INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 22, 103–120 (1996)

AN ALGORITHM FOR SIMULATION OF STEADY FREE SURFACE FLOWS

NATARAJAN RAMANAN AND MICHAEL S. ENGELMAN

Fluid Dynamics International, 500 Davis St., Suite 600, Evanston, IL 60201, U.S.A.

SUMMARY

An algorithm to simulate steady, viscous free surface flows is presented in this paper. A Picard-type approach wherein the flow and free surface updates are performed alternately is utilized to iterate for a solution. The procedure is intended for large-scale two- or three-dimensional problems. A surface-intrinsic co-ordinate system which facilities representation of general free surface shapes is used. Using a Galerkin finite element method (GFEM), two free surface updates, namely kinematic and normal stress updates are formulated. It is shown that the effects of surface tension, surface tension gradients and imposition of contact angles can be simulated elegantly within the framework of the GFEM. A novel feature of the updates is that the deformations are sought in a direction normal to the current iterate free surface shape, with the result that the method is ideally suited when used in conjunction with an automatic mesh generator. With the normal stress update a volume constraint can also be imposed. A segregated method is utilized to solve iteratively one degree of freedom at a time for the solution of the flow variables. As a result, the memory and disc space requirements are minimal. Sample problems in extrusion, coating and crystal growth are presented to clearly illustrate the convergence behaviour and accuracy of the algorithm.

KEY WORDS: free surface; free boundary problem; segregated method; finite elements; extrusion

1. INTRODUCTION

An interfacial region of molecular proportions is present between two immiscible fluids, the extent of which provides for a smooth change in physical properties. From a continuum viewpoint the tangible influence of this interface is an interfacial tension, a consequence of molecular forces of attraction. It is this force that accounts for the pressure difference across drops and bubbles. Engineering problems which possess interfaces of immiscible fluids are wide and varied. Common examples that are of current interest include extrudate swell, crystal growth from melts, coating flows, electrolytic cells, etc. The ability to predict the interface shapes in conjunction with fluid flow and heat flow in the bulk fluids is of utmost importance in these and other engineering problems. For a continuum treatment of viscous fluid flow the existence of such an interface poses additional complexities through non-linear boundary conditions that need to be satisfied thereon. Such non-linearity is further accentuated in situations where non-uniform temperature, species or electric field effect spatial gradients in the interfacial tension are present. It is for this reason that these problems are intractable to analytical techniques except under excessively simplifying assumptions and have been pursued as an active area of research in the realm of computational fluid dynamics.

In the following sections an algorithm to simulate steady, viscous external flows is presented. For compactness and clarity we devote our attention to cases where the interface demarcates the fluid of interest and a tenuous vapour. In addition, we shall assume that the fluid is viscous and incompressible. The special case of free boundary problems of inviscid fluids has received considerable attention in the

CCC 0271–2091/96/020103–18

Received December 1993
Revised July 1994

literature. The subject of viscous free surface flows, which has received much less scrutiny, is the focus of our attention.

The numerical solution of viscous free surface problems was pioneered by the studies of Nickell *et al.*[1] and Harlow and Welch.[2] The two studies approached the problem through different representations of the free surface and solution methods. Nickell *et al.*[1] assumed the free surface to be piecewise continuous, whereas Harlow and Welch[2] represented the surface as a collection of piecewise discontinuous segments (a staircase type). The former is ideally suited to a boundary element or finite element discretization of the equations and researchers of free surface algorithms in those areas continued their efforts based on Nickell *et al.*'s[1] approach. Harlow and Welch's[2] approach, on the other hand, lends itself suitably to a finite-difference- or control-volume-based discretization of the equations and the research community[3–5] in those areas progressed along these lines.

In the finite element area, the study of Nickell *et al.*[1] is noteworthy in that it was the earliest application of the finite element method to the solution of free surface flows. By first assuming a free surface shape, they solved for the flow solution subject to the satisfaction of normal and tangential stress conditions. Then they determined the free surface deformation necessary to satisfy the other boundary condition of no normal flow. They repeated in succession the solution of flow variables and free surface updates until a converged solution was attained for both the free surface shape and the flow field. Although their study was performed for a fluid with vanishing surface tension, from an algorithmic viewpoint it proved the feasibility of a Picard-type approach to the solution of free surface flows. In addition, although Nickell *et al.*[1] used the kinematic constraint as the boundary condition to update the free surface, if the fluid has a finite surface tension, one of the stress boundary conditions can be used to do so. Silliman and Scriven[6] comprehensively addressed the issue of proper choice of boundary conditions to perform free surface updates. Working with two-dimensional geometries, they provided two free surface updates, namely kinematic and normal stress iterations. They claimed that the former is suitable for small or vanishing surface tension, while the latter should be preferred for large or finite surface tension problems. The key feature of their algorithms is the ability to represent the free surface as a height function. Other studies that relied on the height function approach include those by Fredericksen and Watts[7] and Keunings.[8] Orr *et al.*[9] addressed many meniscus problems in 3D by representing the free surface height as a function of the horizontal co-ordinates.

The Picard approaches of Nickell *et al.*[1] and Silliman and Scriven[6] possess only a linear rate of convergence. To obtain quadratic convergence rates, Saito and Scriven[10] proposed a Newton-based algorithm using splines that also overcame some of the limitations of the height function representation of Silliman and Scriven.[6] The Newton-based approach was adopted for 2D problems by many others, including Kistler and Scriven,[11] Kheshgi and Scriven,[12] Coyle *et al.*[13] and Engelman and Sani.[14] Karagiannis *et al.*[15] extended the same method to three-dimensional problems and have reported solutions for extrudate swell of fluid flow out of square, rectangular and other non-trivial shapes of dies. Their analysis ignores the effects of surface tension. The alternative to the Newton method of Saito and Scriven[10] is the total linearization method of Cuvelier and Schulkes,[16] who also solved a coupled set of equations along with the free surface shapes. Ho and Patera[17] provide yet another alternative to the height function method. They developed a transient algorithm based on a local surface co-ordinate system suitable for periodic or closed surfaces. The arbitrary Lagrangian–Eulerian (ALE) method of Ramaswamy and Kawahara[18] offers another approach to the solution of viscous free surface flows.

It is evident that an algorithm that handles general free surface shapes with the ability to handle contact angle conditions is highly desirable for the simulation of viscous free surface problems. While the Newton method proposed by Saito and Scriven[10] performs quite satisfactorily in two-dimensional geometries, an extension of the same to three dimensions will be memory- and CPU-time-intensive, as illustrated by the work of Karagiannis *et al.*[15] We believe that a Picard-type iteration in combination with a segregated solution approach will substantially reduce the demand on computational resources. With this in mind we present one such algorithm that requires just C^0 continuity of the surface, includes effects of interfacial tension and imposes contact angle conditions naturally in the weak form.

