LA-UR-96- 0088

CONF-960162--3

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36

TITLE: **VOLUME TRACKING OF INTERFACES HAVING SURFACE TENSION IN TWO AND THREE DIMENSIONS**

AUTHOR(S): D(ouglas) B. Kothe, T-3
W(illiam) J. Rider, CIC-3
S(tewart) J. Mosso, X-3
J(erry) S. Brock, T-3
J. I. Hochstein, U. of Memphis

SUBMITTED TO: *34th Aerospace Science Meeting and Exhibit, January 15–18, 1996, Reno, Nevada*

Los Alamos

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

FORM NO. 836 R4
ST. NO. 2629 5/81

ROUGH DRAFT

AIAA 96-0859

Volume Tracking of Interfaces Having Surface Tension in Two and Three Dimensions

D.B. Kothe, W.J. Rider, S.J. Mosso, and J.S. Brock
Los Alamos National Laboratory
Los Alamos, NM

J.I. Hochstein
The University of Memphis
Memphis, TN

VOLUME TRACKING OF INTERFACES HAVING SURFACE TENSION IN TWO AND THREE DIMENSIONS*

D.B. Kothe, W.J. Rider, S.J. Mosso, and J.S. Brock
Los Alamos National Laboratory
Los Alamos, NM

J.I. Hochstein
The University of Memphis
Memphis, TN

Abstract. Solution algorithms are presented for tracking interfaces with piecewise linear (PLIC) volume-of-fluid (VOF) methods on fixed (Eulerian) two-dimensional (2-D) structured and three-dimensional (3-D) structured and unstructured grids. We review the theory of volume tracking methods, derive appropriate volume evolution equations, identify and present solutions to the basic geometric functions needed for interface reconstruction and volume fluxing, and provide detailed algorithm templates for modern 2-D and 3-D PLIC VOF interface tracking methods. We discuss some key outstanding issues for PLIC VOF methods, namely the method used for time integration of fluid volumes (operator splitting, unsplit, Runge-Kutta, etc.) and the estimation of interface normals. We also present our latest developments in the continuum surface force (CSF) model for surface tension, namely extension to 3-D and variable surface tension effects. We identify and focus on key outstanding CSF model issues that become especially critical on fine meshes with high density ratio interfacial flows, namely the surface delta function approximation, the estimation of interfacial curvature, and the continuum surface force scaling and/or smoothing model. Numerical results in two and three dimensions are used to illustrate the properties of these methods.

1. Introduction

The accurate modeling of interfacial flows requires high fidelity algorithms for the kinematics and dynamics of interfaces. Algorithms for interface kinematics must address the discrete representation of the interface and its advection through the computational domain. Algorithms for interface dynamics must model physics specific to and localized at the interface (e.g., phase change and surface tension). The numerical techniques chosen to model interface kinematics and dynamics are especially important in finite-difference Eulerian methods designed to simulate flows with interfaces of arbitrarily complex topology. In these schemes the computational grid remains stationary, so an interface algorithm must minimize diffusion by maintaining a compact interface thickness without sacrificing the robustness necessary to meet the topology demands. The algorithm must be amenable to three dimensions, and incorporation of additional interface physics should be straightforward.

We are interested in modeling a general class of interfacial flows, which are defined to be any flow, ranging from incompressible to high-speed, that involves multiple fluids with differing properties. Interfaces delineating these fluids are characterized by abrupt changes in fluid properties, and might also be the site of localized phenomena such as surface tension or phase change. We are interested in modeling fluids that are not in general intimately mixed, i.e., there remains a discernible (but in general topologically complex) interface between the fluids that resolvable in the computational model. By resolvable we mean that fluid parcels bounded by an interface must be larger than the mesh spacing. Examples of the interfacial flows we wish to model are free surface flows, where a water/air interface is characterized by surface tension and sharp changes in viscosity and density; and impact dynamics, such as rod a penetrating a plate, where metal/metal and metal/air interfaces are characterized by abrupt changes in material strength properties. In such cases, the ability to represent interface dynamics accurately dictates whether or not the overall flow is modeled reliably.

Our goal is to model interfacial flows, whether it be free surface flows [4,5] or impact dynamics [40], with the same underlying interface (kinematic and dynamic) algorithm. Our interface algorithms fall under the general class of immersed interface methods [20], based on the pioneering work of Peskin [22]. In these methods, a fixed (Eulerian) grid is *not* in general aligned with interfaces. Interfaces are instead allowed to have arbitrarily

*Send correspondence to Doug Kothe (*dbk@lanl.gov*), MS-B216, Fluid Dynamics Group T-3, Los Alamos National Laboratory (LANL), Los Alamos, NM, 87545. This work performed under the auspices of the U.S. DOE by LANL under Contract W-7405-ENG-36.

complex topology, which is a design feature necessary to model important topological features such as coalescence and breakup. Special methods must therefore be devised to model the representation of the interface on the grid as well as its movement across the grid. Interface dynamics, or those physics specific to the interface, are modeled as a localized volumetric force ("source term"), which typically takes the form of the product of an approximate delta function and the appropriate interface physics per unit interfacial area. The magnitude of the interface volumetric force typically falls to zero at some prescribed distance away from the interface. The interface is not a perfect discontinuity, instead being a transition region having a width of at least one mesh spacing.

This paper is another in a recent series of studies focused on the design and implementation of robust and accurate methods for modeling interfacial flows [2,24,33,34,38,39]. While our attention in the past has been primarily on modeling a general class of free surface flows [3,4,5], our current efforts are targeted toward the reliable simulation of casting processes currently in use at the Los Alamos National Laboratory [37]. These casting processes encompass a wide breadth of complex physical phenomena, ranging from the (possibly turbulent) fluid flow and free surface dynamics, which govern the mold filling process, to the heat transfer and phase change process, which are important as the part cools and solidifies. Grain growth and material response (e.g., residual stress buildup) commence as the part cools yet further.

Our current focus is on the development of numerical methods needed to model the free surface flow during the mold filling stage of a casting process. Mold filling involves the insertion of approximately incompressible liquid metal alloys into complex 3-D molds that are be constructed from sand, wax, stainless steel, graphite, etc. The insertion usually takes place via gravity-pour or pressure-injection into a pre-designed system of runners, sprues, and gates. The mold-filling process is characterized by topologically-complex metal/air interfaces having high density ratios (> 5000) and high surface tension values (10^2 - 10^3 dynes/cm).

Perhaps the most challenging aspect of developing simulation tools for modern casting processes is the faithful representation of the complex 3-D mold/part geometries. We have therefore chosen to partition these geometries with fully unstructured meshes, which has necessitated the generalization and extension of our current structured, orthogonal mesh algorithms. Efforts are therefore currently underway to extend the 2-D incompressible flow algorithm documented in [2] to 3-D unstructured meshes, and will be the subject of a future paper. The purpose of this paper is to detail our recent efforts in developing algorithms for volume tracking of interfaces having surface tension, which are needed in modeling the mold filling stage of a casting process.

This paper is outlined as follows. In section (2) we review the theory of volume tracking methods and derive appropriate volume evolution equations. In section (3) we identify and present solutions to the basic geometric functions needed for interface reconstruction and volume fluxing, and provide detailed algorithm templates for modern 2-D and 3-D PLIC (piecewise linear interface calculation) VOF tracking methods. We discuss some key outstanding issues for PLIC VOF methods, such as the method used for time integration of fluid volumes (operator splitting, unsplit, Runge-Kutta, etc.). In section (4) we discuss our recent efforts in the estimation of interface normals, which is of crucial importance for volume tracking methods and surface tension models. In section (5) we present our latest developments in the continuum surface force (CSF) model for surface tension. In particular, we identify key outstanding CSF model issues that become especially critical on fine meshes with high density ratio interfacial flows, namely the surface delta function approximation, the estimation of interfacial curvature, and the continuum surface force scaling and/or smoothing model. Finally, in section (6), 2-D and 3-D numerical results are presented to illustrate the properties of our methods.

2. Volume-Based Methods for Tracking Fluid Interfaces

As a first step in our discussion of volume tracking methods we will derive an equation for the evolution of volumes and volume fractions. It is our intention that this will motivate further development of the numerical method and place it on firm footing technically. This counters the popular view that volume tracking is purely heuristic and lacks rigor.

Our derivation will start from a basic princple and the Reynolds transport theorem. Let V be the total fluid volume and f^k be the fractional volume of the k th fluid, defined as

$$V^k = \int f^k dV; \quad M^k = \int f^k \rho^k dV \tag{1}$$

where . For an Eulerian grid the volume of computational cells is invariant

$$V = \sum_k V^k; \quad \sum_k f^k = 1. \tag{2}$$

$$\frac{d\rho^k}{dt} = -\rho^k(\nabla \cdot \mathbf{u}^k) = -\rho^k(\nabla \cdot \mathbf{u}) \tag{3}$$

since we assume that $\mathbf{u}^k = \mathbf{u}$. Since $\rho^k = M^k/V^k$ and $dM^k/dt = 0$, then

$$\frac{dV^k}{dt} = V^k(\nabla \cdot \mathbf{u}). \tag{4}$$

Substituting $V^k = f^k V$ into (4), we obtain

$$\frac{df^k}{dt} = \frac{\partial f^k}{\partial t} + \mathbf{u} \cdot \nabla f^k = 0. \tag{5}$$

which can be written as

$$\frac{\partial f^k}{\partial t} + \nabla \cdot (\mathbf{u} f^k) = f^k(\nabla \cdot \mathbf{u}). \tag{6}$$

From (6), we see that the volume fraction f^k is a Lagrangian invariant. Integrating (6) over volume:

(7)

$$\sum_k \nabla \cdot (\mathbf{u} V^k) = \sum_k V^k(\nabla \cdot \mathbf{u}) , \tag{8}$$

and, after rearrangment,

$$(\nabla \cdot \mathbf{u})\, V = \sum_k \nabla \cdot (\mathbf{u} V^k) \tag{9}$$

since the velocity field is not a function of the kth fluid. This gives us an expression for the divergence of velocity that may not be the same as the naive result. Furthermore, we can use this expression with (2.5) to assure that the discretization is volume filling. A similar approach can be used on the evolution equation for f^k.

