A Numerical Method for Two Phase Flow Consisting of Separate Compressible and Incompressible Regions

Rachel Caiden [*†‡]
Ronald P. Fedkiw [* §]
Chris Anderson [†‡]

June 15, 2000

Abstract

We propose a numerical method for modeling two phase flow consisting of separate compressible and incompressible regions. This is of interest, for example, when numerically modeling the combustion of fuel droplets or the shock induced mixing of liquids. We use the level set method to track the interface between the compressible and incompressible regions, as well as the Ghost Fluid Method (GFM) to create accurate discretizations across the interface. The GFM is particularly effective here since the equations differ in both number and type across the interface. The numerical method is presented in two spatial dimensions with numerical examples in both one and two spatial dimensions, while three dimensional extensions are straightforward.

[*]Research supported in part by ONR N00014-97-1-0027.
[†]Research supported in part by AFOSR Grant F-49620-96-1-0327.
[‡]Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095
[§]Computer Science Department, Stanford University, Stanford, California 94305

1 Introduction

Problems with large density ratios, e.g. the combustion of fuel droplets or the shock induced mixing of liquids, are still rather difficult problems for modern computational fluid dynamics. These problems all concern the interaction of liquid droplets with a compressible gas medium. In general, there are three classical approaches to such problems; one can treat both phases as compressible, the gas as compressible and the liquid as incompressible, or both phases as incompressible.

When gas and liquid phases are treated as compressible, it is customary to model both with the fully compressible Navier-Stokes equations and a different equation of state for each phase. The change in equation of state is known to cause oscillations in numerical solutions near phase interfaces. These oscillations can be suppressed e.g. see [19] and [18] where the oscillations caused by the numerical method in [22] are removed. However, the suppression schemes have a side-effect that fluid properties can be smeared near interfaces. More details on the successful application of these types of numerical methods can be found in [1], [25], [32] and [31]. Numerical smearing across interfaces can be avoided using the Ghost Fluid Method (GFM) first proposed in [11] for two phase compressible flow and later extended to shocks, deflagrations, and detonations in [12]. While the Ghost Fluid Method can yield solutions with sharp fluid interfaces, a completely compressible treatment can be limiting because of the difference in sound speed between the liquid phase and gas phases. The more restrictive CFL condition in the liquid phase dictates a small time step for both phases, and this leads to inefficient numerical methods. In addition, a completely compressible approach is limited to liquids (or other materials) for which there are acceptable models for their compressible evolution.

To address such difficulties, we propose using the approach where the gas is modeled as a compressible fluid and the liquid is modeled as an incompressible fluid. The method can be viewed as a phase decomposition approach in which a high-resolution shock capturing scheme for the compressible flow is coupled with a standard incompressible flow solver for the liquid. The sharp liquid-gas interface is captured with the level set method [21]. Near the interface the Ghost Fluid Method is used to treat the boundary conditions in a manner that admits sharp discontinuities while still allowing for smooth discretizations across the interface. One important feature of our method is that we do not evolve the solution using operator splitting; in each time step both phases are updated simultaneously. Thus, the method avoids the time discretization errors that are associated with time-split schemes. The equations are solved with third order TVD Runge Kutta schemes in time and third order ENO schemes in space see [30, 13, 11, 17]. A method where the compressible and incompressible phases are also treated separately is presented in [14]. However, the method in [14] is restricted to one spatial dimension and it was not clear how to extend that technique to multiple spatial dimensions without ill-advised dimensional splitting.

In our procedure we are treating the liquid phase as incompressible. An alternate possibility that still retains the compressible nature of the liquid phase and avoids the time-step restriction of the difference in sound speeds would be to employ numerical methods designed specifically for low Mach number flow, e.g. [20] proposed a one dimensional numerical method based on asymptotics which was more recently extended to apply to a large class of standard compressible flow solvers in multiple dimensions [26]. In [5], this problem was treated with a semi-implicit method that was only implicit on those terms related to the speed of sound. See also [23] which generalized the work in [5]. A related method appears in [36] which splits the equations into an explicit advection phase and an implicit nonadvection phase. This particular method has been used to produce phenomenal images of fluid motion, see e.g. [35]. There are many other notable methods and the reader is referred in particular to [10] which uses a Hodge decomposition and [27] which addresses cancelation difficulties with low Mach number flows. The general technique that we outline for evolving the gas and liquid phases using separate models would apply to a method where the incompressible algorithm is replaced with one of the low Mach number solvers mentioned above; we leave this to future work.

Lastly, there are several methods that model both phases as incompressible, [34] , [4], [33] , [6] and [17], however, this approach is ruled out because our interest is in flows where compressible effects in the gas phase are important.

In the second section we describe the equations that are used to evolve the compressible fluid, the incompressible fluid, and the level set function. In addition, this section addresses the boundary conditions and coupling at the compressible/incompressible interface. The third section discusses the general time stepping strategy including the details required to advance each phase for one Euler time step. Section four addresses higher order TVD Runge Kutta methods and adaptive time stepping. Section five presents computational results that demonstrate the efficacy of our procedure. The numerical method is presented in two spatial dimensions with computational results in both one and two spatial dimensions. Three dimensional extensions are straightforward.

2 Equations and Their Discretization

2.1 Compressible Flow

The basic equations for two dimensional compressible flow are the Euler equations which can be written as:

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (E + p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ (E + p)v \end{pmatrix}_y = 0 \tag{1}$$

where t is the time, x and y are the spatial dimensions, ρ is the density, u and v are the velocities, E is the total energy per unit volume, and p is the pressure. The total energy is the sum of the internal energy and the kinetic energy,

$$E = \rho e + \frac{\rho(u^2 + v^2)}{2} \tag{2}$$

where e is the internal energy per unit mass. The pressure can be written as a function of density and internal energy, $p = p(\rho, e)$. For the sake simplicity only a gamma law gas, $p = (\gamma - 1)\rho e$, is considered in this paper. Note that the effects of viscosity, thermal conductivity, and mass diffusion are ignored in the compressible gas. The compressible flow equations are discretized using 3rd order accurate ENO methods. See [30, 13] for more details.

2.2 Incompressible Flow

The equations for incompressible flow can be deduced from the compressible flow equations by setting the divergence of the velocity field, $\vec{V} =< u, v >$, to zero obtaining,

$$\vec{V}_t + \vec{V} \cdot \nabla \vec{V} + \frac{\nabla p}{\rho} = \frac{\mu \triangle \vec{V}}{\rho} \tag{3}$$

$$\nabla \cdot \vec{V} = 0 \tag{4}$$

where both ρ and μ are assumed to be constant in the incompressible region. The equations are discretized on a MAC grid using the projection method [9] which allows equation (3) to be rewritten as

$$\frac{\vec{V}^\star - \vec{V}^n}{\triangle t} + \vec{V} \cdot \nabla \vec{V} = \frac{\mu \triangle \vec{V}}{\rho} \tag{5}$$

and

$$\frac{\vec{V}^{n+1}-\vec{V}^{\star}}{\triangle t}+\frac{\nabla p}{\rho}=0 \tag{6}$$

where the convection terms are discretized with standard 3rd order Hamilton Jacobi ENO methods [11, 16], and the viscous terms are discretized with standard second order central differencing. Once $\vec{V}^{\star}$ has been computed, the Poisson equation

$$\Delta p^{\star}=\nabla\cdot\vec{V}^{\star} \tag{7}$$

is discretized with Dirichlet boundary conditions on the pressure. This equation is derived by taking the divergence of equation (6) noting that the divergence of $\vec{V}^{n+1}$ is identically zero. Also note that the pressure has been rescaled using

$$p^{\star}=\left(\frac{\triangle t}{\rho}\right)p \tag{8}$$

to define a scaled pressure, $p^{\star}$. After solving equation (7) for the scaled pressure, the appropriate form of equation (6) given by

$$\vec{V}^{n+1}-\vec{V}^{\star}+\nabla p^{\star}=0 \tag{9}$$

is used to obtain $\vec{V}^{n+1}$.

2.3 The Level Set Method

The level set equation

$$\phi_t+\vec{V}\cdot\nabla\phi=0 \tag{10}$$

is used to track the interface between the compressible and the incompressible regions. $\phi\le 0$ designates the incompressible fluid and $\phi>0$ designates the compressible fluid. Hamilton Jacobi WENO methods [11, 16] are used to advect the level set function according to equation (10) and to reinitialize the level set function according to

$$\phi_t+S(\phi_0)(|\vec{\nabla}\phi|-1) \;=\; 0 \tag{11}$$

which was first proposed in [33]. The level set function is used to define the unit normal at every grid point as

$$\vec{N}=\frac{\vec{\nabla}\phi}{|\vec{\nabla}\phi|}=<n_1,n_2> \tag{12}$$

using central differencing where $\vec{N}$ points from the incompressible fluid into the compressible fluid. In the rare case that the denominator is identically zero, one sided differencing is used to calculate ϕ_x and ϕ_y in order to allow at least one nonzero value to be calculated. The curvature at each grid point is defined as

$$\kappa = \nabla \cdot \vec{N} = \frac{(\phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy} + \phi_x^2 \phi_{yy})}{(\phi_x^2 + \phi_y^2)^{1.5}} \tag{13}$$

and discretized using standard central differencing. In order to ensure that under-resolved regions do not erroneously contribute large surface tension forces, thresholding is applied to the curvature so that it satisfies

$$|\kappa| \leq \frac{1}{\min(\triangle x, \triangle y)} \tag{14}$$

2.4 Interface Boundary Conditions

In order to obtain a numerical method that can treat the interface between compressible and incompressible flow one must first address the boundary conditions and coupling mechanisms at the interface. Since the interface is a contact discontinuity moving with the local fluid velocity, $\vec{V}$, the Rankine Hugoniot jump conditions imply that $[p] = 0$ and $[V_N] = 0$, i.e. both the pressure and the normal velocity, $V_N = \vec{V} \cdot \vec{N}$, are continuous across the interface, see e.g. [11].

In the presence of thermal conduction, the temperature is continuous across the interface. In this paper, thermal conductivity effects are ignored introducing an uncoupled variable across the interface. When considering compressible flow, this can be thought of as an equation of state variable, e.g. ρ or e. We choose the entropy, S, as the equation of state variable since the entropy obeys a simple advection equation of the form

$$S_t + \vec{V} \cdot \nabla S = 0 \tag{15}$$

away from shocks implying that the entropy is not convected across the interface (which moves at speed V_N in the normal direction) [11]. In the incompressible flow, both density and internal energy obey equation (15) as well with S replaced by either ρ or e respectively [24]. Thus, similar to the entropy, information in these variables does not cross the interface. In the presence of viscosity, the tangential velocities are continuous, and the $[p] = 0$ boundary condition needs to be modified to account for the viscous stress, see e.g. [17]. In this paper, the compressible fluid is inviscid implying that there is no viscous coupling across the interface so that $[p] = 0$ remains valid while the tangential velocities are uncoupled across the interface. The nonzero incompressible viscosity only acts internal to the incompressible fluid. In addition, note that the tangential velocities obey equation (15) as well implying that information in these variables does not cross the interface.

In order to design a numerical method, the interface needs well defined values of all the independent variables. This can be achieved by specifying ρ, $\vec{V}$, and e on the incompressible side of the interface and S, $\vec{V}$, and p on the compressible side of the interface. All the uncoupled variables can be determined using one sided extrapolation to the interface. These variables are the tangential velocities on both sides of the interface, the incompressible density and internal energy, and the compressible entropy. In the Ghost Fluid Method, these interface values are not directly used, but instead these interface values are captured using one sided extrapolation of these quantities into ghost cells on the opposite side of the interface. Note that both the incompressible density and internal energy are treated as spatially constant so that no numerical treatment of these variables is needed. In fact, the incompressible internal energy can be completely omitted from the problem. The $[V_N] = 0$ jump condition implies that the normal velocity is continuous across the interface and that both the compressible and incompressible normal velocity must be considered when determining the unique value of the interface normal velocity which is used on both sides of the interface. Once the interface normal velocity has been determined, all that remains is the compressible pressure, and since all other interface values are determined, this variable is actually uncoupled! Therefore the interface value of the compressible pressure is determined with one sided extrapolation and its interface value can be captured with ghost cells similarly to the variables that obey equation (15). This is quite surprising since the interface separating two phase compressible flow requires the same coupling for the pressure that is required for the normal velocity [11].

The interface normal velocity can be determined using any number of interpolation techniques. However, one should be careful to realize that the interface normal velocity should be defined in a way that is consistent with incompressible flow. That is, since the incompressible region and its boundary should behave in a way that conserves area (or volume in three dimensions), the interface normal velocity needs to be consistent with the interior incompressible flow providing an extra global constraint on the interface normal velocity which is related to the compatibility condition, see

[24]. For this reason, the interface normal velocity is determined solely from the incompressible fluid values. This gives the interface velocity an incompressible character that helps to alleviate area (or volume) loss. Once again, the exact interface velocity is not actually computed, but captured using one sided extrapolation from the incompressible region. At this point, one might have legitimate concerns over the coupling mechanisms, that is, while the compressible fluid sees an incompressible interface velocity, the incompressible fluid is oblivious to the compressible velocity field. However, the incompressible fluid is coupled to the compressible fluid in the projection step. The compressible interface pressure is used as a Dirichlet boundary condition when solving a Poisson equation in the incompressible region, and the results are used to update the incompressible velocity field providing the proper coupling. Note that using the compressible pressure in this way also enforces the $[p] = 0$ boundary condition. In the presence of surface tension, the compressible pressure is not used directly, but is first modified according to the appropriate $[p] = \sigma\kappa$ jump condition.

3 Solution Advancement

In this section we describe how our procedure advances the solution one Euler time step. Higher order TVD Runge Kutta methods can be obtained as a combination of Euler time steps and simple averaging as explained in the next section.

At the beginning of a time step the level set function, ϕ, is defined at all grid nodes. The zero contour of the level set, $\{(x,y)\,|\,\phi(x,y)=0\}$, delineates the interface between compressible and incompressible fluids. The values of the compressible fluid are indicated by $\phi > 0$ and those of the incompressible fluid are indicated by $\phi \leq 0$. The compressible fluid values of mass, momentum and energy, designated by $\vec{U}$, are known at the nodes of a non-staggered grid while the incompressible fluid velocities are known at staggered MAC grid locations. The MAC grid values of ϕ are defined using averaging of the nodal values, e.g. $\phi_{i+\frac{1}{2},j} = \frac{\phi_{i,j}+\phi_{i+1,j}}{2}$.
To advance the solution consists of carrying out three calculations:

1. Extending the incompressible and compressible fluids across the interface using the ghost fluid technique.

2. Computing $\vec{U}^{n+1}$, ϕ^{n+1}, and $\vec{V}^{\star}$, i.e. advance the compressible fluid and the level set function one time step, and compute the intermediate value of the incompressible velocity field $\vec{V}^{\star}$.

3. Projecting $\vec{V}^{\star}$ onto its divergence free component in the region defined by $\phi^{n+1} \leq 0$ to obtain $\vec{V}^{n+1}$ for the incompressible fluid. Note that this step also accounts for the interface forces imposed by the compressible pressure.

