = 0$ )

to choose the free parameter  $P$  of the PDGF model to account for the probe radius of the MS. For simple geometries, the proposed PDGF surface with an appropriate  $H_p$  value gives a good approximation to the MS. This is further demonstrated by a seven-atom system in Fig. 2.

It is also of great interest to verify that the proposed PDGF surface is free of geometric singularities. To this end, we consider geometric parameters where singularities occur in the molecular surface (MS). Figure 3 depicts three molecular systems which admit cusps and/or self-intersecting surfaces. Parameter  $\tau$  is the probe radius that enforces the cavity constraint [4]. The molecular surface of the diatomic system has a pair of cusps when the atomic distance is in a certain interval. The molecular surface of a three-atom system has a sharp self-intersecting surface. The molecular surface of a four-atom system admits both cusps and self-intersecting surfaces. Unlike the MS, surfaces generated with the proposed PDGF surfaces are very smooth at these geometric parameters. The MMSs looked significantly different from the traditional MS surface while the current PDGF surfaces appear much more similar to MS surface. Different PDGF surfaces can be generated by choosing different potential terms.

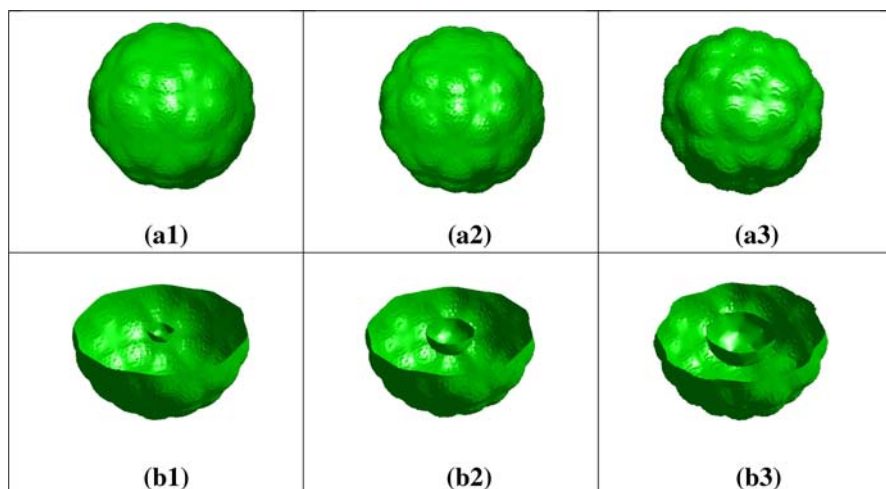
An important issue in biomolecular surface modeling is the ability to capture any open and internal cavities. Following the approach of the MMS generation [4], we



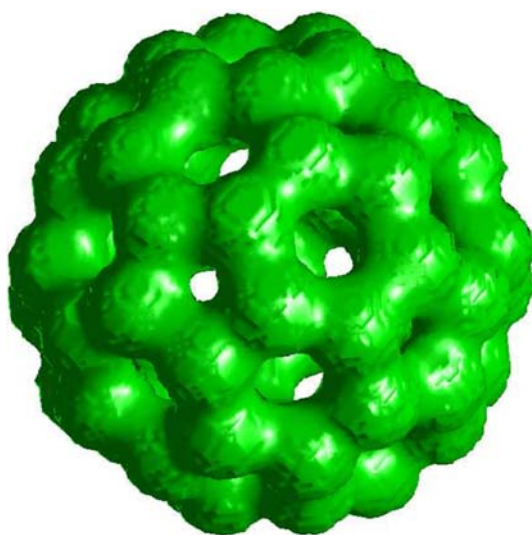
**Fig. 3** *Top row* Molecular surfaces of two-, three- and four-atom systems; *bottom row* corresponding PDGF surfaces. For the diatomic system,  $(x, y, z) = (-3.15, 0, 0)$  and  $(3.15, 0, 0)$ ; The van der Waals radius is set to  $r_v = 2 \text{ \AA}$ . For the three-atom system,  $(x, y, z) = (-2.3, 0, 0)$ ,  $(2.3, 0, 0)$ , and  $(0, 3.984, 0)$ ;  $r_v = 1.5 \text{ \AA}$ ,  $r_p = 1.5 \text{ \AA}$ , and  $\tau = 0.9 \text{ \AA}$ . For the four-atom system,  $(x, y, z) = (-3, 0, 0)$ ,  $(0, -2.6, 0)$ ,  $(3, 0, 0)$ , and  $(0, 2.6, 0)$ ;  $r_v = 1.5 \text{ \AA}$ ,  $r_p = 1.5 \text{ \AA}$ , and  $\tau = 0.9 \text{ \AA}$

set  $S(x, y, z, t) = 0$  on the domain outside a solvent accessible surface during the computation. Here the solvent accessible surface is attained by rolling a probe radius  $\tau$  on the van der Waals surface of the molecule. We consider a buckyball molecule (buckminsterfullerene  $C_{60}$ ). By an appropriate choice of the solvent excluding radius  $\tau$ , a cavity is produced for the buckyball as shown in Fig. 4. Moreover, by choosing relatively smaller  $r_v$  and  $\tau$ , a topologically interesting PDGF surface for the buckyball molecule is created in Fig. 5.

We have demonstrated that the surface generated by the PDGF is a singularity-free approximation to the MS, and is capable of capturing cavities. Thus, we provide an alternative way to generate smooth surfaces for complex proteins. We finally consider a quantitative way to study the difference between the PDGF surface and MS by means of an electrostatic analysis. By defining the PDGF surface as the solvent–solute dielectric interface, the electrostatic potentials of proteins can be attained via the numerical solution of the Poisson–Boltzmann equation. Twenty three proteins, a test set used in previous studies [25, 89], are employed to validate our approach. For all structures, extra water molecules that are attached to proteins are excluded and hydrogen atoms are added to obtain full all-atom models. Partial charges at atomic sites and atomic van der Waals radii in angstroms are taken from the CHARMM22 force field [47]. By setting the PDGF surface and MS as the dielectric boundaries, electrostatic free energies of solvation,  $\Delta G$ , are computed by using the MIBPB-II solver [89] at mesh size  $h = 0.5 \text{ \AA}$ . For a comparison, results from the fourth order geometric equation (FOGE), which is described in detail in the next section, are also presented. In all test cases, the dielectric coefficient  $\epsilon$  is set to 1 and 80 respectively



**Fig. 4** The PDGF surfaces of the buckyball. *a1* van der Waals radius  $r_v = 2.0 \text{ \AA}$  and  $\tau = 0.5 \text{ \AA}$ ; *a2*  $r_v = 1.7 \text{ \AA}$  and  $\tau = 0.5 \text{ \AA}$ ; *a3*  $r_v = 1.4 \text{ \AA}$  and  $\tau = 0.5 \text{ \AA}$ . In *b1*–*b3*, cross section views of the corresponding (a) parts are shown with half of the data of  $S$  removed



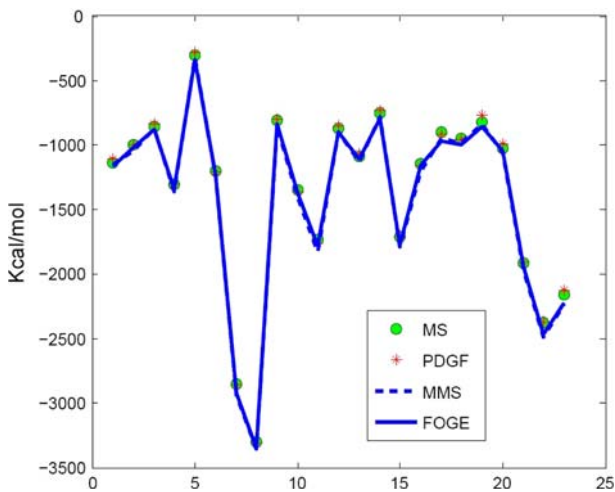
**Fig. 5** The PDGF surface of the buckyball with scaled atomic radius  $r_v = 0.8 \text{ \AA}$  and  $\tau = 0.25 \text{ \AA}$

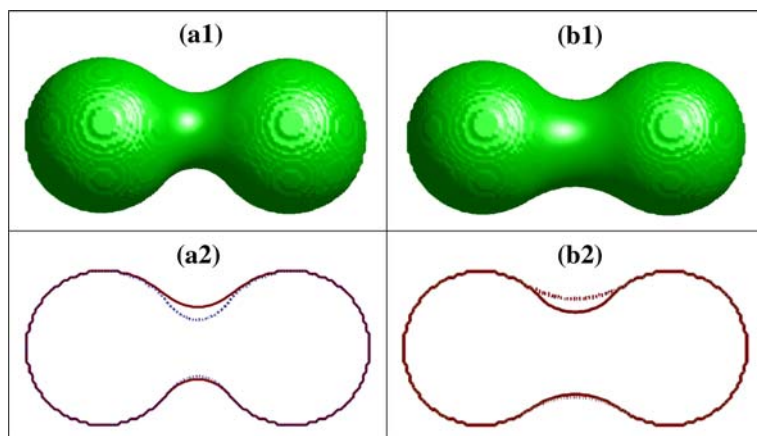
for the protein and solvent. A probe radius of  $r_p = 1.4 \text{ \AA}$  was used for the MS. The PDGF surface is generated with the probe radius of  $\tau = 0.85 \text{ \AA}$  to enforce the cavity constraint. The numerical results of electrostatic free energies of solvation are listed in Table 1. Good consistency can be observed among results of the PDGF surface, the FOGES surface and MS. In fact, this consistency can be observed clearly from Fig. 6. For reference, we also list the solvation free energies associated with MMSs in the table. It can be seen that differences in solvation free energies between the PDGF surface and MS are generally less than those between the MMS and MS. Similar energetic results

**Table 1** Electrostatic free energies of solvation calculated by using the MIBPB

PDB ID	No. of atoms	MS	MMS	PDGF	FOGE
1ajj	519	-1136.9	-1150.4	-1106.4	-1168.8
1bbl	576	-994.2	-1043.5	-999.9	-1017.4
1bor	832	-854.1	-875.0	-836.0	-875.5
1bpi	898	-1304.0	-1358.7	-1301.2	-1341.6
1cbn	648	-305.4	-315.9	-285.0	-340.6
1fca	729	-1201.3	-1245.5	-1205.3	-1233.9
1frd	1478	-2852.2	-2944.8	-2859.2	-2919.3
1fxd	824	-3299.8	-3356.0	-3311.0	-3351.3
1hpt	858	-811.6	-857.5	-799.0	-834.6
1mbg	903	-1350.9	-1407.9	-1362.7	-1376.1
1neq	1187	-1730.7	-1833.0	-1757.9	-1784.8
1ptq	795	-871.8	-902.45	-852.9	-899.6
1r69	997	-1088.9	-1131.6	-1067.5	-1111.7
1sh1	702	-754.6	-779.7	-728.8	-792.6
1svr	1435	-1713.5	-1786.8	-1703.2	-1773.2
luxc	809	-1141.6	-1206.8	-1156.0	-1158.5
1vii	596	-902.5	-949.5	-915.8	-966.8
2erl	573	-950.2	-986.1	-952.1	-1000.1
2pde	667	-819.5	-833.3	-767.7	-854.7
451c	1216	-1025.0	-1061.9	-992.3	-1044.4
1a2s	1272	-1912.8	-1973.6	-1906.6	-1934.1
1a63	2065	-2375.5	-2486.4	-2367.8	-2469.3
1a7m	2809	-2158.0	-2233.6	-2126.9	-2223.1

$$H_p = -0.6666$$

**Fig. 6** Comparison of electrostatic free energies of solvation of 23 proteins listed in the same order as those of Table 1



