Journal of Computational Physics **168,** 445–463 (2001)
doi:10.1006/jcph.2001.6713, available online at http://www.idealibrary.com on IDEAL®

An Adaptive Mesh Algorithm for Evolving Surfaces: Simulations of Drop Breakup and Coalescence

Vittorio Cristini,[1] Jerzy Blawzdziewicz, and Michael Loewenberg

Department of Chemical Engineering, Yale University, New Haven, Connecticut 06520-8286
E-mail: michael.loewenberg@yale.edu

Received August 10, 2000; revised December 15, 2000

An algorithm is presented for the adaptive restructuring of meshes on evolving surfaces. The resolution of the relevant local length scale is maintained everywhere with prescribed accuracy through the minimization of an appropriate mesh energy function by a sequence of local restructuring operations. The resulting discretization depends on the instantaneous configuration of the surface but is insensitive to the deformation history. Application of the adaptive discretization algorithm is illustrated with three-dimensional boundary-integral simulations of deformable drops in Stokes flow. The results show that the algorithm can accurately resolve detailed features of deformed fluid interfaces, including slender filaments associated with drop breakup and dimpled regions associated with drop coalescence. Our algorithm should be useful in a variety of fields, including computational fluid dynamics, image processing, geographical information systems, and biomedical engineering problems.

1. INTRODUCTION

The evolution of a surface typically depends on one or more surface-defined dynamical quantities. These quantities determine the local length scale L, which must be resolved in numerical simulations. For example, the dynamics of deformable drops is determined by the balance of hydrodynamic and capillary stresses thus, L is set by the local radius of curvature. If surfactants are adsorbed on the interface, variations of surface tension may also influence L. For the evolution of a biological membrane, L may depend on the distribution of strain.

[1] Present address: Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455-0132.

0021-9991/01 $35.00

The simplest surface discretization for numerical simulations involves a uniform structured mesh with fixed topology. However, in dynamic simulations this approach is limited to modest deformations of the surface, because the required length-scale distribution evolves. Thus, fixed meshes must resolve the anticipated global minimum of L everywhere.

Adaptive algorithms have been developed for discretization of static structures [1–3]. To a more limited extent, remeshing algorithms have been developed for numerical simulations of deformable interfaces. Most of these algorithms involve local mesh refinement and reconnection [4, 5]; others involve global reconstruction [6] when significant loss of interface resolution is detected.

Herein, we present an adaptive discretization algorithm for evolving surfaces, which is based on the minimization of the mesh configurational energy. The mesh energy is an integral measure of how local edge lengths l deviate from an optimal value $l_0 \sim L$. The minimum mesh energy corresponds to the optimal mesh, which depends only on the instantaneous configuration of the surface. The optimal mesh is achieved by the minimization of mesh energy through a sequence of local restructuring operations, with dynamics that are defined by the mesh energy function.

Our discretization algorithm, including the definition of the mesh energy function and local mesh restructuring, is described in Section 2. Mesh adaptivity is illustrated in Section 3. In Section 4, the algorithm is applied to boundary-integral simulations of deformable drops. Other applications and extensions of the algorithm are discussed in Section 5.

2. ALGORITHM

2.1. Mesh-Energy Function

Our adaptive mesh algorithm relies on the mesh-energy function

$$E = \frac{1}{2} \sum_{\text{edges}} \gamma^2, \tag{1}$$

where the edge tensions are

$$\gamma = l - l_0, \tag{2}$$

where l_0 is the optimal edge length. The mesh is thus analogous to a system of springs.

The optimal edge length is related to the local length scale L,

$$l_0 = \alpha \langle L \rangle_{\text{loc}}, \tag{3}$$

where the constant α sets the resolution of the surface, and $\langle \cdots \rangle_{\text{loc}}$ is a weighted average over a local region S_{loc} with $O(l_0)$ extent. Local averaging is needed to achieve a smooth distribution of l_0 on a discretized surface.

Minimization of the mesh energy (1) implies

$$l \approx l_0; \tag{4}$$

the corresponding number of nodes is given by $N \sim \int_S l_0^{-2}\, dS$, where the integration is performed over the entire surface.

2.1.1. *Local Length-Scale Distribution*

In general, the length-scale distribution L is given by the local minimum

$$L = \min(L_0, L_1, L_2, \ldots), \tag{5}$$

where $L_1, L_2, \ldots$ are the relevant local length scales for a particular problem, and L_0 is a prescribed global upper bound.

In many applications uniform resolution of solid angle is desirable. Thus, L_1 is set by the local curvature, e.g.,

$$L_1^{-2} = \frac{1}{2}\left(\kappa_{\min}^2 + \kappa_{\max}^2\right), \tag{6}$$

where $\kappa_{\min}$ and $\kappa_{\max}$ are the minimum and maximum local curvatures. An additional length scale L_2 becomes relevant for problems involving interactions between closely spaced surfaces. For these problems, L_2 may depend on the quadratic variations of local gap width h between nearly parallel surfaces,

$$L_2 = c_2(L_R h)^{\frac{1}{2}}. \tag{7}$$

Here L_R is the reduced local radius of curvature defined by

$$L_R^{-1} = L_1^{-1} + L_1'^{-1}, \tag{8}$$

where L_1 and L_1' are given by (6), with L_1' evaluated at the closest point on the nearby surface. The prescribed constant c_2 determines the relative weights given to the length scales L_1 and L_2. Other local length scales can be similarly incorporated into the distribution (5).

2.2. Mesh Restructuring

Mesh-energy minimization is achieved through energy-reducing local restructuring, which involves a dynamical spring-like relaxation and topological transformations. These procedures are described below; certain details are provided in Appendix A.

The mesh restructuring procedures are iteratively performed until criteria (11), (12), and (A.1) are satisfied everywhere. The algorithm has $O(N)$ computational cost.

2.2.1. *Mesh Relaxation*

For fixed mesh topology, the energy (1) is reduced by the relaxation of a dynamical system of massless springs with tensions γ and damping coefficients of unity. Accordingly, the position $\mathbf{x}$ of a node evolves on the surface with a velocity $\dot{\mathbf{x}}$ proportional to the vector sum of spring forces acting on the node,

$$\dot{\mathbf{x}} = \hat{\mathbf{P}} \cdot \sum_{j=1}^{N_c} \hat{\mathbf{e}}_j \gamma_j, \tag{9}$$

where the summation is over the edges that emanate from the node, N_c is the node coordination number, $\hat{\mathbf{e}}_j$ is the unit vector parallel to edge j, and $\hat{\mathbf{P}} = \mathbf{I} - \mathbf{nn}$ is the surface projection tensor.

