JOURNAL OF COMPUTATIONAL PHYSICS **125,** 83–103 (1996)
ARTICLE NO. 0081

A Newton–Raphson Pseudo-Solid Domain Mapping Technique for Free and Moving Boundary Problems: A Finite Element Implementation*

P. A. SACKINGER, P. R. SCHUNK, AND R. R. RAO

Manufacturing & Environmental Fluid Dynamics Department (1511), *Sandia National Laboratories, Albuquerque, New Mexico 87185-0827*

Received February 23, 1995; revised September 8, 1995

An implicit, pseudo-solid domain mapping technique is described that facilitates finite element analysis of free and moving boundary problems. The technique is based on an implicit, full-Newton strategy, free of restrictions on mesh structure; this leads to many advantages over existing domain mapping techniques. The fully coupled approach using Newton's method is particularly effective for problems with strong coupling between the internal bulk physics and the governing physics at unknown free boundary locations. It is also useful when the distinguishing conditions which constrain the free boundary shape provide only an implicit dependence on the boundary location. Unstructured meshes allow for efficient resolution of internal and boundary layers and other regions of strong local variations in the solution and they also reduce the amount of user interaction required to define a problem since the meshes may be generated automatically. The technique is readily applied to steady or transient problems in complex geometries of two and three dimensions. Examples are shown that include free and moving boundary problems from solidification and capillary hydrodynamics.

1. INTRODUCTION

Free and moving boundary problems in fluid dynamics, heat transfer, and other disciplines pose a challenge to computational techniques because the computational domain, or boundary shape, must be determined, together with any field variables internal to the domain. From an industrial perspective, many materials processing issues involve free boundary (FB) or moving boundary (MB) problems; solidification and capillary hydrodynamics, the focus of two of the example applications presented here, are important to the production and quality control of metals, semiconductors, and various coatings. Robust, accurate, and efficient computational methods for locating free and moving boundaries, together with the internal physics, are essential for analysis and design of these processes. This is a particular challenge in the face of the nonlinear nature of many of the conservation equations and of the coupling between the boundary or interface shapes and the internal field variables.

Among the several computational approaches available for FB or MB problems, the best choice depends on the particular set of field equations, boundary conditions, parameter ranges of interest, and the range of domain topologies that need to be simulated. Each computational technique offers its own balance between efficiency, accuracy, and robustness, all of which are desirable objectives for any computational approach to analyzing FB or MB problems [1, 2].

The most *accurate* techniques parameterize the free or moving boundary as a mathematical curve (two dimensions) or surface (three dimensions) in space, i.e., *boundary parameterization* techniques, so that the boundary conditions may be applied precisely at an interface with a well-represented location, orientation, and curvature. Moreover, exact boundary parameterization makes possible the solution of distinctly different field equations, according to the governing physics in each region of the computational domain. Clearly, if accuracy of the boundary shape representation is an overriding concern, this precludes the use of interface-tracking schemes based on representing unknown FB or MB boundary shapes with partially filled cells of finite thickness. Included in the latter approach are techniques employing a "concentration function," such as the volume of fluid (VOF) approach [3], and schemes relying on material marker particles, such as the marker-and-cell (MAC) technique [4]. These approaches trade a less accurate boundary shape representation for a faster speed of solution, especially for certain problems with discontinuous evolution of domain topology.

The purpose of this research is to make a boundary-conforming *domain* mapping technique as robust as possible. Here the term *robust* implies a technique that will most often succeed in converging to the solution, if a solution exists. If there are circumstances where coaxing convergence requires an analyst to modify the approach, then

* This work was performed at Sandia National Laboratories for the U.S. Department of Energy under Contract DE-AC04-94AL85000.

0021-9991/96 $18.00

these modifications should be reasonably intuitive, such as starting with an initial guess that resembles the final solution, or continuing in a parameter from one solution to another. Domain-mapping techniques are distinguished from *boundary*-mapping techniques; in the latter case the primary focus is on parameterizing the shape of an unknown free boundary with one less dimension than the computational domain. Thus, this discussion omits some promising new techniques based on the level-set approach [5] that, too, respect a need for accuracy in representing boundaries with strong coupling to curvature-dependent forces.

Frequently, the solution of various nonlinear partial differential equations in different parts of the computational domain will be of as much interest as the shape of the boundary *per se.* A recurring challenge to computational techniques addressing FB and MB problems is that of keeping an updated interior discretization of the domain that is highly responsive to the boundary shape parameterization. The interior discretization is required to obtain accurate representations of the field variables which, in turn, will affect the boundary shape through the boundary conditions that apply there. The key step is to relate the free boundary position to the internal mesh discretization with algebraic or differential equations. The approach advocated in this work is to use two criteria to select the equations: the first is that these equations place no restrictions on the internal mesh structure, and the second is that the equations fit within a Newton–Raphson solution framework.

One such set of mesh-positioning equations meeting these guidelines are those describing the deformation of an elastic solid continuum under boundary loads. This *pseudo-solid* domain mapping procedure is suitable for a range of FB and MB problems, where the topology of the initial guess domain and the final domain are similar. The essential restriction is that the connectivity of the domain remain the same; e.g., a simply-connected domain is not permitted to evolve into a doubly-connected domain. So-called *distinguishing conditions* (DCs) [6] serve to constrain the position or motion of the boundaries of the general deforming domain. These constraints, applied to the pseudo-solid deformation, differ from boundary conditions that are conventionally applied to solid materials in that they are not expressed specifically in terms of the mesh displacement or tractions, but rather in terms of whatever variables are relevant to the FB or MB problem.

Lynch and coworkers were the first to exploit this pseudo-solid concept. Lynch and O'Neill [7] used the static equilibrium equations for an isotropic, elastic pseudo-solid in two dimensions (the plane stress equations) to advance the mesh configuration at each time step. Variations on this basic approach may be found in the work of other researchers (e.g., [8]). However, similar to Lynch's approach, all subsequent techniques which employ the pseudo-solid mesh movement concept do so in a decoupled manner: solution of the field equations is performed on a mesh that is fixed; then satisfaction of the boundary conditions associated with the free boundary are used to update the estimate of the mesh. If the boundary motion is explicitly prescribed, or if the boundary evolution can be tied to the field variables explicitly through a boundary condition (see [9] and p. 19 of [10],) then updating the shape of the domain through a simple mapping or more complicated mappings is possible [11–13].

Here, this pseudo-solid approach is taken a step further by embedding it in a full Newton–Raphson framework. Full Newton iteration with an implicitly defined mapping has a history in the arena of finite element methods for free and moving boundary problems; this history is briefly examined in Section 2. The pseudo-solid mesh motion concept is then discussed in Section 3. There, in addition to developing the underlying equations, the connection is made between the internal mesh motion and the physics of interest at the free or moving boundary. This connection is made through the application of the distinguishing conditions, which is discussed in a general context in Section 3.3 and in the setting of specific applications in Section 5. Section 4 contains a discussion of how the pseudo-solid mesh motion equations and the equations governing the physics of interest are solved with the Galerkin/finite element method. Newton–Raphson iteration is the key to solving the set of nonlinear algebraic equations resulting from the finite element formulation, and the process of building the Jacobian matrix is aided through the use of exact analytic expressions for each nonzero entry. As a consequence and as the key innovation of this work, the advantages of the Newton iteration scheme are obtained for the pseudo-solid domain mapping technique, relaxing the restrictions on grid structure that have hobbled previous Newton-based techniques for FB and MB problems.

Section 5 demonstrates the Newton, pseudo-solid approach with several example problems taken from coating and polymer processing and melting and solidification. It is shown that the full Newton iteration strategy adopted here is a useful step forward in terms of robustness of computational techniques for free and moving boundary problems.

2. NEWTON'S METHOD FOR FB AND MB PROBLEMS

The power of full Newton iteration has been demonstrated repeatedly for a variety of different free and moving boundary problems, including flow of viscous incompressible fluids with significant surface tension [14], solidification with heat transfer and fluid flow [15], and directional solidification of binary alloys with a curvature-dependent melting temperature [16]. A full Newton approach has been

used for FB and MB problems in the context of finite element methods [2], finite difference methods [17], and boundary element methods [18]. These and other references may be consulted for a discussion of the details required to implement a full Newton iteration scheme for different discretization approaches. Here, only the essential steps for the full Newton finite element approach are introduced so that a comparison may be made between the pseudo-solid technique and two previous techniques: the method of spines [14] and the elliptic mesh generation technique [19].

In general, a mathematical description of a FB or MB problem results in a set of coupled partial differential equations. Discretization of these equations leads to a set of coupled nonlinear algebraic equations. A vector of unknowns, ordered so that the field variables associated with the conservation equations appear first, followed by the discretized mesh mapping variables, can be partitioned as

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}^c_{\mathscr{D}} \\ \mathbf{u}^c_{\Gamma} \\ \mathbf{u}^d_{\mathscr{D}} \\ \mathbf{u}^d_{\Gamma} \end{bmatrix}. \tag{1}$$

Here, the superscripts c and d denote unknowns associated with conservation equations and mesh-point displacement equations, respectively. A further subordering of the unknowns is indicated by the subscripts, as interior values of the conserved field variables appear first, denoted by the $\mathscr{D}$ subscript, followed by values taken on the boundaries and interfaces that are denoted by the Γ subscript. Of course, the actual ordering of the unknowns is unimportant, and the particular ordering may be chosen for computational efficiency and convenience.

Associated with the vector of unknowns is an equal-length vector of independent residual equations which arise from the discretization of the governing differential equations, boundary conditions, and other general constraints in the problem,

$$\mathbf{R}(\mathbf{u}) = \begin{bmatrix} \mathbf{R}^c_{\mathscr{D}} \\ \mathbf{R}^c_{\Gamma} \\ \mathbf{R}^d_{\mathscr{D}} \\ \mathbf{R}^d_{\Gamma} \end{bmatrix} = \mathbf{0}. \tag{2}$$

Residual equations for interior values of the conserved field variables and of the displacement equations will depend on the particular choice of discretization technique. The residual equations for the boundary values of the conservation variables incorporate any Dirichlet or Neumann conditions for the field variables that apply at the boundary. Finally, residual equations for the boundary values of the mesh-displacement mapping include the conditions distinguishing the surface positions in the FB or MB problem. By *distinguishing condition,* a condition is meant which is used to effectively constrain the unknown location of the free or moving boundary. These conditions may depend on the physics of the problem at hand, or they may be used to prescribe the position or motion of the boundary (see Section 5 *et seq.*)

Newton's method for finding roots to (2) relies on computing a Jacobian matrix that describes the sensitivity of all equations with respect to all unknowns. Given an initial guess $\mathbf{u}^{[0]}$, a correction vector may be computed at each Newton iteration, n, as the solution to a linear system and added to the guess:

$$\mathbf{J}(\mathbf{u}^{[n]})\boldsymbol{\delta}^{[n]} = -\mathbf{R}(\mathbf{u}^{[n]}), \tag{3}$$

$$\mathbf{u}^{[n+1]} = \mathbf{u}^{[n]} + \boldsymbol{\delta}^{[n]}. \tag{4}$$

At each iteration, the current estimate of the full unknown vector $\mathbf{u}^{[n]}$ is used to calculate the Jacobian matrix and the residual vector.

It is illustrative to show the basic structure of the Jacobian matrix. It includes 16 submatrices as follows:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial \mathbf{R}^c_{\mathscr{D}}}{\partial \mathbf{u}^c_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^c_{\mathscr{D}}}{\partial \mathbf{u}^c_{\Gamma}} & \dfrac{\partial \mathbf{R}^c_{\mathscr{D}}}{\partial \mathbf{u}^d_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^c_{\mathscr{D}}}{\partial \mathbf{u}^d_{\Gamma}} \\ \dfrac{\partial \mathbf{R}^c_{\Gamma}}{\partial \mathbf{u}^c_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^c_{\Gamma}}{\partial \mathbf{u}^c_{\Gamma}} & \dfrac{\partial \mathbf{R}^c_{\Gamma}}{\partial \mathbf{u}^d_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^c_{\Gamma}}{\partial \mathbf{u}^d_{\Gamma}} \\ \dfrac{\partial \mathbf{R}^d_{\mathscr{D}}}{\partial \mathbf{u}^c_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^d_{\mathscr{D}}}{\partial \mathbf{u}^c_{\Gamma}} & \dfrac{\partial \mathbf{R}^d_{\mathscr{D}}}{\partial \mathbf{u}^d_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^d_{\mathscr{D}}}{\partial \mathbf{u}^d_{\Gamma}} \\ \dfrac{\partial \mathbf{R}^d_{\Gamma}}{\partial \mathbf{u}^c_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^d_{\Gamma}}{\partial \mathbf{u}^c_{\Gamma}} & \dfrac{\partial \mathbf{R}^d_{\Gamma}}{\partial \mathbf{u}^d_{\mathscr{D}}} & \dfrac{\partial \mathbf{R}^d_{\Gamma}}{\partial \mathbf{u}^d_{\Gamma}} \end{bmatrix}. \tag{5}$$

The upper left two-by-two block of this matrix expresses the sensitivity of the field equations to corresponding unknowns, e.g., a momentum equation with respect to a velocity component. The lower right two-by-two submatrix expresses the sensitivity of chosen mesh-position equations with respect to all the position degrees of freedom, e.g., the coordinates of the mesh points. This submatrix system may sometimes be solved separately, as the position degrees of freedom $\mathbf{u}^d_{\mathscr{D}}$ may be explicitly represented by the mesh-position equations, as in the spine parameterization technique described below. In that case, the only implicit variables are those reflecting the free boundary position (i.e., the $\mathbf{u}^d_{\Gamma}$); the distinguishing conditions at the free or moving boundaries are used to account for these variables as a part of the matrix system.