**Fig. 7** The PDGF surfaces of a diatomic molecule. *a1*  $H_p = -0.427 \left( \frac{4}{\sqrt{x^2 + (y-5.5)^2 + z^2}} \right)^7$ ; *b1*  $H_p = -0.999 + 0.488 \left( \frac{4}{\sqrt{x^2 + (y-5.5)^2 + z^2}} \right)^7$ ; *a2* A comparison of  $z = 0$  cross sections of *a1* (solid line) and a case with  $H_p = 0$  (dash line); *b2* A comparison of  $z = 0$  cross sections of *b1* (solid line) and a case with  $H_p = -0.999$  (dash line). The coordinates of the diatomic molecule are  $(x, y, z) = (-2.4, 0, 0)$  and  $(2.4, 0, 0)$ . The center of the potential is at  $(0, 5.5, 0)$ . The van der Waals radius is set to  $r_v = 2.0 \text{ \AA}$  for both atoms

suggest that these PDGF surfaces may not require extensive reparameterization of biomolecular radii used in implicit solvent modeling [76]. The biological significance of these differences is to be investigated.

Finally, we illustrate the flexibility of the proposed method for the local modification of biomolecular surfaces. To demonstrate the principle, we consider only simple non-constant potentials. We expect that our method works similarly for more realistic potentials, such the Lennard–Jones potential and the classical dipolar potential [43, 74, 76].

To this end, we set

$$H_p(r) = H_{p0} + d \left( \frac{\sigma}{r} \right)^l \quad (30)$$

where  $H_{p0}$  is a constant,  $l$  an integer,  $d$  the depth of the potential well and  $\sigma$  the reference distance at which the potential becomes very weak. Parameter  $d$  can be either negative or positive representing an attractive potential or a repulsive potential, respectively. Parameter  $l$  controls the decay rate of the potential. In practice, the combination of both positive and negative terms with different  $l$  values can be used as in the Lennard–Jones potential. However, we consider only a single term here to illustrate our method. Figure 7 shows two cases, both having  $l = 7$  and  $\sigma = 4.0$ . In the first case, we set  $H_{p0} = 0$  and an attractive  $d = -0.427$ . A significant change in the necked part of the surface is produced as shown in Fig. 7(a1) and (a2). Since  $l$  is sufficiently large, the change in the surface is localized to one side of the necked part. In the second case, we choose  $H_{p0} = -0.999$  to enlarge the necked part of the



surface. We then set a repulsive  $d = 0.488$  to create a localized indent on one side of the necked part as shown in Fig. 7(b1) and (b2). More dramatic changes in the surface can be generated with appropriate selection of  $H_p$ . These numerical experiments indicate that the proposed PDGF is a robust approach for the generation of desirable locally modified biomolecular surfaces.

### 3.3 High order geometric flows

#### 3.3.1 Formulation

High order forced and gradient-controlled diffusion operators were introduced for image analysis [80]

$$\frac{\partial S(\mathbf{r}, t)}{\partial t} = \sum_{q=0}^m \nabla \cdot \left[ D_q(|\nabla S(\mathbf{r}, t)|) \nabla \nabla^{2q} S(\mathbf{r}, t) \right] + P(S(\mathbf{r}, t), |\nabla S(\mathbf{r}, t)|), \tag{31}$$

where  $S(\mathbf{r}, t)$  is an image function,  $D_q(|\nabla S|)$  are gradient sensitive diffusion coefficients.  $P(S, |\nabla S|)$  is a gradient sensitive forcing term. It can be a nonlinear function of  $S$  as in chemical kinetics. A fourth order version of Eq. (31) was used for noisy image restoration [80], and the treatment of medical magnetic resonance images [46]. Based on Eq. (31), combinations of forward and backward diffusion operators were also utilized for image deblurring [31, 72]. A fourth order variational-based PDE was proposed by Chan et al. [12] for noisy image restoration. You and Kaveh [88] introduced a fourth order curvature diminishing diffusion equation. An analysis of these fourth order equations has recently been carried out in Sobolev space  $H^1$  by Bertozzi and Greer [6, 35, 36]. Xu and Zhou discussed the existence and uniqueness of weak solutions of this class of equations [86].

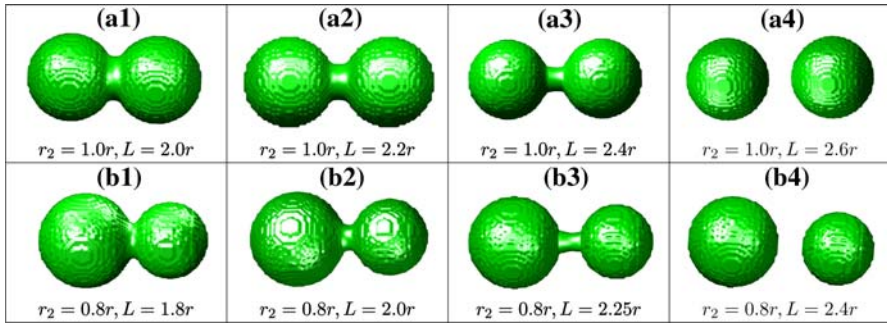
In the present work, we propose the following generalized high order geometric PDE for surface formation and evolution

$$\frac{\partial S}{\partial t} = (-1)^q \sqrt{g(|\nabla \nabla^{2q} S|)} \nabla \cdot \left( \frac{\nabla(\nabla^{2q} S)}{\sqrt{g(|\nabla \nabla^{2q} S|)}} \right) + P(S, |\nabla S|), \tag{32}$$

where  $g(|\nabla \nabla^{2q} S|) = 1 + |\nabla \nabla^{2q} S|^2$  is the generalized Gram determinant and  $P$  is a general production term representing possible forces, including the potential effect in biomolecular surface formation. When  $q = 0$  and  $P = 0$ , Eq. (32) reduces to the mean curvature flow equation, while when  $q = 1$  and  $P = 0$ , it is a surface diffusion flow. This equation can also be regarded as a variation of the high order anisotropic diffusion equation (31). Obviously, we could modify Eq. (32) by including both forward and backward diffusion operators.

#### 3.3.2 Numerical experiments

Since the effect of the forcing term has been discussed in earlier sections, we focus our attention on the effect of the fourth order geometric equation (FOGE) obtained by

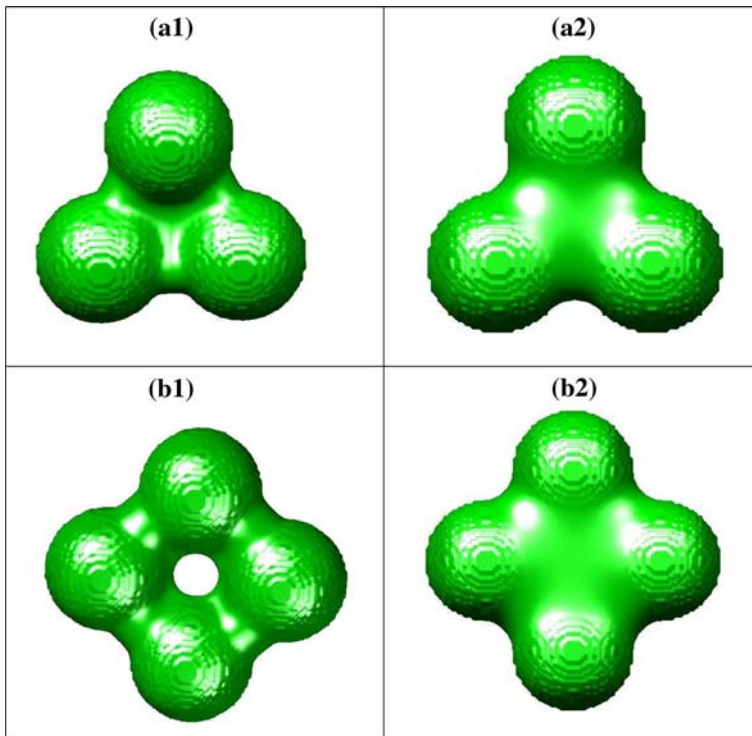


**Fig. 8** The FOGE surface of a diatomic molecule. In the *first* row, two atoms have the same radius while the atomic distance is changed. In the *second* row, both the radius of the second atom and the atomic distance are changed

setting  $P = 0$  and  $q = 1$  in Eq. (32). As in the MMS generation [4], standard finite difference schemes and forward Euler time integration are used to numerically evolve the surface toward its steady state configuration. Finally we extract the desired surface from the hypersurface function by choosing a level set of  $S$ . In the following studies, we are particularly interested in the biological surfaces generated with the FOGE, in comparison with the MMSs.

We first consider the generation of the surface of the FOGE for a diatomic molecule. By setting the radius of the main atom (left one) being  $r_1 = r$ , the radius  $r_2$  of the other atom (right one) and the distance  $L$  between the atomic centers being given in terms of  $r$ . It is known for the MMS generation [4] that there is a critical length  $L_c$  above which the surface enclosing two connected atoms becomes disconnected. The same is true for the FOGE surface, see Fig. 8. In comparison with the MMS, it can be observed from Fig. 8 that the catenoid type connecting tube in the FOGE surface is usually thinner than that for the MMS. Moreover, by considering different cases with varying length  $L$ , a linear dependence of the critical value  $L_c$  with respect to  $r_1$  and  $r_2$  can be identified as  $L_c \simeq 1.25(r_1 + r_2)$  for the surface of the FOGE, which is slightly different from that of the MMS, i.e.,  $L_c \simeq 1.213(r_1 + r_2)$ .

We next consider a comparison between the surface of the FOGE and MMS for the cases of three atoms and four atoms. Similar studies have been done in [4] to demonstrate that the MMS is free of singularities. The same conclusion can be drawn for the surface generated by the proposed FOGE. We focus on the difference between the surfaces of the FOGE and MMS, see Fig. 9. Here, both surfaces are generated without a cavity constraint. The MMS is known to have the minimal surface in terms of total area surrounding all van der Waals (VDW) atoms. In comparison with the MMS, it is clear from Fig. 9 that the FOGE surface actually has smaller surface area for the connecting tube than that of the MMS, whereas the total surface area of VDW patches of the FOGE surface is larger than that of the MMS so that the FOGE surface has a larger total surface area. Consequently, the FOGE surface is able to represent more atomic details than the MMS. Furthermore, it is interesting to see that the FOGE surface can generate a cavity where the MMS does not.