The dynamics of (9) involves a spectrum of relaxation times. Fast modes correspond to local $O(l_0)$ displacements with $O(1)$ relaxation times; slow modes are associated with long-wave perturbations of the node density $\delta\rho$. The relaxation of long-wave perturbations is described by a macroscopic diffusion equation

$$\dot{\delta}\rho \simeq -k^2\delta\rho, \tag{10}$$

where k is the magnitude of the wave vector normalized by the characteristic edge length of the mesh. The slowest modes correspond to $k \sim N^{-\frac{1}{2}}$, and have $O(N)$ relaxation time.

Relaxation of slow modes, with $O(N^2)$ computational cost, is circumvented by topological mesh restructuring described below. Thus, the relaxation process is terminated when

$$|\dot{\mathbf{x}}| < \epsilon, \tag{11}$$

where the equilibration parameter $\epsilon > N^{-\frac{1}{2}}$ controls the extent to which the mesh energy is minimized.

2.2.2. *Topological Transformations*

Topological restructuring operations are used to provide access to mesh configurations with lower energy than accessible by mesh relaxation, and to accelerate the equilibration of slow long-wave modes. The operations described here include node addition, subtraction, and reconnection.

Addition and subtraction of nodes. The criterion for node addition (subtraction) is based on the minimization of mesh energy through Eq. (4). Accordingly, nodes are added (subtracted) in a local region S_{loc}, unless

$$|n - n_0| \lesssim \Delta N, \tag{12}$$

where n_0 and n are the optimal and actual number of nodes in S_{loc}, and ΔN is the number of simultaneously added (subtracted) nodes in this region. The optimal number of nodes is

$$n_0 = S_{\text{loc}}\langle\rho_0\rangle'_{\text{loc}}, \tag{13}$$

where

$$\rho_0 = c_0\alpha^{-2}L^{-2} \tag{14}$$

is the node density corresponding to the prescribed resolution, and c_0 is a geometric factor. For a mesh of equilateral triangles $c_0 = 2 \cdot 3^{-\frac{1}{2}}$. The local averaging operator $\langle\cdots\rangle'_{\text{loc}}$ is similar to that in Eq. (3).

Node reconnection. Herein, unstructured meshes are sought, thus node reconnection is based on mesh isotropy, under the assumption that the mesh energy function (1) is isotropic. For meshes with triangular elements, isotropy is achieved by a Delaunay algorithm [4, 7–9].

2.3. Evolving Surfaces

2.3.1. *Surfaces Defined by Node Positions*

In many applications, the surface is defined only by node positions. In such cases, the node placement and tangential repositioning used in our algorithm require an accurate local representation of the surface. Many suitable representations are available [10–13]; in our implementation, we used a local parabolic representation [14].

2.3.2. *Evolution*

Our algorithm is particularly useful for applications involving discretization of evolving surfaces. In our approach, mesh restructuring and surface evolution are decoupled. Surface evolution is achieved by time-stepping the equation of motion

$$\dot{\mathbf{x}} = \mathbf{u}, \tag{15}$$

where $\mathbf{u}$ is the velocity of the surface at $\mathbf{x}$. The surface resolution is maintained by minimizing the mesh energy after each time step. The resulting mesh is thus insensitive to the history of evolution. In most applications, the computational cost of mesh restructuring is negligible compared to the cost of surface evolution.

3. EXAMPLES OF MESH RESTRUCTURING

For the applications presented in this article, the global maximum length scale L_0 in Eq. (5) is equal to the volume-equivalent particle radius. The initial mesh is derived from a regular mesh on an icosahedron by projection onto the prescribed initial particle shape [15]. In this section, we present two simple examples that illustrate the adaptivity of our mesh restructuring algorithm.

3.1. Static Irregular Surface

An adaptively structured mesh on a particle with an irregular prescribed shape is shown in Fig. 1. The restructured mesh was obtained using resolution constant $\alpha = 0.2$ in Eq. (3), which corresponds to 362 nodes on a spherical surface. The restructured mesh has 1312 nodes, as required for resolution of the high curvature regions, according to Eqs. (5) and (6).

Nodes with high coordination numbers form in the low- to high-curvature transition regions, as seen in Fig. 1b. The restructured mesh has two nodes with coordination number $N_c = 9$, and 35 nodes with $N_c = 8$; the remaining nodes have coordination numbers in the range 5–7.

3.2. Evolving Surface

A simple example of mesh restructuring on an evolving surface is shown in Fig. 2. The surface is initially spherical, and evolves according to (15) with velocity given by the linear

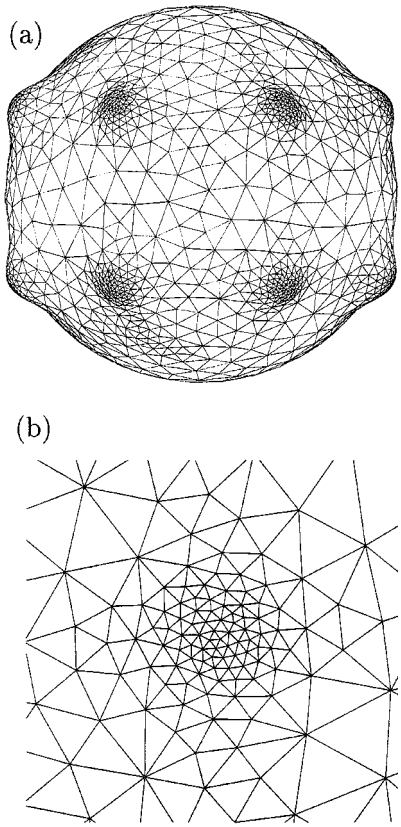

FIG. 1. Adaptively structured mesh on particle with prescribed shape (a); close-up of surface asperity (b). Mesh resolution $\alpha = 0.2$.

field

$$\mathbf{u} = \mathbf{E} \cdot \mathbf{x}, \tag{16}$$

where $\mathbf{E}$ is the velocity gradient corresponding to incompressible axisymmetric strain.

Uniform resolution of solid angle is maintained with resolution constant $\alpha = 0.3$, which corresponds to 162 nodes on the initially spherical surface, and 685 nodes on the final spheroidal configuration shown. Curvatures of 25 are attained at the tips. The tension profiles shown in Fig. 2b demonstrate that optimal local edge length (4) is approximately maintained.