The extension of the flow variables across the interface allows the calculations in step 2 to be implemented using standard difference formulas without regard to the position of the interface. The success of the procedure depends critically upon the manner in which the fluids are extended across the interface; the procedure used here is an extension of the Ghost Fluid Method [11]. Since our spatial discretization uses a combination of staggered and non-staggered grids some additional complexity is introduced into the technique. However, this complexity is tolerated because the use of a staggered MAC grid for the incompressible fluid greatly simplifies the implementation of the projection calculation in step 3.

We begin more detailed descriptions of the steps 1-3 with a discussion of the methods for extending the incompressible and compressible fluids.

3.1 Incompressible Fluid Extension

Incompressible velocities need to be defined at ghost nodes in the compressible region in order to advance the incompressible velocity field. In section 2.4 it is concluded that these values should be obtained by extrapolation from their values in the incompressible region. Constant extrapolation in the normal direction to the interface can implemented by solving

$$I_\tau + \vec{N} \cdot \vec{\nabla} I = 0 \tag{16}$$

in fictitious time τ for $I = u$ on the subset of the MAC grid where the u component of the incompressible velocity field is defined, and separately for $I = v$ on the subset of the MAC grid where the v component of the incompressible velocity field is defined. Instead of time marching, a first order accurate solution to the steady state of equation (16) can be obtained using the fast (velocity) extension method in [2] (which is based on the Fast Marching Method, see e.g. [28]). We prefer this method as it substantially reduces the computational execution time.

3.2 Compressible Fluid Extension

The compressible fluid extension at nodes within the incompressible region (i.e. compressible ghost fluid nodes), is defined by the values of its velocity, its entropy and its pressure. As discussed in section 2.4, the interface boundary conditions dictate that the entropy, the pressure, and the tangential velocity at the ghost fluid points be extrapolated from their values in the compressible region. The values of the entropy and pressure are extrapolated using the fast extension method in [2]. In order to construct a ghost cell velocity, we follow the procedure in [11]. The idea is to extrapolate the entire compressible velocity field to the ghost points using [2] and then obtain the tangential velocity at the ghost points by computing the difference between the extrapolated velocity and its normal component. The total velocity is then computed as

$$\vec{V} = \left(\vec{V}_I \cdot \vec{N}\right) \vec{N} + \left(\vec{V}_{ext} - \left(\vec{V}_{ext} \cdot \vec{N}\right) \vec{N}\right) \tag{17}$$

where the first term is the normal component of the incompressible velocity and the second term is the tangential component of the extrapolated compressible velocity. The incompressible velocity, $\vec{V}_I$, needs to be defined at the ghost nodes. If the extrapolated values of the incompressible velocity are defined first (as outlined above), then simple averaging can be used to obtain the incompressible velocity at each ghost node. Note that equation (17) does not require explicit knowledge of the tangent plane making it easy to implement in three dimensions. Once the ghost node values for the velocity, pressure and entropy have been defined, the conserved variables at the ghost nodes can be reassembled.

3.3 Computing U^{n+1}, ϕ^{n+1}, and $\vec{V}^\star$

With the compressible ghost fluid values defined, compressible fluid values are advanced one time step by applying the ENO discretization procedure at points in the compressible region. Note that a band of ghost nodes are updated in time as well so that they are appropriately defined in case the level set changes sign making them real fluid grid nodes. Since the normal velocity of the interface is defined by the incompressible velocity field, this velocity field is used in equation (10) for the evolution of the level set function. Thus, to advance ϕ in time, the velocity in equation (10) is computed at the grid nodes using simple averaging of the extended incompressible velocity field. Finally, $\vec{V}^\star$ is computed by applying the ENO discretization procedure to all points within the incompressible region including a band about the interface.

3.4 Projecting the incompressible fluid increment $\vec{V}^\star$

Once $\vec{V}^\star$ and ϕ^{n+1} have been computed the discrete Poisson equation with Dirichlet boundary conditions

$$\Delta p^\star = \nabla \cdot \vec{V}^\star \tag{18}$$

is used to obtain the scaled pressure in the region where $\phi^{n+1} \leq 0$. The boundary conditions for $p^\star$ are obtained from p^{n+1} at all compressible points adjacent to the incompressible region using the formula

$$p^\star = \left(\frac{\triangle t}{\rho^I}\right)\left(p^{n+1} + \sigma\kappa\right)$$

where the $\frac{\triangle t}{\rho^I}$ multiplier accounts for the scaling, ρ^I is the incompressible density, and the $\sigma\kappa$ term accounts for the jump in pressure due to surface tension forces, i.e. $[p] = \sigma\kappa$. Note that the curvature is computed at each grid point using the level set function, ϕ^{n+1}. To solve (18) we use a preconditioned conjugate gradient (PCG) method with an Incomplete Choleski preconditioner [15]. Once $p^\star$ has been computed $\vec{V}^{n+1}$is obtained with the relation $\vec{V}^{n+1} = \vec{V}^\star - \nabla p^\star$.

4 Runge Kutta and Adaptive Time Stepping

Since both second and third order TVD Runge Kutta schemes [29] can be written as a convex combination of simple Euler steps, see [29, 17], it is straightforward to generalize the first order time discretization discussed in section 3 to third order TVD Runge Kutta. One difficulty in implementing Runge Kutta methods in problems with interfaces arises when nodal values change character as the interface moves (e.g. one may inadvertently average incompressible and compressible flow values). However, the use of the ghost fluid technique circumvents this difficulty. First, the values of the level set can be averaged directly. Second, the values of the compressible fluid can be averaged using the appropriate ghost cell values where necessary. Third, the incompressible velocity can be averaged using the extended values of the $\vec{V}_{MAC}$ velocity field where needed. Note that the values of $\vec{V}_{MAC}$ in the ghost cells are determined by one sided extrapolation of the incompressible velocity, and thus do not exactly satisfy the divergence free condition although they do have incompressible character. This can cause slight jumps in the pressure at the interface as a larger than normal pressure gradient is needed to enforce exact incompressibility. Also, when using these extended velocities in a Runge Kutta averaging procedure the resulting velocity field is not exactly divergence free. However, the numerical results show that the area loss is small especially when compared to any standard level set calculation. Thus, these slightly compressible edge velocities do not seem to be a significant source of error. Note that one could remove these errors entirely by defining the extended velocity field using a divergence free constraint similar to the process outlined for free surface flows, see e.g. [7] and [8].

Adaptive time stepping is used where the overall time step is the minimum of the compressible and incompressible time steps, i.e.

$$\triangle t = .5 \min(\triangle t^C, \triangle t^I) \tag{19}$$

where we have chosen a CFL restriction of .5. For compressible flow, the convective time step restriction

$$\triangle t^C \left(\frac{|u| + c}{\triangle x} + \frac{|v| + c}{\triangle y} \right) \leq 1 \tag{20}$$

needs to be satisfied at every grid point where $c = \sqrt{\frac{\gamma p}{\rho}}$ is the speed of sound. For incompressible flow, every grid point needs to satisfy

$$\triangle t^I \left(\frac{(C_{cfl} + V_{cfl}) + \sqrt{(C_{cfl} + V_{cfl})^2 + 4(S_{cfl})^2}}{2} \right) \leq 1 \tag{21}$$

where

$$C_{cfl} = \frac{|u|}{\triangle x} + \frac{|v|}{\triangle y} \tag{22}$$

is for the convection terms,

$$V_{cfl} = \frac{\mu}{\rho} \left(\frac{2}{(\triangle x)^2} + \frac{2}{(\triangle y)^2} \right) \tag{23}$$

is for the viscous terms, and

$$S_{cfl} = \sqrt{\frac{\sigma \kappa}{\rho \left(\min\{\triangle x, \triangle y\} \right)^2}} \tag{24}$$

is for the surface tension forces [17].

5 Numerical Examples

In this section, we report on numerical examples which demonstrate the accuracy and convergence behavior of the method. In particular, these examples show that the fluid quantities are not smeared out near the interface nor do the numerical solutions exhibit nonphysical oscillations. Also, all the two dimensional numerical examples had less than $\frac{1}{2}\%$ area loss on the finest grids. The calculations performed here used 3rd order accurate TVD Runge Kutta methods and adaptive time stepping as discussed in section 4. Unless otherwise specified, the two dimensional examples include the effects of viscosity and surface tension with $\mu = .001137\frac{kg}{ms}$ and $\sigma = .0728\frac{kg}{s^2}$. These effects are not present in one spatial dimension.

5.1 One Dimensional Case

In one spatial dimension, the incompressible flow equations are greatly simplified. Equation (4) becomes $u_x = 0$ implying that the incompressible velocity is constant. Equation (3) then becomes

$$u_t + \frac{p_x}{\rho} = 0 \tag{25}$$

implying that equation (5) is just $u^\star = u^n$. Equation (7) becomes $p^\star_{xx} = 0$ implying that the incompressible pressure is merely a straight line connecting the values of $p^\star$ on the left and right boundaries. In fact, equation (9) becomes

$$u^{n+1} - u^n + \frac{p^\star_{right} - p^\star_{left}}{L} = 0 \tag{26}$$

where L is the length of the incompressible region.

5.1.1 Example 1

Consider a 1m domain with 200 grid cells. The domain is filled with a compressible gas with $\gamma = 1.4$, $\rho = 1.226\frac{kg}{m^3}$, $u = 0\frac{m}{sec}$ and $p = 1 \times 10^5 Pa$, except for a $.2m$ incompressible droplet in the center of the domain with $\rho = 1000\frac{kg}{m^3}$, $u = 100\frac{m}{sec}$ and $p = 1 \times 10^5 Pa$. Since the incompressible droplet is moving to the right in a gas which is originally at rest, a compression wave will form in the gas ahead of it and an expansion wave will form in the gas behind it as shown in figure 1 at $t = 7.5 \times 10^{-4}$ seconds where the *red* region represents the compressible fluid and the *blue* region represents the incompressible fluid. The density, velocity and pressure all drop across the few grid cell thick compression wave, although the density jump is too small to be seen in the figure. The density and pressure drop while the velocity rises across the smooth expansion wave which is resolved by the grid, although once again, the density change is too small to be seen in the figure. Figure 2 shows similar behavior with an incompressible density of $\rho = 10\frac{kg}{m^3}$. Note that the lighter droplet is slowed down faster by the compressible gas, and as a result secondary expansion waves with significant amplitude stretch between the droplet and the lead compression and expansion waves. A grid refinement study was preformed on both calculations using grids of 200, 400, and 800 cells. The incompressible velocity was used for the comparison with Aitken extrapolation [3]. The computed velocities of $99.7216\frac{m}{sec}$, $99.7189\frac{m}{sec}$ and $99.7175\frac{m}{sec}$ from the coarsest to the finest mesh yield a convergence rate of .9475 for the $\rho = 1000\frac{kg}{m^3}$ case, and the computed velocities of $75.6466\frac{m}{sec}$, $75.4843\frac{m}{sec}$ and $75.4043\frac{m}{sec}$ yield a convergence rate of 1.0206 for the $\rho = 10\frac{kg}{m^3}$ case. Figure 3 shows the results obtained with 800 grid cells for the $\rho = 10\frac{kg}{m^3}$ case to illustrate the behavior of the variables under mesh refinement.

5.1.2 Example 2

In this example, the ambient compressible medium has $\rho = 1.58317\frac{kg}{m^3}$, $u = 0\frac{m}{sec}$ and $p = 98066.5Pa$. A shock wave is initially located at $x = .1m$ with a post shock state of $\rho = 2.124\frac{kg}{m^3}$, $u = 89.981\frac{m}{s}$, and $p = 148407.3Pa$ to the left of $x = .1m$. The shock wave travels to the right impinging on the incompressible droplet with initial state of $\rho = 1000\frac{kg}{m^3}$, $u = 0\frac{m}{sec}$ and $p = 98066.5Pa$ causing both reflected and transmitted waves as shown in figure 4 at $t = 1.75 \times 10^{-3}$ seconds. Note that the transmitted wave is too weak to be seen in this figure, although it can clearly be seen in figure 5 which shows the same calculation with an incompressible density of $\rho = 10\frac{kg}{m^3}$. Figure 6 shows the calculation at an earlier time of $t = 9 \times 10^{-4}$ seconds with a density of $10\frac{kg}{m^3}$, shortly after the shock has initially impinged on the droplet. Note that the transmitted wave has traversed the droplet at infinite speed and is now entering the gas on the far side. A grid refinement study was preformed on both calculations using grids of 200, 400, and 800 cells using the incompressible velocity for the comparison. The computed velocities of $.544424\frac{m}{sec}$, $.5444639\frac{m}{sec}$ and $.5444742\frac{m}{sec}$ from the coarsest to the finest mesh yield a convergence rate of 1.0617 for the $\rho = 1000\frac{kg}{m^3}$ case, and the computed velocities of $40.9873\frac{m}{sec}$, $40.8685\frac{m}{sec}$ and $40.8074\frac{m}{sec}$ yield a convergence rate of .9593 for the $\rho = 10\frac{kg}{m^3}$ case. Figure 7 shows the results obtained with 800 grid cells for the $\rho = 10\frac{kg}{m^3}$ case to illustrate the behavior of the variables under mesh refinement.

5.2 Two Dimensional Case

5.2.1 Example 3

Consider a $[0m, 1m] \times [0m, 1m]$ domain with 100 grid cells in each direction. Similar to example 1, the domain is filled with a compressible gas with $\rho = 1.226\frac{kg}{m^3}$, $u = v = 0\frac{m}{sec}$ and $p = 1 \times 10^5 Pa$, except for a $.2m$ radius incompressible droplet in the center of the domain with $\rho = 1000\frac{kg}{m^3}$, $u = 100\frac{m}{sec}$, $v = 0\frac{m}{sec}$ and $p = 1 \times 10^5 Pa$. This incompressible droplet moves to the right causing a compression wave in the gas ahead of it and an expansion wave in the gas behind it. Figure 8 shows a one dimensional cross section of these waves at $t = 5 \times 10^{-4}$ seconds. Figures 9 and 10 show the pressure contours and the velocity field at the same time. Figure 11 shows the initial level set location as compared to the location at $t = 2.5 \times 10^{-3}$ seconds using 50, 100, and 200 grid cells in each direction. Careful examination of the right hand side of the level set location shows first order accurate convergence in the location of the interface. An area loss study was undertaken using the method outlined in the appendix. Initially, the area of the droplet is $.04\pi$. The area loss was .23%, .16%, and .0125% on grids with 50, 100, and 200 cells in each direction respectively. Similar results for $\rho = 10\frac{kg}{m^3}$ are shown in figures 12 and 13 where the area loss was .49%, .31%, and .13%. Notice that the lighter droplet has been deformed and slowed at a faster rate than the heavier droplet. Also note that the calculation on the finest mesh is starting to show signs of Kelvin-Helmholtz instability as demonstrated by the small wiggles in the interface location. This instability occurs when the tangential velocity is discontinuous across an interface as is required by the imposed no-slip interface boundary condition. On coarser grids, the numerical viscosity can nonphysically damp out this effect.