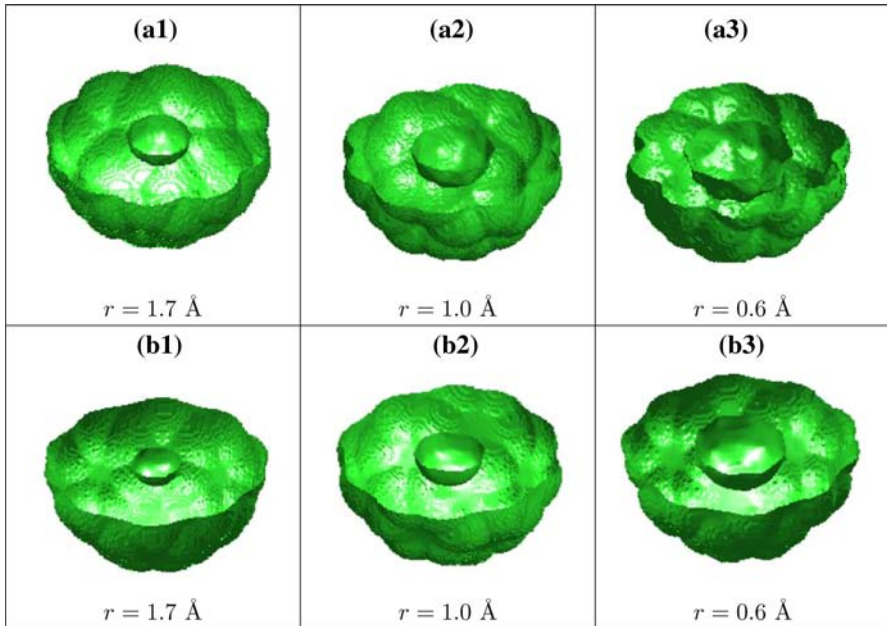


**Fig. 9** Comparison of 3- and 4-atom surfaces generated by the FOGE (*left column*) and MMS (*right column*) without constraint

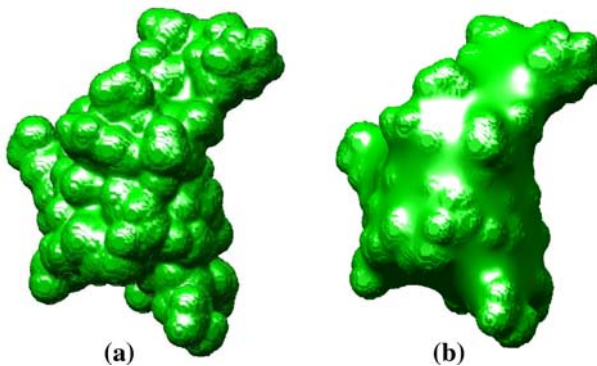
Inaccessible internal cavities and open cavities (pockets) are commonly encountered in macromolecules. The cavity constraints introduced in the MMS generation [4] can be similarly applied in the FOGE surface generation. As an example, we consider a buckyball molecule (buckminsterfullerene  $C_{60}$ ). We set  $S(x, y, z, t) = 0$  on the domain outside a solvent accessible surface during the computations of both the MMS and FOGE surfaces. Here the solvent accessible surface is attained by rolling a probe radius  $\tau = 1.5\text{\AA}$  on the van der Waals surface of the molecule. The results shown in Fig. 10 clearly indicate that the cavities of the FOGE surface are much larger than those of the MMS. This suggests that the FOGE surface is well suited for certain biological applications where large cavities are desirable.

We next test the FOGE surface for some complex biomolecules. The FOGE surface of a protein (PDB ID: 1ajj) is given in Fig. 11. It is interesting to see that atomic details are captured in the FOGE surface, but not so well in the MMS, while the FOGE surface still maintains the smoothness property of the MMS.

We finally consider the electrostatic free energies of solvation of 23 proteins computed with FOGE surfaces. The treatment of the protein data is the same as that described in the last section. Results are listed in Table 1 and plotted Fig. 6. It is seen that FOGE surfaces often provide smaller solvation values. This is consistent



**Fig. 10** A comparison between the cavities of  $C_{60}$  generated by the FOGE (the *first row*) and those generated by the mean curvature flow (the *second row*)



**Fig. 11** Biomolecular surfaces of protein 1ajj. **a** Obtained with the FOGE. **b** Obtained with the mean curvature flow

with the fact that FOGE surfaces are more skinny than MMSs as shown in Figs. 9 and 11.

#### 4 Computational algorithms for the mean curvature flow

The generation of surfaces is important and has been a bottleneck for molecular dynamics simulations in the implicit solvent models. In this section, we study efficient

computational algorithms for solving nonlinear diffusion equations. In particular, we focus on the mean curvature flow equation

$$\frac{\partial S}{\partial t} = \sqrt{g} \nabla \cdot \left( \frac{\nabla S}{\sqrt{g}} \right). \tag{33}$$

Standard finite differences with forward Euler time stepping are commonly utilized to solve this type of equations [10, 14]. In our earlier work [4] a simple discretization scheme was used. Specifically, the forward Euler method is employed for time integration, while the second order central difference scheme was used for the spatial discretization. Stability constraints provide the major obstacle that slows down the computation. In the present study, numerical algorithms based on a semi-implicit scheme, a Crank–Nicolson and two alternating-direction implicit (ADI) methods [16] are designed and tested. Comparison is made for the different approaches in terms of numerical accuracy and efficiency.

### 4.1 Numerical algorithms

#### 4.1.1 Explicit Euler scheme

In a detailed form, the three-dimensional (3D) mean curvature flow equation (33) can be rewritten as

$$\begin{aligned} \frac{\partial S}{\partial t} = & \frac{(1 + S_x^2 + S_y^2)S_{zz} + (1 + S_x^2 + S_z^2)S_{yy} + (1 + S_y^2 + S_z^2)S_{xx}}{1 + S_x^2 + S_y^2 + S_z^2} \\ & - 2 \frac{S_x S_y S_{xy} + S_x S_z S_{xz} + S_z S_y S_{yz}}{1 + S_x^2 + S_y^2 + S_z^2}. \end{aligned} \tag{34}$$

We adopt the following notations in the rest of discussions. Let the index  $i, j, k$  represent the location  $(x_i, y_j, z_k)$  and  $h$  be the uniform grid size. We consider a discrete time  $t_n := n\tau$  where  $n$  is a non negative integer and  $\tau$  is the time step size. We denote  $S_{ijk}^n$  to the discretized form of  $S(x_i, y_j, z_k, t_n)$ . An explicit scheme for the mean curvature flow can be given as

$$S_{ijk}^{n+1} - S_{ijk}^n = [v_x \delta_x^2 + v_y \delta_y^2 + v_z \delta_z^2] S_{ijk}^n + \tau f_{ijk}^n, \tag{35}$$

where

$$v_x = D_{1ijk}^n \tau, \quad v_y = D_{2ijk}^n \tau, \quad v_z = D_{3ijk}^n \tau, \tag{36}$$

$$f_{ijk}^n = \left\{ -2 \frac{+(\delta_x S)(\delta_y S)(\delta_{xy} S) + (\delta_x S)\delta_z S(\delta_{xz} S) + (\delta_y S)\delta_z S(\delta_{yz} S)}{1 + (\delta_x S)^2 + (\delta_y S)^2 + (\delta_z S)^2} \right\}_{ijk}^n, \tag{37}$$

$$D_{1ijk}^n = \left\{ \frac{1 + (\delta_y S)^2 + (\delta_z S)^2}{1 + (\delta_x S)^2 + (\delta_y S)^2 + (\delta_z S)^2} \right\}_{ijk}^n, \tag{38}$$

$$D_{2ijk}^n = \left\{ \frac{1 + (\delta_x S)^2 + (\delta_z S)^2}{1 + (\delta_x S)^2 + (\delta_y S)^2 + (\delta_z S)^2} \right\}_{ijk}^n, \tag{39}$$

$$D_{3ijk}^n = \left\{ \frac{1 + (\delta_x S)^2 + (\delta_y S)^2}{1 + (\delta_x S)^2 + (\delta_y S)^2 + (\delta_z S)^2} \right\}_{ijk}^n, \tag{40}$$

where

$$\{\delta_x S\}_{ijk}^n = \delta_x S_{ijk}^n = \left[ S_{(i+1)jk}^n - S_{(i-1)jk}^n \right] / 2h, \tag{41}$$

$$\{\delta_y S\}_{ijk}^n = \delta_y S_{ijk}^n = \left[ S_{i(j+1)k}^n - S_{i(j-1)k}^n \right] / 2h, \tag{42}$$

$$\{\delta_z S\}_{ijk}^n = \delta_z S_{ijk}^n = \left[ S_{ij(k+1)}^n - S_{ij(k-1)}^n \right] / 2h, \tag{43}$$

$$\begin{aligned} \{\delta_{xy} S\}_{ijk}^n &= \delta_{xy} S_{ijk}^n \\ &= \left[ S_{(i+1)(j+1)k}^n + S_{(i-1)(j-1)k}^n - S_{(i+1)(j-1)k}^n - S_{(i-1)(j+1)k}^n \right] / 4h^2, \end{aligned} \tag{44}$$

$$\begin{aligned} \{\delta_{xz} S\}_{ijk}^n &= \delta_{xz} S_{ijk}^n \\ &= \left[ S_{(i+1)j(k+1)}^n + S_{(i-1)j(k-1)}^n - S_{(i+1)j(k-1)}^n - S_{(i-1)j(k+1)}^n \right] / 4h^2, \end{aligned} \tag{45}$$

$$\begin{aligned} \{\delta_{yz} S\}_{ijk}^n &= \delta_{yz} S_{ijk}^n \\ &= \left[ S_{i(j+1)(k+1)}^n + S_{i(j-1)(k-1)}^n - S_{i(j+1)(k+1)}^n - S_{i(j-1)(k+1)}^n \right] / 4h^2 \end{aligned} \tag{46}$$

and

$$\delta_x^2 S_{ijk}^n = \left[ S_{(i-1)jk}^n - 2S_{ijk}^n + S_{(i+1)jk}^n \right] / h^2, \tag{47}$$

$$\delta_y^2 S_{ijk}^n = \left[ S_{i(j-1)k}^n - 2S_{ijk}^n + S_{i(j+1)k}^n \right] / h^2, \tag{48}$$

$$\delta_z^2 S_{ijk}^n = \left[ S_{ij(k-1)}^n - 2S_{ijk}^n + S_{ij(k+1)}^n \right] / h^2. \tag{49}$$

We can rewrite the explicit scheme (35) in matrix–vector notation as

$$S^{n+1} - S^n = A(S^n)S^n + F(S^n). \tag{50}$$

This scheme is explicit, since  $S^{n+1}$  can be calculated from  $S^n$  directly, and it has the advantage of not requiring much memory. However, it is not very efficient, because a very small step size is required to guarantee the stability of the algorithm. This motivates us to consider the following semi-implicit scheme in the present study.