It should be noted that this development is quite similar to that found in [1] where a similar result is found for the computation of the divergence of velocity.

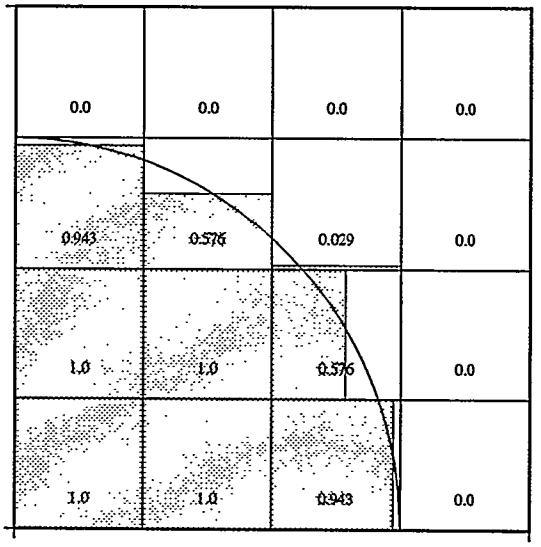

Figure 1: Reconstruction of a circular arc based upon the SLIC VOF approximation of piecewise constant interfaces in each cell. Numbers shown are cell volume fractions.

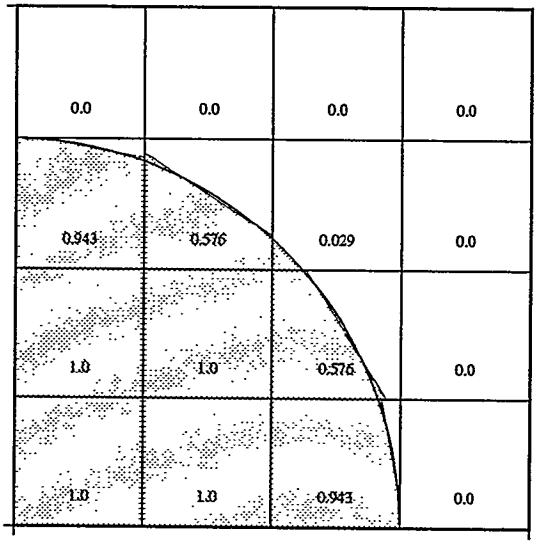

Figure 2: Reconstruction of a circular arc based upon the PLIC VOF approximation of piecewise linear interfaces in each cell. Numbers shown are cell volume fractions.

3. Implementation of Multi-Dimensional

PLIC VOF Interface Tracking Methods

In this paper we will describe the implementation of PLIC methods for incompressible flows. This setting provides a strong competing constraint on the design of the PLIC method. In addition to the volume filling rule, the total volume of a material is conserved over the entire mesh. In using the volume filling rule, discrete error can be absorbed by the compressibility of the fluid. In the setting of incompressible flow, this is an often unacceptable error.

The key point of this development is that the discrete equation (2.8) will not be zero. Another point of distinction is between operator split methods and those that are not. For operator split methods on incompressible flows, one can utilize that fact that

$$\frac{\partial u}{\partial x} = -\frac{\partial v}{\partial y} \tag{10}$$

For non-operator split or unsplit methods we will use $\nabla \cdot \mathbf{u} = 0$ and a redistribution algorithm to assure that the volume of a material is conserved on the mesh. In practice either this approach or the volume-filling derivation leads to pleasing and convergent numerical results.

$$\mathbf{x} \cdot \hat{\mathbf{n}} - \rho = 0 \tag{11}$$

$-\infty < \rho < \infty$

$$\mathbf{x}_0 \cdot \hat{\mathbf{n}} - \rho \begin{cases} < 0, \text{ if } \mathbf{x}_0 \text{ behind plane} \\ = 0, \text{ if } \mathbf{x}_0 \text{ on plane} \\ < 0, \text{ if } \mathbf{x}_0 \text{ in front of plane} \end{cases} \tag{12}$$

3.1 2-D Structured Meshes

Our approach to constructing VOF methods in two dimensions differs greatly from the standard here. In almost every description (and implementation) the reconstruction of the volume in a cell and the construction of the numerical flux of a volume has been done in a "case-by-case" manner. By "case-by-case" we mean that the geometric reconstruction is strictly analyzed for specific cell-interface topologies and results are used with a case-by-case classification to define the method. We will break with this tradition for several reasons: the number of cases proliferates in 3-D or for unsplit integration methods, it is complex and pedagogically difficult and it lends itself to defining a method that is heuristic rather than rigorous.

By contrast the approach we outline below is straightforward and provides a algorithmic framework that is simple and extensible. Furthermore, it allows us to tap into a significant amount of work done in the area of computational geometry. The VOF method and its PLIC variant in particular can be broken up into a set of geometric primitives that lead to a concise statement of the method.

Let us examine on of these geometric primitives: how a line truncates a volume. Given a cell (or a flux volume) defined by a set of vertices $\mathbf{x}_v$, and a line defined by (12), we are to determine which portion of polygon (cell) lies inside the line. Clearly, the line divides 2-space into two portions depending on our definition the normal $\hat{\mathbf{n}}$ to the line.

Algorithm 1 [Polygon Truncated by a Line]

1. *Determine which of the vertices, $\mathbf{x}_v$ lie inside the fluid. Here we use the direction formed by a point on the line and a vertex to determine its identity vis-a-vis the line.*
2. *Traverse the polygon and collect the vertices that are inside the fluid and the intersection of the line with segments that connect vertices that are and are not inside the fluid.*

Another key part of the algorithm is the computation of areas bounded by a polygon. Axisymmetric calculations also fall easily into this framework.

We compute the lines location in a polygon via Brent's method as it is a robust method to do this.

Other essential portions of the algorithm deal with visualization of the reconstruction and solution. Useful utility routines such as the length of an interface in a cell (another good visualization technique!).

Algorithm 2 [PLIC Reconstruction]

1. *Compute the normals to the line in the mixed cell*
2. *Make an initial guess of the constant for the line*
3. *Find the constant that gives the correct volume in a cell (Brent's method).*

Algorithm 3 [Mixed Cell Detection]

1. *Flag cells as mixed if $\varepsilon \le f \le 1 - \varepsilon$*
2. *Flag cells as mixed if a cell is full ($f > 1 - \varepsilon$) and one of its face neighbors is empty.*

Algorithm 4 [Active Cell Detection]

1. *Flag a cell as being active if one of the cells in its domain of dependence is mixed.*

Algorithm 5 [Isolated Cell Detection]

1. *Flag a cell as being isolated if it is mixed and none of its neighbors is mixed.*

Algorithm 6 [Operator-Split 2-D PLIC Method]

1. *Flag all mixed, active and isolated cells.*
2. *Compute the discrete divergence of velocity in the mixed, active and isolated cells.*
3. *Reconstruct the interface in all mixed cells.*
4. *Compute the volume fluxes δV_k in the mixed, active and isolated cells.*

5. Advance the volumes in time

$$\tilde{f}_{ij}^{k} = f_{ij}^{k,n} - \frac{\delta V_{i+1/2j}^{k,n} - \delta V_{i-1/2j}^{k,n}}{V_{ij}} + \delta t \nabla \cdot \mathbf{u}_{ij} f_{ij}^{k,n} \quad (13)$$

$$f_{ij}^{k,n+1} = \tilde{f}_{ij}^{k} - \frac{\delta \tilde{V}_{ij+1/2}^{k} - \delta \tilde{V}_{ij-1/2}^{k}}{V_{ij}} + \delta t \nabla \cdot \mathbf{u}_{ij} f_{ij}^{k,n+1} \quad (14)$$

$$\tilde{f}_{ij}^{k} \tilde{V}_{ij} = f_{ij}^{k,n} V_{ij}^{L} - \left(\delta V_{i+1/2j}^{k,n} - \delta V_{i-1/2j}^{k,n} \right) \quad (15)$$

$$f_{ij}^{k,n+1} V_{ij} = \tilde{f}_{ij}^{k} \tilde{V}_{ij} - \left(\delta \tilde{V}_{ij+1/2}^{k} - \delta \tilde{V}_{ij-1/2}^{k} \right) \quad (16)$$

$$V_{ij}^{L} = V_{ij}(1 + \delta t \nabla \cdot \mathbf{u}_{ij}) \quad (17)$$

$$\tilde{V}_{ij} = V_{ij}^{L} - V_{ij} \delta t \frac{\partial u}{\partial x} = V_{ij}\left(1 + \delta t \frac{\partial v}{\partial y} \right) \quad (18)$$

$$\frac{f_{ij}^{k,n+1} - f_{ij}^{k,n}}{\delta t} + \frac{\partial}{\partial x}(f_{ij}^{k,n} u) + \frac{\partial}{\partial y}(\tilde{f}_{ij}^{k} v) = \nabla \cdot \mathbf{u}_{ij} f_{ij}^{k,n} \quad (19)$$

$$\frac{f_{ij}^{k,n+1} - f_{ij}^{k,n}}{\delta t} + \frac{\partial}{\partial x}(f_{ij}^{k,n} u) + \frac{\partial}{\partial y}(f_{ij}^{k,n} v) = \nabla \cdot \mathbf{u}_{ij} f_{ij}^{k,n} \quad (20)$$

$$\varepsilon_{ij} = \delta t \frac{\partial}{\partial y}\left[\frac{u_{ij} v_{ij}}{1 + \delta t \partial v_{ij}/\partial y} \frac{\partial f_{ij}^{n}}{\partial x} \right] \quad (21)$$

Algorithm 7 [Non-Operator-Split 2-D PLIC method]

1. Flag all mixed, active and isolated cells.
2. Compute the discrete divergence of velocity in the mixed, active and isolated cells.
3. Reconstruct the interface in all mixed cells.
4. Compute the volume fluxes in the mixed, active and isolated cells, δV^{k}.
5. Advance the volumes in time using

$$f_{ij}^{k,n+1} = f_{ij}^{k,n} - \frac{\delta V_{i+1/2j}^{k,n} - \delta V_{i-1/2j}^{k,n}}{V_{ij}} - \frac{\delta V_{ij+1/2}^{k,n} - \delta V_{ij-1/2}^{k,n}}{V_{ij}} + \frac{\delta t}{2}(f_{ij}^{k,n} + f_{ij}^{k,n+1}) \nabla \cdot \mathbf{u}_{ij} \quad (22)$$

6. Apply a bounds check or redistribute any $f_{ij}^{k,n+1} > 0$ or $f_{ij}^{k,n+1} < 1$.

Algorithm 8 [Volume Redistribution]

1. Look within the domain of dependence for a cell, if there is an over or undershoot, distribute the nonphysical quantity to mixed cells in the domain of dependence.

