Proceedings of FEDSM'02
2002 ASME Fluids Engineering Division Summer Meeting
July 14-18, 2002, Montreal, PQ, Canada

FEDSM2002-31125

MODELING HIGH DENSITY RATIO INCOMPRESSIBLE INTERFACIAL FLOWS

Markus Bussmann
Douglas B. Kothe[*]
James M. Sicilian

CCS-2, Methods for Advanced Scientific Simulations
Mail Stop B-250, Los Alamos National Laboratory
Los Alamos, NM 87545
Email: bussmann@lanl.gov, dbk@lanl.gov, sicilian@lanl.gov

ABSTRACT

We present an approach to modeling incompressible interfacial flows on fixed meshes that yields solutions at any density ratio. There are two aspects of the methodology that are crucial for obtaining accurate high density ratio solutions: a consistent approach to mass and momentum conservation, by using mass flux information from an interface advection algorithm as the basis for the momentum advection calculation, and a careful evaluation of pressure gradients near the interface. Our particular implementation couples a volume tracking algorithm with a predictor/projection solution of the flow equations on unstructured meshes. We present the methodology, and then the results of several calculations.

INTRODUCTION

A key characteristic of incompressible interfacial flows is the density ratio between fluids. For liquid/gas flows a typical value is 10^3-10^4, as is the case for water or molten metal flows in an atmospheric air environment. The ambient gas does not always play a significant role in the dynamics of the heavy fluid, and the influence diminishes as the density ratio increases. But there are nonetheless phenomena where the gas phase will affect the liquid even at very high density ratios. Examples include gas entrainment in a liquid phase, bubble dynamics, and aerodynamic effects of a gas phase on a liquid.

There are today an impressive variety of techniques used to model interfacial flows (Scardovelli and Zaleski, 1999; Tryggvason et al., 2001; Osher and Fedkiw, 2001; Hou et al., 2001), and yet a review of the literature leads one to conclude that many implementations of various algorithms encounter difficulties as the density ratio increases. Some will suggest that an algorithm has been designed to yield solutions for a particular range of density ratios, and so dismiss the fact that the algorithm fails to model other regimes. We would argue, on the other hand, that a high density ratio capability is a desirable characteristic of any algorithm, and that the ability (or not) of a model to yield solutions for such flows tells us much about the algorithm itself.

We focus our attention in this paper on Eulerian (fixed grid) schemes using an immersed interface method to solve for arbitrarily-complex interface topologies. There are two aspects of such a model that we have found to be required for modeling high density ratio flows. The first is a tighter coupling of the interface advection algorithm to the flow solver. We impose a consistency between mass (interface advection) and momentum (the flow solver) conservation, by using the mass fluxes calculated by the interface advection algorithm as the basis for the momentum advection calculation performed by the flow solver. This is in contrast to most approaches of using an interface advection algorithm strictly to move an interface with a given velocity field, and then solving for velocities independent of any mass flux information that the interface advection algorithm might provide (Scardovelli and Zaleski, 1999; Tryggvason et al., 2001; Osher

[*] Address all correspondence to this author.

and Fedkiw, 2001). The idea of consistency between mass and momentum conservation isn't new; (Rudman, 1998) used such an approach in a staggered grid scheme on a cartesian mesh. But because the mass and momentum control volumes are staggered, Rudman was forced to introduce a subgrid mesh (twice as fine in each of the coordinate directions) on which to solve for mass conservation, in order to obtain mass fluxes across the faces of the momentum control volumes. We present in this paper a collocated scheme (all fluid variables reside at cell centers) that alleviates this restriction, and hence makes it easier to enforce consistency between mass and momentum conservation.

The second aspect that is crucial is the careful evaluation of pressure gradients used to enforce solenoidality. Consider for a moment a simple hydrostatic problem, of two fluids of very different densities one atop the other. The resulting pressure distribution may be continuous, but ∇P at the interface is not, and any calculation of ∇P must recognize and respect the discontinuity. The difficulty is not usually apparent in the heavy fluid, but rather in the light, which can be dramatically affected by erroneous local estimates of ∇P.

In what follows, we present a brief overview of the equations and of our particular algorithm, followed by the results of several simulations.

