INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **33**: 375–403

A finite element method for free surface flows of incompressible fluids in three dimensions. Part I. Boundary fitted mesh motion

Richard A. Cairncross[a,*], P. Randall Schunk[b], Thomas A. Baer[b], Rekha R. Rao[b] and Phillip A. Sackinger[b]

[a] *Chemical Engineering Department, Drexel University, Philadelphia, PA, U.S.A.*
[b] *Sandia National Laboratories, Albuquerque, NM, U.S.A.*

SUMMARY

Computational fluid mechanics techniques for examining free surface problems in two-dimensional form are now well established. Extending these methods to three dimensions requires a reconsideration of some of the difficult issues from two-dimensional problems as well as developing new formulations to handle added geometric complexity. This paper presents a new finite element formulation for handling three-dimensional free surface problems with a boundary-fitted mesh and full Newton iteration, which solves for velocity, pressure, and mesh variables simultaneously. A boundary-fitted, pseudo-solid approach is used for moving the mesh, which treats the interior of the mesh as a fictitious elastic solid that deforms in response to boundary motion. To minimize mesh distortion near free boundary under large deformations, the mesh motion equations are rotated into normal and tangential components prior to applying boundary conditions. The Navier–Stokes equations are discretized using a Galerkin–least square/pressure stabilization formulation, which provides good convergence properties with iterative solvers. The result is a method that can track large deformations and rotations of free surface boundaries in three dimensions. The method is applied to two sample problems: solid body rotation of a fluid and extrusion from a nozzle with a rectangular cross-section. The extrusion example exhibits a variety of free surface shapes that arise from changing processing conditions. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: capillarity; extrusion; finite element method; free surface; pseudo-solid mesh motion; three dimensional

1. INTRODUCTION

More and more researchers and practitioners in the field of computational mechanics, heat transfer, and mass transfer are now solving three-dimensional models. Until recently, however,

* Correspondence to: Chemical Engineering Department, Drexel University, 32nd & Chestnut Streets, Philadelphia, PA 19104, U.S.A.

Received 26 October 1998
Revised 29 June 1999

problem solving in three dimensions has been noticeably lacking in many branches of mechanics, with most applications and attempts being exploratory rather than for practical design. This was particularly the case for incompressible computational fluid dynamics (CFD), with many prominent researchers opting for two-dimensional or axisymmetric expedients because of computer memory and speed limitations. Times are changing and, even for free and moving boundary problems in fluid dynamics, full three-dimensional analysis is becoming more common. Continued improvements in computer hardware are partly responsible, but other reasons exist. Algorithms that allow for complex three-dimensional boundary tracking have improved tremendously; solid model-based mesh generation has become more routine for moderately complicated three-dimensional models; and linear algebra and solver algorithm technology have been advanced so that new parallel computer architectures can be exploited. Although the authors firmly believe many interesting and unsolved two-dimensional free surface problems still exist (or at least ones that can be handled satisfactorily in two dimensions for the purpose intended), there are far more unsolved problems that require a full three-dimensional approach. The goal of attaining routine and efficient solutions of free and moving boundary problems in three dimensions may still be several years away, and in reaching it, robust and reliable algorithms for mesh movement and solver technology will be indispensable.

Of the several available and proven approaches to free and moving boundary problems in computational mathematics, perhaps the most challenging and most rewarding in terms of accuracy and wide spread applicability are based on boundary conforming moving/adapting meshes. Within this class the most popular techniques are often referred to as arbitrary Lagrangian–Eulerian (ALE) methods [1], which are based on the concept that the mesh moves in a fashion related to the kinematics of the material only over a portion of the domain, typically the boundary. Advances to ALE-based approaches have been numerous but curiously disjointed until 1991, when Soulaimani *et al.* [2] recognized that years of work in solving free boundary problems with finite element methods (FEM) and full Newton iteration led by Scriven *et al.* [3–5] are in essence the same as the much touted ALE approach, the only difference being that the latter were typically developed for moving boundary problems (i.e., transient), rather than free boundary problems (generally steady state). Recent reviews have appeared in the literature, which have covered all significant advances in moving and boundary adapting mesh techniques [6]. Example applications continue to appear, with the most significant developments in moving mesh/ALE techniques pertaining to three-dimensional problems, the subject taken up in this paper [7–10]. It should be noted here that outside the boundary fitted class of approaches of the sort addressed here, moving boundary problems have been solved with fixed grid Eulerian-boundary tracking (e.g., volume of fluid methods [11]) as well as with several different specialty methods (e.g., boundary element and vortex methods) for some time and even in three dimensions [6,12]. Although possessing many advantages over the chosen approach in many application areas, these alternative methods have serious disadvantages with respect to small scale viscous free surface flows of the sort on which this paper is focused.

Clearly most extensions to three dimensions of well-developed conventional two-dimensional algorithms are far from trivial to program, and are often costly to execute. The work presented here is no exception. The objective of this paper, and its follow-on [13], is to extend the

pseudo-solid domain mapping technique of Sackinger *et al.* [14] to three dimensions and further apply it to technologically important problems in manufacturing, culminating with an extrusion coating problem that includes a parameterized dynamic/static contact line. Many noteworthy precedents to this work exist and should be acknowledged. First and foremost is the work by Lynch *et al.* [15] who conceived the concept of the pseudo-solid domain mapping approach, which they applied in an explicit and segregated fashion to thermal problems. Second, the developments of Saito and Scrivan [3] with follow-on papers by Kistler and Scriven [4] and Christodoulon and Scriven [16] are noteworthy as that series of work brought together the powerful Newton–Raphson method with free boundary parameterization. Christodoulou and Scriven [5] set another precedent with a progression from algebraic-based mesh parameterization to differential equation-based mesh motion schemes, using the elliptic grid generation approach developed by Thompson *et al.* [17]. Many others have since applied the approaches of these landmark works. The most relevant to the present paper is the work of Wambersie and Crochet [18] who extended the spine-parameterization technique to three dimensions; the work of Karagiannis *et al.* [19] who made an early successful attempt at embedding that approach in Newton's method, in three dimensions even though computer hardware was so limiting then that no practical applications could be accomplished; and most recently, Catillo and Patera [8] who extended a fully parameterized mesh motion scheme to a technologically important roll coating problem in three dimensions, albeit with a segregated approach. Hanumanthu [20] used a novel combination of elliptic mesh generation and algebraic mesh motion to predict three-dimensional free surface in a film split of wire-wound rod coating.

Even with modern computer hardware there are still many research challenges to be met. First and foremost is to extend the concept of moving grid techniques to three dimensions; this becomes especially challenging on the boundaries. The second challenge involves differential geometry on three-dimensional surfaces; computing normal and tangent vectors consistently and in a way that allows for accurate accounting of surfaces; computing normal and tangent vectors consistently and in a way that allows for accurate accounting of surface curvature (e.g., capillary) effects, complex geometry, and contact line physics is much more difficult than in two dimensions. Perhaps the most daunting challenge is that our chosen approach—which offers many advantages, as pointed out throughout this paper—faces serious mathematical difficulties. Several are inherent in solving the Navier–Stokes system of equations for incompressible flows, regardless of the dimension, but become more severe in three dimensions. These include the incompressibility constraint that couples the velocity components with pressure, thereby precluding the use of purely explicit time marching procedures; and the viscous term leads to a broad eigenvalue spectrum, which makes the equation system stiff, and hence adds to the need for an implicit treatment. Perhaps the most challenging difficulties numerically are brought about by the free surfaces whose locations are unknown *a priori*, thus making the equation system severely non-linear, even when inertia is negligible and the liquid is Newtonian.

Even though there are many ways to circumvent these difficulties, no algorithm has been advanced that successfully copes, in a robust manner, with all them in three-dimensional free surface flows. Nonetheless, we feel the approach presented here significantly contributes to the state-of-the-art. The general physical theory describing free surface flows is presented first.

This includes the governing conservation laws for incompressible flows and the associated boundary conditions. Included is a description of our pseudo-solid mesh motion scheme, fully generalized for the three dimensions and large deformation.

The next section addresses the implementation of our scheme in a Galerkin finite element method (GFEM) framework. This section includes a brief description of the standard form of the Galerkin weak formulation, with a stabilized continuity equation. Owing to its favorable convergence properties with iterative solvers, we chose to employ the least-squares pressure stabilized Petrov–Galerkin form of the continuity equation [21]. This section also explains the application of boundary conditions to the fluid momentum and the mesh motion equations. Applying physical and geometric constraints on mesh motion while avoiding excessive mesh distortion is probably the most difficult part of the implementation. Our method rotates the mesh motion equations on surfaces and edges prior to boundary condition application to minimize artificial shearing of elements near boundaries. The third section discusses the methods used to solve the resulting non-linear algebraic equations with Newton's method. Here there were several challenges that had to be overcome, including the tuning of the iterative matrix solvers and the use of Newton's method and continuation to attain steady state solutions of highly deformed domains.

The final section of this paper addresses some simple three-dimensional applications that share many features with liquid phase manufacturing; more advanced applications are addressed in the second paper in this series. The first example is a fluid confined in a cup, which is spinning in solid body rotation. This is a simple example where our method agrees with the analytical solution. The second example is extrusion of a liquid out of a nozzle of square or rectangular cross-section. The shape and trajectory of the extruded jet/sheet varies with nozzle orientation, flow rate, surface tension, and gravity. Some results from the rectangular nozzle are suggestive of edge effects in curtain coating, where slow-moving fluid near the edge causes bending of the falling sheet.

The second paper in this series [13] extends our finite element formulation to handle solid/liquid contact lines in three dimensions. In two dimensions those lines manifest themselves as points and are relatively easy to track computationally. The data structure needed for a contact line in three dimensions, however, is non-trivial, being a collection of points that lies at the intersection of two geometry surfaces. In addition, the often-applied slip regions on the moving contact lines become a narrow strip of elements in three dimensions, necessitating difficult and skillful mesh design for routine analysis.