We should take a few moments to acknowledge some of the characteristics of the method from the standpoint of numerical analysis, its stability, consistency and accuracy. As constructed each of our integration techniques is a consistent discretization of the volume evolution equation. Because the methods have been derived to be positive schemes, they are stable under the following Courant condition:

$$\delta t = \mathrm{MAX}_{ij}\left(\frac{\delta x}{|u_{ij}|}, \frac{\delta y}{|v_{ij}|} \right) \quad (23)$$

From the Lax Equivalence theorem, the method is therefore convergent (although second-order is elusive).

VOF methods are often thought to not have dissipation, and in the direction normal to flow this is the case, however for flow tangential to the interface the method is diffusive in a manner similar to upwind methods. Similarly, the dispersion is present tangentially, and in the normal direction as well. In the normal direction analysis is difficult, and the results depend strongly on the local topology of a cell and the interface that is being tracked.

Other important properties of a good algorithm include the symmetry of the solution, dealing with more than two fluids and compressible flow. Also, of interest are its properties when the solution is underresolved and grossly underresolved (i.e. how does the method fail? does it degrade gracefully?).

3.2 3-D Structured Meshes

3.3 3-D Unstructured Meshes

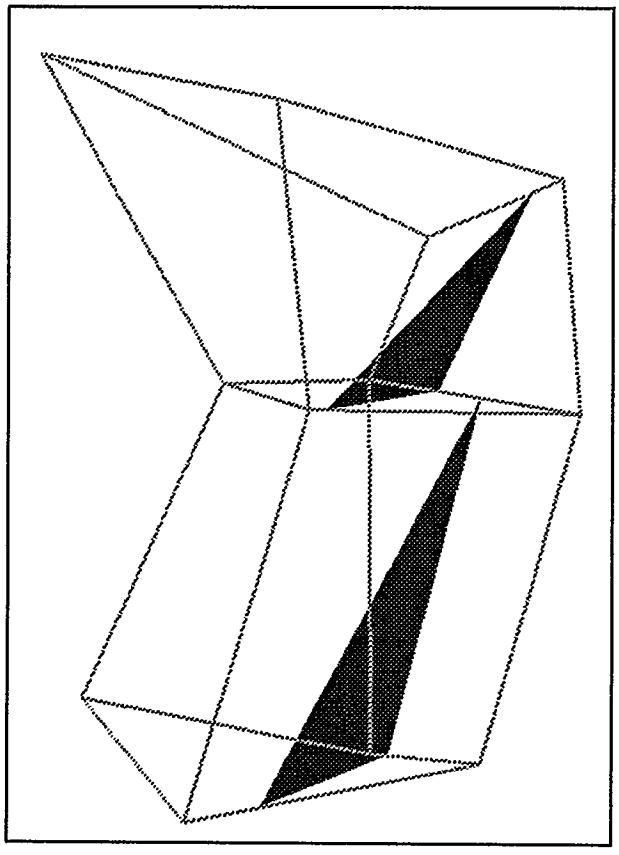

Figure 3:

4. Estimation of Interface Normals From Fluid Volume Data

$$\nabla f = \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \tag{24}$$

$$A \cdot \nabla f^{LS} = b \tag{25}$$

$$A = \begin{bmatrix} \delta x_i^2 d_i^{-1} & \delta x_i \delta y_i d_i^{-1} & \delta x_i \delta z_i d_i^{-1} \\ \delta x_i \delta y_i d_i^{-1} & \delta y_i^2 d_i^{-1} & \delta y_i \delta z_i d_i^{-1} \\ \delta x_i \delta z_i d_i^{-1} & \delta y_i \delta z_i d_i^{-1} & \delta z_i^2 d_i^{-1} \end{bmatrix} \tag{26}$$

$$b = \begin{bmatrix} \delta x_i \delta f_i d_i^{-1} \\ \delta y_i \delta f_i d_i^{-1} \\ \delta z_i \delta f_i d_i^{-1} \end{bmatrix} \tag{27}$$

$$\delta x_i = x_n - x_0;\ \delta y_i = y_n - y_0;\ \delta z_i = z_n - z_0;\ \delta f_i = f_n - f_0 \tag{28}$$

$$d_i = \sqrt{\delta x_i^2 + \delta y_i^2 + \delta z_i^2} \tag{29}$$

$$f_x^{VA} = \frac{1}{V}\int_0^1\int_0^1\int_0^1 \begin{vmatrix} f_\xi & y_\xi & z_\xi \\ f_\eta & y_\eta & z_\eta \\ f_\zeta & y_\zeta & z_\zeta \end{vmatrix} d\xi d\eta d\zeta \tag{30}$$

$$\nabla f^{GG} = \frac{1}{V}\sum_f f_f A_f \tag{31}$$

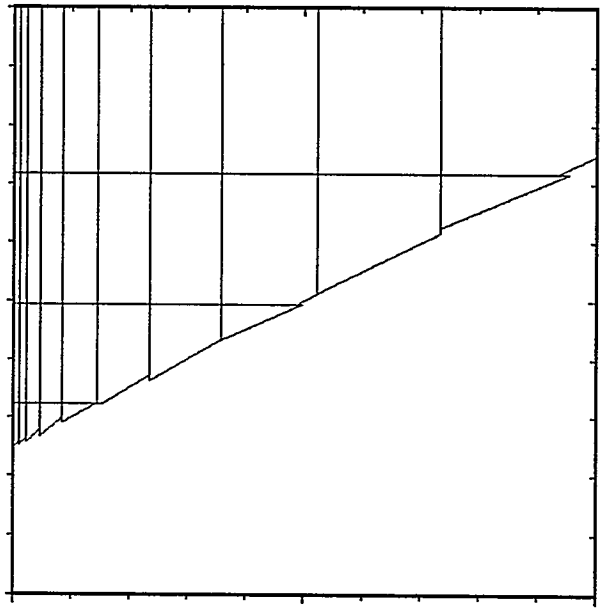

Figure 4: PLIC VOF reconstruction of a line on a Cartesian tensor-product mesh using Youngs' expression for the interface normal [45].

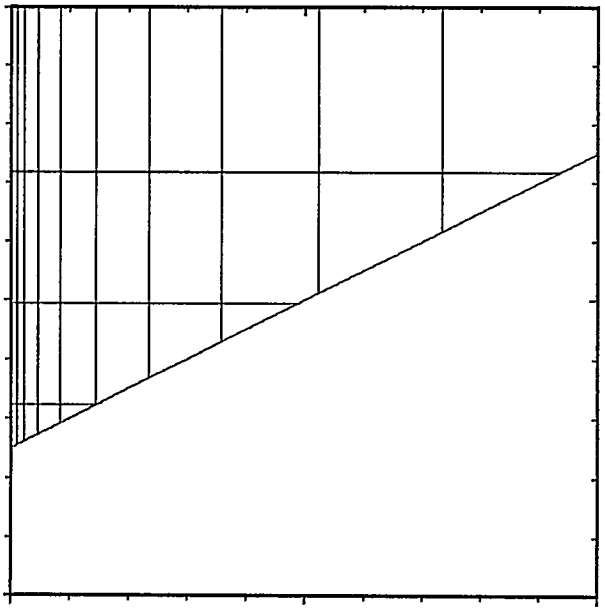

Figure 5: PLIC VOF reconstruction of a line on a Cartesian tensor-product mesh using the fast least squares method for the interface normal

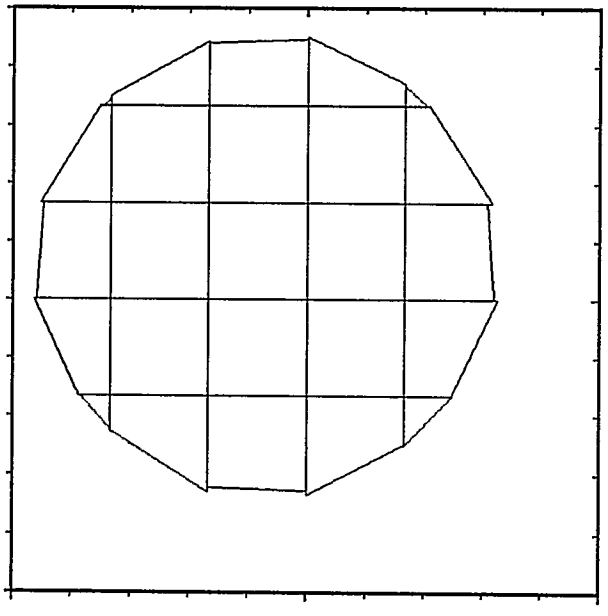

Figure 6: PLIC VOF reconstruction of a circle on a Cartesian tensor-product mesh using Youngs' method for the interface normal [45].

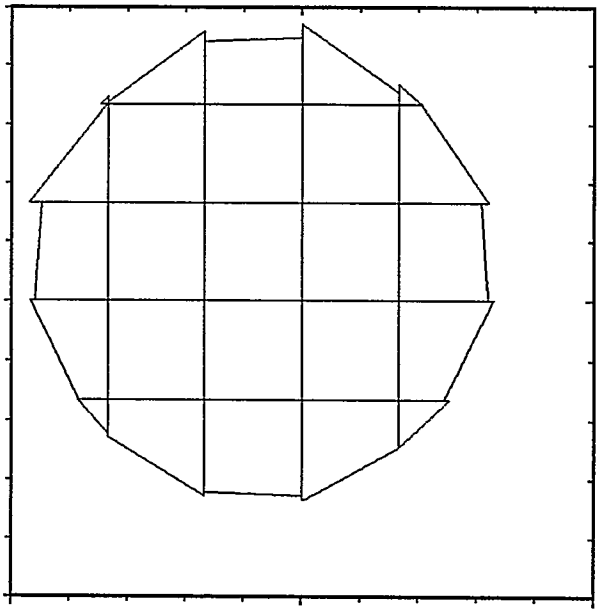

Figure 7: PLIC VOF reconstruction of a line on a Cartesian tensor-product mesh using the fast least squares method for the interface normal

4.1 PLIC VOF Methods: Outstanding Issues

5. Recent Developments in the CSF Model for Surface Tension

5.1 Background

The basic premise of the CSF methodology is to model physical processes specific to and localized at fluid interfaces (e.g, surface tension, phase change) by applying the process to fluid elements everywhere within the interface transition regions. Surface processes are thereby replaced with volume processes whose integral effect properly reproduces the desired interface physics. The CSF method has proven successful in a variety of studies [2-18]. The CSF method lifts all topological restrictions (typically inherent in models for surface tension) without sacrificing accuracy, robustness, or reliability. It has been extensively verified and tested in two-dimensional flows through its implementation in a classical algorithm for free surface flows, where complex interface phenomena such as breakup and coalescence have been predicted.