FORMULATION

We begin with equations of conservation of mass and momentum. To simplify the presentation, we choose to neglect both viscous and surface tension effects, although we have performed calculations that include such phenomena:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{1}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla p + \rho \mathbf{g} \tag{2}$$

$\mathbf{u}$ represents velocity, ρ density, p pressure, and $\mathbf{g}$ the gravitational acceleration.

Although individual fluids are incompressible, we consider multiple immiscible fluids (of different densities) within a domain, and so retain ρ within the bracketed terms on the LHS of equation 2. However, as the density of any fluid particle remains constant,

$$\frac{D\rho}{Dt} = 0 \tag{3}$$

and so:

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

Equation 2, then, is an Eulerian expression of conservation of fluid momentum, subject to the incompressibility constraint embodied by equation 4. Equation 1 describes the transport of density, or put more simply, the transport of different fluids within the domain.

We discretize equation 2 to first order in time, and by introducing an interim "predicted" velocity $\mathbf{u}^{\star}$, divide the resulting equation in two:

$$\frac{\rho^{n+1}\mathbf{u}^{\star} - \rho^{n}\mathbf{u}^{n}}{\Delta t} = -\nabla \cdot (\rho \mathbf{u}\mathbf{u})^{n} \tag{5}$$

$$\frac{\rho^{n+1}\mathbf{u}^{n+1} - \rho^{n+1}\mathbf{u}^{\star}}{\Delta t} = -\nabla P^{n+1} + \rho^{n+1}\mathbf{g} \tag{6}$$

Equation 5 is an explicit formula for $\mathbf{u}^{\star}$, and includes all forces except gravity and pressure gradients. Combining equations 5 and 6 exactly reproduces the time discretization of equation 2; no additional approximation results from this decomposition, which is made simply for computational convenience. Equation 6 relates $\mathbf{u}^{n+1}$ to $\mathbf{u}^{\star}$; combining equation 6 with equation 4 yields:

$$\nabla \cdot \frac{\nabla P^{n+1}}{\rho^{n+1}} = \nabla \cdot (\frac{\mathbf{u}^{\star}}{\Delta t} + \mathbf{g}) \tag{7}$$

We solve equation 7 for P^{n+1}, and complete the timestep by evaluating $\mathbf{u}^{n+1}$ via equation 6.

We now present details of the algorithm used to solve equations 1 and 5-7. Our implementation is on an unstructured hexahedral mesh, with the primary variables $\mathbf{u}$ and p located at cell centers. To assess solenoidality, we also calculate the velocity field at cell face centroids, designated $\mathbf{u}_f$.

Given values of $\mathbf{u}^n$, $\mathbf{u}^n_f$, and p^n, we advance the solution to time $n+1$ in the following manner:

1. Solve equation 1 for ρ^{n+1} using $\mathbf{u}^n_f$. We begin by defining a volume fraction f_k as the fraction of each cell volume V occupied by fluid k:

$$f_k = V_k / V \tag{8}$$

A cell density is then related to the volume fractions via:

$$\rho = \sum f_k \rho_k \tag{9}$$

Equation 1 may then be written:

$$\frac{\partial (f_k \rho_k)}{\partial t} + \nabla \cdot (f_k \rho_k \mathbf{u}) = 0 \tag{10}$$

and since each ρ_k is constant, we obtain an evolution equation for the f_k:

$$\frac{\partial f_k}{\partial t} + \nabla \cdot (f_k \mathbf{u}) = 0 \tag{11}$$

We utilize a multidimensional PLIC (piecewise linear interface calculation (Rider and Kothe, 1998)) volume tracking algorithm for unstructured meshes (Kothe et al., 1999) to solve equation 11 for f_k^{n+1}. The algorithm consists of two steps: a planar reconstruction of fluid-fluid interfaces within a cell, corresponding exactly to the f_k^n and to estimates of the orientations of the interfaces (evaluated as gradients of the f_k^n); and then a geometric calculation of volume fluxes of different materials across cell faces.