The submatrices on the upper right express the sensitivity of the field equations with respect to mesh degrees of

freedom; their size will depend on the choice of domain mapping equations. Importantly, any spatial gradients in these discretized residual equations need to be differentiated with respect to the mesh displacement variables that affect the location of the mesh via the mesh-position equations. The only other notable submatrix of the Jacobian is the sensitivity of the boundary residual equations (i.e., the distinguishing conditions, $\mathbf{R}_\Gamma^d$) with respect to the field variables, $\mathbf{u}_\Gamma^c$ and $\mathbf{u}_\mathscr{D}^c$. This is the submatrix on the lower left. These derivatives provide a means of coupling the overall mesh mapping with the satisfaction of distinguishing conditions.

Moreover, to assure strong (quadratic) convergence it is necessary to compute the Jacobian matrix with analytical expressions. Analytical evaluation requires an analytically differentiable mapping between the mesh position equations and the spatial discretization. For the finite element method, advantage can be taken of the analytical representation of the parametric mapping from the global to the local element coordinate system (see [20]). For the finite difference method, some analytical expression for the internodal distances as functions of spatial variables must be used [17]. Presented in the remainder of this section are two choices of mesh position equations that allow for analytical Jacobian evaluation.

The object then is to compute the Jacobian matrix (5) and the residual equations (2). For the spine, elliptic, and pseudo-solid mapping techniques discussed below, the only difference is in the existence and form of the $\mathbf{R}_\mathscr{D}^d$ components of the residual vector; all remaining components of the residual vector are unchanged. The Jacobian matrix constructed for the three different techniques will differ, however, since it will contain sensitivities to the different kinds of position degrees of freedom in the solution vector (1). A succession of increasingly flexible forms for $\mathbf{R}_\mathscr{D}^d$ occurs from spine to elliptic to pseudo-solid mapping techniques.

Spine techniques are based on the concept of a height function, with interface heights above a base mathematical surface representing the unknowns [21]. The node point positions internal to the computational domain are expressed as algebraic functions of spine height, relying upon the structure of the grid to quickly identify the proper spine (see Fig. 1). The internal mesh point displacements $\mathbf{u}_\mathscr{D}^d$ are removed from the unknown vector and the corresponding residual equations $\mathbf{R}_\mathscr{D}^d$ are removed from the overall residual vector and used to determine $\mathbf{u}_\mathscr{D}^d$ from the given $\mathbf{u}_\Gamma^d$ as required. For the finite element method, full Newton implementations of spine techniques with increasing sophistication have been used for a variety of free and moving boundary problems [20, 14, 22, 23, 24]. Despite their inherent efficiency, spine techniques suffer from drawbacks: the technique fails when the interface or boundary distortions seek a multi-valued height function; considerable expertise

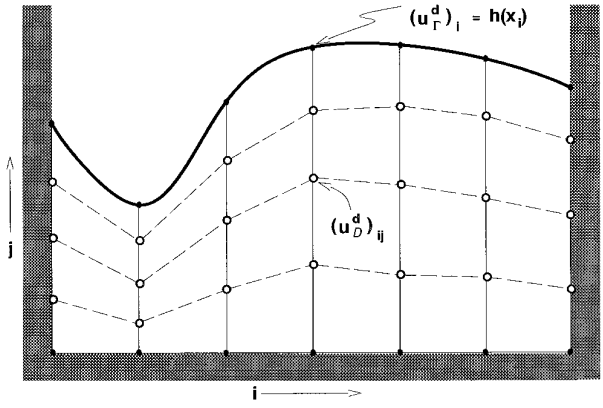

FIG. 1. A basic schematic of spine approaches to representing the unknown shape of an interface. The underlying structure of the grid permits the boundary deformations to be propagated into the interior of the computational domain. Sliding, pivoting, and curved spines are extensions of this approach that can help when the interface deformation tends too close to multi-valuedness. Free boundary shape approximated using height function. Grid structure provides: $(\mathbf{u}_\mathscr{D}^\mathbf{d})_{ij} = \alpha_j(\mathbf{u}_\Gamma^\mathbf{d})_i$.

and manual intervention is needed initially to fit a complicated shape with spines; and, finally, the restriction of the structured mesh can result in greater relative element distortion than if unstructured meshes were permitted.

The limitation to single-valued height functions of the spine techniques may be eliminated by resorting to domain mappings based on the concept of elliptic mesh generation (EMG) [25, 26]. The mesh point locations are then determined as constant coordinate surfaces in the reference coordinate system (see Fig. 2). A full Newton finite element implementation of EMG for free boundary problems was proposed in Ref. [19] and with variations in [27]. The EMG approach is generally more flexible than spine techniques for solving free and moving boundary problems. Drawbacks of the EMG approach are that it requires special handling for irregular domain shapes and that, to date, unstructured meshes have not been accommodated. However, once the conceptual hurdle is made to placing mesh points onto the reference domain without restricting the grid structure to lie upon constant coordinate surfaces, this leads naturally to a formulation that resembles the pseudo-solid approach introduced next.

3. PSEUDO-SOLID APPROACH

A pseudo-solid deforming in a Lagrangian fashion makes for a natural mapping between topologically similar domains. Essentially, for the particular case of a reversible elastic deformation, the pseudo-solid approach is equivalent to using elliptic mesh generation equations to perform the mapping with the condition of minimizing a strain energy functional. However, the pseudo-solid approach does not necessarily require the use of a conservative con-

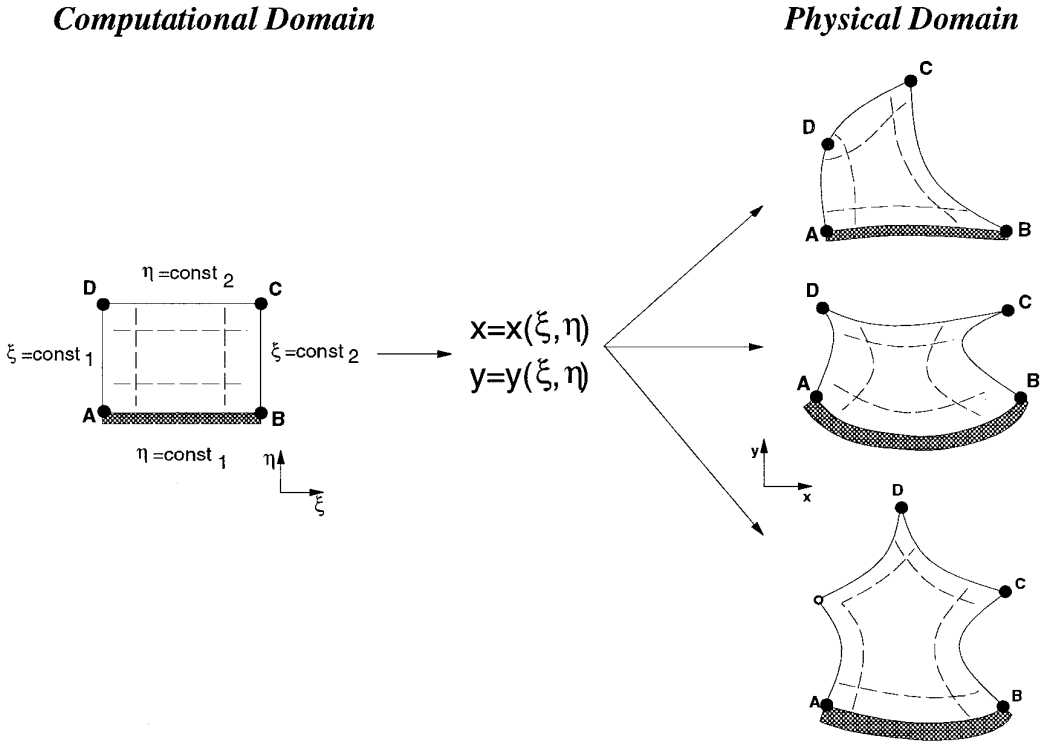

FIG. 2. The boundary-fitted elliptic grid generation approach for representing complex domain shapes can simulate more distortions of the boundaries than spine-based approaches, but level sets of the orthogonal ξ-η coordinate system form boundaries that can create ''special points'' if the topology of the physical domain is described by an odd number of sides.

stitutive model for which a simple variational statement may be found.

One aspect of the implicit approach advocated here is that the deformation field which maps an initial domain into a deformed domain is defined on the deformed domain. In one sense, this is done purely as a convenience; since most of the interesting physics will be computed on the deformed domain, the mechanics of the pseudo-solid might as well be viewed from the same convenient perspective. Thus, the instantaneous configuration of a deformed pseudo-solid is used for interpolating all variables, including the deformation field that was required to bring the initial configuration to the deformed configuration. In essence, the term ''implicit'' refers to the way the free boundary problem is cast. The field variables and the deformation of the computational domain are all unknown *a priori;* they are determined as functions on the deformed domain.

In the development here, the choice is to treat the displacement field as mapping a continuum region, potentially making the technique amenable to other discretization schemes besides the finite element method that is employed here (Section 4). Discrete models of nodes connected by springs with extensional and torsional stiffnesses have been used successfully for mesh deformation, particularly in the context of adaptive methods (e.g., [28–30]). While the discrete approach is convenient for finite difference or finite volume computations, the approach taken here is to use the finite element weighted residual equations directly. The discrete spring and continuum solid approaches result in identical equation systems for certain situations (e.g., low order finite elements), but the convenience of assembling finite element equations for the stress in the pseudo-solid simultaneous to the assembly of the finite element equations for the conservation equations of interest make the finite element weighted approrach more appealing and convenient in the present context.

Notably, *every* boundary of the pseudo-solid is considered to be a moving or free boundary, either under prescribed geometry, kinematics, or under constraints or kinematics governed by the physical problem of interest. Thus, even if the boundary stays at a fixed spatial location, it is useful to formulate the problem so that the pseudo-solid, while conforming to a fixed boundary, may slide along it.

The following two sections describe the actual full Newton pseudo-solid approach for solving free and moving boundary problems. First, the presentation focuses on the fully time dependent ALE formulation, with the emphasis on the mapping. Second, pertinent governing equations for the pseudo-solid and for sample problems with incompressible flow and heat transfer are developed. Third, the

specific assumptions are discussed that permit solution of steady free boundary problems directly, i.e., without marching a transient solution continuously through time to a steady state. Finally, the significance of the distinguishing conditions to the pseudo-solid approach is discussed along with specifics concerning their application.

3.1. *ALE Formulation*

The pseudo-solid mapping approach developed below actually belongs to the category of arbitrary Lagrangian Eulerian (ALE) techniques, insofar as a reference system is chosen to conform to the Lagrangian deformations of the pseudo-solid. The reference frame is not necessarily either Lagrangian or Eulerian with respect to the fluid motion, but arbitrary. The ALE approach has roots in the solid mechanics community where large deformation problems such as rolling and extrusion have created interest in techniques that provide a general reference frame for casting the conservation equations that was not prone to the severe mesh deformations of a strictly Lagrangian approach. Large material deformations are endemic to problems from fluid dynamics as well, and ALE techniques have renewed interest for free and moving boundary problems where deforming material surfaces are involved. From its origins [31], the ALE approach has been developed by a number of researchers, particularly within the finite element community (e.g., [32, 33, 8]).

The general idea of an ALE formulation is that the material and the mesh both move with some velocity relative to a referential coordinate system. This alters the conservation equations so that two velocities generally appear in the governing equations, that of the material and that of the mesh. From the standpoint of the pseudo-solid, the mesh will move in a Lagrangian or material fashion, following some deformation of the pseudo-solid from an initial condition to a final configuration that solves a FB problem at each time step.