In order to illustrate the effects of viscosity and surface tension, we shrink the domain to $[0m, 1 \times 10^{-5}m] \times [0m, 1 \times 10^{-5}m]$ for the $\rho = 10\frac{kg}{m^3}$ case. Figure 14 shows a one dimensional cross section of these waves at $t = 5 \times 10^{-9}$ seconds. Note the jump in pressure due to surface tension effects. Figure 15 shows the initial level set location as compared to the location at 2.5×10^{-8} seconds using 50, 100, and 200 grid cells in each direction where the area loss was .225%, .107%, and .006% respectively. Note that the smaller droplet has a rounder shape as compared to the larger droplet in figure 13.

5.2.2 Example 4

Consider a $[0m, 1m] \times [0m, 1m]$ domain with 100 grid cells in each direction. Similar to example 2, the ambient compressible medium has $\rho = 1.58317\frac{kg}{m^3}$, $u = v = 0\frac{m}{sec}$ and $p = 98066.5Pa$. A shock wave is initially located at $x = .1m$ with a post shock state of $\rho = 2.124\frac{kg}{m^3}$, $u = 89.981\frac{m}{s}$, $v = 0\frac{m}{sec}$ and $p = 148407.3Pa$ to the left of $x = .1m$. The shock wave travels to the right impinging on the incompressible droplet with initial state of $\rho = 10\frac{kg}{m^3}$, $u = v = 0\frac{m}{sec}$ and $p = 98066.5Pa$, with radius $.2m$ at the center of the domain, causing both reflected and transmitted waves as shown in the one dimensional cross sections in figure 16 at 1.25×10^{-3} seconds. Figures 17 shows the velocity fields at the same time. Figure 18 shows the initial level set location as compared to the location at $t = 2.5 \times 10^{-3}$ seconds using 50, 100, and 200 grid cells in each direction where the area loss was 1.6%, .52%, and .43% respectively. Note that the calculation on the finest mesh is starting to show signs of Kelvin-Helmholtz instability.

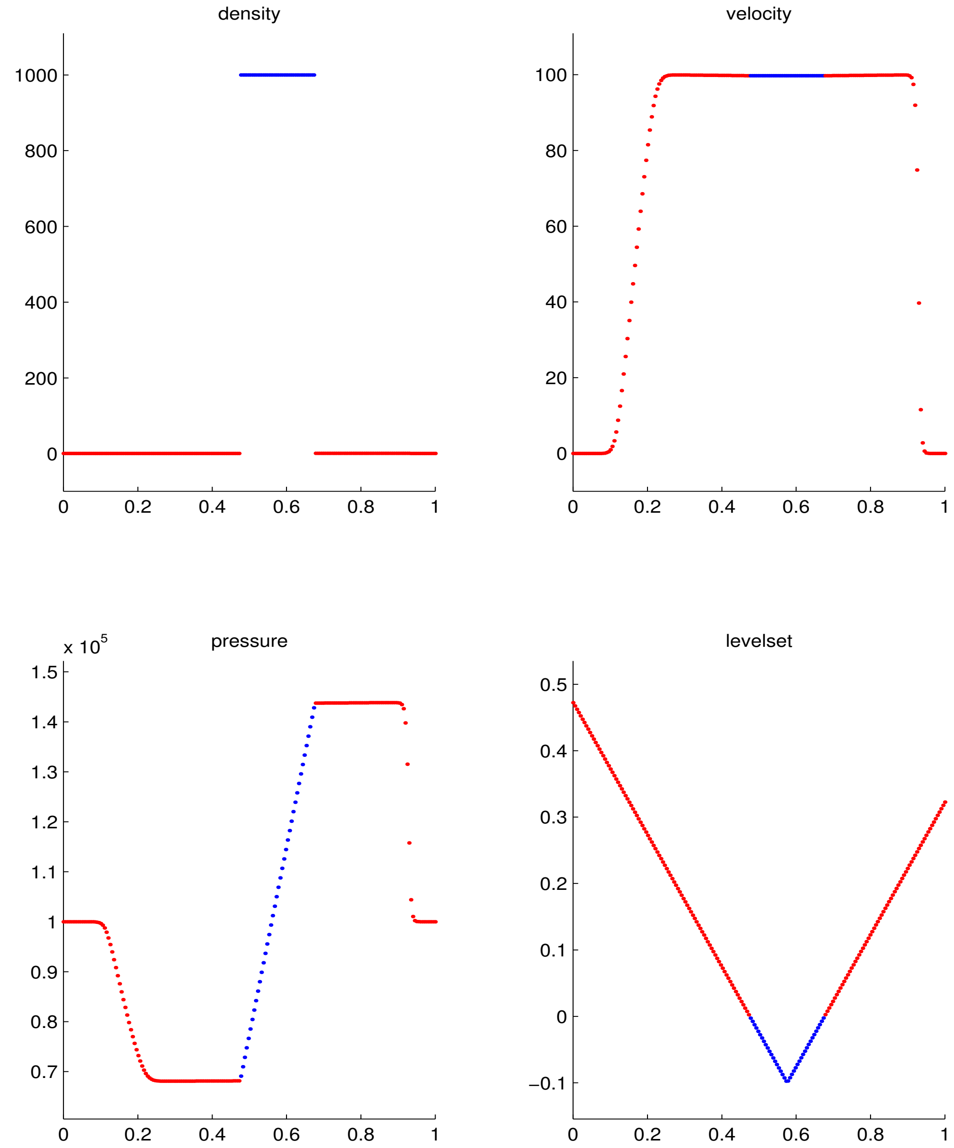

Figure 1: Incompressible $\rho = 1000\frac{kg}{m^3}$ droplet traveling to the right at $t = 7.5 \times 10^{-4}$ seconds.

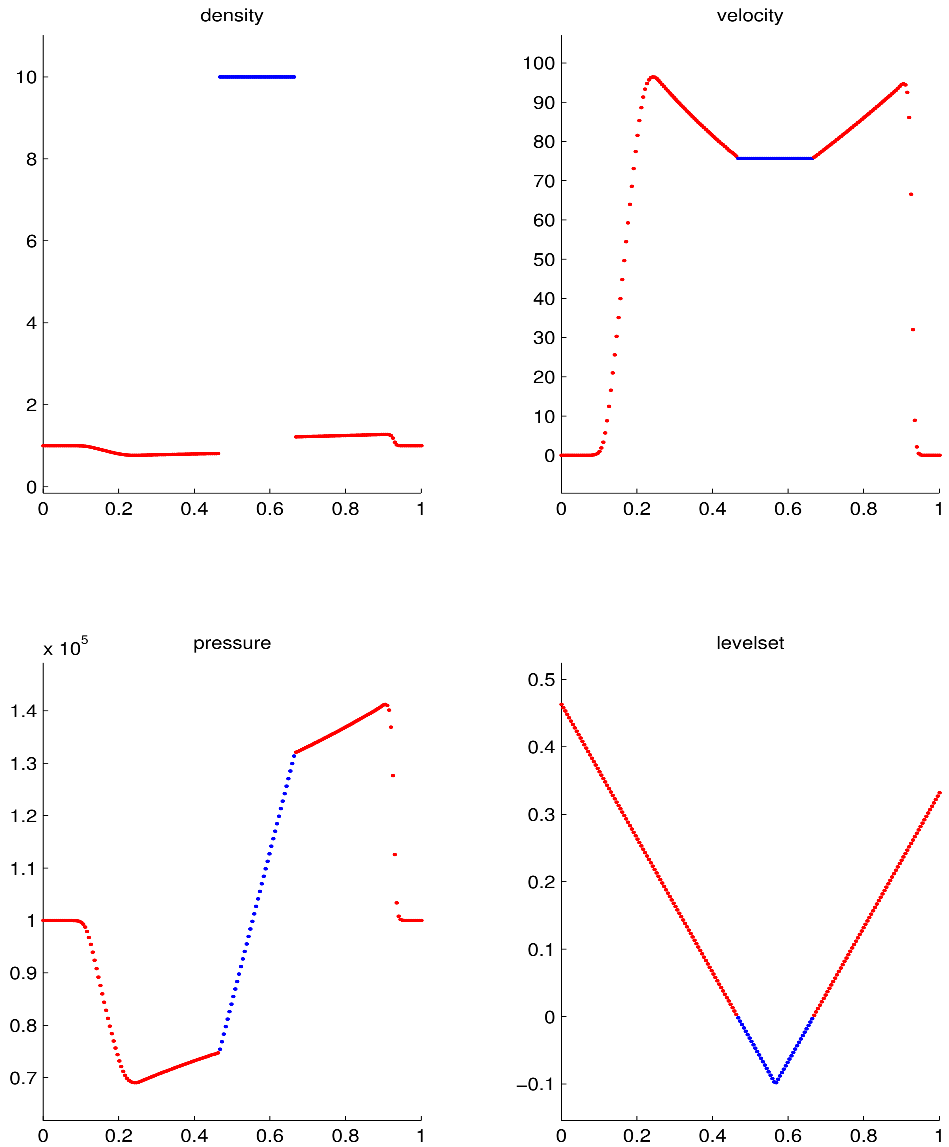

Figure 2: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right at $t = 7.5 \times 10^{-4}$ seconds (200 grid cells).

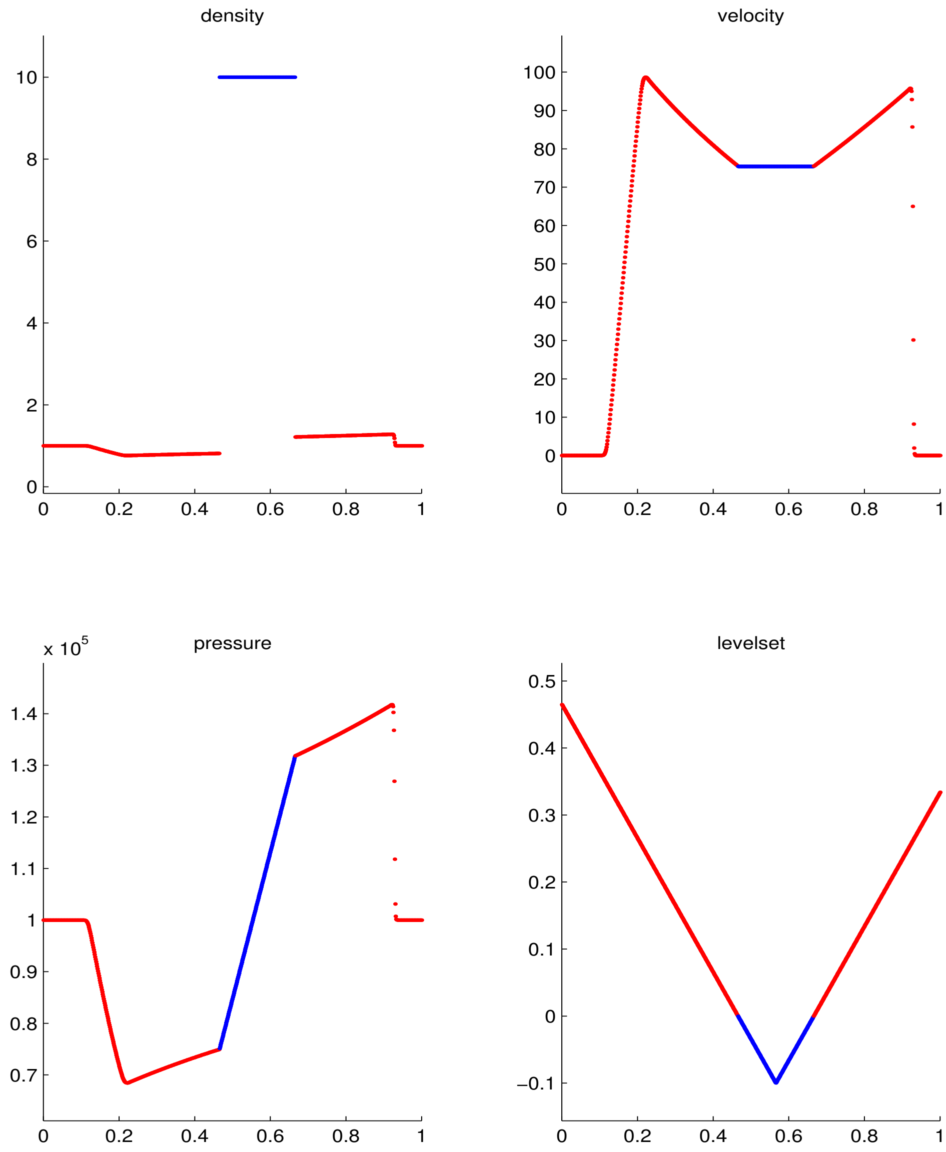

Figure 3: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right at $t = 7.5 \times 10^{-4}$ seconds (800 grid cells).

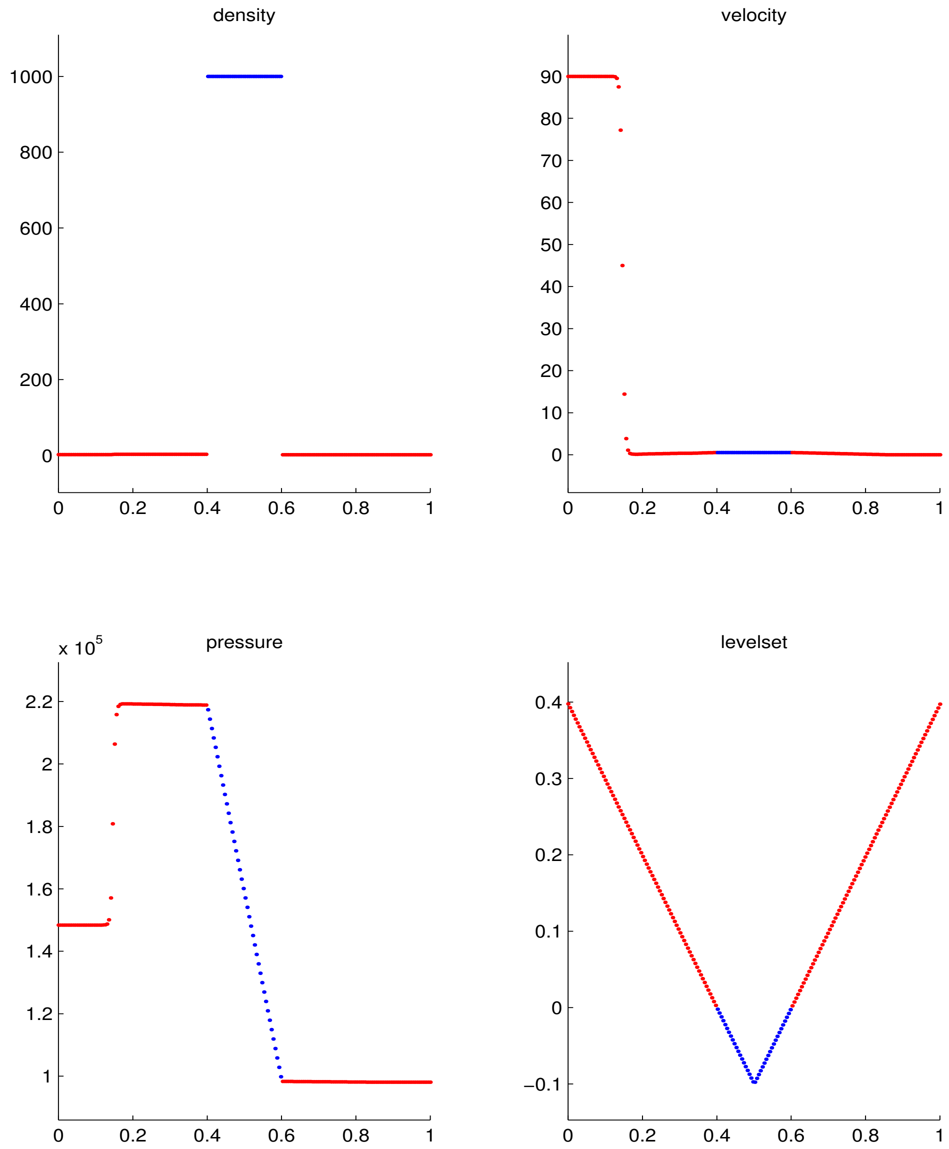

Figure 4: Shock wave impinging on an incompressible $\rho = 1000\frac{kg}{m^3}$ droplet at $t = 1.75 \times 10^{-3}$ seconds.