4. DYNAMICS OF DROPS WITH DEFORMABLE INTERFACES

In this section, our adaptive mesh-restructuring algorithm is used in three-dimensional boundary integral simulations of deformable drops in Stokes flow. We consider neutrally buoyant drops with volume $\frac{4\pi}{3}a^3$ and viscosity $\lambda\mu$, in a continuous-phase fluid with viscosity μ; constant interfacial tension σ is assumed. Under these conditions, drop dynamics depends on the viscosity ratio λ, and the capillary number $\mathrm{Ca} = \mu a G/\sigma$, where G is the imposed strain rate. The behavior of drops in shear flow $\mathbf{u} = Gy\mathbf{e}_x$, and axisymmetric straining flow

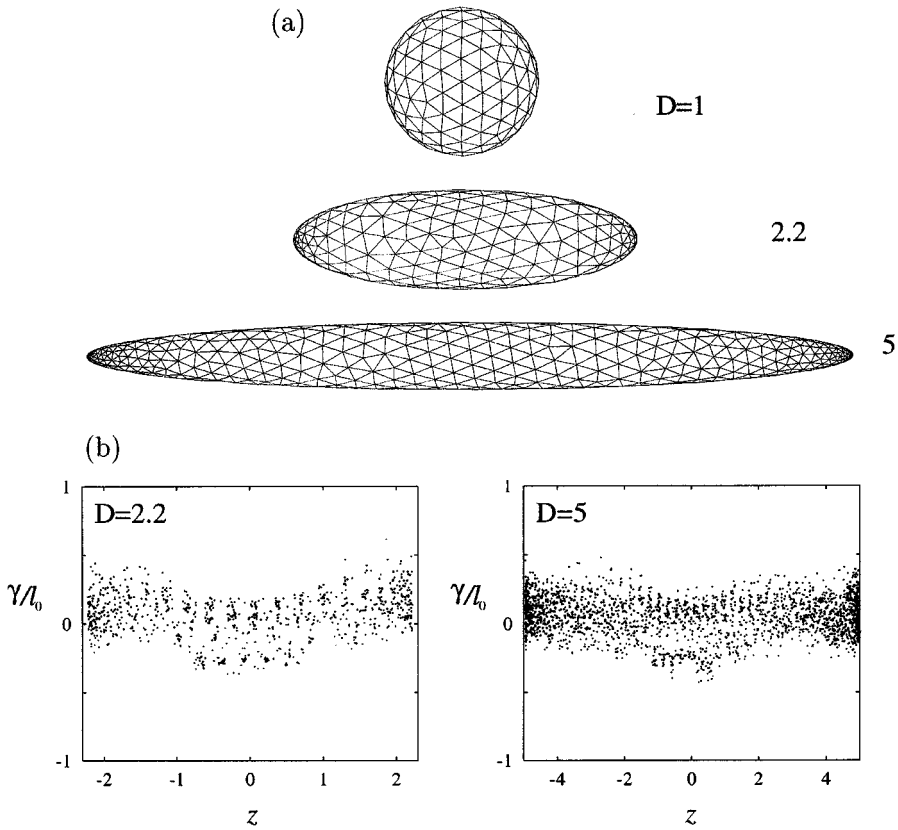

FIG. 2. Adaptive mesh with resolution $\alpha = 0.3$ on initially-spherical surface undergoing axisymmetric strain (a), major axis D as labeled; normalized tension profiles along the major axis (b).

$(u_r, u_z) = G(-\frac{1}{2}r, z)$ is considered. Henceforth, length is rescaled by the drop radius a, and time is rescaled by the capillary relaxation time $\mu a/\sigma$.

4.1. Evolution Equation

Interface evolution is described by Eq. (15) with node velocities obtained by solving a boundary-integral equation on the drop interfaces S [16, 17]

$$\mathbf{u}(\mathbf{x}) = \frac{2}{\lambda+1}\mathrm{Ca}\,\mathbf{u}^{\infty}(\mathbf{x}) - \frac{2}{\lambda+1}\int_S \mathbf{G}(\mathbf{x}'-\mathbf{x})\cdot\mathbf{f}(\mathbf{x}')\,\mathrm{d}^2\mathbf{x}' + \frac{\lambda-1}{\lambda+1}\int_S [\mathbf{T}(\mathbf{x}'-\mathbf{x})\cdot\mathbf{u}(\mathbf{x}')]\cdot\mathbf{n}(\mathbf{x}')\,\mathrm{d}^2\mathbf{x}'. \tag{17}$$

The surface traction jump,

$$\mathbf{f} = (\kappa_{\min} + \kappa_{\max})\mathbf{n}, \tag{18}$$

results from surface tension and is proportional to the mean curvature, $\mathbf{n}$ is the outward

normal vector, and $\mathbf{u}^{\infty}$ is the prescribed incident flow. The kernel functions **G**, **T** are the single-layer and double-layer Green functions for Stokes equations [17]. According to Eq. (17), node velocities on the interface are obtained without solving for the velocity elsewhere in the fluid.

4.1.1. *Numerical Implementation*

Equation (17) was solved by iteration [18] with eigensolutions corresponding to rigid-body motion and expansion purged [17]. The boundary integrals were evaluated by trapezoid-rule on the nodes of the mesh with singularity subtractions to regularize the integrands [15]. Curvatures and normal vectors were evaluated at nodes using a parabolic surface representation [14]. Second-order time stepping of (15) was used, with time step controlled by the capillary-stability criterion $\Delta t \ll \mu l_{\min}/\sigma$, where $l_{\min}$ is the global minimum edge length. The fluid velocity calculation has $O(N^2)$ computational cost, which dominates the $O(N)$ cost of mesh restructuring.

4.1.2. *Accuracy*

Node velocities obtained by the above procedure are $O(N^{-1})$ accurate [19]. Extrapolation with respect to the number of nodes was used to obtain $O(N^{-2})$ accuracy of the quantities shown in Figs. 7c and 9.

The accuracy of our three-dimensional simulations was verified by considering two test problems with axial symmetry (cf., Figs. 4 and 7c). The analytical azimuthal integration, allowed by this symmetry, reduces Eq. (17) to a one-dimensional integral equation [17], which can be solved with high precision. The required one-dimensional mesh was restructured, according to the procedure described in Appendix B.

