

A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions

Keh-Ming Shyue *

Department of Mathematics, National Taiwan University, Taipei 106, Taiwan

Received 18 June 2005; received in revised form 25 October 2005; accepted 27 October 2005

Available online 9 December 2005

Abstract

We present a simple volume-of-fluid approach to interface tracking for inviscid compressible multicomponent flow problems in two space dimensions. The algorithm uses a uniform Cartesian grid with some grid cells subdivided by tracked interfaces, approximately aligned with the material interfaces in the flow field. A standard volume-moving procedure that consists of two basic steps: (1) the update of a discrete set of volume fractions from the current time to the next and (2) the reconstruction of interfaces from the resulting volume fractions, is employed to find the new location of the tracked interfaces in piecewise linear form at the end of a time step. As in the previous work by LeVeque and the author to front tracking based on a surface-moving procedure (R.J. LeVeque, K.-M. Shyue, Two-dimensional front tracking based on high-resolution wave propagation methods, *J. Comput. Phys.* 123 (1996) 354–368), a conservative high-resolution wave propagation method is applied on the resulting slightly non-uniform grid to update all the cell values, while the stability of the method is maintained by using a large time step idea even in the presence of small cells and the use of a time step with respect to the uniform grid cells. We validate our algorithm by performing the simulation of a Mach 1.22 shock wave in air over a circular R22 gas bubble, where sensible agreement of some key flow features of the computed solutions are observed when direct comparison of our results are made with the existing experimental and numerical ones that appear in the literature. Other computations are also presented that show the feasibility of the algorithm together with a mixture type of the model equations developed by the author (K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with Mie–Grüneisen equation of state, *J. Comput. Phys.* 171 (2001) 678–707) for practical multicomponent problems with general compressible materials characterized by a Mie–Grüneisen form of the equation of state.

© 2005 Elsevier Inc. All rights reserved.

AMS: 65M06; 76L05; 76M20; 76T05

Keywords: Volume tracking; Wave propagation method; Multicomponent flows; Mie–Grüneisen equation of state; Impact problem

* Tel.: +8862 3366 2866; fax: +8862 2391 4439.

E-mail address: shyue@math.ntu.edu.tw.

1. Introduction

Our goal in this paper is to describe a simple volume tracking approach for the efficient numerical resolution of material interfaces arising from inviscid compressible multicomponent problems in two space dimensions. We use a Eulerian formulation of the equations in which the principal motion of the single-component flow is governed by the basic conservation laws for the mass, momenta in the x - and y -direction, and energy as

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho Eu + pu \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho Ev + pv \end{pmatrix} = 0, \quad (1)$$

respectively. Here ρ denotes the density, u is the particle velocity in the x -direction, v is the particle velocity in the y -direction, p is the pressure, and E is the specific total energy. To complete the model, the constitutive law for the thermodynamic behavior of the fluid component of interests is assumed to satisfy a Mie–Grüneisen equation of state of the form

$$p(\rho, e) = p_{\text{ref}}(\rho) + \Gamma(\rho)[\rho e - \rho e_{\text{ref}}(\rho)], \quad (2)$$

where e represents the specific internal energy, $\Gamma = (1/\rho)(\partial p/\partial e)|_{\rho}$ is the Grüneisen coefficient, and $p_{\text{ref}}, e_{\text{ref}}$ are the properly chosen states of the pressure and internal energy along some reference curve (e.g., along an isentrope, a single shock Hugoniot, or other empirically fit curves) in order to match the experimental data of the material being examined, see [54] and references cited therein for more details. As usual, we have $E = e + (u^2 + v^2)/2$.

As in the previous work by LeVeque and the author [31] to general front tracking for nonlinear hyperbolic systems of conservation laws in two dimensions, our approach to volume tracking is based on a fully discretized version of the high-resolution finite volume method in wave-propagation form. The algorithm uses an underlying uniform Cartesian grid with some rectangles subdivided by the tracked interfaces into two parts, where discontinuities in the material interfaces (or slip lines for that matter) are expected. Rather than employing a marker-and-cell representation of the interface and a surface-moving procedure to advance the interface location as before, which can be done for some simple cases (cf. [51,52]) but is very troublesome (at least in view of the code development aspect) in dealing with situations such as the splitting or merging of interfaces and the changes of complex multidimensional interfacial topology (cf. [9]), we adopt a volume-fraction based interface-moving procedure for an easy treatment of the aforementioned issues instead, see [19,38] for the other possible alternates. That is to say, we introduce a scalar field Y in the algorithm that is defined to be the fraction of each cell that is on one particular side of the interface. In the current interest of two-dimensional compressible flow problems, this volume-fraction function Y would be updated in each time step by solving an evolution equation of the form

$$\frac{\partial Y}{\partial t} + u \frac{\partial Y}{\partial x} + v \frac{\partial Y}{\partial y} = 0 \quad (3)$$

with a standard wave-propagation method. Here u and v are the particle velocities governed by the full Euler equations (1) which is unlike the case for many incompressible flow problems where a divergence-free constraint should be fulfilled on them for the fundamental agreement of the mathematical model being used. When this is done, the new interface can then be reconstructed in a cell-by-cell fashion from the information on Y , yielding a set of disconnected line segments (this is as opposed to a connected line segment case obtained by using a typical surface-moving procedure [6,31]) in the cells containing parts of the interface, see Section 3 for more exposition.

An important property of the wave-propagation form of the method is that reasonable time steps can be taken even if some of the subcells created by the tracked interfaces are orders of magnitude smaller than the underlying Cartesian cells. Uniform time steps are used throughout the computation, with the time step chosen so that the usual CFL (Courant–Friedrichs–Levy) condition is satisfied relative to the size of the regular grid cells. The method would typically remain stable in this case because near the small subcells the numerical

domain of dependence is extended automatically based on a large time step technique developed by LeVeque for general hyperbolic systems of partial differential equations (cf. [24–27]). High-resolution results can be achieved in a straightforward manner in most part of the domain, except possibly for cells near the tracked interfaces. This is discussed further in Section 2.

It should be mentioned that as in our surface-based front tracking algorithm [31], the basic philosophy of the proposed volume tracking method is that the subdivision of cells is not assumed to give the definitive location of the true interface, but is viewed rather as an approximate location yielding a refined grid that is better able to represent the discontinuous solution than the Cartesian grid alone. As a result, there is minimal smearing of the interface in the solution. To prevent the occurrence of spurious oscillation in the pressure when such a scenario occurs in the multicomponent context, the algorithm uses a mixture type of the model system described in [54] as the basis, and a standard finite volume method based on the wave-propagation formulation as for the consistent and accurate approximation of the system. Numerical results presented in Section 5 show that this gives an efficient way for the improvement of the resolution of the interfaces, and the correct convergence behavior of the weak solution for shock waves.

It is without doubt that there are many instances to have the desire of keeping the interface sharp, and consequently the advantage of a tracking method over a fixed grid capturing method. One good example among them is in the simulation of compressible two-phase flows with very different fluid component separated by interfaces, and under extreme flow condition, see [18,32,37,65,66,69] for an example. In this case, despite some simple situations (cf. [54]), numerical experiences indicate that it would be very difficult in general to devise a pressure-oscillation-free interface-capturing method to deal with the occurrence of more than one fluid component within a grid cell. In fact, it is one of the main motivations of the present attempt to develop a robust volume tracking method that is viable for a wide variety of two-phase flow problems with real materials.

There have been quite a few efforts over the years to develop efficient volume-of-fluid interface tracking methods for various practical applications. In regards to incompressible flow problems governed by the Navier–Stokes equations, for example, representative works in this direction are such as Hirt and Nichols for free boundary problems [16], Welch and Wilson for phase transition problems [67], and Gueyffier et al. [13], Lafaurie et al. [22], Rudman [47,48], and Sussman and Puckett [59] for two-phase flow problems. In regards to the current compressible flow problems governed by the inviscid Euler equations, there are works done by Chern and Colella [5] for nonlinear hyperbolic systems of conservation laws (see [1] also for a version of the method with the local adaptive mesh refinement included), and Miller and Puckett [37] for problems in multiple condensed matters.

The approach we take here is in essence similar to that of Chern and Colella [5] in which they also use a volume-of-fluid approach to keep track of the interface location, and then apply high-resolution conservative finite volume methods on the resulting grid. The key difference is that they use a more traditional flux difference form, together with a flux redistribution algorithm to maintain stability in the presence of small cells. It has been demonstrated before (cf. [31,52]) that our wave-propagation form can give good results when it is combined with a surface-based front tracking algorithm in one and two dimensions. One major objective of this paper is to show that this method can still be very effective when it is used in conjunction with a volume-based interface tracking method for two-dimensional compressible hydrodynamics problems.

The format of this paper is outlined as follows. In Section 2, we review the basic idea of a finite volume method in wave-propagation form on a slightly non-uniform Cartesian grid with a moving interface. In Section 3, we describe the standard piecewise linear interface construction method based on a given set of discrete volume fractions. The volume tracking algorithm for inviscid compressible flow is discussed in Section 4. Results of some sample validation tests as well as application of the method to practical problems are shown in Section 5.

2. Wave-propagation methods

To set the groundwork for the later development of a volume-based interface tracking algorithm, we begin our discussion by reviewing a finite volume method in wave-propagation form for the numerical approximation of general hyperbolic systems of conservation laws in two space dimensions,

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} + \frac{\partial g(q)}{\partial y} = 0. \quad (4)$$

Here $q \in \mathbb{R}^m$ denotes the vector of m conservative quantities, and $f(q)$, $g(q)$ denote the flux vectors, see (1) for an example. For all practical purposes as concerned below, it is essential to describe the method on a slightly non-uniform grid that is composed of a set of regular cells with a fixed mesh spacing Δx and Δy in the x - and y -direction, respectively, and some irregular cells which are formed by inserting a piecewise linear representation of the tracked interfaces into the grid, yielding the subdivision of some regular cells into two subcells, see Fig. 1a for an illustration.