2. PHYSICAL THEORY

2.1. Navier–Stokes equations

Fluid flow within the domain is described by the incompressible Navier–Stokes equations in three dimensions for a Newtonian fluid. The equation of conservation of momentum in non-dimensional form with constant density and constant viscosity is

$$Re\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) = \nabla \cdot \boldsymbol{T} + St\boldsymbol{f} \tag{1}$$

The equation of conservation of mass, the continuity equation, is

$$\nabla \cdot \boldsymbol{u} = 0 \tag{2}$$

where

$$\boldsymbol{T} = -p\boldsymbol{I} + \nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T, \quad Re = \frac{\rho UL}{\mu}, \quad \text{and} \quad St = \frac{\rho g l^2}{\mu U} \tag{3}$$

$\boldsymbol{u}$ is the velocity vector, p is the pressure, $\boldsymbol{T}$ is the stress tensor, $\boldsymbol{I}$ is the identity tensor, ρ is density, U is a characteristic speed, L is a characteristic length scale, μ is the absolute viscosity, $\boldsymbol{f}$ is a unit vector in the direction of gravity, and g is the acceleration due to gravity. The Reynolds number, Re, and Stokes number, St, represent the relative magnitudes of inertia and gravity with respect to viscous forces. The pressure has units of the viscous pressure $p = p'L/(\mu U)$, where p' is the dimensional pressure.

2.2. Boundary conditions on the Navier–Stokes equations

Equations (1)–(3) apply everywhere within the fluid domain subject to conditions applied on the boundaries of the domain. Along solid boundaries a no-slip condition is applied, so the fluid velocity is zero, $\boldsymbol{u} = 0$. In cases where a solid wall is moving, then $\boldsymbol{u} = \boldsymbol{u}_{\text{wall}}$.

At the inflow and outflow planes, conditions of fully developed flow are applied

$$-\boldsymbol{n} \cdot \boldsymbol{T} = p_{\text{ext}}\boldsymbol{n} \tag{4}$$

$\boldsymbol{n}$ is an outward pointing unit normal to the boundary and p_{ext} is the external pressure acting on the boundary. In cases where the boundary has a vertical orientation, the external pressure varies in the vertical direction ($p_{\text{ext}} = p_0 + \rho g(z - z_0)$). While the inflow or outflow pressure only becomes hydrostatic at an infinite distance from the zone of flow rearrangement, it is a common approximation in practice and does not adversely affect the solution when the inflow and outflow planes are placed far enough upstream or downstream of the zone of flow rearrangement [6,16].

At free surfaces, mass and momentum must be conserved. In steady state free surfaces flows without evaporation, the free surface is a material surface and mass conservation requires that the fluid velocity normal to the surface must be zero

$$\boldsymbol{n} \cdot \boldsymbol{u} = 0 \tag{5}$$

This equation is normally called the kinematic boundary condition. Although it is a restriction on the fluid velocity at the free surface, it is customary to use it as a boundary condition on the mesh deformation [6]. In unsteady problems, the normal fluid velocity is equal to the normal motion of the boundary, $\boldsymbol{n} \cdot \boldsymbol{u} = \boldsymbol{n} \cdot \dot{\boldsymbol{x}}_{\text{fs}}$, where $\dot{\boldsymbol{x}}_{\text{fs}}$ is the velocity of the free surface.

At fluid–gas interfaces, surface tension causes a jump in stress across the interface. Conservation of momentum at free surfaces relates the internal stress to the external pressure, p_{ext}, via the capillary pressure, for constant surface tension

$$-\boldsymbol{n}\cdot\boldsymbol{T}=\boldsymbol{n}p_{\text{ext}}+2H\frac{1}{Ca}\boldsymbol{n} \tag{6}$$

H is the surface mean curvature and $Ca=\mu U/\sigma$ is the capillary number. In this equation, the external stress is assumed to be purely hydrostatic pressure, $p_{\text{ext}}\boldsymbol{I}$. For the general calculation of capillary pressure of three-dimensional surfaces, the mean curvature, H, can be related to the surface divergence of the normal vector [22]

$$2H=-\nabla_{\text{s}}\cdot\boldsymbol{n} \tag{7}$$

σ is the surface tension and $\nabla_{\text{s}}=(\boldsymbol{I}-\boldsymbol{nn})\cdot\nabla$ is the surface gradient operator, with $(\boldsymbol{I}-\boldsymbol{nn})$ the surface identity tensor and ∇ the bulk gradient operator. Further development of the equation is given in the appendix. Equation (6) is used as a boundary condition on the stress in the Navier–Stokes equations. At the outflow plane where the free surface exits the computational domain, surface tension exerts a force tangentially across the perimeter of the exit plane, $\boldsymbol{m}\sigma$. $\boldsymbol{m}$ is an outward pointing unit vector that is tangent to the surface and normal to the bounding curve along the edge of the surface; $\boldsymbol{m}$ is also called a binormal.

2.3. Free surface models with boundary filled mesh

As discussed in Section 1, we have chosen to use a computational mesh that conforms to the fluid domain for these computations. The edges of the mesh conform to both the free and fixed boundaries of the domain. Thus, the mesh must deform as the fluid surfaces deform. This section describes the theory we use to govern mesh motion.

Our approach corresponds to the ALE mesh motion formulation discussed in solid mechanics literature [14]. We treat the mesh as a fictitious elastic solid, which deforms in response to boundary loads. As the mesh boundary conforms to the domain occupied by the fluid, the mesh interior moves as though it was a compressible elastic solid. Such a method is called pseudo-solid mesh motion. The main advantages of this approach are that it is easily applied to unstructured three-dimensional meshes and that it provides an intuitive framework for understanding how the mesh deforms.

2.3.1. Pseudo-solid mesh motion. The basic premise of pseudo-solid mesh motion is that the initial mesh position is a stress-free state of the fictitious solid, $\boldsymbol{X}$. Then as the mesh boundaries move, the mesh deforms to a new stressed state, $\boldsymbol{x}$. The relationship between the stressed state and the stress-free state is the deformation field

$$\boldsymbol{x}=\boldsymbol{X}+\boldsymbol{d} \tag{8}$$

where $\boldsymbol{d}$ is the displacement field that represents how the material points in the pseudo-solid have deformed to accommodate fictitious elastic stresses. In numerical computations, the displacement field is the solution variable and the initial stress-free state field, $\boldsymbol{X}$, is predetermined by a standard mesh generation package. On the interior of the mesh, there is no explicit relationship between the displacement field and fluid motion.

Motion of the pseudo-solid in the mesh interior is governed by a quasi-static momentum equation

$$\nabla \cdot \boldsymbol{S} = 0 \tag{9}$$

where $\boldsymbol{S}$ is the Cauchy stress in the pseudo-solid. Of course, any form of the solid momentum equation is equally valid because this equation governs motion of a fictitious material. The form in Equation (9) has proven effective in two-dimensional calculations [14]. The pseudo-solid stress $\boldsymbol{S}$ is related to the deformation field through a constitutive equation. We have found that Hookean elasticity generally provides acceptable mesh motion behavior

$$\boldsymbol{S} = \lambda_s e \boldsymbol{I} + 2\mu_s \boldsymbol{E} \tag{10}$$

where λ_s and μ_s are the Lamé constants for the pseudo-solid, e is the volume strain, and $\boldsymbol{E}$ is the Eulerian strain tensor.

The choice of strain tensor, $\boldsymbol{E}$, turns out to be important to the quality of the solution obtained. The finite strain Eulerian strain tensor in terms of gradients in the displacement field is

$$\boldsymbol{E} = \frac{1}{2}[\nabla \boldsymbol{d} + \nabla \boldsymbol{d}^T - \nabla \boldsymbol{d} \cdot \nabla \boldsymbol{d}^T] = \frac{1}{2}[\boldsymbol{I} - \boldsymbol{F}^{-T}\boldsymbol{F}^{-1}] \tag{11}$$

The superscripts T, -1, and $-T$ refer to the transpose, inverse, and inverse-transpose of a tensor. In this equation all gradients of the displacement are in the deformed (Eulerian) state, but the deformation gradient, $\boldsymbol{F}$, is the Lagrangian deformation gradient with respect to the undeformed state ($\boldsymbol{F}^{-1}$ is the Eulerian deformation gradient with respect to the deformed state) [23]. Most previous attempts at pseudo-solid mesh motion utilized a small deformation, linear strain tensor, where in the small strain limit, the third term on the right-hand side is ignored. The volume change, or dilation, of the pseudo-solid is the determinant of the deformation gradient, $\boldsymbol{F}$.

$$\frac{V}{V_0} = \det|\boldsymbol{F}|, \quad \boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \boldsymbol{I} + \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{X}} \tag{12}$$

where V_0 is the volume in the undeformed stress-free state. This form is often called the Lagrangian deformation gradient because the derivatives are with respect to the undeformed co-ordinates. Then the volume strain is

$$e = 3(\det|\boldsymbol{F}|^{1/3} - 1) \tag{13}$$

This form reduces to $tr|\nabla \boldsymbol{d}|$ under small strains [24].

The use of the finite strain forms $\boldsymbol{E}$ and e are important for the prediction of flows in which the domain experiences large deformation and large rotation in particular. Figure 1 shows predictions of the flow of a two-dimensional sheet jet, which falls under gravity. With small

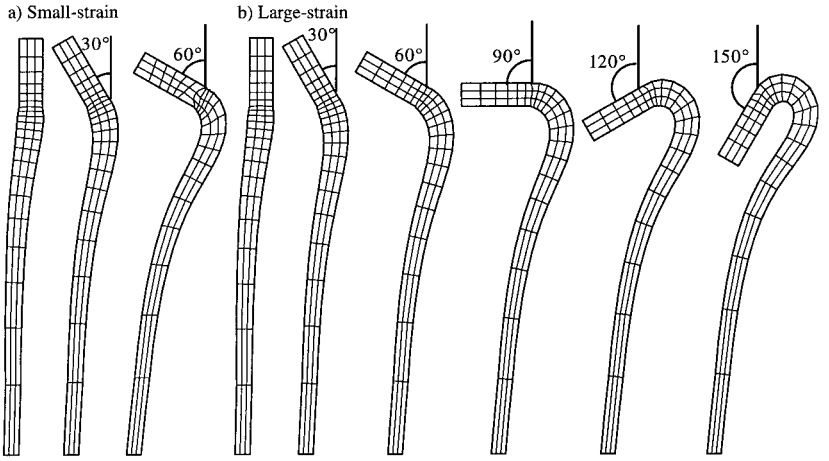

Figure 1. Comparison of (a) small strain mesh motion and (b) finite strain mesh motion for two-dimensional calculations of extrusion of a liquid sheet from a slot with $Re = 0.25$, $St = 0.05$, and $Ca = 70$. The small strain formulation creates undesirable mesh distortion when the nozzle is rotated 60° but the finite strain formulation allows rotation of 150° or more.

strain elasticity, the mesh breaks down before the nozzle is rotated 90°, but with finite strain elasticity the nozzle can rotate 150° or more without detrimental mesh distortion. In the small strain elasticity results with the nozzle rotated 60°, the elements near the nozzle exit become skewed as though non-uniform shear stresses were exerted on the mesh there. This shearing occurs because the small deformation strain tensor produces artificial stresses in a body under solid body rotation. Rotation of the nozzle cause non-uniform artificial shear stresses where the mesh bends sharply and results in undesirable mesh distortion. Use of the finite strain part of the strain tensor eliminates the artificial stresses due to rotation and alleviates the undesirable mesh distortion.

