Compressible two-phase fluid informatics with subgrid modeling

Mingyu Sun

A general numerical method with subgrid modeling has been developed for the compressible two-phase fluids with arbitrary length scale of interfacial fluid structures. The method provides a framework for the sharp interface method that can resolve an interface with a length scale sufficiently large than the grid size; it can also accommodate the diffused interface modeling such as a two-fluid model. It can be an efficient method for modeling complex multiphase phenomena in which the compressibility is not negligible, such as cavitations, volcano eruptions, and meteor impact in atmosphere.

3.1. Introduction

Bubble growth and collapse are two basic phenomena in gas-liquid two-phase flows, including a boiling liquid, cavitation bubbles induced by a pulsed energy source or a shock wave. A collapsing bubble may eventually reduce its original size to a minute fraction, releasing a shock wave and light. In addition to bubbly flows, the phenomena with different or/and changing length scales prevails in nature, such as the breakup of a water drop, fuel atomization, the evolution of ocean waves, and volcano eruptions. The motivation of this work is to develop an accurate, efficient and robust method for these problems with disparate length scales under general conditions under high pressure and density ratios.

Consider an interface or a particle with the length scale, d, say the diameter of a spherical particle, or equivalent diameter if non-spherical. Compared with the grid spacing that is restricted by computer resources, the length scales in two-phase phenomena can be generally divided to three scales, under-resolved ($d \ll \Delta x$), transitional ($d = O(\Delta x)$) and resolved ($d \gg \Delta x$) scales, as shown in Fig. 3.1. According to the length scale that can be best modeled, numerical methods developed for two-phase flows can be divided to two categories.

The sharp interface or resolved-interface method can deal with flows containing particles sufficiently larger than the grid size. Popular methods under this category include the volume of fluid (VOF, [6,14,31]), the level-set[23,34], among many others (e.g. [9,13,39,40,44]). These methods can resolve the interface sharply, in one or two cells. The VOF method, more precisely, the volume-tracking method, can be formulated conservatively, but the level-set is generally non-conservative. The VOF method encounters difficulties when dealing with particles in the transitional scale [5]. The treatment for subgrid particles is impossible by using up-to-date methods under this category.

For the simulation of under-resolved particles, $d \ll \Delta x$, the diffuse interface methods (two-fluid models) are often used. The reader is referred to [20] for other unresolved-interface methods. These models assume a local mixture, distinguishing at least the volume fraction of two phases in a grid cell. Each phase is assumed to have own pressure and velocity or averaged/relaxed ones. This approximation allows strong numerical simplicity and eliminates the explicit treatment for interfaces. Depending upon the assumption adopted in such a model, the number of governing equations can vary from four to seven [1,2,7,21,30,35] for 1-D flows. In practice, these models are supplemented by instantaneous relaxation or averaging procedures (e.g. [30,35]). This approach can simulate two-phase phenomena with any length scales, but an interface can only be resolved with a fairly wide stencil. An interface separating two pure fluids can never be resolved sharply as it should be.

Consider a control volume or a cell consisting of two phases on a fixed Eulerian grid. It interacts with its neighboring volumes that are occupied by either one or two phases, and simultaneously the two phases inside interact with each other. The former represents the phenomena with length scales of the grid spacing and larger, and the latter represents those with subgrid scales. The former inter-cell interactions are simulated in the Lagrangian frame, and the latter internal or subgrid interactions is solved by a thermodynamic and dynamic subgrid closure model. A general and robust subgrid closure model for two-material cells is proposed. The conservative quantities of the entire cell are apportioned between two materials, and then pressure and velocity are fully or partially equilibrated by modeling subgrid wave interactions. An unconditionally stable and entropy-satisfying solution of the processes has been successfully found. The solution is valid for arbitrary level of relaxation. The model is numerically designed with care for general materials, and is computationally efficient without recourse to subgrid iterations or sub-cycling in time. The model is implemented and tested in the Lagrange-remap framework. The generality, robustness and efficiency of the model make it useful in principle in algorithms, such as arbitrary Lagrangian-Eulerian methods, VOF methods and even some mixture models, for compressible two-phase flow computations.

The modeling of the inter-cell interactions of the model, based on the Largrange-remap method, is detailed in section 3.2.A. The Volume-tracking technique developed for subgrid-scale fluid particles is introduced in section 3.4.

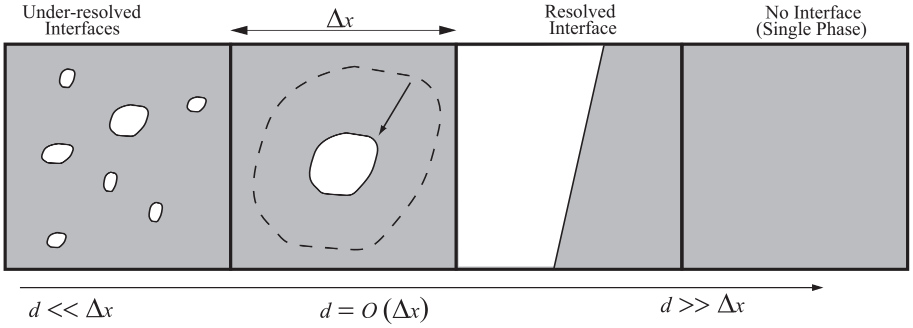

Figure 3.1. The length scale (d) compared with the grid spacing (Δx) in two-phase flows.

The thermodynamic and dynamic subgrid closure model is summazied in section 3.5.

3.2. Lagrange-remap strategy for single phase flows

In the cell-centered framework, each cell represents a Lagrange particle. No material interface will be treated in this step. The change of the conservative quantities of the particle is governed only by the pressure and velocity at its faces. The face pressure and velocity are evaluated from the effective quantities of pressure, density, velocity and sound speed. Only the explicit method is summarized in this section, and the extension to the implicit method is quite straightforward. The Lagrange-remap method consists of two steps:

1. the Lagrange step, in which the change of volume, momentum and energy are updated only from the pressure and velocity at cell faces, to be introduced in section 3.2.A.;
2. the remap step, in which the Lagrange solutions of two phases are projected back to the Eulerian grid conservatively, to be introduced in section 3.2.B.

The subgrid closure modeling is performed for the Lagrange cell, so it is used after the Lagrange step, but before the remap step.

3.2.A. Lagrange step

For a Lagrangian cell, the change of volume in time Δt is given by the face velocity,

$$\chi_v = \Delta t(u^*_{i+1/2} - u^*_{i-1/2}). \tag{3.1}$$

Similarly, the change of momentum and that of energy are given by

$$\chi_m = -\Delta t(p^*_{i+1/2} - p^*_{i-1/2}), \tag{3.2}$$

$$\chi_E = -\Delta t(p^*_{i+1/2}u^*_{i+1/2} - p^*_{i-1/2}u^*_{i-1/2}), \tag{3.3}$$

where $p^*_{i+1/2}$ and $u^*_{i+1/2}$ are pressure and velocity at cell face between left cell i and right cell $i+1$. They are given by the acoustic Riemann solver (3.20) and (3.21). For achieving second-order accuracy in both time and space, the left and right states used in the Riemann solver are advanced by $\Delta t/2$ and linearly extrapolated from the cell center, say for pressure on the left,

$$p^L_{i+1/2} = p_i + \Phi^L_i[(\nabla p)_i\frac{\Delta x}{2} + (p_t)_i\frac{\Delta t}{2}] \tag{3.4}$$

where time derivative $p_t = -\rho a^2 u_x$. The velocity $u_{i+1/2}$ can be obtained similarly by replacing p by u with $u_t = -p_x/\rho$. All quantities and their gradients are calculated from cell centers. Notice that it is the material time derivatives that is used in (3.4). The slope limiter Φ_i is used to suppress possible numerical oscillations near sharp discontinuities. The one-parameter family of minmod limiter is used, for $\theta \in [1,2]$,

$$\Phi_i = \begin{cases} 1 & g_i = 0, \\ \max(0, \min(1, \theta\frac{g_{i-1/2}}{g_i}, \theta\frac{g_{i+1/2}}{g_i})) & g_i \neq 0. \end{cases} \tag{3.5}$$

where g_i is the gradient of the function centered at cell i. The limiter is most dissipative for $\theta = 1$ and is least dissipative for $\theta = 2$. We set $\theta = 1.6$ for the slope limiter in the Lagrange step, Φ^L_i.

Having the total change of volume, momentum and energy of the mixed cell, one may update the state of two phases inside using the subgrid closure model introduced in section 3.5..

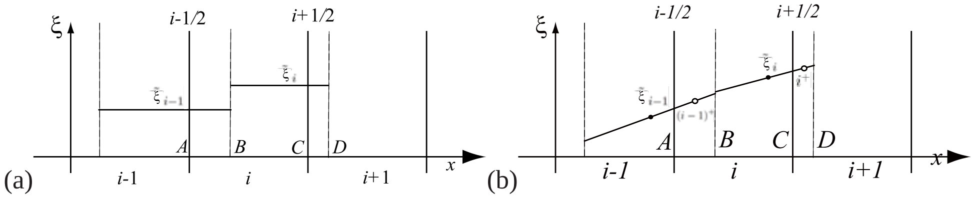

Figure 3.2. The remap step: (a) first order remap; (b) second-order remap. Variable ξ represents a quantity to be remapped.

3.2.B. Remap step

All quantities of each phase updated in the Lagrange step are defined in the Lagrangian frame, and they are projected to the original grid in the remap step in order to start a new cycle.

In the remap step, the conservative quantities in the Eulerian cell, i, are found by remapping the Lagrangian solution. As shown in Fig.3.2, they are the sum of two portions, AB and BC. Supposing the Lagrangian solution is piecewise constant, we get

$$\Omega_i \mathbf{U}_i^{n+1} = (u^*_{i-1/2}\Delta t)\tilde{\mathbf{U}}_{i-1} + (\tilde{\Omega}_i - u^*_{i+1/2}\Delta t)\tilde{\mathbf{U}}_i, \tag{3.6}$$

where $\mathbf{U}_i$ denotes the vector of conservative quantities in cell i, and the variables with tilde denote those of the Lagrange cell. The projection can be reformulated as the finite volume method

$$\Omega_i \mathbf{U}_i^{n+1} = \tilde{\Omega}_i \tilde{\mathbf{U}}_i - \Delta t(u^*_{i+1/2}\mathbf{U}^*_{i+1/2} - u^*_{i-1/2}\mathbf{U}^*_{i-1/2}), \tag{3.7}$$

where, for $u^*_{i+1/2} > 0$,

$$\mathbf{U}^*_{i+1/2} = \tilde{\mathbf{U}}_i. \tag{3.8}$$

The face velocity $u^*_{i+1/2}$ must be the same as that used in (3.1) and (3.3). If the conservative quantities in the remap fluxes are defined from a piecewise constant state as (3.8), it is first order accurate. The second-order accuracy is achieved by performing the linear interpolation [41] from the upstream cell,

$$\mathbf{U}^*_{i+1/2} = \tilde{\mathbf{U}}_i + \mathbf{\Phi}_i^r(\nabla\tilde{\mathbf{U}})_i\Delta_i^+, \tag{3.9}$$

where $\mathbf{\Phi}_i^r$ is the slope limiter used in the remap step, and

$$\Delta_i^+ = \tilde{x}_{i^+} - \tilde{x}_i = (x_{i+1/2} + u\Delta t/2) - (x_i + u\Delta t) = (x_{i+1/2} - x_i) - u\Delta t/2. \tag{3.10}$$

The interpolated value is located at the center between the solid grid interface and the dashed particle interface, denoted by empty circles in Fig. 3.2b. The state at this central point represents the average over the portion, CD. Numerically, it is intensive quantities, $\mathbf{R} = (\rho, u, e)$, that are interpolated. The conservative quantities are then evaluated by,

$$\mathbf{U}^*_{i+1/2} = \tilde{\mathbf{U}}(\tilde{\mathbf{R}}_i + \mathbf{\Phi}_i^r(\nabla\tilde{\mathbf{R}})_i\Delta_i^+), \tag{3.11}$$

which is numerically more robust. The use of non-conservative state quantities neither reduces the order of accuracy, nor violates the conservation laws since the overall conservation is always maintained in (3.7).

The method to enforce the monotonicity of the remap step is similar to the construction of a limiter in the MUSCL approach for hyperbolic equations. We use the same one-parameter minmod slope limiter (3.5) with $\theta = 2$ for $\mathbf{\Phi}_i^r$.

