An Adaptive Tetrahedral Mesh for Multiphase Domains with Evolving Interfaces

Finite-Element Simulations of Deformable Drops

R. Hooper, V. Cristini[1], J. Lowengrub[2], C. W. Macosko and J. J. Derby

Department of Chemical Engineering and Materials Science, University of Minnesota, 421 Washington Avenue S.E., Minneapolis, MN 55455

E-mail: hooper@cs.umn.edu; cristini@cems.umn.edu; lowengrb@math.umn.edu; macosko@umn.edu; derby@umn.edu

Received 2001

An adaptive mesh algorithm is presented, and applied to finite-element simulations of drop breakup and coalescence in viscous flow. In this algorithm, which has applications beyond the context, a three-dimensional multiphase domain enclosing interfaces is discretized by an unstructured mesh of tetrahedra constructed from the interfaces and other domain boundaries. All boundaries are discretized by triangulated surface meshes, where the triangular elements coincide with faces of the tetrahedra.

After each time step of simulation, the boundary mesh is reconstructed first, using an energy-minimization remeshing algorithm for triangulated surfaces based on loral restructuring operations, such as edge swapping, addition/subtraction of elements, and dynamical spring-like relaxation of node positions to an equilibrium configuration. The volume mesh is then generated from the boundary using an advancing-front/local-reconnection technique based on a node density function and Delaunay tetrahedra.

The resulting adaptive discretization maintains resolution of local length scales to a user-prescribed accuracy. Application of the algorithm is illustrated with finite-element simulations of deformable drops in Stokes flow. Steady drop shapes, the evolution of slender fluid filaments to breakup, and hydrodynamic interactions with boundaries and multiple drops are accurately described.

1. INTRODUCTION

The complex evolution of 3-dimensional geometries is encountered in a wide variety of physical and biological systems and is of critical importance in many in-

[1]Also at the School of Mathematics

[2]School of Mathematics, University of Minnesota, 206 Church Street, Minneapolis, MN 55455

dustrial problems. An important example is the morphology development of multiphase fluids in which strong flow in various geometries causes the interfaces to deform significantly. Finite-element methods (FEM) provide a natural choice to describe evolution of such systems using numerical simulations. They allow the solution of problems on irregularly shaped domains and can include such complex physical phenomena as inertia, thermal effects, species transport, surfactant effects and non-Newtonian behavior. Boundary-element methods have also been developed, although they are limited to a more restricted class of physical systems.

Application of FEM to problems on complex domains has led to the development of several powerful 3-dimensional mesh generation algorithms [1, 2, 3, 4]. Similar techniques have been applied to problems involving mesh distortion [5, 6]. However, little has been done for problems with very large interface deformations where the problem dynamics are dominated by the shape of the interface, e.g. large deformations involving capillarity. The main difficulty in performing such simulations is in accurately resolving very dissimilar length scales within the same domain at tolerable computational cost. Most moving boundary problems using FEM have employed mesh tracking schemes which require the finest mesh resolution to be maintained during the entire deformation and are therefore restricted to problems involving modest shape changes [7, 8, 9].

This difficulty has partially been overcome by an adaptive surface remeshing algorithm applied within the context of boundary integral simulations of multiphase Newtonian Stokes flow [10, 11] . For this type of problem, the boundary integral method requires only surface meshes to be discretized. Here, we combine the surface remeshing algorithm of Cristini [11] with efficient and robust 3-dimensional meshing provided by the commercial software package, Hypermesh, to produce an adaptive, 3-dimensional FEM. The ability of our algorithm to accurately describe significant drop deformations is demonstrated by simulations of deformable drop systems of Newtonian fluids in various flow scenarios. Advantage is taken of the mathematical form of the Newtonian Stokes equations to test the accuracy of the adaptive FEM algorithm without additional complicating effects arising from convective terms in the governing transport equations. The additional numerical framework needed to deal with such terms is reserved for the next chapter in which 3D drop deformations in viscoelastic fluids is addressed.

We begin by presenting the adaptive surface and volume remeshing algorithms in Section 2. The equations governing Newtonian drop deformations are summarized in Section 3 followed by a section describing the finite element method (FEM) used to approximate a solution to the governing equations. Results for which adaptive remeshing is performed after every time step are presented in Section 5 followed by our conclusions in the last section.

2. ADAPTIVE REMESHING ALGORITHM

Before describing the physical problem of Newtonian Stokes flow and the FEM used to approximate a solution, the adaptive remeshing algorithm is presented. In short, drop free surfaces are reconstructed after each time step using a mesh of triangles which maintains a specified resolution of surface curvature. This is accomplished using a surface remeshing algorithm developed by Cristini [11] for use in boundary integral simulations of Newtonian drop dynamics. The discretized

drop free surfaces are passed along with a triangulated surface mesh defining the outer domain boundaries, i.e. box, sphere, cylinder, extruder die, etc., to a commercial mesh generator which constructs an unstructured mesh of linear (4-node) tetrahedra. The triangulated drop free surfaces and outer boundary correspond to faces of adjacent tetrahedra. A brief description of the surface and domain meshing algorithms follows.

2.1. Surface remeshing

The algorithm used for adaptive reconstruction of triangulated drop surfaces meshes is described in detail in [11]. We provide a brief description of the details here.

The resolution of curvature of drop surfaces is kept uniform over the triangulated surface by adding elements in regions of high curvature and removing elements in lesser curved regions. A prescribed level of accuracy is maintained during a simulation by adaptively remeshing after each time step and utilizing a node density function

$$\rho = C_0 <\kappa>^2 \ , \tag{1}$$

where ρ is the surface node density on the drop surface(s), κ is a smooth measure of the local curvature of the surface, and C_0 is a constant that determines the resolution. Correspondingly, the instantaneous number of nodes, N_S, that discretizes the drop free surface is a function of the drop shape,

$$N_S = C_0 \int_S <\kappa>^2 \, \mathrm{d}S. \tag{2}$$

In practice, the constant, C_0 is set by specifying the number of nodes, N_0 used to discretize the initial shape (always spherical in this work) of a single drop. The required density (1) of nodes is maintained during a simulation through a sequence of local restructuring operations summarized next.

The surface mesh is modeled as a dynamic system of damped massless springs connecting the nodes: each spring has a tension $l - l_0$, where l is the instantaneous edge length and $l_0 \sim \rho^{-1/2}$ is the equilibrium length (to be approached iteratively). Equilibration velocities of the nodes are determined as the resultant of local spring tensions projected onto the surface. The system of springs has minimum-energy equilibrium states corresponding to zero equilibration velocities of all nodes, i.e. $l = l_0$. An equilibrium configuration is approached iteratively to within a specified tolerance by evolving the node positions with the equilibration velocities. Between equilibration steps, optimal node connectivity is maintained by local reconnections which adhere to a Delaunay criterion [12].

The triangulated surface(s) defining the outer fluid domain boundary are not remeshed but instead maintain the same discretization throughout the entire transient simulation.

2.2. Three-dimensional mesh generation

Since 3-dimensional unstructured mesh generation is a challenge worthy of a thesis in and of itself [5], the benefit of many man-years in this field is realized in

this work by resorting to a commercial mesh generation package, Hypermesh. A 3-dimensional mesh of unstructured tetrahedra is generated from the triangulated surfaces of the bounding region and the drop free surfaces. Node and connectivity information defining the surface meshes is passed to Hypermesh via an import translator created from C libraries supplied with the licensed software distribution. The algorithm used by Hypermesh is essentially a 3-dimensional analogue of the 2-dimensional surface meshing algorithm described in the previous subsection [13]. Nodes are initially placed in the domains (drops and outer fluid) using an advancing-front algorithm governed by a simple functional dependence, e.g. linear, exponential or Gaussian, to transition between surface meshes of differing resolution. New nodes are then inserted into the domain, and new elements are created by subdividing existing elements using the new nodes. Mesh quality is subsequently improved by local reconnection based on combined Delaunay and min-max criteria implemented iteratively. Node positions and element connectivities of both surface and domain meshes are exported using an output template into a form used by the FEM solver.

Because of proprietary issues related to the commercial software, the remeshing performed after each time step could not be done in core memory. The importing of surface meshes and exporting of full tetrahedra meshes had to be done using files. Nevertheless, the total time required for remeshing never exceeded more than 5% of the total computation time.

3. PROBLEM STATEMENT

The adaptive remeshing scheme defined above is applied to drop deformations in Newtonian systems. Specifically, nonbuoyant immiscible fluid drops of volume $\frac{4}{3}\pi {R_0}^3$ and viscosity $\hat{\mu}$ are considered to lie within a fluid of viscosity μ. The drop surfaces are surfactant-free, and no temperature gradients are present so that the interfacial tension, σ, is constant.

Flow conditions are assumed for which the Reynolds number based on the drop size is low, and thus the Stokes equations describe fluid motion, i.e.

$$\nabla \cdot \mathbf{T} = \mathbf{0} \tag{3}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{4}$$

All fluid phases are assumed to obey Newton's constitutive equation so that the total fluid stress is,

$$\mathbf{T} = -p\mathbf{I} + \lambda_i \left(\nabla \mathbf{v} + \nabla \mathbf{v}^T\right) \tag{5}$$

where p is the isotropic pressure, and $\mathbf{I}$ is the identity tensor. The above equations have been made dimensionless using characteristic time $\tau_{\dot{\gamma}} = \dot{\gamma}^{-1}$, length R_0 and stress $\mu\dot{\gamma}$ with $\dot{\gamma}$ being the characteristic rate of straining of the imposed flow. The coefficient λ_i is equal to 1.0 within the outer fluid and to the viscosity ratio $\lambda = \hat{\mu}/\mu$ within the drop phases. It follows [14] that the relevant parameters for drop deformation are the capillary number,

$$Ca = \tau_\sigma / \tau_{\dot{\gamma}}, \tag{6}$$

which is the ratio of capillary relaxation time $\tau_\sigma = \mu R_0/\sigma$ to flow time defined above, the viscosity ratio λ, and the geometric relationship of the drops to the outer

domain boundaries, i.e. geometric parameters reflecting hydrodynamic interactions with the bounding walls. Stationary deformed drop configurations exist in the flow below a critical $O(1)$ capillary number [14]. Above the critical capillary number, drops continuously elongate until breakup occurs.

The drops are subjected to flows described by

$$\mathbf{v}^{\infty} = \mathbf{D} \cdot \mathbf{x} \tag{7}$$

imposed at the outer boundary of the suspending fluid with $\mathbf{D}$ denoting the dimensionless velocity gradient and $\mathbf{x}$ denoting position. Particular types of flows which are relevant to previous studies of drop deformations and which are used in this work include uniaxial extensional flow with $D_{11} = 1.0$, $D_{22} = D_{33} = -0.5$ and all other $D_{ij} = 0$; and simple shear with $D_{12} = 1.0$ and all other $D_{ij} = 0$.

At the drop surfaces, both dynamic and kinematic boundary conditions apply and are given respectively as,

$$[\mathbf{n} \cdot \mathbf{T}]_{\Gamma} - \kappa \mathbf{n} = \mathbf{0} \tag{8}$$

$$\mathbf{n} \cdot (\dot{\mathbf{x}}_{\Gamma} - \mathbf{v}) = 0 \tag{9}$$

where $\mathbf{n}$ is the unit normal vector directed outward from drop surface Γ, $[\cdot]_{\Gamma}$ denotes the jump of the quantity between the brackets over the interface Γ in the direction $\mathbf{n}$, and $\dot{\mathbf{x}}_{\Gamma}$ denotes the velocity of nodes located on the drop surface(s).

Initial conditions are provided by specifying the initial drop shapes to be spherical and discretized by N_0 nodes each and the velocity fields to be zero within all fluid domains. At time, $t = 0$, the forcing flow of eq. (7) is imposed.

4. FINITE ELEMENT METHOD

The finite element method used in this work is derived from Zhou & Derby [15, 16]. A detailed description of their method applied to single-phase Newtonian particle sintering can be found in [15]. While differing by the inclusion of additional fluid phases, no assumptions of problem symmetry and the way in which mesh displacements are treated, this work is primarily distinguished from theirs by decoupling the solution of the equations governing fluid flow from the mesh movement. This accommodates the adaptive remeshing features described in Section 2 and allows for additional numerical techniques to be discussed in detail below.

4.1. Numerical Formulation

The FEM recasts the 'strong' form of momentum and mass conservation given by eqs. (3) and (4), respectively, into a 'weak' variational form amenable to discretization over irregular domains. In particular, a Galerkin formulation is obtained by weighting and integrating over the problem domain the strong form of the equations using test (weight) functions selected from a function space also used as a basis for approximating the problem unknowns (velocity $\mathbf{v}$ and pressure p in this work). The function space defines a basis for 3-dimensional linear tetrahedra [17]. Following the derivation of Hughes [18], a pressure-stabilized form can be obtained which allows the same basis to be used for velocities and pressure. Without resorting to a stabilized formulation, which in essence is a consistent penalty method, a mixed

basis with velocities approximated by higher order polynomials than used to approximate the pressure would be required in order to satisfy the 'LBB' or 'inf-sup' constraint [19]. Violation of this constraint leads to an over-specified problem which manifests spurious pressure modes (non- convergent node-to-node oscillations). The final steps involved in recasting the governing equations include integrating the momentum equation by parts, applying the divergence theorem, and substituting into the resulting surface integral the 'natural' boundary condition of eq. (8) [20, 17]. The resulting form of the recast momentum and continuity equations are given, respectively, as follows:

$$\begin{aligned} \mathbf{R}_{\mathrm{m}}(\mathbf{x}) &= \int_{\Omega} \left(-p\,\mathbf{I} + \lambda_i \left(\nabla\mathbf{v} + \nabla\mathbf{v}^T\right)\right) \cdot \nabla\phi(\mathbf{x},\mathbf{x}')\ \mathrm{d}\Omega \\ &+ Ca^{-1} \int_{\Gamma} \phi(\mathbf{x},\mathbf{x}')\kappa\mathbf{n}\ \mathrm{d}\Gamma = \mathbf{0}, \end{aligned} \tag{10}$$

$$R_{\mathrm{c}}(\mathbf{x}) = \int_{\Omega} \phi(\mathbf{x},\mathbf{x}')\nabla\cdot\mathbf{v}\ \mathrm{d}\Omega + \alpha \sum_j h_j^2 \int_j \nabla\phi(\mathbf{x},\mathbf{x}')\cdot\nabla p\ \mathrm{d}\Omega_j = 0. \tag{11}$$

In the above equations, $\mathbf{x}'$ is the integration variable, $\mathbf{v}$ and p are fluid velocity and pressure, ϕ is a linear tetrahedral basis function, and Ω represents the volume domains. Field variables consist of three velocity components of $\mathbf{v}$ and a scalar pressure field, p, which are discretized using the same continuous linear basis functions ϕ. As an example, pressure at a given location is

$$p(\mathbf{x}) = \sum_j p(\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j), \tag{12}$$

where the summation is over all nodes of the mesh, and the basis function ϕ is linear within the elements that share node j, and zero elsewhere. The domain integrals in equations (10)–(11) are evaluated over each element by Gaussian quadrature using four Gaussian integration points. This is done at the element level by mapping elements in the domain into a standard element and employing isoparametric mapping [17]. The latter implies that spatial positions within an element are interpolated in the same manner as the velocity and pressure degrees of freedom. The nodal values of $\mathbf{v}$ and p are the problem degrees of freedom which must be obtained at each time step.