2. GOVERNING EQUATIONS

For an elegant exposition of the algorithm we will restrict our attention to the steady, incompressible flow of a Newtonian fluid. The interface is assumed to exist in the presence of a vapour of vanishing viscosity. The Navier–Stokes equations under such assumptions are

$$u_{i,i} = 0 \tag{1}$$

$$\rho u_j u_{i,j} = T_{ij,j} + f_i. \tag{2}$$

In these equations $\mathbf{u}$ is the velocity, $\mathbf{T}$ is the total stress and $\mathbf{f}$ is the body force. The fluid density is represented by ρ.

The total stress tensor $\mathbf{T}$ comprises the pressure p and the deviatoric stress $\boldsymbol{\tau}$:

$$T_{ij} = -p\delta_{ij} + \tau_{ij}.$$

A constitutive equation is necessary to define the relationship between stress and strain. This can be written for a Newtonian fluid as

$$\tau_{ij} = \frac{\mu}{2}(u_{i,j} + u_{j,i}).$$

The governing equations (1) and (2) are subject to the boundary conditions

$$u_i = \bar{u} \quad \text{on } \Gamma_u, \tag{3}$$

$$n_j T_{ij} = \bar{T}_i \quad \text{on } \Gamma_t, \tag{4}$$

where Γ_u and Γ_t signify the boundaries on which Dirichlet conditions on velocity and stress are applied. In addition to the above boundary conditions, the following conditions must be satisfied at the fluid–vapour interface Γ_f:

$$u_i n_i = 0, \tag{5}$$

$$n_i T_{ij} t_j = t_i \sigma_i, \tag{6}$$

$$n_i T_{ij} n_j = -p_0 + \sigma H, \tag{7}$$

where $\mathbf{n}$ is the outward-pointing normal and $\mathbf{t}$ is a tangential direction on the free surface. In the above equations p_0 is the ambient pressure, σ is the surface tension and H is the total curvature of the fluid–vapour interface. Since the flow is assumed steady, the kinematic constraint is simply a statement of no normal flow. Spatial gradients in surface tension, as seen from equation (6), can induce tengential stresses on the bulk fluid. Such gradients can be the cause of non-uniform temperature or surfactant distribution. The comma ($= \nabla_s$) operator here is the surface gradient operator.[19] At the interface the difference in normal stress between fluid and vapour is compensated by the forces of interfacial tension.

If the flow is assumed to be slow, the equations can be made dimensionless by characteristic scales of L, U and $\mu U/L$. The Navier–Stokes equations can then be cast in a non-dimensional form as

$$u_{i,i} = 0, \tag{8}$$

$$Re\, u_j u_{i,j} = T_{ij,j} + e_i \frac{Bo}{Ca}, \tag{9}$$

where $Re = \rho UL/\mu$, $Ca = \mu U/\sigma_0$ and $Bo = \rho g L^2/\sigma_0$, subject to Dirichlet conditions on velocity and stress. The surface tension σ_0 is the reference value. In the momentum equation the body force is assumed to be only due to gravitational attraction that acts in the direction specified by the unit vector **e**. On the free surface the interfacial boundary conditions dictate

$$u_i n_i = 0, \tag{10}$$

$$Ca\, n_i T_{ij} t_j = t_i \sigma_{,i}, \tag{11}$$

$$n_i T_{ij} n_j = -p_0 + \frac{1}{Ca} H. \tag{12}$$

In the above equations, three dimensionless groups appear. Re is the Reynolds number, the ratio of inertial and viscous forces. The capillary number Ca measures the relative importance of viscous and surface tension forces. The ratio of gravity forces to interfacial tension is the Bond number Bo. The ratio of Bond and capillary numbers is often referred to as the Stokes number. We now turn to the weak form of these equations to describe how the free surface boundary conditions come into play.

3. GALERKIN FINITE ELEMENT METHOD

3.1. Weak formulation of governing equations

Using the Galerkin finite element method and the Green–Gauss theorem, the weak form of equation (2) can be written as

$$\int_V \psi u_{i,i} = 0, \tag{13a}$$

$$\int_V Re\varphi_k u_j u_{i,j}\,\mathrm{d}v - \int_V \varphi_{k,i} p\,\mathrm{d}v + \int_V \varphi_{k,j} u_{i,j}\,\mathrm{d}v + \int_V \varphi_{k,j} u_{j,i}\,\mathrm{d}v = \int_\Gamma n_j T_{ij} \varphi_k\,\mathrm{d}s + \int_V \frac{Bo}{Ca} \varphi_k e_i\,\mathrm{d}v, \tag{13b}$$

where ψ is the basis function for pressure and φ is the basis function for velocities in the computational domain V. The traction terms $\bar{T}$ can be applied to Γ_t where the stresses are prescribed, while on Γ_f the free surface boundary conditions of equations (11) and (12) are applied.

The weak form of the equations can be solved either in a coupled or in an uncoupled fashion. Saito and Scriven[10] solve the set of equations using a Newton method which allows for simultaneous evaluation of flow field and free surface position. Such a method is quadratic in convergence rates but memory- and CPU-intensive, as discussed previously. For this reason we opt to use a fixed point iteration wherein the flow solution and free surface update are performed successively in an iterative fashion. When the flow solution is to be obtained on a free surface iterate, only two of the three equations (10)–(12) can be satisfied on the surface. As observed by Silliman and Scriven,[6] the choice of the distinguished boundary condition is contingent upon the capillary number. At low capillary numbers the normal stress balance is the proper condition to update the free surface. In high or infinite capillary number situations the kinematic constraint is the distinguished condition that should be used to update the shape. In the following subsections we provide a description of these two updates performed on general free surface shapes. We will, however, first provide details of how we obtain the flow solution for a given free surface iterate.

3.2. *The segregated method*

For the solution of the flow field we used the segregated algorithm of Haroutunian *et al.*[20] This algorithm is a variant of the popular SIMPLER algorithm of Patankar.[21] A brief description of the two-dimensional version of this method is provided here for completeness.