3.3. Lagrange-remap strategy for two-phase flows

3.3.A. The effective state of the two-phase cells

The individual states in the mixed cell have been updated in the subgrid closure model. Most numerical methods adopted in the Lagrange step are designed for cell with one pressure and velocity. The effective state of the two-phase cell must be constructed to be consistent with the subgrid closure.

3.3.A.1. *Density*

There is no ambiguity to define an effective density of a two-phase cell from mass conservation,

$$\tilde{\rho} = \sum_k \alpha_k \rho_k. \tag{3.1}$$

Although it is the unique solution that maintains the conservation of mass, it is not necessarily the most accurate one in all flow situations. For instance, when a shock wave interacts with the mixed cell but travels in only one material, the most accurate solution is that of the material. The effective density may be interpreted as a probability-weighted value of two phase quantities, which holds for velocity and pressure as well.

3.3.A.2. *Velocity*

The velocity can be calculated from the conservation of momentum,

$$M\tilde{u} = \sum_k M_k u_k, \tag{3.2}$$

or

$$\tilde{u} = \sum_k \alpha_k \rho_k u_k / \sum_k \alpha_k \rho_k. \tag{3.3}$$

The effective velocity in the direction normal to an interface is reasonable since the time for velocity relaxation is short[17], which is also seen from the decay rate in (3.24). However, it remains questionable for interfaces having a strong shear discontinuity for multi-dimensional applications.

3.3.A.3. *Pressure*

The effective pressure is defined as

$$\tilde{p} = \sum_k \alpha_k p_k. \tag{3.4}$$

This definition is in consistent with the Gibbs relation for the mixed cell [10].

3.3.A.4. *Sound speed*

An effective sound speed of the mixed cell is necessary for modeling the effect of compressibility. Before deriving the sound speed for the non-equilibrium subgrid closure proposed in last sections, two limit sound speeds are reviewed first. One limit corresponds to the frozen sound speed, where no subgrid interactions or relaxations are taken into account. Knowing constant α_k resulted from equal volumetric strain, one may differentiate the pressure using the equal strain assumption to get

$$\tilde{\rho}\tilde{a}^2 = \sum_k \alpha_k \rho_k a_k^2. \tag{3.5}$$

The assumption of equal volumetric strain itself is clearly incorrect if it is applied to a mixed cell of air and water for instance, as commented by Benson[4]. It generally leads to a pressure change proportional to the square of sound speed.

On the other hand, if pressure equilibrium is assumed, one may get another sound speed same as that developed in homogeneous two-phase flows, known as the Wood's equation [42],

$$\tilde{\rho}\tilde{a}^2 = (\sum_k \frac{\alpha_k}{\rho_k a_k^2})^{-1}. \tag{3.6}$$

It is also called equilibrium sound speed (e.g. [17]).

We shall derive the effective sound speed of the mixed cell, resulting from the partition of external quantities (section 3.5.A.) and the subgrid interactions (section 3.5.B.). It turns out that the new effective sound speed just varies between (3.5) and (3.6), depending upon the level of relaxation. Suppose the mixed cell of pressure $\tilde{p}$ is expanded or compressed by a volume of χ_v. The partition of volume and subgrid interactions lead to the pressure, under isentropic assumption,

$$\tilde{p}^* = \sum_k \alpha_k^* p_k^* = (\alpha_l + \frac{\delta\Omega_l}{\Omega})[p_l - \frac{\rho_l a_l^2}{\Omega_l}(\chi_{vl} + \delta\Omega_l)] + (\alpha_r - \frac{\delta\Omega_l}{\Omega})[p_r - \frac{\rho_r a_r^2}{\Omega_r}(\chi_{vr} - \delta\Omega_l)]. \tag{3.7}$$

The subgrid closure model is actually entropy-nondecreasing. Since the entropy change of individual phase is of second order of smallness, to be shown in section 3.5.C., the terms of pressure dependence on the entropy have been neglected. The pressure change is then, after some manipulations

$$\delta\tilde{p} = \tilde{p}^* - \tilde{p} = \frac{\chi_v}{\Omega}[(\sum_k \alpha_k \rho_k a_k^2) - \beta\frac{(\rho_l a_l^2 - \rho_r a_r^2)^2}{\rho_l a_l^2/\alpha_l + \rho_r a_r^2/\alpha_r}], \tag{3.8}$$

in which high order terms have been neglected. Using $\delta\tilde{p} = \frac{\tilde{\rho}\tilde{a}^2}{\Omega}\chi_v$, one gets

$$\tilde{\rho}\tilde{a}^2 = (\sum_k \alpha_k \rho_k a_k^2) - \beta\frac{(\rho_l a_l^2 - \rho_r a_r^2)^2}{\rho_l a_l^2/\alpha_l + \rho_r a_r^2/\alpha_r}. \tag{3.9}$$

The parameter β results from the finite decay of pressure difference in (3.29), for $t = \Delta t$,

$$\beta = 1 - e^{-\frac{2N\bar{c}}{\rho_l a_l + \rho_r a_r}\Delta t}. \tag{3.10}$$

The parameter $\beta \in [0, 1]$, or $N \in [0, +\infty)$, characterizes the level of relaxation. For $\beta = 0$ or $N = 0$, the cell is frozen without relaxtion, and (3.9) is just the frozen sound speed (3.5). For $\beta = 1$ or $N = +\infty$, the cell is fully relaxed. The effective sound speed (3.9) takes its minimum value; this minimum value is exactly the equilibrium sound speed (3.6).

3.4. Volume-tracking of subgrid-scale fluid particles

In VOF, the interface in a cell is not tracked explicitly, but reconstructed and approximated by a simple geometry. The simple line (piecewise constant) interface calculation (SLIC) [22] assumes the geometry is a line parallel to one of the grid lines. Currently most widely used method is based on the piecewise linear interface calculation (PLIC). A historic review of the piecewise constant and linear reconstructions can be found in [28,29]. In PLIC, the surface normal vector, $\mathbf{n}$, is required to construct the linear interface

$$\mathbf{n} \cdot \mathbf{r} + h = 0, \tag{3.1}$$

where h is the line constant. The normal vector is inferred from the spatial distribution of ϕ. If ϕ is a smooth function, the normal vector satisfies

$$\mathbf{n} = \nabla \phi. \tag{3.2}$$

A few numerical methods have been proposed for the calculation of the surface normal. Youngs [46] used the finite-difference method to discretize the gradient in (3.2) directly. Puckett [26] approximated the surface normal vector from the volume fraction in a 3×3 block of cells using an iterative method. The efficiency of the iterative method was improved by Pilliod using an algorithm, ELVIRA, which can reconstruct all linear interfaces exactly [25]. Scardovelli and Zaleski [33] described these methods in concise formulas, and presented two least-square fit techniques. The reconstruction using a spline interface was attempted by López et al. [19]. The volume fraction is advanced from the geometry of the reconstructed interface, by multidimensional/unsplit schemes or one-dimensional/operator-split schemes. Continuous efforts have been made to improve the advection algorithms [11,28,33,46].

It is quite well known that the VOF method will be accurate whenever the radius of curvature is large with respect to the grid size. It will be less accurate for scales comparable, and may lose all details for scales smaller than one grid spacing. Consider fluid structures with the characteristic length, d, say, the thickness of a thin filament, or the diameter of a circular particle. If d reduces to the order of the grid size, Δx, the VOF method numerically splits or merges them. It is called numerical surface tension by Rider and Kothe [28]. Černe et al. [5] investigated the behavior of the VOF method in simulating these small interface structures, and found that the error of the interface reconstruction increases rapidly when $d/\Delta x \leq 3$. This is in agreement with the fact that a minimum of three grid cells are required to resolve a circular particle with certain accuracy. They also found that the advection errors occur as well. The small particle moves faster for $d/\Delta x < 2$. Subgrid particles ($d/\Delta x < 1$) were not considered in their study.

For resolving subgrid particles, there are two difficulties behind the PLIC-VOF method. Consider an isolated subgrid particle in a cell, as illustrated in Fig. 3.3a. The subgrid particle generally stays in the cell, surrounded by the dark phase in neighboring cells. The first difficulty is that the gradient of volume fraction (3.2) cannot provide a meaningful solution. How to update the volume fraction is another difficulty. The simulation on the finer grid can resolve this problem; however it increases the computational cost. In order to enhance the accuracy of VOF at low grid resolution, more information on the interface in addition to the volume fraction is necessary. López et al. attempted to resolve thin filaments by using makers [18]. Zhang & Liu represented particles explicitly as piecewise polygons, and calculated new material areas inside interface cells via polygon-clippings in a discrete manner [47]. Dyadechko & Shashkov proposed the moment-of-fluid method that keeps track of the cell-wise material centroids, in addition to the volumes [8]. All these methods improve the accuracy in resolving interfaces with large curvatures or fine structures. However, none of these methods have shown the capability to track and advect a subgrid-scale particle, and the implementation of them in an existing PLIC-VOF code is not straightforward.

This work tries to improve the accuracy of the PLIC-VOF method at low grid resolution, and to expand its capability in dealing with subgrid particles [36]. The idea is to treat the surface normal vector as independent variables that are integrated along with the advection algorithm, instead of calculating them from (3.2). The equations for the surface normal vector imposed with the unit normal constraint ($|\mathbf{n}| = 1$) were tested by Raessi et al [27], and accurate surface curvatures for resolvable circular interfaces have been obtained. In this work, the treatment for surface normals has been developed for resolving subgrid particles. Non-conservative equations for surface normals are proposed and generally tested in this work, and also the unit normal constraint is not imposed.

3.4.A. The surface normal and the motion of a small particle

We shall consider the mathematical relationship between the surface normal and the motion of a particle in this section. Let domain D occupied by a resolvable or a subgrid particle be a simply connected domain bounded by the surface $\Gamma : \phi = c$. For the scalar field ϕ that is smooth enough to define $\mathbf{n} = \nabla\phi$, we have, using the Stokes' theorem,

$$\int_D \mathbf{n} dA = 0. \tag{3.3}$$

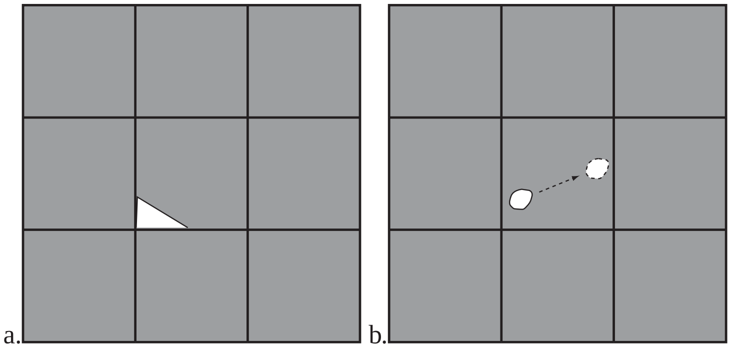

Figure 3.3. The difficulty in advecting a subgrid particle by the volume-tracking method alone. The PLIC-VOF method may reconstruct a subgrid particle, given the phase volume and the normal vector. However, the normal vector cannot be properly calculated from (3.2) only from the distribution of volume fraction for the isolated particle. Even if a vector is assigned as shown in (a) by certain rule based on the volume fraction alone, the particle will be trapped in the cell. Consider a moving particle as shown in (b). Although the particle has traveled a short distance within the cell, the PLIC-VOF method will reconstruct the same linear segment in the cell because the volume fraction and thus its distribution are unchanged.

In order to characterize the size of a deformable particle, we introduce length scale, d, such that the surface of the particle can be bounded by a circle with the minimum diameter d. Let the normal vector at the centroid of the particle be $\mathbf{n}_c$, we have

Theorem 1: The surface normal at the centroid of a particle is of zero magnitude within the error of $O(d^{\,2})$,

$$\mathbf{n}_c = \mathbf{0} + O(d^2).$$

Proof: Approximating $\mathbf{n}$ by the Taylor series expansion at the centroid gives

$$\int_D \mathbf{n} dA = \int_D [\mathbf{n}_c + (\nabla \mathbf{n}) \cdot (\mathbf{r} - \mathbf{r}_c) + O(d^2)] dA = 0, \tag{3.4}$$

where $O(d^2)$ represents omitted second and higher order terms. Using the definition for the centroid, $\int_D (\nabla \mathbf{n}) \cdot (\mathbf{r} - \mathbf{r}_c) dA = 0$, gives

$$\int_D [\mathbf{n}_c + O(d^2)] dA = 0.$$

This completes the proof.