4.2. Stationary Drop Shapes

Figure 3 depicts stationary drop shapes in shear flow for two viscosity ratios, under conditions that produce strongly distorted interfaces, but not breakup. These results were obtained from long-time simulations of drops with spherical initial shapes. The adaptivity

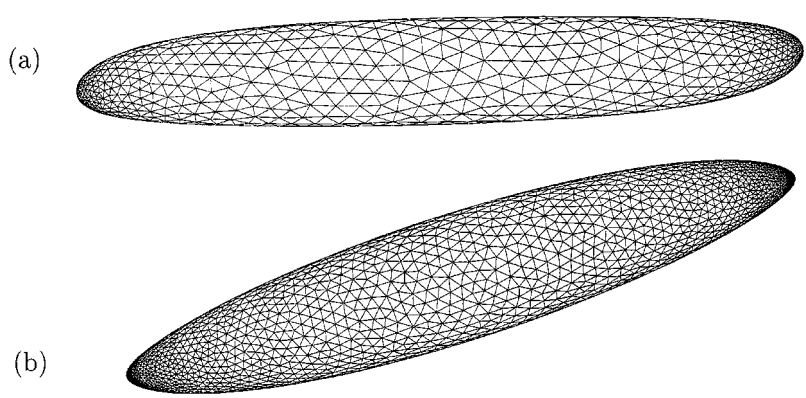

FIG. 3. Stationary drop shapes in shear flow; Ca = 1.43, $\lambda = 3$ (a), Ca = 0.8, $\lambda = 0.01$ (b). Velocity in horizontal direction, velocity gradient vertical.

of the mesh to the distorted drop interfaces is evident; high curvature regions ($\kappa \approx 20$ for $\lambda = 0.01$) are resolved.

4.3. Drop Breakup

The results depicted in Figs. 4 and 5 illustrate adaptive discretization on surfaces with a widely varying time-dependent length-scale distribution.

Figure 4 depicts the relaxation and breakup of a drop after elongation in an axisymmetric straining flow. The sequence of drop shapes obtained from three-dimensional simulations is compared to axisymmetric calculations at given values of time normalized by the breakup time. The breakup time predicted by three-dimensional simulations differs from the axisymmetric result by 1%.

An example of drop breakup in shear flow is shown in Fig. 5. Other examples, including a detailed comparison with an experimentally observed drop shape, have been published elsewhere [20].

Pinch-off. Three-dimensional adaptive-mesh simulations that extend beyond pinch-off events are feasible using the mesh-splicing algorithm described in Appendix C; an example is shown in Fig. 6. Typically, the neck region undergoes subsequent breakup yielding small

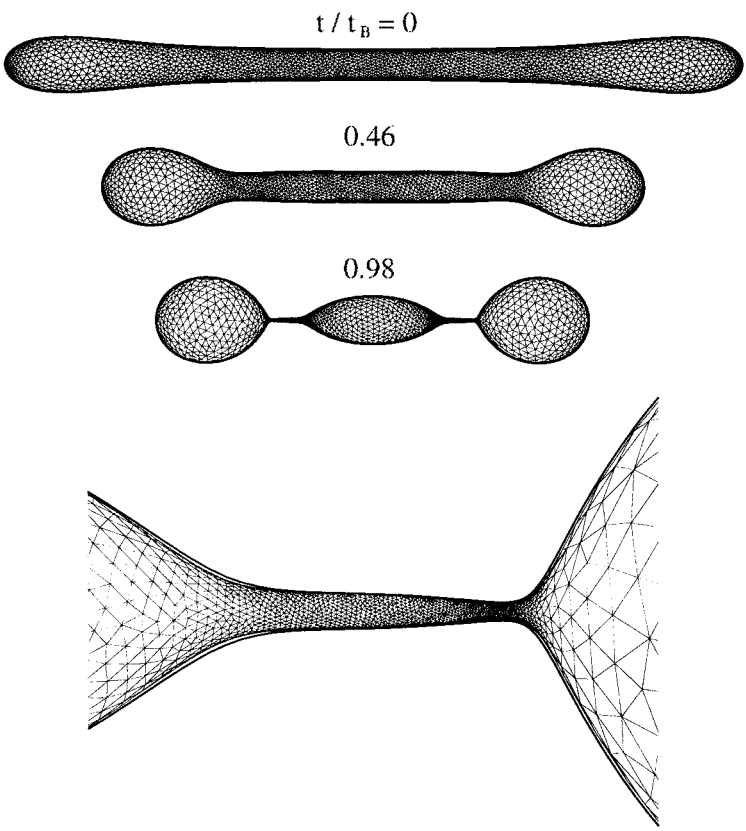

FIG. 4. Breakup of $\lambda = 1$ drop in a quiescent fluid after elongation in axisymmetric straining flow, Ca $= 0.13$; indicated times are normalized by breakup time. At $t = 0$, the flow stops, and the drop length is 11.2. The breakup time is $t_B = 24.5$. Three-dimensional (mesh) and axisymmetric (solid contour) simulations are shown. Inset shows neck region corresponding to the final configuration.

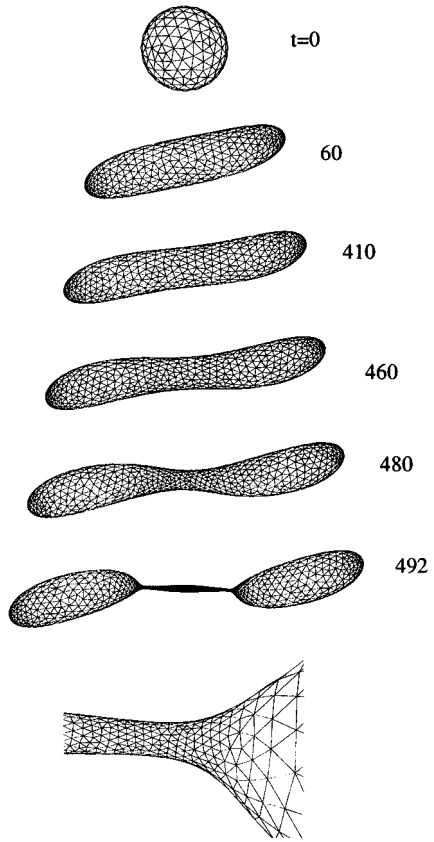

FIG. 5. Breakup of an initially spherical drop in shear flow Ca = 0.42, $\lambda = 1$; inset shows neck region at $t = 492$. Flow oriented as in Fig. 3.