When the computational domain is discretized, the weak form of the Navier–Stokes equations results in a set of linear algebraic equations

$$\mathbf{K}_u\mathbf{u} - \mathbf{C}_x\mathbf{p} = \mathbf{f}_x, \tag{14}$$

$$\mathbf{K}_v\mathbf{v} - \mathbf{C}_y\mathbf{p} = \mathbf{f}_y, \tag{15}$$

$$\mathbf{C}_x^{\mathrm{T}}\mathbf{u} + \mathbf{C}_y^{\mathrm{T}}\mathbf{v} = 0, \tag{16}$$

where $\mathbf{K}_u$ and $\mathbf{K}_v$ are the advection–diffusion matrices of equation (13b), $\mathbf{C}_x$ and $\mathbf{C}_y$ are the pressure matrix terms and $\mathbf{f}_x$ and $\mathbf{f}_y$ are the forcing vectors. To solve this set, the segregated algorithm sets up a linear algebraic equation for each degree of freedom. While each of the momentum equations provides a matrix equation for each velocity degree of freedom, one needs a matrix equation for pressure. A Poisson-type matrix equation is derived for this purpose at the discrete level by multiplying the momentum equations by the inverses of $\mathbf{K}_u$ and $\mathbf{K}_v$ respectively followed by multiplication by $\mathbf{C}_x^{\mathrm{T}}$ and $\mathbf{C}_y^{\mathrm{T}}$ respectively. The procedure resulting from this manipulation is referred to as the *pressure projection method* and is discussed in detail by Haroutunian *et al.*[20]

3.3. *Kinematic update*

In cases where the surface tension effects are small in comparison with the viscous forces (i.e. $Ca \gg 1$), the use of the kinematic constraint as the distinguished condition is the preferred approach. In this update, the normal and tangential stress balance conditions are satisfied during the flow solution. These equations enter the momentum equations through the boundary terms of equation (13b).

The normal stress can be replaced by equation (12) and the tangential stress by equation (11) to give

$$\int_{\Gamma_f} \varphi_k n_j T_{ij}\,ds = \int_{\Gamma_f} \varphi_k(-p_0 n_i - \sigma n_i n_{j,j})ds + \int_{\Gamma_f} \varphi_k \sigma_{,i}\,ds. \tag{17}$$

Note that in the above equation Γ_f is that part of the boundary where a free surface is present. The curvature in the normal stress balance has been replaced by the surface gradient of the normal vector. Using the surface divergence theorem[19]

$$\int_{\Gamma} (\sigma\varphi_k)_{,i}\,ds = \int_{\partial\Gamma} \sigma\varphi_k m_i\,ds + \int_{\Gamma} \sigma\varphi_k n_i n_{j,j}\,ds, \tag{18}$$

we can combine the equations to give

$$\int_{\Gamma_f} \varphi_k n_j T_{ij}\,ds = -\int_{\Gamma_f} \varphi_k p_0 n_i\,ds - \int_{\Gamma_f} \sigma\varphi_{k,i}\,ds + \int_{\partial\Gamma_f} \sigma\varphi_k m_i\,ds, \tag{19}$$

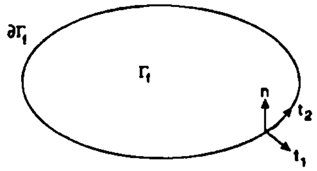

Figure 1. Representation of free surface boundary

where **m** is the outward-pointing normal of the boundary $\partial\Gamma_f$ of the free surface (see Figure 1). The integral on $\partial\Gamma_f$ is simply

$$\int_{\partial\Gamma_f} \sigma\varphi_k m_i \,\mathrm{d}s = \sigma\varphi_k m_i|_1 + \sigma\varphi_k m_i|_2 \tag{20}$$

in two dimensions and $\partial\Gamma_f$ is a line integral in a 3D geometry that ensures the satisfaction of contact angles in a weak sense. These equations are simply the 3D extension of those provided by Ruschak.[22] It should be noted that equation (19) also accounts for spatial gradients in surface tension.

To perform the update, we need to find the new location of the free surface where the kinematic constraint is satisfied in a weighted residual sense. In a weighted sense the kinematic constraint is then

$$\int_{\Gamma_f} \theta_k u_i n_i \,\mathrm{d}s = 0, \tag{21}$$

where θ is the weight function for free surface deformation. Since the integrand is a scalar product, we choose to evaluate it in a local tangential–normal co-ordinate system. To be sure that we do not make any assumptions about the shape of the free surface, we represent the surface in terms of the local tangential co-ordinate directions. If (x, y) represent the local tangential directions and z represents the normal direction, let us then assume the equation of the updated surface to be

$$z = g(x, y). \tag{22}$$

The normal to this free surface will be

$$\mathbf{n} = \frac{1}{\sqrt{(1 + g_x^2 + g_y^2)}}(-g_x, -g_y, 1). \tag{23}$$

If we represent the deformation g using the basis function δ, i.e.

$$g = \delta_i g_i, \tag{24}$$

and substitute this into the kinematic constraint, we get

$$\int_{\Gamma_f} \theta_k(\delta_{i,1}u_1 + \delta_{i,2}u_2)g_i = \int_{\Gamma_f} \theta_k u_3 \,\mathrm{d}s. \tag{25}$$

In the above equation (u_1, u_2) are the local tangential velocities and u_3 is the normal velocity. Since the constraint equation using equations (10) and (23) is a first-order partial differential equation, we use a streamline-upwinded Petrov–Galerkin formulation. The weight function is chosen as

$$\theta = \delta + \beta\mathbf{u} \cdot \nabla, \tag{26}$$

where $\beta = h/2|u|$. The mesh length h is the element size computed in the streamwise direction on the surface.

Notice that the predicted deformation is directly a function of the normal velocity at the current iterate. In matrix form this equation is simply

$$\mathbf{K}_f\mathbf{g} = \mathbf{f}, \tag{27}$$

where

$$\mathbf{K}_{\mathrm{f}} = \int_{\Gamma_f} (\theta_k \delta_{i,1} u_1 + \theta_k \delta_{i,2} u_2) \mathrm{d}s, \qquad \mathbf{f} = \int_{\Gamma_f} \theta_k u_3 \, \mathrm{d}s. \tag{28}$$

If there are any fixed contact lines, g is taken to be zero. The contact angle conditions are imposed as part of the stress boundary conditions (equation (19)). Since the kinematic constraint equation is a first-order partial differential equation, Dirichlet-type boundary conditions can be enforced at the inflow only. This observation is also supported by the study of Renardy and Renardy.[23]

3.4. *Normal stress update*

In problems where the surface tension forces dominate the viscous forces (i.e. $Ca \ll 1$), the distinguished condition is the normal stress balance. In this update the tangential stress balance and the condition of no normal flow are imposed as part of the flow solution. As before, the integration of the momentum equations generates on the boundary a tangential stress term

$$\int_{\Gamma_f} \varphi_k t_l n_i T_{ij} t_j \, \mathrm{d}s = \int_{\Gamma_f} \varphi_k \sigma_{,l} \, \mathrm{d}s, \tag{29}$$

where the comma is the surface gradient operator. This term can be evaluated in the local co-ordinate system as

$$\nabla_s \sigma = \mathbf{t}_1 \frac{\partial \sigma}{\partial x_1} + \mathbf{t}_2 \frac{\partial \sigma}{\partial x_2}. \tag{30}$$

The above stress is a consequence of non-uniform surface tension caused by temperature or surfactant distribution on the interface. The other boundary condition of zero normal flow can be invoked in a straightforward manner. However, as pointed out by Engelman *et al.*,[24] *consistent* normals should be evaluated to ensure conservation of global mass flow.