2.3.2. Boundary conditions on mesh motion. As mentioned above, the interior of the pseudo-solid deforms in response to boundary motion. Thus, boundary conditions on the mesh dictate the solution of the mesh motion problem. As the mesh momentum equation is a vector equation, three mesh boundary conditions must be applied on every surface of the mesh.

There are two main kinds of boundary conditions applied on mesh surfaces. Dirichlet conditions and distinguishing conditions. *Dirichlet conditions* specify the value of the displacement field at each node along a specified boundary; thus they specify both boundary shape and the distribution of nodes on the boundary. Normally, Dirichlet conditions are specified for all three displacement fields on a boundary, but this is not necessary. Sometimes only one or two boundary conditions are explicitly specified, but in such a case the remaining conditions are implicitly deemed shear-free conditions as discussed below (i.e., one could specify a Dirichlet condition on the vertical displacement and allow the mesh to be shear-free in the horizontal directions).

Distinguishing conditions are equations that prescribe the shape of the free surface but not the distribution of nodes along the surface. Common examples of distinguishing conditions are planar or cylindrical surfaces and other geometric shapes. An equation for a planar boundary is

$$ax + by + cz + d = 0 \tag{14}$$

a, b, c, and d are constants determining the location and orientation of the plane and x, y, and z are the Cartesian co-ordinates. The equation constrains the motion of the planar boundary in the normal direction, but the nodes should be able to redistribute tangentially in any fashion. Allowing the nodes to redistribute tangentially to minimize the pseudo-solid shear stresses on the boundary often gives acceptable mesh motion [14]

$$\boldsymbol{tn}\colon\ \boldsymbol{S} = 0 \tag{15}$$

where $\boldsymbol{t}$ is any tangent on the surface and $\boldsymbol{n}$ is the normal to the surface. Using the FEM discussed in the next section, this equation is automatically weakly satisfied when a mesh boundary condition is not explicitly specified.

The kinematic boundary condition of Equation (5) is also a distinguishing condition on mesh motion, i.e., the boundary moves to make the normal component of fluid velocity equal to zero

$$\boldsymbol{n} \cdot (\boldsymbol{u} - \dot{\boldsymbol{x}}_{\mathrm{fs}}) = 0 \tag{16}$$

This condition makes the free surface a material surface in the normal direction, i.e., no mass crosses the surface. Similar to geometric distinguishing conditions it is convenient to allow the nodes on a free surface to redistribute to minimize pseudo-solid stresses according to Equation (15).

At edges where two or more surfaces meet, the relative importance of the adjoining boundary conditions needs to be resolved. The normal hierarchy is that Dirichlet conditions take precedence over distinguishing conditions, and geometric conditions take precedence over free boundary conditions. However, care must be taken not to loose valuable boundary information. For example, when two planar boundaries intersect, the resulting edge is a straight line; in such a case both planar conditions still need to be applied to separate components of the mesh motion equation and the third component of the mesh motion equation allows tangential motion along this edge to minimize mesh shear stress. Details of this implementation are discussed below.

3. FINITE ELEMENT FORMULATION

3.1. Galerkin finite element formulation

To discretize the fluid momentum equation, the continuity equation, and the quasi-static equilibrium equation for the pseudo-solid mesh motion, we use the FEM. When the finite

element method is applied, the solution of the relevant set of continuum equations is approximated by a finite set of basis or shape functions multiplied by unknown nodal coefficients. The unknowns of interest are velocity, pressure, and mesh displacement. For viscous incompressible flow problems, the velocity and pressure fields interpolation must satisfy the Ladyzhenskaya–Babuska–Brezzi (LBB) condition that requires the velocity interpolant to be of higher order than the pressure interpolant [25,26]

$$\boldsymbol{u} \approx \boldsymbol{u}_h = \sum_{i=1}^{n} \boldsymbol{u}_i \phi_i(\boldsymbol{x}) \tag{17}$$

where $\boldsymbol{u}$ denotes the true velocity solution, $\boldsymbol{u}_h$ denotes the approximate finite element solution, $\boldsymbol{u}_i$ represents the velocity at the node i, and ϕ_i is the velocity basis function associated with node i.

Among the valid pairs of interpolating velocity and pressure, basis functions are quadratic for velocity with bilinear or linear discontinuous for pressure [27]. For two-dimensional problems, we use an approximation for the pressure shape function $\psi_i(\boldsymbol{x})$, which may be bilinear or linear continuous

$$p \approx p_h = \sum_{i=1}^{m} p_i \psi_i(\boldsymbol{x}) \tag{18}$$

No constraint analogous to the LBB condition exists for the mesh displacement interpolation functions since the pseudo-solid mesh deformation algorithm has the property of a compressible solid

$$\boldsymbol{d} \approx \boldsymbol{d}_h = \sum_{i=1}^{m} \boldsymbol{d}_i \phi_i(\boldsymbol{x}) \tag{19}$$

The trial functions (17)–(19) are then substituted into differential equations in place of the continuous solution to form residual solutions. For instance, the momentum residual, R^{m} is

$$R^{\mathrm{m}} = Re \frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} + Re(\boldsymbol{u}_h - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u}_h + \nabla \cdot [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)] - St\boldsymbol{f} \tag{20}$$

In Equation (20), the time derivative has been transformed to a total time derivative, which follows the motion of a finite element nodes. $\boldsymbol{u}_m$ is the velocity of the nodes, which is equal to the speed of nodal displacement

$$\boldsymbol{u}_m = \dot{\boldsymbol{x}} = \dot{\boldsymbol{d}} = \frac{\mathrm{d}\boldsymbol{d}}{\mathrm{d}t} \tag{21}$$

In steady state problems, the nodal velocity is zero everywhere, $\boldsymbol{u}_m = 0$.

The residual equations are then multiplied by weight functions and integrated over the domain to obtain a set of non-linear algebraic equations for the nodal coefficients. For GFEM, the weight functions are chosen to be the shape functions themselves. The nodal unknowns, $\boldsymbol{u}_i$, $\boldsymbol{d}_i$, and p_i, are then found such that the weighted residual equations are approximately zero. The vector momentum equation is converted to three scalar components by dot products with the co-ordinate basis vectors $\boldsymbol{e}_\beta$, $R^{\mathrm{m},\beta} = \boldsymbol{e}_\beta \cdot R^{\mathrm{m}}$. Each vector component is weighted separately so that weighted residuals are recovered for each component of the velocity vector, $\boldsymbol{u}_h^\beta$. This yields a volume integral over V of the form

$$\boldsymbol{R}_i^{\mathrm{m},\beta} = \int_V \phi_i \boldsymbol{e}_\beta \cdot \left(Re\left(\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} + (\boldsymbol{u}_h - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u}_h \right) + \nabla \cdot [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)] - St\boldsymbol{f} \right) \mathrm{d}V = 0 \tag{22}$$

There are three momentum residuals associated with every node. Because the stress term contains second derivatives of the velocity, the divergence of total stress in Equation (22) must be integrated by parts. This gives

$$\begin{aligned} R_i^{\mathrm{m},\beta} = & \int_V \phi_i \boldsymbol{e}_\beta \cdot \left(Re\left(\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} + (\boldsymbol{u}_h - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u}_h \right) - St\boldsymbol{f} \right) \mathrm{d}V \\ & - \int_V \nabla(\phi_i \boldsymbol{e}_\beta) : [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)] \, \mathrm{d}V + \int_A (\phi_i \boldsymbol{n} \boldsymbol{e}_\beta : [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)]) \, \mathrm{d}A = 0 \end{aligned} \tag{23}$$

This application of the divergence theorem creates surface integrals over the surface boundary, A. Neglecting these surface terms in the formulation results in the so-called natural boundary condition for the traction on the surface [28], e.g.

$$\int_A \phi_i \boldsymbol{n} \boldsymbol{e}_\beta : [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)] \, \mathrm{d}A = 0 \tag{24}$$

where $\boldsymbol{n}$ is the outward facing normal. The effect is to apply zero traction on this boundary in a weak integrated sense.

The continuity residual, R^{c}, is

$$R^{\mathrm{c}} = \nabla \cdot \boldsymbol{u}_h \tag{25}$$

and the corresponding weighted residual equations are recovered using the pressure basis functions as weights.

$$R_i^{\mathrm{c}} = \int_V \psi_i (\nabla \cdot \boldsymbol{u}_h) \, \mathrm{d}V = 0 \tag{26}$$

The pseudo-solid mesh displacement residual, R^{d}, like the fluid momentum equation, is also a vector equation of a simple form under the assumptions of no inertia, no relative pseudo-solid material motion, and no external forces

$$R^{\mathrm{d}} = \nabla \cdot \boldsymbol{S} \tag{27}$$

where $\boldsymbol{S}$ is the total stress tensor for the pseudo-solid response. We weight this residual with the mesh displacement basis function, ϕ_i, and again use the divergence theorem

$$R_i^{\mathrm{d}} = \int_V \nabla(\phi_i \boldsymbol{e}_\beta) : \boldsymbol{S} \, \mathrm{d}V - \int_A (\phi_i \boldsymbol{n}\boldsymbol{e}_\beta : \boldsymbol{S}) \, \mathrm{d}A = 0 \tag{28}$$

Equations (23) and (26) yield the classical Galerkin finite element formulation of the incompressible Navier–Stokes equations and require the use of an LBB compliance element. In the following section, we will give the equations for the pressure-stabilized Galerkin least-squares (GLS) form of the flow equations that allow circumvention of the LBB requirement and equal-order interpolation of velocity and pressure. Here we only give a brief description of the method, the details can be found in the work Hughes and others [29,30].