Given an initial guess of the domain shape, any spatial coordinates may be expressed as a function of the material coordinates of the pseudo-solid spanning the domain $\mathscr{D}_{\text{initial}}$, viz.,

$$\mathbf{x}_P = \mathbf{X}_{\text{initial}}$$

as shown in Fig. 3, where $\mathbf{x}_P$ denotes the current spatial coordinates of a point P in the pseudo-solid continuum. The initial configuration of the pseudo-solid is $\mathbf{X}_{\text{initial}}$, and at each time plane it is assumed to be given and to be topologically similar to the final deformed configuration. Reference shall be made to a fixed initial configuration of the pseudo-solid during the course of the Newton iteration since a map is sought from a specific initial configuration to a specific final deformed configuration. That is, the mapping depends on both the initial configuration and the final configuration.

As a consequence of the field equations, boundary conditions, distinguishing conditions, and other possible constraining conditions, the solution of the free boundary problem will provide a displacement field that maps the initial domain $\mathscr{D}_{\text{initial}}$ to the final domain $\mathscr{D}_{\text{final}}$, and a material point of the pseudo-solid will be displaced to a new position,

$$\mathbf{x}_P = \mathbf{X}_{\text{final}} = \mathbf{X}_{\text{initial}} + \mathbf{d}. \tag{6}$$

A central objective of the FB or MB problem is, therefore, the determination of the mapping $\mathbf{d}$.

Equation (6) is essentially a Lagrangian description of the pseudo-solid material. By augmenting the free boundary problem of interest with an additional problem of determining the deformation of the pseudo-solid, the continuum may be utilized to embed the grid points associated with the discretized problem. A careful distinction is made between the pseudo-solid material that spans the computational domain and whatever physical material (if any) that spans the same. Indeed, this technique for mesh deformation may be applied to such problems as electrostatics, with potential surfaces bounding a region of vacuum being identified as free surfaces. The pseudo-solid material is only an artifice carried for convenience in moving a computational mesh. And while the deformation of the pseudo-solid is viewed from a Lagrangian standpoint, the real materials often will be best viewed from an Eulerian standpoint. The mapping between the two is performed as needed using (6).

The required displacement field $\mathbf{d}$ need only be regular or invertible, but it is not necessarily restricted to small values. This feature is exploited in the solution of MB and FB problems where the initial guess of the domain shape may require large pseudo-solid strains to achieve the solution after a few Newton iterations. The time-dependent formulation for transient problems with moving boundaries places no *a priori* restrictions on the size of $\mathbf{d}$ at each time plane. However, the accuracy and stability of the transient algorithm may still be affected by the need to construct good approximations for the pseudo-solid *velocity,* $d\mathbf{d}/dt$, which figures into the conservation laws through an ALE term [34]. For FB problems, the full *motion* of the pseudo-solid to its deformed state is not sought, rather, it is only the final displacement that is of interest.

A key feature of the *implicit* approach to the problem is that the displacement $\mathbf{d}$ is regarded as a function of $\mathbf{x}$ on the *deformed* domain, i.e.,

$$\text{Find } \mathbf{d}(x) \text{ for } x \in \mathscr{D}_{\text{final}}. \tag{7}$$

Thus, the mapping back to the original configuration is

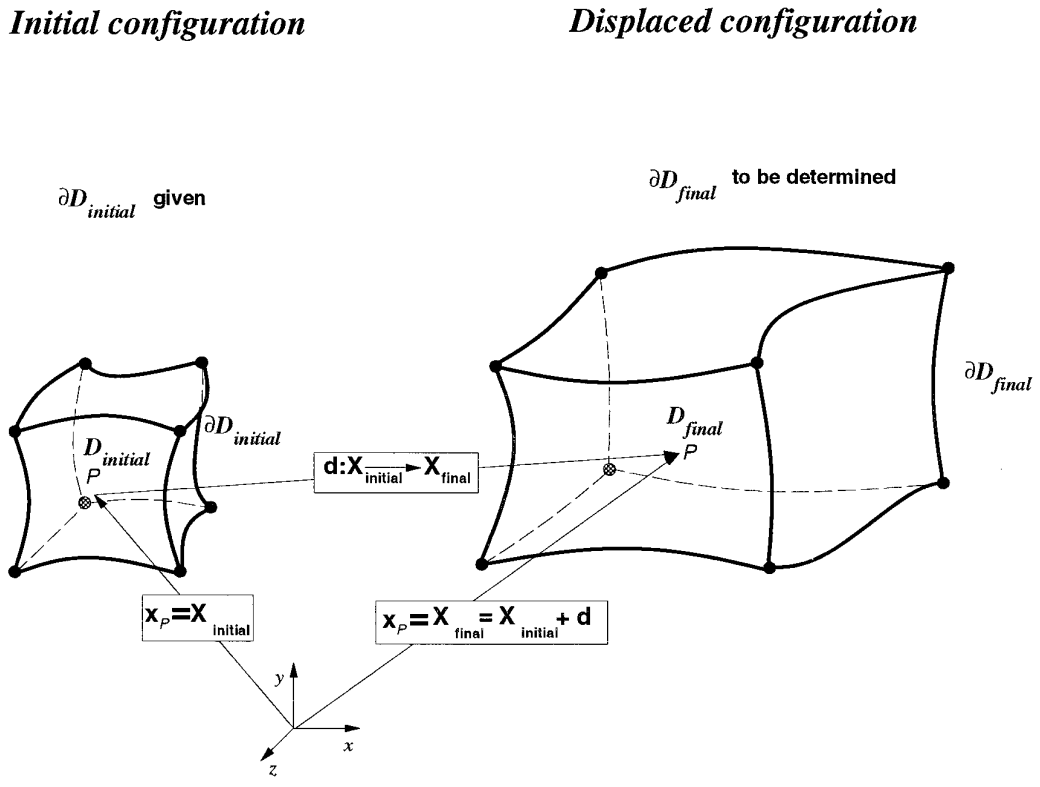

FIG. 3. Pseudo-solid approach. The initial domain undergoes a displacement mapping given by **d**, to become the final domain. The *implicit* nature of the mapping is that **d** is regarded as a function of position on the final domain (or whatever the current best guess is for the final domain during the Newton iteration).

closely tied to the current configuration. An inherent feature in using the implicit mapping of (7) is that $\mathbf{d}(x)$ must be determined iteratively because the displacement mapping **d** required to achieve the solution of MB and FB problems is likely to be nonlinear. Newton's method is an effective means of solving for this kind of implicitly posed nonlinear problem.

For the most part, since the pseudo-solid is merely an artifice to embody the mesh, almost any constitutive model would suffice. Simple models that are not dependent upon the path or history of deformation are preferable, simply for the sake of convenience. The only objective to be met in the choice of constitutive model is to minimize mesh distortion. A small strain elastic model is chosen here and has been used successfully for a range of FB and MB problems. Recent work has indicated more sophisticated nonlinear constitutive models would make better choices to minimize mesh distortion; this will be the subject of future presentations.

For transient problems, an account must be made of the evolution of field variables and the shape of the computational domain. If there are spatial gradients of any field variables, then the motion of the pseudo-solid through the computational domain will cause field variables sampled at fixed points of the pseudo-solid to exhibit temporal variations due to this relative motion. This effect is incorporated into the governing conservation equations using extra terms to account for the pseudo-solid velocity effect on the time-derivative. The solutions to the transient problem will therefore have nonzero pseudo-solid velocity, i.e.,

$$\frac{\partial \mathbf{d}}{\partial t} \neq 0.$$

A key difference between the steady and transient formulations is that the former permits large unrestricted displacements of the pseudo-solid, while the transient formulation requires $\mathbf{d}(\mathbf{x}, t)$ to be a continuous function of time as well as of space. Temporal continuity is required so that the velocity of the pseudo-solid is well defined and can be used as needed to evaluate ALE terms of the conservation equations. No such restriction applies to the Newton iterates on **d** for FB problems, although the robustness and the global convergence properties of the Newton iteration might be affected by the severity of the deformation.

3.2. *Governing Equations*

The differential equations which govern the internal motion of the pseudo-solid material are taken directly from the theory of continuum mechanics. Begin with Cauchy's equation for the balance of forces internally within the pseudo-solid material:

$$\rho_m \frac{\partial \dot{\mathbf{d}}}{\partial t} = \nabla \cdot \mathbf{S} + \rho_m \mathbf{f}_m. \tag{8}$$

Here, ρ_m is the density of the pseudo-solid, $\mathbf{S}$ is the Cauchy stress tensor, and $\mathbf{f}_m$ is the body force per unit volume acting upon the pseudo-solid in the interior. Note that (8) does not contain any reference velocity for the mesh due to the choice of a material frame for the pseudo-solid.

Here the body force $\mathbf{f}_m$ is zero. Certainly, it is possible that some kind of solution adaptivity could be built-in through a specially prepared body force term, which could be used to focus the mesh towards regions of high solution curvature. This presentation defers this intriguing pursuit of adaptivity and focuses on conforming the pseudo-solid at boundaries and interfaces only. Without body forces, the internal motion of the pseudo-solid will be governed entirely by the constitutive equation and by whatever deformations or stresses that are exerted on the boundaries or internal interfaces.

Another arbitrary choice in this implementation is to make the pseudo-solid inertialess, i.e.,

$$\rho_m = 0.$$

Such a choice is not mandatory, of course. For steady, free-boundary problems the choice is irrelevant since all time-dependent terms are neglected, includng the acceleration term in (8) and the static mechanical equilibrium equations are solved instead.

Zero density of the pseudo-solid has other consequences. At each time plane of a MB problem, which is solved like a FB problem, the mesh is required to respond instantaneously. The constraint on mesh motion is degenerate in the sense that no time derivatives of $\mathbf{d}$ appear in the evolution equation for the mesh itself, and an elliptic equation governs the mesh displacement at each time step. The added expense of an implicit solution of these equations averts any potential added expense associated with acoustic transients, where the hyperbolic equations for $\rho_m \neq 0$ are solved instead. Nevertheless, despite its omission from the mesh stress equations, the time derivative of the displacement does appear in other equations and care must be taken to accurately determine $\dot{\mathbf{d}}$ as the mesh deforms through the computational domain for MB problems.

Applying the assumptions of zero body force and zero density yields a static-equilibrium equation for the pseudo-solid response:

$$0 = \nabla \cdot \mathbf{S}. \tag{9}$$

A simple isotropic linear elastic constitutive equation (Hooke's Law) for the pseudo-solid is

$$\mathbf{S} = \lambda \, \mathrm{tr}(\mathbf{E}) \, \mathbf{I} + 2 \, \mu \mathbf{E}, \tag{10}$$

where λ and μ are Lamé coefficients, $\mathbf{I}$ is the identity tensor, $\mathbf{E}$ is the Eulerian infinitesimal strain tensor,

$$\mathbf{E} = \tfrac{1}{2} [\nabla \mathbf{d} + \nabla \mathbf{d}\dagger], \tag{11}$$

and $\mathrm{tr}(\mathbf{E})$ is the trace of the strain tensor. The gradients of the displacement $\mathbf{d}$ are in terms of spatial coordinates. For example, in Cartesian coordinates,

$$(\nabla \mathbf{d})_{ij} = \frac{\partial d_j}{\partial x_i}.$$

The Lamé coefficients express relative shear and extensional moduli of the pseudo-solid. These values may be chosen to help retain an optimal mesh during deformation. Arbitrarily, the choice of

$$\mu = \lambda = 1$$

is made, corresponding to a Poisson ratio $\nu = 0.25$, thus avoiding the degenerate case of an incompressible computational domain ($\nu = 0.5$).

Continuum equations for a sample range of applications are put forth to augment the equations developed above for the deformation of the pseudo-solid. Our interest has been focused mainly on problems in transport phenomena, specifically the flow of viscous incompressible fluids with free surfaces, potentially coupled with heat and mass transfer. As mentioned above, there are potentially a great many other free and moving boundary problems that would benefit from the full-Newton pseudo-solid approach, certainly many more applications than the limited selection shown here.

For an *incompressible* fluid, overall conservation of mass is written simply as

$$\nabla \cdot \mathbf{v} = 0, \tag{12}$$

where $\mathbf{v}$ is the fluid velocity with respect to fixed spatial coordinates. Momentum conservation for the fluid is written

$$\rho \, (\overset{*}{\mathbf{v}} + \mathbf{v} \cdot \nabla \mathbf{v}) = \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}, \tag{13}$$

where ρ is the fluid density, and $\overset{*}{\mathbf{v}}$ is the time derivative of the velocity field including the ALE velocity of the mesh, $\boldsymbol{\sigma}$ is the total Cauchy stress *of the fluid,* and $\mathbf{g}$ is the body force (per unit mass) acting on the fluid. The advected derivative is defined for any conserved quantity ψ as

$$\overset{*}{\psi} = \frac{\partial \psi}{\partial t} - \dot{\mathbf{d}} \cdot \nabla \psi, \tag{14}$$

where the velocity of the mesh is $\dot{\mathbf{d}}$. This derivative is akin to the material derivative that is used in differential conservation statements; it incorporates into the time rate of change that contribution resulting from the reference pseudo-solid moving through a gradient of a field. Generally, the time derivatives like $\overset{*}{\mathbf{v}}$ will be zero for free boundary problems and nonzero for moving boundary problems.

