INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **36**: 841–865 (DOI: 10.1002/fld.159)

An assessment of a parallel, finite element method for three-dimensional, moving-boundary flows driven by capillarity for simulation of viscous sintering

Hua Zhou and Jeffrey J. Derby*,[1]

Department of Chemical Engineering and Materials Science, Army HPC Research Center, and Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, U.S.A.

SUMMARY

A parallel, finite element method is presented for the computation of three-dimensional, free-surface flows where surface tension effects are significant. The method employs an unstructured tetrahedral mesh, a front-tracking arbitrary Lagrangian–Eulerian formulation, and fully implicit time integration. Interior mesh motion is accomplished via pseudo-solid mesh deformation. Surface tension effects are incorporated directly into the momentum equation boundary conditions using surface identities that circumvent the need to compute second derivatives of the surface shape, resulting in a robust representation of capillary phenomena. Sample results are shown for the viscous sintering of glassy ceramic particles. The most serious performance issue is error arising from mesh distortion when boundary motion is significant. This effect can be severe enough to stop the calculations; some simple strategies for improving performance are tested. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: finite element method; moving boundary; surface tension; viscous sintering

1. INTRODUCTION

There are many fluid flows of engineering relevance that involve the time-dependent movement of a free surface where capillary forces are important. Such flows include those associated with coating processes [1,2], the motion of drops or bubbles [3], meniscus-defined crystal growth systems [4–6], and many others. In this paper, we present a finite element method to compute three-dimensional, time-dependent free surface flows where capillarity is important and apply this method to describe an important system which features such flows, namely the viscous sintering of glassy ceramic particles.

* Correspondence to: Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, MN 55455-0132, U.S.A. Fax: +1 612 6267246.
[1] E-mail: derby@tc.umn.edu

Received 12 September 1998
Revised 10 October 2000

Viscous sintering occurs when a compact of glassy particles is heated to a temperature high enough so that the glass flows under the action of surface tension forces. The resulting flow fills the necks between the particles and shrinks the compact, until a dense state is reached. The first quantitative approaches to model viscous sintering assumed simple but *ad hoc* geometric forms for surface evolution and derived analytical expressions describing the rate at which this evolution proceeds [7–9]. Early computational models for viscous sintering computed the evolution of two-dimensional geometries but where hampered by the difficulties associated with the accurate computation of surface curvature [10–12]. Recent two-dimensional models for planar configurations of circular particles [13–16] and axisymmetric arrangements of spherical particles [17,18] have successfully simulated viscous sintering in simple particle arrangements.

While the studies mentioned above were able to provide important insights, any real ceramic system is characterized by a three-dimensional arrangement of powder particles. In a three-dimensional system, particle rearrangement can lead to significant anisotropic shrinkage [19], which can result in unwanted residual stresses in the sintered ceramic material. Zhou and Derby [20] recently presented the first fully three-dimensional simulation of the viscous sintering of a three-particle system. In this paper, we lay out the approach for these calculations and present some new results.

Moving-boundary flows in which surface tension is important have long been studied in many disciplines; however, relatively few approaches have been applied to the solution of three-dimensional problems. Ho and Patera [21,22] developed a spectral element method that employed a three-dimensional, variational form of the free-surface traction condition and applied their method to test the stability of several falling film flows. Legat and Marchal [23] employed a three-dimensional, finite element to predict shape extrudates from a die. Pozrikidis and co-workers [24,25] have developed boundary element methods suitable to describe three-dimensional creeping flow with moving boundaries in three dimensions. Loewenberg and Hinch [26] developed similar techniques to study the deformation of viscous drops in three-dimensional flows. Thess and Orszag [27] studied three-dimensional time-dependent surface tension-driven convection by a pseudo-spectral method which is based on Fourier series in the horizontal direction and Chebyshev polynomial series in the vertical direction.

In this paper, we present a finite element method to compute three-dimensional, moving-boundary flows where capillarity is significant. The mathematical formulation of our system is presented in Section 2. Section 3 describes the methods employed, including accurate front-tracking by an arbitrary Lagrangian–Eulerian formulation [28], a robust representation of surface the curvature using the ideas put forth by Ruschak [29] and Patera and Ho [21,22], and a flexible pseudo-solid mesh moving algorithm (used here with an unstructured mesh of tetrahedral elements) coupled with implicit temporal integration, as per the method put forth by Sackinger *et al.* [30]. This section also describes the implementation of our code on a data-parallel supercomputer. In Section 4 sample results are presented for the viscous sintering problem. We comment on the strengths and weaknesses of our approach in Section 5. While the current approach appears promising for many situations, there remain serious limitations arising from mesh distortion when significant surface motion must be accommodated.

2. GOVERNING EQUATIONS

We consider the motion of an incompressible, Newtonian fluid, which is bounded by a free surface. The governing equations are taken to be the Navier–Stokes momentum balance along with continuity

$$\rho \frac{\partial v}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = \rho \mathbf{g} + \nabla \cdot \mathbf{T} \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0 \tag{2}$$

where ρ is the fluid density, $\mathbf{v}$ is the velocity vector, t is time, ∇ is the divergence operator, $\mathbf{g}$ is the gravitational vector, and $\mathbf{T}$ is the total stress tensor.

We express the total stress tensor as a function of pressure and velocity gradients using the constitutive equation for a Newtonian fluid

$$\mathbf{T} = -p\mathbf{I} + \eta(\nabla \mathbf{v} + \nabla \mathbf{v}^T) \tag{3}$$

where, p is the pressure, $\mathbf{I}$ is the identity tensor, η is the fluid viscosity, and the superscript T denotes the transpose operator.

For the cases we consider here, the fluid is bounded by a free surface where a kinematic boundary condition holds

$$\mathbf{n} \cdot \left(\mathbf{v} - \frac{\partial \mathbf{x}_s}{\partial t} \right) = 0 \tag{4}$$

as well as a force balance

$$\mathbf{n} \cdot \mathbf{T} + p_g \mathbf{n} - \gamma \kappa \mathbf{n} = 0 \tag{5}$$

In the above equations, $\mathbf{n}$ denotes an outward-pointing unit vector normal to the fluid surface, $\partial \mathbf{x}_s / \partial t$ is the time derivative of the surface position, p_g is the pressure of the gas phases surrounding the fluid, γ is the surface tension, and κ is the mean curvature of the fluid surface [31].

For moving-boundary problems with significant capillary effects, an appropriate non-dimensionalization of the momentum equation can be obtained by normalizing the spatial co-ordinates $\mathbf{x}$ with a characteristic length R, the components of the stress tensor $\mathbf{T}$ with γ/R, the fluid velocity field $\mathbf{v}$ with γ/η, time t with $R\eta/\gamma$, and surface curvature κ with $1/R^2$. The momentum equation becomes

$$Su \frac{\partial \mathbf{v}}{\partial t} + Su \mathbf{v} \cdot \nabla \mathbf{v} = Bo \frac{\mathbf{g}}{g} + \nabla \cdot \mathbf{T} \tag{6}$$

where $Su = \rho \gamma R / \eta^2$ is the Suratman number and $Bo = \rho R^2 g / \gamma$ is the Bond number.

We further simplify this general problem for the case of the viscous sintering of glassy ceramic particles, where R is on the order of 10^{-7}–10^{-4} m, ρ is on the order of 10^3 kg m^{-3}, η is on the order of 10^6–10^9 Pa s, and γ is on the order of 0.1 N m^{-1}. We then have $Su \approx 10^{-23}$–10^{-14} and $Bo \approx 10^{-9}$–10^{-3}. Since $Su \ll 1$ and $Bo \ll 1$, we ignore the intertial term and gravitational terms, and the momentum equation becomes the Stokes equation

$$\nabla \cdot \mathbf{T} = 0 \tag{7}$$

with

$$\mathbf{T} = -p\mathbf{I} + (\nabla\mathbf{v} + \nabla\mathbf{v}^T) \tag{8}$$

Without loss of generality for an incompressible fluid, we also set the pressure of the exterior gas phase to zero. The force balance at the surface then simplifies to

$$\mathbf{n} \cdot \mathbf{T} - \kappa\mathbf{n} = 0 \tag{9}$$

Note that all terms of the above equations are non-dimensional. Our description of the capillary-driven flow during viscous sintering is described by these three equations, Equations (7)–(9), along with the dimensionless forms of Equations (2) and (4).

3. NUMERICAL METHODS

The numerical solution of the above equations is made challenging by the coupling among the field equations for velocity and pressure, the interfacial conditions of capillarity, and the changing shape of the fluid domain. The following subsections address the methods employed to meet these challenges. Section 3.1 presents the spatial discretization of the governing equations by the finite element methods. Section 3.2 describes mesh generation and mesh movement strategies associated with the front-tracking methods we employ. Section 3.3 outlines temporal discretization, the solution of the resulting set of differential algebraic equations, and the parallel implementation of out methods.

3.1. Finite element formulation

We employ a pressure-stabilized Petrov–Galerkin (PSPG) finite element method [32–34] with equal-order interpolation for velocity and pressure over tetrahedral elements. We comment further about mesh generation and movement in the subsequent section. Using this method, the components of the velocity field and the pressure field are approximated as follows:

$$\mathbf{v}_x = u\mathbf{e}_x = \sum_i u^i(t)\phi^i(x, y, z)\mathbf{e}_x \tag{10}$$

$$\mathbf{v}_y = v\mathbf{e}_y = \sum_i v^i(t)\phi^i(x, y, z)\mathbf{e}_y \tag{11}$$

$$\mathbf{v}_z = w\mathbf{e}_z = \sum_i w^i(t)\phi^i(x, y, z)\mathbf{e}_z \tag{12}$$

$$p = \sum_i p^i(t)\phi^i(x, y, z) \tag{13}$$

where u^i, v^i, w^i, and p^i are unknown time-dependent coefficients which interpolate the velocity components and pressure respectively; $\mathbf{e}_k$ are unit co-ordinate vectors associated with the kth Cartesian co-ordinate, x, y, or z; ϕ^i are linear basis functions defined over a tetrahedral element; and the summation occurs over all nodes in the mesh.