3.2. Galerkin least-squares/pressure stabilization formulation

The standard finite element formulation weights the momentum equation and the continuity equation with the shape function used to describe the interpolation of the variables. When pressure stabilization is used, the momentum equation is projected into the gradient of the Galerkin weight function and added onto the weighted residual of the continuity equation. This gives a non-zero diagonal term in the continuity equation and results in a modified continuity residual of the form

$$\begin{aligned} R_i^{\mathrm{c}} = & \int_V \psi_i [\nabla \cdot \boldsymbol{u}_h] \, \mathrm{d}V \\ & + \int_V \tau_{pspg} \nabla \psi_i \cdot \left[Re \left(\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} + (\boldsymbol{u}_h - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u} \right) + \nabla \cdot [p\boldsymbol{I} - (\nabla \boldsymbol{u}_h + \nabla \boldsymbol{u}_h^T)] - St \boldsymbol{f} \right] \mathrm{d}V = 0 \end{aligned} \tag{29}$$

We define τ_{pspg} following the work of Tezduyar [30] but with a slight variation since we want it to be constant throughout the entire domain, while Tezduyar used an element-level constant value. τ_{pspg} is defined differently depending on the average Reynolds number of the flow

$$Re_{\mathrm{avg}} = \frac{\rho_{\mathrm{avg}} \| \boldsymbol{u}_h \| h_{\mathrm{avg}}}{\mu_{\mathrm{avg}}} \tag{30}$$

Written in terms of the average element size h_{avg}, the global L_2 velocity norm and the average viscosity and density, τ_{pspg} is

$$\tau_{pspg} = \frac{c}{12} \left(\frac{h_{\mathrm{avg}}}{L} \right)^2 \frac{\mu}{\mu_{\mathrm{avg}}}, \quad Re_{\mathrm{avg}} \leq 3 \tag{31}$$

$$\tau_{pspg} = \frac{c}{2\,Re_{\text{avg}}} \frac{h_{\text{avg}}}{L} \frac{\mu}{\mu_{\text{avg}}}, \quad Re_{\text{avg}} > 3 \tag{32}$$

where c is a positive scalar parameter that can be varied.

The viscous stress term in Equation (29) must be handled in a special way, because it will be zero for bilinear interpolation [29]. By including the incompressibility constraint, we may write this term as

$$R_i^{\text{c}}[\nabla^2 \boldsymbol{u}] = -\int_V \tau_{pspg} \nabla \varphi_i \cdot [\nabla^2 \boldsymbol{u}_h] \, \mathrm{d}V \tag{33}$$

and after integrating it by parts, we have

$$R_i^{\text{c}}[\nabla^2 \boldsymbol{u}] = -\int_V (\tau_{pspg} \varphi_i \nabla \cdot [\nabla^2 \boldsymbol{u}_h]) \, \mathrm{d}V - \int_A (\tau_{pspg} \varphi_i \boldsymbol{n} \cdot [\nabla^2 \boldsymbol{u}_h]) \, \mathrm{d}A \tag{34}$$

Under the assumption of constant viscosity, the first term on the right-hand side of Equation (34) is a variation of the divergence of velocity, which should go to zero weakly as the incompressibility constant is satisfied. The velocity diffusion term, $\nabla^2 \boldsymbol{u}$, can be found from the balance of the momentum equation, which contains terms with first derivatives of the velocity only

$$\nabla^2 \boldsymbol{u} = Re\left(\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} + (\boldsymbol{u} - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u}\right) + \nabla p - St\,\boldsymbol{f} \tag{35}$$

Using this information, we can rewrite Equation (34) as

$$R_i^{\text{c}}[\nabla^2 \boldsymbol{u}_h] = \int_V \tau_{pspg} \varphi_i \nabla^2 [\nabla \cdot \boldsymbol{u}_h] \, \mathrm{d}V - \int_A \tau_{pspg} \varphi_i \boldsymbol{n} \cdot \left[Re\left(\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} + (\boldsymbol{u}_h - \boldsymbol{u}_m) \cdot \nabla \boldsymbol{u}_h\right) + \nabla p - St\,\boldsymbol{f} \right] \mathrm{d}A \tag{36}$$

The first term on the right-hand side of Equation (36) is assumed to be identically zero for incompressible flows and is therefore ignored. The second term on the right-hand side is added to the continuity equation only on the domain boundaries, since it cancels out at inter-element boundaries when continuous interpolation of the field variables is used and the τ_{pspg} term is assumed to be constant. If discontinuous variables are used, special care must be taken to include the jump variables between inter-element boundaries [21]. We choose τ_{pspg} as a global variable, so that it is consistent throughout the domain and ensures that the boundary integral cancels out at inter-element boundaries. Practically, the second term has non-trivial contributions only on inflow and outflow boundaries.

It is important to note that if the boundary term is not included for Stokes flow, the solution will be incorrect for the inflow and outflow boundaries. For Stokes–Poiseuille flow, this boundary term eliminates the oscillations in the velocity and pressure fields near the inflow and outflow boundaries [29].

3.3. Application of boundary conditions to finite element formulation

Both the fluid momentum and the pseudo-solid momentum equations are vector equations and require *explicit* or *implicit* specification of three boundary conditions, one for each dimension, on all surfaces, edges, and vertices. The continuity equation is a scalar equation, which enforces continuity of mass and does not require any boundary conditions. The momentum equations are resolved into individual components via the basis vectors $\boldsymbol{e}_\beta$ as shown above; so each component equation balances momentum in the given basis direction. The application of boundary conditions in a numerical scheme either specifies the boundary term in the momentum balance (a *weak* boundary condition) or replaces the momentum balance with another constraint (a *strong* boundary condition). In the latter case, the constraint is penalized by a large number (10^{12}) and added to the appropriate equation, effectively replacing the equation in favor of the constraint. Note that the terms *strong* and *weak* are merely labels for two methods of applying boundary conditions; they do not imply that a strong boundary condition is enforced any more 'strongly' than a weak boundary condition, or vice versa.

3.3.1. Boundary conditions on fluid momentum equations. For the problems in this paper, the boundary conditions on the fluid momentum equations are straightforward to apply. Dirichlet conditions replace the equation associated with the component to which they apply

$$R_i^{\mathrm{m},\beta} = v_i^\beta - v_{\mathrm{DC}}^\beta \tag{37}$$

where v_{DC}^β is the velocity applied to component β at node i.

The capillary and hydrostatic pressure conditions are weak conditions and modify all three components of the momentum balance by replacing the boundary traction integral in Equation (23)

$$R_i^{\mathrm{m},\beta} = \cdots + \int_A \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{in}\, \mathrm{d}A \tag{38}$$

$\boldsymbol{T}_{in}$ is the total stress in the fluid at the boundary. A pressure boundary condition replaces $-\boldsymbol{n}\cdot\boldsymbol{T}_{in}$ by p_{ext}^n, where p_{ext} is the pressure external to the boundary; because p_{ext} is known, this integral is easily evaluated. The capillary stress boundary condition (6) replaces $-\boldsymbol{n}\cdot\boldsymbol{T}_{in}$ by $\boldsymbol{n}p_{\mathrm{ext}} + 2H\boldsymbol{n}/Ca$ for a constant surface tension. Evaluation of the capillary terms requires integration by parts to remove second derivatives of the surface position. Details of the derivation are given in Appendix A, but the result is that the boundary traction integral becomes [20,31]

$$\begin{aligned} -\int_A \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{in}\, \mathrm{d}A = {} & \int_A \phi_i (\boldsymbol{e}_\beta \cdot n) p_{\mathrm{ext}}\, \mathrm{d}A - \int_A \frac{1}{Ca} (\boldsymbol{I} - \boldsymbol{nn}) : \nabla(\boldsymbol{e}_\beta \phi_i)\, \mathrm{d}A \\ & + \int_C \phi_i \frac{1}{Ca} (\boldsymbol{e}_\beta \cdot \boldsymbol{m})\, \mathrm{d}C \end{aligned} \tag{39}$$

This integral replaces the boundary integral in Equation (23). The second term on the right-hand side represents the capillary pressure and surface tension gradients on the surface, and the last term represents the surface tension exerted along the edges of the free surface, C. In the extrusion example problem discussed later, this last term is only applied along the perimeter of the outflow plane where the free surface leaves the domain.

3.3.2. Boundary conditions on mesh equations. Two mesh distinguishing conditions are used in this paper to constrain and deform the mesh. For nodes on planar boundaries, a mesh equation at the node is replaced by the plane condition

$$R_i^{\mathrm{d},n} = ax_i + by_i + cz_i + d \tag{40}$$

where x_i, y_i, and z_i represent the co-ordinates of the nodal point. Equation (36) may also be expressed equivalently as

$$R_i^{\mathrm{d},n} = \boldsymbol{n}_p \cdot \boldsymbol{d}_i \tag{41}$$

where $\boldsymbol{n}_p$ is the unit normal to the plane and $\boldsymbol{d}_i$ is the vector of displacements of that node. The superscript d,n indicates that this equation replaces the normal component of the displacement equation for node i. This distinguishing condition is applied as a strong boundary condition.

For nodes on free surface boundaries, a mesh equation at that node is replaced by a weighted residual of the kinematic condition

$$R_i^{\mathrm{d},n} = \int_A \phi_i \boldsymbol{n} \cdot (\boldsymbol{u} - \dot{\boldsymbol{x}}_{\mathrm{fs}})\, \mathrm{d}A \tag{42}$$

It is likewise applied to the mesh equation at node i as a strong boundary condition.

An important distinction to note is that the planar condition is evaluated only at the nodal points. In this sense, it is a strong collocated boundary condition. In contrast, the kinematic condition is obtained by integration over the surfaces that share each node. Hence, it is a strong integrated condition.

3.3.3. Rotation of vector equations prior to boundary condition application. The boundary conditions on the pseudo-solid momentum equations are often difficult to apply because these distinguishing conditions can apply on arbitrarily oriented surfaces. Spurious mesh stresses can appear if the surface projection of the distinguishing condition is non-zero. Therefore, it is important that the distinguishing condition constrains only the normal motion of the surface and permits the nodes to redistribute freely in the tangential direction. In this case it is convenient to express the pseudo-solid momentum equation in normal and tangential components. The normal component is replaced by the distinguishing condition, and the tangential components allow shear-free distribution of the nodes. Sackinger *et al*. [14] demonstrated the effectiveness of this procedure for two-dimensional problems.

A general vector equation contains three components, R_x, R_y, and R_z, in Cartesian co-ordinates. The normal component in this vector equation is equal to the dot product of the unit vector normal to the surface and the vector equation; similar operations are performed for the surface tangent vectors. Then, the vector equation rotated into normal and tangential form becomes

$$\begin{bmatrix} R_i^n \\ R_i^{t1} \\ R_i^{t2} \end{bmatrix} = \begin{bmatrix} \boldsymbol{n}_i^T \\ \boldsymbol{t}_{1,i}^T \\ \boldsymbol{t}_{2,i}^T \end{bmatrix} \begin{bmatrix} R_i^x \\ R_i^y \\ R_i^z \end{bmatrix} \tag{43}$$

$\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are the two perpendicular tangents in the surface. In general, the rotation procedure can be performed before or after integration of the volumetric terms of the equations. For simplicity we have chosen to rotate the equations after numerical integration. This may introduce some spurious tangential shear stress into the mesh, but the effect is small and disappears with refinement. Furthermore, the exact tangential position of the nodes is not physically relevant (it is a pseudo-solid) and this rotation scheme produces nearly shear-free nodal distribution.