### 4.1.2 Semi-implicit scheme

From Eq. (50), we consider a modified discretization, i.e.,

$$S^{n+1} - S^n = A(S^n)S^{n+1} + F(S^n), \tag{51}$$

which leads to a semi-implicit scheme [70]

$$(1 - A(S^n)) S^{n+1} = S^n + F(S^n). \tag{52}$$

It is implicit in the sense that this scheme does not give the solution  $S^{n+1}$  directly—one needs to solve a linear system. However, in order to avoid solving a nonlinear algebraic system at each time step, we compute nonlinear terms with the solution obtained in the previous time step, and thus, this is a semi-implicit scheme. The advantage of the implicit nature is that the computation could be *more* stable so that it does not suffer from such a serious time step size restriction. Consequently, the computation can be fully adapted to the desired accuracy without the need of choosing a small time step.

However, being able to use a large time increment does not necessarily imply that the semi-implicit scheme is superior, since it gives rise to a new problem. It requires one to solve a linear system which has a large sparse matrix. Applying a direct algorithm such as Gaussian elimination would lead to enormous memory storage and computation effort in 3D. Hence, iterative algorithms have to be applied. Classical iterative methods like Gauss–Seidel do not need additional storage and convergence can be guaranteed for some special  $A$ , but the convergence rate is rather slow. Biconjugate gradient method can also be used in our semi-implicit scheme. The Fast Fourier transform (FFT) was used in Ref. [70]. However, due to the protection of the van der Waals surfaces in our surface evolution, the FFT cannot be directly applied.

### 4.1.3 Crank–Nicolson scheme

In Eq. (34), the right hand side can be regarded as a multi-variable function  $F(S_x, S_y, S_z, S_{xx}, S_{yy}, S_{zz}, S_{xy}, S_{yz}, S_{xz})$ ,

$$\begin{aligned} \frac{\partial S}{\partial t} = F = & \frac{(1 + S_x^2 + S_y^2)S_{zz} + (1 + S_x^2 + S_z^2)S_{yy} + (1 + S_y^2 + S_z^2)S_{xx}}{1 + S_x^2 + S_y^2 + S_z^2} \\ & - \frac{2S_x S_y S_{xy} + 2S_x S_z S_{xz} + 2S_z S_y S_{yz}}{1 + S_x^2 + S_y^2 + S_z^2}. \end{aligned} \tag{53}$$

The Crank–Nicolson scheme is given as

$$S^{n+1} - S^n = \frac{\Delta t}{2} (F^{n+1} + F^n), \tag{54}$$

with  $F^{n+1}$  being linearized by a multi-variable Taylor expansion

$$\begin{aligned} F^{n+1} = F^n + & \frac{\partial F}{\partial S_x}(S_x^{n+1} - S_x^n) + \frac{\partial F}{\partial S_y}(S_y^{n+1} - S_y^n) + \frac{\partial F}{\partial S_z}(S_z^{n+1} - S_z^n) \\ & + \frac{\partial F}{\partial S_{xx}}(S_{xx}^{n+1} - S_{xx}^n) + \frac{\partial F}{\partial S_{yy}}(S_{yy}^{n+1} - S_{yy}^n) + \frac{\partial F}{\partial S_{zz}}(S_{zz}^{n+1} - S_{zz}^n) \\ & + \frac{\partial F}{\partial S_{xy}}(S_{xy}^{n+1} - S_{xy}^n) + \frac{\partial F}{\partial S_{yz}}(S_{yz}^{n+1} - S_{yz}^n) + \frac{\partial F}{\partial S_{xz}}(S_{xz}^{n+1} - S_{xz}^n) \end{aligned} \tag{55}$$

where derivatives of  $F$  should be evaluated at time instant  $t_n$ . The resulting scheme is given by

$$\begin{aligned}
 & S^{n+1} - \frac{\Delta t}{2} \left[ \frac{\partial F}{\partial S_x} S_x^{n+1} + \frac{\partial F}{\partial S_y} S_y^{n+1} + \frac{\partial F}{\partial S_z} S_z^{n+1} + \frac{\partial F}{\partial S_{xx}} S_{xx}^{n+1} + \frac{\partial F}{\partial S_{yy}} S_{yy}^{n+1} \right. \\
 & \quad \left. + \frac{\partial F}{\partial S_{zz}} S_{zz}^{n+1} + \frac{\partial F}{\partial S_{xy}} S_{xy}^{n+1} + \frac{\partial F}{\partial S_{yz}} S_{yz}^{n+1} + \frac{\partial F}{\partial S_{xz}} S_{xz}^{n+1} \right] \\
 & = S^n + \Delta t F^n - \frac{\Delta t}{2} \left[ \frac{\partial F}{\partial S_x} S_x^n + \frac{\partial F}{\partial S_y} S_y^n + \frac{\partial F}{\partial S_z} S_z^n + \frac{\partial F}{\partial S_{xx}} S_{xx}^n + \frac{\partial F}{\partial S_{yy}} S_{yy}^n \right. \\
 & \quad \left. + \frac{\partial F}{\partial S_{zz}} S_{zz}^n + \frac{\partial F}{\partial S_{xy}} S_{xy}^n + \frac{\partial F}{\partial S_{yz}} S_{yz}^n + \frac{\partial F}{\partial S_{xz}} S_{xz}^n \right], \tag{56}
 \end{aligned}$$

with appropriate second order central finite difference schemes for all spatial derivatives. It can be proved that with a smooth solution, the Crank–Nicolson scheme is second-order in time and second-order in space if appropriate central finite difference schemes are applied to the spatial discretization. Moreover it is stable. But convergence is slow in the generation of MMSs, which involves rapidly changing solutions. For rapidly changing solutions, the matrix in the linearized system is far from being diagonally dominant, which is the main cause of the slow convergence.

#### 4.1.4 Alternating direction implicit (ADI) schemes

In the following, to fully take advantages of a semi-implicit scheme, we adapt a splitting algorithm based on an alternating direction implicit (ADI) method, which is a variation of the Crank–Nicolson scheme. The traditional Douglas ADI method, widely used for linear diffusion equations, is always stable. In the present case, the additional advantage of the ADI-based scheme is that a fast Thomas algorithm can be applied to speed up the computation dramatically. In order to develop the ADI-based scheme, the nonlinear mean curvature equation has to be linearized first. Many different linearizations are possible. In the present work, we discuss two simple ADI schemes.

*ADI (a).* We write Eq. (34) as

$$\frac{\partial S}{\partial t} = S_{zz} + S_{yy} + S_{xx} - G(S). \tag{57}$$

where

$$G(S) = \frac{S_x^2 S_{xx} + S_y^2 S_{yy} + S_z^2 S_{zz} + 2S_x S_y S_{xy} + 2S_x S_z S_{xz} + 2S_z S_y S_{yz}}{1 + S_x^2 + S_y^2 + S_z^2}. \tag{58}$$

The nonlinear source term  $G$  is treated from the previous time step while the linear ones are considered on the current time step. This results in a linear problem at each discrete time step. Inspired by the Crank–Nicolson scheme and factorization by introducing



additional terms of order  $O(\tau^2)$ , we modify Eq. (57) to give

$$\begin{aligned} & \left(1 - \tau \frac{\delta_x^2}{2}\right) \left(1 - \tau \frac{\delta_y^2}{2}\right) \left(1 - \tau \frac{\delta_z^2}{2}\right) S_{ijk}^{n+1} \\ &= \left[ \left(1 + \tau \frac{\delta_x^2}{2}\right) \left(1 + \tau \frac{\delta_y^2}{2}\right) \left(1 + \tau \frac{\delta_z^2}{2}\right) - \tau^3 \frac{\delta_x^2 \delta_y^2 \delta_z^2}{4} \right] S_{ijk}^n - \tau G(S_{ijk}^n), \end{aligned} \tag{59}$$

where  $G(S_{ijk}^n)$  is the second order central difference discretized of  $G(S)$ . Now the following multi-step algorithm (3D Douglas-like method) can be adopted

Step 1

$$\left(1 - \tau \frac{\delta_x^2}{2}\right) S_{ijk}^{n+\frac{1}{3}} = \left(1 + \tau \frac{\delta_x^2}{2} + \tau \delta_y^2 + \tau \delta_z^2\right) S_{ijk}^n - \tau G(S_{ijk}^n); \tag{60}$$

Step 2

$$\left(1 - \tau \frac{\delta_y^2}{2}\right) S_{ijk}^{n+\frac{2}{3}} = S_{ijk}^{n+\frac{1}{3}} - \tau \frac{\delta_y^2}{2} S_{ijk}^n; \tag{61}$$

Step 3

$$\left(1 - \tau \frac{\delta_z^2}{2}\right) S_{ijk}^{n+1} = S_{ijk}^{n+\frac{2}{3}} - \tau \frac{\delta_z^2}{2} S_{ijk}^n. \tag{62}$$

The first step only introduces the diffusion operator in  $x$ -direction. Therefore, the resulting set of algebraic equation is tridiagonal. Similarly, the second and third steps involve the diffusion operators in  $y$ - and  $z$ -directions respectively. Therefore the efficient Thomas algorithm can be applied. Moreover, it is numerically verified by varying  $\tau$  that this ADI scheme, denoted as ADI (a), is stable. However, the accuracy is somewhat affected due to the introduction of some second derivative terms into the source term. In other words, a small time step (but still much greater than that of an explicit scheme) is required in order to achieve a desired accuracy. This motivates us to design another ADI scheme in order to use a larger time step for the computation.

ADI (b). In Eq. (35), nonlinear terms  $v_x, v_y, v_z$ , and  $f_{ijk}^n$  can be calculated from the previous time step while the linear terms are considered at the current time step. This also leads to a linear problem at each time step. We modify Eq. (35) to give

$$\left(1 - \frac{v_x}{2} \delta_x^2 - \frac{v_y}{2} \delta_y^2 - \frac{v_z}{2} \delta_z^2\right) S_{ijk}^{n+1} = \left(1 + \frac{v_x}{2} \delta_x^2 + \frac{v_y}{2} \delta_y^2 + \frac{v_z}{2} \delta_z^2\right) S_{ijk}^n + \tau f_{ijk}^n. \tag{63}$$

By introducing additional terms of order  $O(\tau^2)$ , this equation is further modified as

$$\begin{aligned} & \left(1 - \frac{A_x}{2}\right) \left(1 - \frac{A_y}{2}\right) \left(1 - \frac{A_z}{2}\right) S_{ijk}^{n+1} \\ &= \left[ \left(1 + \frac{A_x}{2}\right) \left(1 + \frac{A_y}{2}\right) \left(1 + \frac{A_z}{2}\right) - A_x A_y A_z / 4 \right] S_{ijk}^n + \tau f_{ijk}^n, \end{aligned} \quad (64)$$

where

$$A_x = v_x \delta_x^2, \quad A_y = v_y \delta_y^2, \quad A_z = v_z \delta_z^2. \quad (65)$$

Then the following multi-step implementation can be applied.