For the Newtonian Stokes problem of eqs. (10)–(11) the discretization of field variables using the linear tetrahedral basis leads to a system of linear residual equations $\mathbf{R}(\mathbf{u}) = \mathbf{0}$ for all problem degrees of freedom $\mathbf{u}$ at nodes in the problem domain. To establish the framework needed for straightforward implementation of nonlinear physics such as shear thinning fluids or additional coupled transport (heat and/or mass) phenomena, the equation system was treated using a non-linear solution procedure. Newton-Raphson iterations were performed by solving at each iteration the following linear system of equations,

$$\mathbf{J}(\mathbf{u}^k)\boldsymbol{\delta}^{k+1} = -\mathbf{R}(\mathbf{u}^k) \tag{13}$$

where the Jacobian $\mathbf{J} = \frac{\partial\mathbf{R}}{\partial\mathbf{u}}$ and residual are evaluated using values $\mathbf{u}^k$ of variables at the current iterate k, and updates are obtained from the solution $\boldsymbol{\delta}^{k+1}$ as

$\mathbf{u}^{k+1} = \mathbf{u}^k + \boldsymbol{\delta}^{k+1}$. The linear system of equations is also solved iteratively using a Generalized Minimum Residual (GMRES) technique combined with diagonal preconditioning [21, 22]. Specifically, a variation of "Type 3" diagonal preconditioning described in [22] is used and involves solving the following modified linear system,

$$\left(\mathbf{J}\mathbf{D}^{-1/2}\right)\left(\mathbf{D}^{1/2}\boldsymbol{\delta}\right) = -\mathbf{R}\,. \tag{14}$$

The matrix $\mathbf{D}$ has entries correspnding to $D_{ii}^{1/2} = \sqrt{J_{ii} + \epsilon}$ with ϵ a small parameter taken to be 10^{-6} in this work. This solution methodology has been used with great success by Zhou & Derby for the nonlinear Newtonian Stokes problem involving coupled mesh motion.

4.2. Dynamic Boundary Condition

The boundary integral accounting for capillarity in eq. (10) must be handled carefully. The dimensionless mean surface curvature κ involves second-order spatial derivatives which are identically zero when performed on the linear basis used to describe position, i.e. isoparametric mapping. Two ways of dealing with this issue are proposed.

The first way of dealing with second derivatives associated with surface curvature is to avoid them altogether by integrating the surface integral by parts [23, 24]. This has the effect of transferring a derivative operation from position to the weighting function and yields the following replacement,

$$Ca^{-1}\int_\Gamma \phi(\mathbf{x},\mathbf{x}')\kappa\mathbf{n}\;\mathrm{d}\Gamma = Ca^{-1}\int_\Gamma (\mathbf{I}-\mathbf{nn})\cdot\nabla\phi(\mathbf{x},\mathbf{x}')\;\mathrm{d}\Gamma \tag{15}$$

This technique can be applied at the element level using Gaussian quadrature and was used by Zhou & Derby in conjunction with implicit time-integration and Newton's method to solve the flow equations fully coupled to mesh motion.

An alternative approach is to depart from isoparametric mappings for position. Instead, surface features such as curvature and surface normal vector can be obtained from an analytic representation of the surface in a region local to the point of interest and used directly in the surface integral. A local representation of the surface by a least squares fit to a paraboloid was used successfully by Zinchenko et al. [25] in boundary integral simulations and has been adopted in this work. The fitting procedure is node-based so that a node-based numerical evaluation of the surface integral is preferred over the element-based Gaussian quadrature used to evaluate the weak form of the surface integral. As described in [25],

$$\int_\Gamma \phi(\mathbf{x},\mathbf{x}')\kappa\mathbf{n}\;\mathrm{d}\Gamma \;\approx\; \sum_i \kappa^i\mathbf{n}^i\Delta\Gamma^i \tag{16}$$

$$\Delta\Gamma^i \;=\; \frac{1}{3}\sum_{ie}\Delta\Gamma \tag{17}$$

where the first summation is over all nodes, i, on the drop surfaces, and the second summation is over all drop surface elements ie having node i as a vertex. The basis functions $\phi(\mathbf{x},\mathbf{x}')$ are unity at nodes. Because the fitting procedure is iterative and based on neighboring nodes encompassing several triangular surface elements,

this approach would be extremely difficult, if not impossible, to implement in an implicit time-integration scheme involving Newton-like methods. However, it can be readily implemented in the explicit, decoupled framework of this work. In contrast to under-predicting capillary effects using the weak form of the surface integral, this technique is seen to over-predict capillarity as described in greater detail in the results section.

4.3. Time Integration

The benefit of adaptivity comes at the expense of resorting to an explicit time integration scheme in which solution of the momentum and mass balance equations is decoupled from movement of the mesh. To develop this approach, we define a vector $\mathbf{y}$ to represent degrees of freedom of both the Stokes problem as well as displacements $\mathbf{d}$ of nodes in the mesh, i.e. $\mathbf{y} \equiv [\mathbf{v}, p, \mathbf{d}]^T$. The equations of Section 4.1 can then be expressed compactly as,

$$\mathbf{M}\dot{\mathbf{y}} = \mathbf{F}(\mathbf{y}) \tag{18}$$

where $\mathbf{M}$ is the mass matrix containing the coefficients of time derivatives, $\dot{\mathbf{y}}$, of the degrees of freedom, and $\mathbf{F}$ consists of all terms not containing explicit time derivatives. Noting the absence of explicit time derivatives in the Stokes equations of eqs. (10)–(11), eq. (18) has the following structure:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \cdots & \cdots & \cdots \\ 0 & 0 & M_{dd} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}} \\ \dot{p} \\ \\ \dot{\mathbf{d}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_m \\ R_c \\ \cdots \\ \mathbf{F}_d(\mathbf{v}) \end{bmatrix} \tag{19}$$

where $\mathbf{R}_m$ and R_c are given by eqs. (10) and (11), respectively, and M_{dd} and $\mathbf{F}_d$ are left unspecified until the next section. The decoupling of mesh movement from solution of the field variables $(\mathbf{v}, p)$ which makes adaptive remeshing possible is indicated by the partitioning in eq. (19). The decoupled problem can be rewritten as

$$\begin{bmatrix} \mathbf{0} \\ M_{dd} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{y}}_1 \\ \dot{\mathbf{d}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_d \end{bmatrix} \tag{20}$$

with definitions of $\mathbf{y}_1$ and $\mathbf{F}_1$ inferred from eq. (19).

Equation 20 can be viewed as two distinct steps for the overall problem. The first step defined by $\mathbf{F}_1 = \mathbf{0}$ involves solving for Stokes flow on a fixed mesh and is followed by the second step, $M_{dd}\dot{\mathbf{d}} = \mathbf{F}_d$, which consists of an update of the mesh based on the previously computed velocity. A time stepping algorithm is only required for the second step of the overall problem.

The simplest means of stepping the solution for nodal displacements in time would be to use the explicit Euler scheme which would take the following form:

$$M_{dd}\left(\frac{\mathbf{d}_{t+\Delta t} - \mathbf{d}_t}{\Delta t}\right) = \mathbf{F}_d\left(\mathbf{v}_t\right). \tag{21}$$

However, this method is the least accurate, the most restrictive on time step size for stability and has been found to perform poorly for this class of problems solved

using the boundary integral method [10, 11]. Several choices exist to improve both accuracy and stability of explicit time stepping. The methods are generally classified as multi-step, which require solutions at two or more previous time steps, and single-step predictor-corrector methods which require information for only the current time step interval (explicit Euler being the simplest single-step method). Due to economy and to the success experienced in boundary integral simulations, a second-order accurate single-step method, second-order Runge-Kutta (RK2), is used in this work. Updates to node positions are then computed as follows:

$$M_{dd}\left(\frac{\mathbf{d}_{t+\Delta t}-\mathbf{d}_t}{\Delta t}\right)=\mathbf{F}_d\left(\mathbf{v}_{t+\Delta t/2}\right), \tag{22}$$

where the only difference from eq. (21) is in the velocity used to determine $\mathbf{F}_d$. Whereas explicit Euler uses the velocity $\mathbf{v}_t$ at the beginning of the time step, RK2 uses a prediction for velocity $\mathbf{v}_{t+\Delta t/2}$ at the midpoint of the time step Δt. The method achieves second order temporal accuracy at the expense of a two-stage process involving a predictor for variables at $\Delta t/2$ followed by a corrector update of variables to Δt. The complete algorithm including remeshing is summarized in Table 1.

The time step Δt indicated in the algorithm is varied based on the smallest length scale in the mesh at the current time. This implies a convective stability constraint based solely on physics dominated by competition between viscous and capillary forces and is in contrast to the unconditionally stable implicit time integration scheme employed in prior drop deformation studies of Zhou & Derby. The next section provides specific details for mesh evolution. remeshing, respectively.

4.4. Mesh evolution

For the Newtonian Stokes problem of this work, no explicit time derivatives appear in the governing equations, cf eqs. (3) and (4). Instead, all time derivatives arise from the kinematic boundary condition of eq. (9) which serves to move nodes on the drop surface(s) in a physically consistent manner. Accordingly, the explicit update step for these nodes with positions $\mathbf{x}_\Gamma$ is accomplished using,

$$\frac{\mathrm{d}\mathbf{x}_\Gamma}{\mathrm{d}t}=\mathbf{nn}\cdot\mathbf{v}(\mathbf{x}_\Gamma). \tag{23}$$

The velocity field, $\mathbf{v}$, used to move the drop surface nodes is found by solving the reformulated (weak form) momentum and mass equations on a fixed mesh represented by nodes at $\mathbf{x}(t)$ for the predictor step and $\mathbf{x}(t+\Delta t/2)$ for the corrector step. Whereas $\mathbf{x}(t)$ is obtained directly from a 3D remeshing step (see Section 2.2), values of $\mathbf{x}(t+\Delta t/2)$ not located on the drop surface must be determined by some consistent procedure.

In implicit time integration schemes with mesh displacements coupled to solution of the velocity and pressure fields, nodal displacements are taken care of automatically [26, 27, 28, 15, 16]. However, decoupling the mesh movement via explicit time integration places the burden of devising a consistent scheme on the modeler. A consistent scheme is one that moves nodes not on a kinematic boundary in such a way that element distortions are minimized, element volumes are prevented from collapsing to zero or changing sign and distortions of the bounding outer surface

TABLE 1
Second-order Runge-Kutta time integration algorithm for Newtonian

Stokes flows

Given: Ω_0,

For each time step Δt *until* $t \geq t_{max}$*:*

PREDICTOR

Find $\mathbf{y}_{1,t} = (\mathbf{v}, p)_t$ *:*

$$\mathbf{F}_1 \left(\mathbf{y}_{1,t}, \mathbf{d}_t \right) = \mathbf{0}$$

Find $\Omega_t \overset{\mathbf{d}_{t+\Delta t/2}}{\longrightarrow} \Omega_{t+\Delta t/2}$*:*

$$M_{dd} \left(\frac{\mathbf{d}_{t+\Delta t/2} - \mathbf{d}_t}{\Delta t/2} \right) = \mathbf{F}_d \left(\mathbf{v}_t \right)$$

CORRECTOR

Find $\mathbf{y}_{1,t+\Delta t/2} = (\mathbf{v}, p)_{t+\Delta t/2}$ *:*

$$\mathbf{F}_1 \left(\mathbf{y}_{1,t+\Delta t/2}, \mathbf{d}_{t+\Delta t/2} \right) = \mathbf{0}$$

Find $\Omega_t \overset{\mathbf{d}_{t+\Delta t}}{\longrightarrow} \Omega_{t+\Delta t}$*:*

$$M_{dd} \left(\frac{\mathbf{d}_{t+\Delta t} - \mathbf{d}_t}{\Delta t} \right) = \mathbf{F}_d \left(\mathbf{v}_{t+\Delta t/2} \right)$$

Update time: $t = t + \Delta t$

Remesh:

$$\Omega_t^{old} \mapsto \Omega_t^{new}$$

mesh are either prevented or controlled. The latter condition is needed to allow correct implementation of the imposed flow via eq. (7). One possibility for a consistent explicit scheme is to adapt an implicit scheme such as the pseudo-solid mesh treatment used by Zhou & Derby [15]. The primary drawback to such an approach is the additional computational effort needed to solve the (decoupled) mesh displacement equations each time the mesh is displaced (twice per time step using RK2). We have chosen to employ a much cheaper but more ad hoc scheme based on spines and algebraic weighting of nodal displacements.

The scheme we use in this study for single drop systems can be summarized as follows:

- For a node not on a drop (kinematic) surface, determine whether it lies inside or outside a drop phase.
- For nodes in the outer fluid phase:

  – Find which drop surface element is intersected by the position vector connecting the center of the drop to the node.

  – Interpolate a normal velocity on the drop surface at the point of intersection using the three normal velocities at the triangle vertices.

  – Determine a relative displacement velocity by dividing the distance from the point of intersection to the node by the distance from the point of intersection to the point at which the position vector intersects the outer boundary surface and multiplying this ratio by the interpolated normal velocity at the point of intersection on the drop surface.

- For nodes inside a drop phase:

  – Find the nearest drop surface element to the node.

  – Find the location within the surface element which is intersected by the position vector connecting the center of the drop to the node.

  – Interpolate a normal velocity on the drop surface at the point of intersection using the three normal velocities at the triangle vertices.

  – Determine a relative displacement velocity by dividing the distance from the drop center to the node by the distance from the drop center to the point of intersection and multiplying this ratio by the interpolated normal velocity at the point of intersection on the drop surface.

Expressions in the notation of Section 4.3 are as follows:

$$M_{dd} = \mathbf{I} \tag{24}$$

$$\tag{25}$$

$$\mathbf{F}_d = \begin{cases} \mathbf{nn} \cdot \mathbf{v} & , \ \mathbf{x} \in S \\ \mathbf{0} & , \ \mathbf{x} \in S_{outer} \\ \mathbf{v}_{rel} & , \ \mathbf{x} \notin S, S_{outer} \end{cases} \tag{26}$$

with $\mathbf{v}_{rel}$ the relative displacement velocity found using the interpolation scheme just described.

For multiple drop simulations, we again take advantage of the absence of time derivatives in the Newtonian Stokes equations and simply move all nodes with the velocity field, thereby moving the mesh in a purely Lagrangian way, i.e. $\mathbf{F}_d = \mathbf{v}, \forall \mathbf{x}$. Resulting mesh distortions are larger than for the ad hoc scheme but are prevented from becoming excessive by remeshing after every time step. A purely Lagrangian movement of the mesh could also be used with single drops, but the ad hoc method produces less element distortion and holds some potential for permitting multiple mesh tracking steps to be taken between remeshing.

Both schemes just described provide nodal velocities. Actual displacements are obtained by multiplying the nodal velocities by an appropriate time step. In our implementation, a variable time step $\Delta t = \epsilon l_{\min}$ is used and set by requiring

$$\Delta \mathbf{x} < l_{\min} \ , \tag{27}$$

where $l_{\min}$ is the shortest node-to-node distance over the mesh. The requirement is met by choosing $\epsilon = 0.5Ca$ for $Ca \leq O(1)$ and $\epsilon = 0.5$ for $Ca \gg O(1)$.