To determine the matrix equations for free surface deformation, we turn to the weighted residual form of the normal stress balance, which can be stated as

$$\int_{\Gamma_f} \delta_k (n_i T_{ij} n_j + p_0 - \delta H) \mathrm{d}s = 0. \tag{31}$$

If the free surface is represented as a function in the global co-ordinate system (e.g. $y' = f(x', z')$, then one can rewrite the curvature term using the surface divergence theorem as

$$\int_{\Gamma_f} \delta_k \sigma n_{i,i} = -\int_{\Gamma_f} n_i (\sigma \delta_k)_{,i} \, \mathrm{d}s + \int_{\partial \Gamma_f} \delta_k \sigma n_i m_i \, \mathrm{d}s. \tag{32}$$

Such a procedure is beneficial in that it lowers the order of the equations and an explicit calculation of the curvature is not needed. However, *a priori* knowledge of the surface is necessary to utilize such a procedure. Since the problems of interest do not always fall under this classification, we assume a local representation of the surface as described in the kinematic update.

Specifically, we seek a deformation

$$\mathbf{s} = g\mathbf{n} \tag{33}$$

in a direction normal to the current free surface iterate. The normal $\mathbf{n}'$ to this new perturbed surface is[19]

$$\mathbf{n}' = \mathbf{n} - \mathbf{n} \times \nabla \times \mathbf{s}. \tag{34}$$

Using (33) and (34) and vector identities, it can be shown that the curvature of the new surface (H') can be written in terms of the old curvature (H) as

$$H' = H + g_{,ii} - g(H^2 - 2K) + gg_{,i}H_{,i}, \tag{35}$$

where K, the Gaussian (or second) curvature, is the product of the principal curvatures of the surface. Further, if we make the assumption that the deformations are small, equation (35) can be linearized to

$$H' = H + g_{,ii} - g(H^2 - 2K). \tag{36}$$

Substituting this into the normal stress balance, we have

$$\int_{\Gamma_f} \delta_k\sigma[H + g_{,ii} - g(H^2 - 2K)]\mathrm{d}s = \int_{\Gamma_f} \delta_k(p_0 + n_iT_{ij}n_j)\mathrm{d}s. \tag{37}$$

Using the Green–Gauss theorem,

$$\int_{\Gamma_f} \delta_k\sigma g_{,ii}\,\mathrm{d}s = \int_{\partial\Gamma_f} \delta_k\sigma m_i g_{,i}\,\mathrm{d}s - \int_{\Gamma_f} (\sigma\delta_k)_{,i}g_{,i}\,\mathrm{d}s. \tag{38}$$

With equation (34) and the condition that at the boundaries of the new free surface iterate

$$m_in'_i = 0, \tag{39}$$

it can be shown that

$$m_ig_i = m_in_i, \tag{40}$$

which is simply a correction to the weighted residual equation when the contact angle boundary conditions are not satisfied. Finally, the weighted residual equation is

$$-\int_{\Gamma_f} (\sigma\delta_k)_{,i}g_i\,\mathrm{d}s - \int_{\Gamma_f} \sigma\delta_k g(H^2 - 2K)\mathrm{d}s = -\int_{\partial\Gamma_f} \sigma\delta_k m_in_i\,\mathrm{d}s + \int_{\Gamma_f} \delta_k(p_0 + n_iT_{ij}n_j - \sigma H)\mathrm{d}s. \tag{41}$$

It should be noted that the new deformation g is simply determined by the residual of the normal stress balance and the non-satisfaction of contact angle boundary conditions.

The curvature at any point can be computed by passing a circle through two adjacent points. This procedure has been adopted with success by Goodwin and Homsy.[25] If the curvature does not change rapidly, it can also be evaluated as the surface gradient of the normal vector. The normal at a node is non-unique (for the discretized surface), since the nodes are common to more than one element. A mean normal that best resembles the true normal to the continuous surface must be found. It can be approximated as an average of normals evaluated at integration points close to the node in the various elements. Since the elements are likely to have different surface areas too, it would be better to seek a weighted average based on the surface areas of these normals; such a weighted normal is simply the consistent normal in the terminology of Engelman *et al.*[24] Because these are evaluated for the imposition of zero normal flow, it would be computationally advantageous to use them for curvature calculations. As can be seen from the examples, such an approach provides for a satisfactory computation of curvature. The consistent normal at the midpoint of a quadratic element can be shown to be simply the geometric normal. Though such a value is satisfactory for the imposition of zero normal flow, it can give erroneous results for curvature calculations. To alleviate this difficulty, we approximate the midpoint consistent normal as an average of the normals of the other nodes that belong to the element.

We now proceed with the update procedure for normal stress iteration. If we let the free surface be represented by

$$z = g(x, y), \tag{42}$$

then one can invoke an equation for the deformation of the free surface as

$$\begin{aligned}\int_{\Gamma_f} [\delta_{i,1}(\sigma\delta_k)_{,1} + \delta_{i,2}(\sigma\delta_k)_{,2}]g_{,i}\,\mathrm{d}s + \int_{\Gamma_f} \sigma\delta_k\delta_i(H^2 - 2K)g_{,i}\,\mathrm{d}s \\ = \int_{\Gamma_f} \delta_k(n_iT_{ij}n_j + p_0)\mathrm{d}s + \int_{\Gamma_f} \delta_k\sigma H\,\mathrm{d}s - \int_{\partial\Gamma_f} \delta_k\sigma n_j m_j\,\mathrm{d}s.\end{aligned} \tag{43}$$

The first term on the right side of the equation is the normal stress on the free surface, the second term is the integral of capillary pressure and the last term is the influence due to the imposition of contact angle boundary conditions. The contact angle term is evaluated using the normal of the previous iterate and the proper tangent vector as specified by the contact angle. It should be noted that though this update is formulated for small deformations, no assumptions or linearizations have been made of the curvature terms. Thus the normal stress update is quite suitable for even large deformations of the free surface. However, since the curvature calculations are performed explicitly and low-order basis functions are used, problems which possess sharp changes in the curvature of the free surface may be difficult to model.

As part of the flow solution, pressure is computed only up to an additive constant. This pressure level can be computed in one of two ways. If there is a volume constraint on the fluid, the additional condition that the volume be conserved provides the estimate for the constant. The volume constraint statement is simply

$$\int_{\Gamma_f} g_k\,\mathrm{d}s = 0. \tag{44}$$

Combining this with equation (43), we can obtain the constant to correct the pressure from the flow solution. In problems of unbounded extent the asymptotic behaviour or far-field conditions provide the means of determining the constant. For example, in static capillary problems the pressure on the free surface in the farfield is known, since the curvature effects are negligible or known *a priori*. Using such a condition, one can determine the constant to correct the pressure.