2. Evaluate $\nabla \cdot (\rho \mathbf{u}\mathbf{u})^n$ and then $\mathbf{u}^\star$ utilizing the volume flux information of step 1. Consider Figure 1, that illustrates the advection of material across the right face of a cell containing an interface between two fluids. The advected volume is:

$$\delta V_f = \Delta t A_f \mathbf{u}_f \cdot \hat{n}_f \tag{12}$$

where $\mathbf{u}_f \cdot \hat{n}_f$ is the component of the solenoidal face velocity $\mathbf{u}_f$ normal to the face, and A_f is the face area. Discretizing the advection term of equation 2, we obtain:

$$\Delta t \int \nabla \cdot (\rho \mathbf{u}\mathbf{u})^n dV \approx \sum_f \delta V_f \langle \rho \mathbf{u} \rangle_f^n \tag{13}$$

The volume tracking algorithm calculates the subvolumes $\delta V_{k,f}$ within the flux volume δV_f (see Figure 1), and we introduce $f_{k,f}$ to represent the volume fraction of δV_f associated with a particular material k:

$$f_{k,f} = \frac{\delta V_{k,f}}{\delta V_f} \tag{14}$$

Multiplying the $\delta V_{k,f}$ by corresponding densities yields the mass of each material k crossing a face f:

$$M_{k,f} = \rho_k \delta V_{k,f} \tag{15}$$

and equation 13 may then be written in a way that is wholly consistent with mass advection:

$$\Delta t \int \nabla \cdot (\rho \mathbf{u}\mathbf{u})^n dV \approx \sum_f \sum_k M_{k,f} \langle \mathbf{u} \rangle_f^n \tag{16}$$

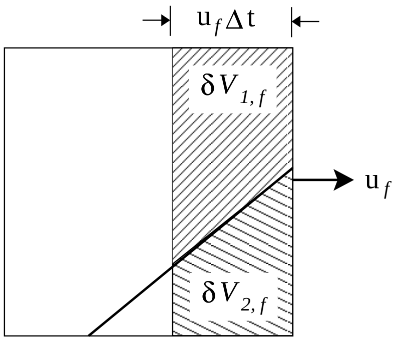

Figure 1. THE ADVECTION OF MATERIAL ACROSS THE FACE OF A CELL CONTAINING AN INTERFACE BETWEEN TWO FLUIDS. THE VOLUME TRACKING ALGORITHM CALCULATES THE INDIVIDUAL FLUX VOLUMES $V_{1,f}$ and $V_{2,f}$.

Note that $\langle \mathbf{u} \rangle_f^n$ is the estimate of the advected momentum per unit mass. For the purposes of this paper, we have chosen this to be a simple upwind value.

3. Interpolate $\mathbf{u}^\star$ and ρ^{n+1} to cell faces to obtain $\mathbf{u}_f^\star$ and ρ_f^{n+1}. Velocity interpolation is done using a least squares linear reconstruction technique similar to that of (Barth, 1993), as we assume that velocity does not vary discontinuously near an interface. The resulting stencil typically encompasses more than just a couple of cells. Density, on the other hand, varies discontinuously, and so we limit the size of the stencil by calculating a face density as a simple average of cell densities on either side of a face.

4. Solve equation 7 for the cell-centered P^{n+1}, where divergences are calculated by summing over cell faces, and ∇P_f is calculated from a stencil corresponding to that of the density interpolation to faces. Equation 6 is then used to calculate a solenoidal (to machine precision) face velocity field:

$$\mathbf{u}_f^{n+1} = \mathbf{u}_f^\star - \Delta t \left(\frac{\nabla P_f^{n+1}}{\rho_f^{n+1}} - \mathbf{g} \right) \tag{17}$$

5. Finally, interpolate $(\nabla P_f^{n+1}/\rho_f^{n+1} - \mathbf{g})$ to cell centers in order to obtain $\mathbf{u}^{n+1}$ from $\mathbf{u}^\star$ via equation 6.