In a finite volume formulation of the method, the cell average of each grid cell is defined by integrating the solution over the cell and dividing by the area of the cell. Hence the approximate value of the cell average of the solution over any given cell S at time t_n can be written as

$$\mathcal{Q}_S^n \approx \frac{1}{\mathcal{M}(S)} \int_S q(x, y, t_n) dx dy, \quad (5)$$

where $\mathcal{M}(S)$ is the measure (area) of this cell. The methods we use are based on solving one-dimensional Riemann problems at each cell edge, and the waves (i.e., discontinuities moving at constant speeds) arising from the Riemann problem are employed to update the cell averages in the cells neighboring each edge (cf. [26,27,31]).

2.1. First-order method

As a first example to explain the basic idea of our first-order method, we consider the cell edge between irregular cells C and D as illustrated in Fig. 1a. We solve the one-dimensional Riemann problem normal to this face, which in this case will be

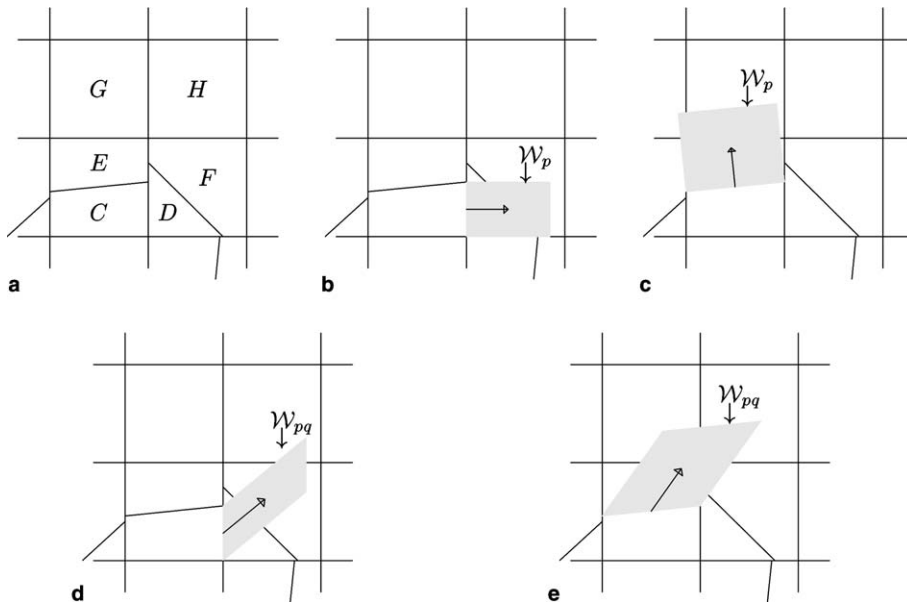


Fig. 1. First-order wave-propagation method. (a) The typical grid created by our volume tracking algorithm. (b) A wave propagating normally from the cell edge between cells C and D updates the solution in both cells D and F . (c) A wave propagating normally from the cell edge between cells C and E updates four neighboring cell averages. (d) A wave propagating both normally and tangentially from the cell edge between cells C and D updates the solution in cells D , F , and H . (e) A wave propagating both normally and tangentially from the cell edge between cells C and E updates five neighboring cell averages. Here \mathcal{W}_p represents the region swept out by the p -wave, and \mathcal{W}_{pq} represents the region swept out by the p -wave and q -wave in the normal and tangential to the cell edge, respectively, over a time step Δt for $p, q = 1, 2, \dots, m$.

$$\frac{\partial q}{\partial t} + \frac{\partial f(q)}{\partial x} = 0,$$

with the initial data given by Q_C^n and Q_D^n . If Roe's approximate solver is chosen to solve the Riemann problem, the above nonlinear equation would be replaced by a linear system of the form

$$\frac{\partial q}{\partial t} + A(Q_C^n, Q_D^n) \frac{\partial q}{\partial x} = 0,$$

where $A(Q_C^n, Q_D^n)$ is a constant matrix that depends on the initial data and is a local linearization of the Jacobian matrix of $f(q)$ about an average state (cf. [29,46,62]). The result is that the jump $Q_D^n - Q_C^n$ is decomposed into eigenvectors of the matrix A ,

$$Q_D^n - Q_C^n = \sum_{p=1}^m \alpha_p r_p,$$

where r_p is the p th eigenvector of A and $A r_p = \lambda_p r_p$ with λ_p the corresponding eigenvalue. The scalar α_p gives the wave strength.

Clearly, if the wave speed $\lambda_p > 0$ the p th wave propagates towards the right, while if $\lambda_p < 0$ it propagates towards the left. Fig. 1b shows a typical p -wave with $\lambda_p > 0$, where part of cell D and part of cell F are affected by this wave over a time step Δt from the current time t_n to the next t_{n+1} . In the simplest case of the wave-propagation method, i.e., a two-dimensional extension of the first-order Godunov's method, these cell averages are updated by

$$Q_S^{n+1} := Q_S^n - \frac{\mathcal{M}(\mathcal{W}_p \cap S)}{\mathcal{M}(S)} \alpha_p r_p \quad (6)$$

for $S = D, F$. Here \mathcal{W}_p represents the region swept out by the wave and $\alpha_p r_p$ is the jump across the wave. Note that we have employed a standard initialization procedure that assigns each cell average to its values at the previous time step, $Q_S^{n+1} := Q_S^n$ for all S .

Now, if we are concerned with the cell edge between cells C and E (this is one segment of the tracked interface), a Riemann problem would be solved in the direction normal to the interface with data Q_C^n and Q_E^n . Note that the Euler equations (1) are isotropic and easily rotated to this frame (cf. [26]). The p -wave indicated in Fig. 1c overlaps four cells and using (6) in a similar manner gives modification of the cell average in each of these cells by the jump across this wave, weighted by the fraction of the cell area covered by the wave.

To improve the stability of this first-order method, it is a customary approach by introducing transverse propagation of waves based on the solutions of Riemann problems in the orthogonal direction (cf. [26,27,31]). Again we use the cell edge between cells C and D in Fig. 1a, as an example. Then for the equation we take the tangential portion of (4), i.e., the portion of equation in the y -direction,

$$\frac{\partial q}{\partial t} + \frac{\partial g(q)}{\partial y} = 0,$$

and for the initial data if we are working with the p -wave we take

$$Q^L = Q_C^n + \sum_{q=1}^{p-1} \alpha_q r_q, \quad Q^R = Q^L + \alpha_p r_p.$$

With the Roe solver, the solution to this Riemann problem yields a splitting of $Q^R - Q^L = \alpha_p r_p$ into eigenvectors of the Roe matrix $B(Q^L, Q^R)$ of the form

$$\alpha_p r_p = \sum_{q=1}^m \beta_q w_q$$

where B is a suitable approximation of the Jacobian matrix of $g(q)$, and $B w_q = \mu_q w_q$. The wave shown in Fig. 1b is thus split into m waves propagating in the y -direction with speed given by the corresponding

eigenvalue μ_q , and a typical one is shown in Fig. 1d for the case $\mu_q > 0$. Here to update the cell averages affected by this wave, we employ an analogous formula as in (6),

$$Q_S^{n+1} := Q_S^{n+1} - \frac{\mathcal{M}(\mathcal{W}_{pq} \cap S)}{\mathcal{M}(S)} \beta_q w_q \quad (7)$$

for $S = D, F$ and H , where \mathcal{W}_{pq} represents the region swept out by the p -wave and q -wave over a time step Δt , and $\beta_q w_q$ is the jump across the wave as usual.

We note that, using the same idea as described above, there is no difficulty to split any p -wave into m subwaves \mathcal{W}_{pq} for $q = 1, 2, \dots, m$, moving oblique to the cell edge rather than normal to it. Each of these waves can then be propagated in a similar manner, computing its intersection with the neighboring cells and updating the cell averages, see Fig. 1e for another illustration of a subwave, $\mu_q > 0$, from the cell edge between cells C and E .

It is important to mention that the method remains fully conservative when the transverse propagation of waves is included in the algorithm. Furthermore, because waves are allowed to propagate through more than one grid cell and are not confined to stay within neighboring cells, this method is typically stable as long as the time step Δt satisfies the usual CFL (Courant–Friedrichs–Lewy) condition relative to the regular cell sizes: Δx and Δy ,

$$v = \frac{\Delta t \max(\lambda_p, \mu_q)}{\min(\Delta x, \Delta y)} \leq 1, \quad (8)$$

even when there are very small irregular cells near the tracked interfaces. Nevertheless, the method is still only first-order accurate, see [26,28] for more details.