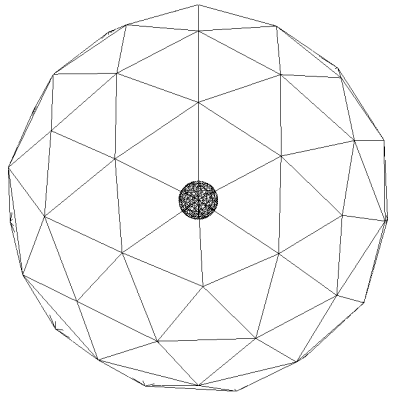
5. RESULTS OF SIMULATIONS

Finite element simulations were performed on single CPU-nodes of an IBM-SP at the Minnesota Supercomputer Institute. A CPU-node consists of four 222 MHz Power3 processors sharing 4 Gb memory. The FEM was implemented in parallel on all four processors using Open-MP. The level of mesh refinement varied during the course of the simulation, becoming greater during later stages as maximum curvature increased. A typical simulation required on the order of 1000 time steps with the last steps requiring about 30 CPU minutes each to complete. Total time required was about 48 CPU hours.

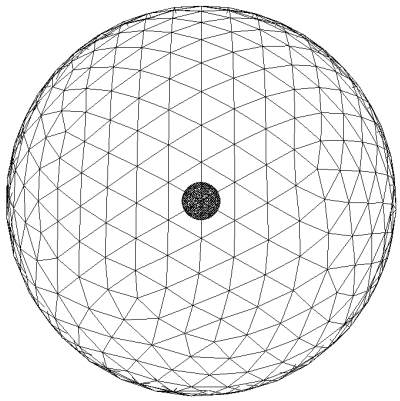
5.1. Subcritical axisymmetric deformation

In this section, we explore the converge characteristics of the adaptive FEM by simulating single drop deformation in a start-up uniaxial extensional flow imposed using eq. (7). This allows comparison of 3D results to highly converged 2D axisymmetric results obtained with the method described by Hooper et al. [28]. Mesh refinement studies are performed by combining initial surface meshes of prescribed resolution for both the drop surface and the outer spherical domain boundary. An example of three such combinations and the corresponding 3-dimensional tetrahedral mesh corresponding to each is given in table 2. The table reports the number of surface nodes $N_{0,outer}$ initially used to discretize the outer domain surface, the initial surface nodes $N_{0,drop}$ on the drop surface, and the coresponding total number of nodes $N_{0,3D}$ initially in the complete 3-dimensional computational domain. The surface mesh pairs are also shown in Figure 1. The spherical outer boundary shape with location at $10R_0$ conforms to that used in the 2D axisymmetric simulations.

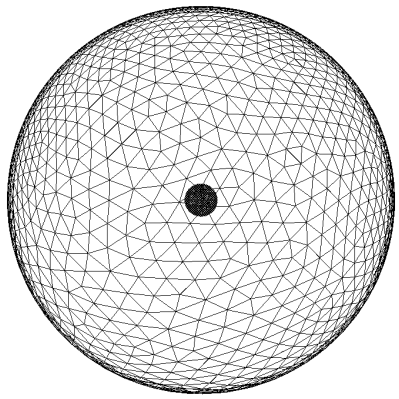
As a first test for convergence, the outer spherical mesh of Mesh 2 is used with four different initial drop surface discretizations comprising $N_{0,drop} = 200, 400, 600$ and 800. Transient drop evolution is shown in figure 2 as a plot of drop length in the direction of extension normalized by initial length $2R_0$ versus time made dimensionless using flow time, $\dot{\gamma}^{-1}$. Line curves are the actual simulation results,



Mesh 1



Mesh 2



Mesh 3

FIG. 1. Initial surface meshes used to construct 3-dimensional unstructured tetrahedral meshes used in convergence tests for uniaxial extensional flows.

TABLE 2
3D meshes used to test convergence in uniaxial extensional flow

Mesh	$N_{0,outer}$	$N_{0,drop}$	$N_{0,3D}$
Mesh 1	59	200	3439
Mesh 2	507	400	6802
Mesh 3	2007	800	11,263

and symbols are values at times corresponding to the unequally spaced time steps of the 2D FEM result. For the 3D adaptive FEM results, values of drop length at these times were found by first fitting each curve to a cubic spline followed by interpolation of the spline data. These points are needed for the convergence analysis described later. Flow conditions were chosen to be fairly severe in order to test the robustness of the 3D adaptive method. We select $\lambda = 0.1$ and $Ca = 0.17$ which corresponds to a subcritical flow for which a stable drop shape is achieved. As the steady state is approached, the viscous stress from the outer fluid must be balanced by those inside the drop along with capillarity. With the drop being an order of magnitude less viscous, larger drop surface curvature is required to establish this balance, and the drop ends exhibit more pronounced curvature as a result. The inset of figure 2 shows a portion of the surface mesh for the final time step simulated using the finest drop surface mesh having $N_{0,drop} = 800$. The maximum mean surface curvature is 3.23 and is well-resolved by the adaptive meshing algorithm which has increased the number of surface nodes from $N_0 = 800$ used to discretize the initially spherical drop surface to $N_S = 1231$. We tested both methods of section 4.2 to account for capillarity. The weak form implementation tended to underpredict capillary effects as evidenced by the inability to provide a stable shape for any of the mesh refinements in table 2. The curves in figure 2 are results obtained using the local paraboloid fit to determine surface normal vectors and curvature used in eq. (10). These clearly indicate an over-prediction of capillarity with less mesh resolution.

The points represented by symbols in figure 2 are now used to ascertain an approximate order of spatial convergence for the 3D adaptive FEM. A convergence of $O(N_0^{-1})$ might be expected based on surface curvature being accurate to $O(\Delta x^2)$ since $\Delta x \sim N_o^{-1}$ with Δx being the minimum surface triangle edge length. Using the values in figure 2 at time $t = 3.25$ the error defined as $E \equiv \left(\frac{l}{2R_0}\right)_{2D} - \left(\frac{l}{2R_0}\right)_{3D}$ is plotted as a function of N_0^{-1} in figure 3. An extrapolation to infinite mesh refinement $N_0 \to \infty$ shows the 3D FEM prediction differs from the accurate 2D FEM result by 0.6%. Performing the extrapolation for each 2D FEM time step, represented by symbols in figure 2, leads to the comparison shown in figure 4. An empirical order of converge was determined by minimizing the error E. The

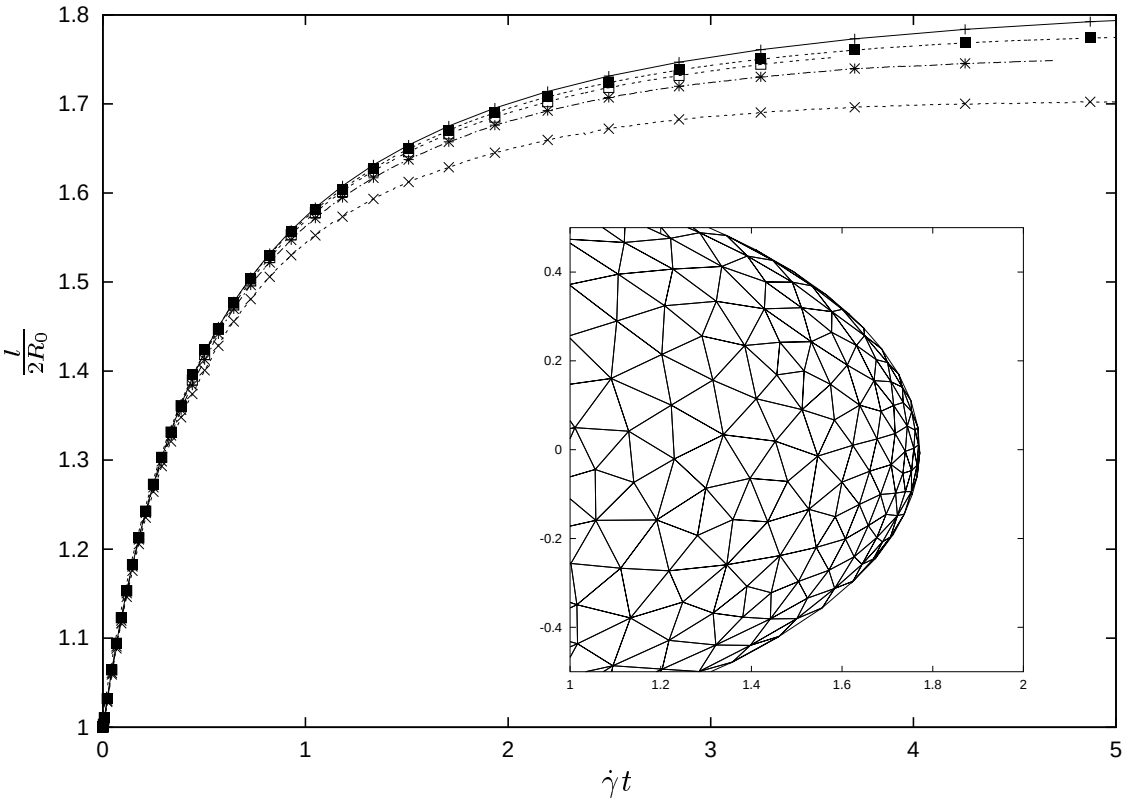


FIG. 2. Drop length $l/2R_0$ as a function of time $\dot{\gamma}t$ during evolution in uniaxial extensional flow; Ca=0.17, $\lambda = 0.1$. Converged 2D axisymmetric finite element results (solid curve); Adaptive 3D finite element results obtained using the outer domain surface mesh for Mesh 2 in Table 2 with varying levels of drop surface mesh refinement: $N_{0,drop} = 200$ (medium dashed curve with x), $N_{0,drop} = 400$ (dash-dotted curve with *), $N_{0,drop} = 600$ (small dotted curve with open squares), $N_{0,drop} = 800$ (small dotted curve with filled squares). Symbols are at time steps of 2D FEM simulation and are interpolated from cubic splines for 3D FEM simulations.

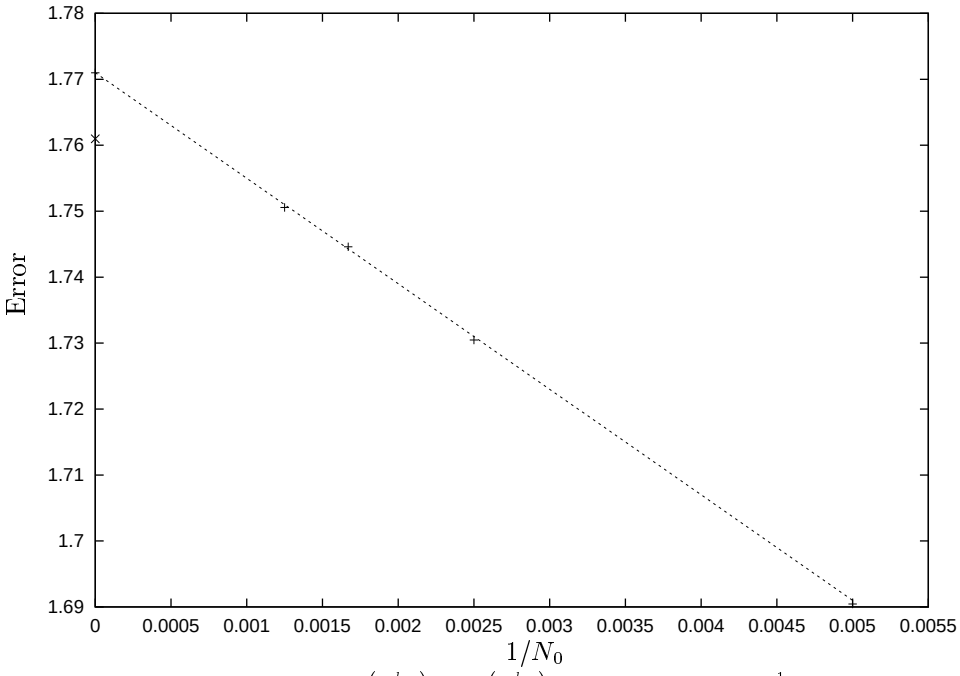


FIG. 3. Error defined as $E \equiv \left(\frac{l}{2R_0}\right)_{2D} - \left(\frac{l}{2R_0}\right)_{3D}$ plotted against N_0^{-1} for time $t = 3.25$ using data of figure 2. Symbol x at $N_0^{-1} = 0$ corresponds to 2D FEM result.

resulting order of convergence of $O(N_0^{-1.4}) \equiv O(\Delta x^{2.8})$ is also shown in figure 4. The error for both orders of convergence are depicted in figure 5.

Whereas the previous convergence test refined only the drop surface mesh while keeping the outer domain surface mesh constant, a subsequent convergence test is now performed in which both the drop and outer domain surface meshes are refined simultaneously. Hence, the meshes in table 1 are used exactly. Figure 6 shows accurate 2D FEM results and 3D FEM results for each of the three mesh refinement levels for uniaxial extensional flow with $Ca = 0.10$ and $lambda = 1.0$. This flow is subcritical with the drop attaining a stable shape but is less severe than previously considered in that the extent of drop deformation is less. As before, in figure 6 line curves represent actual results, whereas symbols reflect interpolations from cubic spline fits for time steps corresponding to the 2D FEM result. Extrapolations to infinite 3D mesh refinement are again performed assuming the error E depends on N_0^{-1} and $N_0^{-1.5}$ with the latter dependence determined empirically. Extrapolated 3D FEM results are shown along with 2D FEM results in figure 7, and errors at each 2D FEM time step for both scalings are shown in figure 8. The empirically determined dependence suggests an order of spatial convergence of the 3D adaptive FEM of $O(\Delta x^3)$.