DISCUSSION

The algorithm as described contains two elements that are crucial for the solution of high density ratio flows. The first is

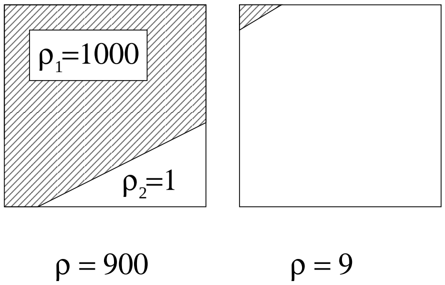

Figure 2. THE CELL DENSITY ρ AS AN INTERFACE PASSES THROUGH. ON THE LEFT, A CELL NEARLY FILLED WITH HEAVY FLUID; ON THE RIGHT, AFTER A SINGLE TIMESTEP, THE CELL NEARLY EMPTY.

the use of the volume flux information provided by the volume tracking algorithm to calculate mass fluxes across faces. It is a simple idea, and yet appears to have been introduced only recently (Rudman, 1998). The reason for this would seem to be related to the fact that most implementations of volume tracking (or VOF) algorithms have been on staggered grids. Until Rudman introduced the subgrid mesh for mass advection, in order to overlap the mass and momentum control volumes, the idea of consistency was apparently not obvious.

The usual alternative to imposing consistency has been to effectively decouple mass from momentum advection, and evaluate $\sum_k M_{k,f}$ for the momentum calculation simply by multiplying some estimate of face density by the fluid volume crossing the face during the timestep. Such an approach is likely fine at low density ratios, when the change of momentum of an interface cell is defined as much by local variations in velocity as it by the difference in densities between fluids. But as the density ratio increases, so does the importance of coupling the mass and momentum calculations. Consider Figure 2, where we depict the state of a cell as an interface passes through. At time n, the cell is nearly filled with a heavy fluid; at the end of the timestep, the reverse is true, and the cell density has changed by two orders or magnitude. If we assume that the velocity distribution in neighboring cells and across the interface is relatively continuous, as will often (but not always - a strong shear flow is one counter-example) be the case, then the accuracy of the momentum calculation is much more a function of the estimates of $\sum_k M_{k,f}$ than of the advected velocities. Any discrepancy between the net sum of mass leaving a cell, $\sum_f \sum_k M_{k,f}$, and the change in cell mass $(\rho^{n+1} - \rho^n)V$, will result in significant errors in the estimate of the interim cell velocity $\mathbf{u}^\star$.

It is worth noting that this discussion of mass and momentum coupling presumes that we solve for momentum conservation via an expression like equation 2. But even this has not always been the case for volume tracking codes. Instead, it has been common practice (e.g. Nichols et al., 1980; Kothe et al., 1991) to solve for conservation of specific momentum, or velocity, by dividing both sides of equation 2 by ρ, on the grounds that the fluids are incompressible. Such an argument would apply in regions removed from fluid interfaces, but certainly not in those cells through which interfaces pass. Such an approach would most surely become problematic as the density ratio increases.

We have implemented this methodology in a volume tracking context, in which volume (and mass) fluxes across cell faces are explicitly calculated. Other interface advection algorithms, such as level sets (Osher and Fedkiw, 2001) or front tracking (Tryggvason et al., 2001), for example, do not calculate volume fluxes in as explicit a manner, and so an implementation of an approach such as the one presented here seems difficult. However, an evaluation of consistency would be of interest; namely, what is the difference between cell density change evaluated via the interface advection algorithm and the net sum of mass exiting a cell, as used to evaluate momentum advection:

$$\rho^{n+1} - \rho^n + \frac{\sum_f \sum_k M_{k,f}}{V} \qquad (18)$$

Although we have chosen to ignore viscous and surface tension effects, they are obviously important to modeling interfacial flows. The usual approach is to evaluate such terms explicitly on the RHS of equation 5. When incorporated into the algorithm presented here, it is crucial that these terms be evaluated using the time $n+1$ density distribution. The reason for this is most easily explained by returning to Figure 2. Momentum advection, now consistent with mass, removes momentum from the cell in proportion to the mass removed. When a cell is emptied, or nearly emptied, of a heavy fluid, there remain very small amounts of mass and momentum relative to what was in the cell at the beginning of the timestep. Momentum fluxes due to viscous, surface tension, and body force terms must then be in proportion to the momentum that is left in the cell. For example, a body force based on the original cell mass results in a completely unrealistic acceleration of the remaining fluid.