2.2. High-resolution method

To achieve high resolution of the method (i.e., second-order accurate on smooth solutions, and sharp and monotone profiles on discontinuous solutions), as usual, we use the solution of the Riemann problem in the direction normal to each cell edge as a basis for the introduction of correction waves in a piecewise-linear form with zero mean value. Having that, we then propagate each of these waves by the corresponding speed over a time step Δt to update the cell averages it overlaps. To demonstrate this, Fig. 2a shows a typical structure of a correction wave at time t_n for a case with $\lambda_p > 0$ from the cell edge between regular cells G and H , i.e., the graph of the three-dimensional piecewise-linear function

$$\mathcal{Z}_p(x, y) = \begin{cases} \sigma_p(x - \bar{x}_G) & \text{if } (x, y) \in [\bar{x}_G - \Delta x/2, \bar{x}_G + \Delta x/2] \times [y_{GH}^a, y_{GH}^b], \\ 0 & \text{if otherwise,} \end{cases} \quad (9)$$

where $\sigma_p = \alpha_p r_p / \Delta x$ is the slope of \mathcal{Z}_p , \bar{x}_G is the x -coordinate of the geometric center of cell G , and y_{GH}^a, y_{GH}^b are the lower and upper ends of the y -coordinate of this cell edge, respectively. Note that, when $\lambda_p < 0$, \mathcal{Z}_p can be made quite easily also by simply changing \bar{x}_G in (9) to \bar{x}_H . The new location of \mathcal{Z}_p moving in the x -direction over a time step Δt is $\mathcal{Z}_p(x - \lambda_p \Delta t, y)$, see Fig. 2b. The cell averages G and H are then modified by

$$\begin{aligned} Q_G^{n+1} &:= Q_G^{n+1} - \frac{1}{2} \frac{|\lambda_p| \Delta t}{\Delta x} (\Delta x - |\lambda_p| \Delta t) \sigma_p, \\ Q_H^{n+1} &:= Q_H^{n+1} + \frac{1}{2} \frac{|\lambda_p| \Delta t}{\Delta x} (\Delta x - |\lambda_p| \Delta t) \sigma_p, \end{aligned} \quad (10)$$

where the correction terms are the volume of the portion of $\mathcal{Z}_p(x - \lambda_p \Delta t, y)$ (weighted by the cell size) that overlaps the grid cell. In practice, the strength of each wave should be limited by using a slope-limiter (cf. [10,29]), and so each α_p in the above σ_p is replaced by a limited value $\tilde{\alpha}_p$ obtained by comparing α_p with the corresponding α_p from the neighboring Riemann problem to the left (if $\lambda_p > 0$) or to the right (if $\lambda_p < 0$). Conservation is clearly maintained in this step with any choice of slopes, because the above two correction terms sum to zero.

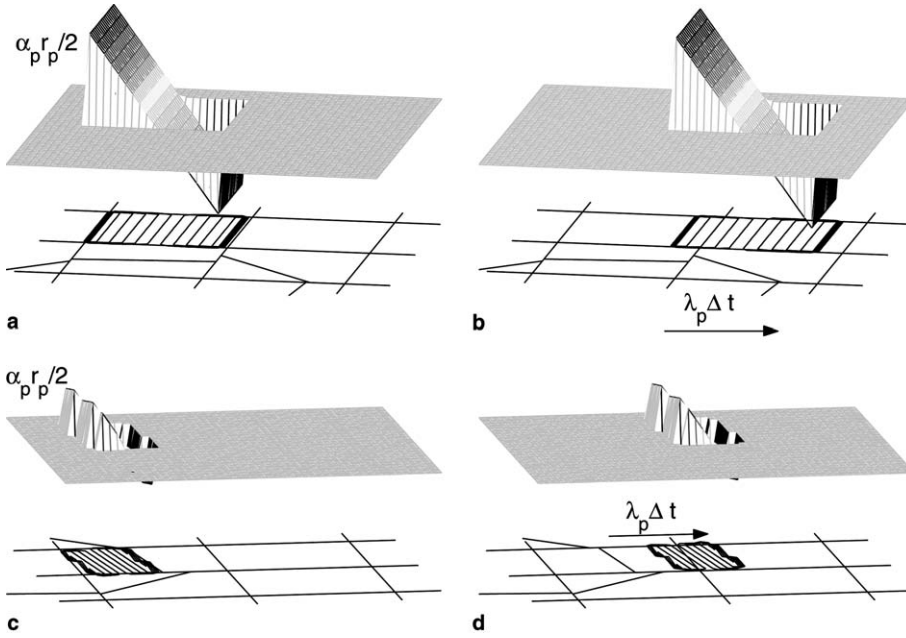


Fig. 2. High-resolution wave propagation method. (a) A correction wave at t_n for a case with $\lambda_p > 0$ from the cell edge between cells G and H . (b) Propagation of the correction wave in (a) with the speed $\lambda_p > 0$ over a time step Δt updates the solution in cells G and H . (c) A correction wave at t_n for a case with $\lambda_p > 0$ from the tracked interface between cells C and E . (d) Propagation of the correction wave in (c) with the speed $\lambda_p > 0$ over a time step Δt updates the solution in neighboring cells. See Fig. 1a for the basic labeling of the grid cell.

To extend the above approach to the more delicate case where the cell edge is adjacent to one or two irregular cells, we may first define a general form of \mathcal{Z}_p in which the normal direction to the cell edge is at an arbitrary angle θ to the x -axis. Without loss of generality, we consider the one between cells C and E shown in Fig. 1a, as an example. Then, for the p -wave, one simple choice of the slope is $\sigma_p = \alpha_p r_p / \Delta d$, where Δd is some measure of the normal distance between cells C and E , such as $\Delta d = \bar{\xi}_E - \bar{\xi}_C$ with $\bar{\xi}_S = \bar{x}_S \cos \theta + \bar{y}_S \sin \theta$ representing the ξ -coordinate of the geometric center of cell S in a local (ξ, η) -coordinate system (i.e., $\xi = x \cos \theta + y \sin \theta$, $\eta = -x \sin \theta + y \cos \theta$), for $S = C, E$ (cf. [26]). With these definitions, for a case with $\lambda_p > 0$, we form the correction wave as

$$\mathcal{Z}_p(\xi, \eta) = \begin{cases} \sigma_p(\xi - \bar{\xi}_0) & \text{if } (\xi, \eta) \in [\bar{\xi}_0 - \Delta d/2, \bar{\xi}_0 + \Delta d/2] \times [\eta_{CE}^a, \eta_{CE}^b], \\ 0 & \text{if otherwise,} \end{cases} \quad (11)$$

where $\bar{\xi}_0$ is the ξ -coordinate of the center of the rectangular region: $[\xi_{CE}, \xi_{CE} - \Delta d] \times [\eta_{CE}^a, \eta_{CE}^b]$, see Fig. 2c for an illustration. Having that, the new location of \mathcal{Z}_p propagating in the normal direction to the cell edge over a time step Δt is $\mathcal{Z}_p(\xi - \lambda_p \Delta t, \eta)$, see Fig. 2d. Now, the cell averages affected by this correction wave can be updated by

$$Q_S^{n+1} := Q_S^{n+1} + \frac{\mathcal{M}(\mathcal{Z}_p \cap S)}{\mathcal{M}(S)}, \quad (12)$$

where $\mathcal{M}(\mathcal{Z}_p \cap S)$ is the measure (volume) of the portion of $\mathcal{Z}_p(\xi - \lambda_p \Delta t, \eta)$ that partly coincides the cell S . It is true that the unlimited slope employed in (11) would be a sensible one to use only if there is no large jump in the solution across the wave. As a general practice, it should be good also to limit the strength of each wave α_p using a slope-limiter in an upwind manner based on the sign of λ_p . But if we now look at the case shown in Fig. 2c, it is not obvious right away how to obtain a limited value $\tilde{\alpha}_p$, because we do not know exactly what is the most appropriate neighboring α_p to be used in the slope limiter. There are some discussions in [26] on how such corrections might be applied on irregular cells, and the algorithm would perhaps be improved by implementing this. However, since there is expected to be a large jump at the tracked interface, this is precisely

where the limiters are expected to minimize the effect of these second-order corrections and to reduce the method to Godunov. It seems reasonable to drop these corrections altogether with considerable simplification of the algorithm.

To end this section, it should be mentioned that the wave-propagation method described here is not limited to solve hyperbolic systems in conservation form, but works equally well to solve hyperbolic systems which are not in conservation form, see [29] for the details.

3. Interface reconstruction scheme

Denote Y_{ij} as being the approximate value of the cell average of the solution (3) over the rectangular region $C_{ij} = \{(x, y) \mid x_i \leq x \leq x_i + \Delta x, y_j \leq y \leq y_j + \Delta y\}$ of grid cell (i, j) . Then, when $Y_{ij} \in (0, 1)$, we expect there is an interface (represents as a linear segment) lying somewhere within C_{ij} . Here we are concerned with a popular piecewise linear interface construction (PLIC) method (cf. [42,45]) for the reconstruction of interfaces from the discrete set of volume fractions Y on uniform Cartesian cells which will be one of the fundamental elements in our volume tracking algorithm, see Section 4. Note that in a standard PLIC method each of the linear segments would be made by first computing an approximate normal vector to the interface within the grid cell, and then determining its location in such a way that the cell volume fraction is preserved in this step. In the following, we review the aforementioned PLIC steps in more details.

3.1. Compute interface normal

As a preliminary, we consider a gradient method of Parker and Youngs [40,70] in which, when $Y_{ij} \in (0, 1)$, the gradient of the volume fraction at the grid cell (i, j) , denoted by ∇Y_{ij} , is computed explicitly by a central-difference like formula of the form

$$\nabla Y_{ij} = \left(\frac{\partial Y}{\partial x}, \frac{\partial Y}{\partial y} \right)_{ij} \approx \left(\frac{Y_R - Y_L}{2\Delta x}, \frac{Y_T - Y_B}{2\Delta y} \right) \quad (13)$$

with the quantities Y_R , Y_L , Y_T , and Y_B defined as

$$\begin{aligned} Y_R &= \frac{1}{4}(Y_{i+1,j-1} + 2Y_{i+1,j} + Y_{i+1,j+1}), & Y_L &= \frac{1}{4}(Y_{i-1,j-1} + 2Y_{i-1,j} + Y_{i-1,j+1}), \\ Y_T &= \frac{1}{4}(Y_{i-1,j+1} + 2Y_{i,j+1} + Y_{i+1,j+1}), & Y_B &= \frac{1}{4}(Y_{i-1,j-1} + 2Y_{i,j-1} + Y_{i+1,j-1}). \end{aligned}$$

Having that, the unit normal of the interface at cell (i, j) may be set to $\vec{N}_{ij} = -\nabla Y_{ij} / |\nabla Y_{ij}|$, and so in this cell \vec{N}_{ij} points in the direction of the most rapid decrease of Y . While the use of (13) is a very simple one for the determination of the interface normal, it is reported in [42], for instance, that this method exhibits only overall first-order accuracy on the location of the interface. To achieve higher-order accuracy, such as the second order, one popular approach is to use the least-squares method of Puckett [42,43] which is described next.