*Step 1*

$$\left(1 - \frac{A_x}{2}\right) S_{ijk}^{n+\frac{1}{3}} = \left(1 + \frac{A_x}{2} + A_y + A_z\right) S_{ijk}^n + \tau f_{ijk}^n; \quad (66)$$

*Step 2*

$$\left(1 - \frac{A_y}{2}\right) S_{ijk}^{n+\frac{2}{3}} = S_{ijk}^{n+\frac{1}{3}} - \frac{A_y}{2} S_{ijk}^n; \quad (67)$$

*Step 3*

$$\left(1 - \frac{A_z}{2}\right) S_{ijk}^{n+1} = S_{ijk}^{n+\frac{2}{3}} - \frac{A_z}{2} S_{ijk}^n. \quad (68)$$

We denote this scheme as ADI (b).

The ADI (b) scheme obviously possesses almost the same features as ADI (a), namely, the efficient Thomas algorithm can also be used here and the scheme is stable for a modified mean curvature flow with a small solution discussed in Sect. 4.2.2. However, it is not stable in our MMS computations because of rapid changes in the solution. Nevertheless, at a given accuracy, the ADI (b) scheme is able to admit a larger time increment than the ADI (a) scheme does.

Moreover, experimental examples demonstrate that the ADI (b) scheme is the most efficient scheme under typical accuracy requirement. It is about four times more efficient than our previous explicit scheme.

## 4.2 Numerical procedure and benchmark test

In this subsection, we carry out numerical experiments to test the performance of the proposed schemes in terms of accuracy and efficiency. The minimal molecular surfaces are generated in these experiments.

### 4.2.1 Computational procedure for MMS generation

First of all, we briefly introduce the procedure to generate MMSs of protein molecules. For a given protein, let the atomic centers be  $\mathbf{r}_i = (x_i, y_i, z_i), i = 1, \dots, n$ , and  $r_i$  represents the radius of the  $i$ th atom. Here  $n$  denotes the total number of the atoms in the protein molecule. We take an enlarged domain to be  $D = \bigcup_{i=1}^n \{\mathbf{r} : |\mathbf{r} - \mathbf{r}_i| < 1.3r_i\}$ . We choose factor 1.3 to guarantee the formation of properly connected MMSs. This is based on our observations for the diatomic system reported above and in [4], which shows that the critical separation distance for the centers of two connected atoms with radii  $r_1$  and  $r_2$  is  $1.213(r_1 + r_2)$ . It was demonstrated in [4] that appropriate connectivity of MMSs is guaranteed if the initial domain  $D$  is chosen with a factor larger than 1.213. For the initial value of  $S$ , we consider an indicator function, namely

$$S(x, y, z, 0) = \begin{cases} S_0, & (x, y, z) \in D \\ 0, & \text{otherwise.} \end{cases} \tag{69}$$

To obtain the minimal molecular surface of biomolecules, we need to protect the van der Waals surface during the evolution. To this end, we consider a simple characteristic function

$$\chi(\mathbf{r}) = \begin{cases} 0, & \forall \mathbf{r} \in D_\chi \\ 1, & \text{otherwise} \end{cases}, \tag{70}$$

where  $D_\chi = \bigcup_{i=1}^n \{\mathbf{r} : |\mathbf{r} - \mathbf{r}_i| < r_i\}$ . Thus, our evolution equation is modified into

$$\frac{\partial S}{\partial t} = \chi(x, y, z) \left[ \frac{(1 + S_x^2 + S_y^2)S_{zz} + (1 + S_x^2 + S_z^2)S_{yy}}{1 + S_x^2 + S_y^2 + S_z^2} + \frac{(1 + S_y^2 + S_z^2)S_{xx} - 2S_x S_y S_{xy} - 2S_x S_z S_{xz} - 2S_z S_y S_{yz}}{1 + S_x^2 + S_y^2 + S_z^2} \right]. \tag{71}$$

At the same time, corresponding modifications have to be made in formulas described in the last section. For instance, We set  $D_{1ijk}^n$  into

$$D_{1ijk}^n = \left\{ \chi(x, y, z) \frac{1 + (\delta_y S)^2 + (\delta_z S)^2}{1 + (\delta_x S)^2 + (\delta_y S)^2 + (\delta_z S)^2} \right\}_{ijk}^n. \tag{72}$$

After evolving Eq. (71), an approximate steady state solution of  $S(x, y, z, t)$  is attained at a large value of  $t = T$  and is a smooth function on a 3D domain with rapid changes near the protected atomic boundaries of  $D_\chi$ . However, the hypersurface  $S(x, y, z, T)$  obtained is not the MMS that we seek. Instead, it gives rise to a family of level surfaces, which include the desired MMS. Note that that  $S(x, y, z, T) \equiv S_0$  on  $D_\chi$ . It turns out that  $S(x, y, z, T)$  is very flat away from the MMS, while it varies sharply at the MMS. In other words,  $S(x, y, z, t)$  is virtually a step function at the desirable MMS.

Therefore, it is easy to extract the MMS as an isosurface,  $S(x, y, z, t) = C$ . It is convenient to choose  $C = (1 - \epsilon)S_0$ , where  $S_0$  is the initial amplitude, and  $\epsilon > 0$  is a very small number. Computationally, by taking  $S_0 = 1000$ , satisfactory results can be attained by using  $\epsilon$  values ranging from 0.004 to 0.01.

Before we proceed to the numerical comparison, we point out that in the semi-implicit scheme, the linear system is solved by the bi-conjugate gradient scheme in the smooth test case and by the successive over-relaxation algorithm (SOR) in test cases with rapid changing solutions, such as proteins and our diatomic system. The choice is based on the speed of the convergence. The SOR is a popular solver for nonlinear differential equations, since it is easy to implement and additional memory is not required. It is also possible to solve the problem as a steady state one, i.e., a boundary value problem. However, our test which is not reported in this paper indicates this approach is not necessarily better than the evolution equation approach in terms of efficiency. In particular, the fast Fourier transformation (FFT) solver fails because of the characteristic function  $\chi(x, y, z)$  in Eq. (71) which is equivalent to the computational domain being inhomogeneous. It is found that if the characteristic function  $\chi$  is applied via a post processing approach (i.e.,  $S$  is updated in the whole domain followed by setting back to  $S_0$  in  $D_\chi$ ), the accuracy of integration cannot be ensured. In the Crank–Nicolson scheme, the linear system is solved by using the bi-conjugate gradient scheme for the smooth test case and by the SOR in the MMS generation. The Crank–Nicolson scheme is of second order convergence in time in the smooth test. However, its convergence is hardly reached because of the rapid changes of the solution. On the other hand, the ADI-based schemes can easily implement the van der Waals surface constraint. They require only a 1D problem’s memory storage. The computational complexity is of  $O(N)$ , where  $N = N_x N_y N_z$ , because of the use of the Thomas algorithm.

#### 4.2.2 Numerical test involving smooth solutions

It is well known that the error of a 3D Douglas ADI scheme for linear diffusion equation is of second-order convergence in space, and of first-order convergence in time. We examine the order of accuracy of the proposed schemes for solving the mean curvature flow equation in the present study. Since our geometric flow equation does not admit a closed form solution, we consider a modified nonlinear diffusion equation with an exact solution. Consequently, numerical results can be analytically validated. In particular, we consider the following modified mean curvature equation

$$\frac{\partial S}{\partial t} = \frac{(1 + S_x^2 + S_y^2)S_{zz} + (1 + S_x^2 + S_z^2)S_{yy} + (1 + S_y^2 + S_z^2)S_{xx}}{1 + S_x^2 + S_y^2 + S_z^2} + \frac{-2S_x S_y S_{xy} - 2S_x S_z S_{xz} - 2S_z S_y S_{yz}}{1 + S_x^2 + S_y^2 + S_z^2} - \frac{(B_2 - B_3)}{B_1} - B_4 \tag{73}$$

$$B_1 = 1 + (\cos(t))^2((\cos(x) \sin(y) \sin(z))^2 + (\sin(x) \cos(y) \cos(z))^2 + (\sin(x) \sin(y) \cos(z))^2) \tag{74}$$

**Table 2** Numerical convergence in space for the proposed schemes

$n_x \times n_y \times n_z$	ADI (b)		ADI (a)		Semi-implicit		Crank–Nicolson	
	$L_\infty$	Order	$L_\infty$	Order	$L_\infty$	Order	$L_\infty$	Order
$4 \times 4 \times 4$	0.214		0.214		0.217		0.216	
$8 \times 8 \times 8$	5.09(-2)	2.07	5.08(-2)	2.07	5.30(-2)	2.03	5.18(-2)	2.06
$16 \times 16 \times 16$	1.23(-2)	2.05	1.23(-2)	2.07	1.48(-2)	1.84	1.31(-2)	1.98

**Table 3** Numerical convergence in time for the proposed schemes

$\tau$	ADI (b)		ADI (a)		Semi-implicit		Crank–Nicolson	
	$L_\infty$	Order	$L_\infty$	Order	$L_\infty$	Order	$L_\infty$	Order
0.4	0.161		0.156		0.362		6.59(-3)	
0.2	9.02(-2)	0.84	8.84(-2)	0.83	0.187	0.95	1.55(-3)	2.09
0.1	4.75(-2)	0.93	4.68(-2)	0.92	9.48(-2)	0.98	3.78(-4)	2.04

$$B_2 = -(\cos(t))^3 \sin(x) \sin(y) \sin(z)(3(\cos(t))^{-2} + 2(\cos(x) \sin(y) \sin(z))^2 + 2(\sin(x) \cos(y) \sin(z))^2 + 2(\sin(x) \sin(y) \cos(z))^2) \tag{75}$$

$$B_3 = 2(\cos(t))^3 \sin(x) \sin(y) \sin(z)((\cos(x) \cos(y) \sin(z))^2 + (\cos(x) \sin(y) \cos(z))^2 + (\sin(x) \cos(y) \cos(z))^2) \tag{76}$$

$$B_4 = \sin(x) \sin(y) \sin(z) \sin(t) \tag{77}$$

The exact solution is designed to be

$$S(x, y, z, t) = \sin(x) \sin(y) \sin(z) \cos(t). \tag{78}$$

This modified equation has the same differential operator as the mean curvature flow equation, but with extra source terms.