Algorithms for interface dynamics must model physics specific to and localized at interfaces, such as phase change and surface tension. In this section we describe

our method for modeling interfacial surface tension, which is formulated with a localized volume force as prescribed by the recent CSF model [3]. Although originally developed for surface tension, the basic approach of the CSF model lends itself quite well to interfacial physics in general, i.e., surface phenomena other than surface tension can be encapsulated easily within the CSF model. Typical examples are phase change and momentum exchange [10], where the surface physics are mass and momentum flux, respectively, transferred across the interface.

The central theme of the CSF model is formulation of interface dynamics into a localized volumetric force, which is quite different from earlier numerical models of interfacial phenomena. The basic premise of the CSF model is to replace interfacial surface phenomena (normally applied via a discrete boundary condition) as smoothly varying volumetric forces derived from a product of the appropriate interfacial physics per unit area and an approximation to the surface (interface) delta function. The CSF formulation makes use of the fact that numerical models of discontinuities in finite volume and finite difference schemes are really continuous transitions within which the fluid properties vary smoothly from one fluid to another (over a distance of $\vartheta(h)$ where h is a length comparable to the resolution afforded by the computational mesh). It is not appropriate, therefore, to apply in these schemes a boundary condition at an interface "discontinuity", which in the case of surface tension is a pressure jump across the interface. Surface tension should instead act on fluid elements everywhere within the transition region.

In the case of surface tension, the relevant surface physics is a force per unit area arising from local interface curvature and local (tangential) variations in the surface tension coefficient. Application of interfacial physics using the CSF model then reduces to application of a localized force in the momentum equation, regardless of interface topology. The CSF model is therefore ideally suited for dynamical interfaces of arbitrary topology. Its simplicity, accuracy, and robustness has led to its widespread and popular use [2-18] in modeling complex interfacial flows that were in many cases previously intractable.

Surface tension modeled with the CSF method places no restrictions on the number, complexity, or dynamic evolution of interfaces having surface tension. Direct comparisons made in modeling surface tension with the CSF model and a popular interface reconstruction model [36] show that the CSF model makes more accurate use of volume fraction data [3]. The normal surface tension force tends to drive interface topologies toward a minimum surface energy configuration. Reconstruction models for surface tension, on the other hand, can sometimes induce numerical noise from computed graininess in the surface pressures, often leading to unphysical disruptions at the interface. The CSF model is also easy to implement, as surface tension is modeled simply by calculating and applying an additional volumetric force in the momentum equation. A small fraction of the total CPU time is expended in modeling surface tension effects.

5.2 Overview of the CSF Model

In the CSF model, surface tension is reformulated as a volumetric force $\mathbf{F}_s$ given by

$$\mathbf{F}_s = \mathbf{f}_s \delta_s , \tag{32}$$

where δ_s is the surface delta function and $\mathbf{f}_s$ is the surface tension force per unit interfacial area [3]:

$$\mathbf{f}_s = \sigma \kappa \hat{\mathbf{n}} + \nabla_s \sigma . \tag{33}$$

In equation (25) above, σ is the surface tension coefficient, ∇_s is the surface gradient [3], $\hat{\mathbf{n}}$ is the interface unit normal, and κ is the mean interfacial curvature, given by [30]

$$\kappa = -\nabla \cdot \hat{\mathbf{n}} . \tag{34}$$

The first term in (25) is a force acting normal to the interface, proportional to the curvature κ. The second term is a force acting tangential to the interface toward regions of higher surface tension coefficient (σ). The normal force tends to smooth and propagate regions of high curvature, whereas the tangential force tends to force fluid along the interface toward regions of higher σ.

Since interfaces having surface tension are tracked with the volume- and particle-based methods mentioned in the previous sections, their topology will not in general align with logical mesh coordinates. Discontinuous interfaces are therefore represented in the computational domain as finite thickness transition regions within which fluid volume fractions vary smoothly from zero to one over a distance of $\vartheta(h)$. The surface delta function, nonzero only within these transition regions, was proposed in the original CSF model to be [3]

$$\delta_s = \frac{|\nabla c|}{[c]} = |\nabla f| = |\mathbf{n}| , \tag{35}$$

where c is the characteristic (color) function uniquely identifying each fluid in the problem and $[c]$ is the jump in the color function across the interface in question. The fluid volume fractions f serve as the color function in this work, so $[c] = 1$.

We have found, as have others [19], that an optimal form for δ_s is one of the key outstanding issues for the

CSF model. An optimal form for δ_s is loosely defined to be one that displays desired convergence and smoothness properties. Our current numerical results, for example, display an undesirable sensitivity to the form used for δ_s. It is therefore crucial that the form of δ_s remain arbitrary, which is not apparent in recent published modifications of the CSF model in which the volume force in equation (25) is approximated as the divergence of a surface stress tensor [13] unnecessarily constrains the CSF model by restricting the form of δ_s to equation (27). This restriction could ultimately limit the usefulness of these continuum stress models unless they accept arbitrary forms for δ_s. The ability of the CSF model to accurately model surface tension effects relies upon the accuracy and smoothness of δ_s, and this issue is discussed in more detail in the following section.

Although the original formulation of the CSF model was motivated primarily by the need to model the normal force in equation (25), there are no restrictions in the underlying theory from including the tangential force as well. This enables the model to properly take into account local spatial variations in σ, which typically arise because of temperature variations and/or the presence of surfactants. We are currently modeling both the normal and tangential forces in equation (25). An example of using the CSF method to model temperature-induced variable surface tension effects can be found in [12].

$$\nabla_s = (\mathbf{i} - n_x \mathbf{n})\,\partial_x + (\mathbf{j} - n_y \mathbf{n})\,\partial_y + (\mathbf{k} - n_z \mathbf{n})\,\partial_z \qquad (36)$$

Given equations (32)-(36), the continuum surface force is easily estimated from first and second order spatial derivatives of the fluid volume fractions. The interface normal vector $\mathbf{n}$ is first computed at cell faces, $\hat{\mathbf{n}}$ follows from normalization, and κ follows from equation (34). The force, which resides at cell centers in our scheme, will be nonzero only within the interface transition region. It is normalized to recover the conventional description of surface tension as the local product $\kappa h \to 0$. Its line integral directed normally through the interface transition region is approximately equal to the pressure jump $\sigma\kappa$.

As stated in reference [3], a wide stencil in general leads to a better estimate of curvature. However, in contrast to the discretizations presented in [3], we have chosen a conservative discretization for the curvature κ and the force $\mathbf{F}_s$. This is motivated by the need to preserve an important physical property of surface tension, namely that the net surface tension force (and also κ) should vanish over any closed surface. A conservative discretization of κ is therefore used, given by:

$$\kappa = -\frac{1}{V}\sum_f \hat{\mathbf{n}}_f \cdot \mathbf{A}_f \qquad (37)$$

where V is the control (cell) volume, $\mathbf{A}_f$ the area of face f (pointing outward) on V, and $\hat{\mathbf{n}}_f$ the unit interface normal on face f of V. For the unit interface normal, a six-cell stencil (in 2-D) is used to compute both components of $\hat{\mathbf{n}}_f$ at each face. The cell-centered curvature κ then results by summing over cell faces, bringing the effective stencil to nine cells. A nonconservative discretization might possibly induce artificial horizontal motion of bubbles that should otherwise rise vertically under the action of buoyancy forces.

5.3 CSF Model: Outstanding Issues

1. *Optimal model for the surface delta function.*
2. *Optimal discretization method for the interface unit normal and curvature.*
3. *Optimal method (if any) used to smooth and/or weight the continuum surface force.*
4. *Optimal method (if any) used to smooth the color function.*
5. *Efficient model for an advanced-time (implicit) curvature.*
6. *Triple points*

As suggested in [3], smoother variations in κ generally result if a mollified volume fraction $\tilde{f}$ is used to compute the face normals in equation (37). A variety of smoothing algorithms (e.g., B-spline or point-Jacobi) have been found to give the desired results, which is the mitigation of high wavenumber contributions to κ (resulting possibly from discretization errors). This should, however, be used with caution because the actual interface geometry could be mollified unphysically. Nevertheless, use of a mollified f in estimating κ was the preferred choice in recent numerical studies detailed in [6,7,18]. Although some examples of the effects of smoothing can be found in [3], it still warrants further investigation, especially as it relates to convergence and consistency. The representation of surface tension in the CSF model as an explicit force is linearly stable only for time steps smaller than a maximum allowable value time step δt_s necessary to resolve the propagation of capillary waves [3]. This constraint is often restrictive, especially when fluid interfaces are undergoing topology changes (e.g., pinch-off) since δt_s is roughly proportional to a higher power ($\sim 3/2$ of the mesh spacing. This restriction can be alleviated with an implicit treatment of κ.