Let \mathcal{E}_{ij} be the sum squared error of volume fractions for the cell (i, j) of the form

$$\mathcal{E}_{ij} = \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} (Y_{mn} - \tilde{Y}_{mn})^2, \quad (14)$$

where Y_{mn} is the original volume fraction for cell (m, n) , and \tilde{Y}_{mn} is the approximate volume fraction for cell (m, n) based on an extended reconstructed interface at cell (i, j) . The aim of the Puckett's method is to obtain a local minimum of \mathcal{E}_{ij} by adjusting ∇Y_{ij} (and so the location of the interface) iteratively in the center of a 3×3 mesh block until convergence. Note that, during each iteration, i.e., for any given ∇Y_{ij} , we have to perform an interface-finding procedure, see Section 3.2, so that the condition $\tilde{Y}_{ij} = Y_{ij}$ is fulfilled in the center cell. Here the numerical method used to solve the minimization problem is a combination of the golden section search and the successive parabolic interpolation (cf. [3] and the NETLIB routine FMIN). To ensure convergence of the correct result by the method, it is important to choose a suitable interval to work with in which \mathcal{E}_{ij} attains a minimum. In practice, this is typically done by a trial-and-error procedure (cf. [15,42,50]) where an initial guess of ∇Y_{ij} is picked up first, say based on (13) or more favorably the Youngs' second method (see below),

and then it is expanded slowly about this initial guess to an interval until \mathcal{E}_{ij} at the endpoints of the interval has a bigger value than the one given by the initial guess.

Note that in Puckett's method ∇Y_{ij} is determined implicitly and also rather costly as a result of the least-squares problem (14). It is interesting to mention that a much simpler method of this type is Youngs' second method [71] where ∇Y_{ij} is obtained by solving a weighted least-squares problem of the form

$$\mathcal{E}_{ij}^\omega = \sum_{m=i-1}^{i+1} \sum_{n=j-1}^{j+1} \omega_{mn} (Y_{mn} - \tilde{Y}_{mn})^2, \quad (15)$$

where in this work ω_{mn} is taken as the inverse distance from the (m,n) th cell center to the (i,j) th cell center, and \tilde{Y}_{mn} is defined to be a Taylor series expansion of the volume fraction from the (m,n) th cell center to the (i,j) th cell center, i.e.,

$$\tilde{Y}_{mn} = Y_{ij} + \nabla Y_{ij} \cdot (x_{m+1/2} - x_{i+1/2}, y_{n+1/2} - y_{j+1/2}).$$

With that, the solution of the minimization problem to (15), denoted by $z = (\nabla Y_{ij})^T$, can be shown to satisfy the linear system of the form

$$A^T A z = A^T b, \quad (16)$$

where

$$A = \begin{pmatrix} -\Delta x / \Delta s & -\Delta y / \Delta s \\ -1 & 0 \\ -\Delta x / \Delta s & \Delta y / \Delta s \\ 0 & -1 \\ 0 & 1 \\ \Delta x / \Delta s & -\Delta y / \Delta s \\ 1 & 0 \\ \Delta x / \Delta s & \Delta y / \Delta s \end{pmatrix}, \quad b = \begin{pmatrix} (Y_{i-1,j-1} - Y_{ij}) / \Delta s \\ (Y_{i-1,j} - Y_{ij}) / \Delta x \\ (Y_{i-1,j+1} - Y_{ij}) / \Delta s \\ (Y_{i,j-1} - Y_{ij}) / \Delta y \\ (Y_{i,j+1} - Y_{ij}) / \Delta y \\ (Y_{i+1,j-1} - Y_{ij}) / \Delta s \\ (Y_{i+1,j} - Y_{ij}) / \Delta x \\ (Y_{i+1,j+1} - Y_{ij}) / \Delta s \end{pmatrix},$$

and $\Delta s = \sqrt{(\Delta x)^2 + (\Delta y)^2}$. In general, one may want to use either a normal equation-based or a QR -based algorithm for the solution of this least-squares problem (15), see [63]. But after a simple algebraic calculation, we find that the matrix $A^T A$ is simply a diagonal matrix of the form

$$A^T A = 2 \operatorname{diag} \left(2 \left(\frac{\Delta x}{\Delta s} \right)^2 + 1, 2 \left(\frac{\Delta y}{\Delta s} \right)^2 + 1 \right),$$

and so we may find the solution of (16) very easily, and so set $\vec{N}_{ij} = -\nabla Y_{ij} / |\nabla Y_{ij}|$ as the normal of the interface for cell (i,j) .

3.2. Determine interface location

Having gotten \vec{N}_{ij} , we now describe a numerical procedure (cf. [45]) for an efficient reconstruction of the location of the interface over the rectangular region C_{ij} . For convenience of the later description, let us assume that $(\bar{x}, \bar{y}) \in C_{ij}$ is a reference point known a priori on the interface and $\vec{N}_{ij} = (\alpha_{ij}, \beta_{ij})$ with $\alpha_{ij}^2 + \beta_{ij}^2 = 1$. Then, the linear-segment representation of the desired location of the interface in cell (i,j) in parametric form may be expressed by $\bar{L}_{ij}(\tau) : (x(\tau), y(\tau)) = (\bar{x} - \beta_{ij}\tau, \bar{y} + \alpha_{ij}\tau)$, where τ is a non-negative parameter value. Clearly, with the proper choice of the interval on τ , $\bar{L}_{ij}(\tau)$ would form a line, subdividing C_{ij} into two parts. Denote C_{ij}^- and C_{ij}^+ to be the region occupied by subcells in the back and in the front of \bar{L}_{ij} , respectively. Our aim here is to determine (\bar{x}, \bar{y}) and so \bar{L}_{ij} in such a way that $\mathcal{M}(C_{ij}^-) / \mathcal{M}(C_{ij}) = Y_{ij}$. Surely, the latter condition ensures the basic conservation of the volume fraction if we set $Y = 1$ for C_{ij}^- and $Y = 0$ for C_{ij}^+ . Note that this setting is consistent with our definition of the direction of \vec{N}_{ij} , i.e., a unit vector pointing in the steepest descent direction of Y . Recall that $\mathcal{M}(S)$ is the measure (area) of the region S .

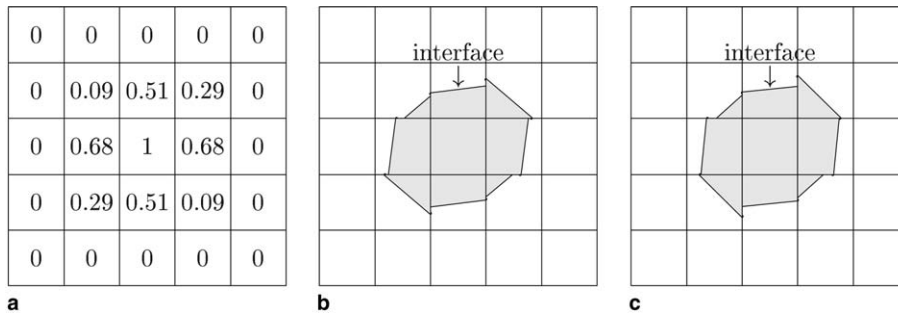


Fig. 3. An example of the reconstruction of the interface location from volume fractions. (a) The original cell-averaged volume fraction functions for an elliptical-shape interface. (b) Result obtained by using the method of Parker and Youngs. (c) Result obtained by using the method of Puckett. Here in the graphs we set $Y = 1$ and $Y = 0$ for grid cells in the shaded and non-shaded region, respectively.

To accomplish this, we define $\varepsilon_{ij}^{(k)} = \mathcal{M}(C_{ij}^{(k)}) / \mathcal{M}(C_{ij}) - Y_{ij}$ as being the error of the cell volume fraction (i, j) at the iteration step k , where $\mathcal{M}(C_{ij}^{(k)})$ is the measure of $C_{ij}^{(k)}$ using a reference point at $(x^{(k)}, y^{(k)})$. With that, by choosing a suitable initial guess: $(x^{(0)}, y^{(0)})$ and $(x^{(1)}, y^{(1)})$ such that $\varepsilon_{ij}^{(0)} \varepsilon_{ij}^{(1)} < 0$, it is not difficult to apply the standard secant method for finding the zero of $\varepsilon_{ij}^{(k)}$, yielding the convergence of the solution $(x^{(k)}, y^{(k)}) \rightarrow (\bar{x}, \bar{y})$ in a superlinear rate. Note that, given \bar{L}_{ij} , the basic geometric information of the splitted grid cell, such as the area and center of the cell, can be found quite easily based on algorithms in computational geometry [39].

Finally, Fig. 3 shows results for a typical situation in a volume-of-fluid algorithm for interface reconstruction from volume fractions, where we have used both the methods of Parker and Youngs, and Puckett on a very coarse 5×5 grid for illustrations, see [42] for an extensive study of the accuracy of these schemes and many others. Note that, in the graphs, we have set $Y = 1$ for grid cells in the shaded region, and $Y = 0$ for grid cells in the non-shaded region. As mentioned earlier, the tracked interface is approximated by a piecewise linear segment with knots at the points where this segment intersects with the grid lines. We use the same data structure as described in [31] for the realization of the tracked interfaces in our computer program.