Turning to the calculation of P^{n+1}, we readily admit that our approach to evaluating ρ_f and ∇P_f are not appropriate on meshes that stray far from orthogonality. We are in the process of implementing a more sophisticated approach. What we have learned, however, is that there must be a consistency between the stencils. Neither ρ nor ∇P is continuous across a fluid interface, even if the configuration is a simple hydrostatic one. However, if we assume velocity to be relatively continuous, then so is $\nabla P/\rho$. Evaluating ρ and ∇P via different interpolations, for example, leads to difficulties that we no longer encounter when the stencils are the same.

Finally, we have seen no signs of pressure field decoupling, a common ailment of cell-centered schemes. Although the implementation is not explicit, the algorithm contains what is a variation of the (Rhie and Chow, 1983) solution to the decoupling problem. Interim velocities $\mathbf{u}^\star$ that do not yet contain a pressure correction are interpolated to cell faces. Pressure gradient corrections are then calculated at cell faces via small stencils, and the resulting gradients, divided by face densities, are averaged to obtain a cell-centered correction. The result is not prone to checkerboarding.

RESULTS

Figure 3 illustrates a simple 2D test of our algorithm. We consider the motion of a drop (cylinder) in an ambient fluid, for different density ratios from one to 10^9, and illustrate five subsequent interfaces for each case. The drop is assigned an initial non-zero velocity; the initial ambient fluid velocity is set to zero. We consider neither viscosity nor surface tension effects. Of interest is the deformation of the drop due to its interaction with the surrounding fluid: the lower the density ratio, the stronger the deformation. At high ratios, as the influence of the ambient fluid diminishes, the solution eventually approaches that of a pure translation of a drop in a void. Note that even at the highest ratios, the results show a slight deviation from a circle; these, however, are related to splitting errors in the tracking of the interface, and not to the flow solution, as the same deviations appear even when we impose the initial velocity field on the simulation, rather than solve for it.

Figure 4 illustrates four views of a 2D collapsing column simulation, calculated on a 2678 cell triangle mesh. The simulation parameters correspond to the experimental results of (Martin and Moyce, 1952). The density ratio is 800, corresponding to a water/air calculation. The results, however, are little affected by the density ratio: we ran the same calculation at a ratio of 10^6, and the results were indistinguishable from those of Figure 4. Figure 5 is a plot of the non-dimensional front position versus time. The results agree well with the data of Martin and Moyce; the difference is of a similar magnitude to that reported elsewhere (e.g. Maronnier et al., 1999; Lock et al., 1998).

Finally, Figures 6 and 7 present views of two simulations of the filling of a hemisphere: Figure 6 with a fluid 1000 times as heavy as the fluid initially in the domain, Figures 7 with a fluid a million times as heavy. The domain is vented near the top of the sphere, to allow the light fluid to escape as the heavy fluid pours in. The results are obviously qualitative, and unlike the results of Figure 4, show slight differences in behaviour as the density ratio changes, because the relative motion of the heavy and light fluids is now much more complex. Nevertheless, the results are very similar, and another demonstration of the efficacy of this approach.