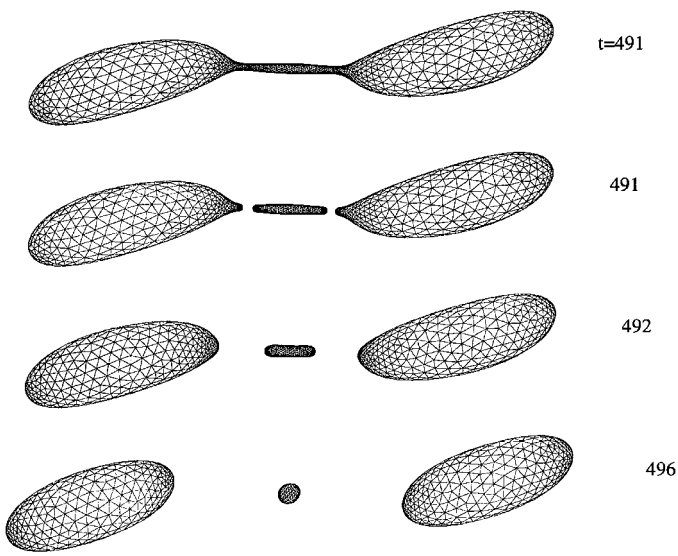

FIG. 6. Pinch-off of a drop in shear flow: same conditions as in Fig. 5.

satellite drops [21, 22], which are not resolved by our simulation. However, the evolution at larger length scales is insensitive to the details of the pinch-off event [23, 24].

4.4. Near-Contact Motion of Deformable Drops

Near-contact motion of deformable drops involves two local length scales (6) and (7), associated with curvature and gap-width variation. Application of our algorithm to such problems is illustrated in Figs. 7–9. The results were obtained using $c_2 = 0.5$ in Eq. (7).

The axisymmetric configuration of drops depicted in Fig. 7a was obtained by the long-time evolution of two initially spherical drops in compressional flow; the near-contact region is shown in Fig. 7b. The system was stabilized by constraining drop centers to the symmetry axis.

The long-time configuration of the drops in Figs. 7a and 7b is stationary because of the balance between pressure-driven film drainage, and the reverse pumping of fluid into the near-contact region by the externally driven recirculation within drops [25]. The gap-width history at the center of the near-contact region (Fig. 7c) obtained by our three-dimensional simulation agrees closely with the axisymmetric result; however, we were unable to resolve the minimum gap-width.

The evolution of a pair of drops in shear flow is illustrated in Figs. 8 and 9. The gap-width history (Fig. 8b) shows that the drops come into close contact gradually (less than

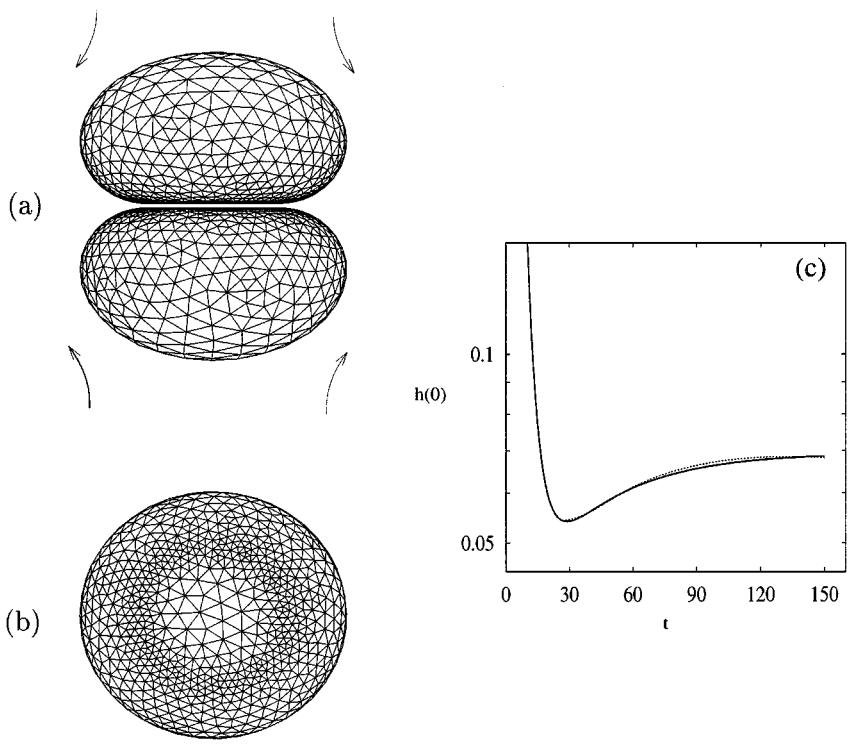

FIG. 7. Near-contact motion of drops in axisymmetric compressional flow Ca $= 0.1$, $\lambda = 1$. Side-view of drops (a), and near-contact region (b) at $t = 150$. Gap-width history at the center of the near-contact region (c); three-dimensional (solid curve) and axisymmetric (dashed curve) simulations are shown. Drops are initially spherical and centered at $(0, 0, \pm 1.6)$.

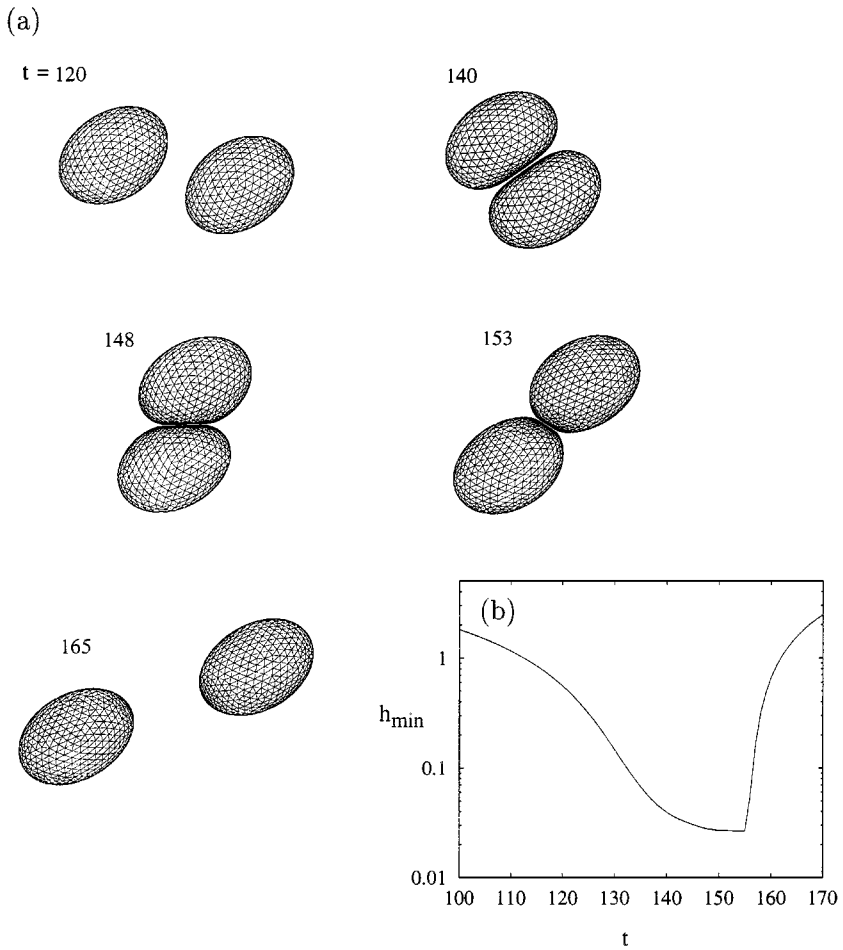

FIG. 8. Evolution of two drops in shear flow (a), gap-width history (b). Flow oriented as in Fig. 3; Ca = 0.135, $\lambda = 1.37$. Drops are initially spherical and centered at $(0, 0, 0)$ and $(-11, 0.51, 0)$.