The order of spatial convergence found here and for the flow scenario considered previously should be considered approximate estimates since many aspects of the adaptive method prevent a clear scaling of the error. For example, use of linear tetrahedra to solve the fluid flow equations with an equal-order interpolation of all field variables would be expected to produce spatial convergence of $\approx O(\Delta x)$ with the length scale now being the minimum within the 3D domain. However,

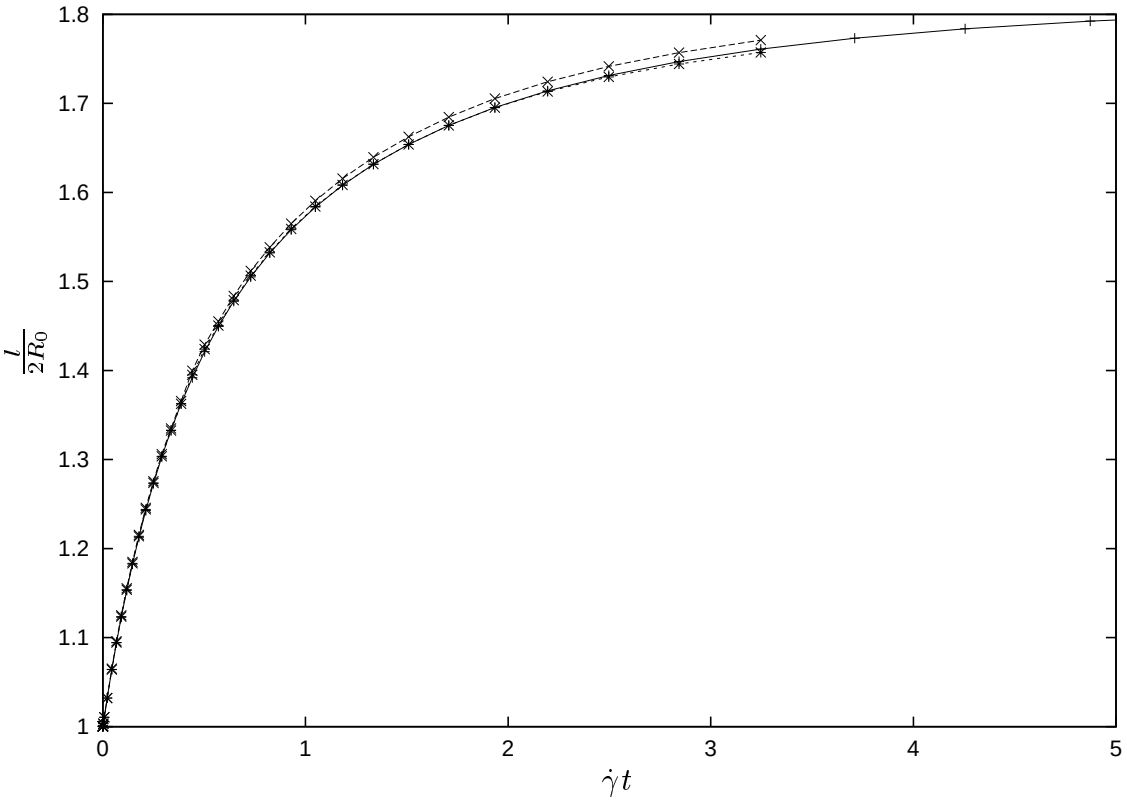


FIG. 4. Drop length $l/2R_0$ as a function of time $\dot{\gamma}t$ during evolution in uniaxial extensional flow; Ca=0.17, $\lambda = 0.1$. Converged 2D axisymmetric finite element results (solid curve with +); Adaptive 3D finite element results extrapolated to $N_0 \to \infty$ using $E \sim N_0^{-1}$ scaling (medium dashed curve with x), Adaptive 3D finite element results extrapolated to $N_0 \to \infty$ using $E \sim N_0^{-1.4}$ scaling (small dashed curve with *).

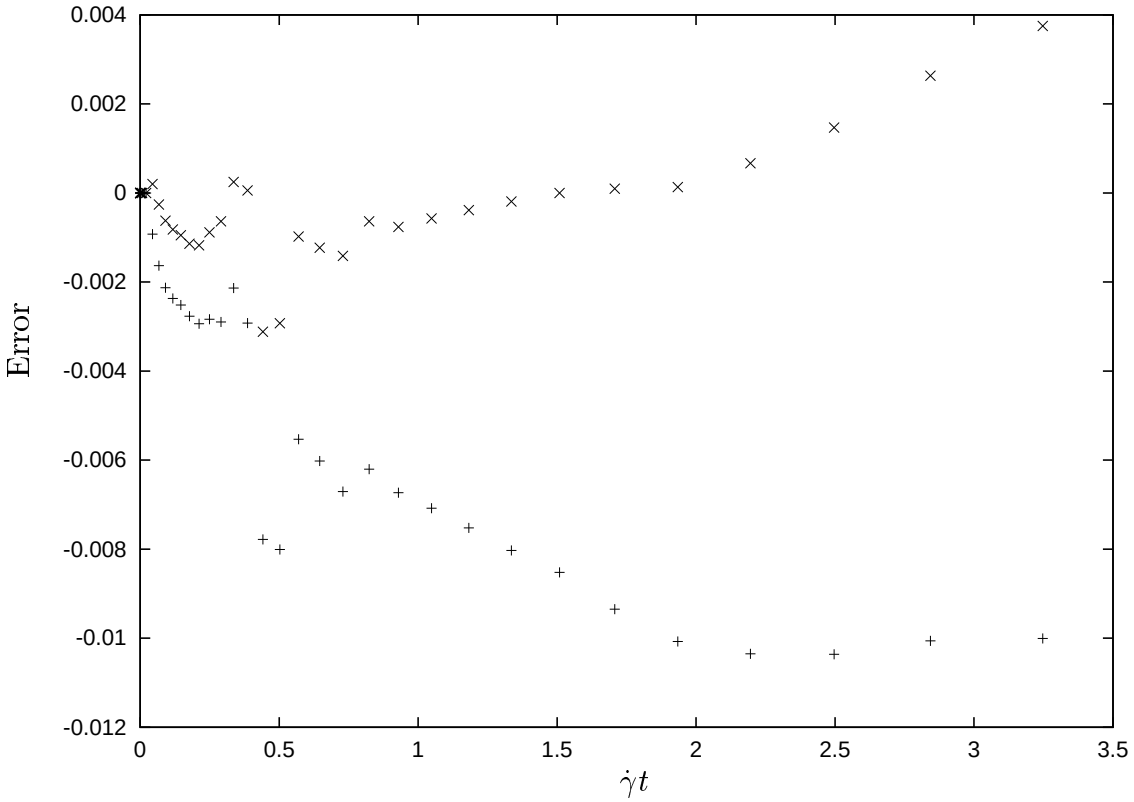


FIG. 5. Extrapolation errors for figure 4 using N_0^{-1} (+) and $N_0^{-1.4}$ (x) scaling of error

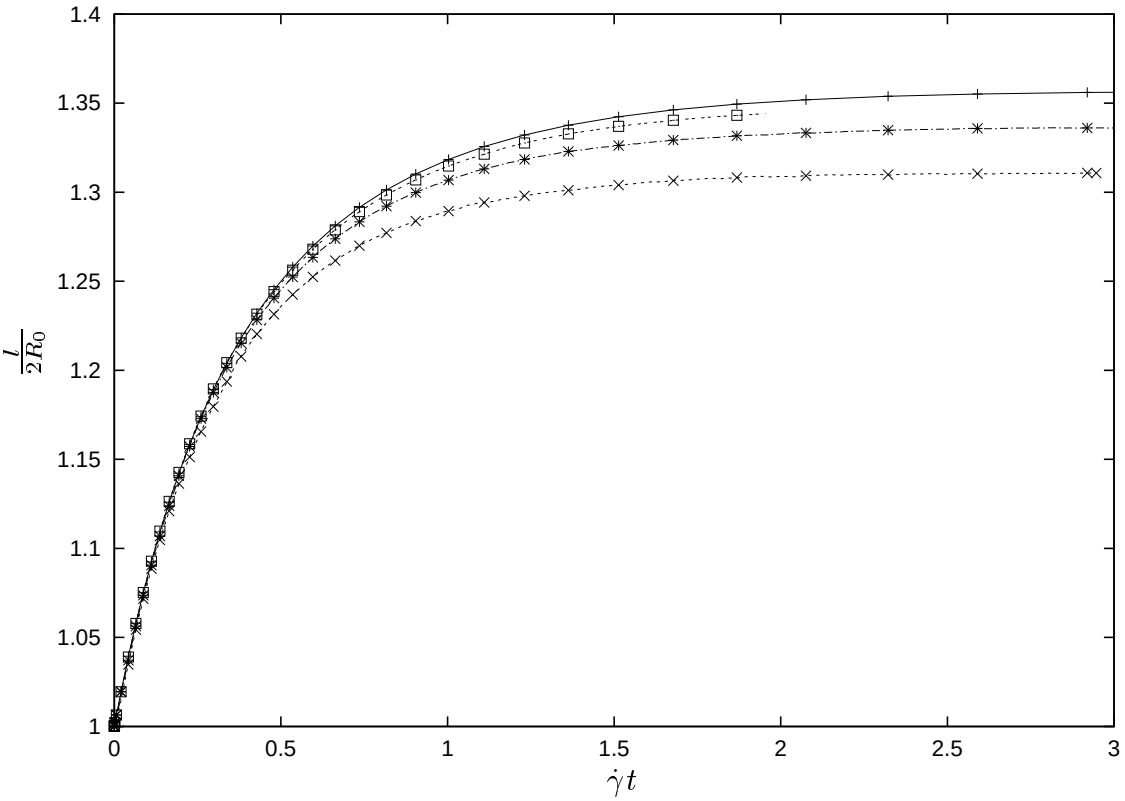


FIG. 6. Drop length $l/2R_0$ as a function of time $\dot{\gamma}t$ during evolution in uniaxial extensional flow; Ca=0.10, $\lambda = 1.0$. Converged 2D axisymmetric finite element results (solid curve with +); Adaptive 3D finite element results obtained using the meshes in Table 2: Mesh 1 (medium dashed curve with x), Mesh 2 (dash-dotted curve with *), Mesh 3 (medium dotted curve with open squares). Symbols are at time steps of 2D FEM simulation and are interpolated from cubic splines for 3D FEM simulations.

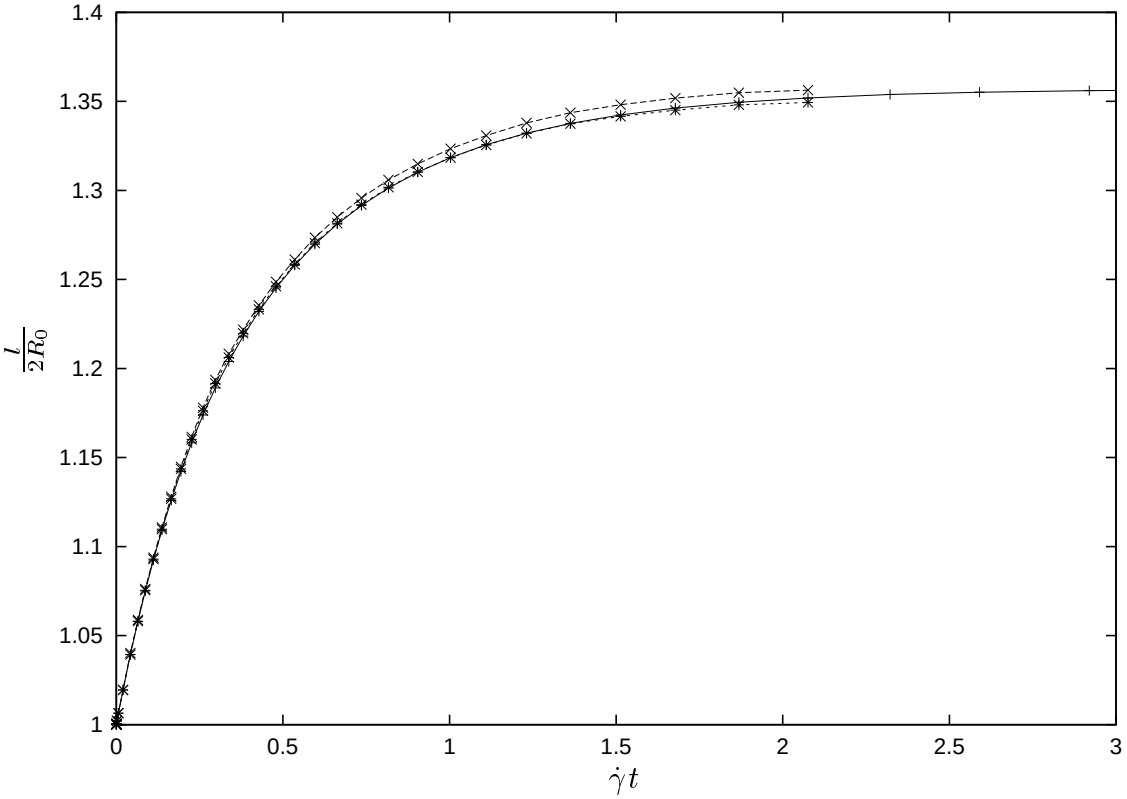


FIG. 7. Drop length $l/2R_0$ as a function of time $\dot{\gamma}t$ during evolution in uniaxial extensional flow; Ca=0.10, $\lambda = 1.0$. Converged 2D axisymmetric finite element results (solid curve with +); Adaptive 3D finite element results extrapolated to $N_0 \to \infty$ using $E \sim N_0^{-1}$ scaling (medium dashed curve with x), Adaptive 3D finite element results extrapolated to $N_0 \to \infty$ using $E \sim N_0^{-1.5}$ scaling (small dashed curve with *).

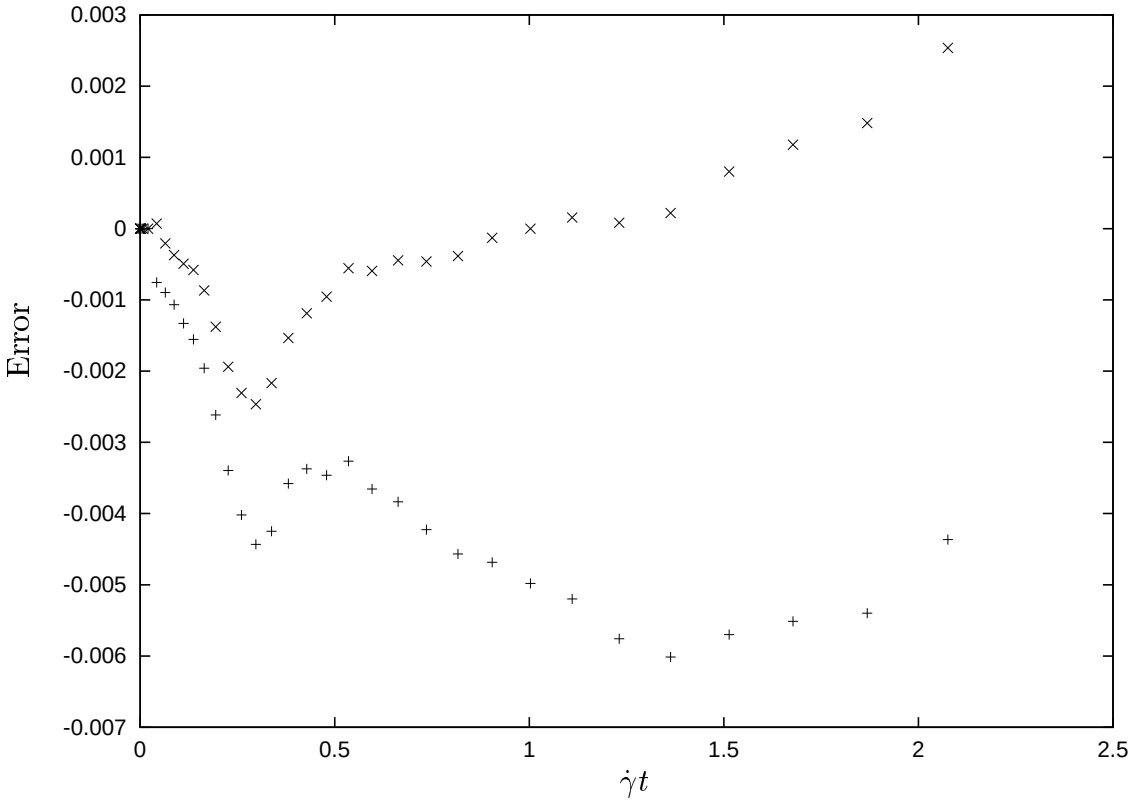


FIG. 8. Extrapolation errors for figure 7 using N_0^{-1} (+) and $N_0^{-1.5}$ (x) scaling of error

these errors may fortuitously offset (be of opposite sign to) those from the $O(\Delta x^2)$ resolution of the drop surface curvature and thereby produce a method with apparently higher order spatial convergence. Accurate identification of the order of convergence of each part of the adaptive 3D FEM and of the method as a whole would be much more involved than the analysis presented in this section. It is not clear whether the return on such significant effort would be of much benefit.