We apply the Galerkin weighting and weak form transformation [35] to the momentum equations. For the sake of brevity, we show only the x-component of the momentum residuals here; the y-component and z-component residuals have a similar form

$$R_x^i = -\int_\Omega p\frac{\partial\phi^i}{\partial x}\,\mathrm{d}\Omega + \int_\Omega\left[\frac{\partial\phi^i}{\partial x}\cdot 2\frac{\partial u}{\partial x} + \frac{\partial\phi^i}{\partial y}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) + \frac{\partial\phi^i}{\partial z}\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)\right]\mathrm{d}\Omega - \int_\Gamma \mathbf{n}\cdot\mathbf{T}\cdot\phi^i\mathbf{e}_x\,\mathrm{d}\Gamma \tag{14}$$

where R_x^i denotes the x-component of the residuals equation at node i in the mesh. The first two terms on the right-hand side of the above equation are volume integrals over the entire domain of the fluid Ω while the last term is a surface integral written over the free surface Γ. We elaborate further on the form of this surface integral below.

The residual equations of the continuity equation are obtained using the PSPG method and have the following form:

$$R_c^i = \int_\Omega \psi^i(\nabla\cdot\mathbf{v})\,\mathrm{d}\Omega + \alpha\sum_K h_K^2\int_K \nabla\psi^i\cdot\nabla p\,\mathrm{d}\Omega \tag{15}$$

where R_c^i denotes the continuity residuals equation at node i in the mesh, α is a stabilization parameter, the summation in the second term is over all elements K in the mesh, and h_K is a measure of the linear size of element K. Consistent with the recommendations in [32] we set $\alpha = 0.25$. The element size, h_K, is taken to be the cube root of the element volume.

The surface integral in Equation (14) contains contributions to the force balance at the surface of the fluid. Here, only capillarity is acting, so we replace the surface integral in Equation (14) with the force balance at the free surface, Equation (9), to obtain

$$-\int_\Gamma \mathbf{n}\cdot\mathbf{T}\cdot\phi^i\mathbf{e}_x\,\mathrm{d}\Gamma = -\int_\Gamma \kappa\phi^i\mathbf{n}\cdot\mathbf{e}_x\,\mathrm{d}\Gamma \tag{16}$$

For problems with significant capillary effects, a proper representation of surface curvature, κ, in this term is critical for accuracy and robustness; many prior models for sintering phenomena were severely limited by inaccuracies in the computation of surface mean curvature [10–12]. The key idea to effectively represent this term, first put forth by Ruschak [29] and later generalized to three-dimensional cases by Patera and Ho [21,22], is to employ a weak-form representation of the left-hand side of Equation (16), so that the mean curvature is expressed in terms of first derivatives of surface shape.

We follow the ideas of Ruschak [29] and Patera and Ho [21,22] by invoking an identity from differential surface geometry [31] to express the first curvature κ of the surface in terms of first-order derivatives. Here, we again show only the x-component of the derivation for brevity

$$\int_{\Gamma} \kappa\phi^i \mathbf{e}_x \cdot \mathbf{n}\, \mathrm{d}\Gamma = -\int_{\Gamma} \nabla_{\Pi}\phi^i \cdot \mathbf{e}_x\, \mathrm{d}\Gamma + \int_{\partial\Gamma} \phi^i \mathbf{e}_x \cdot \mathbf{m}\, \mathrm{d}l \tag{17}$$

where ∇_{Π} is the surface gradient and $\mathbf{m}$ is a unit vector tangential to the surface and normal to any bounding edge of the surface (denoted by $\partial\Gamma$); see Figure 1(a).

Finally, we substitute this identity into Equation (16) and replace the surface integral of the momentum residual Equation (14) to obtain

$$\begin{aligned} R_x^i = & -\int_{\Omega} p \frac{\partial\phi^i}{\partial x}\, \mathrm{d}\Omega + \int_{\Omega} \left[\frac{\partial\phi^i}{\partial x} \cdot 2 \frac{\partial u}{\partial x} + \frac{\partial\phi^i}{\partial y}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) + \frac{\partial\phi^i}{\partial z}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\right] \mathrm{d}\Omega \\ & + \int_{\Gamma} \nabla_{\Pi}\phi^i \cdot \mathbf{e}_x\, \mathrm{d}\Gamma - \int_{\partial\Gamma} \phi^i \mathbf{e}_x \cdot \mathbf{m}\, \mathrm{d}l \end{aligned} \tag{18}$$

The y-component and z-component momentum residuals are derived in an analogous fashion. Notice that the force balance at the free surface has been incorporated directly into the momentum residual equations and that only first derivatives of the surface are needed. This idea is key to the success of this approach, since directly computing surface curvature from second derivatives of surface shape is inherently error-prone.

The remaining tasks for evaluating the momentum residuals involve computing the surface gradient operator ∇_{Π} and the edge vector $\mathbf{m}$. The surface itself is represented by a continuous function of the following form:

$$\mathbf{x}_{\mathrm{s}} = \sum_{K_{\mathrm{s}}} \mathbf{x}_{K_{\mathrm{s}}}(\xi, \eta) = \sum_{K_{\mathrm{s}}} \sum_{j} \mathbf{x}^{(j)}\phi^{(j)}(\xi, \eta) \tag{19}$$

where $\mathbf{x}_{\mathrm{s}}$ denotes the surface, K_{s} represents the surface elements of the mesh, $\mathbf{x}_{K_{\mathrm{s}}}$ are the nodal co-ordinates of the surface, ξ and η are the co-ordinates within the isoparametrically mapped surface element, j represents the local nodes within each surface element, and $\mathbf{x}^{(j)}$ and $\phi^{(j)}$ are the nodal co-ordinates and local basis functions respectively, of surface element K_{s}. Within a given surface element in its isoparametrically mapped reference co-ordinate frame, we compute a set of basis vectors from Equation (19) as follows:

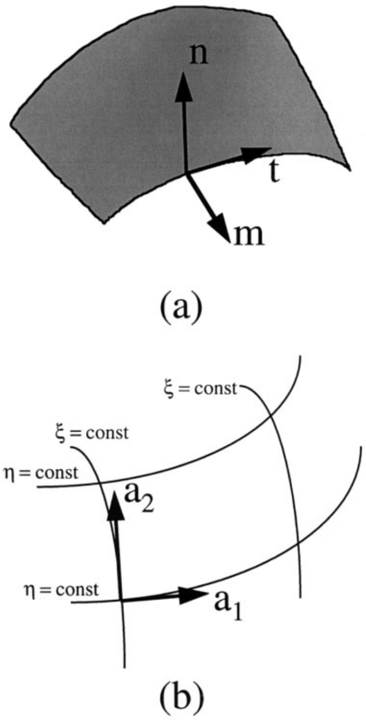

Figure 1. Definitions of normal and tangent vectors to the fluid surface. (a) Geometry of the surface in global co-ordinate frame. (b) Geometry of the surface within a surface element in its isoparametrically mapped reference co-ordinate frame.

$$\mathbf{a}_1 = \left(\frac{\partial \mathbf{x}}{\partial \xi}\right)_{K_s} = \sum_j \mathbf{x}^{(j)} \frac{\phi^{(j)}}{\partial \xi} \tag{20}$$

$$\mathbf{a}_2 = \left(\frac{\partial \mathbf{x}}{\partial \eta}\right)_{K_s} = \sum_j \mathbf{x}^{(j)} \frac{\phi^{(j)}}{\partial \eta} \tag{21}$$

$$\mathbf{a}_3 = \mathbf{a}_1 \times \mathbf{a}_2 \tag{22}$$

where × denotes the vector cross-product. These vectors are depicted in Figure 1(b). The normal to the surface is then represented by

$$\mathbf{n} = \frac{\mathbf{a}_3}{H} \tag{23}$$

where

$$H = |\mathbf{a}_3| = \sqrt{EG - F^2} \tag{24}$$

with

$$E = \mathbf{a}_1 \cdot \mathbf{a}_1 \tag{25}$$

$$F = \mathbf{a}_1 \cdot \mathbf{a}_2 \tag{26}$$

$$G = \mathbf{a}_2 \cdot \mathbf{a}_2 \tag{27}$$

The surface gradient is expressed as [31]

$$\nabla_{\Pi} = \frac{1}{H^2}\mathbf{a}_1\left(G\frac{\partial}{\partial\xi} - F\frac{\partial}{\partial\eta}\right) + \frac{1}{H^2}\mathbf{a}_2\left(E\frac{\partial}{\partial\eta} - F\frac{\partial}{\partial\xi}\right) \tag{28}$$

Finally, we need to calculate **m**, the unit vector tangential to the fluid surface and normal to the edge of the surface. Here, we assume that the edge is the seam between an intersecting fixed plane (which may be a symmetry plane) and the free surface. Let $\mathbf{n}_f$ be the unit normal to the fixed surface and $\mathbf{t}_1$ be the tangential unit vector along the edge (Figure 2(a)). Without loss of generality, we assume that the seam corresponds to the $\eta = 0$ edge of the surface element, as per Equation (19). We define orthogonal unit vectors along the seam, both of which lie in the intersecting fixed plane