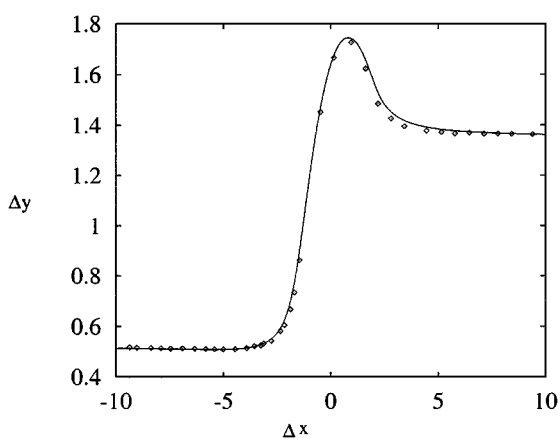

FIG. 9. Relative displacement of drops in velocity gradient direction; numerical simulation (solid curve), experimental measurements by Guido and Simeone [26] (points). Same conditions as in Fig. 8.

3% gap-width), and separate quickly. Our simulation agrees closely with the experimental measurements of Guido and Simeone [26] shown in Fig. 9.

5. CONCLUSIONS

An adaptive mesh-restructuring algorithm, based on the minimization of a mesh energy function, has been presented for simulations of evolving surfaces. Three-dimensional boundary-integral simulations of deformable drops were used to illustrate the adaptivity of meshes on surfaces with widely varying time-dependent length-scale distributions.

Our algorithm should be useful for solving a wide range of problems in computational fluid dynamics, including boundary-integral simulations of emulsions and foams, where the length scales associated with curvature, gap-width variation, and surface-tension gradients (Marangoni stresses) must be simultaneously resolved.

Numerical simulations of red-blood-cell motion in the microcirculation requires the resolution of length scales associated with the distribution of elastic shear and bending strains on the cell membrane [27–29]. During the evolution, highly curved regions form spontaneously and significantly contribute to the red cell dynamics [30]. By maintaining the resolution of surfaces with widely varying length scales, our algorithm makes feasible accurate numerical simulations of red cell motion in microvessels.

Our algorithm can be directly applied to two-dimensional finite-element simulations [31], and an extension to three-dimensional meshes should be possible. Adaptive mesh restructuring has applications in diverse fields, including medical imaging [32, 33], computer graphics [11, 12], and geographical information systems [34].

APPENDIX A: DETAILS OF IMPLEMENTATION

A.1. Mesh Restructuring

Certain details of mesh relaxation and topological transformations are described below.

A.1.1. *Mesh Relaxation*

Dynamic mesh relaxation was performed using second-order integration of Eq. (9) with step size $\Delta t = 0.25$; smaller time steps did not affect the results. The value of the equilibration parameter in Eq. (11) was set to $\epsilon = 0.05$.

Mesh relaxation was performed both globally, and locally on directly affected regions S_{loc} following each topological transformation.

A.1.2. *Topological Transformations*

Node addition. Three-node addition ($\Delta N = 3$) was implemented as shown in Fig. 10. Nodes are added on the surface at points closest to edge midpoints of a selected triangle. Low-coordination number nodes (i.e., $N_c = 3, 4$) are not produced by this procedure. As indicated in Fig. 10, the local averaging region S_{loc} includes all nodes that are first or second neighbors of the vertices of the selected triangle.

Node subtraction. Single-node subtraction ($\Delta N = 1$) was implemented, as shown in Fig. 11 for the case $N_c = 5$. Here, the local averaging region S_{loc} includes all nodes that are

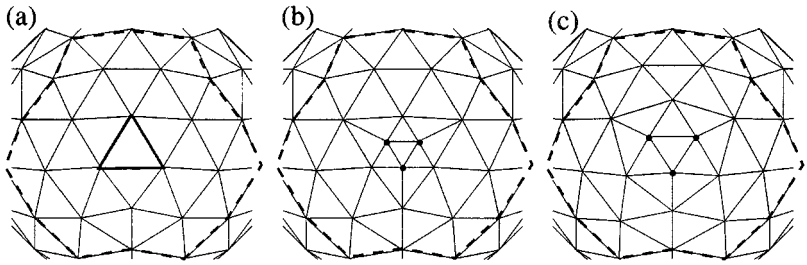

FIG. 10. Node addition procedure: initial mesh (a); after addition (b); after local relaxation (c). Triangle selected for node addition, added nodes, and S_{loc} (dashed contour) are indicated.

first or second neighbors of the vertices of the polygonal region surrounding the subtracted node. A similar approach was used to subtract nodes with $N_c = 6, 7$.

Node reconnection. The criterion for node reconnection is defined in Fig. 12. Accordingly, an edge connecting nodes 1 and 2 is favored over the connection between nodes 3 and 4 if

$$l < s_1 + s_2, \tag{A.1}$$

where l is the distance between nodes 1 and 2, and $s_i = |\mathbf{p}_i \cdot \hat{\mathbf{e}}|$ is the distance from node i to the center of the circle containing nodes $i, 3, 4$ projected along the unit vector $\hat{\mathbf{e}}$ between nodes 1 and 2.

Local node reconnection is performed only if l is small compared to the local radius of curvature, and the projections of the region 1, 2, 3, 4 onto the planes $1, 3, 4$ and $2, 3, 4$ are convex. For meshes with edge lengths smaller than the local radius of curvature, the foregoing procedure is equivalent to Delaunay triangulation.

The surface representation that we used [14] requires $N_c \geq 5$, which imposes an additional coordination-number constraint for node reconnection.