5.2. Subcritical 3D deformation

We next apply the 3D FEM to a fully 3-dimensional transient flow, drop deformation in start-up simple shearing. We choose subcritical flow conditions corresponding to experimental results obtained independently by Guido (private communication), i.e. $\lambda = 1.0$ and $Ca = 0.38$. The outer domain boundary is now constructed as a cubic box having edges of length $20R_0$ in all three dimensions with the center of mass of the drop located at the center of the box. The drop center defines the coordinate origin for application of simple shear using eq. (7). This prevents translation of the drop center by imposing a shear flow by equal and opposite translations of the upper and lower faces of the bounding box. Fully developed shear flow is similarly imposed at the box faces corresponding to the inlet and outlet flow planes. At the remaining two box sides, which have normals in the direction of flow vorticity, the normal velocity is set to zero and a stress-free condition, $\mathbf{n} \cdot \mathbf{T} = \mathbf{0}$, is imposed naturally along the planes. This represents a periodic condition in the vorticity coordinate direction. All specified velocities are imposed as essential boundary conditions by direct modification of the discretized system of equations. At the drop surface, capillarity was treated using the weak form, and normal vectors needed to move nodes were obtained using the paraboloid fit. Domain mesh movement of was performed using the ad hoc procedure based on spines.

The evolution of the dimensionless drop length in the flow direction is shown in figure 9 as a function of dimensionless time. The numerical results are seen to agree with the experimental data to within the accuracy of the experiment ($\sim 10\%$). Additional agreement is shown in the inset which depicts a cross-section of the mesh in the flow plane (having normal vector parallel to flow vorticity) through the center of the drop shown at steady state conditions. The dark curve defines the boundary of the drop determined from the experiment. The use of adaptivity is seen in the higher element density near the drop tips needed to resolve these regions of higher curvature.

5.3. Drop breakup

We now test the ability of the adaptive FEM to handle a more severe 3D simulation of supercritical deformation approaching drop breakup in simple shear flow. Here, $\lambda = 1.0$ and $Ca = 0.44$, and the same initial mesh is used as for the preceding subcritical simple shear simulation. Capillary effects are implemented using the weak form implementation. In figure 10 the drop surface mesh is shown at various times. The efficiency of the transient simulation afforded by adaptivity is evidenced by fewer elements being required early in the simulation with more being added only as needed based on the evolving drop shape. The evolution of dimensionless drop length is compared to converged boundary integral results in figure 12. Good

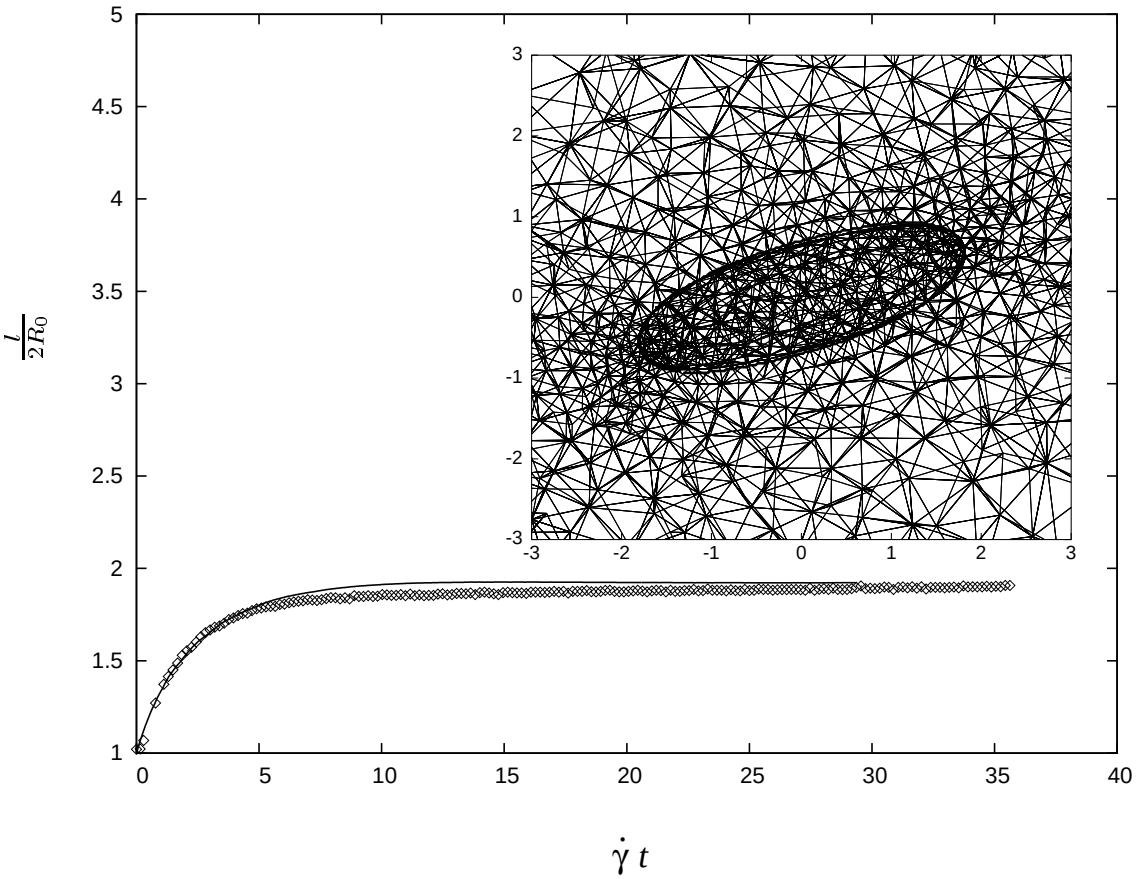


FIG. 9. Dimensionless drop length in the flow direction as a function of dimensionless time, $\gamma = \dot{\gamma}t$, for simple shear with $Ca = 0.38$, $\lambda = 1.0$ and a cubic outer boundary with faces $10R_0$ from the drop center. FEM simulation (solid curve) corresponds to $N_0 = 110$ and is compared to experimental data (open symbols) with the same parameter values obtained by S. Guido (personal communication). Inset: cut through computational mesh showing stationary drop configuration from FEM simulation and from experiment (solid contour).

agreement is maintained to a drop strain, R_{max}/R_0, of about 2.5. Faster drop deformation is predicted by the FEM beyond this point likely as a result of the finite outer domain (located 10 initial drop radii away) and the imposition of fully developed shear flow at the inlet and outlet domain boundaries. The boundary integral formulation implies an outer domain of infinite extent.

In spite of the discrepancy at larger drop deformation, the qualitative features of the approach to breakup are accurately described [29, 10]. The drop elongates and rotates becoming dumbbell in shape and characterized by two symmetric bulbous ends separated by a neck. The bulbs assume a roughly stable size and shape determined by the value of Ca and translate approximately parallel to the flow. Correspondingly the neck shrinks, and two bridges of fluid eventually form that connect the slender neck to the bulbs. The bridges are unstable and would ultimately pinch off in finite time [29]. The approach to pinch-off is evident in figure 11 which shows the mesh cross-section as before for the final time step simulated. The blow-up in the neck region depicts the high concentration of elements required to maintain a specified resolution of curvature. The spatial variation in discretization reinforces how dissimilar length scales are well-resolved by adaptivity and that computational efficiency is realized by adding elements only when and where needed.

5.4. Drop collisions

As a final demonstration of the adaptive 3D FEM, we consider a two-drop system subjected to the same simple shear flow as in the previous two scenarios. Here, $\lambda = 1.0$, $Ca = 0.33$ and the initially spherical drops with unit radius are initially positioned with their centers of mass separated by 6.0 drop radii horizontally and 1.0 drop radius vertically. In contrast to the flows previously considered, the shear flow translates the drops toward each other, forcing them to interact as shown in figure 13 for an adaptive FEM simulation. The drops are prevented from touching (coalescing) due to deformation of the surfaces and the resulting lubrication force that accompanies the squeezing flow developed between the drops. The nonlinearity associated with surface deformation leads to an irreversible interaction in which the final vertical offset of the drops is greater than the initial offset, a phenomenon known as hydrodynamic diffusion.

Figure 14 shows the relative trajectory of the interacting drops as a plot of vertical offset versus horizontal offset of the centers of mass. Two curves for FEM simulations with different refinements of the drop surfaces (but the same refinement of the outer box surface) are shown along with a boundary integral result from the method of Cristini [11] and an extrapolation of the FEM results to infinite refinement as performed in section 5.1. While qualitative agreement is observed, the FEM curves show a consistent shift indicating less vertical offset. At the point of maximum vertical offset, the extrapolated FEM and boundary integral results differ by $\approx 1\%$. A gradual decrease in vertical offset is evident from the outset of the FEM simulations. This vertical migration precedes and is opposite in direction to hydrodynamic interactions of the drops. The same reasoning for the deviation of FEM results from boundary integral simulations in section 5.3 applies here as well, and we believe the present discrepancy to also be due to the presence of the bounding walls used in the FEM simulations. To test this, we performed a

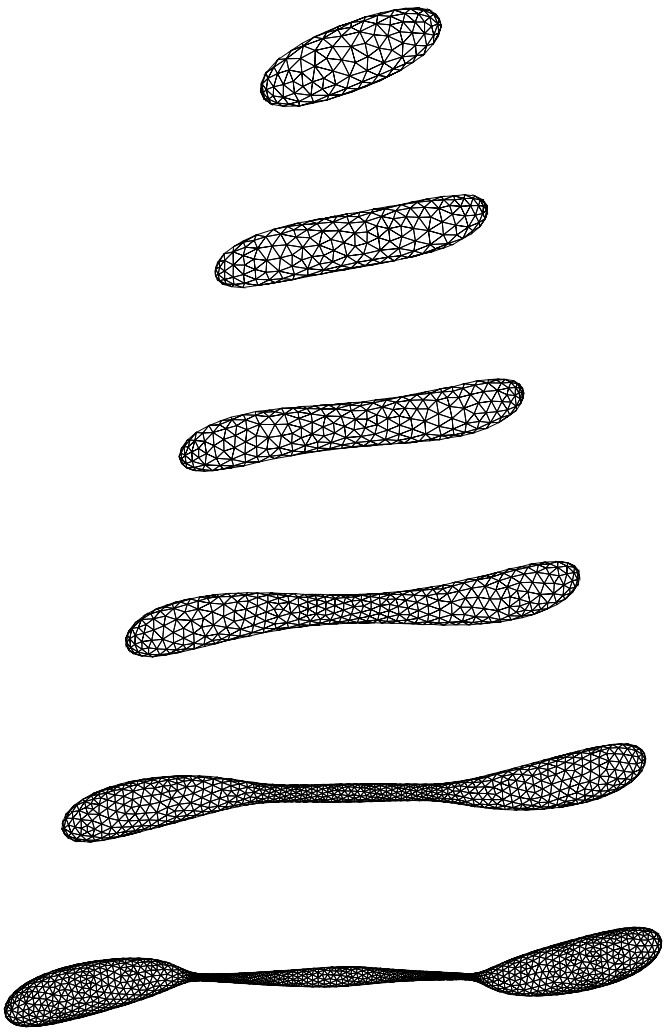


FIG. 10. Drop evolution and onset of breakup in shear flow for $Ca = 0.44$ and $\lambda = 1.0$. Only the surface triangulation is shown. Mesh resolution corresponds to $N_0 = 110$. Dimensionless times $\dot{\gamma}t$ from top to bottom are 3.6, 14.7, 25.1, 35.4, 43.8 and 48.1.

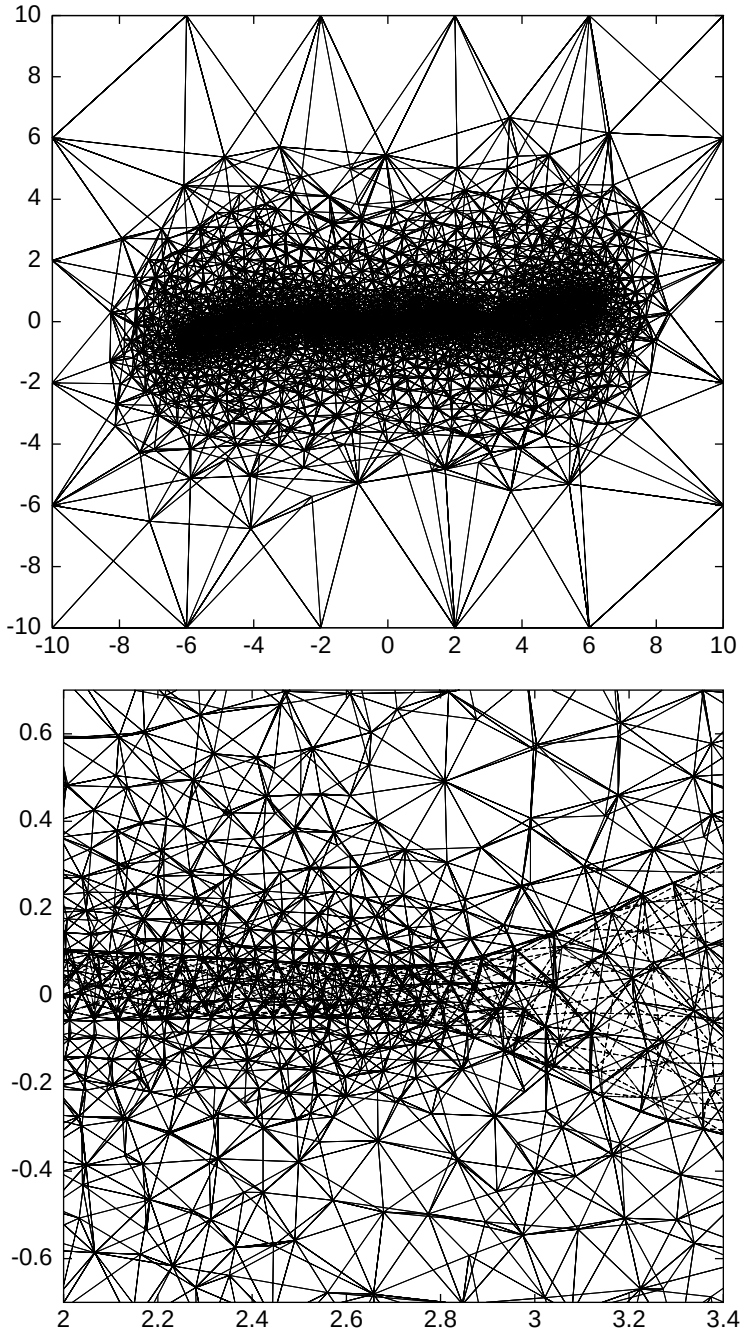


FIG. 11. Final configuration from simulation in figure 10; full domain (top) and blow up of the pinching neck (bottom). Neck thickness $\approx 5\%$ of undeformed drop diameter.