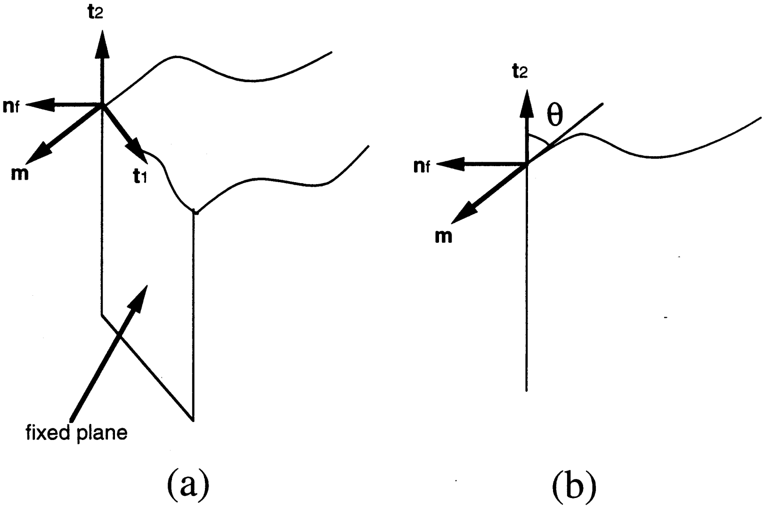

Figure 2. Definitions of normal and tangent vectors at an edge or seam of the fluid surface. (a) Geometry of the surface intersecting a fixed plane. (b) Geometry of the intersection point showing wetting angle θ.

$$\mathbf{t}_1 = \frac{\mathbf{a}_1}{|\mathbf{a}_1|} \tag{29}$$

$$\mathbf{t}_2 = \mathbf{n}_f \times \mathbf{t}_1 \tag{30}$$

where $\mathbf{a}_1$ is the same as defined in Equation (20) and

$$|\mathbf{a}_1| = \sqrt{E} \tag{31}$$

We can now define $\mathbf{m}$ with respect to a convenient geometrical measure involving the intersecting fixed plane and the fluid surface, namely the contact angle θ at which the fluid surface wets the fixed plane (Figure 2(b)). Using these measures, we represent the edge vector as

$$\mathbf{m} = \mathbf{n}_f \sin\theta - \mathbf{t}_2 \cos\theta \tag{32}$$

This representation is used to evaluate the final term on the right-hand side of Equation (14).

While we believe the particular three-dimensional formulation described above to be novel, the underlying idea of representing curvature-induced normal tractions via the weak form is well-established. For example, we have carried out extensive tests of its accuracy and convergence in two-dimensional, finite element simulations of viscous sintering [15,36]. Martinez-Herrara and Derby [15] demonstrated convergence of this numerical approach with high accuracy to the exact, analytical solution of Hopper [37,38] for the coalescence of equal cylinders by creeping viscous plane flow driven by capillarity. The accuracy of the three-dimensional formulation presented here is tested via comparison with the two-dimensional formulation of Martinez-Herrara and Derby in Section 4.1.

3.2. Mesh movement

An important and perhaps underappreciated component of all front-tracking moving-boundary methods is the strategy employed to evolve the mesh in space and time. The simultaneous goals of an effective mesh moving strategy are to accurately follow the boundary motion and to avoid excessive errors introduced by mesh deformation.

The simplest approach is algebraic mesh generation and movement. Here, nodes are placed at specified points between reference surfaces and the moving boundary. Early methods employed a Mongé projection [39] to represent the interface by a single-valued function of co-ordinate, e.g. $y = H(x, t)$, where H represented the height of an interface, and interior nodes were spaced at some fraction of the interface height [40]. The representation of more complicated surface shapes could be accommodated in a similar manner by the method of spines [41], where independent node-generator directors were specified. Such approaches have been used to compute three-dimensional, free-surface flow with relatively simple geometries, such as extrusion flows [23]. Sophisticated multi-domain algebraic methods have been successfully applied to very complicated two-dimensional moving-boundary problems [42]. While algebraic mesh movement methods are relatively straightforward, they must be used in conjunction with structured meshes, and they are relatively limited in their ability to represent complicated domain shapes, especially in three-dimensional problems.

A more flexible approach is to exploit the idea of elliptic mesh generation [43,44] for a moving mesh. In this approach, a domain mapping is generated by the numerical solution of a set of elliptic equations. These equations are derived from a variational problem for the mapping which emphasizes certain characteristics, such as mesh smoothness, orthogonality, or node spacing. The mesh is then laid out along constant-co-ordinate lines of the mapped domain. Christodoulou and Scriven [45] were among the first to couple elliptic mesh generation techniques with the solution of free-surface flow problems. Since then, many advances have been made in the context of free-surface problems; see Christodoulou *et al.* [46] and the references contained therein. However, this approach is difficult to extend to three-dimensional problems [47] and has not to our knowledge been successfully implemented with an unstructured mesh.

Because of our desire to represent complicated, three-dimensional geometries, we employ an unstructured finite element mesh in conjunction with a third approach to mesh movement, the pseudo-solid method. Here, the domain boundary motion is specified by the appropriate physics (the fluid surface motion in our problem), while co-ordinate points within the problem domain move as if the domain volume was an elastic solid. Lynch and co-workers [48] were the first to employ these ideas for the solution of moving-boundary problems. Johnson and Tezduyar [49–51] have also employed this technique for a number of fluid-structure simulations. Recently, Sackinger *et al.* [30] provided a comprehensive overview of this technique and a lucid description of its implementation. We have employed their mesh movement strategy for the problems described here.

Briefly, the approach is as follows (more explicit details are presented in [30]). The initial nodal co-ordinates of the unstructured mesh $\mathbf{x}_0$ are computed by an automated mesh generator acting on the specified initial surface geometry. The nodal positions are then updated at each time step by

$$\mathbf{x}(t) = \mathbf{x}_0 + \mathbf{d}(x, y, z, t) \tag{33}$$

where $\mathbf{x}(t)$ are the time-dependent nodal co-ordinates and $\mathbf{d}(x, y, z, t)$ is the displacement of the domain in time.

The strategy then seeks to describe the mesh as a *pseudo-solid* in as simple a manner as possible. This is accomplished by modifying Cauchy's equation to eliminate inertia and other body forces, yielding

$$\nabla \cdot \mathbf{S} = 0 \tag{34}$$

where $\mathbf{S}$ is the Cauchy stress tensor. We declare the pseudo-solid domain to act as a linear, elastic medium and employ Hooke's constitutive law

$$\mathbf{S} = \lambda \operatorname{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} \tag{35}$$

where λ and μ are Lamé coefficients, both of which are set to unity for out calculations. This choice is motivated by the prior work of Johnson [50] and Sackinger *et al.* [30], who also employed Lamé coefficients of unity. Of interest for future work using the pseudo-solid approach would be the investigation of other values; indeed, non-linear constitutive laws may also be employed to better distribute the mesh with deformation. We do not consider these modifications here, but recent attempts are reported in References [52,53].

The strain tensor $\mathbf{E}$ is given by

$$\mathbf{E} = \frac{1}{2}(\nabla\mathbf{d} + \nabla\mathbf{d}^T) \tag{36}$$

and tr($\mathbf{E}$) in Equation (35) is the trace of the strain tensor. We emphasize that the stress, strain, and displacement of the pseudo-solid described above serve only to move the nodes within the mesh as the shape of the domain evolves; these quantities are independent of and different from the physical definitions of corresponding material quantities.

We represent the nodal displacement vector $\mathbf{d}$ using the same finite element approximation as applied to the velocity field, namely

$$\mathbf{d}_x = d_x\mathbf{e}_x = \sum_i d_x^i(t)\phi^i(x, y, z)\mathbf{e}_x \tag{37}$$

$$\mathbf{d}_y = d_y\mathbf{e}_y = \sum_i d_y^i(t)\phi^i(x, y, z)\mathbf{e}_y \tag{38}$$

$$\mathbf{d}_z = d_z\mathbf{e}_z = \sum_i d_z^i(t)\phi^i(x, y, z)\mathbf{e}_z \tag{39}$$

The Galerkin finite element method is used to solve for $\mathbf{d}$ via the pseudo-solid stress Equation (34). Using standard techniques, the weak form residuals for the k-th component of pseudo-solid displacement are

$$R_{\mathrm{s}}^i = \int_\Omega \mathbf{S}:\nabla(\phi^i\mathbf{e}_k)\,\mathrm{d}\Omega - \int_\Gamma \mathbf{n}\cdot\mathbf{S}\cdot\phi^i\mathbf{e}_k\,\mathrm{d}\Gamma \tag{40}$$

where '$\mathbf{:}$' is the double dot product and $\mathbf{e}_k$ is the unit vector oriented in the k co-ordinate direction.

If a domain surface is bounded by a fixed plane (such as a symmetry plane), the component of the displacement normal to the surface is set to zero by an essential boundary condition

$$\mathbf{d}\cdot\mathbf{n} = 0 \tag{41}$$

and tangential slip conditions are imposed via natural conditions

$$(\mathbf{n}\cdot\mathbf{S})\cdot\mathbf{t} = 0 \tag{42}$$

At the free surface of the fluid, the boundary condition for the pseudo-solid displacement is enforced by setting the time derivative of the displacement to the velocity of the fluid at the surface

$$\frac{\partial \mathbf{d}}{\partial t} = \mathbf{v} \tag{43}$$

This is implemented as an essential condition; the residual equation for the displacement field, Equation (40), for the free surface unknowns are replaced with a weak form of the above condition

$$R_{\mathrm{s}}^{i} = \int_{\Gamma} \phi^{i}\left(\frac{\partial \mathbf{d}}{\partial t} - \mathbf{v}\right)\cdot \mathbf{e}_k \, \mathrm{d}\Gamma = 0 \tag{44}$$

3.3. Temporal integration and parallel solution

We represent the discretized problem as a vector of time-dependent degrees of freedom, formed by the velocity, pressure, and pseudo-solid displacement unknowns

$$\mathbf{y}(t) = (\{u^i, v^i, w^i, p^i, d_x^i, d_y^i, d_z^i,\} i = 1, \ldots, N\})^T \tag{45}$$

where N represents the total number of nodes in the finite element mesh. We then gather the residuals equation for all of the unknowns to form a differential-algebraic equation of differential/algebraic equations (DAEs) [54]

$$\mathbf{M}\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}t} = \mathbf{F}(\mathbf{y}) \tag{46}$$

where the terms involving time derivatives are placed on the left-hand-side of the equation and the remaining terms are represented as $\mathbf{F}(\mathbf{y})$. The unusual feature of the problem considered here is that the only non-zero terms in the mass matrix $\mathbf{M}$ correspond to the time derivatives of the free surface displacement in Equation (44); all remaining equations are algebraic due to the absence of explicit time derivatives in the momentum, continuity, and pseudo-solid displacement equations.