In application, both the plane and kinematic boundary condition replace the normal component of the pseudo-solid mesh equation, R_i^n. The tangential components are retained to allow the mesh to minimize the shear stress at the boundary. The surface traction integral in the tangential components results in a shear-free surface boundary appearing as a natural boundary condition. In simple situations, the choice of which of the three mesh momentum equations to replace by the boundary condition is easy because the surface lines up with a co-ordinate plane (i.e., if the surface were a plane facing in the y-direction, the plane boundary condition would replace the y component of the mesh momentum equation). Such simple rules break down in general cases, especially when the boundaries rotate during the computation. Note too, that *ad hoc* procedures for choosing different natural co-ordinates based on the instantaneous surface orientation can lose continuous differentiability and therefore lose the quadratic convergence of Newton iteration.

An additional problem occurs in the three-dimensional case, which is especially evident when using unstructured grids: the normal–tangential rotation of the equations is not unique because all tangents in the plane of the surface are equally valid. Thus, for efficient numerical computation, the tangents in three-dimensional form must be chosen using a protocol that is independent of which element or node is currently active and is independent of the orientation of the surface. The rotation in Equation (43) is performed individually for each node, i.e., each node has an associated vector of mesh displacements and pseudo-solid momentum equations and these equations are rotated with respect to a normal and two tangents associated with that node. A goal of this rotation algorithm is to calculate a normal vector and two tangent vectors for each surface node.

On the surface of an element there are two local elemental co-ordinates, ξ and η, as shown in Figure 2. The normal to the surface is the normalized cross-product of the change in physical co-ordinates in these two directions

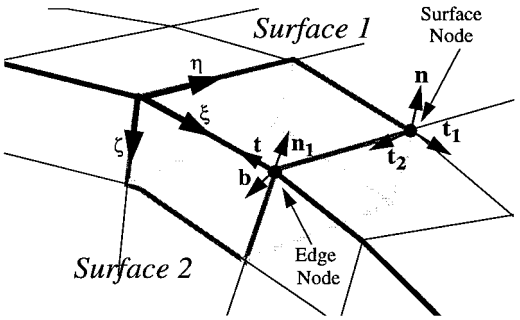

Figure 2. Schematic of normal and tangent vectors at surfaces and edges of an element. Local element computational coordinates are labeled ξ, η, and ζ.

$$\boldsymbol{n} = \frac{\dfrac{\partial \boldsymbol{x}}{\partial \xi} \times \dfrac{\partial \boldsymbol{x}}{\partial \eta}}{\left\| \dfrac{\partial \boldsymbol{x}}{\partial \xi} \times \dfrac{\partial \boldsymbol{x}}{\partial \eta} \right\|} \tag{44}$$

Thus, a surface normal is easy to calculate for a given element side. However, the normal associated with a given node will vary depending on which adjacent element is used to calculate the surface normal. For a given surface node, we use an arithmetic average of the normal calculated from all the adjacent elements.

Likewise, the calculated tangent vectors can vary from element to element. Furthermore, with unstructured meshes, the local elemental co-ordinates in adjacent elements do not necessarily coincide. That is, the local element co-ordinates of a node in a given element may be completely different from those of the same node in a different adjacent element. Thus, a robust algorithm is needed to obtain reproducible tangent vectors for each node on a three-dimensional surface. We have found three methods to be useful in various situations, (1) generation of tangents from seed vectors, (2) using the local element co-ordinates from only one element, and (3) combining the basis vector and seed vector approaches to average the tangents from adjacent elements.

A seed vector can be used to force a tangent to orient in a consistently reproducible direction by projecting the seed vector onto a plane that is perpendicular to the calculated normal

$$\boldsymbol{t}_1 = \frac{(\boldsymbol{I} - \boldsymbol{nn}) \cdot \boldsymbol{s}}{\|(\boldsymbol{I} - \boldsymbol{nn}) \cdot \boldsymbol{s}\|} \tag{45}$$

$\boldsymbol{s}$ is the seed vector, and the resulting tangent is orthogonal to the normal. The second tangent vector is then the cross-product of the normal and the first tangent, $\boldsymbol{t}_2 = \boldsymbol{n} \times \boldsymbol{t}_1$. This method works well for nearly flat surfaces but breaks down when the seed vector is nearly coincident with the normal vector. In particular, in the square jet example problem under large nozzle

rotations ($>90°$), the free surface has all possible orientations, so a single seed vector is guaranteed to break down somewhere on the surface.

Similar to calculation of the normal, local element co-ordinates can also be used to find tangent vectors. If ξ is one of the local element co-ordinates, then an appropriate tangent vector is

$$\boldsymbol{t}_1 = \frac{\dfrac{\partial \boldsymbol{x}}{\partial \xi}}{\left\| \dfrac{\partial \boldsymbol{x}}{\partial \xi} \right\|} \tag{46}$$

This tangent will vary from element to element because the surface orientation changes and because the local co-ordinate ξ can vary from element to element. Thus, arithmetic averaging, as was done with the normal, does not work. Figure 3(a) shows what happens when arithmetic averaging is used to calculate nodal tangent vectors on an unstructured grid. The result is that some portions of the mesh exhibit locally concentrated shearing. The cause of the local shearing is improperly rotated mesh equations; because they were not put in proper normal and tangential form, spurious shear stresses arise that deform the mesh.

To correct for different element orientations adjacent to a node, the tangent calculated by Equation (46) from one element can be used as the seed vector for Equation (45) in an adjacent element. Using this procedure, arithmetic averaging of the tangent vectors produces smooth, robust mesh deformation, as shown in Figure 3(b).

At an edge node where two surfaces intersect, the line tangent (along the curve of intersection), $\boldsymbol{t}$, is well defined by Equation (46) for the line tangent in Figure 2 except for its direction. To determine the direction of the tangent, one of the intersecting surfaces is labeled as the primary surface (surface 1 in Figure 2) whose normal is labeled $\boldsymbol{n}_1$ and the cross-product between $\boldsymbol{n}_1$ and the line tangent is defined as the binormal $\boldsymbol{b}$, which is defined as outward pointing from the primary surface; the direction of the line tangent is chosen so that the cross-product of the normal and the line tangent produces an outward-pointing binormal.

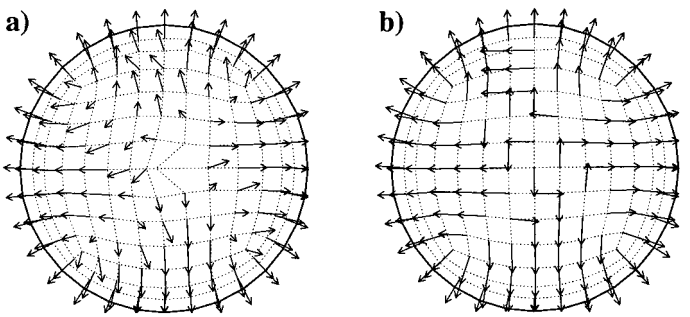

Figure 3. (a) Example of improper mesh distortion caused by incorrect choice of tangent vectors for rotation. The mesh shown is the top of a cylinder and the arrows indicate the calculated first tangent vector at a node. (b) An improved calculation of nodal tangent vectors using the basis-reseed method.

Thus, calculation of the normal, tangent, and binormal vectors at an edge is more straightforward than on a surface.

3.3.4. Rotated conditions and iterative solvers. Because of the typically large matrices associated with three-dimensional free surface problems, application of iterative solution techniques is a necessity. This, however, can lead to difficulties when applying rotated boundary conditions. Iterative solvers typically work best the more diagonally dominant the linear algebra problem. It is possible when applying a rotated condition to reduce if not eliminate diagonal dominance and result in very poor convergence of the iterative solver. For example, if a planar distinguishing condition associated with the y-plane replaces the mesh equation associated with displacements in the x-direction, the result will very likely be failure of the iterative solver to converge. Consequently, considerable care must be exercised in applying rotated conditions so that the equations replaced are associated with the degrees of freedom to which the distinguishing condition is most sensitive. Thus, the correct choice in the above example is to replace the y mesh equation with the planar distinguishing condition instead.

3.4. Methods for solving finite element residual equations

The discretized equations derived in the previous section form a set of non-linear algebraic equations and must be solved for the nodal unknowns. The equation set can be written in vector form, with a dimension equivalent to the total number of nodal degrees of freedom

$$\boldsymbol{F}(\boldsymbol{u}_h, \boldsymbol{p}_h, \boldsymbol{d}_h) = 0 \tag{47}$$

Here $\boldsymbol{u}_h$, $\boldsymbol{p}_h$, and $\boldsymbol{d}_h$ represent the velocity, pressure, and displacement degrees of freedom, in vector form. We linearize this system with the Newton–Raphson method around the initial guess or previous solution, here denoted as the nth iterate

$$\boldsymbol{F}(\boldsymbol{u}_h^n, \boldsymbol{p}_h^n, \boldsymbol{d}_h^n) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{v}}\bigg|^n (\boldsymbol{u}_h^{n+1} - \boldsymbol{u}_h^n) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{p}}\bigg|^n (\boldsymbol{p}_h^{n+1} - \boldsymbol{p}_h^n) + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{d}}\bigg|^n (\boldsymbol{d}_h^{n+1} - \boldsymbol{d}_h^n) = 0 \tag{48}$$

The $(n+1)$ iterate is not yet known. Equation (48) can be rearranged to give

$$\boldsymbol{J}(\boldsymbol{x}^n)(\boldsymbol{x}^{n+1} - \boldsymbol{x}^n) = -\boldsymbol{F} \tag{49}$$

where $\boldsymbol{J}$ is the well-known Jacobian matrix of partial derivatives or sensitivity matrix, and $\boldsymbol{x} \equiv [\boldsymbol{u}_h, \boldsymbol{p}_h, \boldsymbol{d}_h]$ represents a vector of the nodal unknowns. The residual vector $\boldsymbol{F}$ and the Jacobian matrix $\boldsymbol{J}$ are constructed using numerical integration via Gaussian quadrature on an element by element basis. All derivatives with respect to the nodal degrees of freedom are calculated analytically with no approximation, and so we expect quadratic convergence of the iteration process. Details of this procedure for two-dimensional boundary fitted mesh motion problems are given by many previous researchers [4,14,32]. However, few documented attempts at the same fully coupled approach in three dimensions have been made [19].