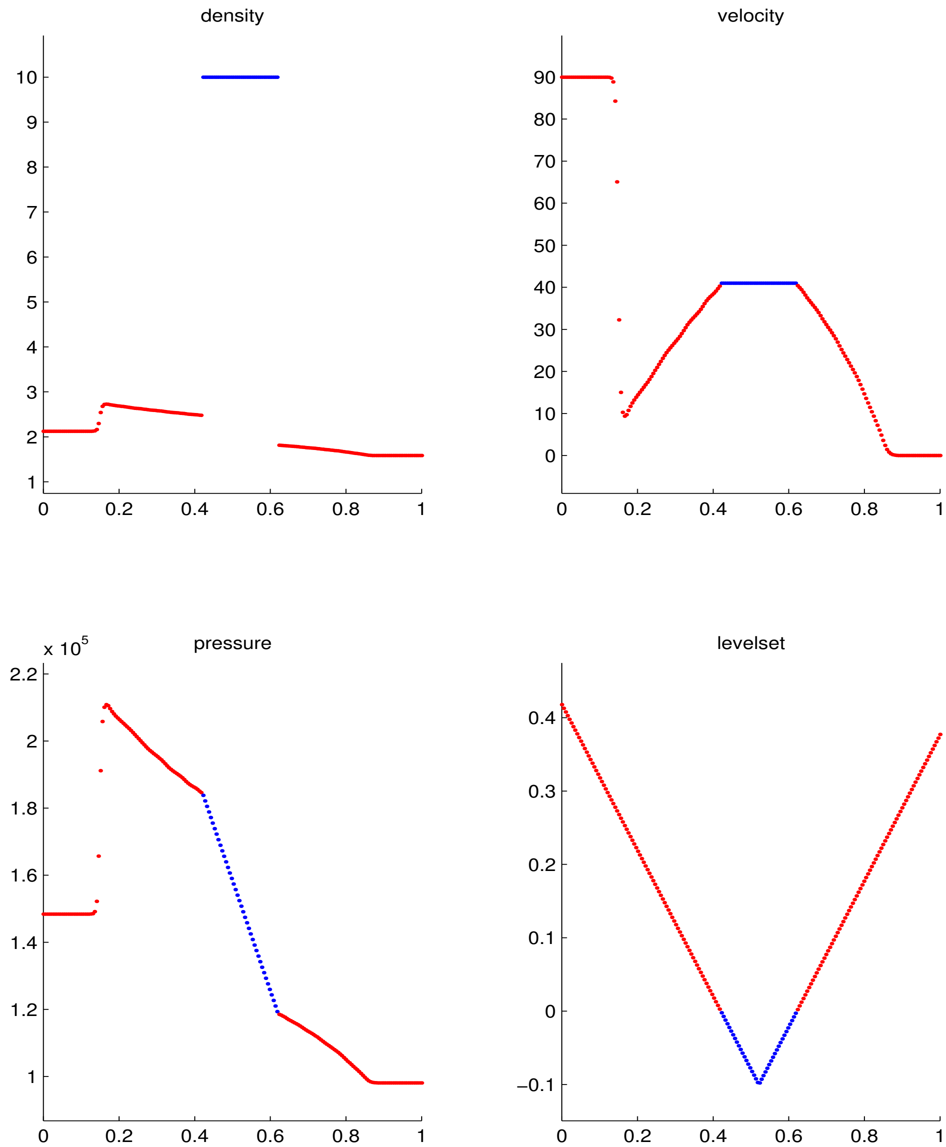

Figure 5: Shock wave impinging on an incompressible $\rho = 10 \frac{kg}{m^3}$ droplet at $t = 1.75 \times 10^{-3}$ seconds (200 grid cells).

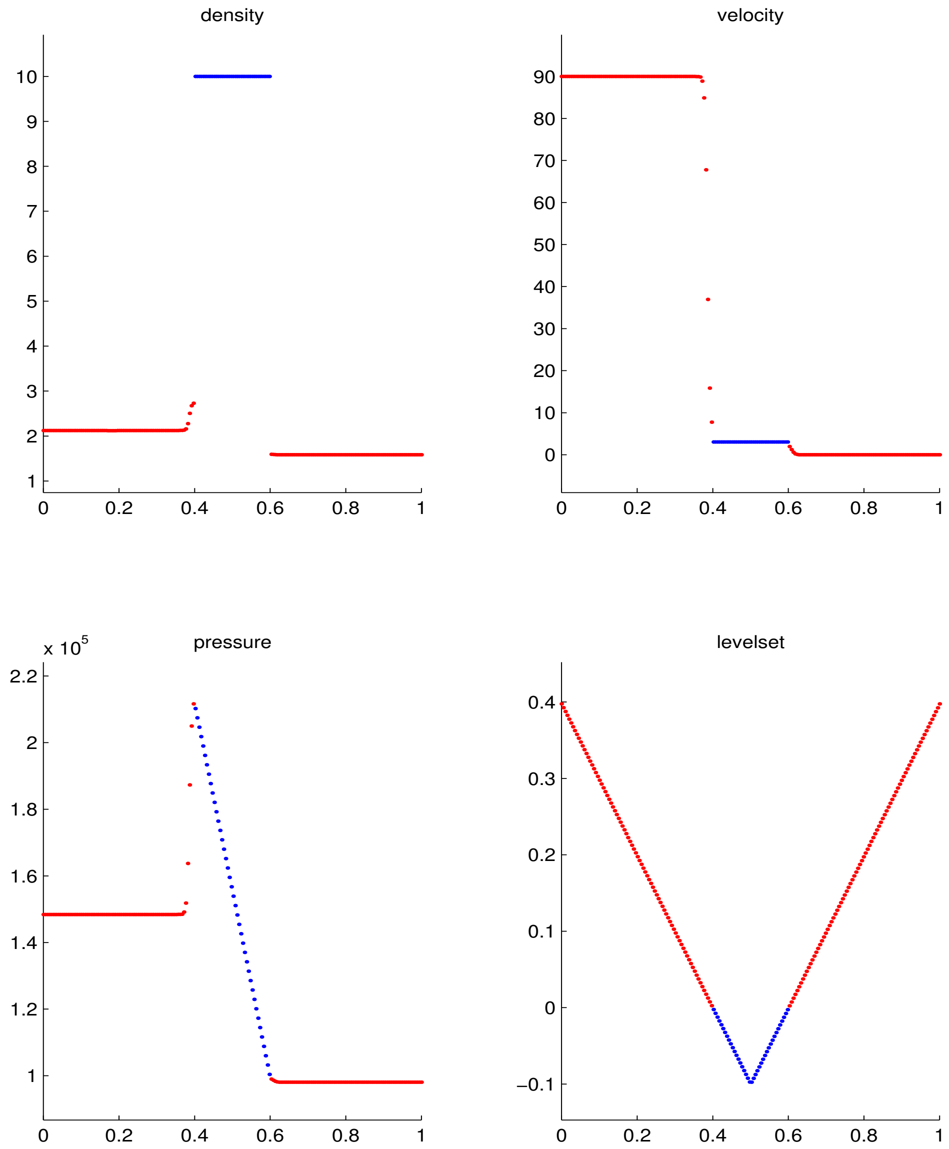

Figure 6: Shock wave impinging on an incompressible $\rho = 10\frac{kg}{m^3}$ droplet at $t = 9 \times 10^{-4}$ seconds.

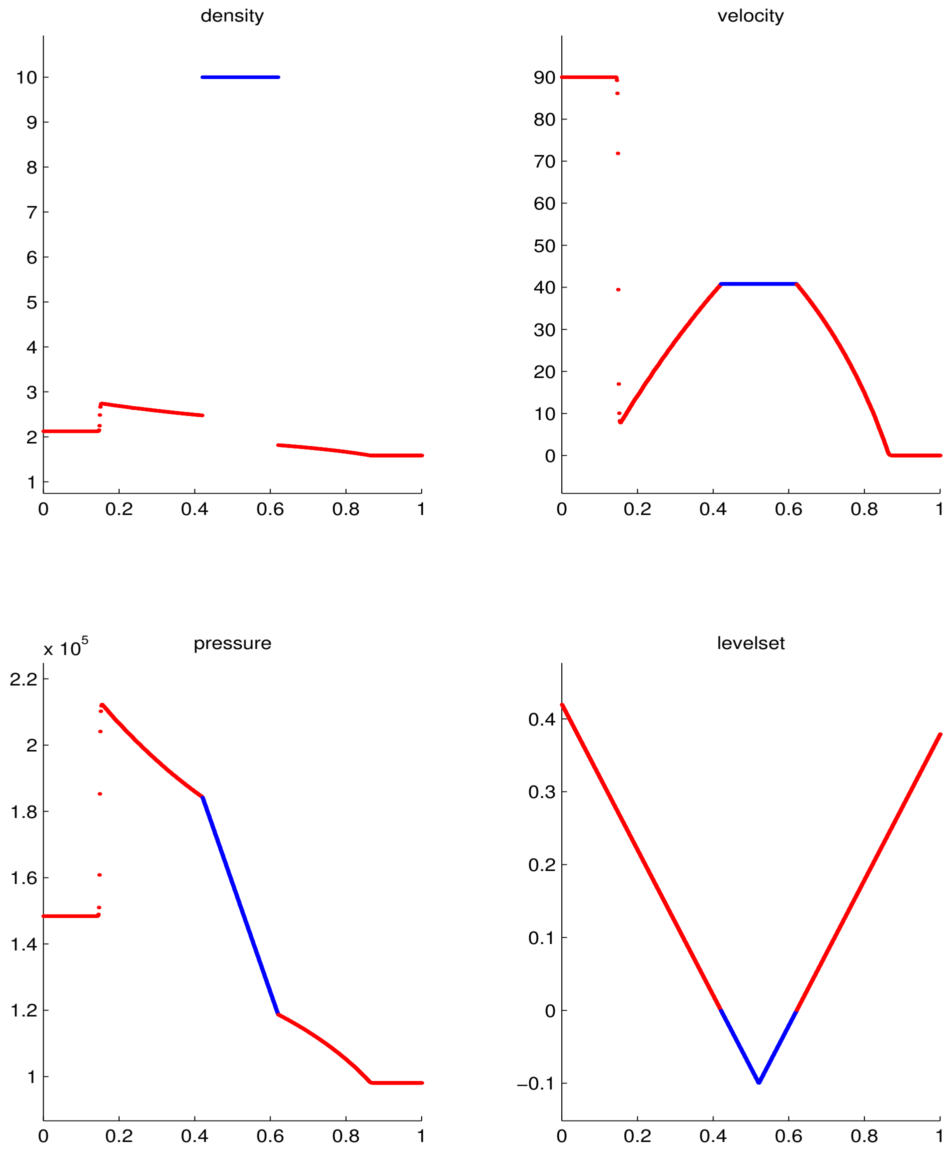

Figure 7: Shock wave impinging on an incompressible $\rho = 10\frac{kg}{m^3}$ droplet at $t = 1.75 \times 10^{-3}$ seconds (800 grid cells).

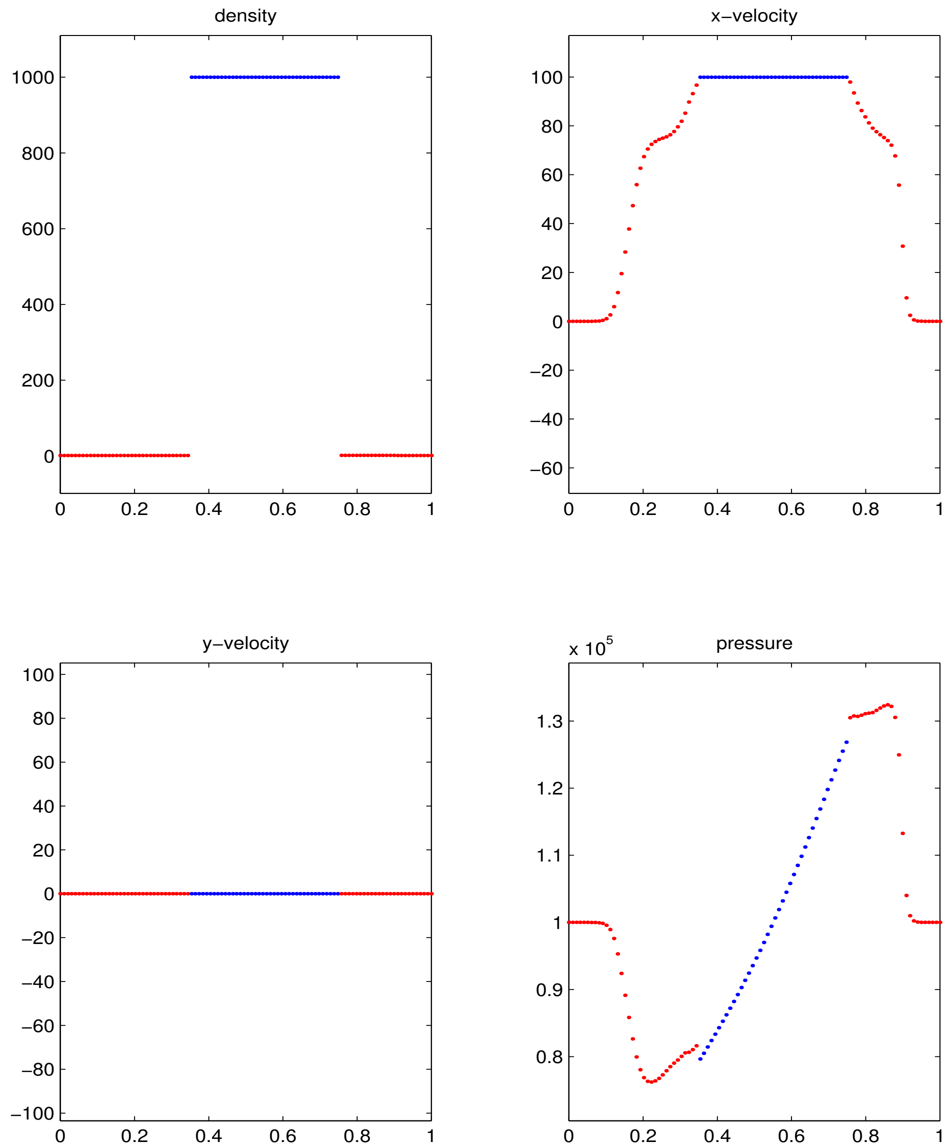

Figure 8: Incompressible $\rho = 1000\frac{kg}{m^3}$ droplet traveling to the right at $t = 5 \times 10^{-4}$ seconds (one dimensional cross section).