This theorem implies that whatever the initial shape of a particle is, if it is shrunk to be very small ($d \to 0$), the surface normal vector at its mass center will be reduced to the zero vector as well. If velocity $\mathbf{u}$ is that of fluid particles, we have

Theorem 2: The zero surface normal moves at the particle velocity of $\mathbf{u}$.

Proof: Consider a circular fluid particle with its centroid moving at the speed of $\mathbf{u}$. Since the zero surface normal is located at the centroid within the error of $O(d^2)$, the zero vector moves at a speed of $\mathbf{u} + O(d^2)$, and then the speed converges to $\mathbf{u}$ for the infinitely small particle $(d \to 0)$.

A surface normal other than the zero vector generally does not move at the speed of $\mathbf{u}$. In addition to the translational motion, the surface normal vector undergoes rotation, compression or expansion, which actually represents the deformation of surfaces. This theorem can be seen from the surface normal equations as well. In the neighborhood of the zero vector, for $l \to 0$ and $m \to 0$, the SN equation equations become, by omitting terms with l and m,

$$\begin{aligned} l_t &\quad + u l_x + v l_y &&= 0, \\ m_t &\quad + u m_x + v m_y &&= 0, \end{aligned} \tag{3.5}$$

which show that both surface normals move at the speed of (u, v). It is emphasized that the conservative equations, however, lead to

$$\begin{aligned} l_t &\quad + u l_x + v m_x &&= 0, \\ m_t &\quad + u l_y + v m_y &&= 0. \end{aligned} \tag{3.6}$$

In short, for a small particle of $O(\Delta x)$, *the centroid of the particle moves at the same velocity as the surface normal vector of zero magnitude, to second-order accuracy*. The zero vector is a ’marker’, not tracked but evolved by the partial differential equations.

In the work of Raessi et al.[27], the normal vector is imposed with the unit normal constraint. Although the normalization of a surface normal vector does not change the orientation and the curvature of an interface, it destroys the solution in the neighborhood of the vector of zero magnitude and creates one or more discontinuities there. For example, the unit vectors point to arbitrary directions at the center of a circle, where two surface normal components are discontinuous. More importantly, a continuous normal vector is required to maintain a small particle moving at the right flow velocity. Relation (3.4) is valid only for a smooth function of $\mathbf{n}$. The information on the vector of zero magnitude will be destroyed by the unit normal constraint. In this work, two surface normal components are solved as they are during the course of integrating the equations.

Coupling the VOF method with the surface normal equations, the present volume tracking method is divided to three procedures,

(a) to reconstruct the piecewise linear interface with a given surface normal in each interface cell such that the interface truncates the cell with a fractional volume equaling the given phase volume in the cell (section 3.4.B.);

(b) to advance the phase volumes using a unsplit algorithm * with a limiter ensuring that all phase volumes lie within bounds of $[0, \Omega]$ (section 3.4.C.), where Ω is the cell volume;

(c) to advance the surface normal vectors (section 4.3).

Procedures (a) and (b) are nothing but a typical PLIC-VOF method except replacing the calculation of surface normal by the surface normal vector obtained from (c). The surface normal vector is integrated by the finite volume method, to be discussed in section 3.4.D.. Since the method combines the PLIC method with the surface normal (SN) equations, we shall refer to the present method as PLIC/SN in what follows.

The discretization procedures to be discussed are so optimized and formulated that they can be readily implemented on any grid system and coupled with a finite-volume flow solver. The only input required from the flow solver is the velocity field. All formulas and discussion are valid for structured and unstructured grids, except for the interface reconstruction procedure that is valid only for rectangular cells. The whole algorithm has been developed on a solution-adaptive unstructured quadrilateral grid[38], and coupled with a compressible flow solver. Only the results on the Cartesian grid with specified velocity fields are reported in this paper for the purpose of comparison and evaluation.

3.4.B. Representation of resolvable and subgrid particles

The present method to resolve different sized particles is illustrated in Fig. 3.4, which contains a circular particle with diameter d varied from $1/2\Delta x$ to $4\Delta x$. The interface representation is the same as what has been developed in the PLIC-VOF methods. The only difference is that the surface normal vector has been defined and updated by solving the equations, so that surface normal calculations are not necessary. With the surface normal vector defined in the cell, any sized particles can be unambiguously defined.

An interface is discretized by linear segments in the cells that intersect with the interface. Each linear segment, starting from and ending at the cell edges, divides the interface cell to two portions having the exact volume of two phases. Two linear segments defined in neighboring cells are in general not connected as shown in Fig. 3.4d, so that the segments themselves are not sufficient to shape a closed interface for a particle. The dark particle is actually bounded by the segments inside the cells together with the wet portion of edges between cells. The center of the subgrid circle shown in Fig.3.4a is located at the grid node, so that the subgrid circle is represented as a symmetric diamond. In general, a subgrid particle is represented by disconnected segments, and its shape changes with the location of its center (see more examples in section 5, Fig. 3.6).

Given the surface normal vector $\mathbf{n}$ and the phase volume, the piecewise linear interface (3.1) is unique in a cell. In general, the line constant h needs to be found by iterative method, because the truncation volume is often a nonlinear function of h[28], especially for axisymmetric and 3-D geometries. For 2-D rectangular cells, the non-iterative method is followed to find h[32].

In short, in order to represent subgrid as well as large particles for any grid system in a simple way, what we follow is no more than

1. *at most one linear material interface is located in a control volume;*
2. *the material interface inside the volume and the material interfaces aligned along volume faces may form a subgrid particle, as shown in Fig.1a.*

For rule (2), no special treatment is really needed for a PLIC algorithm. The material interface aligned along grid lines appears as a natural result of PLIC. It is listed here because this sort of subgrid particles, which are unable to be properly handled in a traditional PLIC/VOF method, might have been merged or deleted.

---

[*] The dimension splitting algorithm necessitates the splitting of the equations for the normal vector as well. The computational cost is nearly doubled for unstructured grids. Therefore, in this work, only the unsplit version is investigated.

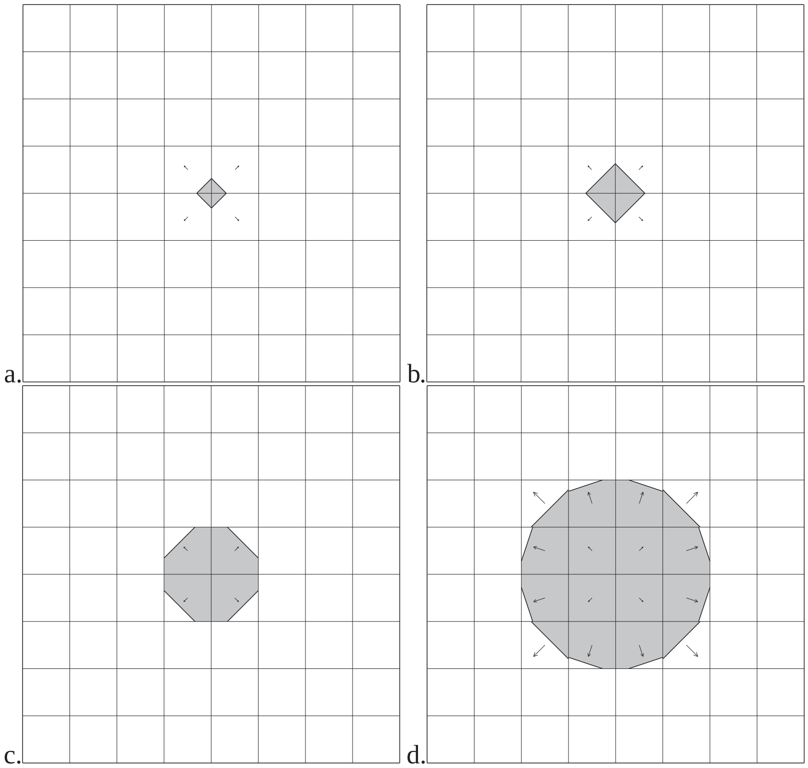

Figure 3.4. Representation of different sized circular particles: a) $d/\Delta x = 1/2$**; b)** $d/\Delta x = 1$**; c)** $d/\Delta x = 2$**;** $d/\Delta x = 4$**.**

No additional restrictions such as the size of a particle, the distance between particles are imposed in the present interface representation. The topological changes of a particle are implicitly handled as most VOF methods. If a particle is divided into two portions with a distance between greater than one grid spacing, it is regarded as the particle breakup, or if the sides of two particles meet in the same cell, they are merged automatically. For resolving two immiscible particles, the distance between them should be large enough to avoid numerical coalescence.

3.4.C. Advection of phase volumes

The advection equation is solved by the finite volume method. The volume flux is evaluated in the direction normal to the grid line, and integrated in a unsplit manner. Given a flow field $\mathbf{u}$, the advection equation is rewritten as

$$\phi_t + \nabla \cdot (\mathbf{u}\phi) = \phi \nabla \cdot \mathbf{u}, \tag{3.7}$$

and

$$\phi_t + \nabla \cdot (\mathbf{u}\phi) = 0, \tag{3.8}$$

for incompressible velocity field. Equation (3.8) is discretized as,

$$\Omega_k^{n+1} = \Omega_k^n - \sum_j (u_n \phi_k \Delta t)_j, \tag{3.9}$$

where u_n is the outward normal velocity across grid interface j, and ϕ_k is the volume fraction of phase k defined at the grid interface, satisfying $\sum_k \phi_k = 1$. Ω_k is the volume of phase k, $\Omega_k = \phi_k \Omega$, satisfying

$$\Omega = \sum_k \Omega_k. \tag{3.10}$$

In order to avoid volume Ω_s^{n+1} going beyond the bound, a simple limiter is imposed on the outflow volume flux with $u_n > 0$,

$$\phi_s^* = \Phi \phi_s. \tag{3.11}$$

It is readily seen that Ω_s^{n+1} is always non-negative with the limiter function. If the volume change is below $\Omega/2$, which is true for the CFL number below 1/2, the slave volume starting from $\Omega_s^n \le \Omega/2$ satisfies $\Omega_s^{n+1} \le \Omega$. The limiter function guarantees the exclusion of the overshoot and undershoot for both phases from numerical results.

It is noted that the Φ limiter is different from the redistribution algorithm adopted in some VOF algorithms. The redistribution algorithm is often triggered when an abnormal volume that is beyond $[0, \Omega]$ is found in the solution, and then redistribute the volume to somehow empirically chosen cells nearby. The Φ limiter is to adjust the volume flux such that the abnormal volume will not appear. This limiting method is simple and general for any grid system, although it cannot replace the use of a more accurate procedure to calculate volume fluxes in the VOF advection algorithm.

Once the interface has been constructed, as discussed in section 4.1, the volume flux $(u_n \phi_s \Delta t)_j$ at grid line j is defined *geometrically* by calculating the volume or the area cut by the grid line shifted upstream by $-u_n \Delta t$, where u_n is the normal velocity defined in the upstream cell. When coupled with a compressible flow solver, the velocity can be defined by the Riemann solver. In this work, since the exact velocity field is specified in all test cases, u_n is simply taken as the exact solution located at the center of the grid line. The method to evaluate volume fluxes is 'naive unsplit'[28]. A modification is made following the DDR method[11]. The DDR method is adopted because of its easy implementation on our finite volume unstructured solver, and it does behave better than the naive unsplit version.

The minimum particle that can be resolved is restricted by the machine accuracy in evaluating volumes and volume fluxes, and in practice the small volumes satisfying

$$\Omega_s/\Omega \leq 4 \times 10^{-11} \tag{3.12}$$

for double precision floating-point computations are removed, and filled with the other phase.