For incompressible flows it is convenient to decompose the fluid stress tensor into an isotropic pressure and a deviatoric stress,

$$\boldsymbol{\sigma} = -P\mathbf{I} + \boldsymbol{\tau},$$

where the deviatoric stress might typically be related to the velocity field through Newton's Law of Viscosity.

Energy conservation for the incompressible fluid is

$$\rho C_p \left(\overset{*}{T} + \mathbf{v} \cdot \nabla T\right) = -\nabla \cdot \mathbf{q} + h, \tag{15}$$

where T is the temperature, C_p is the heat capacity, $\mathbf{q}$ is the diffusive heat flux, and h is a volumetric heat source. Again, advective terms are included to account for both the motion of the fluid relative to the coordinate system ($\mathbf{v}$) and for the motion of the mesh relative to the coordinate system, $\dot{\mathbf{d}}$. In the applications below, Fourier's law is used to represent diffusive heat flux, i.e.,

$$\mathbf{q} = -k\nabla T. \tag{16}$$

Note the deliberate segregation of the advective terms in Eqs. (13) and (15). The ALE mesh velocity is lumped together with the time derivative. This separation is done to ease the solution of steady FB problems where all time derivative terms including the velocity of the pseudo-solid is zero, but the velocity of the fluid may be nonzero.

Other conservation equations may also be of interest, such as those for species conservation or for deformation of real solid materials. There is no inherent difficulty in extending the pseudo-solid domain mapping algorithm to these and other classes of problems. Moreover, with the exception of the "extra" boundary conditions, which are used to distinguish the free surface motion, the usual boundary conditions require no extraordinary treatment.

3.3. *Distinguishing Conditions*

The purpose of this section is to describe the details of the boundary conditions applied to the pseudo-solid, including those which connect its motion to the FB or MB problem of interest, i.e., the distinguishing conditions (DCs). Unlike the essential conditions on displacement or natural conditions on traction applied at the surface of the pseudo-solid, DCs are used to define the position of the free boundary implicitly.

The kinds of DCs available are manifold and depend on the particular kind of FB or MB problem to be solved. Section 5 is merely a sampling of the different kinds of problems and DCs that can be encountered. Indeed, there is often a choice of whether to use a particular boundary or interface constraint as a boundary condition or as a distinguishing condition. Within the context of a Newton–Raphson iteration scheme, however, the choice of DC is not nearly so critical as it is when a lower-order iteration scheme is employed.

There is little difficulty in enforcing such DCs for one-dimensional FB problems; the one "extra" boundary condition arising from the FB problem is used as the one constraint for the one component of displacement that is permissible.

In two or three dimensions, however, mesh displacement is a *vector* quantity that offers two or more displacement directions as degrees of freedom at boundaries or internal interfaces. Correspondingly, the mesh stress equation requires just as many independent boundary conditions. The FB problem will typically provide but one "extra" condition, leaving open the question of what should be done to provide the full complement of boundary conditions upon the displacement vector field for multi-dimensional problems. There is probably more than one answer to this question; one possible answer is described next that provides compatibility with the Newton–Raphson requirement that *all* equations and boundary constraints of the system possess well-defined derivatives (preferably, continuous derivatives).

Compatibility with the Newton–Raphson method is not a trivial issue for free and moving boundary problems. Seemingly simple solutions to choosing the distinguished component of displacement based on

$$\text{Choose } d_\alpha \text{ such that } \mathbf{n} \cdot \mathbf{e}_\alpha = \max_\beta \left(\mathbf{n} \cdot \mathbf{e}_\beta\right), \tag{17}$$

where $\mathbf{n}$ is the unit normal to the surface and the $\mathbf{e}_\beta$ are unit vectors in the coordinate system, do not suffice. The reason is that (17) is not generally differentiable, since a deformable boundary or internal interface may distort to where a different coordinate direction abruptly becomes the preferred distinguished component.

A solution to this dilemma is to rotate the components of

the governing equations locally into normal and tangential components so the normal component may always be identified along a smooth boundary and used for the imposition of distinguishing conditions. Then, a Neumann condition may be applied to the tangential component(s) of the traction so that the pseudo-solid (and, hence, the mesh) may slide freely in those directions. For boundaries that normally would be considered fixed, or those undergoing prescribed kinematics, it is advantageous to consider those, too, as free boundaries, but under a different kind of distinguishing condition to constrain their freedom. Permitting tangential sliding at all external boundaries is an advantage, since it provides for greater overall flexibility with less internal distortion in conforming to different shapes. To be compatible with the Newton–Raphson iteration procedure, however, the rotation of the equations into a local coordinate system must be performed using a differentiable transformation.

Of course there are special points, like corners, which are the junctions or intersections of two or more boundaries and at which it is mathematically impossible to impose a tangential-traction-free constraint (assuming the curves do not join smoothly at the intersections). In the case of corners, multiple distinct constraints apply at the intersection, and the point is located simply using the corresponding set of distinguishing conditions.

Once the rotation of the boundary traction vector equation is accomplished, the normal component is replaced by the distinguishing condition (i.e., an essential boundary condition) for the free boundary, acting much like a potential well which drives the boundary normally to satisfy the DC. The tangential component of the traction vector equation is used to disallow tangential tractions at external boundaries, or to preserve any tangential stress balance at internal interfaces. The constraint at internal interfaces is needed to preserve the mesh conformity that would be lost if internal slippage were permitted.

For specificity and simplicity, consider the two-dimensional system in Fig. 4. A two-dimensional region will have boundaries and internal interfaces that are comprised of piecewise smooth curves that are essentially one-dimensional objects, in that they may be parameterized as a vector-valued function of one real variable (e.g., the arc length along the curve). Along each of these smooth curves is the need to describe unit normal and tangent vectors, with a consistent convention for direction (such as outward) that must be decided upon in advance, so that a continuous representation of $\mathbf{n}(s)$ and $\mathbf{t}(s)$ may be constructed.

Keeping with a two-dimensional prototype, the mesh stress vector at the boundary is decomposed into its spatial coordinate directions and then reconstructed according to the local boundary orientation. If the components of the boundary traction vector acting on the surface of the

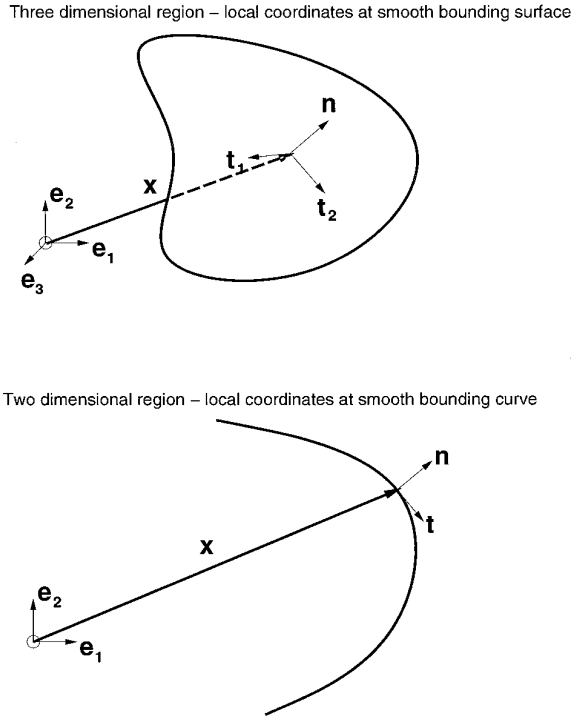

FIG. 4. Smooth portions of the boundary of the pseudo-solid filling the computational domain have local coordinate systems affixed that follow the unit normals and tangents at the boundary.

pseudo-solid in the reference coordinate system are given as

$$\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{18}$$

where the subscripts denote each of the independent coordinate directions and $(\mathbf{e}_1, \mathbf{e}_2)$ denote the unit vectors in those directions. Tractions expressed by (18) are rotated into a local coordinate system based on a normal/tangential basis along the boundary, viz.,

$$\mathbf{f}' = \begin{bmatrix} f_n \\ f_t \end{bmatrix} = \begin{bmatrix} (\mathbf{n}\cdot\mathbf{e}_1) & (\mathbf{n}\cdot\mathbf{e}_2) \\ (\mathbf{t}\cdot\mathbf{e}_1) & (\mathbf{t}\cdot\mathbf{e}_2) \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix},$$

where the unit vectors on the boundary, both the normal $\mathbf{n}$ and the tangent $\mathbf{t}$, are differentiable functions of the boundary position of the pseudo-solid. If the current estimate of a smooth boundary curve is given as

$$\mathbf{x}(s) = \begin{bmatrix} x_1(s) \\ x_2(s) \end{bmatrix}, \quad 0 < s < 1,$$

then expressions for normal and tangential unit vectors are

$$\mathbf{t}(s) = \frac{\dot{x}_1\mathbf{e}_1 + \dot{x}_2\mathbf{e}_2}{\sqrt{\dot{x}_1^2 + \dot{x}_2^2}}, \tag{19}$$

$$\mathbf{n}(s) = -\frac{\dot{x}_1\mathbf{e}_1 + \dot{x}_2\mathbf{e}_2}{\sqrt{\dot{x}_1^2 + \dot{x}_2^2}}, \tag{20}$$

where $\dot{x}_1 = dx/dx$, etc. The sign of the unit vectors is arbitrary, except insofar as the overall consistency of the convention, particularly for the case of deformable internal interfaces in a FB or MB problem, or where any two smooth curves splice together smoothly. The latter form of consistency is pertinent for the finite element and other methods that rely upon small subdomains (elements or cells) that adjoin one another at a boundary.

For constraint equations distributed on a boundary, the equation associated with the normal component of the traction vector f_n is *replaced* with a desired distinguishing condition,

$$r_{\mathrm{DC},k}(\mathbf{u}) = 0, \tag{21}$$

where $\mathbf{u}$ is the vector of unknowns for the FB or MB problem and the k subscript is used to identify the surface and the DC that applies. A distinguishing condition should be differentiable, but it is otherwise unburdened by any restrictions. In particular, Eq. (21) need not have any explicit dependence on the pseudo-solid position $\mathbf{d}$, or for MB problems, on $\dot{\mathbf{d}}$. The examples in Section 5 make this important feature clear. Effectively, the condition (21) acts in the capacity of a normal traction boundary condition which vanishes only insofar as the DC is obeyed.

The process of replacing the normal component of the pseudo-solid traction vector with a distinguishing condition occurs both on exterior boundaries of the computational domain and on interior interfaces that are deformable according to the FB or MB problem. All boundaries that are free or deformable will have distinguishing conditions that apply, even if the conditions are no more than simple geometric descriptions of the boundary (the most basic distinguishing condition).

The constraint equation associated with the tangential component of the traction vector of the pseudo-solid on exterior boundaries vanishes, i.e.,

$$f_t = 0,$$

while at interior interfaces the tangential traction is continuous,

$$f_t|_I - f_t|_{II} = 0.$$

The pseudo-solid is assumed to be continuous and not to admit any internal discontinuities, slip, or fractures. Overall, this pseudo-solid provides a framework for a discretization that is conforming everywhere.

Three-dimensional extensions of these ideas are not difficult in principle, although practically the implementation may be more tedious, since two independent tangent vectors are required to describe a bounding surface and a means of generating such tangent vectors that are globally consistent and smoothly varying along the surface are required for the Newton implementation. Also, three-dimensional regions may have bounding "seams," space curves at which two surfaces intersect. In a general scheme, distinguishing conditions must be applied to two out of three components of the mesh stress vector to position the points onto the seam, but in such a manner that the tangential stress of the mesh vanishes and permits nodes on the seam to slide tangentially along the curve. Special points that are the intersection of three independent distinguishing conditions may be applied to all three components of the mesh force balance, without the intermediate step of local rotation. Just as for the case of two-dimensional FB and MB problems, in three dimensions the consideration of higher order singularities, such as tangential intersections of boundaries and free interfaces, is ignored in this presentation.

4. FINITE ELEMENT FORMULATION

Discussion of the discretization procedure has been postponed to this late stage to emphasize that the technique need not be restricted to finite element formulations. While the FE method offers many advantages and is chosen to solve the continuum equations in this presentation, other discretization techniques might be used instead. Indeed, the only apparent restriction is that of differentiability, which is required for an analytical implementation of Newton's method, and many other spatial discretization techniques (finite difference, spectral) might serve as well.