A.2. Local Averaging Operations

Below are the averaging operations for Eqs. (3) and (13), which we used in our simulations. Accordingly, we define an optimal triangle area

$$A_0 = \frac{3}{2}\left[\sum_{i=1}^{3} \overline{\rho}(i)\right]^{-1}, \tag{A.2}$$

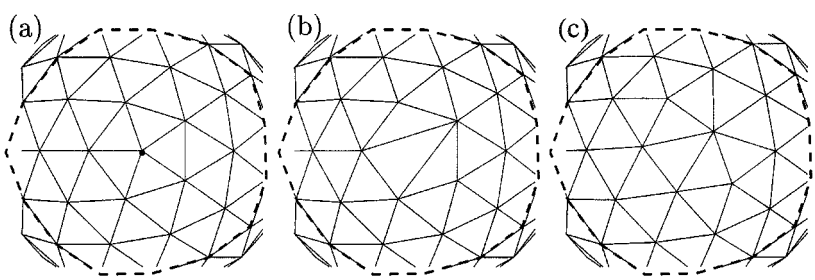

FIG. 11. Same as in Fig. 10, except for node subtraction, with subtracted node indicated.

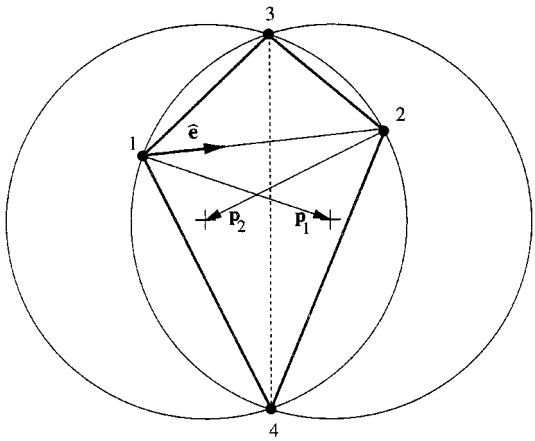

FIG. 12. Schematic for node reconnection procedure. Crosses denote centers of circles containing nodes 1, 3, 4 and 2, 3, 4.

where the mean node density is

$$\bar{\rho}(i) = \frac{1}{2}\left[\rho_0(i) + \frac{1}{N_c}\sum_{j=1}^{N_c}\rho_0(j)\right], \tag{A.3}$$

and ρ_0 is the optimal node density (14). In Eq. (A.2) the summation is over the vertices of the triangle, and in Eq. (A.3) the summation is over the nearest neighbors of node i.

The optimal length of an edge, defined by Eq. (3), is

$$l_0 = \frac{1}{2}\sum_{i=1}^{2} l_0'(i), \tag{A.4}$$

where the summation is over the two nodes joined by the edge, and

$$l_0'(i) = 2^{1/2} c_0^{1/2} \sum w(i) A_0(i)^{1/2} \tag{A.5}$$

is the optimal mesh length scale associated with a node i. In our simulations, $l_0'(i)$ was obtained by summing over the N_c neighboring triangles surrounding the node and the N_c next-neighbors; the weights assigned to the first and second neighbors were $w = \frac{5}{6}N_c^{-1}$ and $w = \frac{1}{6}N_c^{-1}$.

The optimal number of nodes in a local region, defined by Eq. (13), is

$$n_0 = \frac{n S_{\text{loc}}}{S_0}, \tag{A.6}$$

where

$$S_0 = \sum_k A_0(k), \tag{A.7}$$

is the sum of the optimal areas in S_{loc}, and n is the actual number of nodes in the region, with border nodes weighted by the fraction of surrounding triangles that lie within S_{loc}.

APPENDIX B: ONE-DIMENSIONAL MESH RESTRUCTURING

In this appendix, we describe an adaptive mesh-restructuring algorithm for two-dimensional and axisymmetric problems. A spline parameterized by arc-length s provides the local surface representation.

The appropriate node density, analogous to (14), is

$$\rho_0 = \alpha^{-1} L^{-1}. \tag{B.1}$$

A piecewise constant representation of the node density is defined, $\bar{\rho}_0(s) = \rho_0(s_i)$, where s_i is the arc-length of nearest node. The number of new nodes N' is obtained by integration of $\bar{\rho}_0$, with the resolution constant α adjusted to obtain an integer value. The positions of the new nodes s_i' $(i = 1, \ldots, N')$ are given by

$$1 = \int_{s_i}^{s_{i+1}} \bar{\rho}_0(s)\, ds, \tag{B.2}$$

starting from $s_1 = 0$. The local length-scale distribution, corresponding to (6), that maintains uniform resolution of angle is

$$L_1^{-1} = |\kappa|, \tag{B.3}$$

where κ is the line curvature. Near-contact lubrication stresses are resolved by the length scale (7).

APPENDIX C: MESH SPLICING

Here we present an extension of our adaptive discretization algorithm for mesh splicing, which involves the repositioning of selected nodes to target positions $\mathbf{x}'$ using

$$\dot{\mathbf{x}} = -\hat{\mathbf{P}} \cdot (\mathbf{x} - \mathbf{x}'), \tag{C.1}$$

in place of the mesh-relaxation equation (9).

The procedure starts with the identification of two sets of splicing nodes at a prescribed distance on either side of a location to be spliced (e.g., regions with most negative curvature), as illustrated in Fig. 13a. The two sets of splicing nodes are then repositioned to target points $\mathbf{x}'$ that lie on corresponding splicing planes, as shown in Fig. 13b. Repositioning is achieved by globally restructuring the mesh using (C.1) for the sets of splicing nodes.

Surface patches were generated from spherical caps that are approximately tangent to the interface along the (near-circular) contours formed by the projected splicing nodes. Surface patches were discretized by a procedure similar to that described in Section 3, using node addition and subtraction to achieve one-to-one correspondence between nodes at the edge of the patches and nodes on the splicing contours.Edge nodes and nodes on splicing contours

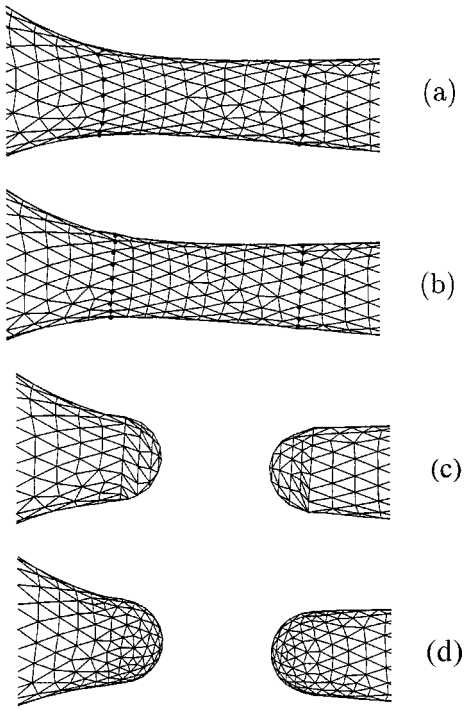

FIG. 13. Mesh splicing procedure used for simulation shown in Fig. 6. Points indicate sets of splicing nodes before (a) and after (b) repositioning. Surface patches before (c) and after (d) relaxation.

are then joined as shown in Fig. 13c. Dynamic relaxation (9) was used to achieve the mesh configuration depicted in Fig. 13d.