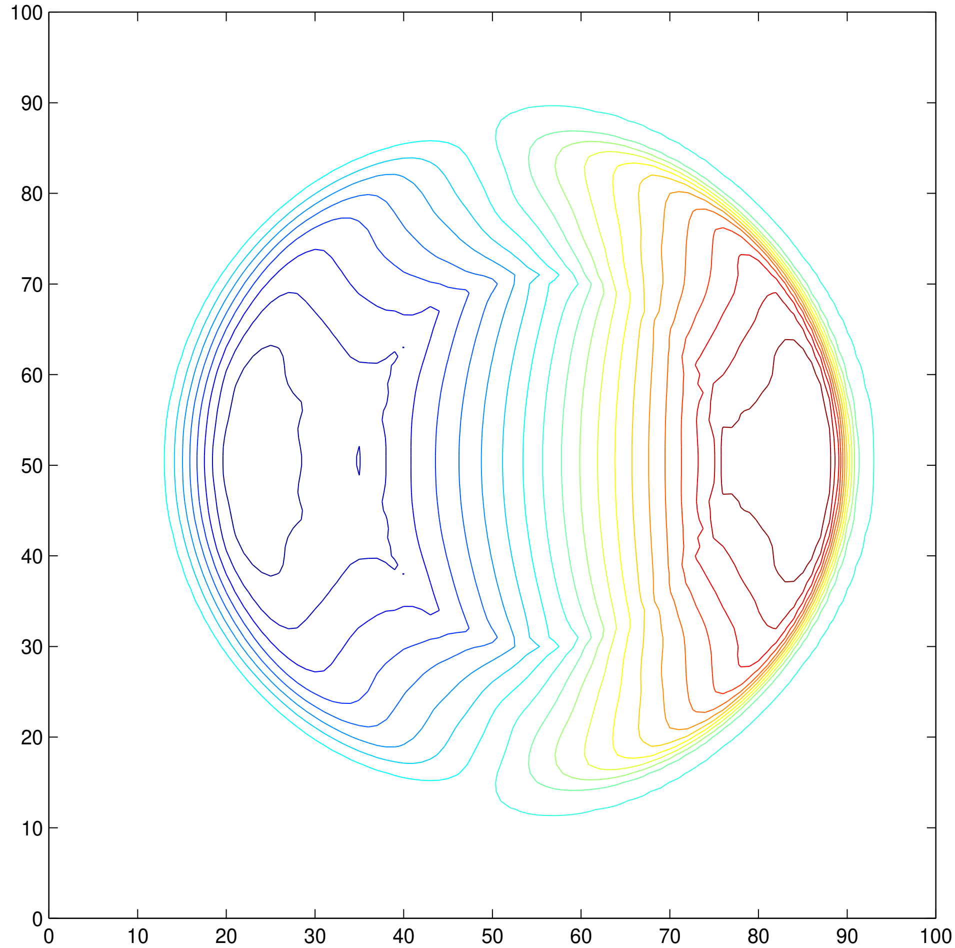

Figure 9: Incompressible $\rho = 1000\frac{kg}{m^3}$ droplet traveling to the right at $t = 5 \times 10^{-4}$ seconds (pressure contours).

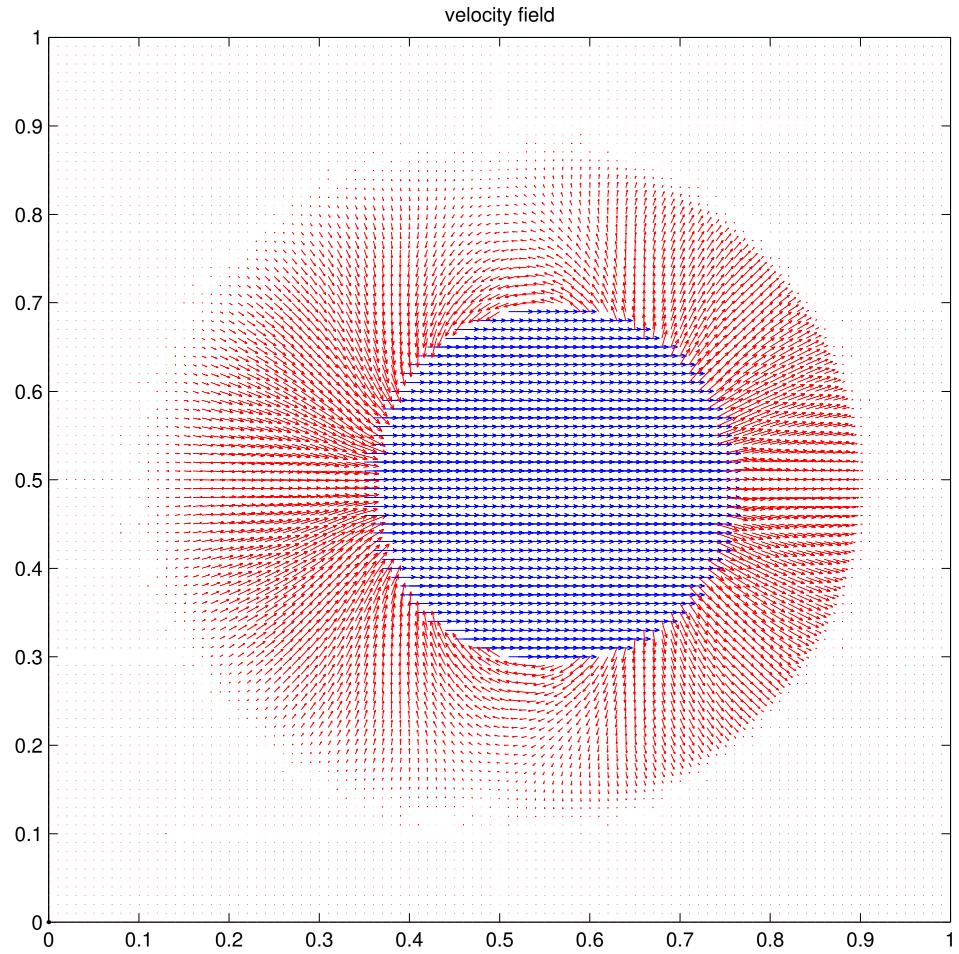

Figure 10: Incompressible $\rho = 1000\frac{kg}{m^3}$ droplet traveling to the right at $t = 5 \times 10^{-4}$ seconds (velocity field).

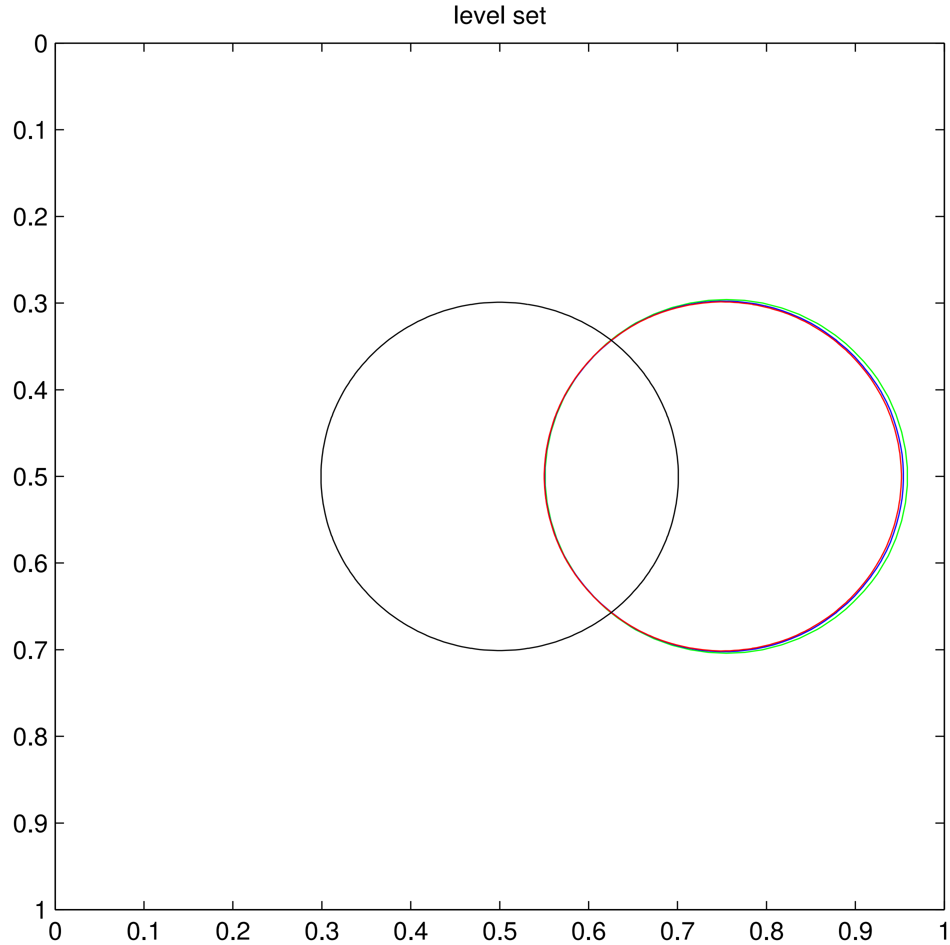

Figure 11: Incompressible $\rho = 1000\frac{kg}{m^3}$ droplet traveling to the right at $t = 2.5 \times 10^{-3}$ seconds as compared to the initial data (*green*, *blue* and *red* are successive levels of mesh refinement).

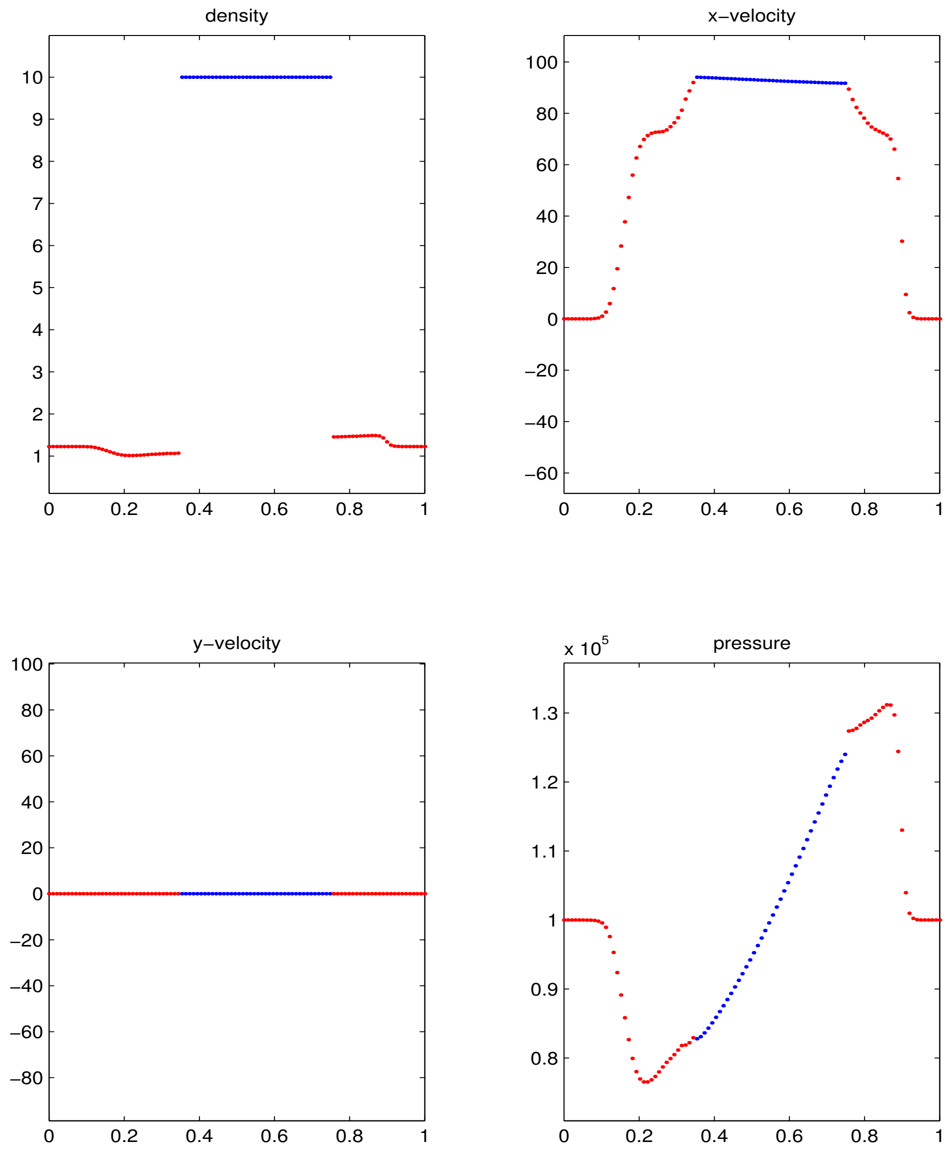

Figure 12: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right at $t = 5 \times 10^{-4}$ seconds (one dimensional cross section).