3.4.D. Discretization of surface normal equations

Both conservative form and non-conservative form of the SN equations have been implemented and investigated. For the sake of clarity, they are rewritten as

$$\mathbf{n}_t + (\mathbf{F}_1)_x + (\mathbf{F}_2)_y = \mathbf{S}, \tag{3.13}$$

where for the conservative form

$$\mathbf{F}_1 = \begin{pmatrix} ul + vm \\ 0 \end{pmatrix} \quad \mathbf{F}_2 = \begin{pmatrix} 0 \\ ul + vm \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{3.14}$$

and for the non-conservative form

$$\mathbf{F}_1 = \begin{pmatrix} ul \\ um \end{pmatrix} \quad \mathbf{F}_2 = \begin{pmatrix} vl \\ vm \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} -mv_x + lv_y \\ mu_x - lu_y \end{pmatrix}. \tag{3.15}$$

System (3.13) is discretized by the finite volume method. Consider a control volume Ω_i bounded by discrete faces with outward surface normal $\mathbf{s} = (s_x, s_y)$.

$$(\mathbf{n}_i)_t = \mathbf{S}_i - \frac{1}{\Omega_i} \sum_j \hat{\mathbf{F}}_j,$$

where the numerical flux $\hat{\mathbf{F}}_j$ is approximated by the upwind scheme, depending on the direction of normal velocity $u_n = \mathbf{u} \cdot \mathbf{s}$,

$$\hat{\mathbf{F}}_j = \begin{cases} \mathbf{F}_1^- s_x + \mathbf{F}_2^- s_y, & \text{if } \ u_n > 0; \\ \mathbf{F}_1^+ s_x + \mathbf{F}_2^+ s_y, & \text{otherwise.} \end{cases} \tag{3.16}$$

Velocity and surface normal at faces are required to define the numerical fluxes.

3.4.D.1. *First-order scheme*

For the first-order scheme in space, velocity and surface normal used to determine the numerical flux are simply those located in the control volume on the upwind side. The source terms are discretized using the central difference scheme.

3.4.D.2. *Second-order scheme*

For achieving second-order accuracy in space, the surface normal and velocity at grid interfaces are located at the center of face, $\mathbf{r}_j^c$. They are interpolated from the center of the volume, following the MUSCL method,

$$M^{-,+} = M_i^{-,+} + (\nabla M)_i^{-,+} \cdot (\mathbf{r}^c - \mathbf{r}_i^{-,+}),$$

where M represents both the velocity and surface normal required to define the numerical flux. Superscripts $^{-,+}$ indicates the values are defined from left and right sides (or upstream and downstream) respectively. For example,

$M_i^{-,+}$ are those values located at $\mathbf{r}_i^{-,+}$ respectively. $(\mathbf{r}_j^c - \mathbf{r}_i^{-,+})$ is the distance between the center of the grid interface and the location of values defined. The method is general for both cell-centered and cell-vertex data structures.

In solving the hyperbolic conservation laws, the limiter is often used to suppress possible numerical oscillations around discontinuities. To investigate the influence of the limiter on the solution of surface normal equations, the MINMOD slope limiter for limiting the gradient $(\nabla M)_i^{-,+}$ is also implemented. The source terms are discretized using the central difference scheme.

For achieving second-order accuracy in time, the two-step Runge-Kutta method is followed for the surface normal equations. However, the interface reconstruction and the volume flux evaluation, described in sections 4.1 and 4.2, are done only once based on the surface normal at the last time step.

3.4.D.3. *Initial and boundary conditions*

For solving the IBV problems of surface normal equations, appropriate initial and boundary conditions must be specified for l and m. For a given interface, only the direction of surface normal vector is available nearby. Others have to be defined. An intuitive way to define these values is to construct a smooth surface function ϕ, such that $\phi = c$ represents the interface. Two types of particles, circle and square, are tested in this paper, and their definitions are summarized as below. For a circle centered at (x_0, y_0), we set

$$\phi(x,y) = \frac{1}{2}[(x-x_0)^2 + (y-y_0)^2],$$

and then differentiate it,

$$\begin{aligned} l(x,y) &= x - x_0, \\ m(x,y) &= y - y_0, \end{aligned} \tag{3.17}$$

which are the initial conditions for a circle. Similarly consider a square centered at (x_0, y_0) with sides parallel to grid lines, represented by

$$\phi(x,y) = \frac{1}{2}\begin{cases} (y-y_0)^2 & |y-y_0| > |x-x_0|; \\ (x-x_0)^2 & |y-y_0| < |x-x_0|; \\ \frac{1}{2}[(x-x_0)^2 + (y-y_0)^2] & |y-y_0| = |x-x_0|, \end{cases}$$

one gets two surface normals for squares,

$$l(x,y) = \begin{cases} 0 & |y-y_0| > |x-x_0| \\ x-x_0 & |y-y_0| < |x-x_0| \\ \frac{1}{2}(x-x_0) & |y-y_0| = |x-x_0|, \end{cases} \tag{3.18}$$

and

$$m(x,y) = \begin{cases} y-y_0 & |y-y_0| > |x-x_0| \\ 0 & |y-y_0| < |x-x_0| \\ \frac{1}{2}(y-y_0) & |y-y_0| = |x-x_0|. \end{cases} \tag{3.19}$$

Notice that the definitions are independent of the size of the circle or square, and it suggests the surface normals for any sized particles are solved equally.

Physical boundary conditions for surface normal vector of a real material interface attached on a wall depend on many factors, especially surface tension and gravity, which is beyond the scope of this paper. We shall focus on the interfaces inside the domain. In practice, for wall, inlet and outlet boundaries, we set

$$\frac{\partial l}{\partial s} = \frac{\partial m}{\partial s} = 0, \tag{3.20}$$

in the direction normal to the boundary.

3.4.E. The advection of a subgrid particle

It has been shown that the zero normal vector moves at fluid velocity in section 3. We will investigate how a subgrid particle can follow the zero vector using the PLIC/SN method.

Consider a 1-D subgrid particle as shown in Fig. 3.5. Without losing generality, it is assumed that the surface normal points from the bright to the dark phase. Suppose the particle is initially located at face A, $x = x_0$. We use a linear function

$$l(x) = x - x_0$$

with zero exactly located at $x = x_0$ as the initial condition for surface normal vector. The averaged surface normal over the cell bounded by faces A and B is then positive, as indicated by the dashed line in the top-right figure. Given the volume of the bright particle and the positive surface normal, the interface is then reconstructed on the left following

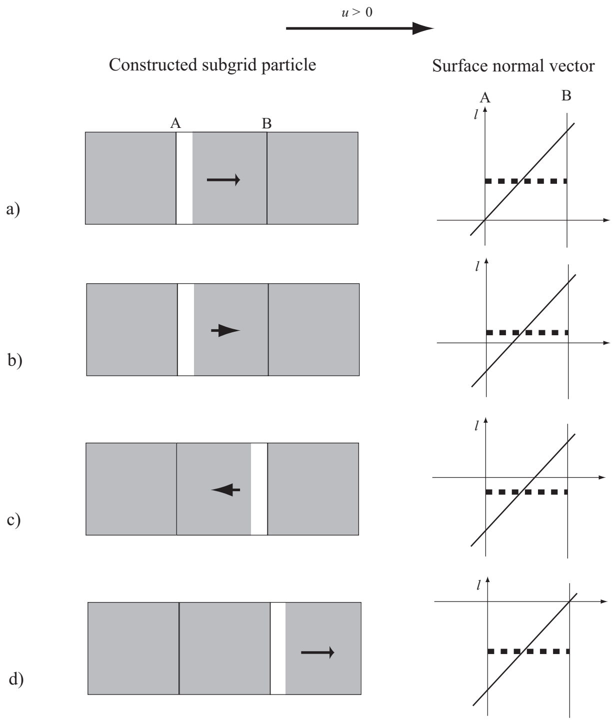

Figure 3.5. A schematic of a 1-D subgrid particle motion controlled by the surface normal vector. The arrow always points from the bright phase to the dark phase in the left column, and its direction is determined by the sign of the averaged surface normal over the cell, the value of which is shown by the dashed line in the right.

the PLIC, as shown in Fig. 3.5a. Notice that the commonly used surface normal calculation for this small subgrid particle is senseless.

Now consider the motion of the particle. Suppose the flow moves at a constant speed to right, traveling through one cell in three steps. The evolution of the surface normal is shown in the right column. It starts with a positive value, and gradually reduces to be negative after the zero vector moves across the central point of the cell. After three steps, the zero vector has traveled from face A to face B, at the same speed as the flow. The corresponding reconstructed interface is shown in the left column. In the first step, from Fig. 3.5a to b, since the subgrid particle is attached on the face A, there is no volume flux through face B, thus the volume of the particle is unchanged. At this moment, the surface normal is still positive, so that the reconstructed interface is the same. Once the surface normal changes its sign, as seen from Fig. 3.5b to c, the particle is reconstructed on the right. The particle leaps from one side to the other side of the cell. In the third step, the particle is moved to the right neighboring cell by the advection algorithm, returning to the initial state but shifted by one grid cell. The procedure above will be repeated.

It is clear that the motion of a subgrid particle is made possible by both the leap controlled by the PLIC/SN and the volume advection through faces. The motion of a subgrid particle in two dimensions is more interesting and complicated. Fig. 3.6 shows all sequential steps of a particle of $d/\Delta x = 0.4$ moving at velocity $(u, v) = (1.0, 0.5)$. For this illustration, the CFL number is taken as 1/8 based only on x-velocity, so that the particle is supposed to move from one node to the other in 16 steps, traveling two grid cells in x-direction, and one cell in y-direction. In most steps, the particle appears as a combination of small pieces in 2-4 cells. Although these small pieces are reconstructed separately by PLIC algorithm in each cell, they are always connected without splitting. It is emphasized that the translation is realized simply by a typical PLIC algorithm with the surface normal defined in the cell. No additional treatment is necessary.

Using the PLIC/SN method, for 1-D geometry the subgrid particle actually stays on the upstream side of the cell before the normal vector is reversed, which happens exactly after the arrival of the zero vector. The leap motion in the cell and the volume advection through faces are not coupled. It is different for a general 2-D geometry. Although the sign of l is reversed between Fig. 3.6d and f, the orientation and location of the reconstructed interface are evolved gradually as shown from Fig. 3.6b to f. The volume flux through the right face is not zero in Fig. 3.6e, so that a small portion is advected to the right cell as seen in Fig. 3.6f. The same is seen in y direction from Fig. 3.6h to j when the sign of m is reversed. As a result of this coupling between the reconstruction and the volume advection, the subgrid particle is located in favor of the downstream sides, which is clearly seen in Figs. 3.6p and q. As interpreted by Černe et al.[5], using a typical PLIC-VOF method, the location error will be accumulated, resulting in the particle moving faster.

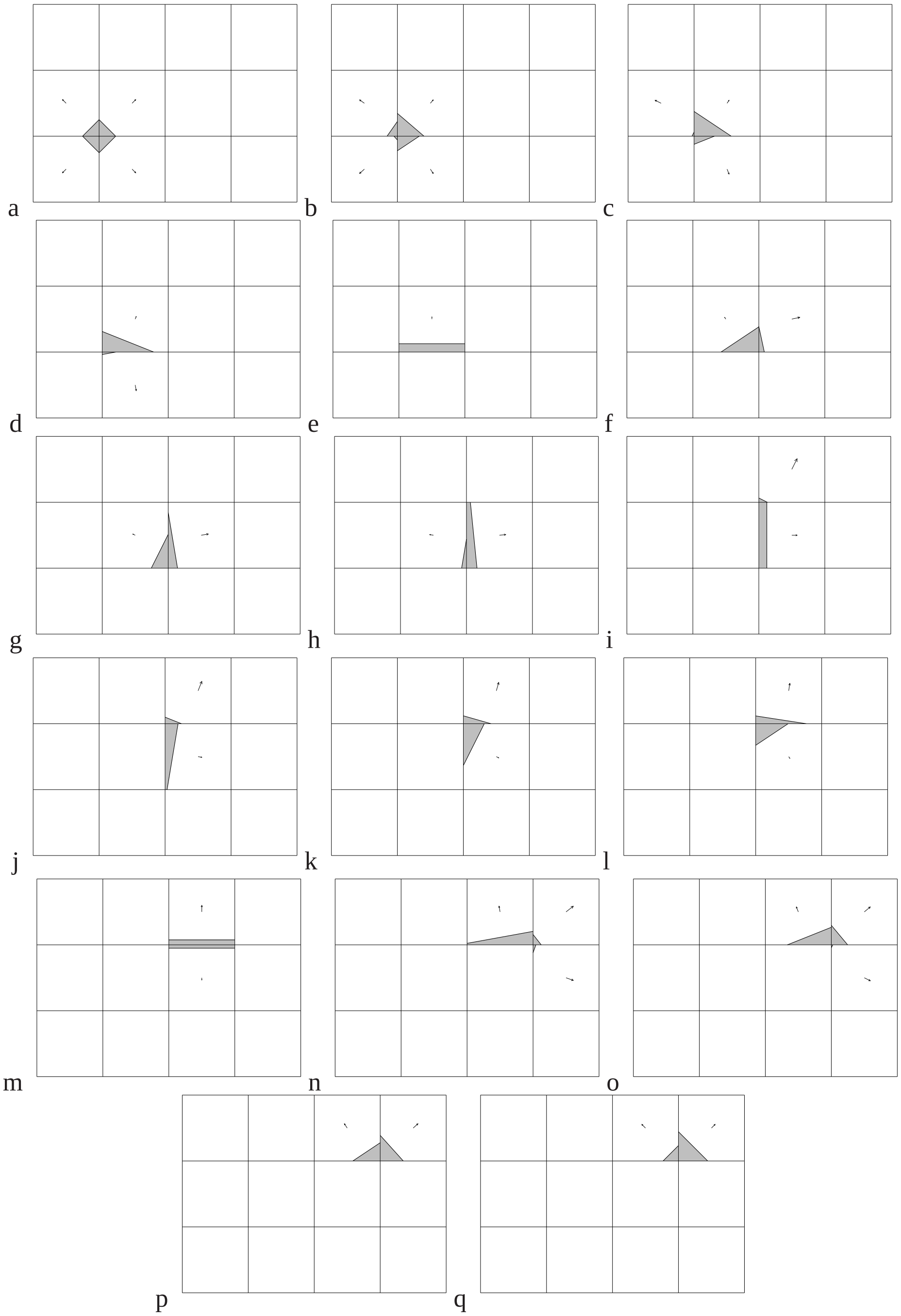

Figure 3.6. Translation of a 2-D subgrid particle of $d/\Delta x = 0.4$ in the velocity field ($u = 1, v = 0.5$), all sequential steps. The particle is supposed to move from one node to the other node in 16 steps, traveling two grid cells in x-direction, and one cell in y-direction.