We choose to solve the above DAE simultaneously using backward-difference integration methods; such approaches have been gainfully employed for two-dimensional moving-boundary problems [15,18,39,55]. The implicit Euler algorithm is used to integrate the discretized governing equations, Equation (46), in time. This method yields the following equation to be solved at each time step:

$$\mathbf{M}(\mathbf{y}_{n+1} - \mathbf{y}_n) - \Delta t\mathbf{F}(\mathbf{y}_{n+1}) = 0 \tag{47}$$

where the subscripts denote the time step and Δt is the time step size. This method is unconditionally stable with respect to time step size [56] and thus permits the use of relatively large time steps, promoting computational economies for time-dependent simulations which would be otherwise limited by numerical instability.

Our time-dependent discretization, Equation (47), is an implicit equation for $\mathbf{y}_{n+1}$, and we employ the Newton–Raphson for its solution at every time step. We choose an initial guess for the vector of unknowns, $\mathbf{y}_{n+1}^0$, and successive updates to the unknowns vector are computed using the following iterative scheme:

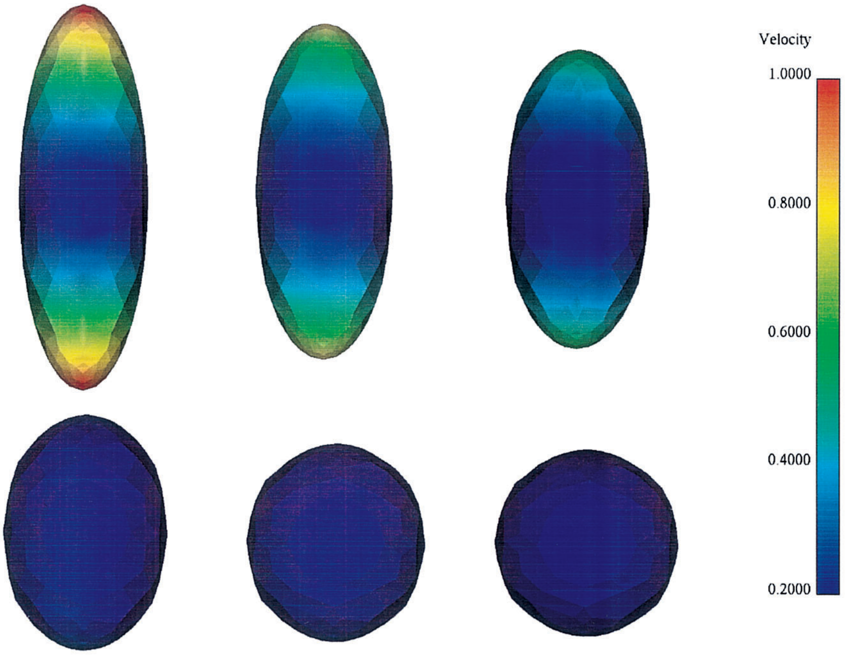

Plate 1. Shape evolution during the sintering of the axisymmetric ellipsoid at $t = 0.0$, 0.2, 0.4, 1.0, 2.0, and 3.0 respectively, from left to right and top to bottom.

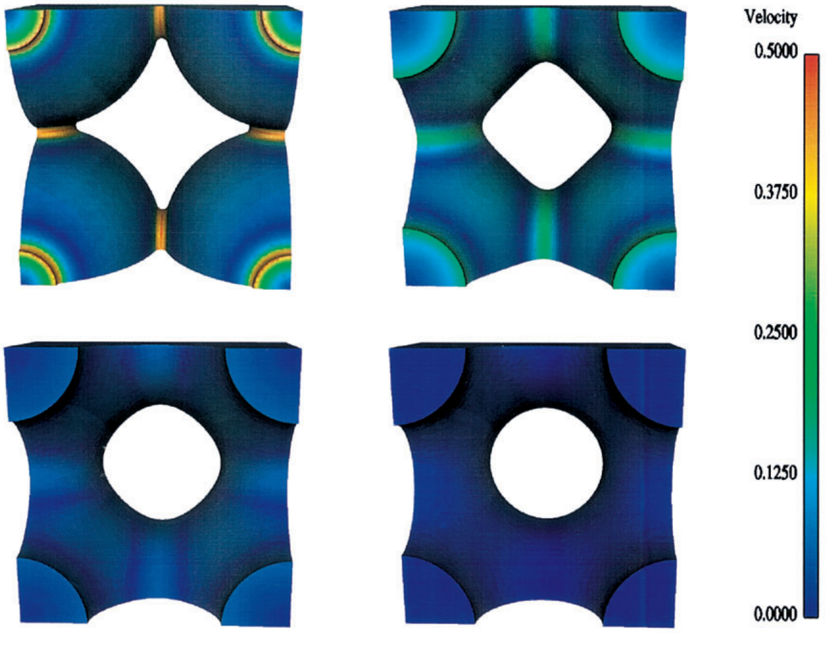

Plate 2. Shape evolution during the sintering of a constrained cubic lattice of spheres at $t = 0.0$, 0.5, 1.0, and 4.0 respectively, from left to right and top to bottom.

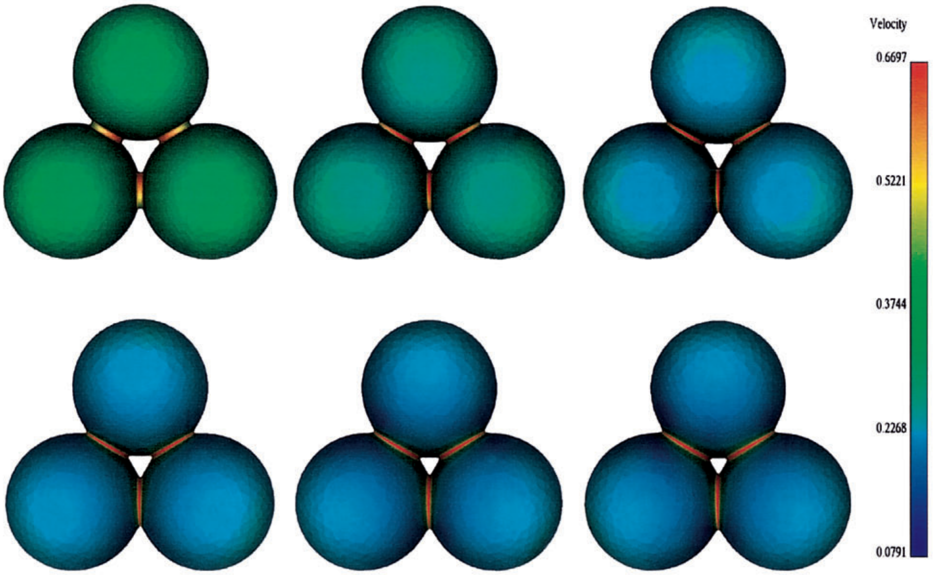

Plate 3. Shape evolution during the sintering of three unconstrained spherical particles at $t = 0$, 0.05, 0.1, 0.15, 0.2, and 0.225 respectively, from left to right and top to bottom.

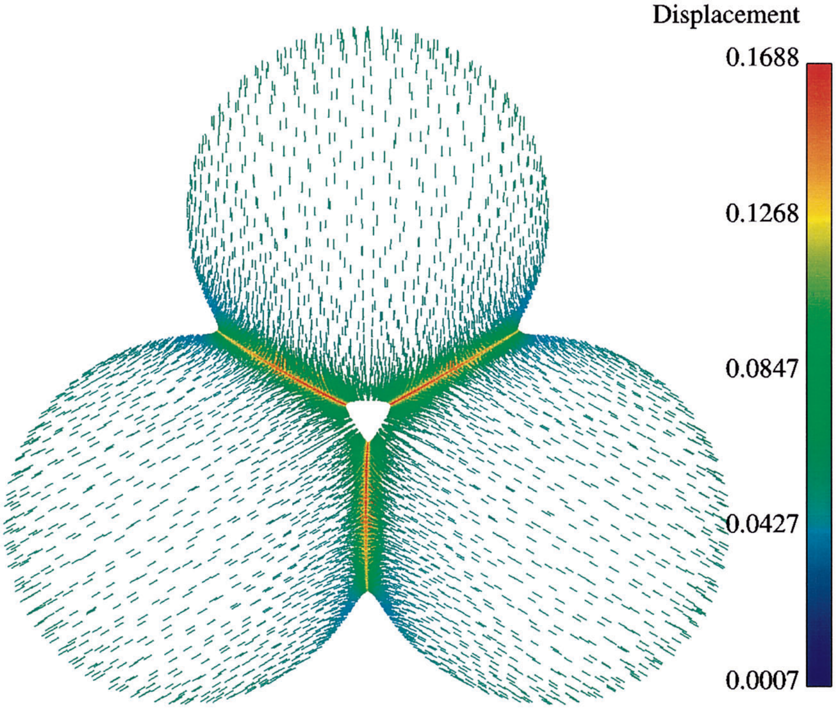

Plate 4. The surface displacement field at $t = 0.225$ for the sintering of three unconstrained spherical particles. The tails of the displacement vectors are located on the initial surface position, and the heads are located on the final surface position.

$$\mathbf{y}^{\ell+1}_{n+1} = \mathbf{y}^{\ell}_{n+1} + \delta\mathbf{y}^{\ell}_{n+1} \tag{48}$$

where ℓ is the iteration counter. The update vector $\delta\mathbf{y}^{\ell}_{n+1}$ is generated by solution of the linear equation

$$\mathbf{J}(\mathbf{y}^{\ell}_{n+1})\delta\mathbf{y}^{\ell}_{n+1} = -\mathbf{R}(\mathbf{y}^{\ell}_{n+1}) \tag{49}$$

where

$$J_{ij} = M_{ij} - \Delta t\left(\frac{\partial F_i}{\partial y_{n+1,j}}\right) \tag{50}$$

are elements of the Jacobian matrix. Particularly important to the performance of this algorithm is the correct Jacobian matrix, which must include the proper coupling between the displacement of the mesh and the field quantities. For the computations presented here, we employed a convergence criterion for the L-infinity norm of the update vector to be from 10^{-5} to 10^{-7}.