The governing conservation equations are discretized using the Galerkin finite element method (GFEM), including (9), (12), (13), and (15). The procedure is essentially a restatement of the problem in weak form using the method of weighted residuals [35, 36]. To avoid excessive discretization error all second-order derivatives are integrated by parts using the divergence theorem.

The pseudo-solid stress is governed by a vector equation with each component denoted by the subscript α:

$$0 = R^d_{i,\alpha} \tag{22}$$

$$= (w_i^{(d)}, \mathbf{e}_\alpha \cdot (\nabla \cdot \mathbf{S})) \tag{23}$$

$$= -\int_{\mathscr{D}} \nabla((w_i^{(d)}\mathbf{e}_\alpha) : \mathbf{S}\, dV + \int_\Gamma w_i^{(d)}\mathbf{n}\mathbf{e}_\alpha : \mathbf{S}\, dS) \tag{24}$$

for the i, α component of the mesh stress residual. The

unit vectors of the coordinate system are denoted using $\mathbf{e}_\alpha$. The index i refers to each of the independent basis functions w, and the $^{(d)}$ superscript refers to basis functions chosen to represent the displacement field. Orthogonality is enforced using an inner product based on the integral over the computational domain,

$$(\cdots) = \int_{\mathscr{D}} \cdots dV.$$

The weak forms are often reduced using the divergence theorem (e.g., (24)) to integrands with lower order derivatives and augmented with boundary flux data included of the form

$$(\cdots)_\Gamma = \int_\Gamma \cdots dS.$$

A weighted residual form of the continuity equation for an incompressible fluid is

$$\begin{aligned} 0 &= R_i^c, \\ &= (w_i^{(P)}, \nabla \cdot \mathbf{v}). \end{aligned} \tag{25}$$

Conservation of momentum for the fluid, another vector equation, has a discretized form

$$\begin{aligned} 0 &= R_{i,\alpha}^m \\ &= (w_i^{(m)}, \mathbf{e}_\alpha \cdot [-\rho(\overset{*}{\mathbf{v}} + \mathbf{v} \cdot \nabla \mathbf{v}) + \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{g}]), \\ &= (w_i^{(m)}, \mathbf{e}_\alpha \cdot [-\rho(\overset{*}{\mathbf{v}} + \mathbf{v} \cdot \nabla \mathbf{v}) + \rho \mathbf{g}]) \\ &\quad - (\nabla(w_i^{(m)} \mathbf{e}_\alpha) : \boldsymbol{\sigma}) + (\mathbf{n} \mathbf{e}_\alpha : \boldsymbol{\sigma})_\Gamma, \end{aligned} \tag{26}$$

and the weighted residual equation for the conservation of energy for the fluid is

$$\begin{aligned} 0 &= R_i^e \\ &= (w_i^{(e)}, [-\rho C_p(\overset{*}{T} + \mathbf{v} \cdot \nabla T) + \nabla \cdot \mathbf{q} + h]), \\ &= (w_i^{(e)}, [-\rho C_p(\overset{*}{T} + \mathbf{v} \cdot \nabla T) + h]) \\ &\quad - (\nabla w_i^{(e)} \cdot (-k \nabla T)) + (\mathbf{n} \cdot ((-k \nabla T)))_\Gamma. \end{aligned} \tag{27}$$

The temperature, fluid velocity, and displacement of the pseudo-solid are represented using isoparametric basis functions. The pressure is represented with a lower order interpolant so as to satisfy the LBB constraint (see, e.g., [37]).

The representation of the different field variables in terms of finite element basis functions is based on standard well-known procedures; the most notable feature, insofar as this development is concerned, is the mapping of spatial coordinates using the finite basis functions W defined on the unit element from the unit element to each of the elements in the mesh (the uppercase distinguishes basis functions on the unit element from those on elements in the actual domain),

$$x_\alpha = \sum_k [X_{\alpha k}^{\text{initial}} + d_{\alpha k}] \, W_k^{(d)}(\boldsymbol{\xi}), \tag{28}$$

where the displacement for each node is added to the initial position of the node embedded in the pseudo-solid material. The index k cycles over the nontrivial contributions from basis functions for the element in question. Here $\boldsymbol{\xi}$ is the vector of local isoparametric coordinates.

The only remaining issue concerns the Newton implementation, which requires that the derivatives of the weighted residual integrals expressed by Eqs. (25)–(27) be taken with respect to the field variables. Most of the derivatives that comprise the Jacobian matrix entries may be determined in a straightforward manner. However, since the mesh is deformable, there is the additional complexity of having to differentiate the residual equations with respect to the displacement field, i.e., all of the $d_{\alpha i}$. The details of this step have been developed previously (cf., [14]) and are reviewed here.

Contributions to the Jacobian matrix resembling those for a fixed grid problem are developed in a standard fashion. For example, the thermal diffusion term contributes

$$\begin{aligned} \frac{\partial R_i^e}{\partial T_j} &= \cdots + \frac{\partial (\nabla w_i^{(e)} \cdot (-k \nabla T))}{\partial T_j} + \cdots \\ &= \cdots + (\nabla w_i^{(e)} \cdot (-k \nabla w_j^{(e)})) + \cdots. \end{aligned}$$

Thus, entries comprising the $\partial \mathbf{R}_{\mathscr{D}}^c / \partial \mathbf{u}_{\mathscr{D} \cup \Gamma}^c$ submatrices of the global Jacobian are readily determined.

The sensitivity of the discretized conservation equations with respect to mesh displacement unknowns is more involved. Since the elements are deformable, the integration limits include displacement variables, and this contribution to the global Jacobian matrix must be realized. First, the element integrals are transformed to a fixed unit element, viz.,

$$\int_{\text{element } e} \cdots dV = \int_{\text{unit element}} \cdots |\mathbf{J}_e| \, dv,$$

where $|\mathbf{J}_e|$ is the determinant of the *elemental* Jacobian matrix for the transformation from the unit element to any particular element (using (28)). The elemental Jacobian matrix is constructed as

$$J_{e,\alpha\beta} = \frac{\partial x_\beta}{\partial \xi_\alpha}. \tag{29}$$

A first vital contribution to the global Jacobian's $\partial \mathbf{R}^c_{\mathscr{D}}/\partial \mathbf{u}^d_{\mathscr{D}\cup\Gamma}$ submatrix terms will be made by portions resulting from the transformation to the unit element. That is, terms from

$$\frac{\partial |\mathbf{J}_e|}{\partial d_{\alpha j}}.$$

Second, every portion of the integrands in the weak form residual equations that are explicit functions of position must be differentiated with respect to mesh displacement to provide contributions to $\partial \mathbf{R}^c_{\mathscr{D}}/\partial \mathbf{u}^d_{\mathscr{D}\cup\Gamma}$. For example, a heat source term that is a function of position, $h(\mathbf{x})$, would contribute

$$\frac{\partial (w_i^{(e)}, h(\mathbf{x}))}{\partial d_{\alpha j}} = \left(w_i^{(e)}, \frac{\partial h(\mathbf{x})}{\partial x_\alpha} w_j^{(d)} \right) + \left(w_i^{(e)}, h(\mathbf{x}) \frac{\partial |\mathbf{J}_e|}{\partial d_{\alpha j}} \right),$$

to the global Jacobian matrix. Any terms utilizing the normal and tangent vectors (cf. (19) and (20)) at a boundary will contribute in a similar way, since they depend upon the current estimate of the boundary shape.

A third kind of contribution to $\partial \mathbf{R}^c_{\mathscr{D}}/\partial \mathbf{u}^d_{\mathscr{D}\cup\Gamma}$ is based upon the implicit dependence of the spatial gradient operators upon the deformable mesh. Terms like

$$\frac{\partial \nabla w_i^{(e)}}{\partial d_{\alpha j}} \tag{30}$$

provide nontrivial contributions that may be determined using (28) and the chain rule. The elemental Jacobian transforms gradient terms between elements in the computational domain and the unit element,

$$\frac{\partial ( )}{\partial \xi_i} = \sum_k \frac{\partial x_k}{\partial \xi_i} \frac{\partial ( )}{\partial x_k},$$

$$= \sum_k J_{e,ik} \frac{\partial ( )}{\partial x_k}.$$

Written more compactly, this transformation may be inverted while the elemental Jacobian is nonsingular

$$\nabla_\xi = \mathbf{J}_e \cdot \nabla$$

$$\nabla = \mathbf{J}_e^{-1} \cdot \nabla_\xi.$$

Thus, terms such as (30) are determined by evaluating

$$\frac{\partial \mathbf{J}_e^{-1}}{\partial d_{\alpha j}}.$$

Using the small strain formulation for the pseudo-solid mechanics, the gradient of the mesh displacement will include sensitivity to the mesh displacement in two ways:

$$\frac{\partial \nabla d_\gamma}{\partial d_{\alpha j}} = \frac{\partial \mathbf{J}_e^{-1} \cdot \nabla_\xi d_\gamma}{\partial d_{\alpha j}},$$

$$= \frac{\partial \mathbf{J}_e^{-1}}{\partial d_{\alpha,j}} \cdot \nabla_\xi d_\gamma + \delta_{\gamma\alpha} \mathbf{J}_e^{-1} \cdot \nabla_\xi d_\gamma.$$

These sensitivities are used in the calculation of the $\partial \mathbf{R}^d_{\mathscr{D}}/\partial \mathbf{u}^d_{\mathscr{D}\cup\Gamma}$ terms of the global Jacobian matrix.

The only interaction of the pseudo-solid deformation with the field variables of the conservation equations occurs at the boundaries. That is, the $\partial \mathbf{R}^d_{\mathscr{D}}/\partial \mathbf{u}^c_{\mathscr{D}\cup\Gamma}$ terms of (5) are zero. Distinguishing conditions provide nonzero contributions for the $\partial \mathbf{R}^d_{\Gamma}/\partial \mathbf{u}^c_{\mathscr{D}\cup\Gamma}$ and the $\partial \mathbf{R}^d_{\Gamma}/\partial \mathbf{u}^d_{\mathscr{D}\cup\Gamma}$ submatrices. The specific contributions depend on the form of the particular DC; different kinds of DCs are illustrated in the next section.

5. EXAMPLE PROBLEMS

The examples in this section were chosen to demonstrate several features and advantages of the Newton pseudo-solid approach. These examples make plain the utility of unstructured grids, particularly for resolving boundary layers and singular field variable behavior at contact points. All calculations were performed on relatively coarse meshes and are meant only to illustrate the technique. One example drawn from a solidification processing application is used to demonstrate the strong, quadratic convergence inherent to Newton's method. This feature not only makes steady state analysis more practical for these problems, but it helps make the transient algorithm more efficient and opens the possibility of conducting computer-aided nonlinear analysis [38, 2]. Also demonstrated is a transient coating startup problem showing that large free surface deformations may be determined with the technique described herein, without severe mesh distortion or mesh-line crossing.

In all cases the original mesh topology was generated with the paving algorithm [39]. The meshes are all four- or nine-node isoparametric quadrilaterals in two dimensions and eight-node isoparametric hexahedral bricks in three dimensions. It is important to point out that the technique being demonstrated here is completely independent of the mesh generator chosen, and the technique is in no way restricted to quadrilateral or hexahedral elements. By and large, the mesh selection was arbitrary; the first convenient mesh was used and the results obtained do not depend upon these choices, apart from the standard approximation errors associated with the finite element method. The coarse meshes used here for illustration purposes would naturally be refined for greater accuracy in solving actual problems.

In all examples the Lamé coefficients are taken as $\lambda = \mu = 1$ (see Eq. (11)), for which case Poisson's ratio was $\nu = 0.25$. The resulting mesh integrity in the example problems appeared to be fairly independent of the choice of the Lamé coefficients, so long as Poisson's ratio remained different from 0.5 (by at least 10%.)

5.1. *Box to Circle*

This first example is somewhat trivial; it is purely a geometric problem; i.e., the only equations solved in the interior of the domain are those for a linear elastic response of a solid material to an encompassing boundary load. The idea is to force a spatially discretized box region (with sides of length $1/\sqrt{2}$) into a circle of unit diameter with a purely normal traction (compressive or tensile), prescribed according to the description in Section 3.3. The distinguishing condition applied to the entire exterior boundary of the original box-shaped configuration is

$$\begin{aligned} r_{\mathrm{DC,circle}}(x, y) &= 0, \quad (x, y) \in \Gamma, \\ &= x^2 + y^2 - 1. \end{aligned} \tag{31}$$

Nodes for every element on the exterior boundary must satisfy (31); the sensitivity of this DC to the pseudo-solid displacement (i.e., $\partial \mathbf{R}_\Gamma^d / \partial \mathbf{u}_\Gamma^d$) is easily determined using (28) as required. In the actual finite element implementation, the conditions for the imposed DCs may be generally applied, either in a weak form, such as

$$\int_\Gamma w_i^{(d)} r_{\mathrm{DC,circle}} J_s \, ds = 0,$$

where J_s represents the elemental Jacobian for the transformation from a fixed unit element edge to an element with an edge that is a distinguished FB, or in a strong statement representing a collocation at nodes or Gauss points (i.e., (31) is enforced for each node point on the boundary).