3.5. Mesh adjustment after free surface update

The free surface updates using both kinematic and normal stress conditions provide the deformation in a direction normal to the current free surface. As a result, if one has access to an automatic mesh generator, the mesh can be updated after a few or every free surface update(s). This is the preferred way, since it allows for a smooth mesh to be generated after one or a few iteration(s). We have implemented two different approaches to mesh adjustment. First we used a spine-based mesh adjustment along the lines of Silliman and Scriven.[6] The mesh adjustment is performed by using the spine directions specified by the user. Since the free surface deformations are predicted along the local normal to the surface, as seen from Figure 2, the deformation is projected along the spine passing through the free surface node and the free surface node is moved to that location along the spine. The interior nodes on the spine are adjusted based on the relative grading of the original mesh. Such a method preserves the integrity of the mesh. The chief drawback of this method is that in three-dimensional problems or problems with moving contact lines on curved boundaries it is sometimes extremely difficult to set up spines to follow the direction of the free surface. As a second approach we remesh the regions adjacent to a moving free surface with an algebraic grid generator. We have found

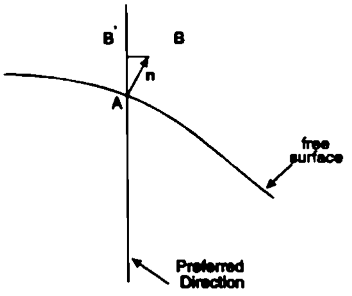

Figure 2. Free surface movement and preferred/spine directions

that the automatic mesh generator produces more efficient solutions, since the nodes are allowed to move in a direction locally normal to the free surface. At the walls or boundaries where the free surface have to move in a direction determined by the geometry, the component of the deformation along that direction is used to move the surface. In the next section we present a few test problems that demonstrate the capabilities of the free surface algorithm.

4. TEST PROBLEMS

4.1. 2D die swell

The die swell problem has essentially become the benchmark problem for testing external free surface flow problems. We have solved this problem with both kinematic and normal stress updates and compared our solutions (Tables I and II) with those of a Newton-based solver of FIDAP.[26] The tables present the comparison on the basis of the die swell rate at various Reynolds numbers and capillary numbers. The comparison shows excellent agreement with the solutions obtained with the Newton-based method. Besides the swell rate, the shape of the free surface also compared very well. The experimental results of Whipple and Hill[27] for this problem are for creeping flow at zero surface tension. The die swell under such circumstances has been observed to be around 20%, which is also in agreement with our simulations. The convergence criterion was based on the relative change in velocities, pressure and free surface deformation. Convergence is assumed to have been reached if all these relative changes are less than 10^{-4}. Typically 10–15 free surface updates are needed to reach this convergence criterion. In accordance with the findings of Silliman and Scriven,[6] the convergence rates are best when the kinematic update is used for $Ca \gg 1$ and the normal stress update for $Ca \ll 1$. In addition, relaxation factors are necessary to converge to the solution. These values are also given in the tables. It is also observed that a partially converged solution of flow variables (i.e. velocity tolerance less than 10^{-4}) between updates converged faster to the final free surface shape in most instances. When simulating creeping flows, such an approach can lead to a spurious solution of the free surface profile with a bulge near the exit section. However, if the flow solution is converged to a tight tolerance before performing free surface updates, such spurious solutions are always averted. Figures 3(a) and 3(b) show the typical convergence behaviour of the update schemes. The L^2-norm of the free surface deformation and the residual of the free surface distinguished condition are plotted as a function of the updates. The figures show a linear convergence rate for the two update schemes.

The computer storage requirements for the solution of this problem are as follows. With the computational domain discretized with 697 nodes, the Newton-based solver required 457,778 integer words to store the stiffness matrix with minimum and maximum bandwidths of 64 and 111 respectively. The segregated solver needed 74,696 integer words and had minimum and maximum

Table I. Comparison of die swell (%) for kinematic update

Re	*Ca*	Relaxation factor (segr.)	Segregated	Newton
10	∞	0	7·51	7·58
10	10	0·6	7·67	7·68
10	1	0·8	7·98	7·98
1	∞	0	19·01	18·99
1	10	0·5	18·21	18·53
1	1	0·8	13·92	13·85

Table II. Comparison of die swell (%) for normal stress update

Re	*Ca*	Relaxation factor (segr.)	Segregated	Newton
10	10^{-3}	0·5	0·09	0·11
10	0·1	0·8	4·71	4·69
10	0·5	0·97	7·98	7·98
1	10^{-3}	0·5	0·08	0·04
1	0·1	0·8	3·41	3·51
1	0·5	0·9	10·51	10·77

bandwidths of 27 and 37 respectively. Since the memory requirements are quite low for such a two-dimensional problem, the Newton-based solver was much quicker in terms of CPU time required to solve the problem to the same accuracy. However, the advantage of such segregated solvers will only be evident in large 2D or 3D problems.

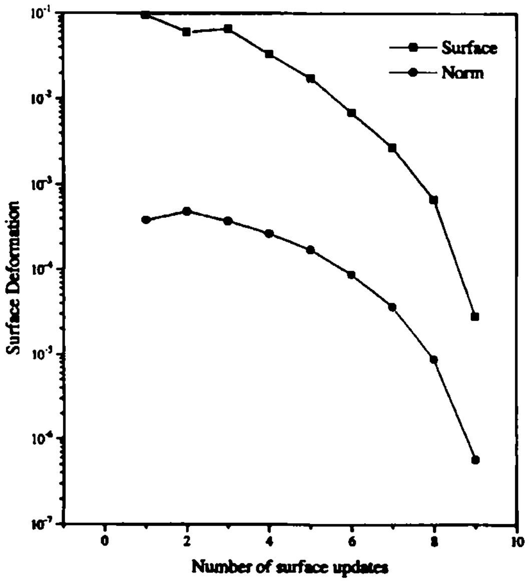

Figure 3(a). Convergence behaviour of kinematic update for $Re = 10$, $Ca = \infty$. The norm of the free surface deformation (surface) and normal velocity (norm) on the free surface are plotted as a function of free surface updates