We obtain significant improvement in the iterative procedure if the initial guess, $\mathbf{y}^{0}_{n+1}$, lies near to the desired solution, $\mathbf{y}_{n+1}$. We provide a good initial guess with a second-order Adams–Bashforth predictor for time step $n+1$

$$\mathbf{y}^{0}_{n=1} = \mathbf{y}^{p}_{n+1} = \mathbf{y}_n + \frac{\Delta t_n}{2}\left[\left(2 + \frac{\Delta t_n}{\Delta t_{n-1}}\right)\frac{\mathrm{d}\mathbf{y}_n}{\mathrm{d}t} - \frac{\Delta t_n}{\Delta t_{n-1}}\frac{\mathrm{d}\mathbf{y}_{n-1}}{\mathrm{d}t}\right] \tag{51}$$

where successive estimates of the time derivatives of all variables are computed recursively by inverting the trapezoid rule [57]

$$\frac{\partial\mathbf{y}_n}{\partial t} = \frac{2}{\Delta t}(\mathbf{y}_n - \mathbf{y}_{n-1}) - \frac{\partial\mathbf{y}_{n-1}}{\partial t} \tag{52}$$

Using these formulae, we estimate the value of all unknowns and their time derivatives. Interestingly, we find better performance using a second-order predictor rather than a first-order predictor, even though the time integration scheme, Equation (47), is only first-order accurate. We also find that a good initial guess for the solution at time step at $n+1$ provides improved performance of the GMRES inner iterations described next.

We have implemented out procedures on massively parallel supercomputers in order to efficiently solve for the large number of degrees of freedom needed for accurate three-dimensional computations. The details of this implementation are provided in a series of references [58–60]; here, we provide only a very short description. The finite element mesh is first partitioned, depending upon the number of processors used, and the mesh data is then distributed across processors using Connection Machine Fortran (CMF) or high performance Fortran (HPF) constructs for the Connection Machine 5 (CM-5) or the Cray T3E respectively. The critical element of our implementation is the efficient solution of the linear algebraic equations arising during each time step. For this we use the restarted generalized minimum residual iterative technique of Saad and Schultz [61], often referred to as GMRES(m), with diagonal preconditioning. For the computations in this paper, we employ a Krylov subspace dimension of 100; within each Newton iteration, 15 restarts were employed for the first two cases, discussed in Sections 4.1 and 4.3, and 100 restarts were used in the three-sphere geometry shown in Section 4.4. Taking advantage of data locality, the element level residuals vector and the element level Jacobian matrix are calculated without inter-processor communication. Communication between processors is efficiently carried out using communication routines from the Connection Machine Scientific Software Library for the CM-5 or MPI for the T3E.

4. RESULTS

In this section, we present sample results of the viscous sintering problem for several initial geometries. For all of these results, we have generated initial unstructured meshes of tetrahedral elements using the automatic mesh generator of Johnson and Tezduyar [49]; initial meshes of varying size and complexity are shown for the test cases in Figure 3. Since the point of this paper is to examine the effectiveness of our overall approach rather than to specifically discuss the results of the sintering simulations, we have not carried out extensive mesh refinement tests to determine the accuracy of the computations performed here. Results presented in Reference [62] clearly demonstrate numerical convergence with finer meshes for several test cases. The computations were performed on the Thinking Machine Corporation CM-5 and the Cray T3E, both multiple processor, distributed memory supercomputers, at the University of Minnesota Army High Performance Computing Research Center.

4.1. Axisymmetric ellipsoid

In this first example, we consider an axisymmetric ellipsoid relaxing to its equilibrium shape by flows driven by capillarity. The initial length of the three major axes are $\ell_x = \ell_y = 0.5$ and $\ell_z = 1.5$; the initial geometry is shown in Figure 3(a). The mesh shown in Figure 3(a) consists of 1715 elements and 437 nodes, comprising 3059 total unknowns. Time integration employed 130 constant time steps to reach a dimensionless time of 4.0. The computation was performed on 64 processors of CM-5, and each time step took about 9.7 s.

Snapshots of the evolving geometry at several points in time are shown in Plate 1. As evident by the surface velocity distribution, the extended ends of the ellipsoid move rapidly inward at early times and slow as the particle shape relaxes toward a sphere, which is the geometry of mechanical equilibrium for this system.

While this is a fully three-dimensional computation, the initially axisymmetric geometry leads to two-dimensional behavior, and we can compare the results of this simulation with the results obtained from a prior two-dimensional model [18]. Figure 4 shows the lengths of the x-axis and the z-axis of the ellipsoid versus time for the two codes. The excellent agreement between the computed results gives us confidence in the veracity of the three-dimensional formulation. The small discrepancy at the end of simulations is due to slight mismatch of the volume of the initial meshes. This discrepancy becomes smaller as the three-dimensional mesh is refined [62].

4.2. Mesh distortion for the ellipsoid

The pseudo-solid formulation allows for simple computation interior mesh movement; however, we also wish to minimize distortion of the elements inasmuch as possible. We have investigated two approaches to reduce element distortion and have tested these approaches on the ellipsoidal geometry put forth in the previous section.

The first idea to reduce element distortion follows after the work of Johnson and Tezduyar [49]. They employed a linear elasticity equation to deform their mesh but dropped the elemental transformation Jacobian from their finite element residual equations. This approach is tantamount to making smaller elements stiffer; the idea is to shift most of the mesh deformation to larger elements, which are better able to accommodate deformation without

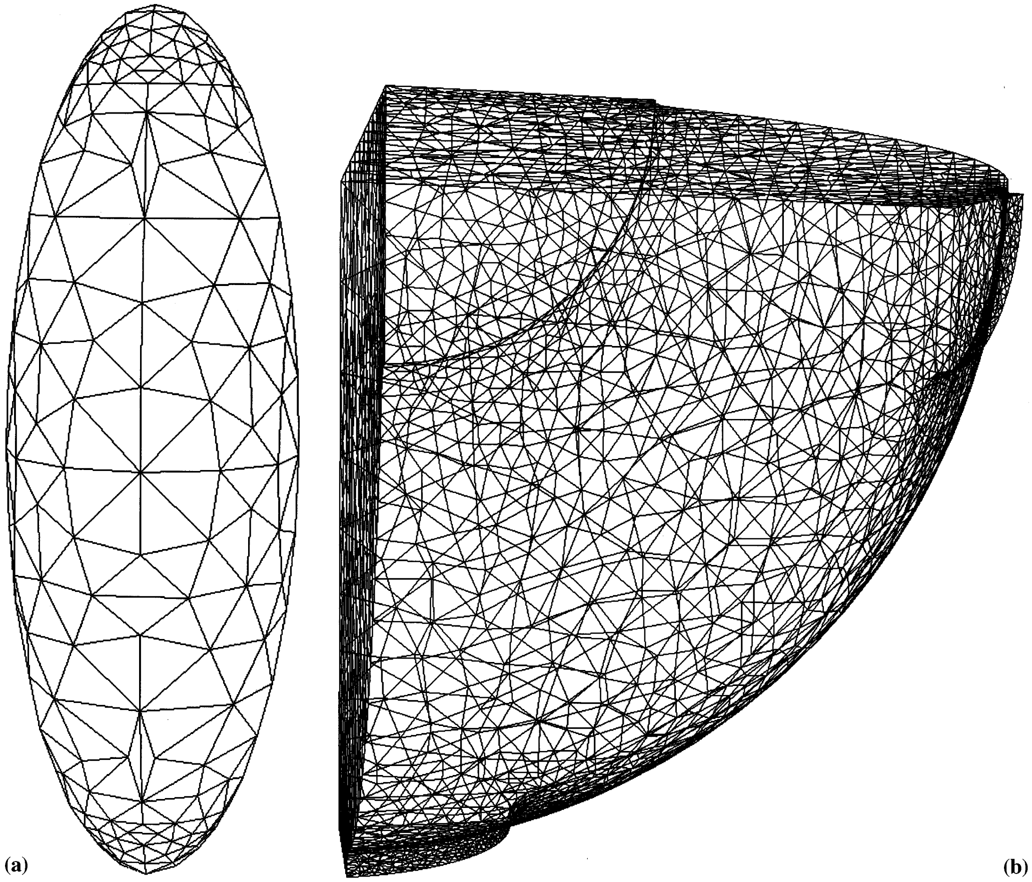

Figure 3. The initial meshes for the three cases considered here; number of elements and nodes given in the text. (a) Axisymmetric ellipsoid. (b) Constrained cubic lattice of spheres. (c) Three unconstrained spherical particles.