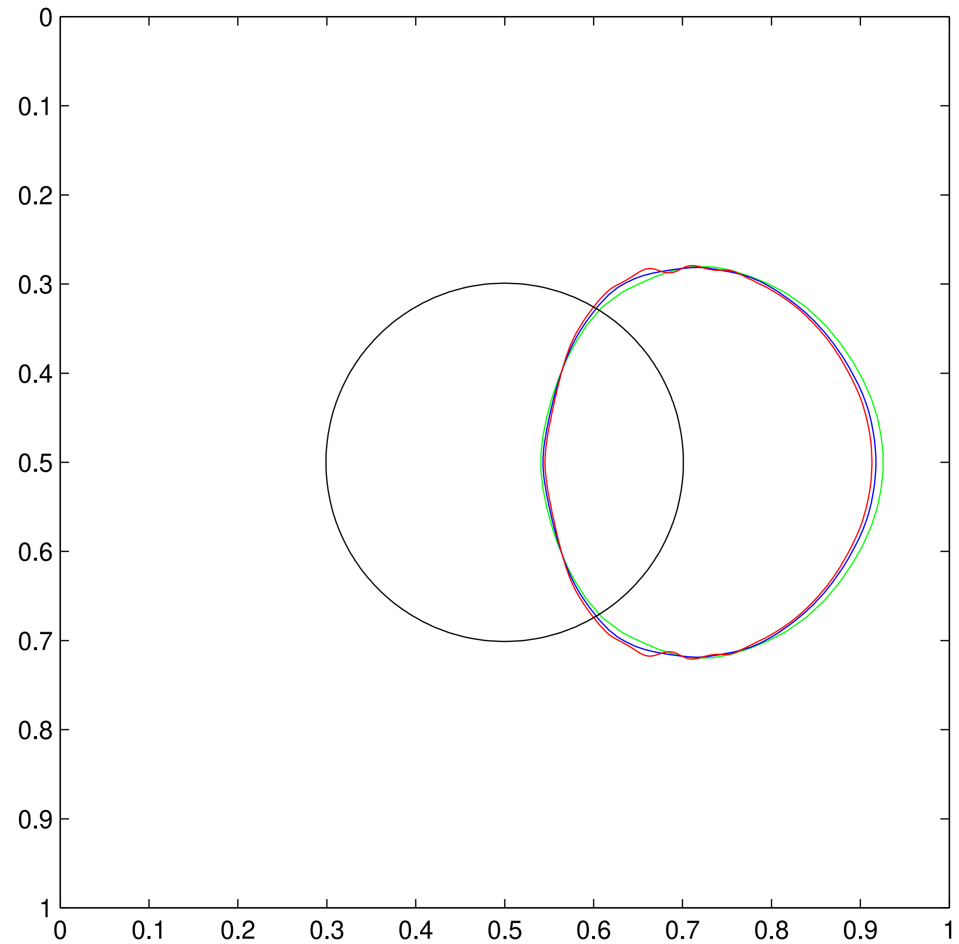

Figure 13: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right at $t = 2.5 \times 10^{-3}$ seconds as compared to the initial data (*green*, *blue* and *red* are successive levels of mesh refinement).

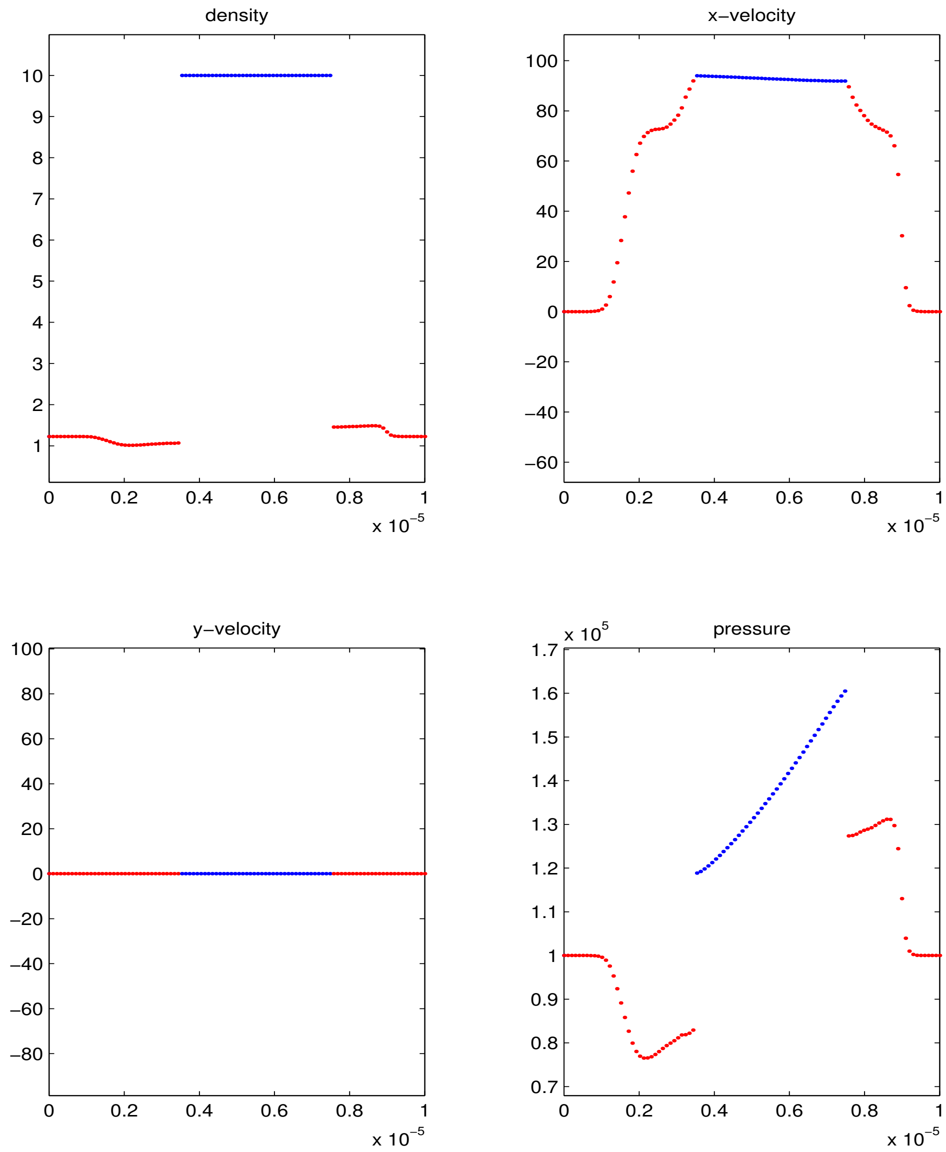

Figure 14: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right on a small domain at $t = 5 \times 10^{-9}$ seconds (one dimensional cross section).

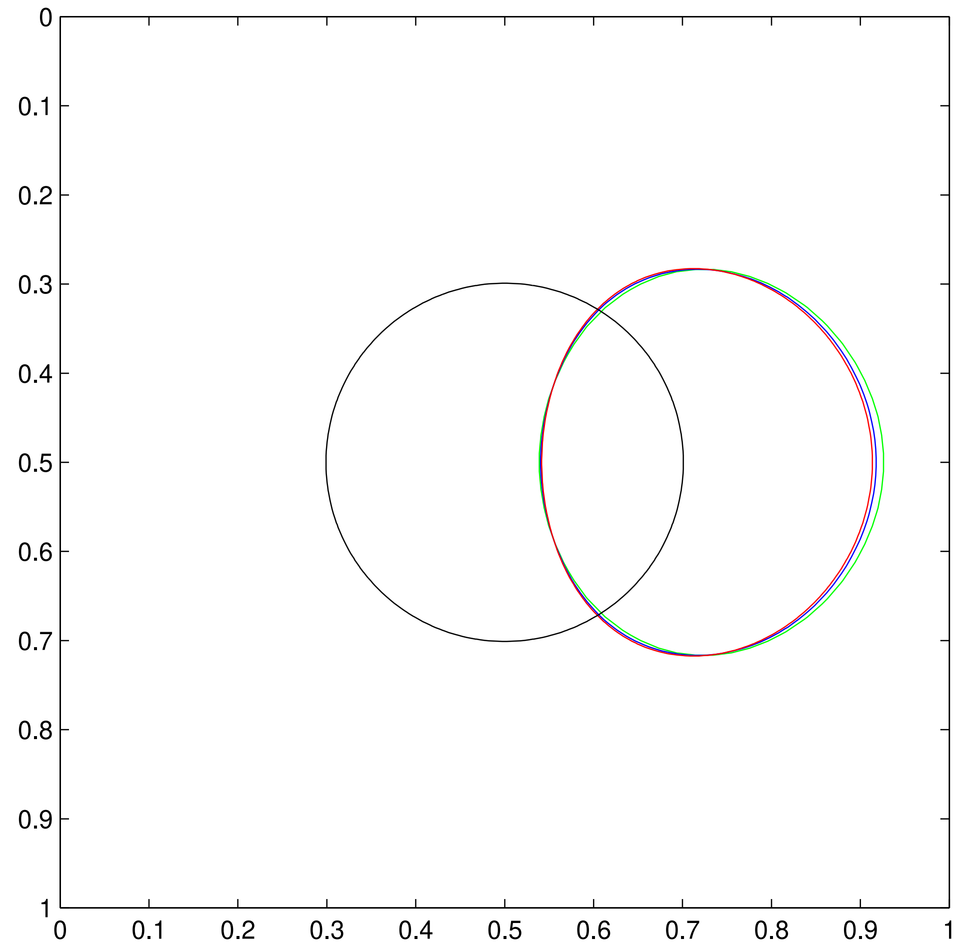

Figure 15: Incompressible $\rho = 10\frac{kg}{m^3}$ droplet traveling to the right on a small domain at $t = 2.5 \times 10^{-8}$ seconds as compared to the initial data (*green*, *blue* and *red* are successive levels of mesh refinement).

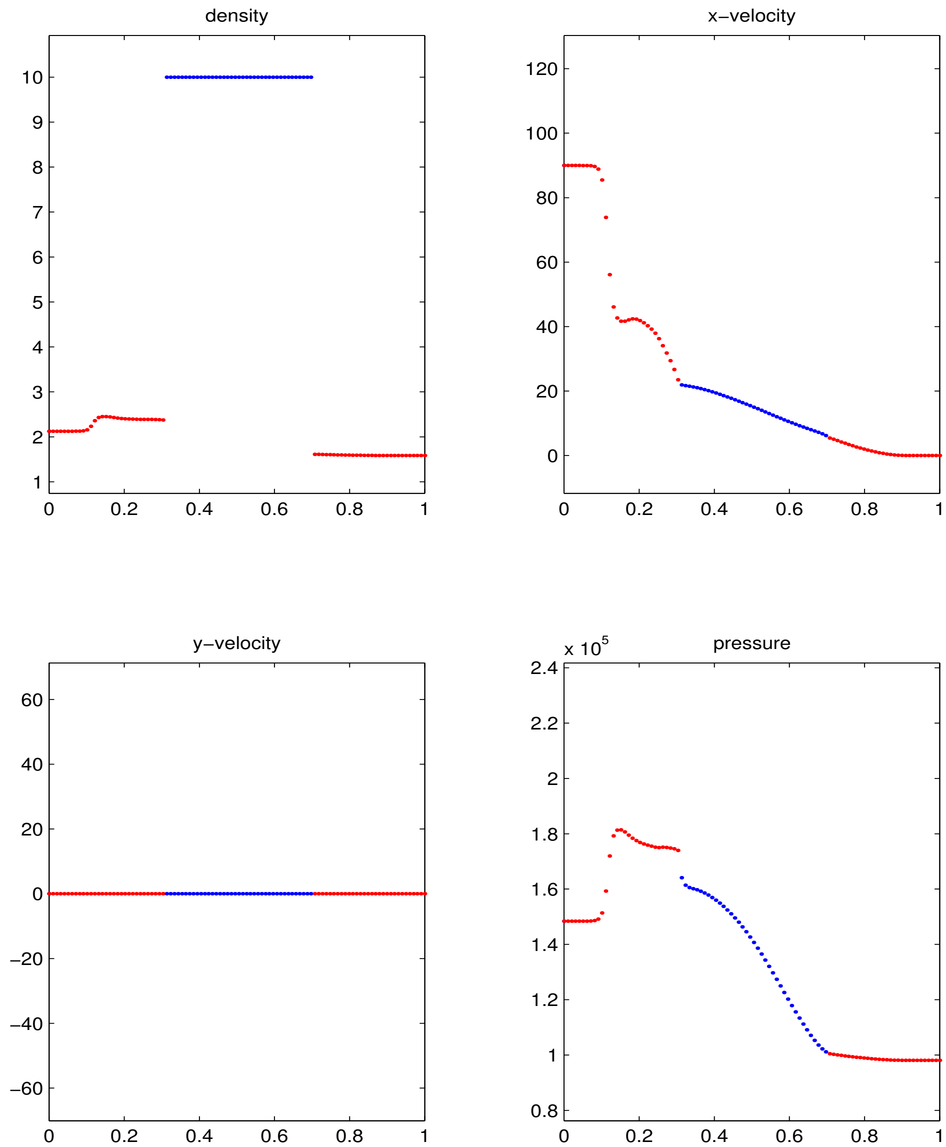

Figure 16: Shock wave impinging on an incompressible $\rho = 10\frac{kg}{m^3}$ droplet at $t = 1.25 \times 10^{-3}$ seconds (one dimensional cross section).

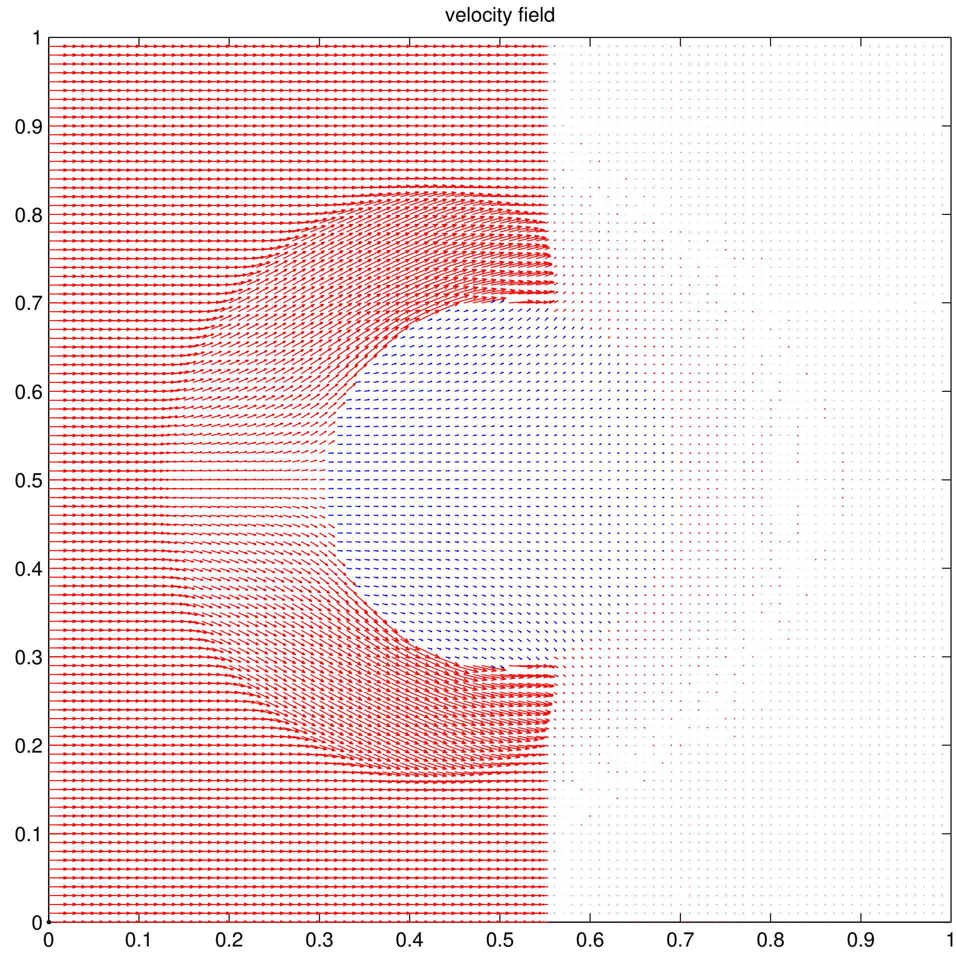

Figure 17: Shock wave impinging on an incompressible $\rho = 10\frac{kg}{m^3}$ droplet at $t = 1.25 \times 10^{-3}$ seconds (velocity field).