By analyzing numerical results, the location error of a subgrid particle can be estimated as,

$$\epsilon_{\max} = \pm\Delta x/2 + d + O(\Delta x^2), \tag{3.21}$$

where three errors are originated from the interface reconstruction, the advection and the surface normal equations respectively. The first is the leap error. The location of a particle within the cell is solely determined by the PLIC together with the normal vector. The subgrid particle is reconstructed either on the left or the right side depending upon the location of zero vector. If the zero vector represents the exact location of the particle, the location error originated from this reconstruction procedure is no more than $\Delta x/2$, which is the first term. The error originated from the advection algorithm will depend on the algorithm used and the location of interface, so d is taken as the simple estimate of the maximum error, if it is smaller than the grid spacing. The numerical error in integrating the surface normal equations is assumed to be second order accurate. The deviation of the exact center of the particle from the zero vector is $O(d^2)$ as discussed in section 3.4.A., which is smaller than $O(\Delta x^2)$ for a subgrid particle. The last term $O(\Delta x^2)$ represents these two errors. All numerical results above show that the location errors in two dimensions are within the bound (3.21).

It is noted that a more precise location of the subgrid particle can be reconstructed in the cell, by taking into account the location of the zero vector that can be obtained from the distribution of the normal vector. However, this necessitates non-trivial modifications in PLIC and advection algorithms.

3.5. Thermodynamic and dynamic subgrid closure model for two-phase cells

Consider a control volume or a cell consisting of two phases. Individual quantities in the cell, such as velocity, density and pressure, are initially specified or updated in the last step; they are in either equilibrium or disequilibrium of velocity and pressure. The entire cell undergoes the total change of momentum, energy and volume in a finite time Δt after interacting with other cells. We suppose that the total change has been calculated by any Lagrange scheme that treats the entire cell as a whole. At the same time, momentum, energy and volume in the cell are transferred from one phase to the other if they are in disequilibrium. The present subgrid closure model for these physical processes contains three components[37]:

1. to distribute the total change of momentum, volume and energy to two phases (section 3.5.A.),
2. to find a partially or fully relaxed state of pressure and velocity, such that the cell may eventually approach pressure and velocity equilibrium between two phases (section 3.5.B.),
3. to construct an effective state of the mixed cell that is used to advance the Lagrange step (section 3.3.A.).

The closure model faithfully follows the conservation of volume, mass, momentum and energy, and it satisfies the entropy inequality (section 3.5.C.).

In the closure model, the interface in a mixed cell can be one or more boundaries that separate two phases. It is a straight line in the VOF method with linear interface reconstruction, while it represents closed lines or surfaces if there are numerous bubbles or drops in the cell. The boundaries between phases are not surfaces in a strict mathematic sense, but are very thin regions, in which the properties change with great abruptness from the properties of one phase to those of the other. We make the fundamental assumption that the fluids are not intermingled, or the grid scale is sufficiently larger than the diffused zone between two fluids. We assume the effect of surface geometry issues, such as area, shape, and number density, on total thermodynamic properties in the equilibrium state is so small that it can be neglected, which is true in most applications. If, however, we consider surface tension and absorption, the geometry issues may become of great importance in determining the thermodynamic properties. Some modifications are required for the closure model proposed, which is beyond the scope of this paper.

Concerning material properties, we assume each phase has individual state quantities satisfying its own EOS, and the quantities of pressure, temperature and sound speed of each phase vary continuously with volume and internal energy. Also each phase has a positive heat capacity at constant volume. For physical systems for which the heat capacity is negative, the closure model should be reexamined. No mass exchange or phase change is considered in this paper, and therefore the phase mass remains constant. A vacuum or a void is a solution of the compressible Euler equations; it is preferably treated as the third phase numerically. In the present work dealing with two phases, the void is implicitly filled by the neighborhood gas phase, as commonly done in the simulation of compressible gas flows. This approximation is reasonable for gas/gas and gas/liquid materials. The void that possibly appears in a solid phase is not handled in this paper.

3.5.A. Partition of external conservative quantities

The interaction of a Lagrangian cell with its neighboring cells results in the total change of energy, volume, and momentum, denoted by χ_E, χ_v, χ_m respectively. We suppose these quantities have been updated by a stable Lagrange scheme, and shall apportion them between two phases in the mixed cell. For brevity, only the changes of left phase are to be defined. Those of right phase are given by the conservation, $\chi_r = \chi - \chi_l$, for all quantities. Here, the left and the right phase are labeled only for the use of the Riemann solver; they are labeled arbitrarily elsewhere.

3.5.A.1. *Partition of momentum*

We assume that two phases undergo the same acceleration. This equal acceleration rate assumption gives the velocity change, same for two phases,

$$\delta u = \frac{\chi_m}{M}, \tag{3.1}$$

where $M = \sum_k M_k$ is the total mass of the cell. The momentum partitioned for the left phase is therefore

$$\chi_{ml} = \frac{M_l}{M}\chi_m. \tag{3.2}$$

The partition of momentum results in a change in kinetic energy,

$$\chi_{em} = \sum_k [\frac{1}{2}M_k(u_k + \delta u)^2 - \frac{1}{2}M_k u_k^2] = \sum_k [M_k u_k \delta u + \frac{1}{2}M_k(\delta u)^2], \tag{3.3}$$

which should be subtracted from the total energy change χ_E. It is noted that the change of kinetic energy (3.3) cannot be simply assigned to each phase, because it may violate entropy inequality and numerically produce negative internal energy. The present partition of momentum only updates the momentum and velocity, and leaves the change of kinetic energy to be treated together with the total energy in section 3.5.A.3..

3.5.A.2. *Partition of volume*

We follow the classic assumption of equal volumetric strain,

$$\chi_{vl} = \alpha_l \chi_v. \tag{3.4}$$

It maintains the positivity of volume if $\chi_v > -\Omega$, where Ω is the total volume of the cell, or Δx in one dimension.

Two materials undergo either expansion or compression process after volume partition. It is preferably for the process to be isentropic, in order to reduce numerical heat fluxes between two phases. Thus, the work done by the pressure force,

$$dw_k = p_k^{\text{isen}} dv_k \tag{3.5}$$

should be subtracted from phase energy, where p_k^{isen} denotes the pressure in the isentropic process. In this paper, a relation that contains variables with subscript k, such as (3.5),(3.6),(3.7) and (3.8), represents two relations, one for $k = l$ and the other for $k = r$, unless the subscript k appears in the sigma summation notation. The pressure in an isentropic process generally depends on volume, so that the work cannot be simply evaluated for general materials from (3.5), although the exact solution is available for the ideal gases. Some iterative procedures are needed. We consider a general method to approximate the work for general EOSs without recourse to iterations. The difficulty here is how to maintain the positivity of internal energy. A discussion on the positivity of internal energy as one of the constraints in optimization based closure model may be found in [3]. Suppose two phase pressures, p_k, become p_k^* after equal volumetric strain, using the isentropic relation

$$p_k^* = p_k + a_k^2 d\rho_k = p_k + a_k^2[M_k/(\Omega_k + \chi_{vk}) - M_k/\Omega_k], \tag{3.6}$$

where a_k is the speed of sound of phase k, and Ω_k is the volume of phase k. If two initial pressures and two end pressures are all positive or all negative, the pressure work is approximated by that of the smaller absolute value,

$$\chi_{wk} = \begin{cases} -\min(p_k\chi_{vk}, p_k^*\chi_{vk}) & p_k\chi_{vk} \geq 0 \\ -\max(p_k\chi_{vk}, p_k^*\chi_{vk}) & p_k\chi_{vk} < 0 \end{cases}. \tag{3.7}$$

If four pressures have different signs, the work is set to be,

$$\chi_{wk} = 0. \tag{3.8}$$

Formulas (3.7) and (3.8) are used instead of using formula (3.5). Although it is difficult to find the exact p_k^{isen} for a general material, one may expect that it lies between the initial pressure p_k and the end pressure p_k^*. The formulas (3.7) and (3.8) assure that the absolute work is not greater than that done in the isentropic process. There is only a second-order difference. That is to say, if an isentropic expansion/compression results in a positive internal energy after the pressure work is subtracted, formulas (3.7) and (3.8) guarantee the positivity of internal energy as well.

3.5.A.3. *Partition of energy*

The total energy consists of internal energy and kinetic energy. Since the kinetic energy has been determined by the updated velocities, the change of the internal energy is,

$$\chi_e = \chi_E - \chi_{em}. \tag{3.9}$$

Also, the work done in volume compression/expansion should be subtracted from the total internal energy. The energy that remains to be partitioned becomes

$$\chi_e^* = \chi_e - \sum \chi_{wk}. \tag{3.10}$$

Suppose the left phase gets a portion of energy χ_e^*, for $\alpha \in [0,1]$,

$$\chi_{el} = \alpha \chi_e^*. \tag{3.11}$$

We determine such a parameter α that the resulting two phases reach the state of maximum entropy. The entropy change due to the addition of internal energy under constant volume is

$$S(\alpha) = \int_0^{\alpha\chi_e^*} \frac{1}{T_l} de_l + \int_0^{(1-\alpha)\chi_e^*} \frac{1}{T_r} de_r. \tag{3.12}$$

Using $de = C_v dT$, one gets

$$S(\alpha) = \int_{T_l}^{T_l+\alpha\chi_e^*/C_{vl}} \frac{C_{vl}}{T} dT + \int_{T_r}^{T_r+(1-\alpha)\chi_e^*/C_{vr}} \frac{C_{vr}}{T} dT. \tag{3.13}$$

where $C_v = (\partial e/\partial T)_v$ is the heat capacity at constant volume. After a few manipulations, one gets the solution of $S'(\alpha) = 0$,

$$\alpha = \frac{C_{vl}}{C_{vl}+C_{vr}}[1 + \frac{C_{vr}(T_r - T_l)}{\chi_e^*}]. \tag{3.14}$$

It is easy to show $S''(\alpha) < 0$ for any positive heat capacity, so that the solution corresponds to the state of maximum entropy. The solution is subject to the constraint, $\alpha \in [0,1]$, numerically expressed as

$$\alpha = \max(0, \min(\alpha, 1)). \tag{3.15}$$

Without this constraint, the state of thermal equilibrium is reached by implicitly introducing some amount of heat flux between two phases.

3.5.B. Modeling of subgrid wave interactions

If two phases in the mixed cell are in disequilibrium, they will interact with each other and evolve obeying the conservation laws. Since there is no net change of total conservative quantities, the mixed cell can be regarded as an isolated system. In this isolated system, two phases will eventually reach pressure and velocity equilibrium; the subgrid wave interactions in the mixed cell are indeed a relaxation process. The thermal equilibrium is generally not achieved if the physical heat transfer is not considered.