Solving the matrix system (49) with direct Gaussian elimination based solvers is generally preferred but rarely practical for three-dimensional problems. The rapidity of fast iterative

solvers is the only recourse, but these solvers are encumbered by the poorly conditional matrices generated by a fully coupled approach. The culprits that tax the best of iterative solver/preconditioners pairs are the continuity condition (26) and the free boundary distinguishing conditions, e.g., Equation (42). These conditions typically result in zero diagonal entries in the matrix, or at best a matrix row that has relatively large numbers off the diagonal. The pressure-stabilized Petrov–Galerkin formulation, as detailed above, improves the condition of the matrix system and has to date allowed us to take advantage of the iterative solvers. It should be mentioned that the arrangement of our distinguishing conditions and conflicting boundary conditions at external mesh boundaries, seams, and corners play an important role in iterative solver performance. A matrix system that may be solvable with the direct methods are only efficiently solved with iterative methods when these conditions are arranged so that the diagonal or near-diagonal entries are significantly populated.

To solve the matrix system we chose the Aztec linear solver package [33], which was developed to provide convenient access to an entire library of different iterative methods. Aztec provides many basic iterative solver solution methods, including conjugate gradient (CG) and generalized minimum residual (GMRES). The package also offers a variety of preconditioners with the most effective (but most expensive) being that which uses incomplete LU (ILU) factorization. The best choice of iterative solvers for a given problem remains as much an art as a science. All of the problems solved in this work employed GMRES with ILU preconditioning. GMRES requires a chosen sub-space size, with which a Gram–Schmidt orthogonalization is preformed for each new search vector [34]. In the example to follow, we chose the subspace size to range between 64 and 100, depending on the problem size and nature.

4. RESULTS AND APPLICATIONS

4.1. Solid body rotation of fluid with a free surface

The free surface of a fluid in solid body rotation without surface tension becomes a paraboloid of revolution at steady state. Thus, this is a simple problem for testing the accuracy of three-dimensional free surface predictions. Figures 4 and 5 display predictions for a fluid in a rotating cup. The mesh is unstructured and confirms to the fluid domain as discussed above. The kinematic condition serves as the distinguishing condition on the upper free surface. There is no-slip between the fluid and the walls or bottom of the cup, so $u_\theta = \omega r$ there; the problem is solved in Cartesian co-ordinates, so u_θ is translated into velocities in the x- and y-directions. The contact line where the free surface meets the cup wall is pinned at a fixed position but the angle is not constrained. Using a fixed contact line simplifies the edge conditions and alleviates the need for a constraint on the volume, which is simplified by the specifying the height of the contact line. As the angular velocity of the cup, ω, increases, the free surfaces sags further into the cup

$$\Delta z(r=0) = \frac{\omega^2}{2g} \tag{50}$$

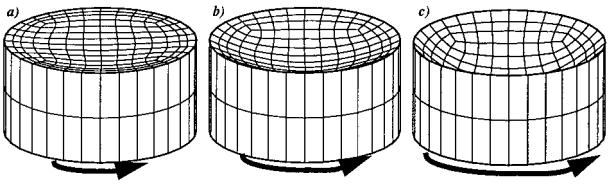

Figure 4. Sample predictions of three-dimensional free surface of a fluid under solid body rotation with angular velocities that cause vertical displacements of (a) $\Delta z = 0.05$, (b) $\Delta z = 0.25$, and (c) $\Delta z = 0.5$.

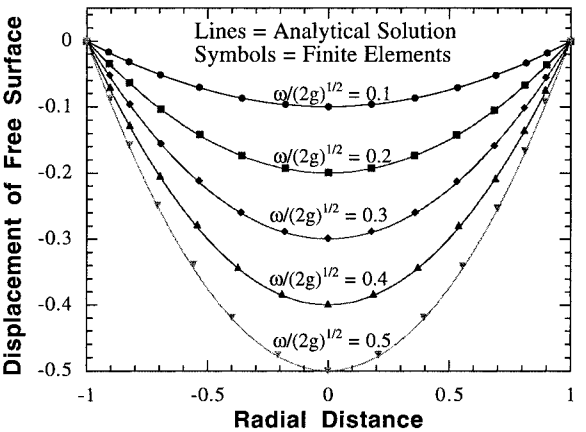

Figure 5. Comparison between free surface shape predicted by finite element calculations and analytical solution for a fluid under solid body rotation at a series of angular velocities.

$\Delta z(r = 0)$ is the depth that the free surface sags at the axis of rotation, and g is the magnitude of gravity; this result is independent of density and viscosity.

Figure 4 shows the mesh at three different angular velocities and clearly shows that increasing the angular velocity causes additional sag in the free surface. Figure 5 shows the comparison between the results predicted by finite elements and Equation (50). For all the cases shown in Figure 5, the finite element predictions nearly match the analytical results. A match between these results and the analytical solution is not trivial because the finite element mesh does not possess cylindrical symmetry; so this example serves as a good test of the three-dimensional fluid mechanics and free surface capability.

4.2. Extrusion from a square nozzle

Many industrial processes involve extrusion of a liquid from an orifice. For example, extrusion is used in the production of polymer films, fibers and coatings. The subsequent evolution of the falling fluid affects the quality of the product. This example problem displays predictions

of fluid exiting from a nozzle of square cross-section and falling under gravity. Figure 6 depicts the typical geometry and boundary conditions for this example problem. This problem contains several features essential to our applications. Fluid flows into the nozzle (through the inflow plane) under an applied pressure and exits the domain through the outflow plane in free-fall. Under appropriate conditions this can lead to a steady state solution. The nozzle is rigid and inclined with respect to gravity so the flow is truly three-dimensional. The nozzle exerts drag on the fluid through no-slip conditions along the walls but fluid exiting the nozzle falls in a free jet under the action of gravity. The jet deforms due to gravitational, viscous, inertial, and capillary forces before exiting the domain. Thus, the cross-section of the domain changes from square to circular between the inlet and outlet.

Plane conditions (14) are applied on the walls of the nozzle and the inflow and outflow planes. There is also an interior planar boundary at the outflow end of the nozzle. The kinematic condition (16) is applied to the entire free surface of the jet.

The fluid enters the nozzle under an applied pressure and has zero velocity (no-slip) along the walls of the nozzle (see Figure 6 for a schematic of the boundary conditions). The free surface is pinned to the lip of the nozzle. The fluid exits the domain with an assumed zero shear and tangential stresses that approximates the physical situation; strictly, the fluid stream

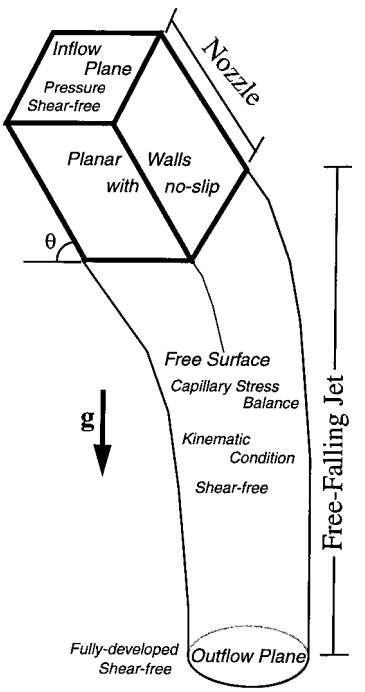

Figure 6. Sketch of the domain and boundary conditions applied in a three-dimensional free surface flow of fluid extruding from a nozzle and falling under gravity.

accelerates under the action of gravity and continues to thin, ultimately breaking up. The conditions chosen for the results in Figure 7 ensure that the shape of the jet is sensitive to the competing effects of gravity, inertia, capillary forces, and viscosity. Using the nozzle half-width and the maximum inlet velocity at the nozzle entrance as the reference velocity, the Reynolds number is about 25 and the Capillary number is 1.7. The Stokes number is 5, using the width of the nozzle as the reference length.

The series of solutions in Figure 7 were predicted using continuation starting from a nozzle oriented vertically. The initial solution of the problem shows that the extruded fluid evolves from a square to a nearly circular cross-section through the combined actions of surface tension and viscosity. To achieve solutions with an inclined nozzle, the nozzle was rotated in 5° increments. As the nozzle rotates, the extruded fluid is thrown further from the nozzle due to the inertia of the existing fluid. When the nozzle rotates past horizontal (90° in Figure 7(d)) the fluid arcs upward slightly before gravity causes it to bend and fall. The maximum achievable rotation is limited by several computational factors: (1) mesh distortion at the nozzle exit, (2) elongation of the elements in the falling jet, and (3) some oscillations in the free surface that occur as the mesh distortion becomes too large. To alleviate these limitations, a finer mesh could be used, or the constitutive equation for the mesh motion could be modified to reduce elongation of elements.

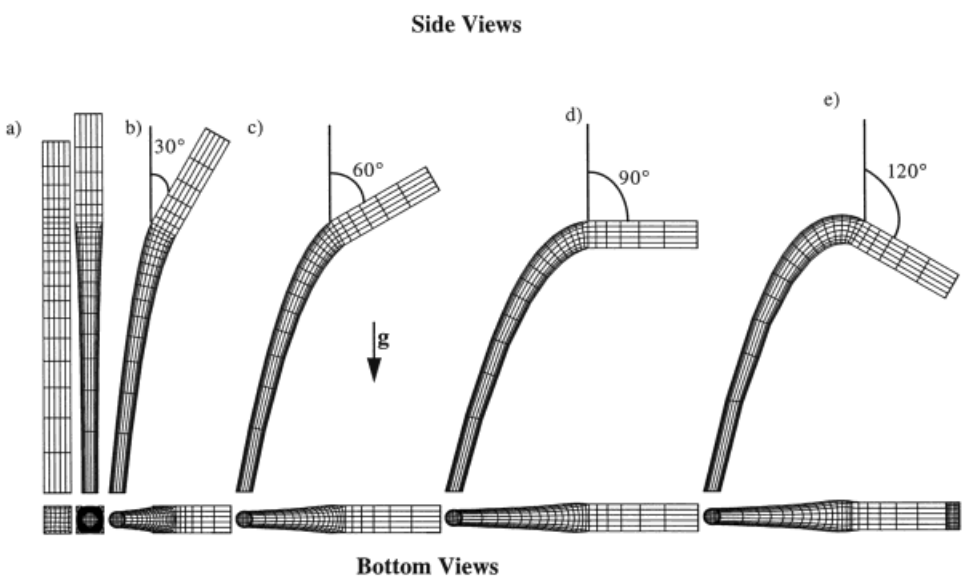

Figure 7. Extrusion of a fluid out of a nozzle of square cross-section. Side and bottom views are shown for a series of rotations of the nozzle with respect to gravity. The first mesh depicts the undeformed mesh. For all cases, the dimensionless parameters are $Re = 25$, $Ca = 1.7$, $St = 5$. (a) No rotation, (b) 30° rotation, (c) 60° rotation, (d) 90° rotation, (e) 120° rotation.