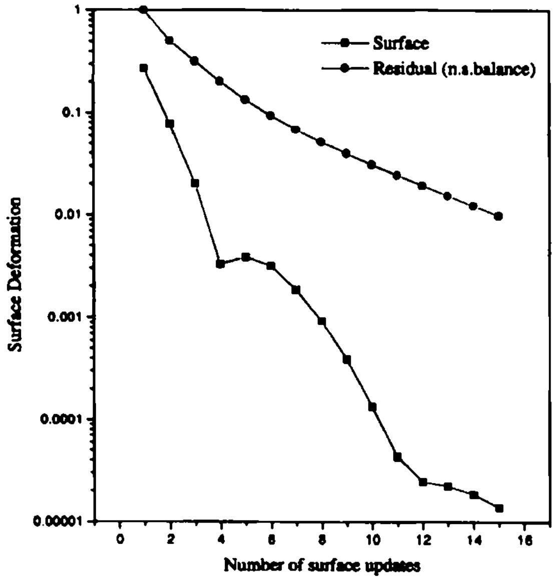

Figure 3(b). Convergence behaviour of normal stress update for $Re = 10$, $Ca = 0 \cdot 1$. The norm of free surface deformation and residual of normal stress balance on the free surface are plotted as a function of free surface updates

4.2. 3D die swell

The extrudate swell from a square die was chosen to show the applicability of the algorithm in three dimensions. Only a quarter-section of the die and extrudate is modelled owing to symmetry. The initial shape of the free surface assumed was simply a projection of the die as shown in Figure 4(a). The flow field in this section was computed with a coupled solver using successive substitution with the free surface treated as a stress-free boundary. The resulting velocity field was used as an initial flow field for the free surface problem. At the corners of the die the normals are computed as consistent normals, which for the initial free surface shape are simply along the diagonal of the square cross-section. The free surface is moved in the direction of the consistent normal during every update. The consistent normals are recomputed after the free surface has been updated. Using a relaxation factor of 0·5 for the free surface, convergence was attained in less than 15 free surface updates.

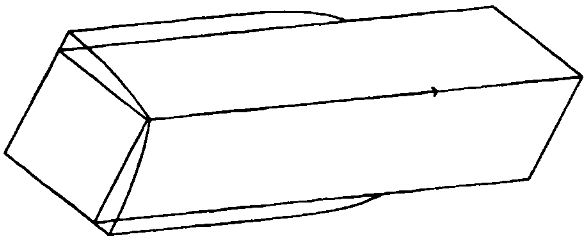

Figure 4(a). Initial and final free surface shapes for extrusion through a square die; $Re = 1$, $Ca = \infty$

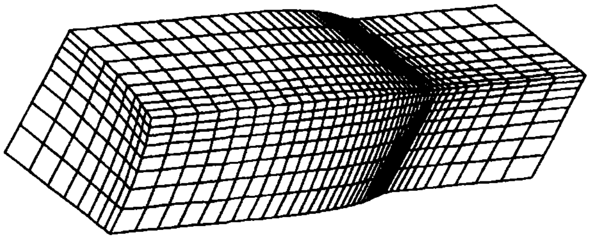

Figure 4(b). Final mesh for extrusion through a square die; $Re = 1$, $Ca = \infty$

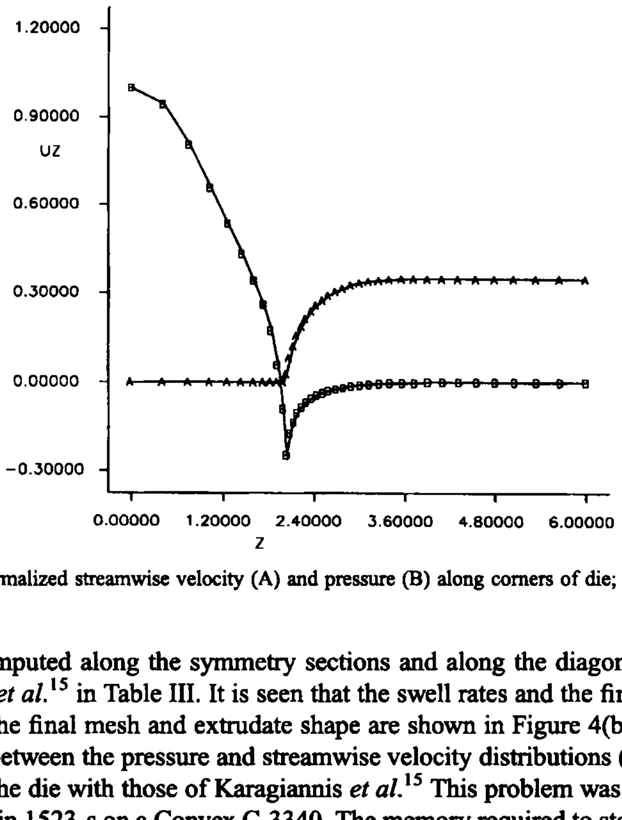

Figure 4(c). Normalized streamwise velocity (A) and pressure (B) along corners of die; $Re = 1$, $Ca = \infty$

The swell rates computed along the symmetry sections and along the diagonal are compared with those of Karagiannis *et al.*[15] in Table III. It is seen that the swell rates and the final free surface shapes compare very well. The final mesh and extrudate shape are shown in Figure 4(b). We have also found excellent agreement between the pressure and streamwise velocity distributions (shown in Figure 4(c)) along the corners of the die with those of Karagiannis *et al.*[15] This problem was discretized with 3159 nodes and converged in 1523 s on a Convex C-3340. The memory required to store the stiffness matrix was about $1 \cdot 1 \times 10^6$ integer words. Karagiannis *et al.*[15] modelled the problem with 5859 nodes. Their simulation took about 1150 s on a Cray X-MP 2/2. Their memory requirements were much higher than ours (estimated to be about 24×10^6 integer words) for 16,500 unknowns with a front width of about 726. It should be noted that out algorithm achieved the same accuracy as theirs with a smaller number of nodes, since we did not have the restriction of using spines.

4.2. *Curtain coating*

In this example we show the utility of the algorithm to solve problems with dynamic contact lines. The curtain coater is a classic example of a dynamic contact line problem that has been studied in detail by Kistler and Scriven.[11] For this test case we choose $Ca = 10$, $Re = 2 \cdot 5$ and set the substrate

Table III. Comparison of swell rates of extrusion through a square die

Re	*Ca*	Swell rate (%) segregated	Swell rate (%) Karagiannis
1	∞	18·9 (centre), 3·1 (diag.)	18·4 (centre), 2·9 (diag.)
10	∞	16·8 (centre), 0·8 (diag.)	—

velocity and inflow velocity to be equal. In addition, we have imposed the dynamic contact angle at the location where the left free surface contacts the free surface to be 160°. The Navier slip model is used to describe the relationship between slip velocity and shear stress. The slip coefficient has been set to 0·01. Figure 5(a) shows the initial guess of the free surface shape. The free surface is allowed to move in the direction of the local normal, except near the dynamic contact line where the movement is specified to be only along the substrate. For reasons of numerical stability we performed the first 10 updates with a high relaxation factor of 0·95 for the free surface and then decreased it to 0·5 and resumed the computations. To obtain a converged solution, close to 50 free surface updates were necessary. The final mesh and free surface shape shown in Figure 5(b) compare very well with the solution reported by Kistler and Scriven.[11] The location of the left free surface which contains the dynamic contact point is compared with the numerical results of Kistler and Scriven[11] in Figure 5(c) and shows very good agreement.