Example global Jacobian entries for the DC represented by (31) can be constructed by differentiation,

$$\frac{\partial r_{\mathrm{DC,circle},i}}{\partial d_{x,j}} = 2x \delta_{ij} w_j^{(d)},$$

$$\frac{\partial r_{\mathrm{DC,circle},i}}{\partial d_{y,j}} = 2y \delta_{ij} w_j^{(d)}.$$

The result of applying this DC with the rotation technique described in Section 3.3 is shown in Fig. 5. Three Newton iterations were required to attain a converged solution. Interestingly, the corners of the box are mapped without any difficulty, although unacceptably distorted elements result. Note the change in overall area of the computational domain from 0.50 for the box to 0.785 for the circle. It was also necessary to pin one point of the original boundary to lie at a specific location on the circle, in order to make the solution of the problem unique. That is, a condition of the form

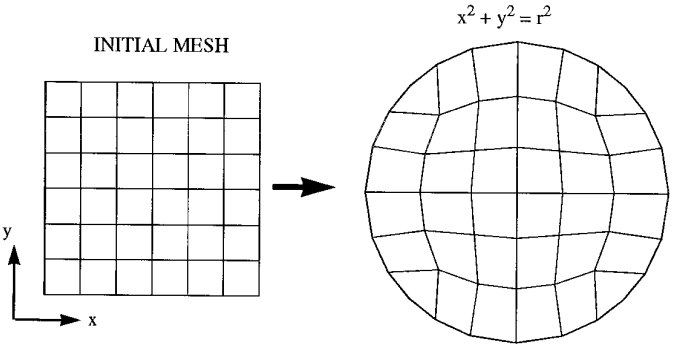

FIG. 5. Illustration of forcing a box-shaped geometry to a circular geometry with purely geometric distinguishing conditions. Note the need for the second DC to fix a point on the boundary and remove the degeneracy associated with any rotation of the pseudo-solid within the circle.

$$\begin{aligned} r_{\mathrm{DC,pt}}(x, y) &= 0, \\ &= x + \sqrt{2} \end{aligned} \tag{32}$$

forces the lower left corner of the box to coincide with one of the two points, where conditions (32) and (31) are simultaneously satisfied. Either solution is acceptable; the one that is actually chosen depends upon the convergence history of the Newton iteration process.

In fact, this need to remove rotational freedom of the pseudo-solid applies in all cases: somewhere in the domain at least one point must be uniquely specified by the applied boundary conditions on the pseudo-solid in order to eliminate the multiplicity of solutions for all possible rotations of the pseudo-solid. This necessary specification occurs naturally in problems with intersection or junction points, as is the case for all the remaining examples. This simple example, however, lacks such a point because the one distinguishing condition describes a globally smooth boundary.

5.2. *Solidification*

In some cases, conditions which distinguish the location of the boundary make no direct reference to the current position or velocity of the interface. In this case the implicit approach is a breakthrough, as this example makes clear.

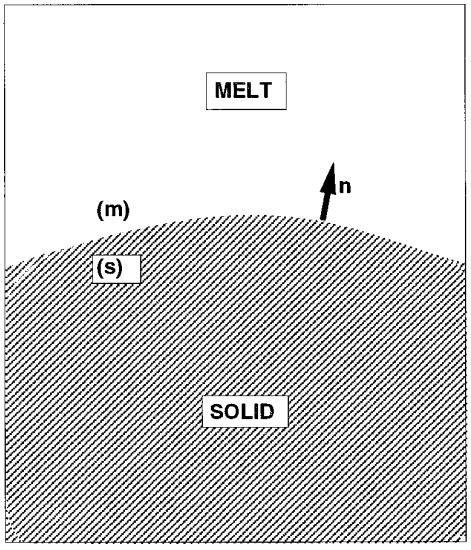

FIG. 6. Simple solidification of a melt at an interface is a representative FB and MB problem. Two distinguishing conditions constrain interfacial behavior: (i) the temperature must be continuous; (ii) the heat fluxes in the two phases are balanced by any latent heat generation at the interface.

Consider a solidification interface between a melt and a solid region of a pure material (see Fig. 6). The temperature in the melt and the solid is continuous across the interface, and a local statement of the conservation of energy balances the heat fluxes in the two regions with the latent heat generation occurring at the interface due to solidification or melting. Thus,

$$T = T_{\mathrm{mp}}, \tag{33}$$

$$\mathbf{n} \cdot \mathbf{q}|_{(s)} - \mathbf{n} \cdot \mathbf{q}|_{(m)} = \rho_{(s)} \nabla H_f^0 \mathbf{n} \cdot (\mathbf{v} - \mathbf{v}_s). \tag{34}$$

Here, T_{mp} is the melting point temperature; the subscripts s and m denote quantities for solid and melt phase, respectively; ΔH_f^0 is the latent heat of fusion; and $\mathbf{v}_s$ is the velocity of the melt surface. In this instance, the heat flux balance expressed by (34) includes the normal velocity of the unknown interface and thus provides a ''lever,'' so that the position of the interface may be updated using this equation. For steady solidification problems, i.e., $\mathbf{v_s} = \mathbf{0}$ and $\mathbf{n} \cdot \mathbf{v} = 0$, or for problems without latent heat effects, no explicit relationship between the position or velocity of the boundary and the internal field variable, T, exists, and (34) no longer provides a direct mechanism for updating the boundary position. Moving boundary problems with no explicit relation between the boundary position or velocity and the internal field variables have been termed *implicit* FB problems [9] and are excellent candidates for a full Newton iteration approach.

Figure 7 shows an example of solving a free-boundary melting problem in a cube for the position of the two interfaces separating three phases; such a problem might arise in the studies of alloy solidification with a liquidus and solidus isotherm to be identified. Using the dimensions of the cube as a length scale, the steady heat equation (15) is solved without convection or heat source terms, but with the equations for the pseudo-solid. The top of the cube-shaped domain is set to a dimensionless temperature of 1.1, and the bottom to zero. Heat is drawn off the remaining sides with a heat transfer model,

$$\mathbf{n} \cdot \mathbf{q} = h(T - T_0), \tag{35}$$

where the heat transfer coefficient $h = 2$ and the reference temperature $T_0 = 0$. These conditions comprise all of the

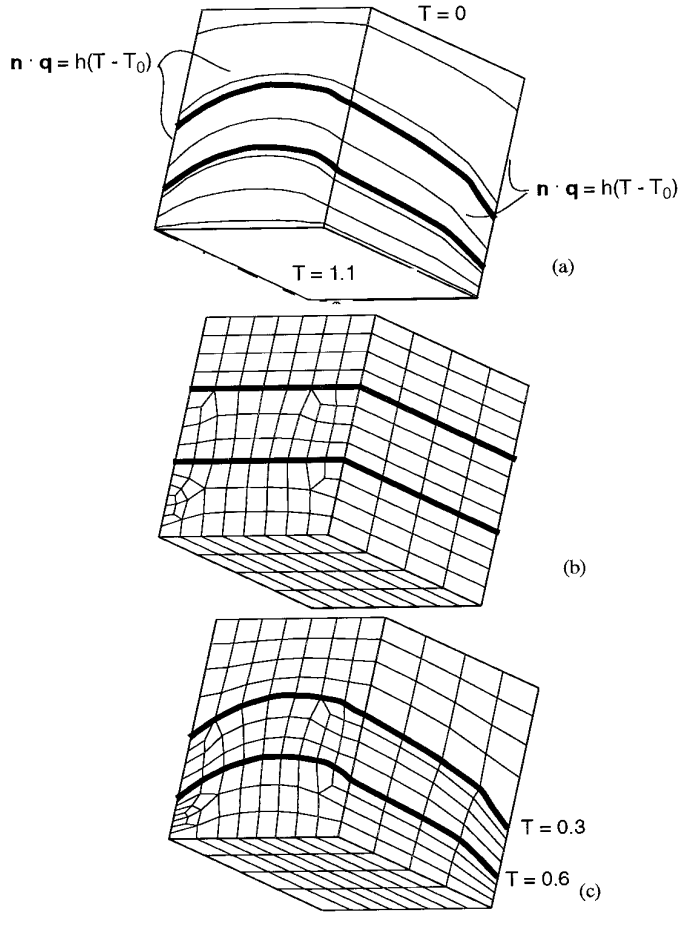

FIG. 7. Three-dimensional melting/solidification problem in a unit box: (a) Equally spaced isotherms between $T = 0$ and $T = 1.1$. Here $h = 0.2$, $T_0 = 0$, and all phases have thermal conductivity $k = 1$. Bold lines are contours placed at $T = 0.3$ and $T = 0.6$. (b) Undeformed mesh. Bold lines indicate free surfaces to be determined using DCs. (c) Deformed mesh with $T = 0.3$ and $T = 0.6$ as distinguishing conditions.

conventional boundary conditions applied to the energy equation in the computational domain.

The mesh used for this simulation is unstructured in a plane and then extruded in the third dimension. In this instance, the lack of structure is gratuitous and was selected only to illustrate the freedom of the technique from grid structure requirements.

The distinguishing conditions on the pseudo-solid consist of two kinds: geometric and thermal. Geometric constraints describing the cube faces are represented as planar surfaces of the form

$$\begin{aligned} r_{\mathrm{DC,plane},m} &= 0, \\ &= a_m x + b_m y + c_m z + d_m, \end{aligned} \tag{36}$$

where the subscript m denotes the external faces of the cube. In addition, thermal distinguishing conditions are used to determine the positions of two internal phase boundaries,

$$\begin{aligned} r_{\mathrm{DC,isotherm},n} &= 0, \\ &= T - T_{\mathrm{mp},n}, \end{aligned} \tag{37}$$

where the n subscript identifies each of the two phase transition temperatures. Equations (37) drive one phase boundary to conform to the $T = 0.3$ isosurface and the other to the $T = 0.6$ isosurface. Corresponding Jacobian entries for the sensitivity of (37) with respect to temperature provide the only nonzero contribution in the row:

$$\frac{\partial r_{\mathrm{DC,isotherm},n,i}}{\partial T_j} = w_j^{(e)}.$$

This is simply the *isotherm–Newton* technique put forth in [6].

A complete solution of the FB problem in this case required one continuation step. First, the heat equation was solved, together with associated external boundary conditions on fixed boundaries, which is a linear problem. During this first step, no internal DCs were applied, but only the external geometric DCs that fixed the shape of the cube. As the second step, the two internal interfaces were released by applying the isotherm distinguishing conditions. The solution of the first problem required but one Newton iteration because the problem was linear; the solution of the second problem that included both of the thermal DCs expressed in (37) required four Newton iterations. The L_2 norm of the residual and Newton update vector for the second step are shown in Fig. 8. The convergence behavior is clearly quadratic.

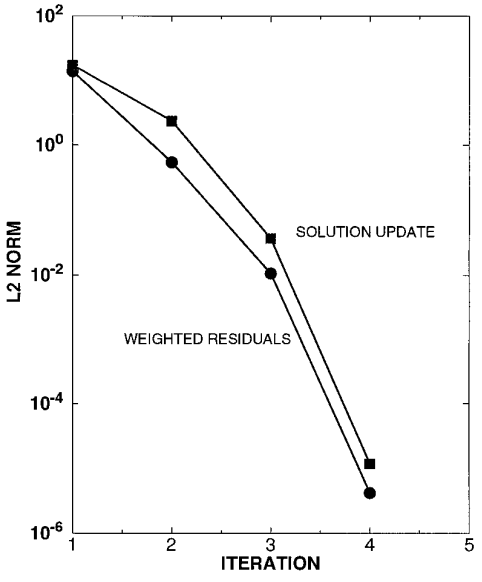

FIG. 8. Newton iteration history for melting problem. The L_2 norm of the residual vector **R** and the solution update vector **u** for the three-dimensional melting problem. Quadratic convergence is evident.

The reason the full-Newton approach works for these problems is that an implicit relation still exists; i.e., there is some sensitivity of the distinguishing condition to some unknown in the problem, such as a field variable like temperature. An implicit relationship, presuming the overall problem is well posed, is sufficient for solving the free or moving boundary problem. A fully coupled Newton iteration scheme makes the dependence more apparent, as nontrivial corrections to the interface position unknowns are determined due to nonzero Jacobian entries coupling these unknowns with the overall problem.

Moreover, while many moving boundary problems are governed by parabolic equations, there are many which are elliptic. These have been termed degenerate by Crank [10] and Sackett [9] since the distinguishing conditions for the free surface do not contain any explicit reference to boundary position or velocity. The full-Newton approach advocated here is suitable for these problems as well.