Figure 8 shows predictions of fluid extruded from a rectangular slot inclined 45° from vertical. The aspect ratio of the slot is 4:1, while this aspect ratio is much lower than typical aspect ratios in curtain coating, for example, it does show the potential of this method for application to curtain coating. No-slip between the fluid and slot walls creates a slow moving fluid near the periphery of the nozzle exit; viscous stresses serve to accelerate this fluid (at the expense of momentum from interior) as the sheet falls and bends under gravity. Because the lateral edges of the sheet contain slower moving fluid, the center of the sheet projects further

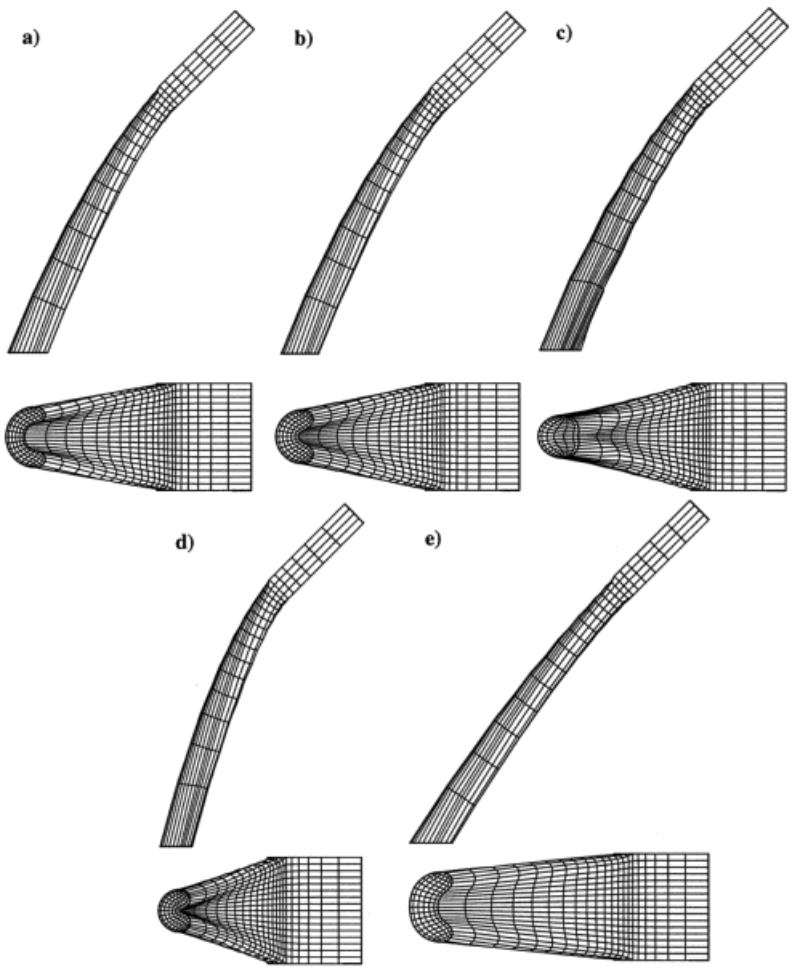

Figure 8. Extrusion of a fluid out of a nozzle with rectangular cross-section rotated through 45° from vertical. Side and bottom views are shown for a series of values of Ca and Re. (a) Low surface tension, $Re = 25$, $Ca = 3.4$; (b) base case, $Re = 25$, $Ca = 1.7$; (c) high surface tension, $Re = 25$, $Ca = 0.85$; (d) low flow rate, $Re = 12.5$, $Ca = 1.7$; (e) high flow rate, $Re = 56$, $Ca = 1.7$.

from the nozzle than the edges, resulting in a curved, falling sheet of liquid. Figure 8(b) shows such a curved sheet under the same conditions (dimensionless parameters) as the square jet in Figure 7.

The shape of the curved falling sheet of liquid is sensitive to the operating conditions. Figure 8(a)–(c) shows the progression in shape as surface tension increases (i.e., the Capillary number decreases at constant Reynolds and Stokes numbers). Under low surface tension (Figure 8(a)), the sheet develops a wide crescent shape, which persists through the outflow plane. As surface tension increases the falling sheet narrows and the crescent becomes thicker. High enough surface tension makes the falling sheet become circular in cross-section. Figure 8(c) shows a sheet that passes from crescent-shaped to nearly circular before passing through the outflow plane.

Varying the fluid flow rate (by varying the inlet pressure) has a similar effect on the shape of the falling liquid sheet. Decreasing the flow rate causes the sheet to spill (fall more steeply) from the slot and the sheet narrows more before it reaches the outflow plane. The result in Figure 8(d) for low flow rate shows that the crescent shape closes up as the edges of the sheet approach and a sharply curved groove forms in the bottom of the sheet. As flow rate increases, the residence time in the computational domain decreases and the fluid exiting the domain has a wider, smoother cross-section with an aspect ratio closer to that of the original slot.

The results in Figures 7 and 8 demonstrate changes in flow topology that can be predicted by the method in this paper. Use of a boundary fitted mesh enables accurate resolution of free surface shapes, which evolve under the combined effects of inertia, gravity, viscosity and surface tension.

5. CONCLUSIONS

This paper presents a new FEM for solving free surface flows of viscous incompressible fluids in three dimensions. To accurately resolve the shape of the free surface, we use a boundary fitted mesh, which deforms with the fluid. Based on the success of two-dimensional free surface methods [4,6] we use a full Newton iteration for simultaneously updating the field variables and the mesh. The use of this fully coupled approach results in a large number of unknowns and a large matrix problem, which is difficult to solve by direct methods. Iterative methods, however, do not converge well for the standard formulation of the continuity equation and some boundary conditions. Pressure stabilization circumvents the LBB requirements and results in better convergence with iterative solvers. At low Reynolds number, the pressure stabilization formulation requires an additional boundary integral to be evaluated around every element if τ_{pspg} varies within the domain or along inflow and outflow boundaries if τ_{pspg} is a constant.

This paper presents a generalization of the Sackinger *et al.* [14] pseudo-solid mesh motion formulation for large deformation and three dimensions. The finite deformation strain tensor is needed to avoid excessive mesh deformation under large rotations. Similarly, when the boundaries rotate significantly, the mesh boundary conditions need to become independent of orientation; this is accomplished by rotating the mesh equations into normal and tangential components along all boundaries. Then the mesh boundary conditions can be applied to

constrain motion in the normal direction while allowing the mesh to redistribute freely in the tangential direction. However, consistent calculation of tangent vector in three dimensions is also important to avoid unwanted mesh distortion. We present three methods of calculating these tangent vectors using a seed vector, the elemental co-ordinates, and a combination of these two methods.

The final formulation is able to track large deformations and rotations of free surfaces in three dimensions. We showed that our formulation accurately predicts the analytical solution for the free surface of a fluid under solid body rotation. The method also predicts how a fluid extruded from a rectangular nozzle bends and falls under gravity. As the nozzle rotates past horizontal, the fluid arcs upward before falling. When the nozzle has a wide aspect ratio, slower moving fluid near the edges causes the falling sheet to become curved, similar to edge effects observed in curtain coating, which require the use of edge guides [35]. The shape of this falling liquid sheet is sensitive to flow rate and surface tension.

This paper provides a powerful framework for predicting free surface flows of fluids in three dimensions using boundary fitted mesh motion. The companion paper to this one [13] extends this formulation to include dynamic contact lines where the extruded fluid impinges on a moving substrate.

APPENDIX A. DERIVATION OF GENERAL CAPILLARY BOUNDARY CONDITION

Appropriate specification of the interfacial momentum balance is crucial to solution of free surface problems. A general form of the interfacial momentum balance has received limited attention in the literature, and some discussions are misleading. In particular, the divergence theorem for three-dimensional curved surfaces is hard to find in the literature, and some authors have assumed it to have the same form as the volume divergence theorem, which is erroneous. Thus, we feel a more detailed discussion is justified.

The weighted residual equation for the Navier–Stokes equation contains an integral of the normal stress over all boundaries of the domain

$$R_i^\alpha = \cdots + \int \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{\text{in}} \, \mathrm{d}A \tag{51}$$

The ellipsis represents all the volume integrals of the Navier–Stokes equations. This surface integral must be evaluated over all boundaries of the domain or be replaced by another boundary condition, e.g., Dirichlet condition. On free surfaces, the normal stress is evaluated by an interfacial momentum balance with capillarity (in dimensional form)

$$-\boldsymbol{n} \cdot (\boldsymbol{T}_{\text{in}} - \boldsymbol{T}_{\text{ext}}) = 2H\sigma\boldsymbol{n} + \nabla_s \sigma \tag{52}$$

Here $\boldsymbol{T}_{\text{in}}$ refers to the stress tensor within the fluid domain and $\boldsymbol{T}_{\text{ext}}$ refers to the state of stress in the external fluid. Normally the stress tensor in the external fluid is considered to be an isotropic pressure (for gases where shear stresses are negligible with respect to the shear stresses

in the fluid) $\boldsymbol{T}_{\text{ext}} = -\boldsymbol{n}p_{\text{ext}}$. As stated in the main text, $H = -\nabla_s \cdot \boldsymbol{n}/2$ is the surface mean curvature, σ is the surface tension, and $\nabla_s = (\boldsymbol{I} - \boldsymbol{nn}) \cdot \nabla$ is the surface gradient operator, with $(\boldsymbol{I} - \boldsymbol{nn})$ as the surface identity tensor and ∇ as the bulk gradient operator.