In the present model, the subgrid wave interactions are supposed to obey the time-dependent conservation laws. This set of equations is imposed with the periodic boundary condition so that the net change is zero. We try to find an approximate but analytical solution of the set of equations that meet two conditions: (1) reaching pressure and velocity equilibrium for time approaches infinity, and (2) satisfying the entropy inequality.

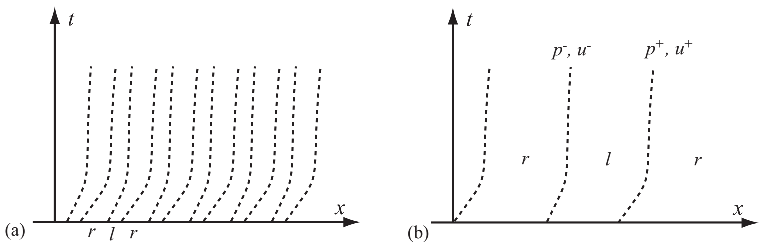

Figure 3.7. Interface configuration inside the interface cell

Suppose there are N uniformly distributed phase pairs in the mixed cell, as shown in Fig. 3.7a, and every phase pair consists of two phases with the same intensive quantities (density, pressure, velocity, temperature, volume fraction) as those defined in the cell. Since the periodic boundary condition is imposed on the phase pair, the problem remains the same for any N. If there are many interfaces in the cell, the parameter N represents approximately the number of interfaces, bubbles, or drops inside. For an interface that separates two pure materials in space, it may make no sense to think of the number of phase pairs. Here, we give a quick estimate of N for a cell associated with only one interface. Consider one phase pair ($N = 1$) in all cells. There are one interface inside and the other interface just at the grid face between two neighboring cells. The interface that separates two pure materials in space should be relaxed at a lower decay rate than the flow configuration of $N = 1$. It will be shown in what follows that the relaxation rate of both velocity and pressure is proportional to the parameter N. We may set a smaller N, say $N = 0.5$, for one interface, considering that there are actually two interfaces associated with a cell for $N = 1$. In general flow situations, it is expected that $N \leq 1$ for one interface in a cell. In multi-dimensions, it might be also possible to model the effect of the geometric properties, such as the length of the interface inside, on the relaxation rate using a real N, not necessarily an integer.

Consider the left phase in a phase pair, as shown in Fig. 3.7b, its solution is governed by both pressure and velocity at left and right boundaries, (p^-, u^-), (p^+, u^+) respectively. The conservation laws yield, at the discrete level for the left phase,

$$(M_l)_t = 0, \tag{3.16}$$

$$(\Omega_l)_t = -N(u^- - u^+), \tag{3.17}$$

$$(M_l u_l)_t = N(p^- - p^+), \tag{3.18}$$

$$(M_l E_l)_t = N(p^- u^- - p^+ u^+), \tag{3.19}$$

where M_l is the mass satisfying $M_l = \rho_l \alpha_l \Omega = \rho_l \Omega_l$. The equations for the right phase can be similarly written. Both velocity and pressure at faces are approximated by the acoustic Riemann solver,

$$u^+ = \bar{u} + (p_l - p_r)/(\rho_l a_l + \rho_r a_r), \tag{3.20}$$

$$p^+ = \bar{p} + \bar{s}(u_l - u_r), \tag{3.21}$$

where $\bar{u} = (\rho_l a_l u_l + \rho_r a_r u_r)/(\rho_l a_l + \rho_r a_r)$, $\bar{p} = (\rho_l a_l p_r + \rho_r a_r p_l)/(\rho_l a_l + \rho_r a_r)$, and $\bar{s} = (\rho_l a_l \rho_r a_r)/(\rho_l a_l + \rho_r a_r)$. Velocity ($u^+$) in (3.20) and pressure (p^+) in (3.21) are exactly the star state of the acoustic Riemann problem with (p_l, u_l, a_l) on the left and (p_r, u_r, a_r) on the right. Similarly, velocity (u^-) and pressure (p^-) are given by

$$u^- = \bar{u} - (p_l - p_r)/(\rho_l a_l + \rho_r a_r), \tag{3.22}$$

$$p^- = \bar{p} - \bar{s}(u_l - u_r). \tag{3.23}$$

This Riemann solver is valid before the sound wave reaches the other side of the phase volume; the system of equations should be integrated under the CFL condition if solved explicitly. It is prohibitive for very small phases using an explicit method. In order to overcome this stiffness, some implicit ingredient is necessary. We attempt to integrate the equations analytically. The solution of the left phase states is sufficient due to the conservation. It turns out that the momentum exchange between two phases can be obtained from the ordinary differential equation (ODE) of the velocity difference,

$$u^D = u_l - u_r,$$

to be discussed in section 3.5.B.1.. Also, the volume exchange under the isentropic assumption can be solved from the ODE of the pressure difference

$$p^D = p_l - p_r,$$

to be shown in section 3.5.B.2.. Special attention is paid to maintaining the positivity of phase volumes. The energy equation (3.19) is solved based on two solutions obtained, and treated at last in section 3.5.B.3.. The maximum entropy state in the possible solutions is pursued to remove the freedom in determining the energy flux.

3.5.B.1. *Momentum exchange*

Consider the momentum equation for left phase (3.18) in this section. Since no mass transfer between two fluids is considered, M_l remains constant, as seen from (3.16). The momentum equation is rewritten as $(u_l)_t = \frac{N}{M_l}(p^- - p^+)$. Similarly for the right fluid one gets $(u_r)_t = -\frac{N}{M_r}(p^- - p^+)$. Subtracting the two velocities leads to

$$u^D_t = \frac{N}{\bar{M}}(p^- - p^+),$$

where $\bar{M} = (M_l M_r)/(M_l + M_r)$. Using the acoustic Riemann solver for two pressures, $(p^- - p^+) = -2\bar{s}(u_l - u_r)$, gives

$$u_t^D = -\frac{2N\bar{s}}{\bar{M}} u^D.$$

This ordinary differential equation yields

$$u^D = u_0^D e^{-\frac{2N\bar{s}}{\bar{M}} t}, \tag{3.24}$$

where u_0^D denotes the initial velocity difference between two phases. The subscript $_0$ for initial values is often removed for the sake of clarity. A constant $\frac{N\bar{s}}{\bar{M}}$ is assumed here. It is clear that the velocity difference will experience exponential decay with time. The resulting momentum change of the left fluid is,

$$\delta_{M_l u_l} = \int_0^t N(p^- - p^+)dt = -2N\bar{s} \int_0^t u^D dt. \tag{3.25}$$

Substituting (3.24) into (3.25) leads to the momentum change for left fluid,

$$\delta_{M_l u_l} = u_0^D \bar{M}(e^{-\frac{2N\bar{s}}{\bar{M}} t} - 1). \tag{3.26}$$

It can be readily seen that for time approaches infinity, the change of the momentum is finite, and the solution approaches the velocity equilibrium. It is therefore valid not only in the early stage of interactions, but also provides a robust long-time solution.

It is noted that after one time step ($t = \Delta t$), the constant $\bar{s}$ introduces two errors of the same order, $O(u^D \Delta t^2)$. One error is introduced in (3.24), resulting from the integration of the ODE using the constant $\bar{s}$. The other error is introduced when the constant $\bar{s}$ is moved outside of the integral in the rightmost term in (3.25). In the order analysis, we assume that the difference between two phase quantities in the mixed cell is of $O(\Delta x)$, and also the difference between the initial and the end state of the same phase is of $O(\Delta t)$. For the sake of clarity, the spatial and the temporal error will not be distinguished hereafter, so that the error introduced by assuming the constant $\bar{s}$ is of the third order of smallness, $O(\delta^3)$, where δ denotes either Δt or Δx.

3.5.B.2. *Volume exchange*

The volume equation (3.17) will be integrated in this section. Consider the isentropic procedure,

$$\frac{dp_l}{d\Omega_l} = -\frac{M_l a_l^2}{\Omega_l^2}, \qquad \frac{dp_r}{d\Omega_r} = -\frac{M_r a_r^2}{\Omega_r^2}.$$

The pressure difference satisfies,

$$\frac{dp^D}{d\Omega_l} = -(\frac{M_l a_l^2}{\Omega_l^2} + \frac{M_r a_r^2}{\Omega_r^2}). \tag{3.27}$$

Volume equation (3.17) becomes, using (3.27),

$$p_t^D = -N\bar{c}(u^- - u^+),$$

where $\bar{c} = \frac{M_l a_l^2}{\Omega_l^2} + \frac{M_r a_r^2}{\Omega_r^2}$. Approximating the velocity difference by the acoustic solver, $(u^- - u^+) = -\frac{2}{\rho_l a_l + \rho_r a_r}(p_l - p_r)$, yields

$$p_t^D = -\frac{2N\bar{c}}{\rho_l a_l + \rho_r a_r} p^D.$$

Assuming the coefficient constant, one gets

$$p^D = p_0^D e^{-\frac{2N\bar{c}}{\rho_l a_l + \rho_r a_r} t}, \tag{3.28}$$

where p_0^D is the initial pressure difference. It is seen that the pressure difference approaches zero for time approaches infinity. The change of the pressure difference satisfies,

$$\delta p^D = p_0^D (e^{-\frac{2N\bar{c}}{\rho_l a_l + \rho_r a_r} t} - 1), \tag{3.29}$$

which gives an immediate volume change, using (3.27),

$$\delta_{\Omega_l} = -\frac{\delta p^D}{\bar{c}}. \tag{3.30}$$

However, (3.30) cannot avoid the appearance of negative volume if $\bar{c}$ is constant. As seen from the definition of $\bar{c}$, it varies abruptly for either phase volume (i.e., either Ω_l or Ω_r) approaching zero. Therefore the constant $\bar{c}$ is used only to update the pressure difference. In order to maintain the positivity of phase volume, we return to equation (3.27), and integrate it over Ω_l directly,

$$\delta p^D = \frac{M_l a_l^2}{\Omega_l} - \frac{M_r a_r^2}{\Omega - \Omega_l} + A_0. \tag{3.31}$$

Notice that the change of the pressure difference δp^D is given from (3.29) for any given t. The volume of the right phase, Ω_r in (3.27), has been replaced by $\Omega - \Omega_l$. Constant A_0 is determined from the initial state, $A_0 = -(\frac{M_l a_l^2}{\Omega_l} - \frac{M_r a_r^2}{\Omega_r})_0$. After simple algebraic calculations, we get a quadratic equation for Ω_l,

$$A_1 \Omega_l^2 - (M_l a_l^2 + M_r a_r^2 + \Omega A_1)\Omega_l + \Omega M_l a_l^2 = 0, \tag{3.32}$$

where $A_1 = \delta p^D - A_0$. The left volume is the root of (3.32),

$$\Omega_l = \frac{(M_l a_l^2 + M_r a_r^2 + \Omega A_1) - \sqrt{(M_l a_l^2 + M_r a_r^2 + \Omega A_1)^2 - 4 A_1 \Omega M_l a_l^2}}{2 A_1}. \tag{3.33}$$

This root is always real, and satisfies $\Omega_l \in (0, \Omega)$. The detailed proof is given in Appendix A. The volume change is then

$$\delta_{\Omega_l} = \Omega_l - (\Omega_l)_0. \tag{3.34}$$

It is stressed that to maintain the positivity of volume for general materials was a challenging task. This is one reason why the equal strain rate assumption without pressure relaxation is often preferred (e.g. [4,10]). We actually provide an analytical positivity-preserving solution of the problem without recourse to iterations, for both partial and full relaxations.