4. Volume tracking algorithm

In each time step, our volume-based interface tracking method for compressible fluid dynamics problems in two dimensions consists of the following steps:

- (1) Take a time step on the current non-uniform grid using a first-order version of the wave-propagation method to update the cell averages of volume fractions governed by (3) at the next time step.
- (2) Find the new interface location based on the volume fractions obtained in step 1 using an interface reconstruction scheme. Some cells will be subdivided, and the values in each subcell must be initialized.
- (3) Take the same time interval as in step 1, but use the high-resolution wave-propagation method to update the cell averages of our model system such as (1) and (2) for single component flow or a fluid–mixture model proposed in [54] for multicomponent flow on the new non-uniform grid created in step 2.

Note that steps 1 and 2 constitute the basic volume-moving procedure for the modeling of the propagation of interfaces, and step 3 is simply the construction of approximate solutions to the physical problems of interests on a carefully chosen non-uniform grid. Since many of the key ingredients of the algorithm have been discussed before, see Sections 2 and 3, our goal here is to provide some more details with application to compressible flow problems.

4.1. Volume fraction update

Without loss of generality, suppose that Fig. 3c is the grid system we are working on at time t_n with the approximate state for the volume fraction of (3) defined as $Y_S^n = 0$ and $Y_S^n = 1$ when each of the grid cells

S is in the non-shaded and shaded regions of the graph, respectively. In the meantime, the state vector for our (single or multiple component) physical model, denoted by Q_S^n , is set by (5) to represent the approximate value of the cell average of the solution over any given cell S . The algorithm uses a less stringent time step which is determined by the CFL condition (8) based on the uniform grid sizes and the appropriate estimate of the wave speeds in Q_S^n .

Notice that in this setup the value of Y^n is only having jump with magnitude 1 across some polygonal edges of irregular cells, and is uniform elsewhere. Thus, in step 1, it should be reasonable to simply employ the wave-propagation method in first-order form (preferably the one with the transverse propagation of waves, see Section 2.1), to find the new volume fraction Y^{n+1} on this non-uniform grid at the next time step. Here the basic solution of the one-dimensional Riemann problem in the direction normal or tangential to the cell edge can be computed quite easily by solving the variable-coefficient linear transport equation (3) with the velocity field came from Q^n for the coefficient, and Y^n for the initial data (cf. [29]), yielding analogously the wave structure illustrated in Fig. 1 for $m = 1$ over a given time step Δt .

With the above comments in mind, Fig. 4a gives numerical result of a one-step calculation for the volume fraction on the uniform rectangular grid, where the computation is performed by assuming a constant velocity field $(u, v) = (1, 1)$ throughout the domain, and a time step $\Delta t = 0.06$. We note that in case the grid cell is an irregular one, the result present in the figure is computed by an area-weighted average of the original data on the subcells; this is necessary in the algorithm because the reconstruction of the new interface location to be done next depends on that. It is not difficult to show that our result is in good agreement as in comparison with the exact solution of this simple model problem.

4.2. New interface reconstruction

When the new volume fraction Y^{n+1} becomes available on the uniform grid, the approximate location of the tracked interface at the next time step can be obtained in principle by applying any interface reconstruction scheme described in Section 3 to the data set. If we now employ the PLIC method of Puckett to the case shown in Fig. 4a, for instance, we would have the disconnected set of linear segments as displayed in Fig. 4b for the representation of the new location of the old tracked interface over the time step $\Delta t = 0.06$. Surely, when comparing to the exact location of the interface, one may argue that our volume-based result does not seem to be superior to the result gotten by using such as a more typical surface-based approach for the same problem (not shown here, see [58] for more details). But as far as demonstrating the essential feature of the method is concerned, this simple example serves well for our purpose here (some sample results that demonstrate the feasibility of our method to practical problems will be present in Section 5). Nevertheless, we want to emphasize again that in our algorithm we do not assume the tracked interface to be the definitive location of the true

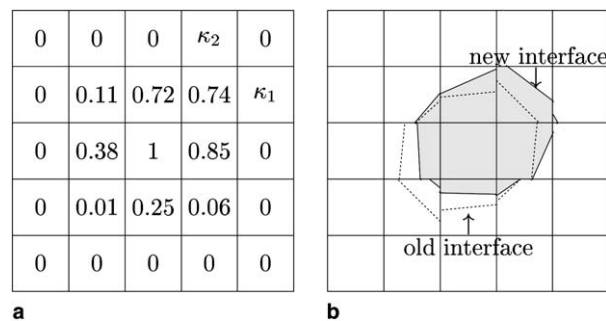


Fig. 4. An example of the volume-moving procedure in our interface tracking algorithm with a uniform velocity field $(u, v) = (1, 1)$ throughout the domain. (a) The updated volume fraction on the uniform rectangular grid based on the solution of (3) with the initial condition shown in Fig. 3c over a time step $\Delta t = 0.06$. Here $\kappa_1 = 5.7 \times 10^{-3}$ and $\kappa_2 = 1.3 \times 10^{-3}$. (b) New interface location obtained by using the PLIC method of Puckett from the volume fractions appeared in (a), where the dotted-line represents the initial location of the tracked interface. Note that the volume fraction at the next time step is set to be unity and zero for grid cells in the shaded and non-shaded region, respectively.

interface, but is only considered as an approximate location yielding a refined grid that is better able to represent the solution for the interface than the uniform grid alone.

It is clear that when a new interface is created, some cells will be subdivided, and the values in each subcell must be assigned. Here the method we use is implemented by taking a data structure that keeps the old and new grid systems separate, and as discussed in [31] doing so would be more efficient than the use of a data structure for an intermediate grid that contains both the old and new subdivisions. In this manner, when the subcell is in front and back of the (directed) new tracked interface within the grid cell, we set the volume fraction at the next time step to be zero and unity, respectively. This yields the new volume fraction that will be used in the next run of our algorithm.

In addition to that, in this work we also need to initialize the state variables for the compressible model of interests at the current time so that they can be updated in the subsequent step. For that, we should distinguish the situation on whether the new tracked interface is inserted into a regular cell or not. Firstly, suppose the new interface splits a regular grid cell A into two subcells B and C . In this instance, one simple way is to assign the cell averages Q_B^n and Q_C^n based on the value Q_A^n as

$$Q_B^n := Q_A^n, \quad Q_C^n := Q_A^n, \quad (17)$$

see [5,30] for some discussions on the other possibilities. Secondly, we have an irregular grid cell that is subdivided by the old interface into subcells A and B , and also by the new interface into the subcells C and D . Now the cell averages Q_C^n and Q_D^n may be initialized by the weighted averages of the values Q_A^n and Q_B^n as

$$Q_S^n := [\mathcal{M}(A \cap S)Q_A^n + \mathcal{M}(B \cap S)Q_B^n] / \mathcal{M}(S) \quad \text{for } S = C, D. \quad (18)$$

4.3. Physical solution update

Our final step of the algorithm is to find the cell averages of the solution of our (single or multicomponent) compressible model system on the new non-uniform grid that contains some of the cell edges as the approximate location of the new interface at the next time step. As before (cf. [31]), this is accomplished in the same manner as described in Section 2 for the solution update of hyperbolic conservation laws on a slightly non-uniform grid. That is to say, with a properly chosen Riemann solver (cf. [52–54]), we first solve the associated one-dimensional Riemann problems in the directions normal and tangential to each of the cell edges which belongs to the original grid system when entered into the algorithm for computation. Having done so, due to the fact that we have chosen the data structure and the initializations (17) and (18) so that there are no waves generated across any polygonal edges of the irregular cell to the new grid, we may then propagate each resulting wave independently of all others to update the cell averages of the aforementioned new grid in cells neighboring to the cell edge over a given time step.

It should be remarked that as in the work of Chern and Colella [5] for interface tracking, or Berger et al. [2] and Colella et al. [7] for handling irregular geometries in a Cartesian grid method, it is quite common to employ a post-processing step of the solution that reassigns the cell average of each irregular cells to a new one using an interpolation scheme based on the cell averages over an extended region neighboring to the grid cell under concerned. Here, in our algorithm, we take the same one-sided interpolation scheme as discussed in [5] for that matter, observing good behavior of the numerical solutions with that for many practical problems, see Section 5 for an example.

To end this section, we want to mention that the basic idea of the two-dimensional volume tracking algorithm described here can be extended in a straightforward manner to three space dimensions. That is, in steps 1 and 3 of the algorithm, we may now apply a variant of the three-dimensional wave-propagation method proposed by Langseth and LeVeque [23] on a slightly non-uniform Cartesian grid to update the volume fractions and the physical solutions, respectively. Clearly, one major endeavor to realize this is to devise a robust numerical procedure so that the measure (volume) of the cubical region swept out by the wave obtained from solving the one-dimensional Riemann problem over Δt and the neighboring hexahedral cells can be computed accurately and efficiently. It is fortunate however that in the field of computational geometry (cf. [8,11,39]) there have been quite a few tools developed for that purpose already, and so we may simply take and implement one of the popular approaches from there in our computer program. As to the interface-reconstruction

step of the algorithm, following the work appeared in [20,21], for instance, it can be done in a similar fashion without too much difficulty. It is important to note that to overcome the outstanding small cell problem when there is a tracked interface cutting through the underlying uniform grid, we are interested in the large time step method as usual. A detailed description of this three-dimensional algorithm (cf. [57]) will be reported elsewhere in the future.