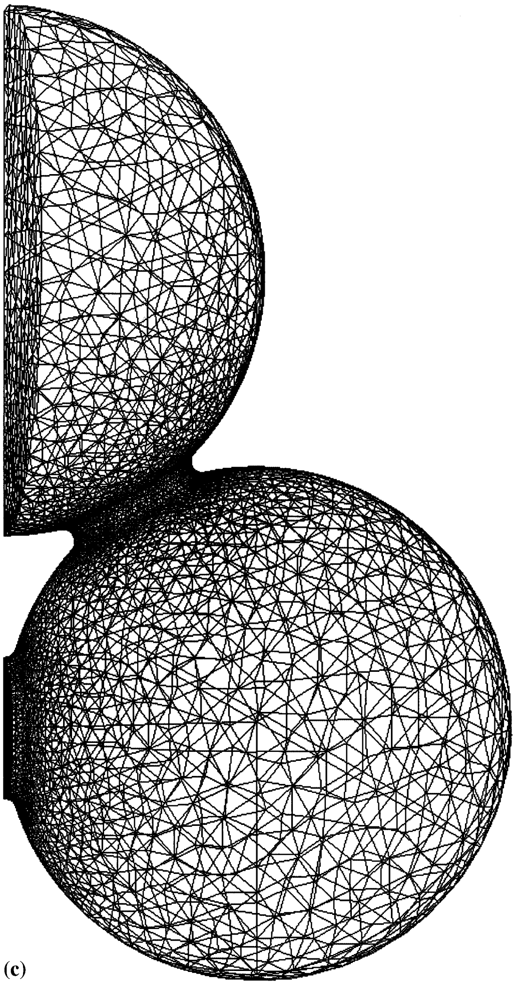

Figure 3 (*Continued*)

excessive distortion. We have repeated the ellipsoidal computation of the prior section with three strategies based on this idea: strategy one solves the original elasticity equations with no modifications, strategy two drops the elemental transformation Jacobian from the elasticity residual equations, and strategy three replaces elemental transformation Jacobian with its inverse, effectively dividing the original elemental residual equation by the square of the transformation Jacobian.

Figure 5 shows the normalized minimum element volume, $V_{\min}$, versus time for these three strategies. The normalized minimum element volume is defined as

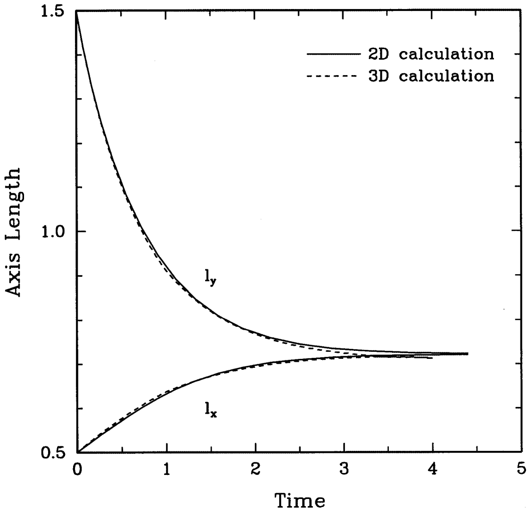

Figure 4. Comparison of three- and two-dimensional computations of the sintering of the axisymmetric ellipsoid.

$$V_{\min} \equiv \min_K \left(\frac{V_K(t)}{V_K(0)} \right) \tag{53}$$

where $V_K(t)$ represents the volume of element K at time t, $V_K(0)$ is the initial volume of element K, and the search for the minimum is conducted through all elements K of the mesh. For the simulations presented here, mesh distortion occurs primarily by the movement of the interface in the normal direction, rather than distortion via shearing motion of the mesh. Elements stretching to a high aspect ratio occupy less volume; hence, the normalized minimum element volume can be roughly interpreted as a measure of mesh distortion. It should be noted, however, that the use of normalized minimum elemental volume is not expected to be a good measure of mesh distortion for all moving-boundary problems solved via the pseudo-solid formulation, since the general motion of the mesh cannot be expected to be simple.

Strategy one (solving the original equations) results in more mesh distortion at all times. Strategies two and three perform better, with strategy three performing the best at early times and strategy two winning out at later times. Note that the discontinuities in the slopes of the curves for strategies two and three are caused by discrete changes in the element number, K, in Equation (53). Based on the results of this test and on its ease of implementation, we have employed strategy two (dropping the elemental transformation Jacobian from the pseudo-elasticity residual equations) for all of the results of this paper.

The second idea involves the definition of the nodal displacement, put forth previously in Equation (33). One approach is to define the initial mesh as the base state, i.e. $\mathbf{x}_0 \equiv \mathbf{x}(0)$, and compute all subsequent displacement vectors relative to this state. We refer to this procedure as the large deformation strategy, since the magnitude of the deformation vector grows

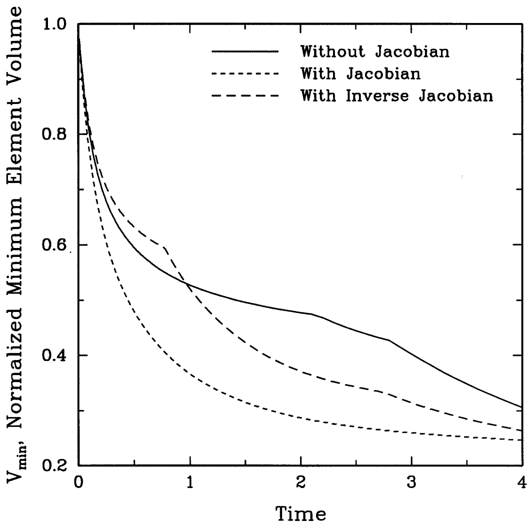

Figure 5. The normalized minimum element volume as a function of time for the three strategies: with Jacobian (strategy one), without Jacobian (strategy two), and with inverse Jacobian (strategy three).

continuously in time. Another approach, which we call the small deformation strategy, is to redefine the nodal reference at every time step, so that $\mathbf{x}_0 \equiv \mathbf{x}(t_n)$. Here, the deformation vector always remains relatively small, since it represents the *relative* mesh deformation between subsequent time steps.

One might argue that the latter small deformation strategy is appealing, since it is more consistent with the physical interpretation of infinitesimal displacements of a linearly elastic medium. On the other hand, one may also argue that this appeal has no mathematical relevance, since the underlying linearity of the pseudo-continuum mechanical equations dictates that the displacement vector of the large deformation approach is simply a linear summation of the displacement vectors of the small deformation strategy. However, this argument strictly holds only in the case of exact arithmetic; discretization error is always present in our numerical computations.

The difference between these strategies is shown in Table I for the computation of the ellipsoid test case with the large and small deformation strategies. The table lists the values of the normalized minimum element volume (see the definition above) as a function of time for the two strategies. Early on, there is little difference between the strategies; after ten time steps the normalized minimum element volume of the large deformation strategy is approximately 7 per cent smaller than that of the small deformation strategy. However, the situation continues to worsen significantly for the large deformation strategy, until a negative volume is obtained at a time of $t = 0.25$. A negative element volume indicates that an element has inverted, i.e. that a node has moved through the opposing element wall. The simulation failed on this time

Table I. The normalized minimum element volume, $V_{\min}$, for the large deformation and small deformation mesh-moving schemes

Time	$V_{\min}$ for large deformation method	$V_{\min}$ for small deformation method
0.01	0.987148	0.987148
0.02	0.981185	0.974772
0.03	0.968687	0.962851
0.04	0.950394	0.951365
0.05	0.930928	0.940294
0.06	0.910873	0.929621
0.07	0.890580	0.919327
0.08	0.870275	0.909396
0.09	0.850112	0.899812
0.10	0.830199	0.890560
...	...	...
0.22	0.360113	0.800908
0.23	0.271512	0.794900
0.24	0.150846	0.789076
0.25	—	0.783430
...	...	...
1.00	—	0.599186
...	...	...
4.00	—	0.303563

step. The small deformation strategy does not suffer this fate and successfully predicts the evolution of the geometry to time $t = 4$, as was shown in Section 4.1. The small deformation strategy is employed for all the results shown here.

4.3. Constrained cubic lattice of spheres

Here we consider spherical particles arranged in a cubic lattice and connected by small necks; see Plate 2. The initial mesh is shown in Figure 3(b); it consisted of 26034 elements and 5212 nodes for a simulation size of 36484 total unknowns. A time step of $\Delta t = 0.005$ was used until $t = 0.5$, after which a larger step size of $\Delta t = 0.01$ was employed until the end of the simulation at $t = 4$. This computation was performed on 64 processors of the Cray T3E. Because of different convergence rates, the wall-clock time spent for each time step varied from approximately 7 min at the beginning of the simulation to less than 0.4 min at the end of the simulation.

The geometry of the system is represented in Plate 2 at several times. In this case, the symmetry planes of the system are constrained, so that no shrinkage occurs. Early, there is rapid neck growth, which is fed by material initially contained in the spherical regions. Since the linear size of the system is fixed, the system approaches an equilibrium minimal surface at the end of the simulation.

4.4. Three unconstrained spherical particles

In this example, we consider a configuration of three particles all connected by necks; see the initial geometry in Plate 3. Two symmetry planes are used to reduce the size of the computational domain; however, the system size is not constrained as in the prior example. Here, the dimensions of the system are allowed to shrink in time, consistent with the behavior of real ceramic compact.

The mesh, shown in Figure 3(c), contained 69016 elements and 12952 nodes, making a simulation size of 90664 total unknowns. The computation was done on the 512-processor partition of CM-5. The computation took 45 time steps to reach a dimensionless time of 0.225, and each step took about 303 s. Total run time was approximately 3.8 h. The computation of the evolution of this complicated geometry was the most challenging of the cases shown here. To conduct this run, the dimension of the Krylow subspace for the GMRES iterative method was increased to 100, and the convergence criterion for the Newton iteration was decreased to 10^{-7} for the L-infinity norm of the update vector. The prior two cases were performed using a subspace size of 15 and a convergence criterion of to 10^{-5}. We comment on the difficulties encountered in this case in the ensuing section.

Plate 3 shows the shape evolution for the three-particle configuration. Especially apparent are the growth of the necks between the particles and the shrinking pore in the center of the configuration. Since all spheres are aligned symmetrically with respect to themselves, the inward movement of the spheres is toward the center of the system and shrinkage is isotropic. This is seen by the nearly uniform bulk displacement of the spheres shown in Plate 4. Examples of anisotropic shrinkage caused by the effects of non-symmetric particle arrangement are discussed in References [20,62].