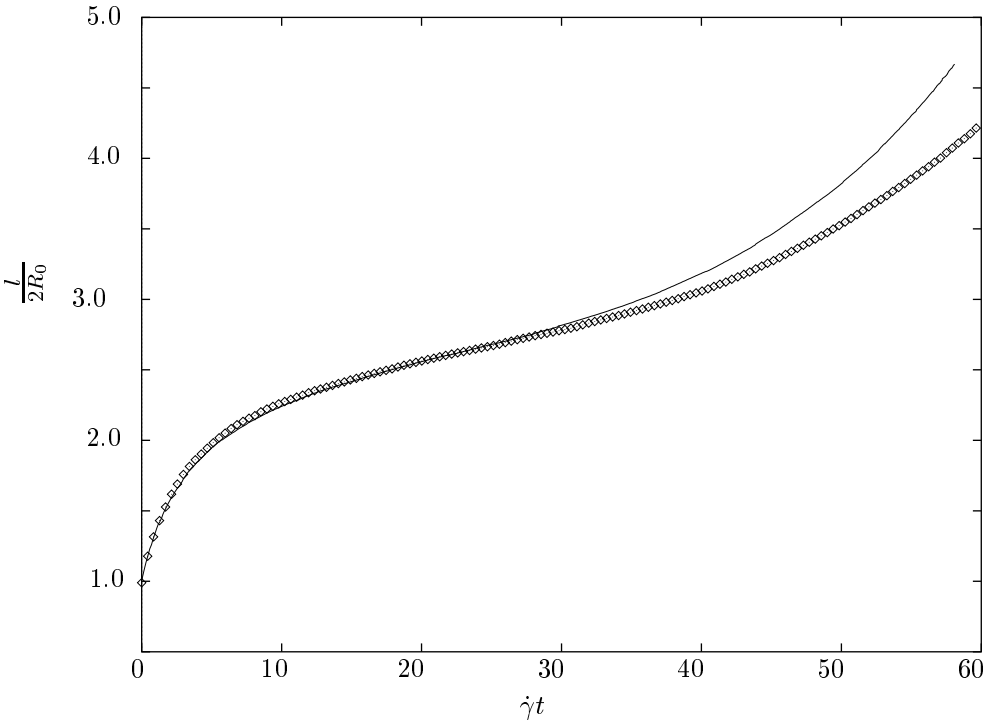


FIG. 12. Dimensionless drop length in the flow direction as a function of dimensionless time for simple shear with $Ca = 0.44$ and $\lambda = 1.0$. FEM with $N_0 = 110$ (solid curve) is compared to boundary integral results (open symbols) using the method of Cristini et al. [11]

subsequent FEM simulation using the same outer bounding box (and discretization) as before but with the drop radius reduced by a factor of 10. By appropriate rescaling of the initial vertical and horizontal offsets and capillary number, this has the effect of moving the bounding walls an order of magnitude farther away. The initial surface of the drop was discretized with the same number of nodes $N_0 = 59$ as the less accurate FEM result of figure 14. The resulting curve is shown in figure 15 along with the boundary integral simulation and the previous FEM simulation with walls $10R_0$ away. The FEM curve for outer boundaries $100R_0$ away shows neither the initial vertical migration nor the consistent vertical shift once the flow is dominated by the interaction of the drops. In fact, the curve follows the much more accurate boundary integral result until numerical inaccuracy associated with under-resolving the flow field between the drops manifests. To see this, figure 17 shows a cut through the 3D mesh near the drops corresponding to figure 13d. At the smallest gap, the two drops are separated by a single layer of tetrahedra. The FEM curve terminates at a point where the 3D meshing algorithm failed to create a mesh.

Additional insight into the performance of the FEM is gleaned by considering the temporal evolution of the gap between the two drops as shown in figure 16 for the finer resolution FEM ($N_0 = 162$) and boundary integral results of figure 14. The semi-log plot enhances the behavior near the closest approach and reveals additional compression arising from the outer boundary in the FEM as already discussed. The noise in the FEM curve for small gap results from discrete changes

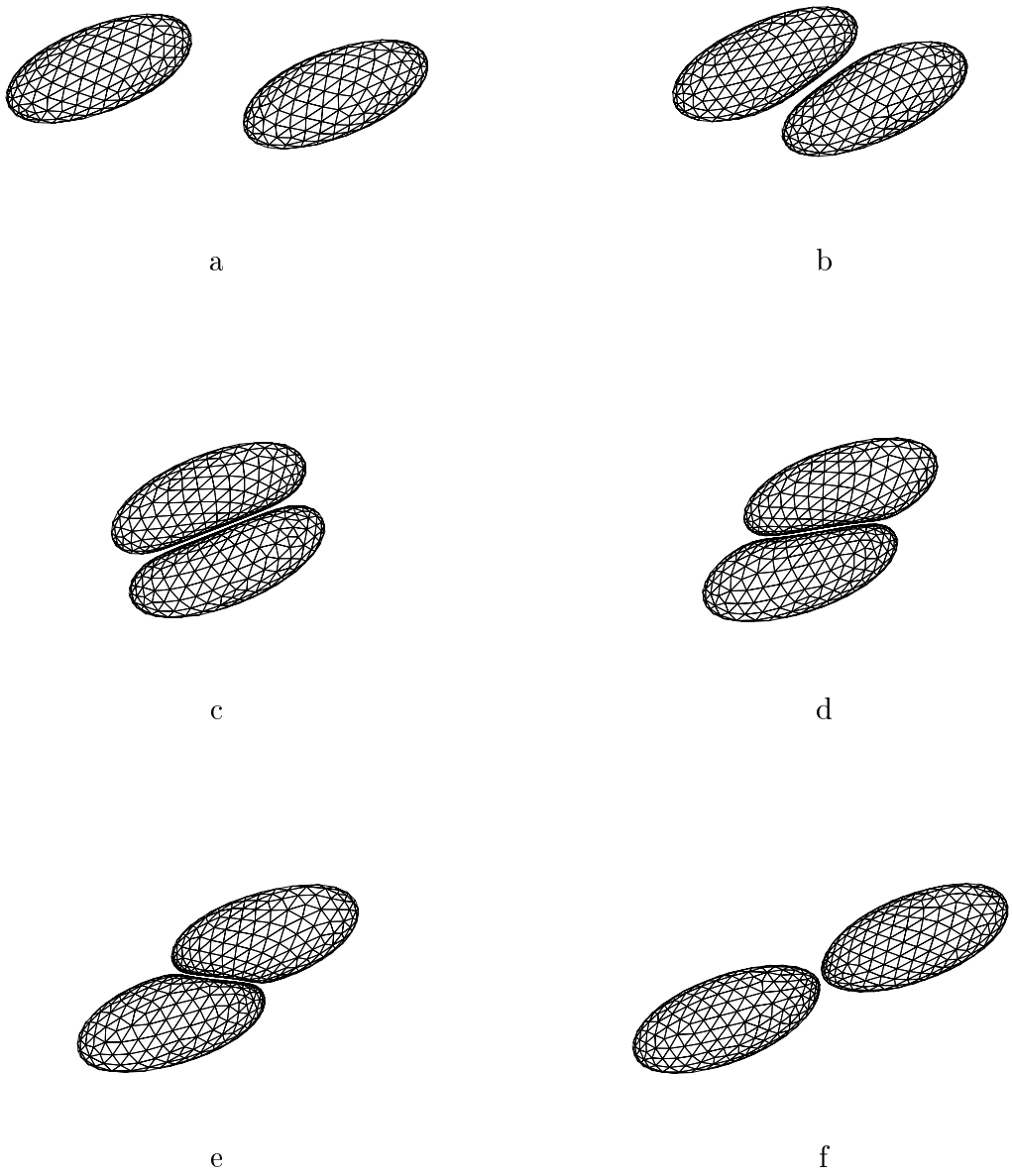


FIG. 13. Drop shape evolution for two equal-sized drops having centers of mass initially offset $\Delta y_{m0} = 0.5$ vertically and $\Delta x_{m0} = 6.0$ horizontally. Flow corresponds to simple shear with $Ca = 0.44$ and $\lambda = 1.0$. Only the surface triangulation is shown. Mesh resolution for each drop corresponds to $N_0 = 162$. Dimensionless times $\dot{\gamma}t$ are as follows: a) 4.92, b) 8.89, c) 10.8, d) 11.7, e) 12.3, f) 13.4.

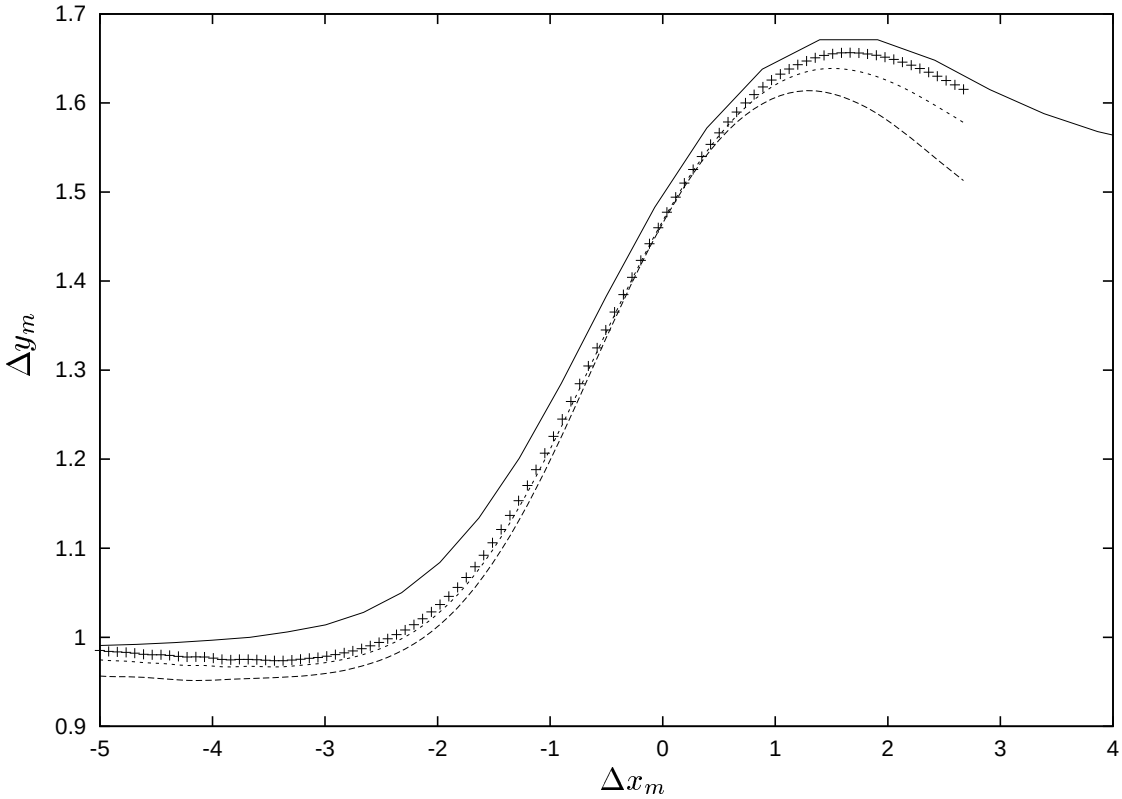


FIG. 14. Vertical offset Δy_m versus horizontal offset Δx_m of drop centers of mass for two equal sized drops with $\lambda = 1.0$ and $Ca = 0.33$ subject to simple shear. FEM curves correspond to $N_0 = 59$ (large dashed curve), $N_0 = 162$ (small dashed curve) and extrapolation to $N_0 \to \infty$ (+) all with outer domain boundary at $10R_0$. Boundary integral result from method of Cristini [11] with $N_0 = 162$ is included (solid curve) for comparison.

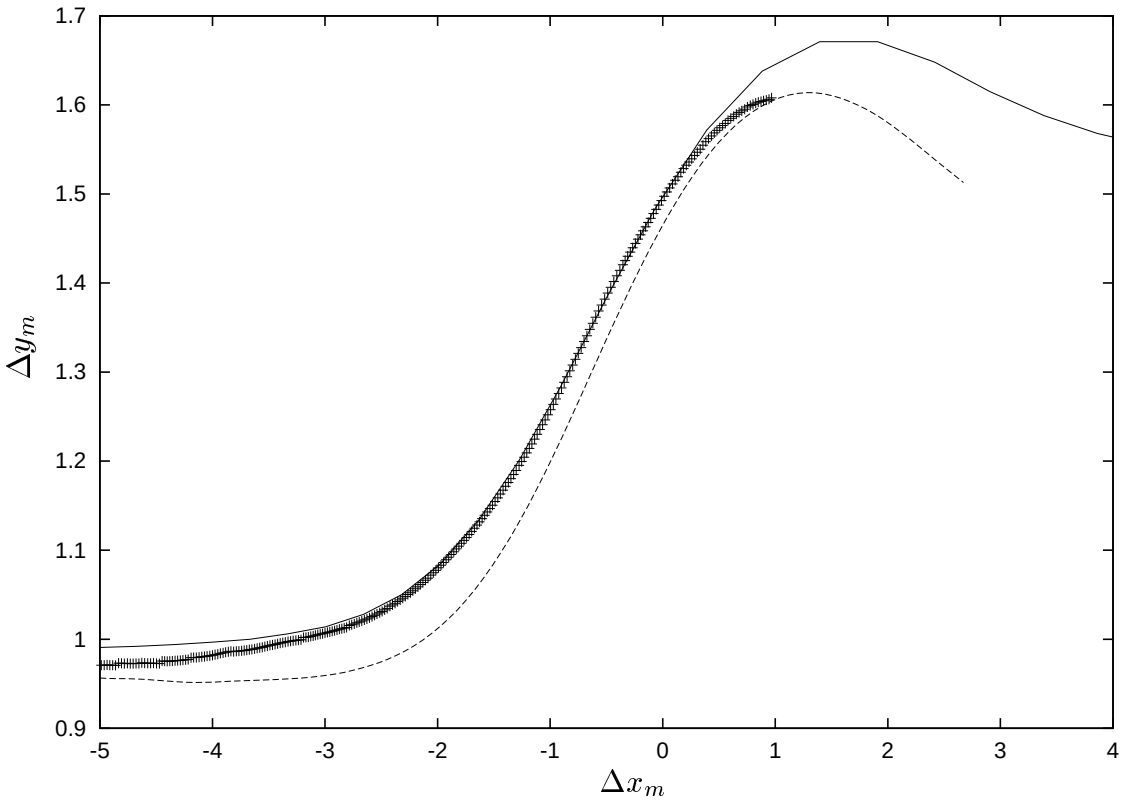


FIG. 15. Vertical offset Δy_m versus horizontal offset Δx_m of drop centers of mass for same conditions as figure 14. FEM curves correspond to $N_0 = 59$ and outer boundary at $10R_0$ (dashed curve) and $100R_0$ (+). Boundary integral result from figure 14 is also shown (solid curve).