Fine-scale control of individual nodes afforded by our adaptive discretization algorithm can also be used to join the meshes of coalescing particles. Singularities, such as cusps and ridges, can be accommodated by controlling the local orientation of edges [34].

ACKNOWLEDGMENTS

This work was supported by a biomedical engineering grant from the Whitaker Foundation, and grants from NASA (NAG3-1935) and NSF (CTS-9624615).

REFERENCES

1. A. J. Stoddart, A. Hilton, and J. Illingworth, SLIME: A new deformable surface, in *Proceedings British Machine Vision Conference 1994*, pp. 285–294.
2. A. J. Bulpitt and N. D. Efford, An efficient 3d deformable model with a self-optimizing mesh, *Image Vision Comput.* **14**, 573 (1996).
3. R. C. Wilson and E. R. Hancock, Bias-variance analysis for controlling adaptive surface meshes, *Comput. Vision Image Understanding* **77**, 25 (2000).
4. D. J. Mavriplis, Unstructured grid techniques, *Ann. Rev. Fluid. Mech.* **29**, 473 (1997).
5. O. S. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).

6. S. Kwak and C. Pozrikidis, Adaptive triangulation of evolving, closed, or open surfaces by the advancing-front method, *J. Comput. Phys.* **145**, 61 (1998).
7. C. L. Lawson, Software for C1 surface interpolation, in *Mathematical Software III*, edited by J. R. Rice (Academic Press, New York, 1977), pp. 161–194.
8. D. T. Lee and B. J. Schacter, Two algorithms for constructing a Delaunay triangulation, *Int. J. Computer Information Sci.* **9**, 219 (1980).
9. L. De Floriani, Surface representation based on triangular grids, *The Visual Computer* **3**, 27 (1987).
10. J. M. Rallison, A numerical study of the deformation and burst of a viscous drop in general linear shear flows, *J. Fluid Mech.* **109**, 465 (1981).
11. F. J. M. Schmitt, B. A. Barsky, and W. H. Du, An adaptive subdivision method for surface fitting from sampled data, *SIGGRAPH* **20**, 176 (1986).
12. R. H. Bartels, J. C. Beatty, and B. A. Barsky, *An Introduction to the Use of Splines in Computer Graphics* (Morgan Kaufmann, Los Altos, CA, 1987).
13. A. Z. Zinchenko and R. H. Davis, An efficient algorithm for hydrodynamical interaction of many deformable drops, *J. Comput. Phys.* **157**, 1 (2000).
14. A. Z. Zinchenko, M. A. Rother, and R. H. Davis, A novel boundary-integral algorithm for viscous interaction of deformable drops, *Phys. Fluids A* **9**, 1493 (1997).
15. M. Loewenberg and E. J. Hinch, Numerical simulation of a concentrated emulsion in shear flows, *J. Fluid Mech.* **321**, 395 (1996).
16. J. M. Rallison and A. Acrivos, A numerical study of the deformation and burst of a drop in an extensional flow, *J. Fluid Mech.* **89**, 191 (1978).
17. C. Pozrikidis, *Boundary Integral and Singularity Methods for Linearized Viscous Flow* (Cambridge Univ. Press, Cambridge, UK, 1992).
18. Y. Saad and M. H. Schultz, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, 856 (1986).
19. M. Loewenberg and E. J. Hinch, Collision of deformable drops in shear-flow, *J. Fluid Mech.* **338**, 299 (1997).
20. V. Cristini, J. Bławzdziewicz, and M. Loewenberg, Drop breakup in three-dimensional viscous flows, *Phys. Fluids* **10**, 1781 (1998).
21. M. Tjahjadi, H. A. Stone, and J. M. Ottino, Satellite and subsatellite formation in capillary breakup, *J. Fluid Mech.* **243**, 297 (1992).
22. J. M. H. Janssen and H. E. H. Meijer, Droplet breakup mechanisms: Stepwise equilibrium versus transient dispersion, *J. Rheol.* **37**, 597 (1993).
23. J. Bławzdziewicz, V. Cristini, and M. Loewenberg, Analysis of drop breakup in creeping flows, *Bull. Am. Phys. Soc.* **42**, 2125 (1997).
24. J. R. Lister and H. A. Stone, Capillary breakup of a viscous thread surrounded by another viscous fluid, *Phys. Fluids* **10**, 2758 (1998).
25. D. Papadopoulos, M. Nemer, X. Chen, J. Blawzdziewicz, and M. Loewenberg, unpublished.
26. S. Guido and M. Simeone, Binary collision of drops in simple shear flow by computer-assisted video optical microscopy, *J. Fluid Mech.* **357**, 1 (1998).
27. T. W. Secomb, Mechanics of blood flow in the microcirculation, in *Biological Fluid Dynamics*, edited by C. P. Ellinton and T. J. Pedley (Company of Biologists, Cambridge, UK, 1995).
28. R. Skalak and N. Özkaya, Biofluid mechanics, *Ann. Rev. Fluid Mech.* **21**, 167 (1989).
29. E. A. Evans and R. Skalak, *Mechanics and Thermodynamics of Biomembranes* (CRC Press, Boca Raton, FL, 1980).
30. T. W. Secomb, R. Skalak, N. Özkaya, and J. F. Gross, Flow of axisymmetric red blood cells in narrow capillaries, *J. Fluid Mech.* **163**, 405 (1986).
31. T. J. R. Hughes, *The Finite-Element Method: Linear Static and Dynamic Finite Element Analysis* (Prentice-Hall Inc., Englewood Cliffs, NJ, 1987).

32. D. McInerney and D. Terzopoulos, A dynamic finite-element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image-analysis, *Comput. Medical Imaging Graphics* **19**, 69 (1995).
33. A. D. Kalvin, C. B. Cutting, B. Haddad, and M. E. Noz, Constructing topologically connected surfaces for the comprehensive analysis of 3D medical structures, *SPIE Image Process.* **1445**, 247 (1991).
34. L. Scarlatos and T. Pavlidis, Hierarchical triangulation using cartographic coherence, *CVGIP: Graphical Models Image Process.* **54**, 147 (1992).