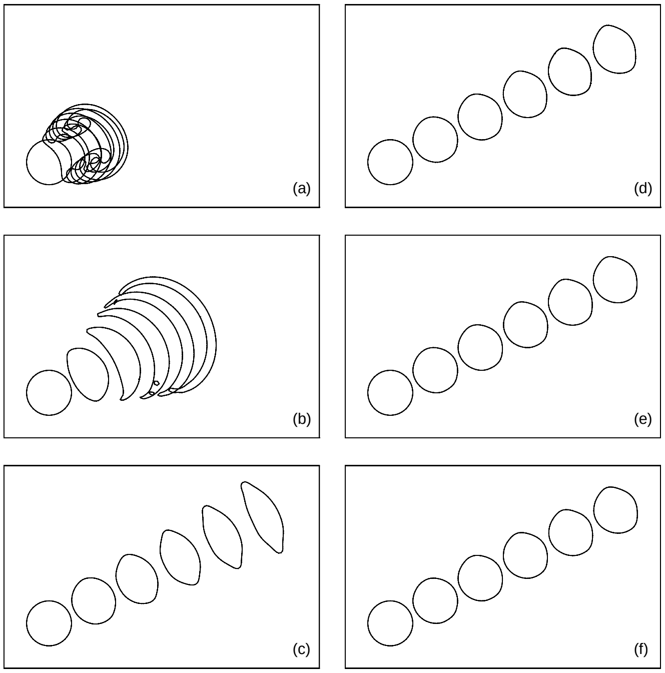

Figure 3. THE MOTION OF A 2D DROP, WITH INITIAL VELOCITY (u,v) = (2,1), IN AN INITIALLY STATIC AMBIENT FLUID, FOR VARIOUS DENSITY RATIOS: (a) 1 (b) 10 (c) 100 (d) 1000 (e) 10^6 (f) 10^9.

CONCLUSIONS

Based on the work of (Rudman, 1998), we have developed a high density ratio incompressible flow model implemented on a collocated mesh. We have identified two aspects of the algorithm that are crucial for the solution of such flows: a consistent approach to mass and momentum advection, and a careful evaluation of pressure gradients near an interface. The algorithm has no difficulty solving for flows of any density ratio.

We intend to develop this methodology further. Our plans include: the development of a more sophisticated interpolation for ∇P and ρ; the extension of this algorithm to flow in the presence of void (zero density) fluid and moving solid boundaries; and a comprehensive verification of the algorithm, especially when we consider viscous and surface tension effects.

REFERENCES

M. Rudman, A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for Numerical Methods in Fluids* 28(2), 357-378 (1998).

R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics* 31, 567-603 (1999).

G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-

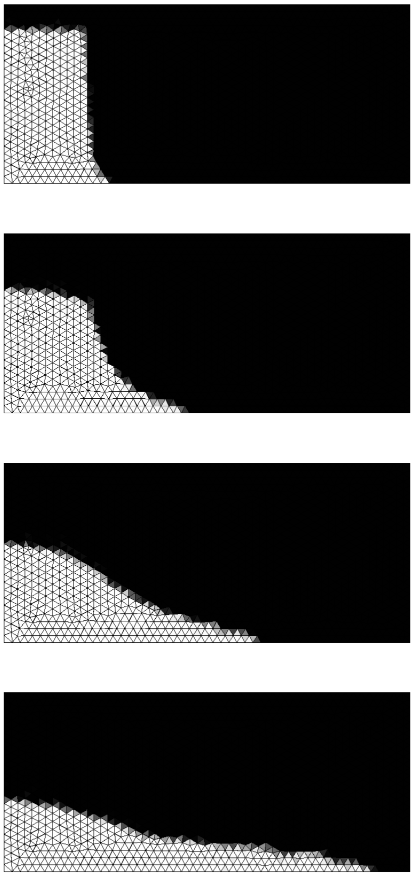

Figure 4. FOUR VIEWS OF THE COLLAPSE OF A WATER COLUMN SURROUNDED BY AIR, CORRESPONDING TO THE EXPERIMENT OF (Martin and Moyce, 1952). A SIMILAR CALCULATION AT A DENSITY RATIO OF 10^6 YIELDED NEARLY IDENTICAL RESULTS.

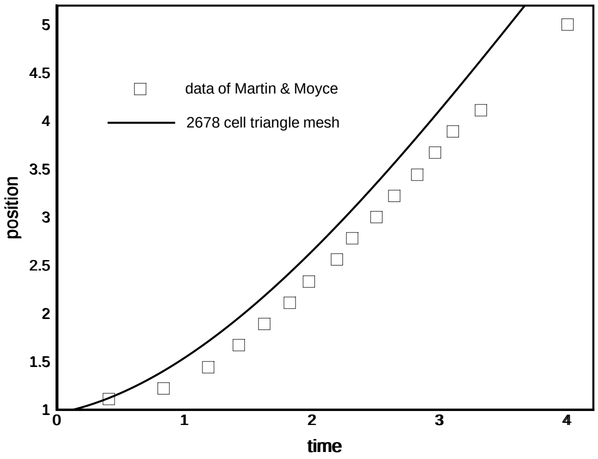

Figure 5. NON-DIMENSIONALIZED FRONT POSITION VERSUS TIME FROM THE RESULTS OF FIGURE 4, COMPARED TO THE DATA OF (Martin and Moyce, 1952).