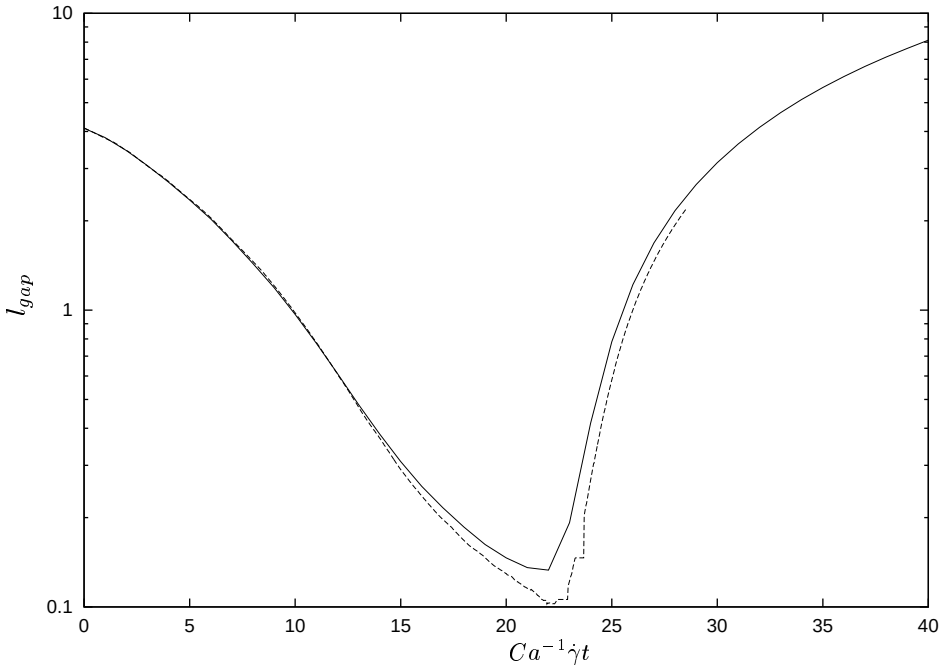


FIG. 16. Shortest distance between drop surfaces l_{gap} versus time made dimensionless using capillary time, $t_\sigma = \frac{\sigma}{\mu R_0} t$ for same conditions as figures 14 and 15. Results are for FEM with $N_0 = 162$ and outer boundary at $10R_0$ (dashed curve) and boundary integral result of figure 14 (solid curve).

in the location of closest approach and is a reflection of the surface discretization. The difference between the curves at the closest approach is 2.8% relative to the initial drop radius.

As a final result comparing FEM and boundary integral simulations which further demonstrates the wall effect and provides a more severe test of the FEM to resolve flow in the gap region, figure 19 shows FEM results for the same conditions as figure 16 but with the initial vertical offset halved, $\Delta y_{m0} = 0.5$. Also shown are two boundary integral simulation results with $\Delta y_{m0} = 0.50$ and $\Delta y_{m0} \approx 0.42$. The latter initial vertical offset was chosen to match the minimum value encountered during the FEM simulation due to the vertical migration induced by the walls. The figure shows that very good agreement between FEM and boundary integral simulations is achieved when the vertical offset of the drop centers is the same going into the regime dominated by the squeezing flow in the gap.

6. CONCLUSIONS

We have demonstrated a novel FEM utilizing an adaptive 3D remeshing capability applied to drop deformations in fully 3D Newtonian systems. Convergence characteristics and accuracy were addressed by quantitative comparison to established axisymmetric FEM and boundary integral numerical results and to independent experimental results. Agreement was good in general with discrepancies being consistent with differences in the way outer boundaries are treated by the FEM and boundary integral method. The results of this chapter indicate this FEM to

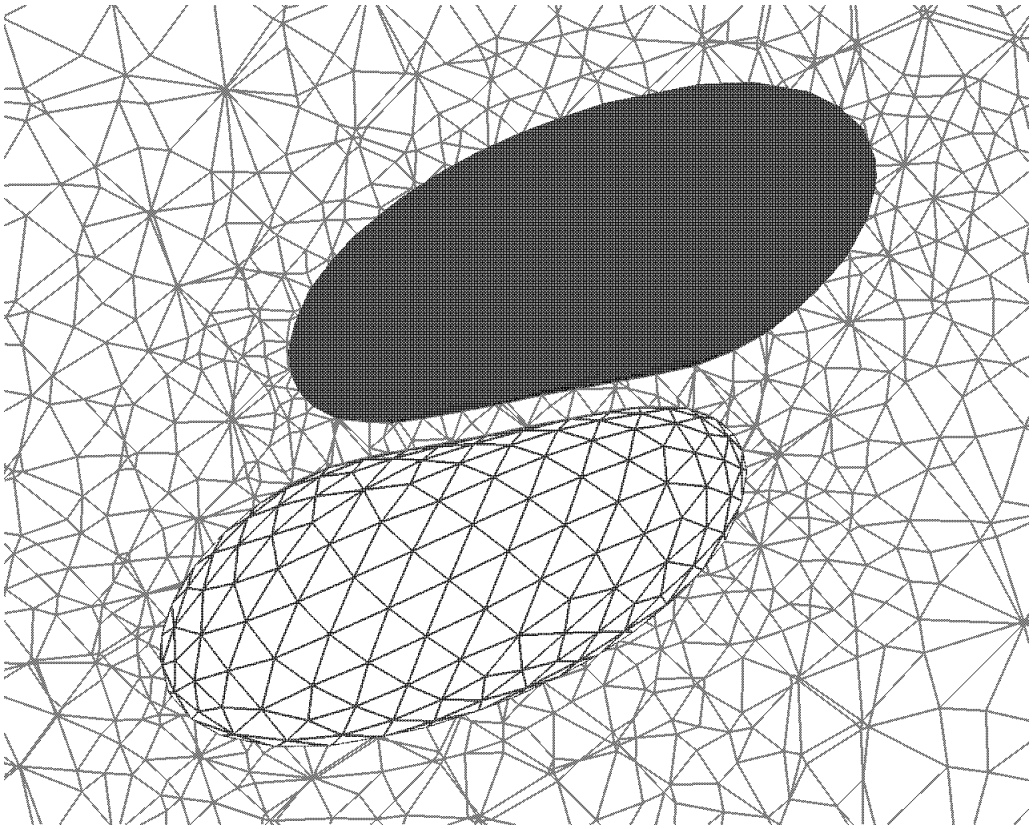


FIG. 17. Cross-section of computational mesh for the simulation of figure 13 corresponding to time $\dot{\gamma}t = 6.41$.

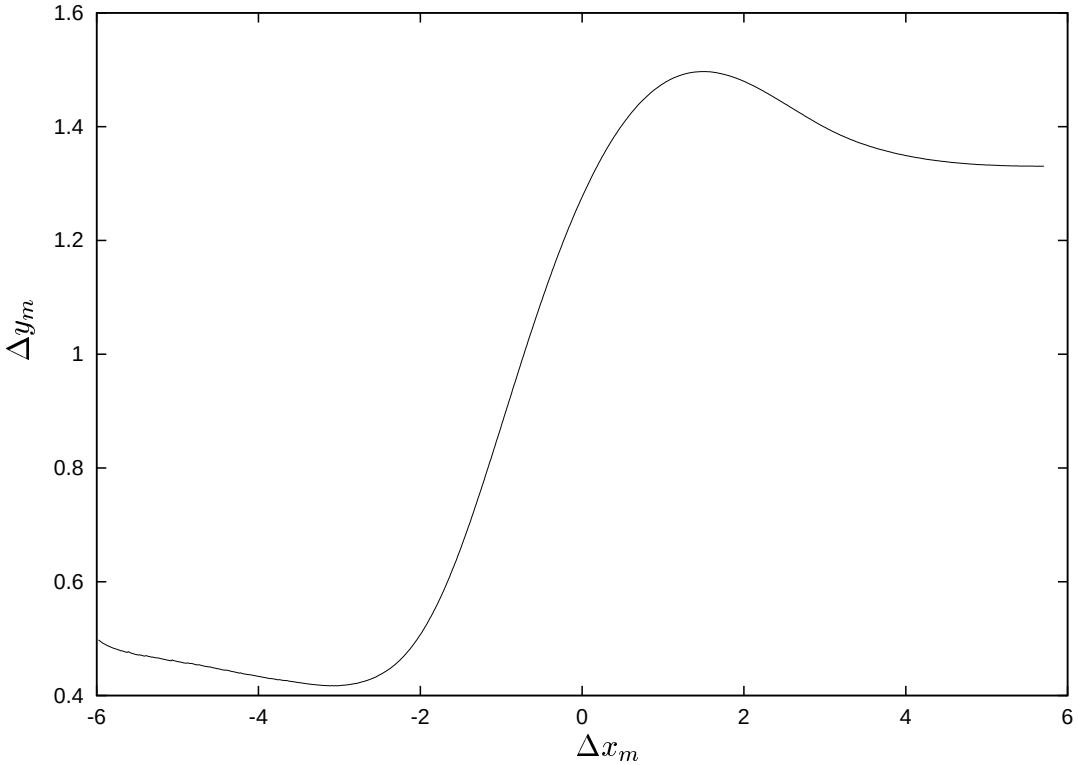


FIG. 18. Vertical offset Δy_m versus horizontal offset Δx_m of drop centers of mass for Ca=0.33 and $\lambda = 1.0$. FEM curve correspond to $N_0 = 162$ and outer boundary at $10R_0$.

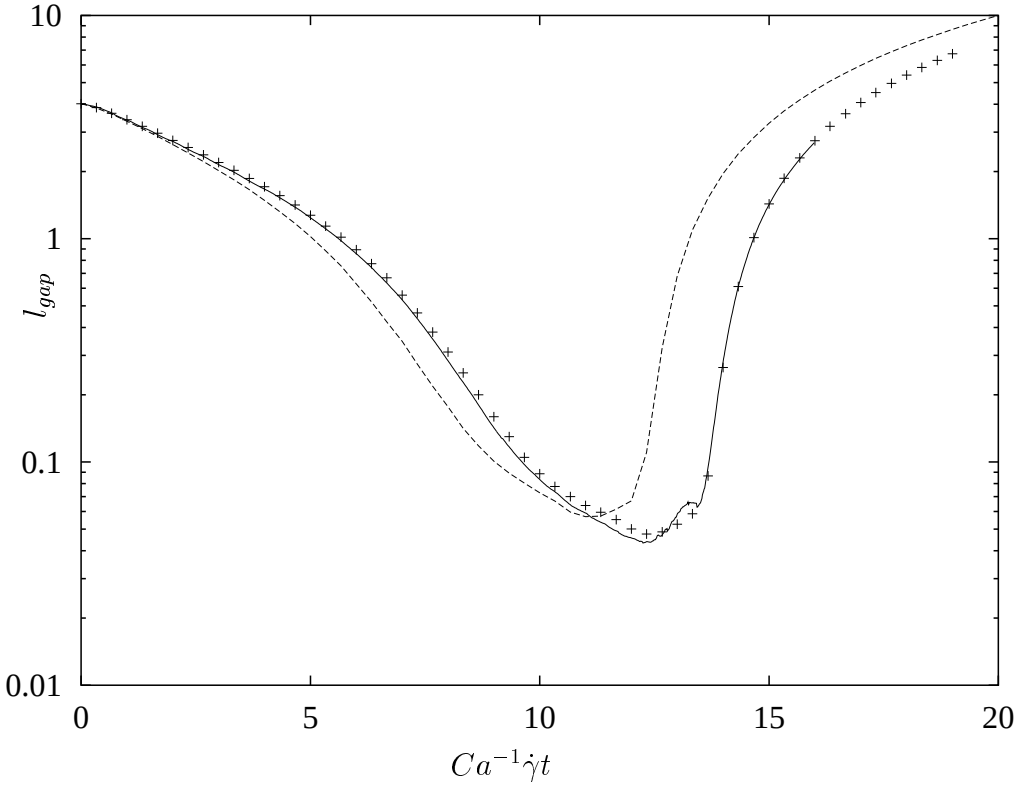


FIG. 19. Shortest distance between drop surfaces l_{gap} versus time made dimensionless using capillary time, $t_\sigma = \frac{\sigma}{\mu R_0} t$ for $\Delta x_{m0} = 6.0$ and $\Delta y_{m0} = 0.5$. Results are for FEM with $N_0 = 162$ and outer boundary at $10R_0$ (solid curve), boundary integral result with same initial vertical and horizontal offsets (dashed curve) and boundary integral result with same initial horizontal offset and initial vertical offset equal to minimum encountered during FEM simulation (plus symbols).

be capable of describing very large deformations such as would be encountered in realistic polymer processing scenarios. Numerical refinements remain which would further improve the accuracy of the method. Among these are a more robust surface mesh density function which more aggressively refines the surface in the gap region for multiple drop interactions. This would result in more than a single layer of 3D tetrahedra being generated in this region and thereby better resolve the local flow field.

In its present form, the FEM developed here is capable of simulating generalized-Newtonian fluid behavior, wall effects and coupled transport phenomena involving diffusion alone. Extension of the method to include convective transport would enable 3D adaptive simulations which include such phenomena as inertia, fluid viscoelasticity, and convection-diffusion of heat and species. Issues relevant to such an endeavor are the subject of the next chapter.

ACKNOWLEDGMENT

The authors thank Dr. Stefano Guido of the Universita' degli Studi di Napoli "Federico II" for providing the experimental data reported in figure 9.

7. APPENDIX: DERIVATIONS/IMPLEMENTATION

Here we present more details for the FEM formulation of this chapter and its implementation. First, the Galerkin form of the governing equations is developed. Building on this, the pressure-stabilized Petrov-Galerkin form is then presented.

7.1. Galerkin $(\mathbf{v}, p)$-FEM formulation

Momentum and mass conservation of eqs. (3) and (4) are cast in a variational formulation over drop domains Ω_1 and outer fluid domain Ω_2 as follows:

$$\int_{\Omega_k} \mathbf{w} \cdot [-\nabla \cdot \mathbf{T}(\mathbf{v}, p)] \, \mathrm{d}\Omega + \int_{\Omega_k} q \nabla \cdot \mathbf{v} \mathrm{d}\Omega = 0 \tag{28}$$

where $\mathbf{w}$ and q are weighting (test) functions, and subscript k denotes the drop ($k = 1$) or outer fluid ($k = 2$) phase domain. Integration by parts can be applied to the first integral to obtain:

$$\int_{\Omega_k} \nabla \mathbf{w} : \mathbf{T}(\mathbf{v}, p) \mathrm{d}\Omega - \int_{\Omega_k} \nabla \cdot [\mathbf{w} \cdot \mathbf{T}(\mathbf{v}, p)] \, \mathrm{d}\Omega + \int_{\Omega_k} q \nabla \cdot \mathbf{v} \mathrm{d}\Omega = 0 \, . \tag{29}$$

Now the divergence theorm is applied to convert the second domain integral to a corresponding surface integral,

$$\int_{\Omega_k} \nabla \mathbf{w} : \mathbf{T}(\mathbf{v}, p) \mathrm{d}\Omega - \int_{\Gamma_k} \mathbf{n} \cdot [\mathbf{w} \cdot \mathbf{T}(\mathbf{v}, p)] \, \mathrm{d}\Omega + \int_{\Omega_k} q \nabla \cdot \mathbf{v} \mathrm{d}\Omega = 0 \tag{30}$$

where $\mathbf{n}$ is the unit normal to the surface of domain Ω_k directed outward from the domain. Writing an expression of the form of eq. (30) for each phase and summing them, keeping in mind $\mathbf{n}_1 = -\mathbf{n}_2$ at $\Gamma \in \Omega_1 \cap \Omega_2$ leads to the following,

$$\int_{\Omega} \nabla \mathbf{w} : \mathbf{T}(\mathbf{v}, p) \mathrm{d}\Omega + \int_{\Gamma} \mathbf{w} \cdot [\mathbf{n} \cdot \mathbf{T}(\mathbf{v}, p)]_{\Gamma} \, \mathrm{d}\Gamma + \int_{\Omega} q \nabla \cdot \mathbf{v} \mathrm{d}\Omega = 0 \, . \tag{31}$$

Here, symmetry of the total stress tensor $\mathbf{T}$ has been exploited, and the resulting integrand for the surface integral corresponds to a jump in stress across the interface. This term allows a natural implementation of dynamic boundary conditions such as eq. (8).