3.5.B.3. *Energy exchange*

Given momentum and volume changes, the energy change is evaluated by integrating (3.19),

$$\delta_{M_l E_l} = \int_0^t N(p^- u^- - p^+ u^+) dt = \int_0^t (\bar{u} N(p_l^- - p_r^+) + \bar{p} N(u_l^- - u_r^+)) dt,$$

which can be simplified to

$$\delta_{M_l E_l} = \hat{u} \int_0^t N(p_l^- - p_r^+) dt + \hat{p} \int_0^t N(u_l^- - u_r^+) dt, \tag{3.35}$$

or

$$\delta_{M_l E_l} = \hat{u} \delta_{M_l u_l} - \hat{p} \delta_{\Omega_l}, \tag{3.36}$$

where $\hat{u}$ is the value of $\bar{u}$ at $t' \in [0, t]$, and $\hat{p}$ is that of $\bar{p}$. Since both velocity and pressure are relaxed over time, it is reasonable to expect that they vary within the bounds

$$\hat{u} \in [u_l, u_r] \tag{3.37}$$

and

$$\hat{p} \in [p_l, p_r]. \tag{3.38}$$

In order to evaluate (3.36), simple choices, such as $\hat{u} = \bar{u}$ and $\hat{p} = \bar{p}$, may violate the entropy inequality, and numerically produce negative energy in some circumstances. We shall find a solution within the bounds (3.37) and (3.38) such that energy change $\delta_{M_l E_l}$ results in the state of maximum entropy in the cell. Consider $\xi_1, \xi_2 \in [0, 1]$, $\hat{u}$ and $\hat{p}$ satisfying (3.37) and (3.38) are expressed as

$$\hat{u} = u_l + \xi_1 (u_r - u_l),$$

$$\hat{p} = p_l + \xi_2 (p_r - p_l).$$

Substituting them into (3.36) gives

$$\delta_{M_l E_l} = u_l \delta_{M_l u_l} - p_l \delta_{\Omega_l} + \chi, \tag{3.39}$$

where

$$\chi = \xi_1 (u_r - u_l) \delta_{M_l u_l} - \xi_2 (p_r - p_l) \delta_{\Omega_l}.$$

By using the solutions for $\delta_{M_l u_l}$ and δ_{Ω_l}, from (3.26) and (3.30) respectively,

$$(u_r - u_l) \delta_{M_l u_l} = (u_r - u_l)^2 \bar{M} (1 - e^{-\frac{2N\bar{s}}{\bar{M}} t}) \geq 0,$$

$$-(p_r - p_l)\delta_{\Omega_l} = (p_r - p_l)\delta_{p_l - p_r}/\bar{c} = (p_r - p_l)^2(1 - e^{-\frac{2N\bar{c}}{\rho_l a_l + \rho_r a_r}t})/\bar{c} \geq 0,$$

it is clear that

$$\chi \in [0, \chi_{max}], \quad \chi_{max} = (u_r - u_l)\delta_{M_l u_l} - (p_r - p_l)\delta_{\Omega_l} \geq 0. \tag{3.40}$$

Now, two parameters, $\hat{u}$ and $\hat{p}$, have been reduced to one parameter χ. We need to find χ such that the resulting state in the cell attains the maximum entropy under the constraint $\chi \in [0, \chi_{max}]$. The problem is similar to that encountered in the partition of energy, discussed in section 3.5.A.3.. After a few algebraic manipulations, we get

$$\chi = \frac{(C_{vr}T_r + B_r)C_{vl} - (C_{vl}T_l - B_l)C_{vr}}{C_{vl} + C_{vr}}, \tag{3.41}$$

subject to the constraint

$$\chi = \min(\chi_{max}, \max(0, \chi)),$$

where $B_l = p_l\delta_{\Omega_l} + \frac{\delta^2_{M_l u_l}}{2M_l}$ and $B_r = p_l\delta_{\Omega_l} - \frac{\delta^2_{M_l u_l}}{2M_r} + (u_r - u_l)\delta_{M_l u_l}$. It is noted that the state of maximum entropy adopted here does not lead to the temperature equilibrium in the mixed cells in general. For instance, an interface with a temperature jump in pressure and velocity equilibrium will not be affected.

This completes all solutions required for the subgrid modeling.

3.5.C. Entropy inequality and numerical heat flux

Suppose the change of conservative quantities are updated by a stable Lagrange scheme that satisfies the entropy inequality, for the effective state,

$$Tds = (\chi_E - \chi_{e\tilde{m}}) + \tilde{p}\chi_v \geq 0,$$

where $\chi_{e\tilde{m}}$ is the change of kinetic energy evaluated from the effective velocity, which is in general different from χ_{em} that is evaluated from phase velocities. Luckily, under the assumption of equal acceleration, it can be shown that $\chi_{em} = \chi_{e\tilde{m}}$, as readily seen from (3.3). This property should be regarded as an advantage of the equal acceleration assumption in the partition of momentum. Thus, one gets,

$$Tds = \chi_e + \tilde{p}\chi_v \geq 0.$$

This guarantees the existence of a reference state in the mixed cell that meets the entropy inequality, since the effective sound speed is in consistent with the closure model. The principle of maximum entropy followed in the subgrid closure model assures that the entropy of the final state is not lower than the reference state, and thus satisfies the entropy inequality

$$\sum_k ds_k \geq 0.$$

It should be noted that the present closure model does not necessarily guarantee that each phase satisfies the entropy inequality,

$$ds_k \geq 0.$$

Consider the mixed cell has undergone an isentropic process in the Lagrange step first. The total heat flux satisfies

$$q = \chi_e + \tilde{p}\chi_v = 0.$$

Some amount of numerical heat flux can be possibly introduced between two phases, when the state of maximum entropy is pursued,

$$ds_k = \frac{q_k}{T_k}, \tag{3.42}$$

where the heat flux q_k is positive for one phase, and negative for the other, because of energy conservation $q = \sum_k q_k = 0$. The amount of the numerical heat flux is constrained by the energy partitioned,

$$\epsilon_q \leq |\chi_e^*| + |\chi_{max}|,$$

where $\epsilon_q = |q_l| = |q_r|$, and χ_e^* is the energy defined in (3.10). The term χ_{max} is of $O(\delta^2)$ as seen from (3.40). It can also be shown that χ_e^* is of $O(\delta^2)$ (Appendix B). Thus, the numerical heat flux satisfies

$$\epsilon_q \leq O(\delta^2). \tag{3.43}$$

That is to say, the numerical heat flux introduced between two phases in the present subgrid closure model is of second-order of smallness. The entropy change of each phase is of the second order, but the total entropy, $ds =$

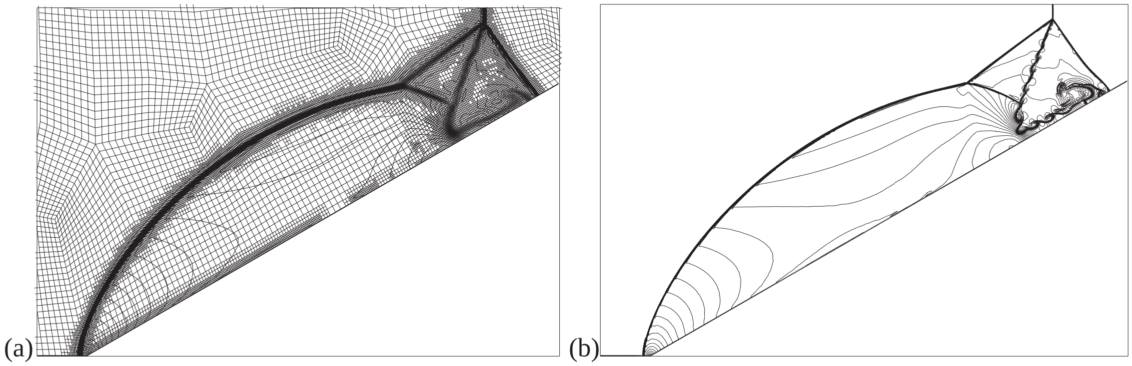

Figure 3.8. Unsteady shock wave (M_s=10) reflection over a 30° wedge: (a) 5-level solution-adaptive unstructured quadrilateral grid, overlapped with isopycnics (b) isopycnics using an adaptive grid 2-level finer than (a).

$|q_l/T_l + q_r/T_r| = \epsilon_q|1/T_l - 1/T_r| \leq O(\delta^3)$, is of third order. Thus, if the mixed cell undergoes an isentropic process in the Lagrange step, the entropy change in the closure model satisfies

$$|O(\delta^3)| \geq \sum_k ds_k \geq 0. \tag{3.44}$$

If the mixed cell experiences an entropy increase in the Lagrange step, after the passage of a shock wave for instance, one has $q > 0$. This non-zero heat is included in χ_e^*, and partitioned by formula (3.14). Since the maximum entropy state is pursued, the low temperature will never be further decreased, so that one gets the inequality,

$$\frac{q}{T_{\min}} + |O(\delta^3)| \geq \sum_k ds_k > 0, \tag{3.45}$$

where $T_{\min} = \min(T_l, T_r)$. The third order term $|O(\delta^3)|$ results from the numerical heat in modeling subgrid isentropic processes, as shown in (3.44).

3.6. Numerical examples

3.6.A. Lagrange-remap for single phase flows

A 2D test problem of the double Mach reflection of a strong shock (M_s=10) over a 30° wedge, which was presented in a review article[43], is investigated. This test case was also calculated with a staggered-mesh Lagrange-Remap method[24]. This calculation is performed using a solution-adaptive unstructured quadrilateral grid [38]. The initial grid has only 67 cells, and it is refined uniformly three levels. Figure 3.8a is a plot of the 5-level adaptive mesh together with corresponding density contours. Figure 3.8b is the solution using two more levels of refinement, in which the finest cell cells corresponds to a 1536×1024 uniform mesh in domain of 1.2×0.8. The shock wave and the shear discontinuity are resolved as well as the best version of the staggered-mesh Lagrange-Remap [24]. In the present result, the flow behind reflected shock wave is free of high-frequency noises.

3.6.B. Lagrange-remap for two-phase flows

The modified Sod shock tube problem with $x_0 = 0.5$, $(\gamma, \rho, u, p)^L = (2, 1, 0, 2)$ and $(\gamma, \rho, u, p)^R = (1.4, 0.125, 0, 0.1)$. A comparative study of this problem using various pressure relaxation closure models for Lagrangian hydrodynamics was performed by Kamm et al. [16].

The problem is calculated using different levels of relaxation by varying parameter N. Three levels of relaxation, $N = 0$ (no relaxation), 0.5 and 5 (well relaxed), are tested. All results at the final time of $t = 0.2$ are shown in Fig. 3.9. The results for parameter N larger than five is almost the same as those of $N = 5$, hardly distinguishable on the scale used in the figures. The results are in a good agreement with those of the sub-scale dynamics model reported in[16]. It is of interest that the level of relaxation hardly affects the flow features away from the interface, such as the left-traveling rarefaction waves and the right-traveling shock wave. A study of the computed result on finer meshes shows an L1 convergence rate of approximately one.

The interface is always resolved sharply in one grid cell without creating any pressure oscillations. Both pressure and velocity are in equilibrium even for the case of $N = 0$. The VOF Lagrange computation of Galera et al. [10] is based on the similar equal strain rate assumption, but they found that the phase quantities exhibit a large jump not only in density but also in pressure in the mixed cell. It is stressed that the present closure model for $N = 0$ contains only the partition of external conservative quantities without any subgrid interaction modeling. A significant improvement is clearly seen; the phase quantities in the mixed cell, such as density and pressure, are resoled similarly