6. Numerical Examples

We now present numerical examples to illustrate the properties of the PLIC VOF interface tracking method

and the CSF model for surface tension. For the PLIC VOF method, we choose simple advection tests: 3-D translation of a cube and sphere and 3-D rotation of a notched brick. The tests are simple in that the flow field moving the bodies does not have vorticity or shear, which induce topological changes in the body (stretching, tearing) that truly challenge interface tracking methods. Such tests have been devised in 2-D where they were used to scrutinize various tracking methods [33]. Similar 3-D tests are currently being devised, and will be used to test the PLIC VOF method more completely in the near future. For the CSF surface tension model, we again choose a seemingly simple test, namely a static drop test in which the surface tension-induced pressure rise inside a 2-D cylindrical or 3-D spherical drop is computed and compared with the known analytic solution. The static drop test, presented in [3] as a validation test of the original CSF model, has since become recognized as one of the most difficult tests for surface tension models based upon the immersed interface methodology. This is because of the difficulty in maintaining an equilibrium position for the drop over many computational cycles. The computed pressure field inside the drop tends to have small numerical variations that induce artificial flow that were recently dubbed "parasitic currents" [13]. The magnitude of these parasitic currents tends to grow with each time step.

6.1 Interface Tracking Tests

Multi-dimensional test problems devised specifically to scrutinize algorithms designed to track discontinuities have not yet become a standard part of the literature. Interface tracking test problems are for the most part lacking, as opposed to those used to test high order continuum advection schemes, where a consistent set of challenging two-dimensional problems are commonly used to judge the relative worth of the methods. The absence of interface tracking test problems has led to our recent study in which we devised a new set of very tough test problems useful for scrutinizing two-dimensional tracking capabilities [33]. We plan to devise equally difficult three-dimensional test problems in future work. Although we are not necessarily advocating these test problems to be the standard barometer by which all tracking methods must be judged, others have begun to use them to test the relative merit of their methods [35]. The intent, however, is to devise tests that thoroughly interrogate an interface tracking method by exposing algorithm weaknesses and well as strengths.

This *numerical surface tension*, or smoothing of high curvature regions, is a characteristic feature of the PLIC VOF method as a consequence of the piecewise linear interface geometry approximation coupled with volume conservation. Smoothing will result if the piecewise linear approximation is not adequate, i.e., when the interface has large sub-cell curvature. Once smoothed, this high curvature information is not recoverable. The degree of smoothing is dictated by the dimensionless product κh, where κ is the local interface curvature and h is the mesh spacing. When κh is of order one (or greater), only one cell (or less) per radius of curvature resolves the interface. Numerical experiments indicate that at least 3-5 cells per radius of curvature, or $\kappa h \leq 1/3$, are needed for faithful representation (minimal smoothing) of the interface geometry. This same constraint applies for the reliable modeling of surface tension [3]. Two possible approaches for overcoming this smoothing problem are (1) a higher-order approximation to the interface geometry (allowing sub-cell curvature), and (2) adaptive mesh refinement (AMR) [42] of high curvature regions. The use of AMR for improved resolution of interface geometries has in fact already been demonstrated in [31] and looks promising. An AMR scheme coupled with the PLIC method might be a more efficient alternative to high resolution interface tracking rather than a higher-order reconstruction.

One final comment before presenting our results. It is *extremely important* to keep in mind that the actual piecewise planar *interfaces* are displayed in the figures that follow, not the *one-half volume fraction isosurface*, as is often shown in other works using VOF interface tracking methods. This practice can be very misleading for several reasons. First, none of the published VOF methods (to the knowledge of the authors) actually use a volume fraction isosurface value for the reconstruction and movement of an interface, instead using a piecewise constant (stairstepped or not) or piecewise linear reconstruction, so associating a volume fraction isosurface with the interface is *incorrect*. Second, volume fraction isosurface plots often give the appearance of mass creation or destruction, as bits of fluid fall above and/or below the specified contour level. Third, volume fraction isosurface plots often infer a much smoother interface than the actual interface reconstructed and used by the interfacial flow algorithms. For these reasons, we will always show actual interfaces (lines in 2-D and planes in 3-D) rather than volume fraction isosurfaces.

6.1.1 3-D Translation (Structured Mesh).

A cube (unit length) or sphere (unit diameter) is placed in the lower corner of either a 16^3 or 32^3 mesh, translated diagonally to the upper corner, and returned to its original position. The total translation distance is two (sphere or cube) diameters (D), one diameter translation to the opposite corner and one diameter translation back. The computational domain is a box spanning

$0.0 \leq (x, y, z) \leq 4.0$, with the cube and sphere being initially centered at $(x, y, z) = 0.875$. Resolution of the cube and sphere is fairly coarse, with only four or eight cells spanning the cube length (L) and sphere diameter. Volume fractions are initialized to unity and zero inside and outside the bodies, respectively. Those cells initially containing an interface have volume fractions initialized according to the portion of the cell truncated by the interface. Error measurements are performed on the differences in volume fraction data observed between the initial and final times. We use an L_1 norm:

$$L_1 = \sum |f^i - f^f| \,/\, \sum |f^i| \,, \tag{38}$$

and an L_2 norm:

$$L_2 = \sqrt{\sum (f^i - f^f)^2 \,/\, \sum (f^i)^2} \,, \tag{39}$$

where f^i and f^f are the initial and final volume fractions, respectively.

It is evident by the initial cube interfaces with L/h equal to 4, shown in Figure (8), that the cube is underresolved with only four cells spanning its width. The piecewise planar approximation of the cube edges is inadequate at this resolution, as seen by the interface planes "chopping off" the cube corners and edges at a 45 degree angle in Figure (8). It is not surprising, therefore, that the cube on this mesh will actually tend toward a sphere after translation, as shown in Figure (9). When

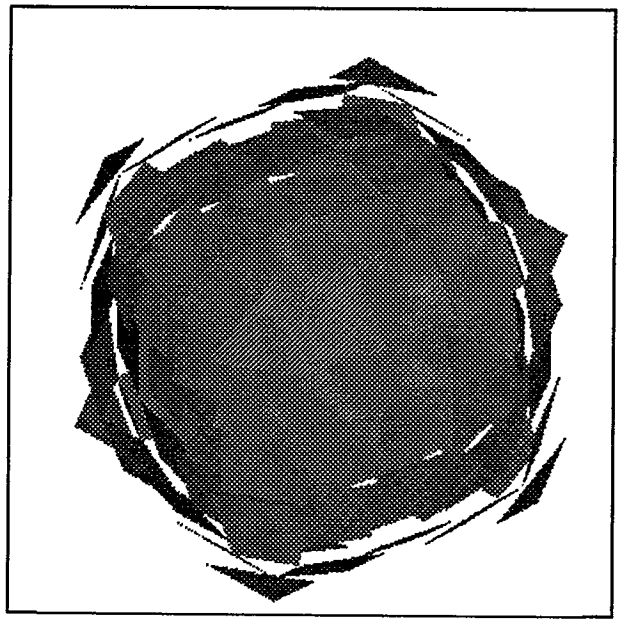

Figure 8: Initial cube interfaces (L/h = 4).

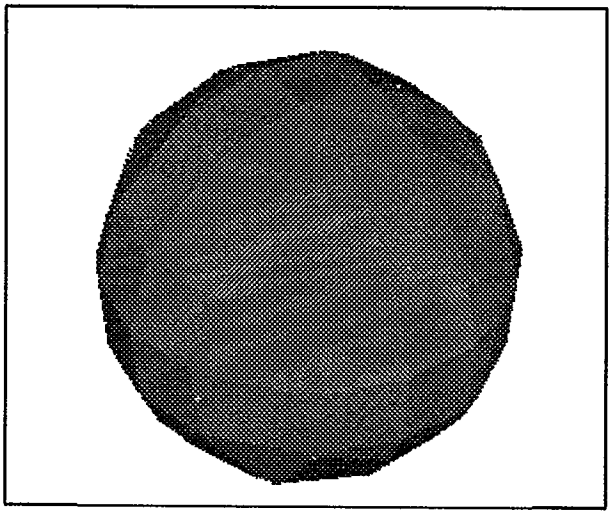

Figure 9: Cube interfaces after being translated diagonally (and returned) a distance of one cube diagonal (L/h = 4).

L/h is equal to 8, however, the cube is adequately resolved, as shown in Figure (10). Its appearance after translation at this resolution in Figure (11) is qualita-

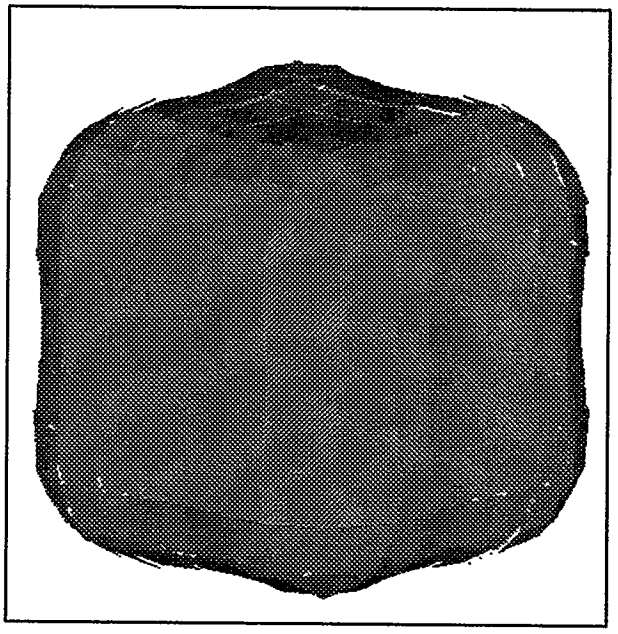

Figure 10: Initial cube interfaces (L/h = 8).

tively the same as its initial shape. The sphere (not

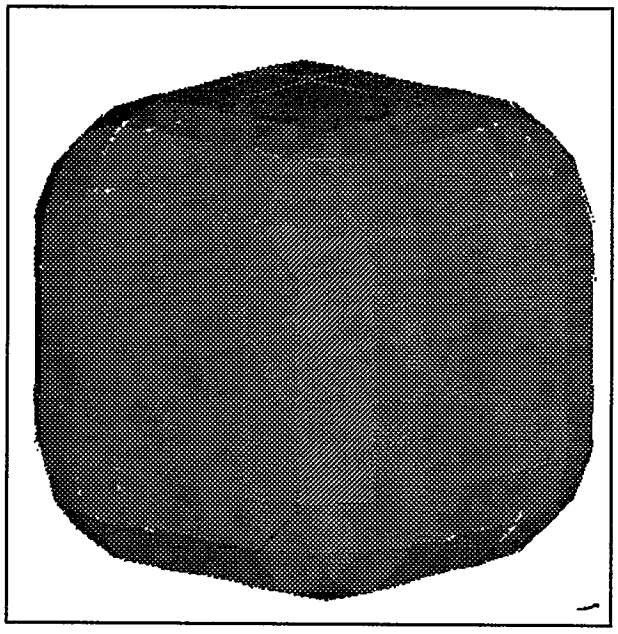

Figure 11: Cube interfaces after being translated diagonally (and returned) a distance of one cube diagonal (L/h = 8).

shown) is translated more accurately than the cube because its curvature, even with D/h of 4, is adequately resolved by the PLIC VOF method, and it does not have any singular edges or corners.