5. CONCLUSIONS

We have developed a three-dimensional finite element model for simulation of free-surface flows where surface tension is important. The problem is formulated as a mathematical moving-boundary problem, in which we solve the field variables and surface motion simultaneously using an implicit temporal integration scheme. Unstructured meshes were employed, with mesh motion carried out using the pseudo-solid method of Sackinger *et al.* [30]. This approach yields an algorithm flexible enough to represent complicated geometries and efficient in its parallel implementation. While we have applied it here to test cases for viscous sintering, i.e. for Stokes flow driven by capillarity, the general formulation is able to represent non-linear flows and other interfacial phenomena, such as Marangoni effects.

While the overall performance of this approach is encouraging, there are two areas in which difficulties can arise. In general, the robustness of the diagonally preconditioned GMRES solution of the linear equations within the Newton iterations can be an issue. In our experience with other solutions of the Navier–Stokes equations, this solution strategy is fast but fragile [60]. Clearly, more robust algorithms are needed for the parallel solution of linear equations in such formulations, and progress is being made [63–65]. Fortunately, we are able to achieve acceptable convergence rates for the problem considered here. We believe that this is due to the

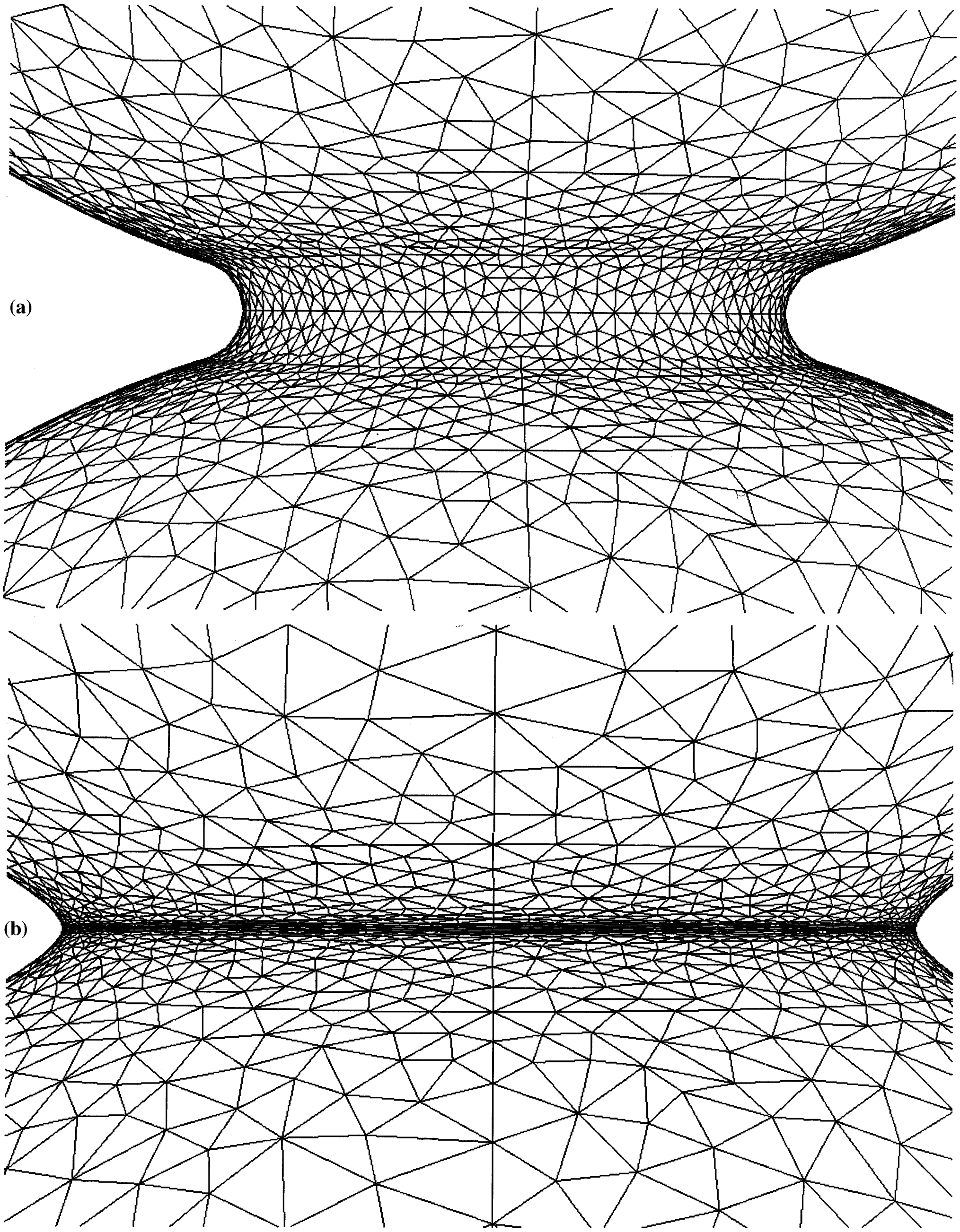

Figure 6. Surface mesh near the neck region for the sintering of three unconstrained spherical particles. (a) Initial mesh, $t = 0$. (b) Final mesh, $t = 0.225$.

nearly symmetric matrices arising from the discretized field equations for this problem and the good initial guesses for the Newton/GMRES iteration available from accurate time integration and the second-order predictor, as described in Section 3.3.

A more difficult issue to address is the degree of mesh distortion which can be tolerated as the system morphology evolves in time. This difficulty does not pose constraints in some problems, such as the first two test cases shown here. However, it does become problematic for more complicated geometries, such as that employed for the three-sphere test case of Section 4.4. The degree of mesh distortion is evident in Figure 6, where the initial and final surface meshes near the neck region are shown for the three-particle simulation. Clearly, the elements in the neck are stretched radially outward and compressed axially. We believe that the resulting high aspect ratio elements contribute substantially to discretization error. This effect has been appropriately referred to as the 'bent element blues' [66]. This discretization error reduces the accuracy of the temporal integration of the system and slows convergence in the Newton/GMRES solves at each time step. Ultimately, the simulation shown Section 4.4 is stopped well before its reaching an equilibrium state, because adequate convergence cannot be achieved.

Future algorithm development will be directed towards techniques to minimize mesh distortion under large surface deformations. Tangential conditions for the surface mesh displacement which favorably redistribute the mesh and the use of non-linear elastic models for the pseudo-solid equations are likely to improve the situation; these ideas have also been put forth by Sackinger *et al.* [30]. Mesh deformation may also be ameliorated using non-linear constitutive laws for the pseudo-solid [52,53], rather than the linear elastic law employed here. Finally, another strategy to handle large deformation is to stop the simulation before mesh distortion causes significant errors, remesh the system geometry, project the solution to the new mesh, and continue the time integration. This strategy has been employed by others, for example, Johnson and Tezduyar [50,51] have solved the three-dimensional problem of spheres setting within a fluid-filled tube. However, the problem considered here poses an additional challenge. The evolution of the fluid surface is not described by analytical shapes, so remeshing must accommodate a domain described only by the finite element representation (Equation (19)). Nevertheless, we are achieving promising initial results using ideas based on the surface remeshing ideas put forth by Cristini [67]. We will report on this approach in a future publication.

This method has already provided powerful tools to obtain new insight to the effects of three-dimensional particle arrangement in viscous sintering [62]. Clearly, many three-dimensional, free-surface problems can be gainfully analysed by this approach. Future advances in parallel computational platforms and algorithmic improvements to minimize mesh distortion effects, such as those discussed above, will make it even more flexible and powerful.

ACKNOWLEDGMENTS

This work was supported in part by the donors of The Petroleum Research Fund, administered by the ACS, a doctoral dissertation fellowship from the University of Minnesota Graduate School to H. Z., and the Army High Performance Computing Research Center under the auspices of the Department of the Army, Army Research Laboratory cooperative agreement DAAH04-95-2-0003/contract DAAH04-95-C-0008, the content of which does not necessarily reflect the position or policy of the government, and no official endorsement should be inferred. Additional computational resources were provided by the

University of Minnesota Supercomputer Institute. We also wish to thank an anonymous reviewer for comments which improved the manuscript.