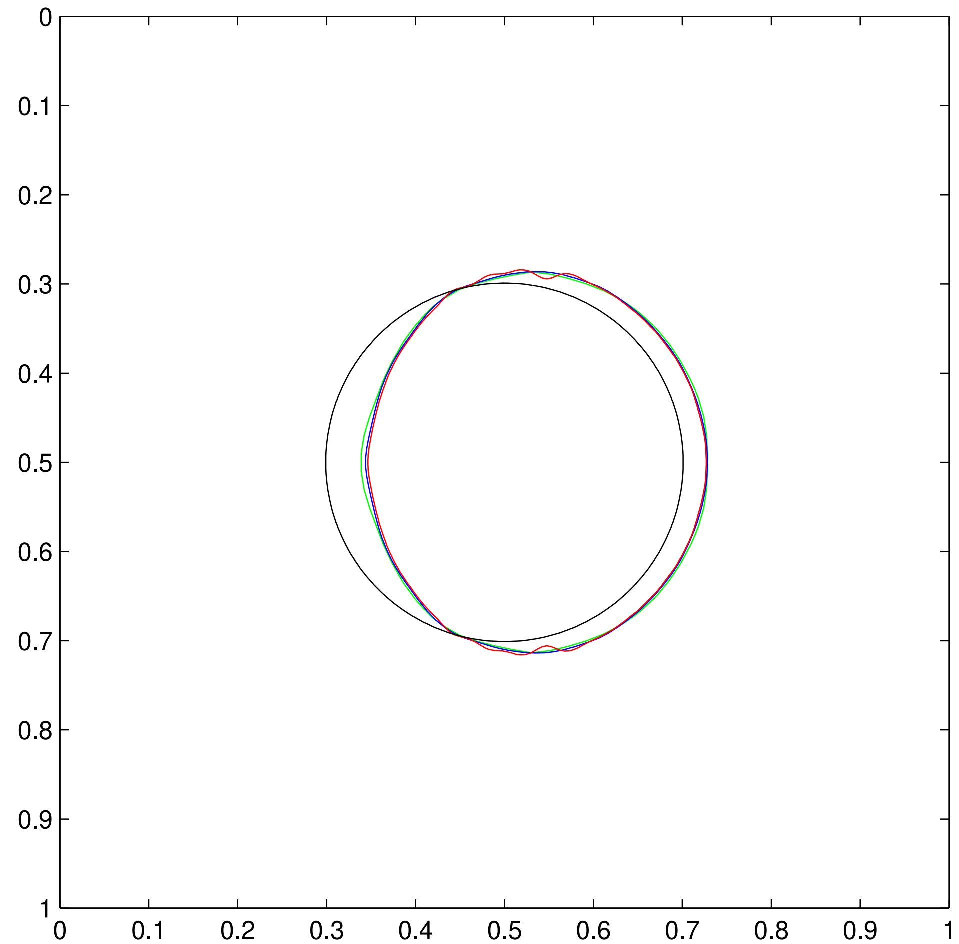

Figure 18: Shock wave impinging on an incompressible $\rho = 10\frac{kg}{m^3}$ droplet at $t = 2.5 \times 10^{-3}$ seconds as compared to the initial data (*green*, *blue* and *red* are successive levels of mesh refinement).

6 Conclusions and Future Work

In this paper we have presented a numerical method for two phase flow where one of the phases is treated as an incompressible flow and one is treated as a compressible flow. The primary computational difficulty in creating numerical schemes that respect the fundamentally different nature of the fluids in these phases is the creation and implementation of appropriate boundary conditions. We derive boundary conditions using "ghost fluid" ideas; the computational results indicate that high quality solutions can be obtained with their use. The test problem we considered was the behavior of an incompressible liquid when subjected to shock waves formed in a high speed gas flow. This test problem was primarily selected to investigate the ability of our proposed method to compute compressible/incompressible flow interactions when the compressible flow contains shocks. It is a separate (and interesting) problem in fluid mechanics to consider the validity of modeling liquid/gas phase interactions as an incompressible/compressible interaction. In future work, the validity of the incompressible assumption for the liquid will be tested by comparing the results obtained with the method presented here with the results obtained with a method where the liquid is modeled as a slightly compressible fluid.

A Unbiased Level Set Contouring

Consider a two dimensional level set function, ϕ, defined on a Cartesian grid. This appendix addresses the construction of an unbiased linear approximation to the zero contour (where $\phi = 0$) of the level set function (commonly used contour routines introduce a directional bias due to the choice of an underlying triangulation).

The standard contour plotting algorithms dictate triangulation of the domain followed by linear interpolation along each edge of each triangle resulting in the determination of the location of the zero values of the level set function along each edge. These zero values occur on two of the edges when the sign of the level set function on one corner is different from the sign on the other two corners, or on none of the edges when the sign of the level set function is the same on all three corners. In the case where the zero values occur on two of the edges, a line segment can be used to connect these two zero values leading to a piecewise linear subcell representation of the zero contour of the level set function. From this zero contour one can easily calculate quantities such as the area enclosed by or the length of the zero contour.

A straightforward way of choosing a triangulation consists of constructing a diagonal in every Cartesian grid cell. Let $\vec{x}_{i,j}$, $\vec{x}_{i+1,j}$, $\vec{x}_{i,j+1}$, and $\vec{x}_{i+1,j+1}$ represent a single grid cell where the subscripts place the points in the obvious locations. Then one could construct the *diagonal* connecting $\vec{x}_{i,j}$ and $\vec{x}_{i+1,j+1}$ or the *off-diagonal* connecting $\vec{x}_{i+1,j}$ and $\vec{x}_{i,j+1}$ as shown in figure 19a and 19b respectively. For each Cartesian grid cell, there are 4 distinct cases to consider. Case 1: all four nodal values have the same sign of ϕ (note that we classify $\phi = 0$ as negative, since we partition the domain into two parts consisting of $\phi \leq 0$ and $\phi > 0$). In Case 1, there is nothing to address since the cell does not contain any part of the zero contour. Case 2: one of the nodal values has a different sign than the other three. Case 3: there are two nodes of each sign and opposite corners are of the same sign. Case 4: there are two nodes of each sign and opposite corners are of opposite sign. Each case is discussed in detail below.

Consider Case 2 with $\phi_{i,j} \leq 0$, $\phi_{i+1,j} > 0$, $\phi_{i,j+1} > 0$ and $\phi_{i+1,j+1} > 0$. In this case, the diagonal determines two triangles which each contain part of the interface, while only one of the two triangles produced by the off-diagonal contains part of the interface, i.e. different answers are obtained depending on whether the diagonal or the off-diagonal is used. See figure 20 for an illustration of the diagonal case (figure 20a) and the off-diagonal case (figure 20b). In the figure, the shaded regions denote $\phi \leq 0$. Presumably, using the extra zero value that lies on the diagonal itself results in a more accurate construction as shown in figure 20a. Otherwise, there is no need for triangles in this case at all as one can construct the representation given by the off-diagonal construction in figure 20b by simply connecting the linearly interpolated zeroes on each side of the grid cell. Note that the diagonal gives extra information (an extra point) when either $\phi_{i,j}$ or $\phi_{i+1,j+1}$ is the point of differing sign, but gives no extra information (no extra point) when either $\phi_{i+1,j}$ or $\phi_{i,j+1}$ is the point of differing sign. For the case where either $\phi_{i+1,j}$ or $\phi_{i,j+1}$ is the point of differing sign, the off-diagonal must be used to pick up extra information (an extra point). Note that this case points out that it is unwise to use diagonals (or off-diagonals) everywhere since the reconstruction is biased. It is better to use an "adaptive" triangulation which always gives extra information, i.e. one should chose the diagonal or off-diagonal in order to obtain a construction similar to figure 20a and not figure 20b.

Consider Case 3 with $\phi_{i,j} \leq 0$, $\phi_{i+1,j} > 0$, $\phi_{i,j+1} > 0$ and $\phi_{i+1,j+1} \leq 0$. In this case, the diagonal construction implies that the line of sight (the line segment connecting two points in space) between $\phi_{i,j}$ and $\phi_{i+1,j+1}$ is contained in $\phi \leq 0$ while the line of sight between $\phi_{i+1,j}$ and $\phi_{i,j+1}$ is not contained in $\phi > 0$ as shown in figure 21a. Similarly, the off-diagonal construction implies that the line of sight between $\phi_{i+1,j}$ and $\phi_{i,j+1}$ is contained in $\phi > 0$, while the line of sight between $\phi_{i,j}$ and $\phi_{i+1,j+1}$ is not contained in $\phi \leq 0$ as shown in figure 21b. In level set notation, the diagonal construction implies that the negative values of the level set have (or are) "merged", while the off-diagonal construction implies that the positive values of the level set are merged. In fact, using a diagonal construction everywhere creates a grid dependence of increased merging in the diagonal direction, while using the off-diagonal construction everywhere creates a grid dependence of increased merging in the off-diagonal direction. Obviously, this is not desirable and some average of these two constructions is desired, especially since the information given (at the grid nodes) does not dictate whether or not merging has occurred. The choice of triangulation itself forces the merging. Using the linearly interpolated zero values on each of the four sides of the cell, one can see that the diagonal construction implies that the point on the bottom of the cell between $\vec{x}_{i,j}$ and $\vec{x}_{i+1,j}$ is connected to the point on the right of the cell between $\vec{x}_{i+1,j}$ and $\vec{x}_{i+1,j+1}$, while the point on the left of the cell between $\vec{x}_{i,j}$ and $\vec{x}_{i,j+1}$ is connected to the point on the top of the cell between $\vec{x}_{i,j+1}$ and $\vec{x}_{i+1,j+1}$ implying that the negative values are merged. Similarly, the off-diagonal construction implies that the point on the bottom of the cell is connected to the point on the left, while the point on the right is connected to the point on top implying that the positive values are merged. Since there are four points to be paired off into two linear segments, there is a total of three ways to make the connections. The diagonal and off-diagonal constructions give only two ways, leaving one possibility inaccessible to these straightforward triangulations. The remaining way to connect the four points consists of connecting the points on opposite sides of the cell giving a construction where neither the positive nor the negative values are merged as shown in figure 21c. In fact, both the positive and the negative values are in contact at a single saddle point formed by the intersection of the two line segments producing an "average" of the two triangulations. While achieving the desired compromise between positive and negative merging, this method does not use triangulation to determine an extra point as opposed to figure 20a. In addition, note that the positive and negative merging cases do not use an extra subcell point either, as they can be constructed by connecting the zero values of the Cartesian cell in the appropriate fashion. In fact, the positive and negative merging cases each contain two line segments similar to Case 2 without triangulation as was shown in figure 20b. Therefore, in order to introduce a new point within this cell to improve the accuracy, we choose the standard average of the four zero values on the Cartesian cell boundary. The zero contour is constructed by connecting this new point to each of the four zero values from which it was formed. Note that this construction can be obtained with an adaptive triangulation where the cell is divided into four triangles defined by the line segments connecting this new zero value inside the cell to each of the four corners of the cell. This adaptive triangulation and the resulting segmentation are shown in figure 21d.

Consider Case 4 with $\phi_{i,j} \leq 0$, $\phi_{i+1,j} \leq 0$, $\phi_{i,j+1} > 0$ and $\phi_{i+1,j+1} > 0$. In this case, one could simply connect the two zero values with a straight line ignoring triangulation as shown in figure 22a. Using triangulation gives a different subcell point depending on whether the diagonal (figure 22b) or the off-diagonal (figure 22c) is used (except for the case where the subcell point happens to be the intersection of the diagonal and the off-diagonal for *both* constructions). To avoid ambiguities one needs to determine which of these two candidates for the intermediate point should be used. Designating the subcell candidates by $\vec{x}_1$ and $\vec{x}_2$ and the zeroes on the Cartesian boundary by $\vec{x}_L$ and $\vec{x}_R$, both points can be used in the construction by connecting $\vec{x}_L$ to $\vec{x}_1$ to $\vec{x}_2$ to $\vec{x}_R$ (figure 22d) or by connecting $\vec{x}_L$ to $\vec{x}_2$ to $\vec{x}_1$ to $\vec{x}_R$

(figure 22e). However, this gives a subcell contour with a possibly large variation. Instead of choosing one or the other, we note that the line segment connecting $\vec{x}_1$ to $\vec{x}_2$ lies on both contours and choose the midpoint of this line segment (the standard average of $\vec{x}_1$ and $\vec{x}_2$) as the subcell zero value and connect this midpoint to each of the zeroes on the Cartesian boundary resulting in a construction with less variation (a shorter length) than one using both $\vec{x}_1$ and $\vec{x}_2$. Once again, this construction can be obtained with an adaptive triangulation where the cell is divided into four triangles defined by the line segments connecting this new zero value inside the cell to each of the four corners of the cell. This adaptive triangulation and the resulting segmentation are shown in figure 22f. It is interesting to note that Case 3 and Case 4 have two corner values of differing sign and require four triangles, while Case 2 has one corner value of differing sign and requires two triangles implying that two triangles are needed for each corner that differs in sign.

Note that one could ignore triangulation altogether simply connecting the two edge points in Case 2 (figure 20b) and in Case 4 (figure 22a), while connecting the points on the opposite sides of the cell in Case 3 (figure 21c). This gives similar answers in each case, although a little less accurate since no extra grid point is determined within the cell.

A.1 Calculating Area

We use the cross product of two vectors to compute the area of triangles, as this is rather robust. For example, consider a triangle with vertices defined by $\vec{x}_a$, $\vec{x}_b$ and $\vec{x}_c$ in counterclockwise order. Then defining $\vec{v}_b = \vec{x}_b - \vec{x}_a$ and $\vec{v}_c = \vec{x}_c - \vec{x}_a$ allows us to define the area as $\frac{\vec{v}_b \times \vec{v}_c}{2}$. Using this definition it is easy to avoid roundoff errors due to thin triangles as they show up as a negative area that can be discarded.