For an equal-order interpolation formulation, the same finite-dimensional basis $\phi^i(\mathbf{x})$ is used for q and each vector component of $\mathbf{w}$ allowing eq. (31) to be discretized using

$$\mathbf{w} = \sum_i a_k^i \phi^i \mathbf{e}_k \tag{32}$$

$$q = \sum_i b^i \phi^i \tag{33}$$

where the sums are performed over all i nodes in the discreatized domain, ϕ^i is a spatially dependent basis function defined at node i, and a_k^i and b^i are arbitrary constants with k ranging from 1 to the number of spatial dimensions. Writing eq. (31) for each i, collecting terms for all a_k^i and b^i and setting their coefficients to zero leads to the following system of residual equations for each component of momentum R_m and for continuity R_c,

$$R_m^i = \int_\Omega \nabla\phi^i \cdot \mathbf{T}(\mathbf{v},p) \cdot \mathbf{e}_m \mathrm{d}\Omega + \int_\Gamma \phi^i \left[\mathbf{n} \cdot \mathbf{T}(\mathbf{v},p)\right]_\Gamma \cdot \mathbf{e}_m \mathrm{d}\Gamma = 0 \tag{34}$$

$$R_c^i = \int_\Omega \phi^i \nabla \cdot \mathbf{v} \mathrm{d}\Omega = 0\,. \tag{35}$$

Appropriate to the Galerkin methodology, the problem degrees of freedom consisting of three components of velocity $\mathbf{v}$ and the pressure p are interpolated using nodal values for each and the same functional basis used for the test functions:

$$v_x(t,\mathbf{x}) = \sum_j v_x(t,\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j) \tag{36}$$

$$v_y(t,\mathbf{x}) = \sum_j v_y(t,\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j) \tag{37}$$

$$v_z(t,\mathbf{x}) = \sum_j v_z(t,\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j) \tag{38}$$

$$p_1(t,\mathbf{x}) = \sum_j p_1(t,\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j) \tag{39}$$

$$p_2(t,\mathbf{x}) = \sum_j p_2(t,\mathbf{x}_j)\ \phi(\mathbf{x},\mathbf{x}_j) \tag{40}$$

where the summation is over all nodes of the mesh having positions $\mathbf{x}_j$, and the coefficients of the summations comprise the time-dependent problem degrees of freedom. The two pressure fields p_1 and p_2 permit a discontinuous pressure at the drop interfaces. Accordingly, pressure p_1 is nonzero within drop phases and at drop boundary nodes, and p_2 is nonzero within the outer fluid phase and at drop boundary nodes. This means of allowing discontinuous pressure across the drop interface(s) preserves bookkeeping based on element-level computations needed for

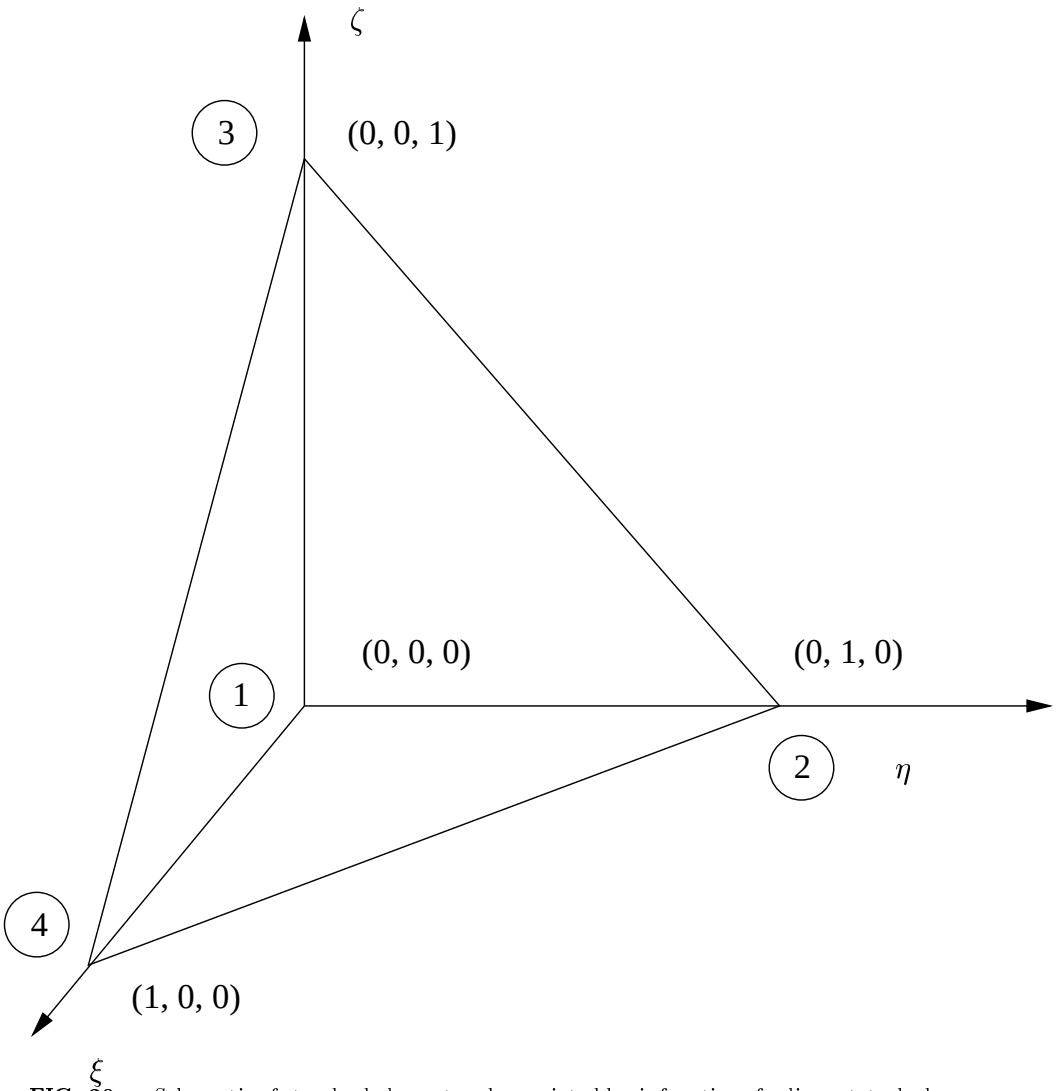


FIG. 20. Schematic of standard element and associated basis functions for linear tetrahedra. Within the standard element, four linear basis functions are defined as follows: $\phi^1 = 1 - \eta - \zeta - \xi$, $\phi^2 = \eta$, $\phi^3 = \zeta$ and $\phi^4 = \xi$.

efficient parallelization at the expense of introducing unused pressure degrees of freedom. The unused pressures decrease for problems involving greater amounts of interfacial nodes. Hence, this approach is least efficient for single drop simulations but quickly improves as the number of drops in multi-drop simulations increases.

The basis functions used both as weighting (test) functions and as interpolating functions for the dependent variables consist of piecewise linear polynomials over a tetrahedron. Such a polynomial can be constructed for each node in the domain, but computations are greatly facilitated by mapping to a standard tetrahedral element shown in Figure 20.

The integrals of eqs. (34) and (35) are performed numerically on the standard element using 4-point Gauusian quadrature approprate to linear tetrahedra [17] .

7.2. Pressure-Stabilized Petrov Galerkin $(\mathbf{v}, p)$-FEM formulation

For equal-order interpolation of velocity $\mathbf{v}$ and pressure p degrees of freedom, spurious pressure modes marked by node-to-node oscillations are known to occur [19]. This stems from violation of the well-known LBB or 'inf-sup' condition and is typically overcome in one of two ways. A mixed formulation can be employed in which the functional basis for velocities is of higher order than that for pressure. This was the approach taken in Chapters 2 and 3 of this thesis. The other popular alternative is to apply a stabilized FEM formulation. This is attractive for 3-dimensional problems involving complex domain shapes due to the relative ease of constructing unstructured tetrahedral meshes. This section presents a pressure-stabilized Petrov-Galerkin (PSPG) FEM formulation which has enjoyed success in similar 3D moving boundary problems involving Newtonain Stokes flow [15, 16].

The PSPG FEM begins by augmenting the variational formulation of eq. (28) as follows:

$$\int_\Omega \mathbf{w} \cdot [-\nabla \cdot \mathbf{T}(\mathbf{v}, p)] \, \mathrm{d}\Omega + \int_\Omega q \nabla \cdot \mathbf{v} \mathrm{d}\Omega$$
$$+ \sum_K \tau_{PSPG} \int_{\Omega_K} [-\nabla \cdot \mathbf{T}(\mathbf{w}, q)] \cdot [-\nabla \cdot \mathbf{T}(\mathbf{v}, p)] \, \mathrm{d}\Omega \;=\; 0 \qquad (41)$$

where the summation is performed over all K elements in the mesh, τ_{PSPG} is an element-based stabilization parameter, and $\mathbf{T}(\cdot, \cdot)$ is the Newtonian Stokes operator given by [30, 31],

$$\mathbf{T}(\mathbf{v}, p) = -p\mathbf{I} + \lambda_i \left(\nabla \mathbf{v} + \nabla \mathbf{v}^T \right) . \qquad (42)$$

The linear tetrahedral basis used in this work leads to second and higher spatial derivatives being identically zero so that the following simplifications apply,

$$\nabla \cdot \mathbf{T}(\mathbf{w}, q) = \nabla q$$
$$\nabla \cdot \mathbf{T}(\mathbf{v}, p) = \nabla p$$

The first two terms of eq. (41) are treated as in the Galerkin FEM formulation. The additional term is a consistent least squares contribution applied at the element level. As recommended in [18] and applied in [15, 16], $\tau_{SUPG} = 0.25 h_K^2$ with h_K half the side length of an equilateral tetrahedron having the same volume as element K. Substituting eqs. (32) and (33) into eq. (41) and applying the manipulations of Section 7.1 leads to the expressions of eqs. (10) and (11).

REFERENCES

1. P. L. George. *Automatic mesh generation, application to finite-element methods.* Wiley, Chichester, 1991.
2. T. J. Baker. Developments and trends in three-dimensional mesh generation. *Appl. Numer. Math.*, 5:275–304, 1989.
3. M. S. Shephard and M. K. Georges. Automatic three-dimensional mesh generation by the finite octree technique. *Int. J. Numer. Methods Eng.*, 32:709–749, 1991.
4. P. L. George, F. Hecht, and E. Saltel. Automatic mesh generator with specified boundary. *Comput. Methods Appl. Mech. Eng.*, 92:269–288, 1991.

5. Andrew A. Johnson. *Mesh generation and update strategies for partallel computations of flow problems with moving boundaries and interfaces.* PhD thesis, Univ. of Minnesota, Minneapolis, 1995.

6. A. A. Johnson and T. E. Tezduyar. Parallel computation of incompressible flows with complex geometries. *Intern. J. Numer. Meths. Fluids*, 24:1321–1340, 1997.

7. P. A. Sackinger, P. R. Schunk, and R. R. Rao. A newton-raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation. *J. Comput. Phys.*, 125:83–103, 1996.

8. R. A. Cairncross, P. R. Schunk, T. A. Baer, R. R. Rao, and P. A. Sackinger. A finite element method for free surface flows of incompressible fluids in three dimensions. part I. boundary fitted mesh motion. *Int. J. Num. Meth. Fluids*, 33:375–403, 2000.

9. T. A. Baer, R. A. Cairncross, P. R. Schunk, R. R. Rao, and P. A. Sackinger. A finite element method for free surface flows of incompressible fluids in three dimensions. part II. dynamic wetting lines. *Int. J. Num. Meth. Fluids*, 33:405, 2000.

10. Vittorio Cristini. *Drop dynamics in Viscous flows.* PhD thesis, Yale University, New Haven, Connecticut, 2000.

11. V. Cristini, J. Bławzdziewicz, and M. Loewenberg. An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence. *J. Comput. Phys.*, 168:445–463, 2001.

12. D. J. Mavriplis. Unstructured grid techniques. *Ann. Rev. Fluid Mech.*, 29:473–514, 1997.

13. D. L. Marcum and N. P. Weatherill. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal*, 33:1619–1625, 1995.

14. Howard A. Stone. Dynamics of drop deformation and breakup in viscous fluids. *Annu. Rev. Fluid Mech.*, 26:65–102, 1994.

15. Hua Zhou. *Three-dimensional finite-element modeling of viscous sintering.* PhD thesis, University of Minnesota, Minneapolis, 1998.

16. H. Zhou and J. J. Derby. A parallel, finite element method for three-dimensional, moving-boundary flows driven by capillarity and its application to viscous sintering. in review, 2000.

17. T. J. R. Hughes. *The finite-element method: linear static and dynamic finite element analysis.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

18. T. J. R. Hughes, L. P. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the babuska-brezzi condition: A stable petrov-galerkin formulation of stokes problem accomodating equal-order interpolations. *Comput. Methods Appl. Mech. Engrg.*, 59:85–99, 1986.

19. P. M. Gresho and R. L. Sani. *Incompressible Flow and the Finite Element Method.* John Wiley & Sons, New York, 1998.

20. J. N. Reddy. *An Introduction to the Finite Element Method.* McGraw-Hill, New York, 1984.

21. Y. Saad and M. Schultz. GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems. *SIAM*, 7:856–869, 1986.

22. A. Yeckel and J. J. Derby. Parallel computation of incompressible flows in materials processing: Numerical experiments in diagonal preconditioning. *Parallel Computing*, 23:1379–1400, 1997.

23. Kenneth J. Ruschak. A method for incorporating free boundaries with surface tension in finite element fluid-flow simulators. *Intl. J. Num. Meth. Engr.*, 15:639–648, 1980.

24. L.-W. Ho and A. T. Patera. Variational formulation of three-dimensional free-surface flows: natural imposition of surface tension to boundary conditions. *Intern. J. Numer. Meths. Fluids*, 13:691–698, 1991.

25. A. Z. Zinchenko, M. A. Rother, and R. H. Davis. A novel boundary-integral algorithm for viscous interaction of deformable drops. *Phys. Fluids A*, 9:1493–1511, 1997.

26. R. W. Hooper, C. W. Macosko, and J. J. Derby. Assessing a flow-based finite element model for the sintering of viscoelastic particles. *Chem. Eng. Sci.*, 55:5733–5746, 1999.

27. Russell Hooper, Valmor DeAlmeida, Jeffrey Derby, and Christopher Macosko. Transient polymeric drop extension and retraction in uniaxial extensional flows. *J. Non-Newt. Fluid Mech.*, 98:141–168, 2001.

28. R. W. Hooper, E. M. Toose, C. W. Macosko, and J. J. Derby. A comparison of finite element and boundary element methods for modeling axisymmetrix polymeric drop deformation. *Int. J. Num. Meth. Fluids*, 2001. in press.

29. V. Cristini, J. Bławzdziewicz, and M. Loewenberg. Drop breakup in three-dimensional viscous flows. *Phys. Fluids*, 10:1781–1783, 1998.

30. L. P. Franca, T. J. R. Hughes, and R. Stenberg. Stabilized finite element methods for the stokes problem. Technical Report 11, Helsinki University of Technology, January 1991.

31. T. E. Tezduyar. *Stabilized finite element formulations for incompressible flow computations*, pages 1–44. Academic Press, Inc., San Diego, CA, 1992.