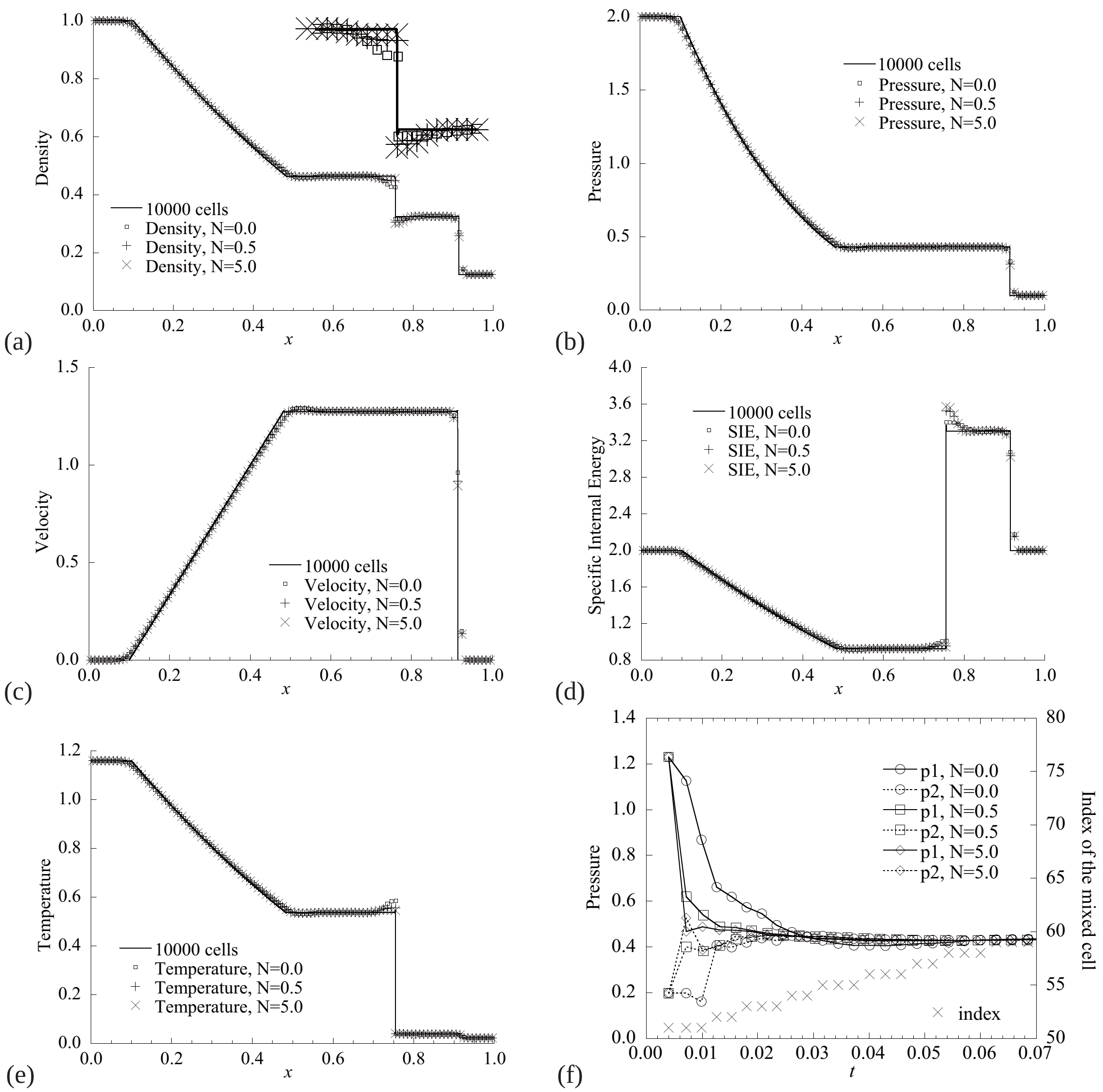

Figure 3.9. Results for the modified Sod shock tube problem at $t = 0.2$ for three levels of relaxation, $N = 0$ (no relaxation), $N = 0.5$ and $N = 5$ (well relaxed) : (a) density; (b) pressure; (c) velocity; (d) specific internal energy; (e) temperature pressure; (f) time-history of two pressures in the mixed cell together with the index of the mixed cell.

as well as those in single phase neighboring cells. The temperatures of both phases in the mixed cell are higher than their corresponding exact ones; this fact implies both phases have experienced an entropy increase since the initial breakdown of discontinuity. The temperature rise of the right phase is hardly seen in the scale used in Fig. 3.9e, but it can be deduced from the rise of internal energy seen in Fig. 3.9d. This temperature increase is allowed in reality, due to turbulent mixing in the early breakdown of a diaphragm for instance. This entropy condition can be verified from density data as well for perfect gases. The numerical densities near the interface must be not higher than the exact solution, as seen in Fig. 3.9a. All results reported by Kamm et al. show the same trend. However, a density increase is seen in the mixed cell in the result of Galera et al.[10]. It is therefore concluded that the present improvement is due to the careful partition of energy (section 3.5.A.3.) that maintains the entropy inequality.

The question remains as to how pressure and velocity in the mixed cell equilibrate without subgrid relaxation. The hint is seen from the history of two phase pressures in the mixed cells (Fig. 3.9f). The pressure data at every time step are all plotted. The material interface is initially located between two cells with the index of 50 and 51, so the index of the mixed cell starts from 51. The interface moves inside the cell for three steps, and then moves to the downstream cell with the index of 52, and so on. For non-zero parameter N, two pressures get equilibrated quickly in two or three steps. For $N = 0$, two pressures equilibrate more slowly. There are two mechanisms for the pressures to equilibrate without relaxation. One is the high pressure decreases more rapidly than the low pressure increases, as seen in the first three steps in the same mixed cell. Another mechanism is the motion of the interface from one cell to another. The phase on the downstream side is repeatedly merged with the same phase in the downstream cell. The subgrid closure model for $N = 0$ does work for such a problem associated with a single moving interface in pressure and velocity equilibrium.

3.7. Concluding remarks

The subgrid closure model proposed in this paper is general and robust for both equilibrium and non-equilibrium two-material cells. It can be constructed as an independent module that provides the updated quantities for each phase at $t = \Delta t$, after apportioning the change of entire cell quantities and modeling subgrid interactions. The updated quantities include individual phase velocity, volume and energy; other quantities, such as density and pressure, are then calculated from the conservation of mass and the EOSs. The entire cell quantities to be apportioned between two materials include volume, momentum and energy. It is not necessary to specify all these quantities. For instance, the closure model should work for the Lagrange scheme based on the single-velocity approximation (e.g. [45]) by setting momentum change $\chi_m = 0$ and $\delta_{M_l u_l} = 0$.

The closure model can handle general materials, and it is computationally efficient in the sense that it requires neither iterations nor sub-cycling in time. The closure model is also robust in the sense that it maintains the positivity of volume and entropy inequality. The model requires no knowledge of neighborhood cells, so that it is a local solver best for parallel processing.

The closure model is useful for the cells that contain two materials inside, no matter how the cells are discretized, either cell-centered or staggered, although it is only tested in the 1D Lagrange-Remap framework in this paper. The closure model proposed is unconditionally stable for any Δt, so that it can also be coupled with the implicit Lagrange schemes with a much larger time step. This can enhance the computational efficiency considerably for low speed two-phase phenomena, in which the compressibility is not negligible, such as cavitating flows. The multidimensional extension of the closure model is possible, to be reported in a separate paper. A 2-D numerical result based on the early version of the closure model, which did not include the positivity treatment for volume exchange as that discussed in section 3.5.B.2., can be found in [15].

References

[1] R. Abgrall and R. Saurel, Discrete equations for physical and numerical compressible multiphase mixtures, J. of Comput. Phys. 186, (2003) 361-396.

[2] M.R. Baer, J.W. Nunziato, A Two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. Int. J. Multiphase flow 186, (1986) 861-889.

[3] A. Barlow, R.N. Hill, M. Shashkov, Interface-aware sub-scale dynamics closure model, 2012 EULAG Model Users' Workshop, 23-27 June, 2012. Loughborough, UK

[4] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mech. and Engr., **99**, 235, (1992)

[5] G. Černe, S. Petelin, I. Tiselj Numerical errors of the volume-of-fluid interface tracking algorithm, Int. J. Numer. Meth. Fluids, **38**, (2002) 329-350.

[6] P. Colella, Volume-of-fluid methods for partial differential equations, in Godunov Methods: Theory and Applications, edited by E.F. Toro (Kluwer, New York, 2001), p.161.

[7] F. Coquel, K.E. Amine, E. Godlewski, B. Perthame, P. Rascle, A numerical method using upwind schemes for the resolution of two-phase flows, J. of Comput. Phys. 136, (1997) 272-288.

[8] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. **227** (2008) 5361-5384.

[9] R.P. Fedkiw, A. Marquina, B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, J. Comput. Phys. **148** (1999) 545-578.

[10] S. Galera, P.H. Maire, J. Breil, A two-dimensioanl unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. **229** (2010) 5755-5784.

[11] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the defined donating region scheme, Int. J. Numer. Meth. Fluids **35** (2001) 151-172.

[12] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: The stream scheme, J. Comput. Phys. **162** (2000) 1-32.

[13] C.W. Hirt, A.A. Amsden, J.L. Cook, Arbitrary Lagrangian-Eulerian computating method for all flow speeds, J. Comput. Phys, **14**, 227, (1974)

[14] C.W. Hirt and B.D. Nichols, Volume of Fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. **39** (1981) 201-225.

[15] D. Igra and M. Sun, Shock-water column interaction, from initial impact to fragmentation onset, AIAA J. **48** (2010) 2763-2771.

[16] J.R. Kamm, M.J. Shashkov, J. Fung, A.K. Harrison,T.R. Canfield, A comparative study of various pressure relaxation closure models for one-dimensional two-material Lagrangian hydrodynamics, Int. J. Numer. Meth. Fluids **65** (2011) 1311-1324

[17] A. Kapila, R. Menikoff, J. Bdzil, S. Son, D. Stewart, Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations, Physics of Fluids, **13** (2001), pp. 3002-3024.

[18] J. López, J. Hernández, P. Gómez and F. Faura, An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows, J. Comput. Phys. **208** (2005) 51-74.

[19] J. López, J. Hernández, P. Gómez and F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. **195** (2004) 718-742.

[20] E. Loth, Numerical approaches for motion of dispersed particles, droplets and bubbles, Prog. in Energy and Combustion Sci. 26 (2000) 161-223.

[21] A. Murrone, H. Guillard, A five equation reduced model for compressible two phase flow problems, J. Comput. Phys. 202, (2005) 664-698.

[22] W.F. Noh, P. Woodward, SLIC (Simple Line Interface Calculation). In Lecture Notes in Physics, van der Vooren AI, Zandbergen PJ (ed) (1976) 330.

[23] S. Osher, and J. A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations, J. Comput. Phys. **79** (1988) 12.

[24] Pember RB, Anderson RW, A comparison of staggered-mesh Lagrange plus remap and cell-centered direct Eulerian Godunov schemes for Eulerian shock hydrodynamics, UCRL-JC-139820, (2000)

[25] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. **199** (2004) 465-502.

[26] E.G. Puckett, A volume of fluid interface tracking algorithm with applications to computing shock wave rarefraction, Proceedings of the 4th International Symposium on Computational Fluid Dynamics, 1991.

[27] M. Raessi, J. Mostaghimi, M. Bussmann, Advecting normal vectors: a new method for calculating interface normals and curvatures when modeling two-phase flows, J. Comput. Phys. **226** (2007) 774-794.

[28] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. **141** (1998) 112-152.

[29] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluids **24** (1997) 671-691.

[30] R. Saurel, R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase Flows, J. Comput. Phys. 150 (1999) 425-467.

[31] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. **31** (1999) 567-603.

[32] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. **31** (2000) 228-237.

[33] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection, Int. J. Numer. Meth. Fluids **41** (2003) 251-274.

[34] J.A. Sethian, P. Smereka, Level set methods for fluid interfaces, Annu. Rev. Fluid Mech. 341, (2003) 341-372.

[35] H.B. Stewart, B. Wendroff, Two-phase flow: models and methods, J. Comput. Phys. 56, (1984) 363-409.

[36] M. Sun, Volume tracking of subgrid particles, Int. J. Numer. Meth. Fluids **66** (2011) 1530-1554.

[37] M. Sun, A thermodynamic and dynamic subgrid closure model for two-material cells, Int. J. Numer. Meth. Fluids, in press, (2013).

[38] M. Sun, K. Takayama, Conservative smoothing on an adaptive quadrilateral grid, *J. Comput. Phys.* **150** (1999) 143-180.

[39] M. Sun, K. Takayama, E.V. Timofeev EV, P.A. Voinovich, Direct numerical study of the dynamical shock-dust interaction, 21th Intl Symposium on Shock Waves, July, 1997, Australia

[40] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162, (2000) 301-337.

[41] B. van Leer, Towards the ultimate conservative difference scheme. V, a new approach to numerical convection, J. of Comput. Phys, **32**, 101, (1979)

[42] A.B. Wood, A Textbook of Sound, G. Bell & Sons Ltd., (1930), New York

[43] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys, **54**, 115, (1984)

[44] T. Yabe, F. Xiao, T. Utsumi, The constrained Interpolation Profile method for multiphase analysis, J. Comput. Phys. 169 (2001) 556-593.

[45] Y.V. Yanilkin, E.A. Goncharov, V.Y. Kolobyanin, V.V. Sadchikov, J.R. Kamm, M.J. Shashkov, W.J. Rider, Multi-material pressure relaxation methods for Lagrangian hydrodynamics, Computers & Fluids, 2012, http://dx.doi.org/10.1016/j.compfluid.2012.05.020

[46] D.L. Youngs, An interface tracking method for a 3D Eulerian hydrodynamics code, Technical Report 44/92/35, AWRE, 1984

[47] Q. Zhang, P.L.F. Liu, A new interface tracking method: The polygonal area mapping method J. Comput. Phys. **227** (2008) 4063-4088.