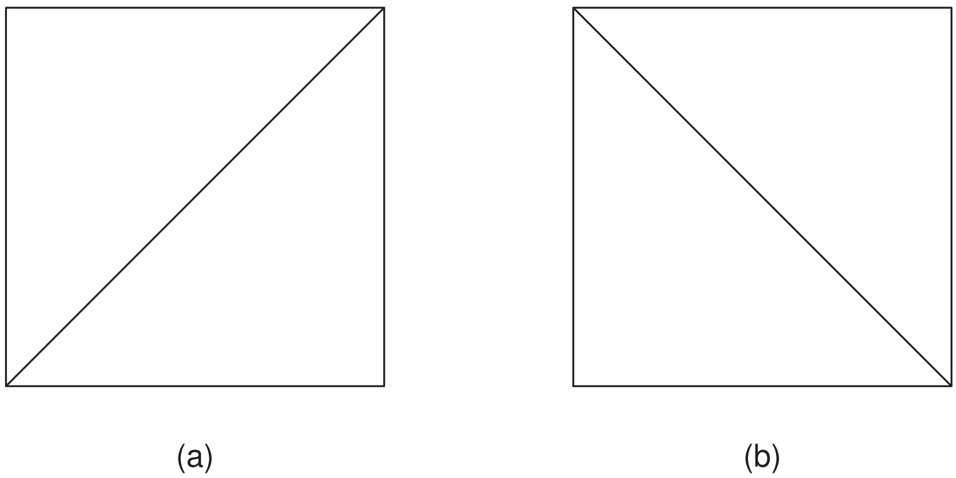

Figure 19: (a) diagonal, (b) off-diagonal.

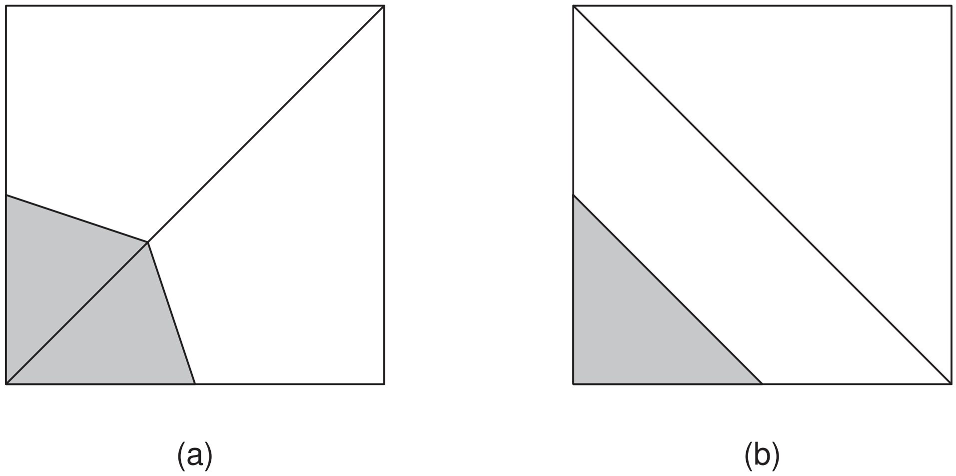

Figure 20: (a) diagonal, (b) off-diagonal.

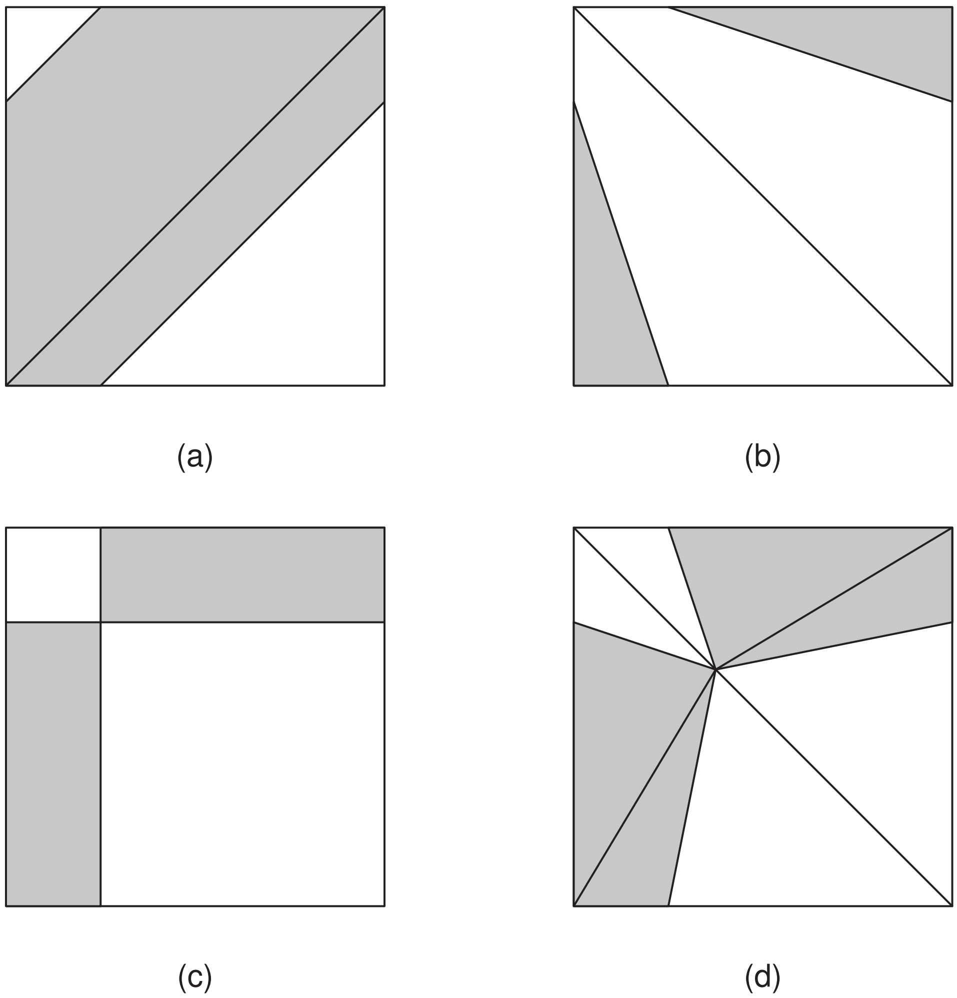

Figure 21: (a) diagonal, (b) off-diagonal, (c) connect opposite edges, (d) adaptive triangulation.

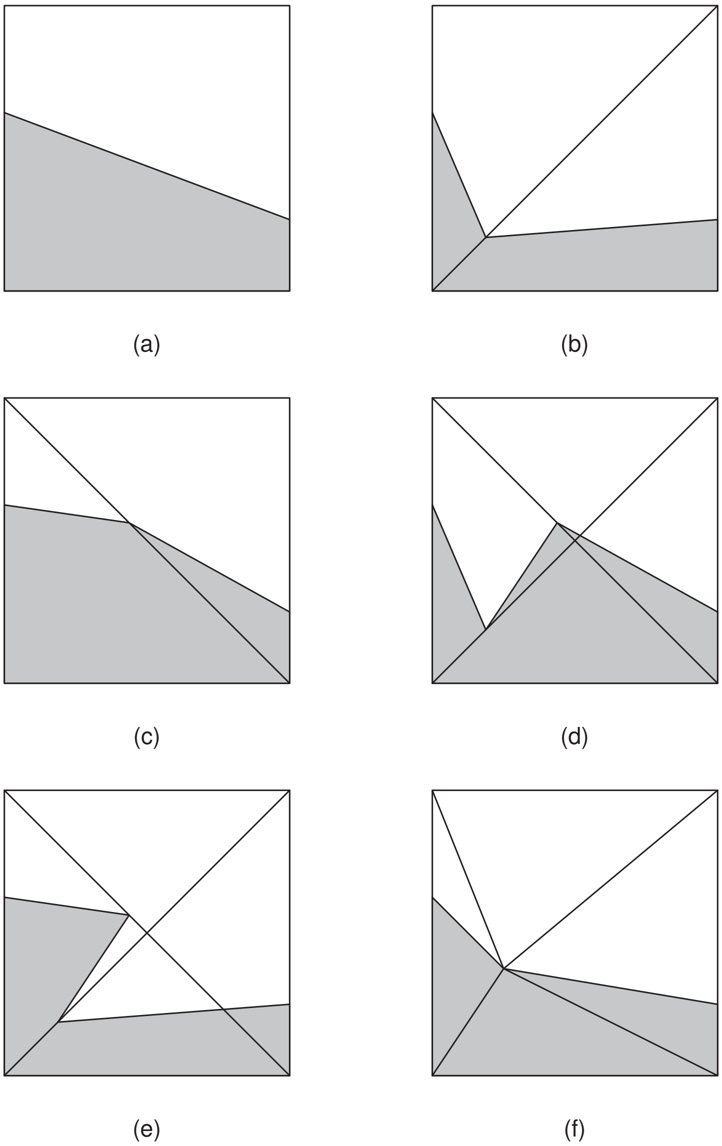

Figure 22: (a) connect opposite edges, (b) diagonal, (c) off-diagonal, (d) both, (e) both, (f) adaptive triangulation.

References

[1] Abgrall, R., *How to Prevent Pressure Oscillations in Multicomponent Flow Calculations: A Quasi Conservative Approach*, Journal of Computational Physics, v. 125, pp. 150-160 (1996).

[2] Adalsteinsson, D. and Sethian, J.A., *The Fast Construction of Extension Velocities in Level Set Methods*, Journal of Computational Physics, vol. 148, pp. 2-22 (1999).

[3] Atkinson, K.E., *Introduction to Numerical Analysis*, 2nd edition, John Wiley and Sons, New York, 1989.

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

[5] Casulli, V. and Greenspan, D., *Pressure Method for the Numerical Solution of Transient, Compressible Fluid Flows*, International J. for Num. Methods in Fluids, v. 4, 1001-1012 (1984).

[6] Chang, Y.C., Hou, T.Y., Merriman, B. and Osher, S., *A Level Set Formulation of Eulerian Interface Capturing Methods for Incompressible Fluid Flows*, J. Comput. Phys., vol. 124, 449-464 (1996).

[7] Chen, S., Johnson, D. and Raad, P., *Velocity Boundary Conditions for the Simulation of Free Surface Fluid Flow*, J. Comput. Phys., vol. 116, pp. 262-276 (1995).

[8] Chen, S., Johnson, D., Raad, P. and Fadda, D., *The Surface Marker and Micro Cell Method*, Int. J. for Num. Methods in Fluids, vol. 25, pp. 749-778 (1997).

[9] Chorin, A.J. *Numerical Solution of the Navier-Stokes Equations*, Math. Comp. 22, 745-762 (1968).

[10] Colella, P. and Pao, K., *A Projection Method for Low Speed Flow*, J. Comput. Phys., vol. 149, pp. 245-269 (1999).

[11] Fedkiw, R., Aslam, T., Merriman, B., and Osher, S., *A Non-Oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (The Ghost Fluid Method)*, J. Computational Physics, vol. 152, n. 2, 457-492 (1999).

[12] Fedkiw, R., Aslam, T., and Xu, S., *The Ghost Fluid Method for deflagration and detonation discontinuities*, J. Computational Physics 154, n. 2, 393-427 (1999).

[13] Fedkiw, R., Merriman, B., Donat, R., and Osher, S., *The Penultimate Scheme for Systems of Conservation Laws: Finite Difference ENO with Marquina's Flux Splitting*, Progress in Numerical Solutions of Partial Differential Equations, Arachon, France, edited by M. Hafez, July 1998.

[14] Fedkiw, R., Merriman, B., and Osher, S., *Numerical methods for a one-dimensional interface separating compressible and incompressible flows*, Barriers and Challenges in Computational Fluid Dynamics, pp 155-194, edited by V, Venkatakrishnan, M. Salas, and S. Chakravarthy, Kluwer Academic Publishers (Norwell,MA), 1998.

[15] Golub, G. and Van Loan, C., *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 1989.

[16] Jiang, G.-S. and Peng, D., *Weighted ENO Schemes for Hamilton Jacobi Equations*, UCLA CAM Report 97-29, June 1997, SIAM J. Num. Analysis (to appear).

[17] Kang, M., Fedkiw, R., and Liu, X.-D., *A Boundary Condition Capturing Method for Multiphase Incompressible Flow*, Journal of Comput. Phys. (submitted).

[18] Karni, S., *Hybrid multifluid algorithms*, SIAM J. Sci. Comput., vol 17 (5), pp. 1019-1039, September 1996.

[19] Karni, S., *Multicomponent Flow Calculations by a Consistent Primitive Algorithm*, Journal of Computational Physics, v. 112, 31-43 (1994).

[20] Klein, R., *Semi-Implicit Extension of a Godunov-Type Scheme Based on Low Mach Number Asymptotics I: One Dimensional Flow*, J. Comput. Phys., v121, pp 213-237, (1995).

[21] Osher, S. and Sethian, J.A., *Fronts Propagating with Curvature Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*, Journal of Comput. Phys., vol. 79, n. 1, pp. 12-49, (1988).

[22] Mulder, W., Osher, S., and Sethian, J.A., *Computing Interface Motion in Compressible Gas Dynamics*, J. Comput. Phys., v. 100, 209-228 (1992).

[23] Patnaik, G., Guirguis, R.H., Boris, J.P. and Oran, E.S., *A Barely Implicit Correction for Flux-Corrected Transport*, Journal of Comput. Phys., vol. 71, pp. 1-20 (1987).

[24] Peyret, R. and Taylor, T.D., *Computational Methods for Fluid Flow*, Springer-Verlag, NY, 1983.

[25] Saurel, R. and Abgrall, R., *A Multiphase Godunov Method for Compressible Multiphase Flows*, J. Comput. Phys., v. 150, n. 2, pp. 425-467 (1999).

[26] Schneider, T., Botta, N., Geratz, K.J. and Klein, R., *Extension of Finite Volume Compressible Flow Solvers to Multi-dimensional, Variable Density Zero Mach Number Flows*, J. Comput. Phys., v155, pp. 248-286, (1999).

[27] Sesterhenn, J., Muller, B. and Thomann, H., On the Cancellation Problem in Calculating Low Mach Number Flows, J. Comput. Phys., vol. 151, pp. 597-615 (1999).

[28] Sethian, J.A., *Fast Marching Methods*, SIAM Review, vol. 41, no. 2, pp. 199-235 (1999).

[29] Shu, C.W. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes*, Journal of Computational Physics, vol. 77, n. 2, pp.439-471(1988).

[30] Shu, C.W. and Osher, S., *Efficient Implementation of Essentially Non-Oscillatory Shock Capturing Schemes II (two)*, Journal of Computational Physics, v. 83, 32-78 (1989).

[31] Shyue, K.-M., *A Fluid-Mixture Type Algorithm for Compressible Multicomponent Flow with Van der Waals Equation of State*, J. Comput. Phys., vol. 156, pp. 43-88 (1999).

[32] Shyue, K.-M., *An Efficient Shock Capturing Algorithm for Compressible Multicomponent Problems*, J. Comput. Phys., vol. 142, pp. 208-242 (1998).

[33] Sussman, M., Smereka, P. and Osher, S., *A level set approach for computing solutions to incompressible two-phase flow*, J. Comput. Phys., vol. 114, pp. 146-154 (1994).

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

[35] Xiao, F., *A Computational Model for Suspended Large Rigid Bodies in 3D Unsteady Viscous Flows*, J. Comput. Phys., vol. 155, pp. 348-379 (1999).

[36] Yabe, T. and Wang, P.-Y., *Unified Numerical Procedure for Compressible and Incompressible Flow*, J. of The Physical Society of Japan, v. 60, n. 7, pp. 2105-2108, July 1991.