Table (1) shows the computed error norms for the translated cube and sphere at two different CFL numbers. As is expected, a CFL number of one yields essentially zero error because the PLIC VOF method is based upon characteristic upwinding. This has been confirmed in our recent 2-D study as well [33]. We have included a smaller CFL number of 0.375 because this is more typical of a conservative value used in an application simulation. Although more simulations would be required to infer a convergence rate, the error norms in Table ((1)) indicate a convergence rate of (slightly less than) first order. This is again consistent with the results in our 2-D study [33]. An improved interface normal, such as the least squares method of Puckett and coworkers [23], should improve the results to second order. Nevertheless, we have found the convergence rate to be problem-dependent, depending upon the CFL number and the body geometry.

Table 1: Volume fraction error norms for a cube and sphere translated diagonally through the mesh and returned. The translation distance is one sphere diameter (D=1) or one cube diagonal (D=$\sqrt{3}$). Results are shown for two different mesh sizes (h) and CFL numbers.

		Cube		Sphere	
h	CFL	L1	L2	L1	L2
0.25	0.375	0.267	0.099	0.041	0.006
0.25	1.0	0.0001	0.0	0.00001	0.0
0.125	0.375	0.174	0.070	0.023	0.002

6.1.2 3-D Rotation (Structured Mesh)*.

Here we reproduce the rotating notched brick problem presented recently by Puckett and Saltzman in [31]. In this problem, a notched brick is rotated with a rotation vector ω directed diagonally through the brick (from corner to corner): $\omega=(i, j, k)\omega_0$, where ω_0 is a constant. One face of the block is notched by the extraction of four smaller cubes from each corner. This problem is challenging for tracking methods because the flow field represents solid body rotation instead of mere translation, the rotation is oriented diagonally to the mesh instead of being mesh-aligned where discretization errors are minimized, and the body undergoing rotation has high curvature regions (the block edges) that are difficult to resolve.

Figure (12) shows the initial notched brick for a

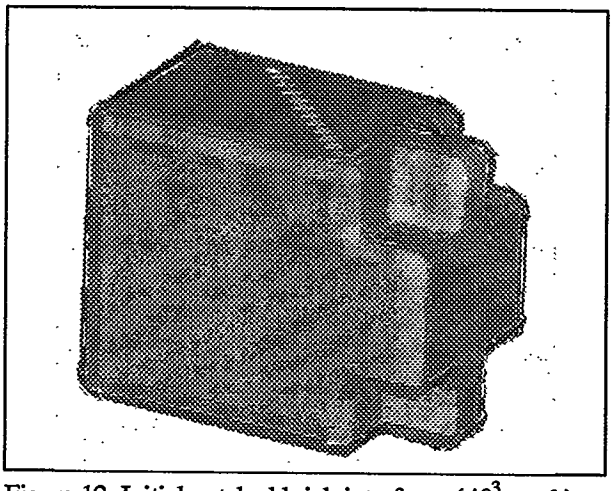

Figure 12: Initial notched brick interfaces (40^3 mesh).

40×40×40 mesh and Figure (13) shows the brick (from a different view) on a mesh that is double the res-

*Animations of the simulations presented in this sections can be found at http://gnarly.lanl.gov/Telluride/Text/movies.html.

olution in every direction ($80 \times 80 \times 80$), which was the

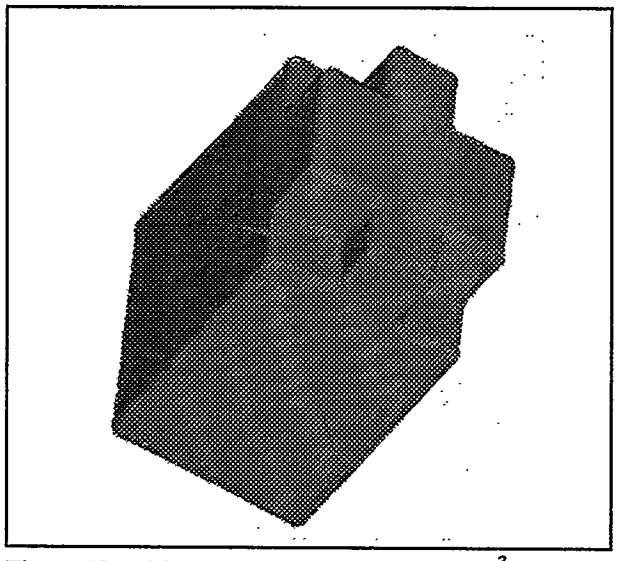

Figure 13: Initial notched brick interfaces (80^3 mesh).

mesh used in the results of [31]. The brick is partitioned with a $25 \times 20 \times 15$ and $50 \times 40 \times 30$ volume of cells, on the coarse and fine mesh, respectively. Two different mesh resolutions are presented to show the consequences of having very high subcell curvature regions (the brick edges) embedded within meshes of various sizes. The brick is rotated one period every 400 time steps, using a CFL number of approximately 0.55 based on velocities at the extreme brick corners [41].

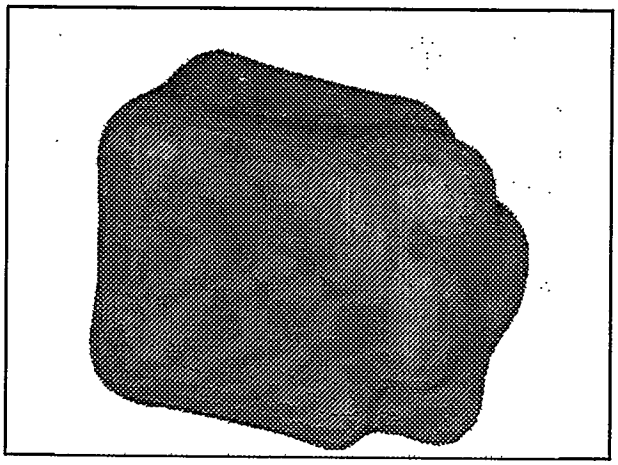

Figure 14: Notched brick interfaces after one revolution (40^3 mesh).

The notched brick on the coarse mesh is shown in Figure (14) after being rotated one revolution. With only a few cells resolving the notches at this resolution, it is obvious that the piecewise planar approximation characteristic of the PLIC VOF method rounds off the high curvature edges until the interface reaches a lower curvature that is resolvable. Because of strict adherence to volume conservation constraints, edge material that is rounded is redistributed locally, leading to the slight bulges along the faces near each edge (the same effect can be seen in the translation problems presented previously). Subsequent revolutions of the notched brick at this resolution cause essentially no additional change in its topology. All of the numerical surface tension effects (rounding of high curvature edges) take place during the first revolution, after which the curvature is resolvable. By *resolvable*, we mean that the piecewise planar approximation is a reasonably accurate approximation of the interface topology.

Figure (15) shows the finer mesh results of the

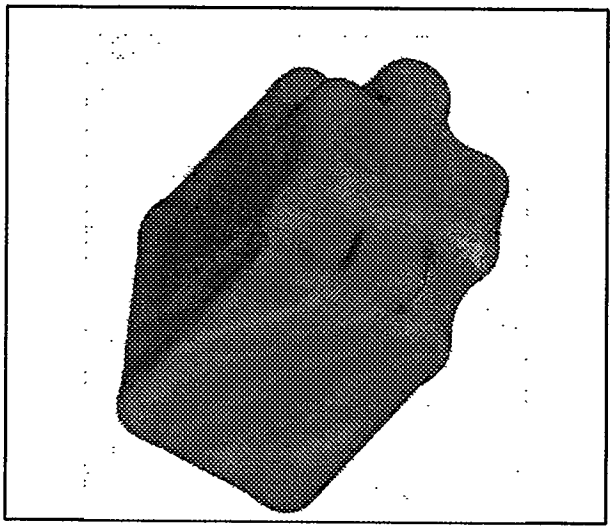

Figure 15: Notched brick interfaces after one revolution (80^3 mesh).

notched brick after one revolution. Here it is evident that edge rounding has been greatly reduced, and the overall notched brick topology is preserved very well. The qualitative difference in results on this finer mesh raises many interesting issues, such as whether or not the amount of numerical surface tension can be quantified and predicted for a given mesh size. It would also be desirable to understand the extent to which these effects will be felt in the proximity of high curvature regions.

The SLIC VOF method generates unacceptable results for this problem, yielding a notched brick that is not even recognizable after only one revolution (see Figure 3d of reference [31]). This is not surprising, as a piecewise constant interface approximation will incur large errors for interfaces oriented arbitrarily to the mesh (as in this rotation problem).

6.1.3 3-D Translation (Unstructured Mesh).

6.2 Static Drop

We now consider a static inviscid spherical drop with surface tension acting along its interface. Gravity is absent, so the only forces acting on the drop are surface tension, which will induce a pressure jump Δp across the drop given by

$$\Delta p = \sigma\kappa, \tag{40}$$

where κ is the drop curvature, equal to $1/R$ in 2-D and $2/R$ in 3-D, and σ is the surface tension coefficient. Assuming the background pressure is zero, the theoretical surface tension-induced drop pressure is

$$p_{theory} = \begin{cases} \sigma/R & \text{in 2-D} \\ 2\sigma/R & \text{in 3-D} \end{cases} \tag{41}$$

where R is the drop radius. As in [3], we can define an average computed drop pressure:

$$p_{drop} = \frac{1}{N}\sum_{n=1}^{N} p_n \quad , \tag{42}$$

where N is the number of cells inside the drop (those having density greater than 99% of the drop density) and p_n is the pressure in cell n. We can also measure the error in the computation with an L_2 norm given by [3]:

$$L_2 = \sqrt{\sum (p_n - p_{theory})^2 / N p_{theory}^2} \quad . \tag{43}$$

For the 2-D drop, the flow field is integrated forward in time many computational cycles, after which the pressure field is examined and compared with its initial value. Ideally the pressure field should exactly cancel the surface tension forces, resulting in a vanishingly small velocity field. At the very least, the pressure and velocity fields should reach steady state, i.e., any computed dynamics are unphysical numerical artifacts.