5.3. *Capillary Hydrodynamics*

Continuous liquid film coating processes epitomize FB and MB problems in capillary hydrodynamics. The example chosen here is a rigorous test of numerical methods because of several inherent features: boundary shapes that, through the normal stress balance at fluid interfaces, involve the *curvature* of the interface; field equations, like the Navier–Stokes equations, that include both significant viscous terms as well as nonlinearities due to advection

and thermophysical property variation; and parameters, such as the capillary number, defined as

$$\mathrm{Ca} = \frac{\mu U}{\gamma},$$

where μ is the viscosity, U is a characteristic velocity of the fluid, and γ is the surface tension. Intuitively, Ca expresses the relative importance of viscous forces to surface tension forces. Values of Ca can cover so large a range that often a change of iteration strategy for explicit solution algorithms is required [40]. Regarding the last feature, early work [40] on interface updating strategies based on a decoupled approach indicated that different strategies were required in different ranges of capillary numbers and, if the wrong strategy were used in a particular parameter range, then either the convergence would be distressingly slow or else the problem would not converge at all. Newton's method, coupling all of the unknowns together simultaneously, was found to be more robust for all capillary numbers.

A choice of the distinguishing condition can be used to locate the free surfaces in this class of problems; whatever boundary constraints remain after the DC is chosen are used as boundary conditions on the momentum equations. Most often the kinematic mass constraint is employed which basically states, in the absence of mass-transfer across the free boundaries, that the surface be a material surface, viz.,

$$\begin{aligned} r_{\mathrm{DC,kinematic}} &= 0, \\ &= \mathbf{n} \cdot (\mathbf{v} - \mathbf{v}_s), \end{aligned} \tag{38}$$

where $\mathbf{v}_s$ is the velocity of surface. Here, the more elaborate boundary conditions that apply to the case of interphase transport are not considered, although such conditions can also be put into the context of a full-Newton free surface algorithm [41].

Another possible choice for a distinguishing condition is to use the normal traction component of the normal stress balance,

$$\begin{aligned} r_{\mathrm{DC,normal\ stress}} &= 0, \\ &= \mathbf{nn} : \mathbf{T} - 2\mathscr{H}\gamma, \end{aligned} \tag{39}$$

where $\mathscr{H}$ represents the mean curvature of the surface. Equation (39) is in fact the only viable choice for a distinguishing condition when there is no flow, i.e., $\mathbf{v} = 0$ everywhere, thereby making it useful for startup continuation strategies and for problems in capillary hydrostatics. In this example, (38) is the distinguishing condition for all capillary surfaces. Moreover, because this example is a steady FB problem, the surface remains motionless, $\mathbf{v}_s = 0$. Other parameter ranges can cause steady coating flows to become unstable; Ref. [2] discusses these problems in greater detail. Whichever choice of DC is made, however, the remaining constraints are used as boundary conditions for the boundary value problem posed by the momentum equations.

Figure 9 illustrates a slot coating flow, which is often employed to apply thin films to moving substrates in the manufacture of paper products, adhesive tapes and films, and many other industrially relevant films. In this flow, liquid emerges from a slot-fed die which is placed at most a few millimeters from a moving substrate. Two menisci form, one upstream of the die slot, bridging the narrow gap, and one downstream of the slot die, forming the final surface of the film. Although this coating configuration can be used to deliver materials which cover a wide range in viscosity and surface tension [42], a deliberate choice here is of the case for which the representative capillary number, $\mathrm{Ca} = O(0.1)$. Under these circumstances surface tension effects are more important than those of viscosity, but not completely dominating.

Four continuation steps were required to attain a solution under the conditions indicated in Fig. 9. In the first step the Navier–Stokes equations were solved, together with boundary conditions for fixed but perfectly slippery surfaces, so as to attain a velocity field that resembles the desired result. In the second step the upstream meniscus was released and (38) applied the distinguishing condition, with 145° contact angles at the static and dynamic contact lines (see Fig. 9). These contact angles at the corners of the domain are also applied as DCs to points in much the same way as the DCs are for the physics that govern fluid and solid surfaces. A vacuum pressure was applied to this meniscus to help stabilize the location of the contact lines. In the third step the downstream meniscus was released, again employing (38) as a distinguishing condition. The incoming flow rate was chosen so that the final film thickness would be close (within 10%) to the slot gap width. A final continuation step was used to lower the flow rate, giving the result illustrated in the figure. After two or three initial iterations using a relaxed Newton algorithm, all the steady state continuation steps converged quadratically in four or five full Newton iterations.

Noteworthy in this simulation is the efficacy of the unstructured mesh. The mesh in the vicinity of the static contact line, dynamic contact line, and static separation line is refined locally in a pointwise manner. It is well known [43] that the fluid stress grows profusely in these regions and locally requires greater spatial resolution to achieve the desired overall accuracy. Where it is desirable, unstructured grids permit easier transitions to structured,

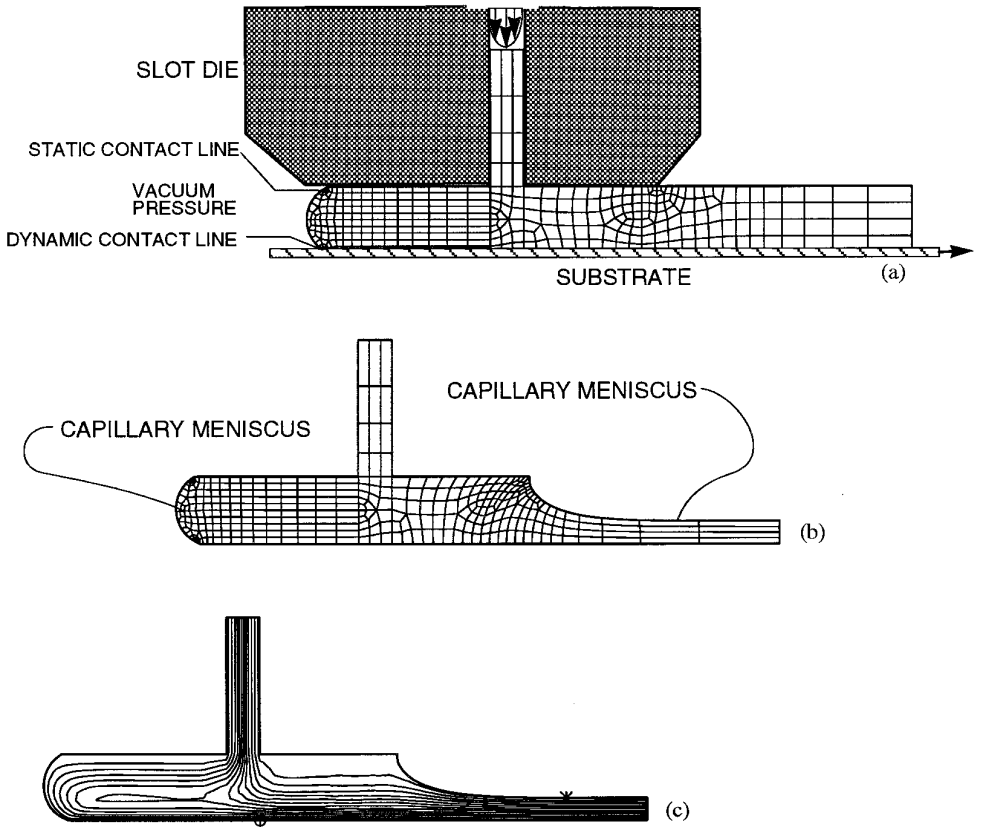

FIG. 9. Slot coating flow with capillary surfaces distinguished with the kinematic boundary condition (38): (a) undeformed mesh; (b) deformed mesh; and (c) corresponding pattern of streamlines. Physical dimensions of the flow are based on the slot gap distance between the slot die and the substrate of 0.5 mm. Density $\rho = 10^3$ kg/m^3, viscosity $\mu = 70$ mPa-s, surface tension $\gamma = 65$ mN/m, substrate speed $U = 0.133$ m/s, and contact angles of 145°.

efficient space-filling mesh topology in the interior of the domain. This flexibility in meshing, coupled with the full Newton algorithm, is one of the most powerful means to compute steady operating states for intermediate ranges of capillary numbers.

5.4. *Transient Problem*

For transient problems the relative motion of the real material and the pseudo-solid, i.e., the mesh, must be accounted for in the governing equations with additional terms as discussed in Section 3.2. In fact, the transient formulation of the technique developed in this work is in essence an ALE method [31]: the mesh moves with the material normal to the interface, as dictated by the condition (38), yet it is adjusted independently of the flow kinematics otherwise. More specifically, the additional term that arises from the transformation of the governing equation into a moving frame of Refs. [33, 44] is equivalent to the extra term that arises from the time-dependent isoparametric mapping in conventional finite-element algorithms [11, 45].

The problem examined here is the startup of a dip-coating process, whereby, after an impulsive start, a substrate is steadily withdrawn from a liquid bath. A film is entrained on the substrate and eventually a steady state is reached at which the film thickness is dictated by a competition between viscous, gravitational, and surface tension forces. The details of steady dip-coating operations of advanced materials has been recently reviewed by Schunk *et al.* [46]. To simulate the startup of the process is much more challenging because of the severe change in domain shape: the domain starts as a simple box and evolves into a shape with two disparate length scales, that of the entrained film thickness and that of the liquid bath. The use of unstructured grids as advocated by this work has made this startup problem more amenable to numerical simulation.

Figure 10 shows three time planes of a startup simulation. The reservoir is 1 cm in width and 1 cm in depth. An inflow is allowed at the lower right of the reservoir. No inflow rate is specified, as it is a key unknown in the problem. At the inflow plane, the pressure is taken as constant,

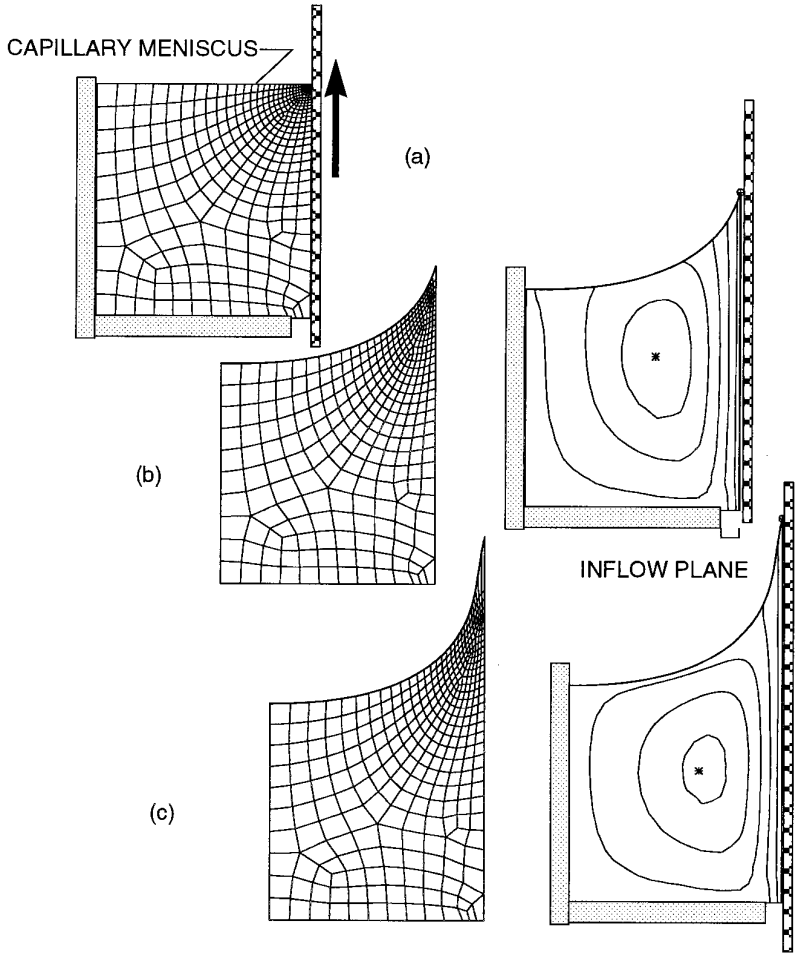

FIG. 10. Dip coating startup flow at three time planes, with the capillary surface distinguished with the kinematic boundary condition (38). Illustrated are: (a) undeformed mesh; (b) deformed mesh and pattern of streamlines at time 0.4 s; and (c) deformed mesh and streamlines at dimensionless time 0.8 s. The physical dimensions of the flow are such that the reservoir width is 1 cm. The liquid density is $\rho = 10^3$ kg/m^3, viscosity $\mu = 70$ mPa-s, surface tension $\gamma = 10$ mN/m, substrate speed $U = 0.01$ m/s, and static contact angle $\vartheta_{\text{left}} = 90°$. The gravitational acceleration is taken as 0.98 m/s^2.

with a value determined by the hydrostatic head associated with the starting depth of the reservoir. The contact angle on the upper left part of the reservoir was specified to be 90°. The moving contact line at the upper right was unconstrained with no contact angle specified. At that point the material velocity field and the contact line speed were dictated by the no-slip condition on the substrate and the kinematic condition (38). All the remaining parameters and properties of this simulation are given in Fig. 10.