REFERENCES

1. Ruschak KJ. Coating flows. *Annual Review of Fluid Mechanics* 1985; **17**: 64–89.
2. Benjamin DF, Scriven LE, *et al*. Coating flows: form and function. *Industrial Coating Research* 1991; **1**: 1–37.
3. Stone HA. Dynamics of drop deformation and breakup in viscous fluids. *Annual Review of Fluid Mechanics* 1994; **26**: 65–102.
4. Brown RA. Theory of transport processes in single crystal growth from the melt. *AIChE Journal* 1988; **34**(6): 881–911.
5. Derby JJ, Atherton LJ, Gresho PM. An integrated process model for the growth of oxide crystals by the Czochralski method. *Journal of Crystal Growth* 1989; **97**: 792–826.
6. Yeckel A, Salinger AG, Derby JJ. Theoretical analysis and design considerations for float-zone refinement of electronic grade silicon sheets. *Journal of Crystal Growth* 1995; **152**: 51–64.
7. Frenkel J. Viscous flow of crystalline bodies under the action of surface tensions. *Journal of Physics* 1945; **IX**(5): 385–391.
8. Eshelby JD. Discussion of 'Seminar on the Kinetics of Sintering'. *Metallurgical Transactions* 1949; **185**: 796–813.
9. Mackenzie JK, Shuttleworth RA. A phenomenological theory of sintering. *Proceedings of the Physical Society* 1949; **62**(12-B): 833–852.
10. Exner HE. Principles of single phase sintering. *Reviews on Powder Metallurgy and Physical Ceramics* 1979; **1**(1–4): 7–251.
11. Hiram Y, Nir A. A simulation of surface tension driven coalescence. *Journal of Colloid Interface Science* 1983; **95**(2): 462–470.
12. Kuiken HK. Viscous sintering: the surface-tension-driven flow of a liquid form under the influence of curvature gradients at its surface. *Journal of Fluid Mechanics* 1990; **214**: 503–515.
13. van de Vorst GAL, Mattheji RMM, Kuiken HK. A boundary element solution for two-dimensional viscous sintering. *Journal of Computational Physics* 1992; **100**: 50–63.
14. van de Vorst GAL. Intergal method for a two dimensional Stokes flow with shrinking holes applied to viscous sintering. *Journal of Fluid Mechanics* 1993; **257**: 667–689.
15. Martinez-Herrera JI, Derby JJ. An analysis of capillary-driven viscous flows during the sintering of ceramic powders. *AIChE Journal* 1994; **40**(11): 1794–1803.
16. van de Vorst GAL. Integral formulation to simulate the viscous sintering of a two-dimensional lattice of periodic unit cells. *Journal of Engineering Mathematics* 1996; **30**: 97–118.
17. Jagota A, Dawson PR. Simulation of the viscous sintering of two particles. *Journal of the American Ceramic Society* 1990; **73**(1): 173–177.
18. Martinez-Herrera JI, Derby JJ. Viscous sintering of spherical particles via finite element analysis. *Journal of the American Ceramic Society* 1995; **78**(3): 645–649.
19. Petzow G, Exner HE. Particle rearrangment in solid state sintering. *Zeitschrift für Metalkunde* 1976; **67**: 611–618.
20. Zhou H, Derby JJ. Three-dimensional finite element analysis of viscous sintering. *Journal of the American Ceramic Society* 1998; **81**(3): 533–540.
21. Ho L-W, Patera AT. A legendre spectral element method for simulation of unsteady incompressible viscous free-surface flows. *Computer Methods in Applied Mechanics and Engineering* 1990; **80**: 355–366.
22. Ho L-W, Patera AT. Variational formulation of three-dimensional free-surface flows: natural imposition of surface tension to boundary conditions. *International Journal for Numerical Methods in Fluids* 1991; **13**: 691–698.
23. Legat V, Marchal J-M. Prediction of three-dimensional general shape extrudates by an implicit iterative sheme. *International Journal for Numerical Methods in Fluids* 1992; **14**: 609–625.
24. Pozrikidis C, Thoroddsen ST. The deformation of a liquid film flowing down an inclined plane wall over a small particle arrested on the wall. *Physics of Fluids A* 1991; **3**: 2546–2558.
25. Kennedy MR, Pozrikidis C, Skalak R. Motion and deformation of liquid drops, and the rheology of dilute emulsions in simple shear flow. *Computers and Fluids* 1994; **23**: 251–278.
26. Loewenberg M, Hinch EJ. Numerical simulation of a concentrated emulsion in shear flow. *Journal of Fluid Mechanics* 1996; **321**: 395–419.
27. Thess A, Orszag SA. Surface-tension-driven Bénard convection at infinite Prandtl number. *Journal of Fluid Mechanics* 1995; **283**: 201–230.
28. Donea J. Arbitary Lagrangian–Eulerian finite element methods. In *Computational Methods for Transient Analysis*, Belytschko T, Hughes TJR (eds). North Holland: Amsterdam, 1983; Chapter 10.

29. Ruschak KJ. A method for incorporating free boundaries with surface tension in finite element fluid-flow simulators. *International Journal for Numerical Methods in Engineering* 1980; **15**: 639–648.
30. Sackinger PA, Schunk PR, Rao RR. A Newton–Raphson pseudo-solid domain mapping technique for free and moving boundary problems: a finite element implementation. *Journal of Computational Physics* 1996; **125**: 83–103.
31. Weatherburn CE. *Differential Geometry of Three Dimensions*. Cambridge University Press: London, 1927.
32. Hughes TJR, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška–Brezzi condition: A stable Petrov–Galekin formulation of Stokes problem accomodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**: 85–99.
33. Hughes TJR, Franca LP. A new finite element formulation for computational fluid dynamics: VII. The Stokes problem with various well-posed boundary conditions: Symmetric formulations that converge for all velocity/pressure spaces. *Computer Methods in Applied Mechanics and Engineering* 1987; **65**: 85–96.
34. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. In *Advances in Applied Mechanics*, Hutchinson JW, Wu TY (eds). Academic Press: New York, 1992; 1–44.
35. Hughes TJR. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1987.
36. Martinez-Herrera JI. Finite element analysis of the sintering of ceramic powders. PhD thesis, University of Minnesota, 1995.
37. Hopper RW. Coalescence of two equal cylinders: exact results for creeping viscous plane flow driven by capillarity. *Journal of the American Ceramic Society (Communications)* 1984; **67**: C262–C264.
38. Hopper RW. Plane stokes flow driven by capillarity on a free surface. *Journal of Fluid Mechanics* 1990; **213**: 349–375.
39. Ungar LH, Ramprasad N, Brown RA. Finite element methods for unsteady solidification problems arising in prediction of morphological structure. *Journal of Scientific Computing* 1988; **3**: 77–108.
40. Nickell RE, Tanner RI, Caswell B. The solution of viscous incompressible jet and free surface flows using finite-element methods. *Journal of Fluid Mechanics* 1974; **65**: 189–206.
41. Kistler SF, Scriven LE. Coating flow theory by finite-element and asymptotic analysis of the Navier–Stokes system. *International Journal for Numerical Methods in Fluids* 1984; **4**: 207–229.
42. Kistler SF, Palmquist KE. A simple, algebraic finite-element discretization of free surface flows. Presentation at AIChE Spring National Meeting, Orlando, FL, March, 1990.
43. Thompson JF, Warsi ZUA, Mastin CW. Boundary-fitted co-ordinate system for numerical solution of partial differential equations—a review. *Journal of Computational Physics* 1982; **47**: 1–108.
44. Thompson JF, Warsi ZUA, Mastin CW. *Numerical Grid Generation*. Elsevier: New York, 1985.
45. Christodoulou KN, Scriven LE. Discretization of free surface flows and other moving boundary problems. *Journal of Computational Physics* 1992; **98**: 1–17.
46. Christodoulou KN, Kistler SF, Schunk PR. Advances in computational methods. In *Liquid Film Coating: Scientific Principles and Their Technological Implications*, Schweizer PM, Kistler SF (eds). Chapman & Hall: New York, 1996; Chapter 9.
47. Saeger RB, Davis HT, Scriven LE. Adaptive elliptic grid generation for 3-D elemental structures. Technical Report 90/73, University of Minnesota Supercomputer Institute, Minneapolis, MN, 1990.
48. Lynch DR, O'Neill K. Continuously deforming finite elements for the solution of parabolic problems, with and without phase change. *International Journal for Numerical Methods in Engineering* 1981; **17**: 81–96.
49. Johnson AA, Tezduyar TE. Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**: 73–94.
50. Johnson AA. Mesh generation and update strategies for parallel computation of flow problems with moving boundaries and interfaces. PhD thesis, University of Minnesota, 1995.
51. Johnson AA, Tezduyar TE. 3D simulation of fluid–particle interactions with the number of particles reaching 100. *Computer Methods in Applied Mechanics and Engineering* 1997; **145**: 301–321.
52. Cairncorss RA, Schunk PR, Baer TA, Rao RR, Sackinger PA. A finite element method for free-surface flows of incompressible fluids in three dimensions, part I: boundary-fitted mesh motion. *International Journal for Numerical Methods in Fluids* 2000; **33**: 375–403.
53. Baer TA, Cairncross RA, Schunk PR, Rao RR, Sackinger PA. A finite element method for free-surface flows of incompressible fluids in three dimensions, part II: dynamic wetting lines. *International Journal for Numerical Methods in Fluids* 2000; **33**: 405–427.
54. Petzold L. Differential/algebraic equations are not ODEs. *SIAM Journal of Scientific and Statistical Computing* 1992; **3**: 367–384.
55. Derby JJ, Atherton LJ, Thomas PD, Brown RA. Finite element methods for analysis of the dynamics and control of Czochralski crystal growth. *Journal of Scientific Computing* 1987; **2**(4): 297–343.

56. Gear CW. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall: Englewood Cliffs, NJ, 1971.
57. Gresho PM, Lee R, Sani RL. On the time-dependent solution of the incompressible Navier–Stokes equations in two and the dimensions. *Recent Advances in Numerical Methods in Fluids* 1980; **1**: 27–79.
58. Salinger AG, Xiao Q, Zhou Y, Derby JJ. Massively parallel finite element computations of three-dimensional, time-dependent, incompressible flows in materials processing systems. *Computer Methods in Applied Mechanics and Engineering* 1994; **119**: 139–156.
59. Xiao Q, Salinger AG, Zhou Y, Derby JJ. Massively parallel finite element analysis of coupled, incompressible flows: a benchmark computation of baroclinic annulus waves. *International Journal for Numerical Methods in Fluids* 1995; **21**: 1007–1014.
60. Yeckel A, Derby JJ. Parallel computation of incompressible flows in materials processing: Numerical experiments in diagonal preconditioning. *Parallel Computing* 1997; **23**: 1379–1400.
61. Saad Y, Schultz MH. GMRES: a generalized minimal algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing* 1986; **7**: 856–869.
62. Zhou H. Three-dimensional simulation of viscous sintering with finite element method. PhD thesis, University of Minnesota, 1999.
63. Saad Y. Highly parallel preconditioners for general sparse matrices. In *Recent Advances in Iterative Methods*, *IMA Volumes in Mathematics and Its Applications*, vol. 60, Golub G, Luskin M, Greenbaum A (eds). Springer: Berlin, 1994; 387–402.
64. Shadid JN, Tuminaro RS. A comparison of preconditioned nonsymmetric Krylov methods on a large-scale mind machine. *SIAM Journal of Scientific and Statistical Computing* 1994; **15**(2): 440–459.
65. Chapman A, Saad Y, Wigton L. High order ILU preconditioners for CFD problems. Technical Report 96/14, University of Minnesota Supercomputer Institute, Minneapolis, MN, 1996.
66. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method: Advection–Diffusion and Isothermal Laminar Flow* (1st edn). Wiley: New York, 1998.
67. Cristini V. Drop dynamics in viscous flow. PhD thesis, Yale University, 2000.