6.2.1 2-D Drop.

Consider a unit square computational domain partitioned with 64×64 uniform orthogonal cells. A circular (cylindrical) drop of radius R=0.25 is centered at $(x, y) = 0.5$. Surface tension forces are computed with the CSF model via equations (24) and (25), except that a surface delta function in cell n is given by

$$\delta_{s,n} = L_n / A_n \quad , \tag{44}$$

where L_n is the length of the PLIC VOF reconstructed interface in cell n, and A_n is the area in cell n. We have found the surface delta function given by equation (35) to gives better results for the static drop than equation (27). The magnitude of the artificial dynamics is greater when the form for δ_s is given by (27) rather than (35) above. The volume fraction field was smoothed prior to the calculation of curvature using one pass of a 9-point Laplacian with a one-half weighting. The resultant continuum surface force was smoothed in the same way.

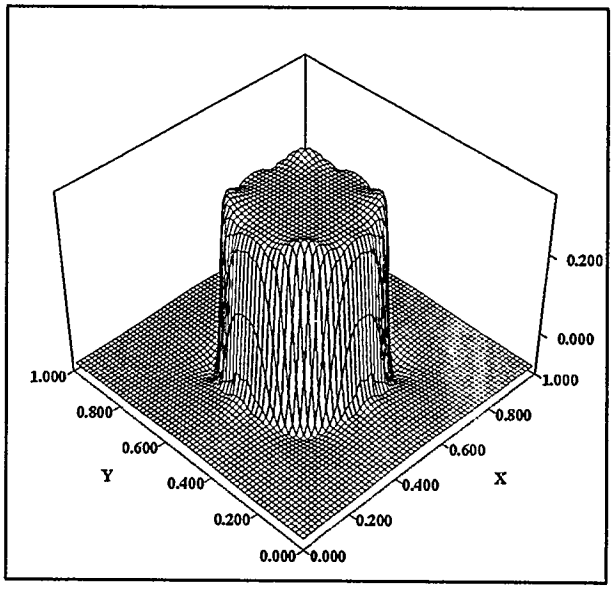

Figure 16: Surface plot of the initial pressure field in a 2-D drop. The drop/background density ratio is 1.

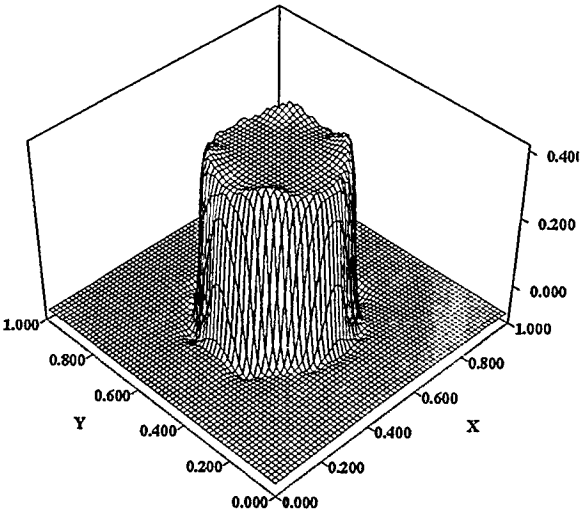

Figure 17: Surface plot of the pressure field in the 2-D drop of Figure (16) after 97 time steps. The CFL number is 0.5.

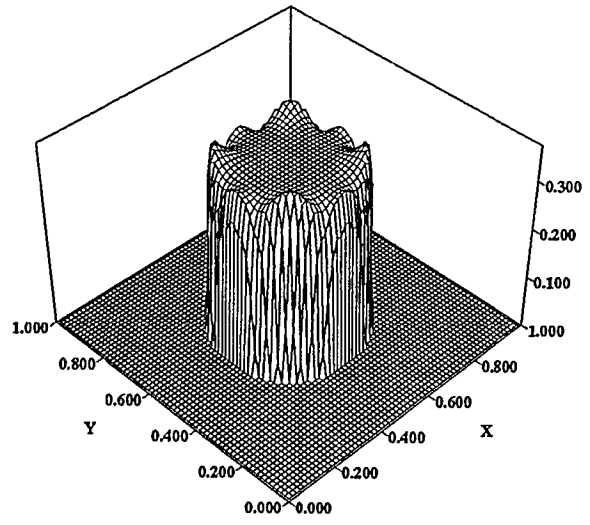

Figure 18: Surface plot of the initial pressure field in a 2-D drop. The drop/background density ratio is 1000.

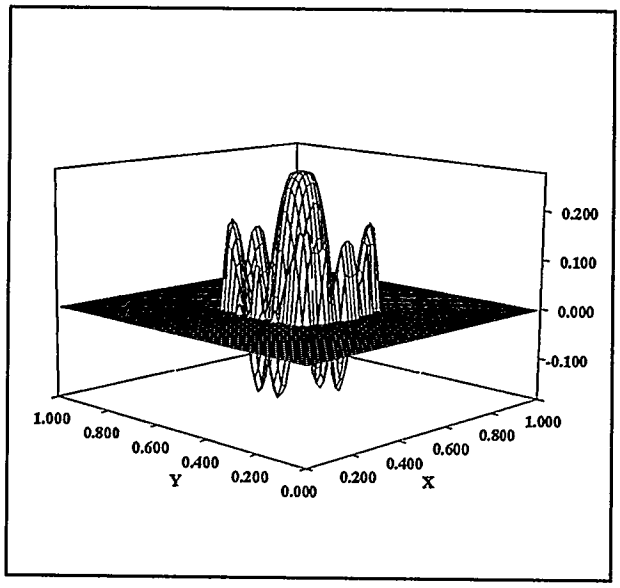

Figure 19: Surface plot of the pressure field in the 2-D drop of Figure (18) after 383 time steps. The CFL number is 0.5.

6.2.2 3-D.

Table 2: Computed average drop pressures for different mesh spacing and drop/background density ratios. The continuum surface force *is not* density scaled.

R/h	ρ_{drop}/ρ_{back}	$\bar{p}_{drop}/p_{theory}$	L_2
4	2	0.70	0.309
8	2	0.71	0.305
4	10	0.84	0.178
8	10	0.82	0.196
4	1000	1.32	0.326
8	1000	1.88	0.913

Table 3: Computed average drop pressures for different mesh spacing and drop/background density ratios. The continuum surface force *is* density scaled according to equation ().

R/h	ρ_{drop}/ρ_{back}	$\bar{p}_{drop}/p_{theory}$	L_2
4	2	0.75	0.274
8	2	0.75	0.264
4	10	0.82	0.223
8	10	0.83	0.204
4	1000	0.85	0.212
8	1000	0.86	0.187

7. Summary and Conclusions

Accurate modeling of interfacial flows, such as the liquid metal filling of molds in casting processes, must have high-fidelity, robust algorithms for interface kinematics (tracking) and dynamics (e.g., surface tension). Our latest algorithmic developments for interfacial flows have been described, in particular volume-based methods for interface tracking and the CSF model for interfacial surface tension. An overall interfacial flow model based on the PLIC VOF interface tracking method and the CSF model for surface tension will be topologically robust, comparatively accurate, and rea-

sonably efficient. This combination of interface algorithms will continue to be a highly competitive and viable approach for the foreseeable future. It is important, however, that PLIC VOF and CSF algorithmic improvements and enhancements targeting the outstanding issues addressed in this paper (and summarized below) continue to evolve in a way that allows these methods to be applied more readily to 3-D interfacial flows having high density ratios, surface tension dominated driving forces, and subcell interface curvature.

We have reviewed VOF interface tracking methods and derived appropriate evolution equations. We have focused on the PLIC VOF method, currently the most modern and accurate member of the VOF family of methods. A detailed 2-D and 3-D PLIC VOF algorithm template has been provided. The entire algorithm can be constructed from a well-defined set of geometric primitives such as line/plane location, line/plane truncation, etc. From the template it is evident that this algorithm is a true interface tracking algorithm, with fluxes derived geometrically rather than algebraically. An implementation based on the templates provided will be robust, efficient, modular, and logically simple. Implementations based on subjective logic (e.g., "look left", "look right") will be less efficient, less general, and more complex.

We have extended the PLIC VOF method to arbitrary 3-D hexahedral (hexes), and have defined the principal geometric task to be the calculation of truncation volumes bounded by doubly-ruled surfaces and an interface plane. Exact analytic solutions have been found for these truncation volumes [], resulting in an efficient implementation. Our solutions should also be valid on non-hex meshes, such as those made up of tetrahedra, pyramids, or prisms, providing these types of cells are viewed as logically degenerate hexes. This will be the subject of future work.

We have identified and begun to address key outstanding issues for PLIC VOF methods. These include an improved estimation of the interface normal (minimizing some norm), accurate (second-order) time integration of the volume fluxes, improved implementation efficiency, better multiple fluid interface (≥ 3) models, and minimization of numerical surface tension. We have found, as has Puckett and coworkers [19,23,32], that an optimal interface normal appears to result from minimization of some norm, which may not be L_2 , but some other norm (e.g., L_∞) that weights the data in a different manner. A fully unsplit time integration scheme gives the most accurate and symmetric results in 2-D, but a carefully devised split scheme remains competitive. A split scheme, however, will break symmetry in certain situations and does not generalize to 3-D unstructured meshes. A 3-D unsplit scheme is therefore highly desirable, but remains a formidable task on unstructured meshes. We have currently pursued simpler approaches, such as generalized splitting and Runge Kutta schemes, but fully unsplit 3-D schemes will be the subject of future work. Finally the numerical surface tension exhibited by PLIC VOF methods has been demonstrated in 3-D, and our standard metric still applies: the product of curvature and mesh spacing must be less than approximately 0.5 for numerical surface tension effects to be minimal. An optimal interface normal calculation will alleviate the effects somewhat, but the solution to resolving any subcell curvature continues to be a higher order VOF method and/or finer mesh resolution.