Consider a cube domain  $[0, 2\pi] \times [0, 2\pi] \times [0, 2\pi]$  and  $t > 0$ . Note that the solution is smooth for this test problem. Table 2 lists the computed errors in different mesh sizes. The standard absolute  $L_\infty$  norm error measurement is employed. Grid numbers in  $(x, y, z, t)$  direction are represented by  $(n_x, n_y, n_z, n_t)$ . Here we let  $T = \tau \times n_t$  and fix  $T = 10, \tau = 0.05$  and thus,  $n_t = 2000$ . The second order convergence in space is observed for the ADI schemes, the Crank–Nicolson scheme and the semi-implicit scheme. In Table 3, we fix spatial mesh size with  $48 \times 48 \times 48$  and set  $T = 10$ . First order convergence in time is obtained for the ADI schemes and the semi-implicit scheme. The Crank–Nicolson scheme shows second order convergence in time and has the best performance. Furthermore, by varying  $\tau$ , it is found that ADI schemes, the Crank–Nicolson scheme and the semi-implicit scheme are all stable in this test case, while the forward Euler scheme needs a very small  $\tau$  to make the computation stable. For instance, if we select a  $32 \times 32 \times 32$  mesh for the forward Euler method, it blows up when we take  $\tau = 0.01$ , whereas we have  $L_\infty = 2.63 \times 10^{-2} = 2.63(-2)$  if  $\tau = 0.005$ .

### 4.3 Numerical test involving rapidly changing solutions

In this subsection, we perform a comparison in terms of efficiency for the proposed schemes in solving the mean curvature flow equation. Both numerical accuracy and the computational efficiency are investigated in a diatomic system, an amino acid and six proteins. For the amino acid and proteins, atomic coordinates are obtained from the protein data bank (PDB) and their atomic van der Waals radii are taken from the CHARMM22 force field [47]. All of the computations are carried out on a SGI Altix 350 workstation with four 1.4 GHz itanium 2 Processors and 4 GB memory. For these test systems, there is no closed form solution available. To benchmark our results, we consider a reference solution by using the explicit Euler scheme with a small step size and large number of iterations, and with  $T = 20$ . This reference solution is essentially a steady state solution and is used to compare with other solutions obtained at a shorter time, but long enough to also bring the solution close to steady state. The relative  $L^2$  error measurement is employed in efficiency studies. We show that some of the proposed schemes are able to deliver a good approximation to the reference solution with a small CPU cost.

#### 4.3.1 A diatomic system

First let us consider a diatomic system with the atomic radius  $2 \text{ \AA}$  and atomic centers at  $\mathbf{r}_1 = (-2.2, 0, 0)$  and  $\mathbf{r}_2 = (2.2, 0, 0)$ . The mesh size is set to  $53 \times 31 \times 31$  with a uniform mesh size of  $h = 0.2 \text{ \AA}$ . It is found that although the ADI (b) scheme is stable over a large time increment for the previous modified mean curvature flow equation, it becomes conditionally stable while being applied to real biomolecular systems, including the diatomic system, because of the rapid spatial variation in the solution. Nevertheless, the critical time step  $\tau$  of the ADI (b) scheme is usually much larger than that of the explicit Euler scheme, so that considerable efficiency gain can still be attained by using such a large  $\tau$ . For the present study, it has been found that ADI (b) scheme admits a time step  $\tau = 0.15$ , which is about 20 times larger than the time step limit  $\tau = 0.008$  in the explicit scheme. On the other hand, by varying  $\tau$ , it is found that the ADI (a) scheme and semi-implicit scheme are stable at a large time increment.

Table 4 lists the results of CPU time and relative  $L^2$  errors with different time increments. Here we set  $T = 3.5$  for the proposed schemes and compute the reference solution by using the explicit scheme with  $\tau = 0.002$  and  $T = 20$ . It is observed that ADI (b) scheme with time step size  $\tau = 0.15$  takes the smallest CPU time if relative  $L^2$  error is required to be less than 2%. The Crank–Nicolson scheme converges slowly and requires a large CPU time. It is omitted from further comparison. One measure of efficiency might be the  $Q$  factor, computed as the product of the CPU time and relative  $L^2$  error. A small  $Q$  factor is desired. Clearly, the ADI (b) has smaller  $Q$  factors. Note that the explicit Euler scheme has the smallest  $Q$  factor. However, one usually wishes to have less CPU cost at an acceptable error, say about 2%. One way to reduce the CPU cost is to use a larger time increment. However,  $\tau = 0.008$  is close to the stability limit of the explicit Euler scheme. Another way to reduce the CPU time is to choose a smaller  $T$ . Indeed, at  $T = 2$ , the relative  $L^2$  error of the explicit Euler scheme is

**Table 4** Comparison of CPU time and errors in a diatomic system ( $T = 3.5$ )

Schemes	$\tau$	CPU time (s)	$L_2$ error (%)	$Q$
Forward Euler	0.008	1.70	0.12	0.2
Crank–Nicolson	0.15	16.7	16.8	280.6
ADI (b)	0.15	0.52	1.83	1.0
	0.10	0.77	1.34	1.0
	0.05	1.56	0.34	0.5
	0.01	7.78	0.06	0.5
ADI (a)	0.15	0.55	9.47	5.2
	0.1	0.84	2.54	2.1
	0.08	1.07	1.61	1.7
	0.01	8.56	0.74	6.3
Semi-implicit	0.8	0.35	15.5	5.4
	0.4	0.52	5.84	2.8
	0.2	0.94	1.75	1.6
	0.15	1.46	0.79	1.2
	0.1	7.19	0.30	2.2
	0.08	9.42	0.06	0.5

2.1%, obtained with the CPU time of 0.97, which gives  $Q = 2$ . Such a result is not as good as those of the semi-implicit and ADI schemes can deliver.

#### 4.3.2 Amino acid

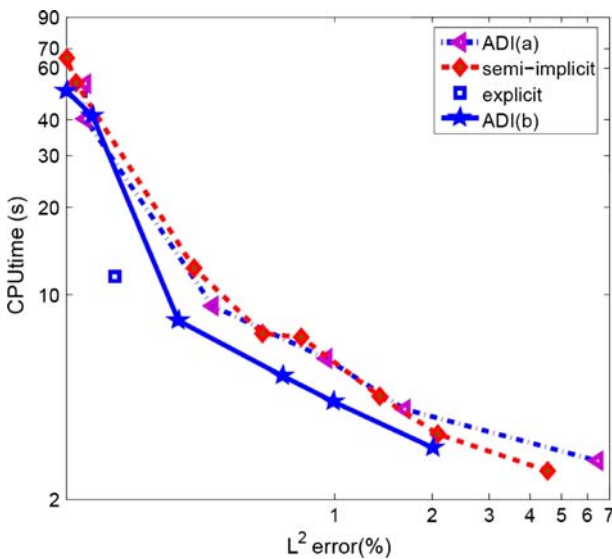
We next consider an amino acid system—tryptophan (Trp), which is listed as a ligand chemical component for a DNA binding regulatory protein, PDB Id: 1wrp. Trp has 27 atoms. Atomic van der Waals radii are taken from the CHARMM22 force field [47], and are in the range of 1–2 Å. In our computation, a mesh of  $63 \times 49 \times 53$  with uniform mesh size  $h = 0.2$  Å is used. It is found that the time step limit for the ADI (b) scheme with an accuracy of better than 2% is still  $\tau = 0.15$  while the time step limit for the explicit scheme to remain stable is  $\tau = 0.008$ . Moreover, by varying the  $\tau$ , it is found that the ADI (a) and semi-implicit schemes are stable. All of these findings agree with the observations from the diatomic system.

Table 5 compares the explicit Euler scheme, the semi-explicit scheme, and the ADI schemes. Here we still set  $T = 3.5$  for the proposed schemes and compute a reference solution using the explicit scheme with  $\tau = 0.002$  and  $T = 20$ . Fig. 12 gives a graphical representation of Table 5 which allows us to find directly the most efficient schemes for a desired accuracy. The ADI (b) scheme appears to out-performs the other methods.

Both the ADI (b) and reference solutions are shown in Fig. 13 for a comparison between the contours at  $x$ ,  $y$ , and  $z$  plane cross sections. Good agreement can be seen. For the purpose of achieving 2% relative  $L_2$  error, the ADI (b) is about 1.5 times more efficient than the semi-implicit and ADI (a) schemes, while it is 4 times faster than the explicit scheme.

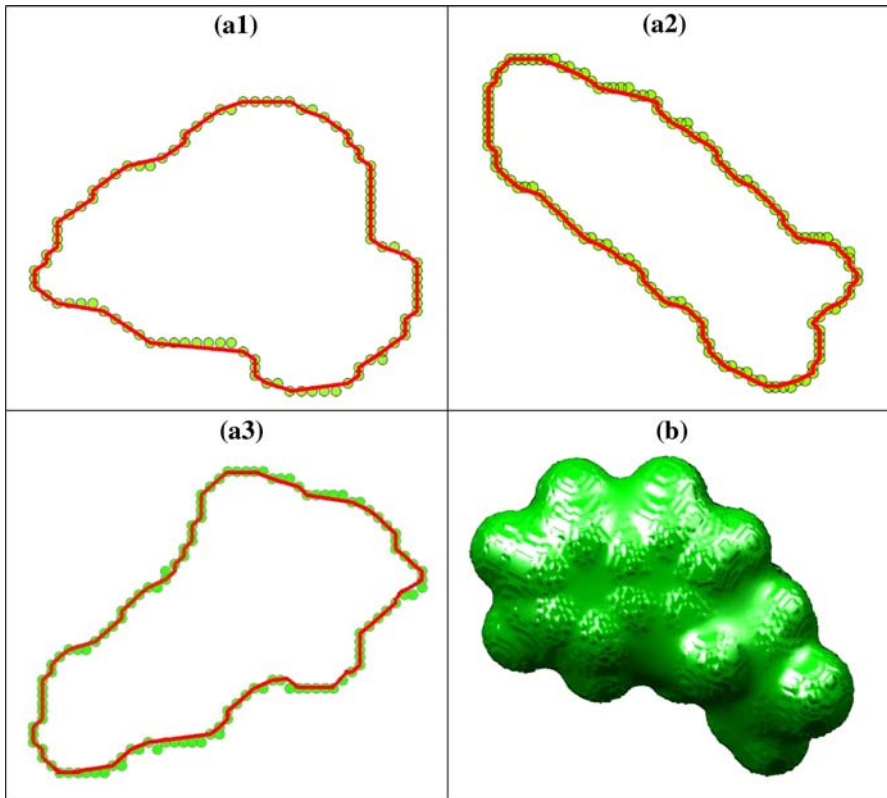
**Table 5** Comparison of CPU time and error in computing the surface of Amino acid Trp

Schemes	$\tau$	CPU time (s)	$L_2$ error (%)
Euler scheme	0.008	11.57	0.21
ADI (b)	0.15	3.0	2.01
	0.10	4.32	0.99
	0.08	5.30	0.69
	0.05	8.17	0.33
	0.01	41.07	0.18
	0.008	50.24	0.15
ADI (a)	0.15	2.71	6.45
	0.10	4.08	1.64
	0.08	6.06	0.94
	0.05	9.71	0.42
	0.01	40.36	0.17
	0.008	53.15	0.17
Semi-implicit	0.8	1.30	9.29
	0.4	2.50	4.53
	0.2	3.34	2.08
	0.15	4.49	1.38
	0.1	7.19	0.79
	0.08	7.40	0.60
	0.05	12.39	0.37
	0.01	53.3	0.16
	0.008	65.19	0.15