4.4. 3D capillary rise on a cylinder

Static capillary problems can be solved with normal stress update for any value of surface tension. The capillary rise on a circular cylinder was solved as a three-dimensional problem to demonstrate this application (Figure 6). Huh and Scriven[28] obtained analytical solutions to this problem. They show that with the contact angle at 60° and the radius set to $0{\cdot}2a$, the capillary rise on the cylinder is $0{\cdot}189a$, where a is the capillary length. Our numerical solution indicates a rise of $0{\cdot}190a$ on the cylinder. The far boundary was set at a radius of $3a$ and was assumed to be flat and fixed. The measured contact angle on the cylinder was 61°.

4.5. Thermocapillary convection in a liquid bridge

This problem is a model of the floating zone process used in crystal growth. The fluid of interest is held between two circular plates maintained at a constant temperature of zero in non-dimensional units. The axis of the liquid bridge that is formed between the two plates coincides with the direction of gravitational acceleration. The fluid is assumed to be present in an environment where it is heated by heat sources whose influence is simulated by a parabolic ambient temperature distribution with a maximum temperature of one in the middle of the liquid bridge. The free surface of the liquid bridge is the outer boundary of the cylinder. The problem is axisymmetric; however, to demonstrate the 3D capabilities, we have solved the flow and free surface shape in a quarter-section of the cylinder.

The flow inside the fluid is driven by a combination of natural convection and thermocapillary convection. The latter develops owing to the heating from the ambient, which sets up a temperature distribution on the free surface and hence a spatial variation in surface tension. The gradients in surface tension then induce a tangential stress on the free surface, causing the fluid to flow in the direction of larger surface tension. The free surface shape is affected primarily by the body forces related to gravitational effects and in small part to the viscous stresses of the fluid. The axisymmetric version of this problem was solved by Zhang and Alexander[29] for the following parameter set: $Bi = 10$, $Pr = 4{\cdot}667$, $Re = 107$, $Ca = 1/70$, $Gr = 2{\cdot}7$ and $G = 0{\cdot}468$. Here Bi is the Biot number, Pr is the

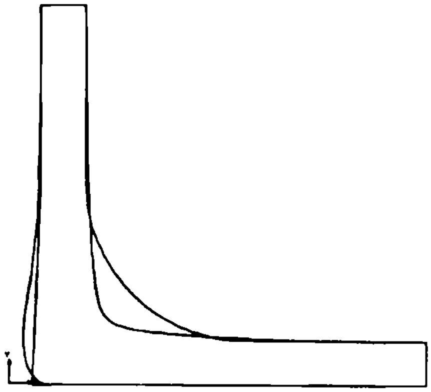

Figure 5(a). Initial and final free surface shapes of curtain coater; $Re = 2{\cdot}5$, $Ca = 10$, $U/V = 1$

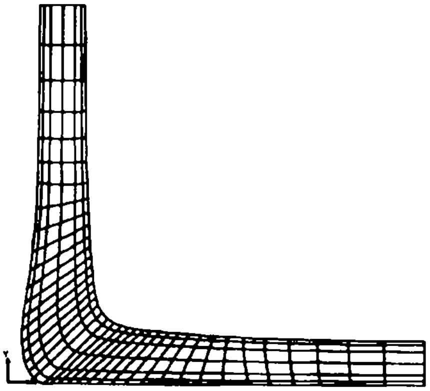

Figure 5(b). Final mesh and free surface shape of curtain coater; $Re = 2{\cdot}5$, $Ca = 10$, $U/V = 1$

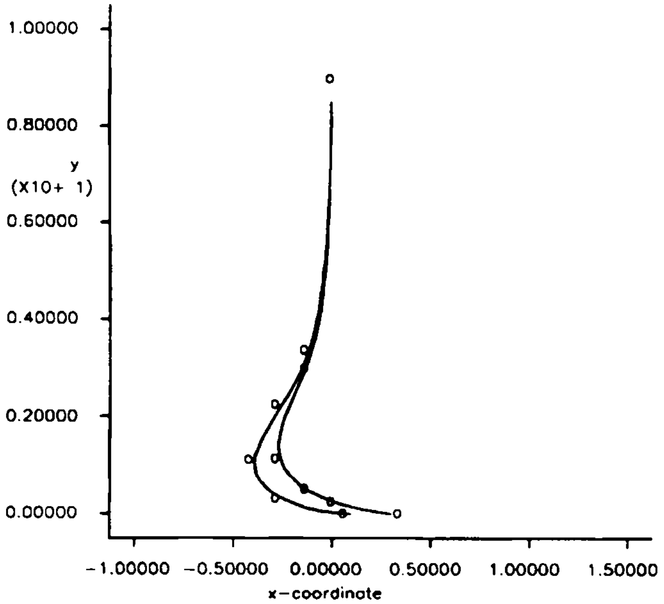

Figure 5(c). Comparison of left free surface with simulations of Kistler and Scriven[11] (o) for $Ca = 10$, $Ca = 3$ ($Re = 2{\cdot}5$, $U/V = 1$)

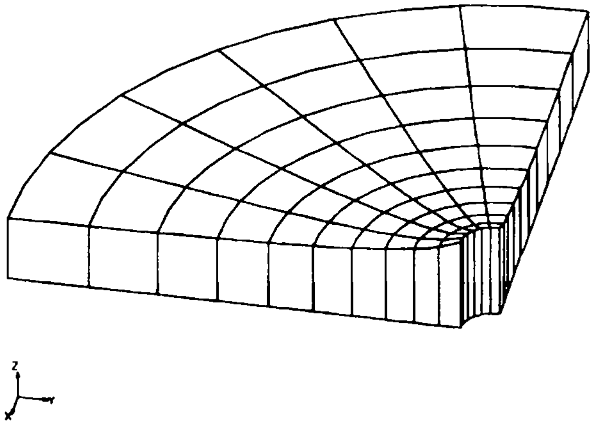

Figure 6. Final mesh of static capillary rise on a cylinder; $a = \sqrt{5}$, contact angle is 60°

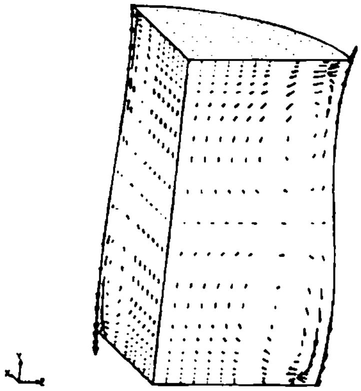

Figure 7(a). Velocity vectors of flow due to thermocapillary convection in a liquid bridge; $Re = 107$, $Pr = 4{\cdot}667$, $Ca = 1/70$, $Bi = 10$, $Gr = 2{\cdot}7$, $G = 0{\cdot}468$