The original topology of the mesh was highly refined towards the moving contact line (upper right corner of the domain). The idea is that elements close to the contact line will experience the largest distortion, and so an initial distribution of many small elements in the vicinity will minimize the local element distortion. The second time plane (Fig. 10b) corresponds to about 0.4 s. At that point, the mesh still has retained most of its original integrity. The third time plane corresponds to about 0.8 s, at which point the element containing the moving contact line has undergone severe distortion. In fact, at this point it is apparent that it would have been best to remesh the deformed domain before such a large distortion of the corner element occurred.

A total of 30 (backward Euler) time steps was required to reach the final time plane at 0.8 s. A variable time-step algorithm was employed which determined the time step, based on the solution error norm. The initial time step was 10^{-4} s and the final time step size was 10^{-1} s. With regard to the time-step size control, surface tension effects are governed by boundary curvature, which historically has been a difficult effect to incorporate into time-dependent numerical schemes for flows of incompressible fluids with free boundaries [47]. For explicit approaches to updating the interface shape, the effects of the surface tension intro-

duce time-step limitations based upon the speed of the capillary-gravity waves traveling across the surface. It is perhaps significant, especially with the low capillary number which characterizes this simulation (Ca of the order of 0.1) that such a small number of time steps are required. This evidence attests further to the power of the fully implicit approach advocated here.

6. CONCLUSIONS

This paper presents a novel technique for solving FB and MB problems. The technique uses a pseudo-solid representation of the mesh, where the mesh responds to distinguishing conditions on domain boundaries and as a linearly elastic solid on the domain interior. A fully coupled Newton–Raphson method is used for solving the nonlinear equation set resulting from finite element discretization of the physics problem and the mesh equations. The power of this solution method has been demonstrated on problems from the areas of capillary hydrodynamics and polymer and metal processing, although its use is not restricted to these areas.

One advantage of this mesh movement algorithm is that it is completely general and can be applied to any numerical discretization that has a grid, such as finite volume and spectral methods, as well as the finite element method presented here. In addition, it is the only fully coupled technique currently available that can take advantage of the efficiency of unstructured grids. The use of Newton's method with direct matrix elimination has well-known advantages that are exploited here: rapid convergence to the solution within four or five Newton iterations, facilitated convergence using first-order parameter continuation strategies, a Jacobian matrix for sensitivity and stability analysis, and straightforward application of implicit distinguishing conditions.

Unfortunately, some of the limitations of previously developed and practiced techniques, i.e., spines and elliptic grid generation, also apply to the pseudo-solid approach advocated here. For any finite discretization the possibility always exists that the deformations will be so great that the individual elements will distort to the point where they do not provide an accurate solution representation.

Also, deformations which result in a fundamental topological change in the computational domain (e.g., droplet breakup), or in the mesh itself, cannot be easily simulated using this technique and are better handled using other algorithms such as those of marker and cell [4] or volume-of-fluid [3]. For problens with extreme distortion or topological changes in the domain the issue of remeshing and interpolation comes to the fore and must be considered. The fully coupled full-Newton implementation also has the disadvantages of being expensive, compared to other methods, due to the larger system of equations needed to represent mesh displacement unknowns and due to the ill-conditioned matrices that defy most iterative techniques and generally submit only to direct elimination. The memory and CPU requirements for direct elimination are known to be high, even for the sparse matrix structures resulting from this formulation. Problems with more than 10^5 total degrees of freedom rapidly become impractical on current computing hardware.

Future studies will be directed towards an investigation of the choice of constitutive equation used to represent the mesh behavior. The Lamé coefficients for the linear elastic model have been chosen in an *ad hoc* manner. A more formal investigation of the optimal choice for these coefficients will be carried out so that they will be tuned to enhance the robustness of the mesh. In addition, nonlinear elastic models will be pursued with the hope that larger deformations of the computational domain can be followed with a minimum of mesh distortion. This would be a step forward in the direction of so-called *r*-adaptivity [48], a course of investigation whose value and importance is reaffirmed by these preliminary results.

One advantage of the mesh movement algorithm presented in this paper is that it provides a natural framework for solving problems with fluid/structure interactions, where the solid is modeled as Lagrangian and the fluid is modeled in an ALE reference frame. Modeling complexity arises because of the coupling at the boundaries between the fluid and solid domain. Fluid/structure interactions will be the focus of our long term research efforts.

ACKNOWLEDGMENTS

The development and implementation of this work was made possible by the LDRD program at Sandia National Laboratories. J. N. Shadid, H. K. Moffatt, and S. Hutchinson graciously provided us with an early version of their reacting flow finite element code that we used as a vehicle for testing the concepts presented here. We wish to acknowledge many valuable suggestions and contributions of our colleagues including R. E. Hogan, J. R. Weatherby, B. Blackwell, R. J. Cochran, D. K. Gartling, and J. A. Schutt. K. S. Chen and R. A. Cairncross provided invaluable support by extending and testing this implementation on various practical problems. We are grateful to the reviewers for their comments and suggestions.

REFERENCES

1. J. M. Floryan and H. Rasmussen, *Appl. Mech. Rev.* **42,** 323 (1989).
2. K. N. Christodoulou, S. F. Kistler, and P. R. Schunk, ''Advances in Computational Methods,'' in *Liquid Film Coating: Scientific Principles and Their Technological Implications,* edited by P. M. Schweizer and S. F. Kistler (Chapman & Hall, New York, 1996), Chap. 9.
3. C. W. Hirt and B. D. Nichols, *J. Comput. Phys.* **39,** 210 (1981).
4. F. H. Harlow and J. E. Welch, *Phys. Fluids* **8,** 2182 (1965).
5. S. Osher and J. A. Sethian, *J. Comput. Phys.* **79,** 12 (1988).
6. H. M. Ettouney and R. A. Brown, *J. Comput. Phys.* **49,** 118 (1983).
7. D. R. Lynch and K. O'Neill, ''Elastic Grid Deformation for Moving Boundary Problems in Two Space Dimensions,'' in *Finite Elements*

in Water Resources III (Wang *et al.,* Eds.), pp. 7.111–7.120, Oxford, 1980, University of Mississippi.

8. T. E. Tezduyar, M. Behr, and J. Liou, *Comput. Methods Appl. Mech. Eng.* **94,** 339 (1992).
9. G. G. Sackett, *SIAM J. Numer. Anal.* **8,** 80 (1971).
10. J. Crank, *Free and Moving Boundary Problems* (Oxford Univ. Press, New York, 1984).
11. D. R. Lynch, *J. Comput. Phys.* **47,** 387 (1982).
12. J. M. Sullivan Jr. and D. R. Lynch, *Int. J. Numer. Methods Eng.* **25,** 415 (1988).
13. R. E. Hogan, B. F. Blackwell, and R. J. Cochran, ''Numerical Solution of Two-Dimensional Ablation Problems Using the Finite Control Volume Method with Unstructured Grids,'' in *6th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, Colorado Springs, CO, 1994,* AIAA Paper No. 94-2085.
14. S. F. Kistler and L. E. Scriven, *Int. J. Numer. Methods Fluids* **4,** 207 (1984).
15. C. J. Chang and R. A. Brown, ''Finite Element Calculation of Buoyancy-Driven Convection Near a Melt/Solid Phase Boundary,'' in *Numerical Properties and Methodologies in Heat Transfer,* edited by T. M. Shih (Hemisphere, Washington, 1983), p. 283.
16. L. H. Ungar and R. A. Brown, *Phys. Rev. B* **29,** 1367 (1984).
17. D. S. Dandy and L. G. Leal, *Int. J. Numer. Methods Fluids* **9,** 1469 (1989).
18. C. Pozrikidis, *J. Fluid Mech.* **188,** 275 (1988).
19. K. N. Christodoulou and L. E. Scriven, *J. Comput. Phys.* **99,** 39 (1992).
20. H. Saito and L. E. Scriven, *J. Comput. Phys.* **42,** 53 (1981).
21. K. J. Ruschak, *Int. J. Numer. Methods Eng.* **15,** 639 (1980).
22. L. Sartor, Ph.D. thesis, University of Minnesota, 1990 (unpublished).
23. D. Wambersie and M. J. Crochet, *Int. J. Numer. Methods Fluids* **14,** 343 (1992).
24. S. F. Kistler and K. E. Palmquist, ''A Simple, Algebraic Finite-Element Discretization of Free-Surface Flows, in *Spring National Meeting, Orlando, FL, 1990, AIChE.*
25. J. U. Brackbill and J. S. Saltzman, *J. Comput. Phys.* **46,** 342 (1982).
26. G. Ryskin and L. G. Leal, *J. Fluid Mech.* **148,** 1 (1984).
27. K. Tsiveriotis and R. A. Brown, *Int. J. Numer. Methods Fluids* **14,** 981 (1992).
28. R. Lohner, K. Morgan, and O. C. Zienkiewicz, ''Adaptive Grid Refinement for the Compressible Euler Equations,'' in *Accuracy Estimates and Adaptive Refinements in Finite Element Computations,* nedited by I. Babuška, O. C. Zienkiewicz, J. Gago, and E. R. de A. Oliveira (Wiley, New York, 1986), p. 281.
29. O. Hassan, K. Morgan, J. Peraire, E. J. Probert, and R. R. Thareja, ''Adaptive Unstructured Mesh Methods for Steady Viscous Flow,'' in *AIAA 10th Computational Fluid Dynamics Conference, 1991,* p. 125.
30. B. Palmerio, *Comput. & Fluids* **23,** 487 (1994).
31. C. W. Hirt, A. A. Amsden, and J. L. Cook, *J. Comput. Phys.* **14,** 227 (1974).
32. T. Belytschko, J. M. Kennedy, and D. F. Schoeberle, *ASME J. Pressure Vessel Technol.* **102,** 62 (1980).
33. T. J. R. Hughes, W. K. Liu, and T. K. Zimmerman, *Comput. Methods Appl. Mech. Eng.* **29,** 329 (1981).
34. W. K. Liu, H. Chang, J.-S. Chen, and T. Belytschko, *Comput. Methods Appl. Mech. Eng.* **68,** 259 (1988).
35. O. C. Zienkiewicz, *The Finite Element Method in Engineering Science* (McGraw–Hill, New York, 1971).
36. B. A. Finlayson, *The Methods of Weighted Residuals and Variational Principles* (Academic Press, New York, 1972).
37. V. Girault and P.-A. Raviart, *Finite Element Methods for Navier–Stokes Equations,* Comput. Math., Vol. 5 (Springer-Verlag, New York, 1980).
38. R. A. Brown, L. E. Scriven, and W. J. Silliman, ''Computer-Aided Analysis of Nonlinear Problems in Transport Phenomena, in *New Methods in Nonlinear Dynamics,* edited by P. Holmes (SIAM, Philadelphia, 1980), p. 289.
39. T. D. Blacker and M. B. Stephenson, Technical Report SAND90-0249, Sandia National Laboratories, 1990 (unpublished).
40. W. J. Silliman and L. E. Scriven, *J. Comput. Phys.* **34,** 287 (1980).
41. P. R. Schunk and R. R. Rao, *Int. J. Numer. Methods Fluids* **18,** 821 (1994).
42. P. M. Schweizer and S. F. Kistler (Eds.), *Liquid Film Coating: Scientific Principles and Their Technological Implications* (Chapman & Hall, New York, 1995).
43. H. K. Moffatt, *J. Fluid Mech.* **18,** 1 (1964).
44. A. Huerta and W. K. Liu, *Comput. Methods Appl. Mech. Eng.* **69,** 277 (1988).
45. H. S. Kheshgi and L. E. Scriven, ''Penalty Finite Element Analysis of Unsteady Free Surface Flows,'' in *Finite Elements in Fluids,* Vol. 5, edited by R. H. Gallagher, J. T. Oden, O. C. Zienkiewicz, T. Kawai, and M. Kawahara (Wiley, New York, 1984), Chap. 19, p. 393.
46. P. R. Schunk, A. J. Hurd, and C. J. Brinker [42, Chap. 13].
47. C. W. Hirt and J. P. Shannon, *J. Comput. Phys.* **2,** 403 (1968).
48. N. Kikuchi, K. Y. Chung, T. Torigaki, and J. E. Taylor, *Comput. Methods Appl. Mech. Eng.* **57,** 67 (1986).