We have reviewed the CSF model for surface tension, presented a 3-D implementation, and identified outstanding issues. Consistent with Puckett and coworkers [19], we have found that a crucial issue is the surface delta function approximation. Our original choice, the magnitude of the interface normal, may not exhibit the desired convergence and behavior on finer meshes. The manner and amount to which the color function is smoothed prior to computing curvature is also an important issue. Too little smoothing allows noise in the curvature field, while too much allows high frequency capillary waves to persist unphysically. Smoothing also appears to be problem dependent. The continuum surface force itself might also be smoothed and/or scaled, as presented in the original CSF formulation. The pressure field dependence on scaling of the continuum surface force is more sensitive for higher density ratio flows. The extent to which these issues are important in the accurate modeling of the interfacial flows encountered in casting processes will be studied further. Algorithmic improvements resulting from these studies will enhance, improve, and extend the popular CSF model for surface tension.

References

[1] P. Colella, H.M. Glaz, and R.E. Ferguson. Multifluid Algorithms for Eulerian Finite Difference Methods. Unpublished manuscript (1995).

[2] W.J. Rider, D.B. Kothe, S.J. Mosso, J.H. Cerutti, and J.I. Hochstein. Accurate Solution Algorithms for Incompressible Multiphase Fluid Flows. Technical Report AIAA 95-0699, AIAA (1995). Presented at the 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV (1995). See also http://www.c3.lanl.gov/~wjr/pubs.html.

[3] J.U. Brackbill, D.B. Kothe, and C. Zemach. A Continuum Method for Modeling Surface Tension. *J. Comput. Phys.*, 100:335-354 (1992).

[4] D.B. Kothe, R.C. Mjolsness, and M.D. Torrey. RIPPLE: A Computer Program for Incompressible Flows with Free Surfaces. Technical Report LA-12007-MS, Los Alamos

National Laboratory (1991).

[5] D.B. Kothe and R.C. Mjolsness. RIPPLE: A New Model for Incompressible Flows with Free Surfaces. *AIAA J.*, 30:2694-2700 (1992).

[6] J.R. Richards, A.N. Beris, and A.M. Lenhoff. Steady Laminar Flow of Liquid-Liquid Jets at High Reynolds Numbers. *Phys. Fluids A*, 5:1703-1717 (1993).

[7] J.R. Richards, A.M. Lenhoff, and A.N. Beris. Dynamic Breakup of Liquid-Liquid Jets. *Phys. Fluids A*, 6:2640-2655 (1994).

[8] H. Haj-Hariri, Q. Shi, and A. Borhan. Effect of Local Property Smearing on Global Variables: Implication for Numerical Simulations of Multiphase Flows. *Phys. Fluids A*, 6:2555-2557 (1994).

[9] H. Liu, E.J. Lavernia, and R.H. Rangel. Numerical Investigation of Micropore Formation During Substrate Impact of Molten Droplets in Plasma Spray Processes. *Atomization and Sprays*, 4:369-384 (1994).

[10] N.B. Morley, A.A. Gaizer, and M.A. Abdou. Estimates of the Effect of a Plasma Momentum Flux on the Free Surface of a Thin Film of Liquid Metal. In ISFNT-3: Third Interational Symposium in Fusion Nuclear Technology, UCLA, Los Angeles, CA (1994).

[11] Q. Shi and H. Haj-Hariri. The Effects of Convection on the Motion of Deformable Drops. Technical Report AIAA 94-0834, AIAA (1994). Presented at the 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV (1994).

[12] G.P. Sasmal and J.I. Hochstein. Marangoni Convection with a Curved and Deforming Free Surface in a Cavity. *J. Fluids Engr.*, 116:577-582 (1994).

[13] B. LaFaurie, C. Nardone, R. Scardovelli, and S. Zaleski. Modelling Merging and Fragmentation in Multiphase Flows with SURFER. *J. Comput. Phys.*, 113:134-147 (1994).

[14] H. Liu, E.J. Lavernia, and R.H. Rangel. Numerical Simulation of Substrate Impact and Freezing of Droplets in Plasma Spray Processes. *J. Phys. D: Appl. Phys.*, 26:1900-1908 (1993).

[15] H. Liu, E.J. Lavernia, and R.H. Rangel. Modeling of Molten Droplet Impingement on a Non-Flat Surface. *Acta Metall. Mater.*, 43:2053-2072 (1995).

[16] J.-P. Delplanque, E.J. Lavernia, and R.H. Rangel. Multi-Directional Solidification Model for the Description of Micro-Pore Formation in Spray Deposition Processes. Presented at the 1995 ASME WAM, San Francisco, CA (1995).

[17] M. Schmid and F. Klein. Fluid Flow in Die Cavities - Experimental and Numerical Simulation. Proceedings of the North American Die Casting Association Conference, Indianapolis, IN, pp. 93-99 (1995).

[18] J.R. Richards, et al. Drop Formation in Liquid-Liquid Systems Before and After Jetting. *Phys. Fluids*, 7:2617-2630 (1995).

[19] I. Aleinov and E.G. Puckett. Computing Surface Tension with High-Order Kernels. Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, pp. 13-18 (1995).

[20] R.J. LeVeque and Z. Li. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM J. Numer. Anal.*, 31:1019-1044 (1994).

[21] H.O. Nordmark. Rezoning for Higher Order Vortex Methods. *J. Comput. Phys.*, 97:366-397 (1991).

[22] C.S. Peskin. Numerical Analysis of Blood Flow in the Heart. *J. Comput. Phys.*, 25:220-252 (1977).

[23] J.E. Pilliod and E.G. Puckett. Second Order Volume-of-Fluid Interface Tracking Algorithms. Unpublished manuscript, to be submitted to *J. Comput. Phys.* (1996).

[24] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, and W.J. Rider. A Second-Order Projection Method for Tracking Fluid Interfaces in Variable Density Incompressible Flows. Submitted *to J. Comput. Phys.* (1995).

[25] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flows. *J. Comput. Phys.*, 114:146-159 (1994).

[26] Y.C. Chang, T.Y. Hou, B. Merriman, and S. Osher. A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows. Unpublished manuscript (1995).

[27] S.O. Unverdi and G. Tryggvason. A Front Tracking Method for Viscous, Incompressible Multi-Fluid Flows. *J. Comput. Phys.*, 100:25-37 (1992).

[28] T.Y. Hou, J.S. Lowengrub, and M.J. Shelley. Removing the Stiffness From Interfacial Flows with Surface Tension. *J. Comput. Phys.*, 114:312-338 (1994).

[29] T.L. Williams. An Implicit Surface Tension Model. Masters Thesis, Department of Mechanical Engineering, Memphis State University (1993).

[30] C.E. Weatherburn. On Differential Invariants in Geometry of Surfaces, with Some Applications to Mathematical Physics. *Quarterly J. Math.*, 50:230-269 (1927).

[31] E.G. Puckett and J.S. Saltzman. A 3D Adaptive Mesh Refinement Algorithm for Multimaterial Gas Dynamics. *Physica D*, 60:84-93 (1992).

[32] E.G. Puckett. A Volume-of-Fluid Interface Tracking Algorithm with Applications to Computing Shock Wave Rarefaction. Proceedings of the 4th Interational Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, pp. 933-938 (1991).

[33] W.J. Rider and D.B. Kothe. Stretching and Tearing Interface Tracking Methods. Technical Report AIAA 95-1717, AIAA (1995). Presented at the 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV (1995). See also http://www.c3.lanl.gov/~wjr/pubs.html.

[34] D.B. Kothe and W.J. Rider. Comments on Modeling Interfacial Flows with Volume-of-Fluid Methods. Technical Report LA-UR-94-3384, Los Alamos National Laboratory (1994).

[35] M. Rudman. Volume Tracking Methods for Interfacial Flow Calculations. Unpublished manscript, Commonwealth Scientific and Industrial Research Organization (1995).

[36] R.S. Hotchkiss. Simulation of Tank Draining Phenomena with NASA SOLA-VOF. Technical Report LA-8163-MS, Los Alamos National Laboratory (1979).

[37] D.B. Kothe, et al. Computer Simulation of Metal Casting Processes: A New Approach. Technical Report LALP-95-197, Los Alamos National Laboratory (1995).

[38] W.J. Rider and D.B. Kothe. A Marker Particle Method for Interface Tracking. Technical Report LA-UR-95-1740, Los Alamos National Laboratory (1995). Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, pp. 976-981 (1995).

[39] D.B. Kothe and W.J. Rider. Practical Considerations for Interface Tracking Methods. Technical Report LA-UR-95-1741, Los Alamos National Laboratory (1995). Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, NV, pp. 638-643 (1995).

[40] D.B. Kothe et al. PAGOSA: A Massively-Parallel, Multi-Material Hydrodynamics Model for Three-Dimensional High-Speed Flow and High-Rate Deformation. Technical Report LA-UR-92-4306, Los Alamos National Laboratory (1992). Proceedings of the 1993 SCS Simulation Multiconference: High Performance Computing, March (1993).

[41] J. Saltzman. Personal Communication, Los Alamos National Laboratory (1994).

[42] M.J. Berger and P. Colella. Local Adaptive Mesh Refinement for Shock Hydrodynamics. *J. Comput. Phys.*, 82:64-84 (1989).

[43] T.J. Barth. Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes. Technical Report AIAA 93-0668, AIAA (1993). Presented at the 31st Aerospace Sciences Meeting and Exhibit, Reno, NV (1993).

[44] W.E. Pracht and J.U. Brackbill. BAAL: A Code for Calculating Three-Dimensional Fluid Flows at All Speeds with an Eulerian-Lagrangian Computing Mesh. Technical Report LA-6342, Los Alamos National Laboratory (1976).

[45] B.J. Parker and D.L. Youngs. Two and Three Dimensional Eulerian Simulation of Fluid Flow with Material Interfaces. Technical Report AWE 01/92, Atomic Weapons Establishment (1992). Presented at the Third Zababakhin Scientific Talks, Kyshtim, Russia (1992).