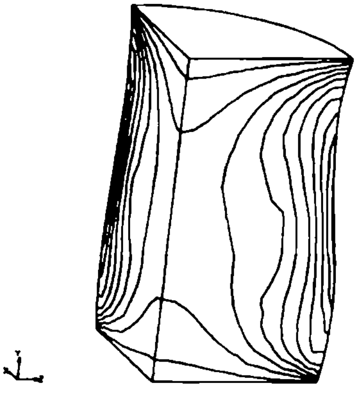

Figure 7(b). Isotherm contours of thermocapillary convection in a liquid bridge

Prandtl number, *Re* is the Reynolds number, *Ca* is the capillary number, *Gr* is the Grashof number and *G* is the dimensionless gravitational acceleration.

Our numerical solution for the above parameter set is shown in Figures 7(a) and 7(b). The flow structure in Figure 7(a) can be explained as follows. The temperatures are hottest near the centre of the free surface and coldest at the location where the free surface meets the discs. As a result, the surface tension is higher near the discs than at the centre. This differential surface tension induces tangential stresses which cause the fluid to flow from the centre of the free surface towards the cold discs. The presence of natural convection weakens the top cell and strengthens the bottom cell, causing differing strengths of circulation for the two cells. Because the Prandtl number of the fluid is moderately large, the convection causes the isotherms to bend with the flow, resulting in the temperature distribution shown in Figure 7(b). It is also seen that the convection results in higher temperature gradients near the discs. Since the gravitational acceleration acts along the axis, the free surface bulges at the bottom and shrinks at the top, because the pressures are higher at the bottom. The flow structure and isotherm distribution compare qualitatively with the results reported by Zhang and Alexander.[29] The only quantitative information that can be used for comparison is the maximum temperature in the crystal melt, which according to our simulations is 0·747, very close to the value of 0·737 obtained by Zhang and Alexander.[29]

5. CONCLUSIONS

An algorithm for the simulation of 3D viscous free surface flows has been presented. The algorithm performs fixed point iteration to execute alternately flow and free surface updates until convergence is achieved. Two updates, namely kinematic and normal stress iterations are formulated to address the extremes of large and small capillary numbers. It is observed that the contact angle boundary conditions and influence of surface tension gradients can be handled quite naturally through a conventional Galerkin finite element method. The free surface is represented using a local co-ordinate system, with the result that general surface shapes can be addressed. The algorithm predicts deformations in a direction locally normal to the previous iterate's shape and can work quite naturally with an automatic mesh generator. When the normal stress balance is uncoupled from the flow solution, the pressures can only be solved with respect to an arbitrary pressure constant. This pressure level can be identified either through a volume constraint or from a far-field condition. Examples of extrudate swell in 2D and 3D, curtain coating, capillary rise on a cylinder and thermocapillary convection in a liquid bridge have been provided to show applicability. Currently, in the next phase of this study, we are extending this algorithm to the simulation of transient flows.

REFERENCES

1. R. E. Nickell, R. I. Tanner and B. Coswell, *J. Fluid Mech.*, **65**, 189–206 (1974).
2. F. H. Harlow and J. E. Welch, *Phys. Fluids*, **8**, 2182 (1965).
3. C. W. Hirt, B. D. Nichols and N. C. Romero, 'SOLA—a numerical solution algorithm for transient fluid flows', *Los Alamos Scientific Laboratory Rep. LA-5852*, 1975.
4. B. D. Nichols and C. W. Hirt, *J. Comput. Phys.*, **9**, 434 (1971).
5. C. W. Hirt and B. D. Nichols, *J. Comput. Phys.*, **39**, 201–225 (1981).
6. W. J. Silliman and L. E. Scriven, *J. Comput. Phys.*, **34**, 287–313 (1980).
7. C. S. Fredericksen and A. M. Watts, *J. Comput. Phys.*, **39**, 282–304 (1981).
8. R. Keunings, *J. Comput Phys.*, **62**, 199–220 (1986).
9. F. M. Orr, R. A. Brown and L. E. Scriven, *J. Colloid. Interface Sci.*, **60**, 137–147 (1977).
10. H. Saito and L. E. Scriven, *J. Comput. Phys.*, **42**, 53–76 (1981).
11. S. F. Kistler and L. E. Scriven, *Int. j. numer. methods fluids*, **4**, 207–229 (1984).
12. H. S. Kheshgi and L. E. Scriven, in *Finite Elements in Fluids*, Vol. 5, Wiley, Chichester, 1984.
13. D. J. Coyle, C. W. Macosko and L. E. Scriven, *J. Fluid Mech.*, **171**, 183–207 (1986).
14. M. S. Engelman and R. L. Sani, *Numer. Methods Lamin. Turbul. Flow*, **3**, 389–400 (1084).

15. A. Karagiannis, A. N. Hrymak and J. Vlachopoulos, *AIChE J.*, **34**, 2088–2094 (1988).
16. C. Cuvelier and R. M. S. M. Schulkes, *SIAM Rev.*, **32**, 355–423 (1990).
17. L. W. Ho and A. T. Patera, *Comput. Methods Appl. Mech. Eng.*, **80**, 355–366 (1990).
18. B. Ramaswamy and M. Kawahara, *Int. j. numer. methods fluids*, 7, 1053–1075 (1987).
19. C. E. Weatherburn, *Differential Geometry of Three Dimensions*, Cambridge University Press, London, 1927, p. 239.
20. V. Haroutunian, M. S. Engelman and I. Hasbani, *Int. j. numer. methods fluids*, **17**, 323–348 (1993).
21. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
22. K. J. Ruschak, *Int. j. numer. methods eng.*, **15**, 639–648 (1980).
23. M. Renardy and Y. Renardy, *J. Comput. Phys.*, **93**, 325–335 (1991).
24. M. S. Engelman, R. L. Sani and P. M. Grasho, *Int. j. numer. methods fluids*, **2**, 225–238 (1982).
25. R. Goodwin and G. M. Homsy, *Phys. Fluids A*, **3**, 515–528 (1991).
26. M. S. Engelman, *FIDAP 7.0 Manuals*, Fluid Dynamics International, Evanston, IL, 1993.
27. B. A. Whipple and C. T. Hill, *AIChE J.*, **24**, 664 (1978).
28. C. Huh and L. E. Scriven, *J. Colloid Interface Sci.*, **30**, 323–337 (1969).
29. Y. Zhang and I. Alexander, *Int. j. numer. methods fluids.*, **14**, 197–215 (1992).