Combining the weighted residual and the interfacial momentum balance gives

$$-\int_A \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{\text{in}} \, \mathrm{d}A = \int_A \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{n}) p_{\text{ext}} \, \mathrm{d}A + \int_A \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{n}) 2H\sigma \, \mathrm{d}A + \int_A \phi_i \boldsymbol{e}_\beta \cdot \nabla_s \sigma \, \mathrm{d}A \tag{53}$$

The terms on the right-hand side represent the external pressure, the capillary pressure, and the surface tension gradients. Following Ruschak [36] it is desirable to simplify this equation to remove explicit calculation of the surface tension gradient and the curvature. The surface divergence theorem [22] relates surface and volume integrals of vector fields on a curved surface

$$\int_A \nabla_s \cdot \boldsymbol{v} \, \mathrm{d}A = \int_C (\boldsymbol{v} \cdot \boldsymbol{m}) \, \mathrm{d}C - \int_A 2H(\boldsymbol{v} \cdot \boldsymbol{n}) \, \mathrm{d}A \tag{54}$$

$\boldsymbol{v}$ is an arbitrary vector field on the surface, $\boldsymbol{n}$ is the unit vector normal to the surface, and $\boldsymbol{m}$ is an outward pointing unit vector, which is tangent to the surface and normal to the boundary curve along the edge of the surface.

If the vector field is $\boldsymbol{v} = \boldsymbol{e}_\beta \phi_i \sigma$ then the surface divergence theorem gives

$$\int_A \nabla_s \cdot (\boldsymbol{e}_\beta \phi_i \sigma) \, \mathrm{d}A = \int_C \phi_i \sigma (\boldsymbol{e}_\beta \cdot \boldsymbol{m}) \, \mathrm{d}C - \int_C \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{m}) 2H\sigma \, \mathrm{d}A \tag{55}$$

Then invoking the chain rule of differentiation on the left-hand side and rearranging gives

$$\int_A \phi_i \boldsymbol{e}_\beta \cdot \nabla_s \sigma \, \mathrm{d}A + \int_A \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{n}) 2H\sigma \, \mathrm{d}A = \int_C \phi_i \sigma (\boldsymbol{e}_\beta \cdot \boldsymbol{m}) \, \mathrm{d}C - \int_A \sigma \nabla_s \cdot (\boldsymbol{e}_\beta \phi_i) \, \mathrm{d}A \tag{56}$$

This equation replaces the last two terms of Equation (53) and gives a residual equation that does not require the surface tension gradient or the curvature to be calculated explicitly

$$-\int_A \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{\text{in}} \, \mathrm{d}A = \int_A \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{n}) p_{\text{ext}} \, \mathrm{d}A - \int_A \sigma \nabla_s \cdot (\boldsymbol{e}_\beta \phi_i) \, \mathrm{d}A + \int_C \phi_i \sigma (\boldsymbol{e}_\beta \cdot \boldsymbol{m}) \, \mathrm{d}C \tag{57}$$

Alternatively the surface divergence can be eliminated by using the surface identity tensor

$$-\int_A \phi_i \boldsymbol{e}_\beta \boldsymbol{n} : \boldsymbol{T}_{\text{in}} \, \mathrm{d}A = \int_A \phi_i (\boldsymbol{e}_\beta \cdot \boldsymbol{n}) p_{\text{ext}} \, \mathrm{d}A - \int_A \sigma (\boldsymbol{I} - \boldsymbol{nn}) : \nabla (\boldsymbol{e}_\beta \phi_i) \, \mathrm{d}A + \int_C \phi_i \sigma (\boldsymbol{e}_\beta \cdot \boldsymbol{m}) \, \mathrm{d}C \tag{58}$$

Both surface tension gradients and capillary pressure result implicitly from the application of this residual equation. This equation is co-ordinate invariant and only requires calculating $\boldsymbol{e}_\alpha$,

$\boldsymbol{n}$, and bulk gradients in the co-ordinates of interest. When made dimensionless, this equation becomes the same as Equation (39).

This form of the capillary condition is a generalization of Ruschak's original formulation and is easily applied to both two-dimensional and three-dimensional problems. In addition, this formulation applies to curvilinear as well as Cartesian co-ordinates through appropriate specification of the gradient operator using scalar factors. For Cartesian co-ordinates, this equation simplifies to the form shown by Hanumanthu [20]

$$-\int_A \phi_i \boldsymbol{n} : \boldsymbol{T}_{\text{in}}\, \mathrm{d}A = \int_A \phi_i \boldsymbol{n} p_{\text{ext}}\, \mathrm{d}A - \int_A \sigma \nabla_s \phi_i\, \mathrm{d}A + \int_C \phi_i \sigma \boldsymbol{m}\, \mathrm{d}C \tag{59}$$

The computer program, GOMA, with which all the results in this paper were generated, always uses Equation (58) to apply the capillary stress balance regardless of whether the problem is two-dimensional, axisymmetric, or three-dimensional.

REFERENCES

1. Hirt CW, Amsden AA, Cook JL. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics* 1974; **14**: 227–253.
2. Soulaimani A, Fortin M, Dhatt G, Quellet Y. Finite element simulation of two and three dimensional free surface flows. *Computational Methods and Applications in Mechanical Engineering* 1991; **86**: 265–296.
3. Saito H, Scriven LE. Study of a coating flow by the finite element method. *Journal of Computational Physics* 1981; **42**: 53–76.
4. Kistler SF, Scriven LE. Coating flows. In *Computational Analysis of Polymer Processing*, Pearson JRA, Richardson SM (eds). Applied Science Publishers: London, 1983; 243–299.
5. Christodoulou KN, Scriven LE. Discretization of free surface flows and other moving boundary problems. *Journal of Computational Physics* 1992; **99**: 39–55.
6. Christodoulou KN, Kistler SF, Schunk PP. Advances in computational methods for free surface flows. In *Liquid Film Coating*, Kistler SF, Schweizer PM (eds). Chapman and Hall: London, 1997; 297–366.
7. Ushijima S. Three dimensional arbitrary Lagrangian–Eulerian numerical prediction methods for nonlinear free surface oscillation. *International Journal for Numerical Methods in Fluids* 1998; **26**: 605–623.
8. Catillo MEG, Patera AT. Three-dimensional ribbing instability in symmetric forward roll film coating processes. *Journal of Fluid Mechanics* 1997; **335**: 323–359.
9. Zhang YM, Cao ZN, Kovacevic R. Numerical analysis of fully penerated weld pools in gas tungsten arc-welding. *Proceedings of the Institution of Mechanical Engineers Part C*, *Journal of Mechanical Engineering Science* 1996; **210**: 187–195.
10. Ramanan N, Engelman MS. An algorithm for simulation of steady free surface flows. *International Journal for Numerical Methods in Fluids* 1996; **22**: 103–120.
11. Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics* 1981; **39**: 210–225.
12. Floryan JM, Rasmussen H. Numerical methods for viscous flows with free/moving boundaries. *Applied Mechanics Reviews* 1989; **42**: 323–341.
13. Baer TA, Cairncross RA, Schunk PR, Sackinger PA, Rao RR. A finite element method for free-surface flows of incompressible fluids in three dimensions. Part II: dynamic wetting lines. *International Journal for Numerical Methods in Fluids* 2000; **33**: 405–427.
14. Sckinger PA, Schunk PR, Rao RR. A Newton–Raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation. *Journal of Computational Physics* 1996; **125**: 83–103.
15. Lynch DR. Unified approach to simulation on deforming elements with application to phase change problems. *Journal of Computational Physics* 1982; **47**: 387–411.
16. Christodoulou KN, Scriven LE. The fluid mechanics of slide coating. *Journal of Fluid Mechanics* 1989; **208**: 321–354.
17. Thompson JF, Warsi UA, Mastin CW. Boundary fitted coordinate systems for numeric solution of partial difference equations, a review. *Journal of Computational Physics* 1982; **47**: 1–108.

18. Wambersie D, Crochet MJ. Transient finite element method for calculating steady state three dimensional free surface flows. *International Journal for Numerical Methods in Fluids* 1992; **14**: 343–360.
19. Karagiannis A, Hrymak AN, Vlachopoulos J. Three dimensional extrudate swell of creeping Newtonian jets. *American Institute of Chemical Engineers Journal* 1988; **34**: 2088–2094.
20. Hanumanthu R. Patterned roll coating. Ph.D. Thesis, University of Minnesota, UMI, Ann Arbor, MI, 1996.
21. Hughes TRJ, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computational Method and Applications in Mechanical Engineering* 1986; **59**: 85–99.
22. Weatherburn CE. *Differential Geometry of Three Dimensions*. Cambridge University Press: London, 1939.
23. Malvern LE. *Introduction to the Mechanics of a Continuous Medium*. Prentice-Hall: Englewood Cliffs, NJ, 1969.
24. Ogden RW. *Non-Linear Elastic Deformations*. Wiley: New York, 1984.
25. Oden JT, Carey GF. *Finite Elements*, *Mathematical Aspects*, vol. IV. Prentice-Hall: Englewood Cliff, NJ, 1983.
26. Reddy JN, Gartling DK. *The Finite Element Method in Heat Transfer and Fluid Dynamics*. CRC Press: Boca Raton, FL, 1994.
27. Fortin M, Fortin A. Experiments with several elements for viscous incompressible flows. *International Journal for Numerical Methods in Fluids* 1985; **10**: 911.
28. Finlayson BA. *Nonlinear Analysis in Chemical Engineering*. McGraw-Hill: New York, 1980.
29. Droux JJ, Hughes TRJ. A boundary integral modification of the Galerkin least squares formulation for the Stokes problem. *Computational Methods and Applications in Mechanical Engineering* 1994; **113**: 173–182.
30. Tezduyar TE, Shih R, Mittal S, Ray SE. Incompressible flow computations with stabilized bilinear and linear equal order interpolation velocity–pressure elements. University of Minnesota Supercomputer Institute Research Report UMSI90/165, 1990.
31. Brown RA. Finite element methods for the calculation of capillary surfaces. *Journal of Computational Physics* 1979; **33**: 217–235.
32. Schunk PR, Sackinger PA, Rao RR, Chen KS, Cairncross RA, Baer TA, Lebreche DA. GOMA 2.0—A full Newton finite element program for free and moving boundary problems with coupled fluid/solid, momentum, energy, mass, and chemical species transport: user's guide. Sandia National Laboratories Technical Report. SAND97-2404, 1997.
33. Hutchinson SA, Shadid JN, Tuminaro RS. Aztec users guide, version 1.1. Sandia National Laboratories Technical Report, SAND95-1559, October, 1995.
34. Golub GH, Van Loan CF. *Matrix Computations*. Johns Hopkins University Press: Baltimore, MD, 1985.
35. Miyamoto K, Katagiri Y. Curtain coating. In *Liquid Film Coating*, Kistler SF, Schweizer PM (eds). Chapman and Hall: London, 1997.
36. Ruschak KJ. A three-dimensional linear stability analysis for two-dimensional free boundary flows by the finite element method. *Computers and Fluids* 1985; **11**: 391–401.