5. Numerical results

We now present some sample numerical results obtained using our algorithm described in Section 4 for tracking material interfaces arising from compressible multicomponent problems in two dimensions. Without stated otherwise, we have carried out all the tests in this section using the Courant number $\nu = 0.9$ defined by (8), the Youngs' second method in the interface reconstruction scheme (for the sake of its simplicity and reasonably good prediction of the interface normal), and the MINMOD limiter in the high-resolution version of the finite-volume wave-propagation method. For comparison purposes, in many cases considered below, we have also included non-tracking results when the interface capturing version of the method is employed to the problems.

Example 5.1. To begin with, we consider a well-studied shock-bubble interaction problem that involves the collision of a shock wave in air with a circular gas bubble (cf. [14,33,44]). To setup the problem, we take a shock tube of size: $(x, y) \in [0, 445] \times [0, 89] \text{ mm}^2$, and use an initial condition that consists of a planar leftward-moving Mach 1.22 shock wave in air located at $x = 275 \text{ mm}$ traveling from right to left, and a stationary gas bubble with center $(x_0, y_0) = (225, 44.5) \text{ mm}$ and of radius $r_0 = 25 \text{ mm}$ in front of the shock which contains an R22 gas inside. Here both the air and R22 are modeled as perfect gases so that we have $\Gamma = \gamma - 1$ ($\gamma > 1$ denotes the usual ratio of specific heats) and $p_{\text{ref}} = e_{\text{ref}} = 0$ in the Mie–Grüneisen equation of state (2). For this problem, inside the R22 gas bubble the state variables are

$$(\rho, u, v, p, \gamma, Y) = (3.863 \text{ kg/m}^3, 0, 0, 1.01325 \times 10^5 \text{ Pa}, 1.249, 1),$$

and outside the bubble they are

$$(\rho, u, v, p, \gamma, Y) = (1.225 \text{ kg/m}^3, 0, 0, 1.01325 \times 10^5 \text{ Pa}, 1.4, 0)$$

and

$$(\rho, u, v, p, \gamma, Y) = (1.686 \text{ kg/m}^3, -113.5 \text{ m/s}, 0, 1.59 \times 10^5 \text{ Pa}, 1.4, 0)$$

in the preshock and postshock regions, respectively. We note that this gives us one example that the (air–R22) material interface is accelerated by a shock wave coming from the light-fluid to the heavy-fluid region, and the early stage of the resulting wave pattern after the interaction would consist of a transmitted shock wave, an interface, and a reflected shock wave.

To solve this two-fluid problem numerically, our algorithm (tracking or non-tracking version) uses a mixture type of the model equations proposed in [52] as the basis for approximation. Results of a sample run obtained using each of the tracking and non-tracking algorithms with an underlying uniform 3560×356 Cartesian grid are shown in Fig. 5 where a sequence of the schlieren-type images of the density in some short distances around the gas bubble is presented at nine different times $t = 55, 115, 135, 187, 247, 318, 342, 417$, and $1020 \mu\text{s}$ (measured relative to the time where the incident shock first hits the upstream bubble wall). From the figure, it is easy to observe the sensible improvement of the non-tracking (i.e., interface capturing) result near the interface when the volume tracking algorithm is employed in the computation. It is also interesting to see that, as far as the qualitative structure of the solution is concerned, our tracking results are in good agreement with the existing experimental and numerical solutions appeared in [44, Fig. 7]. The approximate location of the tracked interfaces for the volume tracking run is displayed in Fig. 6, noticing the complicated topological changes of the interfaces during the computations. We note that it would be a notorious task to do if we want to track the interface of this degree of complexity by using an alternative surface-based interface tracking method like the one discussed in [31], for example.

As in [44], to get a quantitative assessment of several prominent flow features, we make a diagnosis of the space–time locations of the incident shock wave, the upstream bubble wall, the downstream bubble wall, the

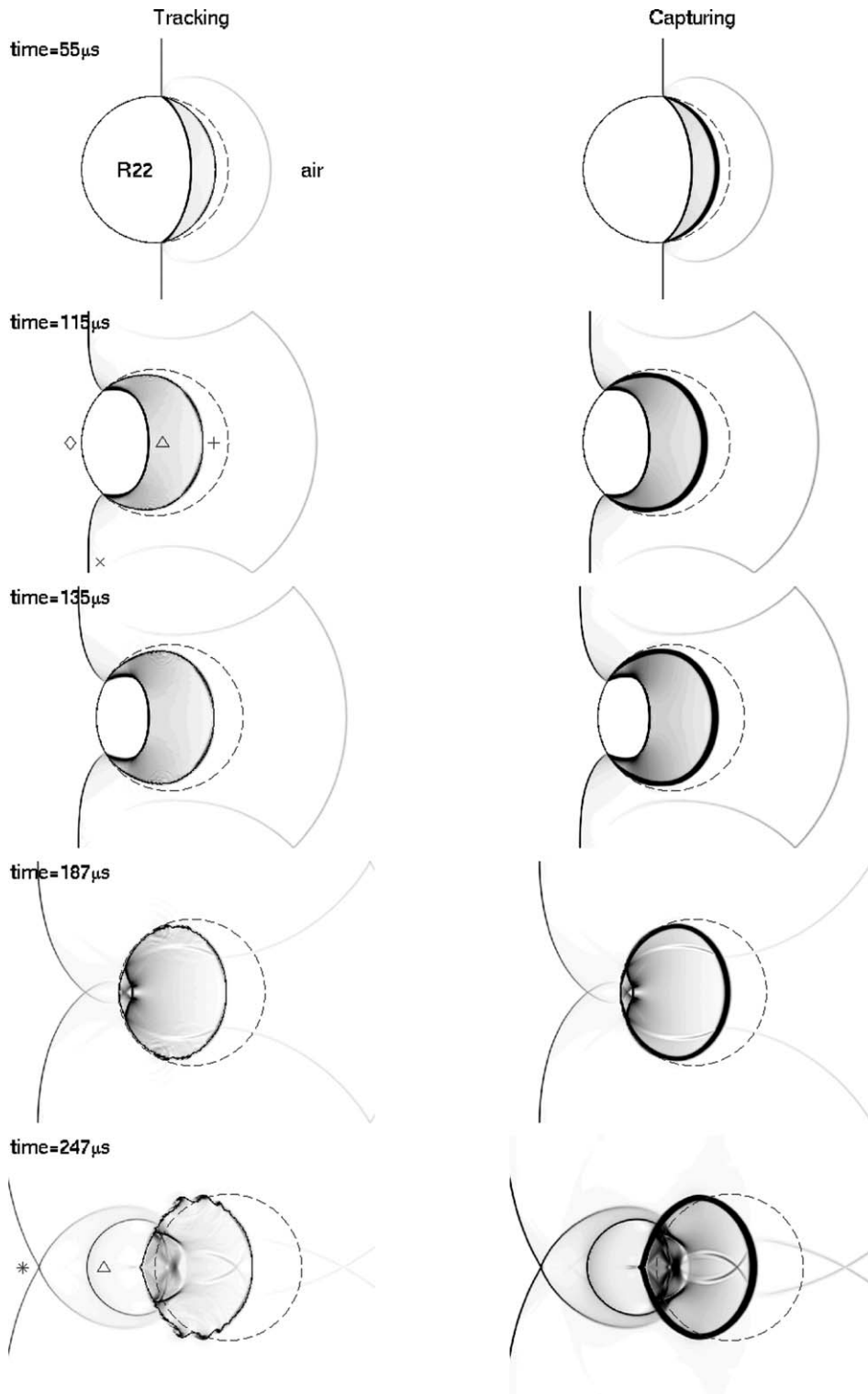


Fig. 5. Numerical results for a planar Mach 1.22 shock wave in air interacting with a circular R22 gas bubble. A sequence of the schlieren-type images of the density obtained using each of the tracking and non-tracking version of the methods is shown at nine different times $t = 55, 115, 135, 187, 247, 318, 342, 417$, and $1020 \mu\text{s}$ with an underlying uniform 3560×356 Cartesian grid.