**Fig. 12** Loglog plots of the efficiency against the accuracy for the surface formation of the Trp





**Fig. 13** Visual comparison of the contours of  $S(x, y, z, T)$  between the ADI (b) solution (*solid lines*) with  $\tau = 0.15$ ,  $T = 3.5$  and the reference solution (*dots*) for Trp. *a1* contours of  $S(x = 0, y, z, T)$ ; *a2* contours of  $S(x, y = 0, z, T)$ ; *a3* contours of  $S(x, y, z = 0, T)$ ; *b* isosurface of Trp

#### 4.3.3 Proteins

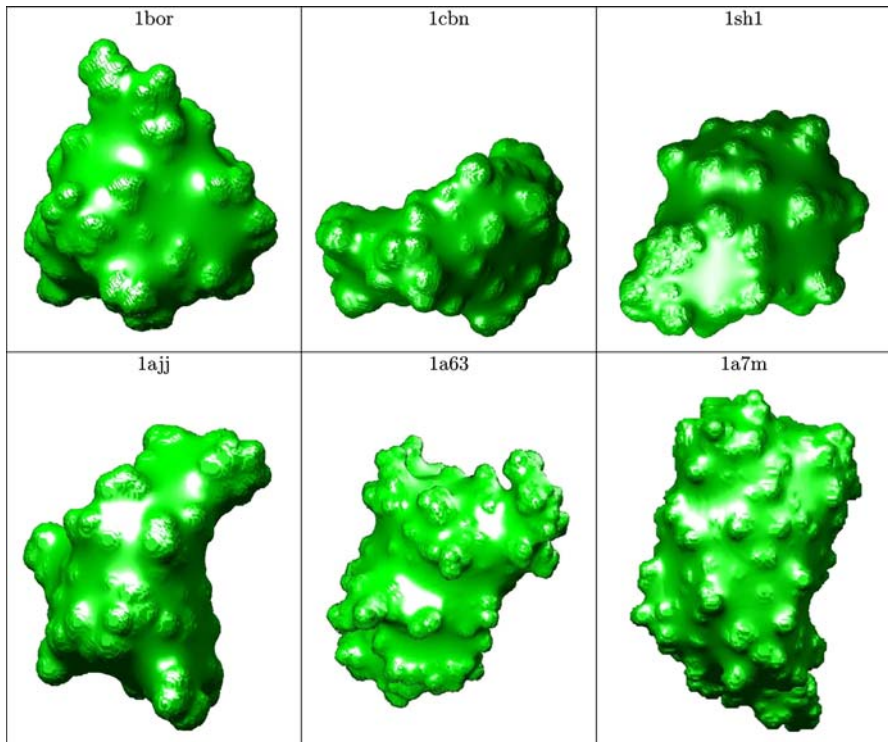
We further explore the performance of our schemes by considering a set of six proteins whose coordinates are available in the Protein Data Bank (PDB). The number of atoms and computational meshes of these six proteins are listed in Table 6. The MMSs of these proteins are generated by using  $h = 0.4 \text{ \AA}$ . Based on the accuracy considerations and the previous results, we choose  $\tau = 0.3$  for the ADI (a),  $\tau = 0.4$  for the ADI (b),  $\tau = 0.4$  for the semi-implicit and  $\tau = 0.03$  for the explicit scheme. It is noted that the bigger (or more complex) the protein is chosen, the longer evolution time is needed to reach the steady state surface, because of larger amount of total deformation involved. Therefore we set  $T = 15$  for these protein computations. Table 7 gives the final CPU time comparison. It is seen from the table that the ADI (b) scheme is the fastest scheme. The final MMSs generated by the ADI (b) scheme with  $\tau = 0.4$  are given in Fig. 14. Therefore, the conclusion drawn from the diatomic and the amino acid systems is in fact quite general. We have found that the proposed ADI (b) scheme is the most efficient for the desired level of accuracy in generating MMSs for different biomolecular systems.

**Table 6** Tested proteins

PDB ID	Number of atoms	Mesh
1ajj	519	$69 \times 76 \times 83$
1cbn	648	$81 \times 72 \times 84$
1sh1	702	$72 \times 75 \times 83$
1bor	832	$84 \times 84 \times 89$
1a63	2065	$163 \times 101 \times 103$
1a7m	2809	$106 \times 153 \times 129$

**Table 7** Efficiency comparison for proteins. CPU time in second is reported

Scheme	1ajj	1cbn	1sh1	1bor	1a63	1a7m
ADI (a)	18	23	18	28	74	110
ADI (b)	14	17	14	23	63	107
Semi-implicit	22	35	25	33	100	190
Explicit	46	71	58	89	234	359

**Fig. 14** Minimal molecular surfaces of six proteins

## 5 Conclusion

In the multiscale implicit solvent models, surfaces are required to separate biomolecules and continuum solvent. Previously, a mean curvature flow approach was proposed for the formation of minimal molecular surfaces [4]. However, local inhomogeneity exists in density, pressure and other macroscopic variables near the biomolecular surface due to a variety of interactions, such as polar–nonpolar, ion–counterion, etc. These interactions induce surface responses to supplement the geometric driving forces. This paper proposes new partial differential equations (GPDEs) for the biological surface formation and evolution. Normally, geometric equations make use of intrinsic geometric forces, such as curvature forces, for the formation and post processing of surfaces. In the present study, we propose GPDEs that utilize not only intrinsic geometric forces, but also potential forces for the formation and evolution of biological surfaces. Apart from geometric and potential driving biomolecular surface flows, stochastic geometric flows were also proposed to account for density and pressure variations, and for the realization of the canonical and grand canonical ensembles. Generalized geometric flows that ensure the surface area is decreasing and the volume is preserved are proposed, consistent with physical requirements. High order geometric flows, which permit control near the boundary, are also proposed. Applications are carried out for proposed potential driving geometric flows and high order geometric flows. Numerical experiments are performed with the Eulerian formulation. Appropriate balance between the geometric and potential forces are considered for biological surface formation. We show that with a proper selection of the potential term, a good approximation to the conventional molecular surface can be achieved. Local modifications of biological surfaces are illustrated with appropriate position dependent potential terms. All proposed new biomolecular surfaces are free of geometric singularities.

The second part of this paper is devoted to the design and testing of computational algorithms for the generation of MMSs. Four algorithms, a semi-implicit scheme using successive over-relation (SOR), a Crank–Nicolson scheme and two alternating direction implicit (ADI) schemes, ADI (a) and ADI (b), have been proposed. Through a designed benchmark problem with a closed form solution, the proposed schemes are shown to be stable for smooth solutions by varying the time increment. Both ADI schemes make use of the Thomas algorithm that reduces the computational complexity. The Crank–Nicolson scheme is of second order convergence in both space and time for the modified mean curvature equation. However, it does not work well for the MMS involving rapidly changing solutions. All other schemes are of second order convergence in space and first order in time for the modified mean curvature equation. Several biomolecular systems have been considered to validate our algorithms. The ADI (b) scheme is found to be conditionally stable when it is applied to real biomolecular systems which involve rapid changes in solutions. Nevertheless, this does not affect the efficiency gain of the ADI (b) scheme, since it usually allows a much larger time step than the forward Euler method. Although both ADI (a) and semi-implicit schemes are stable, they require a smaller time step than ADI (b) to achieve a given level of accuracy. Consequently, the ADI (b) scheme is an efficient scheme for the generation of the MMS. It is

usually four times faster than the forward Euler scheme based on a coarse mesh. Such an efficiency gain can be larger when a denser mesh is used. We finally note that the proposed fast algorithms can be easily applied to the solution of many other PDEs.

**Acknowledgments** This work was supported in part by NSF grant DMS-0616704. PWB was partially supported by DARPA FunBio HR-0011-05-0057. GWW was partially supported by NSF grant IIS-0430987 and NIH grant CA127189. SZ partially supported by NSF grant DMS-0731503.

## References

1. Ashbaugh HS (2000) Convergence of molecular and macroscopic continuum descriptions of ion hydration. *J Phys Chem B* 104:7235–7238
2. Baker NA (2005) Improving implicit solvent simulations: a Poisson-centric view. *Curr Opin Struct Biol* 15:137–143
3. Bates PW, Wei GW, Zhao S (2006) The minimal molecular surface, arXiv:q-bio/0610038v1 [q-bio.BM]
4. Bates PW, Wei GW, Zhao S (2008) Minimal molecular surfaces and their applications. *J Comput Chem* 29:380–391
5. Brakke K (1978) The motion of a surface by its mean curvature, *Mathematical Notes*, vol 20. Princeton University Press, Princeton, NJ
6. Bertozzi AL, Greer JB (2004) Low-curvature image simplifiers: global regularity of smooth solutions and Laplacian limiting schemes. *Commun Pure Appl Math* 57:764–790
7. Blomgren PV, Chan TF (1990) Color TV: total variation methods for restoration of vector valued images. *IEEE Trans Image Process* 7:304–309
8. Bostrom M, Tavares FW, Bratko D, Ninham BW (2005) Specific ion effects in solutions of globular proteins: comparison between analytical models and simulation. *J Phys Chem B* 109:24489–24494
9. Carstensen V, Kimmel R, Sapiro G (1997) Geodesic active contours. *Int J Comput Vis* 22:61–79
10. Cecil T (2005) A numerical method for computing minimal surfaces in arbitrary dimension. *J Comput Phys* 206:650–660
11. Cerutti DS, Baker NA, McCammon AJ (2007) Solvent reaction field potential inside an uncharged globular protein: a bridge between Implicit and Explicit Solvent Models? *J Chem Phys* 127:155101
12. Chan TF, Marquina A, Mulet P (2000) High-order total variation-based image restoration. *SIAM J Sci Comput* 22:503–516
13. Cheng L-T, Dzubiella J, McCammon JA, Li B (2007) Application of the level-set method to the solvation of nonpolar molecules. *J Chem Phys* 127:084503
14. Chopp DL (1993) Computing minimal-surfaces via level set curvature flow. *J Comput Phys* 106:77–91
15. Chorny I, Dill KA, Jacobson MP (2005) Surfaces affect ion pairing. *J Phys Chem B* 109:24056–24060
16. Ciarlet PG, Lions JL (1990) *Handbook of numerical analysis, finite difference methods (part 1), Solution of Equations in  $\mathbb{R}^n$* , vol 1. Elsevier Science, Amsterdam
17. Clarenz U, Diewald U, Dziuk G, Rumpf M, Rusu R (2004) A finite element method for surface restoration with boundary conditions. *Comput Aided Geom Des* 21(5):427–445
18. Connolly ML (1985) Molecular surface triangulation. *J Appl Crystallogr* 18:499–505
19. Cortis CM, Friesner RA (1997) Numerical solution of the Poisson–Boltzmann equation using tetrahedral finite-element meshes. *J Comput Chem* 18:1591–1608
20. Droske M, Rumpf M (2004) A level set formulation for Willmore flow. *Interfaces Free Boundaries* 6(3):361–378
21. Dzubiella J, Swanson JMJ, McCammon JA (2006) Coupling nonpolar and polar solvation free energies in implicit solvent models. *J Chem Phys* 124:084905
22. Eisenhaber F, Argos P (1993) Improved strategy in analytic surface calculation for molecular systems: Handling of singularities and computational efficiency. *J Comput Chem* 14:1272–1280
23. Escher J, Mayer UF, Simonett G (1998) The surface diffusion flow for immersed hypersurfaces. *SIAM J Math Anal* 29(6):1419–1433
24. Feig M, Brooks III CL (2004) Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Curr Opin Struct Biol* 14:217–224