Rawahi, W. Tauber, J. Han, S. Nas and Y.-J. Jan. A front tracking method for the computations of multiphase flow. *Journal of Computational Physics* 169(2), 708-759 (2001).

S. Osher and R.P. Fedkiw. Level set methods: An overview and some recent results. *Journal of Computational Physics* 169(2), 463-502 (2001).

T.Y. Hou, J.S. Lowengrub and M.J. Shelley. Boundary integral methods for multicomponent fluids and multiphase materials. *Journal of Computational Physics* 169(2), 302-362 (2001).

W.J. Rider and D.B. Kothe. Reconstructing volume tracking. *Journal of Computational Physics* 141(1), 112-152 (1998).

D.B. Kothe, M.W. Williams, K.L. Lam, D.R. Korzekwa, P.K. Tubesing and E.G. Puckett. A second-order accurate, linearity-preserving volume tracking algorithm for free surface flows on 3-D unstructured meshes. In *Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conference*, Paper No. FEDSM99-7109 (1999).

T.J. Barth. Recent developments in high order K-exact reconstruction on unstructured meshes. In *31st Aerospace Sciences Meeting and Exhibit*, Paper No. AIAA-93-0668 (1993).

B.D. Nichols, C.W. Hirt and R.S. Hotchkiss. A solution algorithm for transient fluid flow with multiple free boundaries. Technical report LA-8355, LASL (1980).

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, LANL (1991).

C.M. Rhie and W.L. Chow, Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 21(11), 1525-1532 (1983).

J.C. Martin and W.J. Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London, Series A* 244, 312-324 (1952).

V. Maronnier, M. Picasso and J. Rappaz. Numerical simulation of free surface flows. *Journal of Computational Physics* 155(2), 439-455 (1999).

N. Lock, M. Jaeger, M. Medale and R. Occelli. Local mesh adaptation technique for front tracking problems. *International Journal for Numerical Methods in Fluids* 28(4), 719-736 (1998).

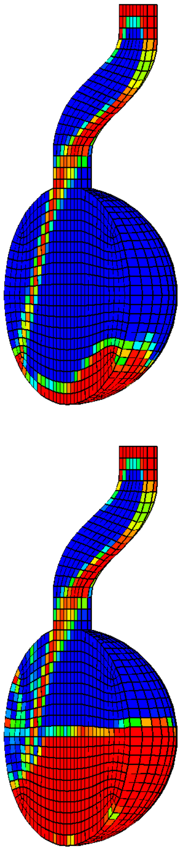

Figure 6. FOUR VIEWS DURING THE FILLING OF A VENTED HEMISPHERE. DENSITY RATIO 1000.

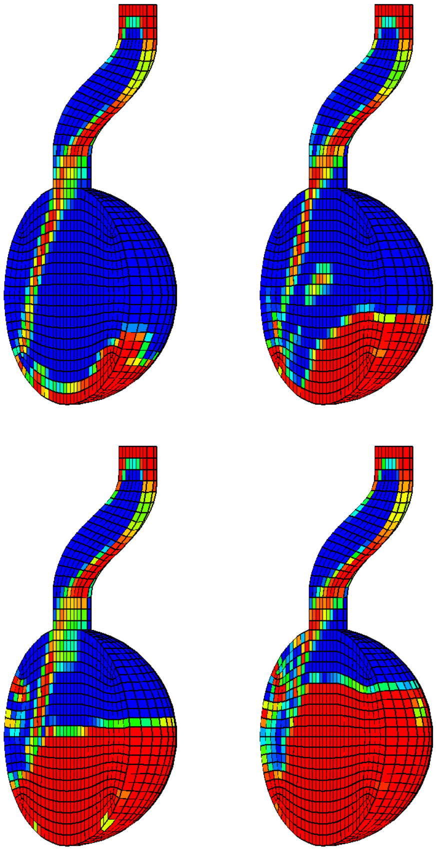

Figure 7. FOUR VIEWS DURING THE FILLING OF A VENTED HEMISPHERE. DENSITY RATIO 10^6.