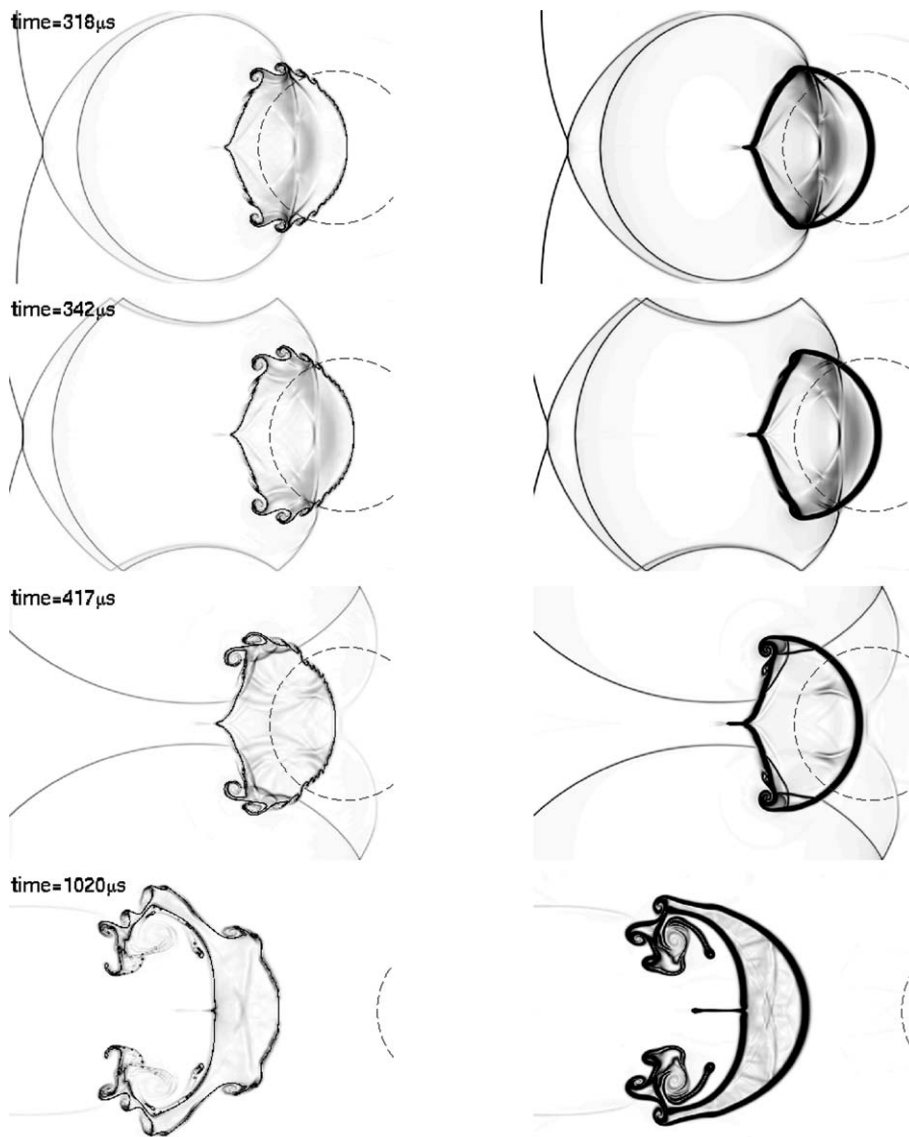


Fig. 5 (continued)

refracted shock wave, and the transmitted shock wave at some selected times, and report our findings in Fig. 7 only up to time $t = 250 \mu\text{s}$, see Fig. 5 for an illustration of the sample positions of the marked symbols plotted in the figure and also Fig. 12 of [44] for comparison. Here we obtain the position of the incident shock, marked by symbol “ \times ”, by checking the jumps in the pressure to see whether there are greater than a prescribed tolerance, at the horizontal sections of 5 mm above the bottom boundary. For the positions of the refracted shock (marked by symbol “ Δ ” when time $t \leq 202 \mu\text{s}$), and the two transmitted shocks (marked by symbols “ $*$ ” and “ \triangle ” when time $t > 202 \mu\text{s}$), we do the same thing as in the incident shock case, but the pressure jumps at the axis of symmetry are checked instead. Now, as to the positions of the upstream and downstream bubble walls which are marked by symbols “ $+$ ” and “ \diamond ”, respectively, there are two cases to be considered. That is, in the volume tracking run, we record the right-most and left-most of location of the tracking interface at the axis of symmetry, while in the non-tracking run we check the volume fractions to see if there are close to the midpoint value 1/2.

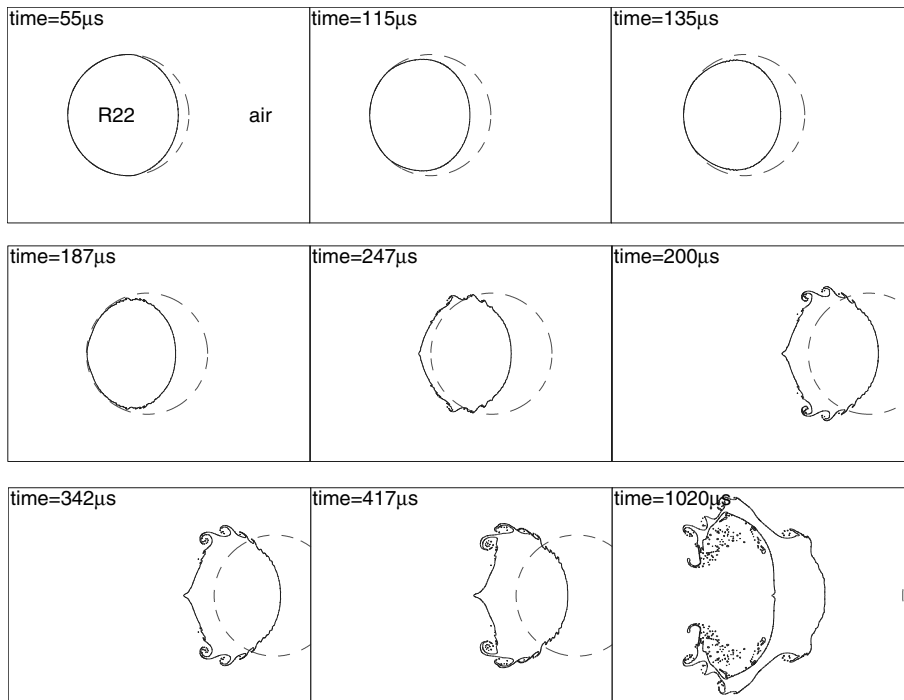


Fig. 6. Approximate locations of the interfaces for the volume tracking run shown in Fig. 5, where the dashed line in each subplot is the initial location of the air–R22 interface.

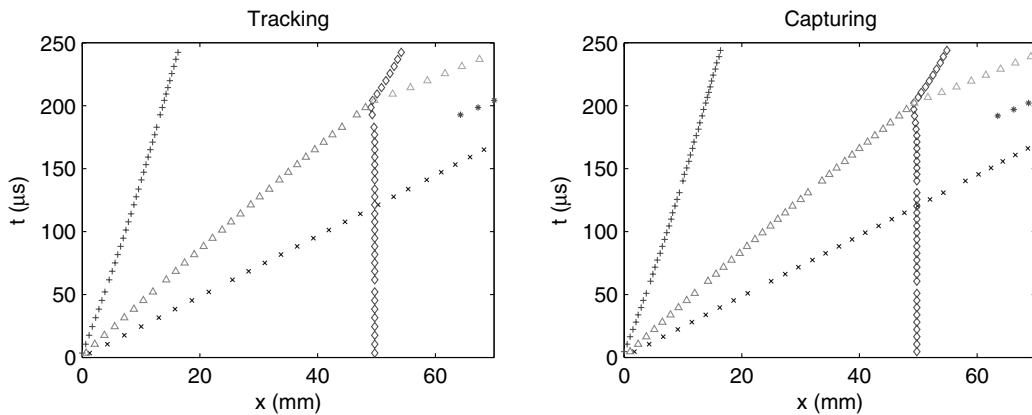


Fig. 7. Space-time locations of the incident shock wave (marked by symbol “x”), the upstream bubble wall (marked by symbol “+”), the downstream bubble wall (marked by symbol “◇”), the refracted shock (marked by symbol “△”), and the transmitted shock (marked by symbols “*” and “△”) for the runs shown in Fig. 5. These trajectories can be used to estimate the speed such as V_s , V_u , V_d , V_R , V_T in the order mentioned above.

With these trajectories, it is a common practice to perform a linear least-squares fit of each set of points in Fig. 7 separately, and take the slope of the respective line as one measure of the wave speed of interests. A comparison of the various computed velocities obtained using our tracking and non-tracking algorithms with those reported in the literature [14,44] is summarized in Table 1, where V_s represents the speed of the incident shock wave during the time $t \in [0, 250] \mu s$, V_{ui} (V_{uf}) is the speed of the initial (final) upstream bubble wall when time $t \in [0, 400] \mu s$ ($t \in [400, 1000] \mu s$), V_{di} (V_{df}) is the speed of the initial (final) downstream bubble wall when time $t \in [200, 400] \mu s$ ($t \in [400, 1000] \mu s$), V_R is the speed of the refracted shock when time $t \in [0, 202] \mu s$, and

Table 1

A comparison of the computed velocities obtained using our tracking and non-tracking algorithms for Example 5.1 with those reported in the literature [14,44]; see the text for the definition of the notations used in the table

Velocity (m/s)	V_s	V_R	V_T	V_{ui}	V_{uf}	V_{di}	V_{df}
Experiment [14]	415	240	540	73	90	78	78
Quirk and Karni [44]	420	254	560	74	90	116	82
Our result (tracking)	411	243	538	64	87	82	60
Our result (capturing)	411	244	534	65	86	98	76

V_T is the speed of the transmitted shock when time $t \in [202, 250] \mu s$. From the table, with reference to the experimental velocities, it is clear that our algorithm (either tracking or non-tracking) gives a better prediction of the speeds for shock waves such as V_s , V_R , and V_T than the numerical results of Quirk and Karni. In the other instance to the prediction of the speeds for bubble walls, we still observe a better agreement of the downstream case: V_{di} and V_{df} , but a slightly less accurate ones for the upstream case: V_{ui} and V_{uf} . It should be mentioned that because the needle-shape like bubble wall in the downstream is not being tracked by the algorithm, see Figs. 5 and 6, we do not find a good match of the speed V_{df} for that. Nevertheless, due to the sensitivity of the interface structure on the grid resolution used in the computation as well as the uncertainty on the experimental error, it should be fair to say that we have come up with a reasonable set of data for the bubble velocities in this application.