25. Feig M, Onufriev A, Lee MS, Im W, Case DA, Brooks III CL (2004) Performance comparison of Generalized Born and Poisson methods in the calculation of electrostatic solvation energies for protein structures. *J Comp Chem* 25:265–284
26. Feng XB, Prohl A (2004) Analysis of a fully discrete finite element method for the phase field model and approximation of its sharp interface limits. *Math Comput* 73:541–567
27. Fixman M (1979) The Poisson–Boltzmann equation and its application to polyelectrolytes. *J Chem Phys* 70:4995–5005
28. Forsman J (2004) A simple correlation-corrected Poisson–Boltzmann theory. *J Phys Chem B* 108: 9236–9245
29. Forsten KE, Kozack RE, Lauffenburger DA, Subramaniam S (1994) Numerical solution of the nonlinear Poisson–Boltzmann equation for a membrane–electrolyte system. *J Phys Chem* 98:5580–5586
30. Gage M, Hamilton RS (1986) The heat equation shrinking convex plane curves. *J Diff Geom* 23:69–96
31. Gilboa G, Sochen N, Zeevi YY (2004) Image sharpening by flows based on triple well potentials. *J Math Imaging Vis* 20:121–131
32. Gogonea V, Osawa E (1994) Implementation of solvent effect in molecular mechanics. 1. Model development and analytical algorithm for the solvent-accessible surface area. *Supramol Chem* 3: 303–317
33. Gomes J, Faugeras O (2001) Using the vector distance functions to evolve manifolds of arbitrary codimension. *Lect Notes Computer Sci* 2106:1–13
34. Grant JA, Pickup BT, Nicholls A (2001) A smooth permittivity function for Poisson–Boltzmann solvation methods. *J Comput Chem* 22:608–640
35. Greer JB, Bertozzi AL (2004) H-1 solutions of a class of fourth order nonlinear equations for image processing. *Discrete Continuous Dyn Syst* 10:349–366
36. Greer JB, Bertozzi AL (2004) Traveling wave solutions of fourth order PDEs for image processing. *SIAM J Math Anal* 36:38–68
37. He L, Kao C-Y, Osher S (2007) Incorporating topological derivatives into shape derivatives based level set methods. *J Comput Phys* 225:891–909
38. Honig B, Nicholls A (1995) Classical electrostatics in biology and chemistry. *Science* 268:1144–1149
39. Huiskens G (1987) The volume preserving mean curvature flow. *J Reine Angew Math* 382:35–48
40. Im W, Beglov D, Roux B (1998) Continuum solvation model: Computation of electrostatic forces from numerical solutions to the Poisson–Boltzmann equation. *Comput Phys Commun* 111:59–75
41. Lawson HB (1980) *Lectures on minimal submanifolds*, Publish or Perish, Berkeley
42. Lee B, Richards FM (1973) Interpretation of protein structures: estimation of static accessibility. *J Mol Biol* 55:379–400
43. Levy RM, Gallicchio E (1998) Computer simulation with explicit solvent: recent progress in the thermodynamic decomposition of free energies and in modeling electrostatic effects. *Annu Rev Phys Chem* 49:531–567
44. Li YY, Santosa F (1996) A computational algorithm for minimizing total variation in image restoration. *IEEE Trans Image Process* 5:987–995
45. Lu Q, Luo R (2003) A Poisson–Boltzmann dynamics method with nonperiodic boundary condition. *J Chem Phys* 119:11035–11047
46. Lysaker M, Lundervold A, Tai XC (2003) Noise removal using fourth-order partial differential equation with application to medical magnetic resonance images in space and time. *IEEE Trans Image Process* 12:1579–1590
47. MacKerell AD Jr, Bashford D, Bellott M, Dunbrack JD, Evanseck MJ, Field MJ, Fischer S, Gao J, Guo H, Ha S, Joseph-McCarthy D, Kuczera L, Lau FTK, Mattos C, Michnick S, Ngo T, Nguyen DT, Prodhom B, Reiher WE, Roux B, Schlenkrich M, Smith JC, Stote R, Straub J, Watanabe M, Wiorkiewicz-Kuczera J, Yin D, Karplus M (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J Phys Chem* 102:3586–3616
48. Mikula K, Sevcovic D (2004) A direct method for solving an anisotropic mean curvature flow of plane curves with an external force. *Math Methods Appl Sci* 27:1545–1565
49. Mumford D, Shah J (1989) Optimal approximations by piecewise smooth functions and associated variational problems. *Commun Pure Appl Math* 42:577–685
50. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 79:12–49
51. Osher S, Rudin L (1990) Feature-oriented image enhancement using shock filters. *SIAM J Numer Anal* 27:919–940

52. Osher S, Rudin L (1991) Shocks and other nonlinear filtering applied to image processing. *Proc SPIE Appl Digital Image Process XIV* 1567:414–430
53. Osher S (1993) A level set formulation for the solution of the dirichlet problem for Hamilton–Jacobi equations. *SIAM J Math Anal* 24:1145–1152
54. Osher S, Fedkiw RP (2001) Level set methods: an overview and some recent results. *J Comput Phys* 169:463–502
55. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Machine Intell* 12:629–639
56. Richards FM (1977) Areas, volumes, packing and protein structure. *Annu Rev Biophys Bioeng* 6:151–176
57. Roux B, Simonson T (1999) Implicit solvent models. *Biophys Chem* 78:1–20
58. Rudin L, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithm. *Physica D* 60:259–268
59. Sanner MF, Olson AJ, Spehner JC (1996) Reduced surface: an efficient way to compute molecular surfaces. *Biopolymers* 38:305–320
60. Sapiro G, Ringach D (1995) Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans Image Process* 5:1582–1586
61. Sapiro G (1996) From active contours to anisotropic diffusion: Relation between basic PDE's in image processing. *Proc. ICIP, Lausanne*
62. Sarti A, Malladi R, Sethian JA (2002) Subjective surfaces: a geometric model for boundary completion. *Int J Comput Vis* 46:201–221
63. Sbert C, Sole AF (2003) 3D curves reconstruction based on deformable models. *J Math Imag Vis* 18:211–223
64. Schneider R, Kobbelt L (2000) Generating fair meshes with  $G^1$  boundary conditions, *Geometric Modeling and Processing, Hong Kong, China*, pp 251–261
65. Schneider R, Kobbelt L (2001) Geometric fairing of irregular meshes for free-form surface design. *Comput Aided Geom Des* 18(4):359–379
66. Sethian JA (2001) Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *J Comput Phys* 169:503–555
67. Sharp KA, Honig B (1990) Electrostatic interactions in macromolecules: theory and applications. *Annu Rev Biophys Biochem* 19:301–332
68. Shen JJ (2006) A stochastic-variational model for soft Mumford-Shah segmentation. *Int J Biomed Imaging* 92329:1–14
69. Simonett G (2001) The Willmore flow for near spheres. *Differential Integral Equations* 14(8):1005–1014
70. Smereka P (2003) Semi-implicit level set methods for curvature and for motion by surface diffusion. *J Sci Comput* 19:439–456
71. Sochen N, Kimmel R, Malladi R (1998) A general framework for low level vision. *IEEE Trans Image Process* 7:310–318
72. Sun YH, Wu PR, Wei GW, Wang G (2006) Evolution operator based single-step method for image processing. *Int J Biomed Imaging* 83847:1–27
73. Swanson MJ, Henchman RH, McCammon JA (2004) Revisiting free energy calculations: a theoretical connection to MM/PBSA and direct calculation of the association free energy. *Biophys J* 86:67–74
74. Tan C, Tan Y-H, Luo R (2007) Implicit nonpolar solvent models. *J Phys Chem B* 111:12263–12274
75. Vorobjev YN, Scheraga HA (1997) A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *J Comput Chem* 18:569–583
76. Wagoner J, Baker NA (2004) Solvation forces on biomolecular structures: a comparison of explicit solvent and Poisson–Boltzmann models. *J Comput Chem* 25:1623–1629
77. Wang XQ, Du Q (2008) Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches. *J Math Biol* 56:347–371
78. Wang Y, Zhou HM (2006) Total variation wavelet-based medical image denoising. *Int J Biomed Imaging* 2006:89095
79. Warwicker J, Watson HC (1982) Calculation of the electric-potential in the active-site cleft due to alpha-helix dipoles. *J Mol Biol* 154:671–679
80. Wei GW (1999) Generalized Perona–Malik equation for image restoration. *IEEE Signal Process Lett* 6:165–167
81. Wei GW, Jia YQ (2002) Synchronization based image edge detection. *Europhys Lett* 59:814–819

82. Wei GW, Sun YH, Zhou YC, Feig M (2005) Molecular multiresolution surfaces, arXiv:math-ph, 0511001, 1 Nov 2005
83. Wei GW, Bates PW, Zhao S (2007) Geometric flows on biological surfaces, In: Mathematics of DNA structure, function, and interactions. IMA, 16–21 September 2007. <http://www.ima.umn.edu/2007-2008/W9.16-21.07/abstracts.html>
84. Willmore TJ (1993) Riemannian geometry. Clarendon Press, Oxford
85. Xu G, Pan Q, Bajaj C (2006) Discrete surface modelling using partial differential equations. *Comput Aided Geom Des* 23(2):125–145
86. Xu M, Zhou SL (2007) Existence and uniqueness of weak solutions for a fourth-order nonlinear parabolic equation. *J Math Anal Appl* 325:636–654
87. Yoshizawa S, Belyaev AG (2002) Fair triangle mesh generation with discrete elastica. *Geometric modeling and processing*, Saitama, Japan, pp 119–123
88. You YL, Kaveh M (2000) Fourth-order partial differential equations for noise removal. *IEEE Trans Image Processing* 10:1723–1730
89. Yu SN, Geng WH, Wei GW (2007) Treatment of geometric singularities in the implicit solvent models. *J Chem Phys* 126:244108
90. Zauhar RJ, Morgan RS (1985) A new method for computing the macromolecular electric-potential. *J Mol Biol* 186:815–820
91. Zhang Y, Xu G, Bajaj C (2006) Quality meshing of implicit solvation models of biomolecular structures. *Comput Aided Geom Des* 23:510–530
92. Zhao HK, Chan TF, Merriman B, Osher S (1996) A variational level set approach to multiphase motion. *J Comput Phys* 127:179–195