Finally, to see how our algorithms works for the pressure, in Fig. 8 we record the time history of the runs shown in Fig. 5 at six stations, $x_p = 3, 11, 27, 43, 67$, and 99 mm, downstream of the R22 gas bubble along the axis of symmetry. As compared this result with the one shown in Fig. 14 of [44], for example, we again notice

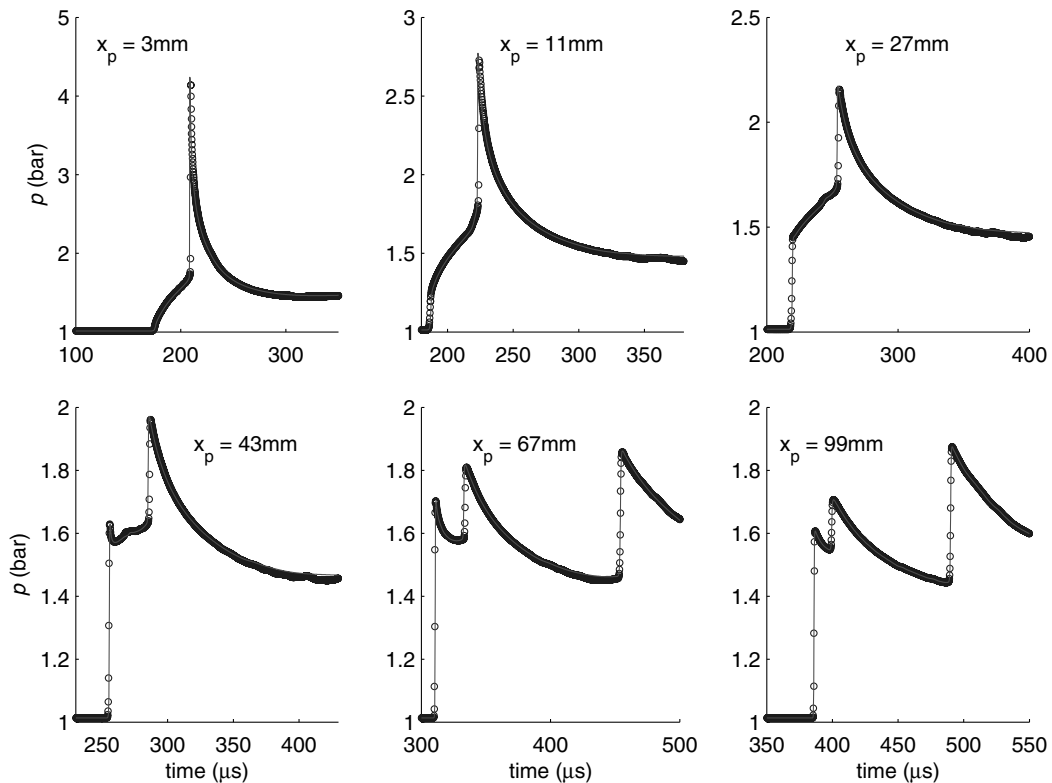


Fig. 8. Plot of the time history of the pressure for the runs shown in Fig. 5 at six stations, $x_p = 3, 11, 27, 43, 67$, and 99 mm, downstream of the R22 gas bubble along the axis of symmetry. The dotted points are the results obtained using the volume tracking method, and the solid lines are the results with interface capturing.

the sensible agreement of the solution behavior to the problem. Note that in carrying out the test we have used the solid wall boundary condition on the top and bottom, and the non-reflecting boundary condition on the left and right sides.

Example 5.2. Our next example of tracking material interfaces is concerned with a model underwater explosion problem (cf. [12,55]). In this test, we take a rectangular domain $(x, y) \in [-2, 2] \times [-1.5, 1] \text{ m}^2$, and consider the initial condition that is composed of a horizontal air–water interface at the $y = 0$ axis and a circular gas bubble in water with the center $(x_0, y_0) = (0, -0.3) \text{ m}$ and of the radius $r_0 = 0.12 \text{ m}$. Here above the air–water interface, the fluid is a perfect gas at the standard atmospheric condition,

$$(\rho, p, \gamma, Y) = (1.225 \text{ kg/m}^3, 1.01325 \times 10^5 \text{ Pa}, 1.4, 1).$$

Below the air–water interface, in region inside the gas bubble the fluid is a perfect gas also with the state variables

$$(\rho, p, \gamma, Y) = (1250 \text{ kg/m}^3, 10^9 \text{ Pa}, 1.4, 1),$$

and in region outside the gas bubble the fluid is water modeled by the simplified form of the Mie–Grüneisen equation of state (2) as

$$p = (\gamma - 1)\rho e - \gamma \mathcal{B}, \quad (19)$$

where \mathcal{B} is a pressure-like constant, and the state variables for that are

$$(\rho, p, \gamma, \mathcal{B}, Y) = (10^3 \text{ kg/m}^3, 1.01325 \times 10^5 \text{ Pa}, 4.4, 6 \times 10^8 \text{ Pa}, 0).$$

Note that, initially, all the fluid components of interests is in a stationary and equilibrium state, but due to the pressure difference between the fluids, breaking of the underwater bubble occurs instantaneously, yielding a circularly outward-going shock wave in water, an inward-going rarefaction wave in gas, and an interface lying in between that separates the gas and the water. Soon after this shock wave is diffracted through the nearby air–water interface, it is known (cf. [12,55]) that the topology of the underwater bubble will undergo a change from the original circular-shape to an oval-like shape. As time evolves, this gas bubble would continue rising upward, causing the subsequent deformation of the horizontal air–water interface.

Fig. 9 shows the schlieren-type images of the density and pressure at four different times $t = 0.2, 0.4, 0.8$, and 1.2 ms obtained using our algorithms (tracking and non-tracking) with a 400×250 grid. From the density plot, we clearly observe the improvement on the use of the volume tracking method over the non-tracking method to the sharpness of the solution structure near the interfaces. Moreover, from the pressure plot, we see the smooth variation of the solution near the interface, without introducing any spurious oscillations. As far as the global structure of the solution is concerned, our result agrees quite well with the result shown in [12], for instance, where a more sophisticated surface tracking method is used in the computation. In Fig. 10, we present the location of the tracked interfaces for the volume tracking run shown in 9. Note that here only the circular bubble interface is tracked initially, and the horizontal air–water interface is tracked in the algorithm only in time when its location deviates from the initial equilibrium position $y = 0$ axis which is aligned with the underlying grid. The cross-section of the density and pressure for the same run along line $x = 0$ is drawn in Fig. 11, giving some quantitative information about the differences of the tracking and non-tracking results at the selected times. The boundary conditions employed in the current computations are the solid wall boundary on the bottom, and the non-reflecting boundary on the remaining sides.

Example 5.3. To show how our algorithm works on tracking material interfaces with complex equation of state, we consider the simulation of the interaction of a shock wave in molybdenum with a block of encapsulated MORB (Mid-Ocean Ridge Basalt) liquid (cf. [37,54]). Here the computational region we take is a unit square domain, and we use an initial condition that is composed of a planar rightward-moving Mach 1.163 shock wave in molybdenum at $x = 0.3 \text{ m}$ traveling from left to right, and a rectangular region $[0.4, 0.7] \times [0, 0.5] \text{ m}^2$ of MORB liquid in front of the shock. For this problem, inside the region of the MORB liquid, we have the state variables

$$(\rho, u, v, p, Y) = (2260 \text{ kg/m}^3, 0, 0, 0, 1),$$

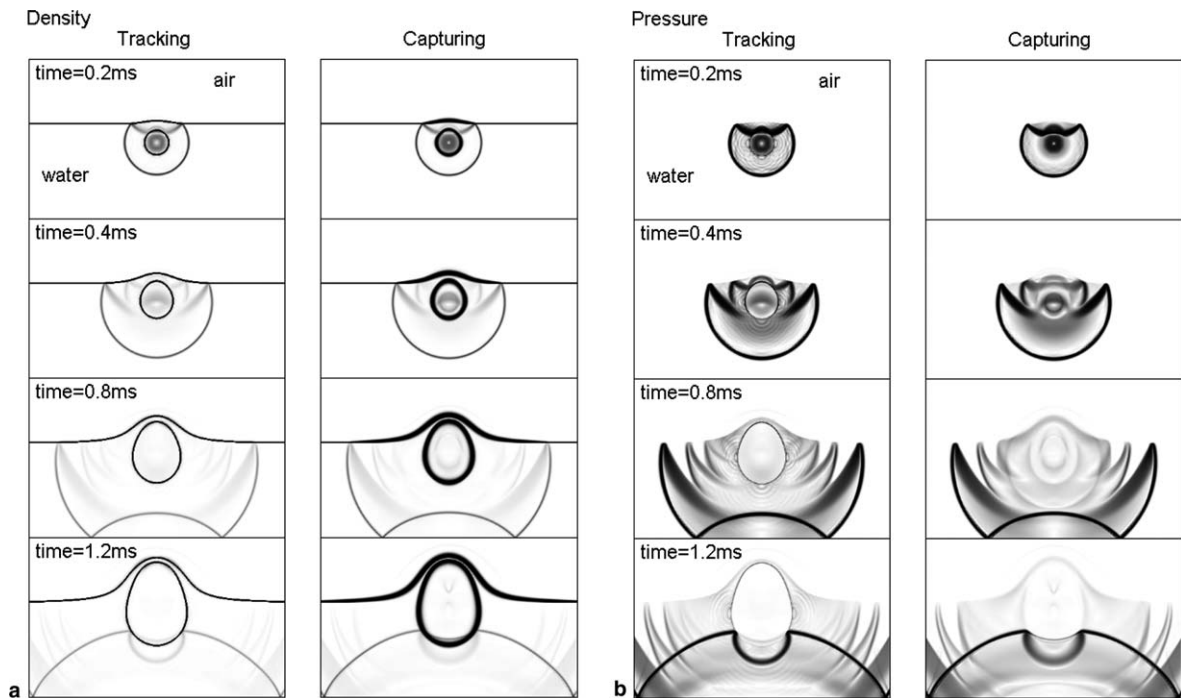


Fig. 9. Numerical results for the simulation of a model underwater explosion problem. Schlieren-type images are shown for (a) the density, and (b) the pressure at four different times $t = 0.2, 0.4, 0.8$, and 1.2 ms obtained using both the tracking and capturing version of the methods with a 400×250 grid.

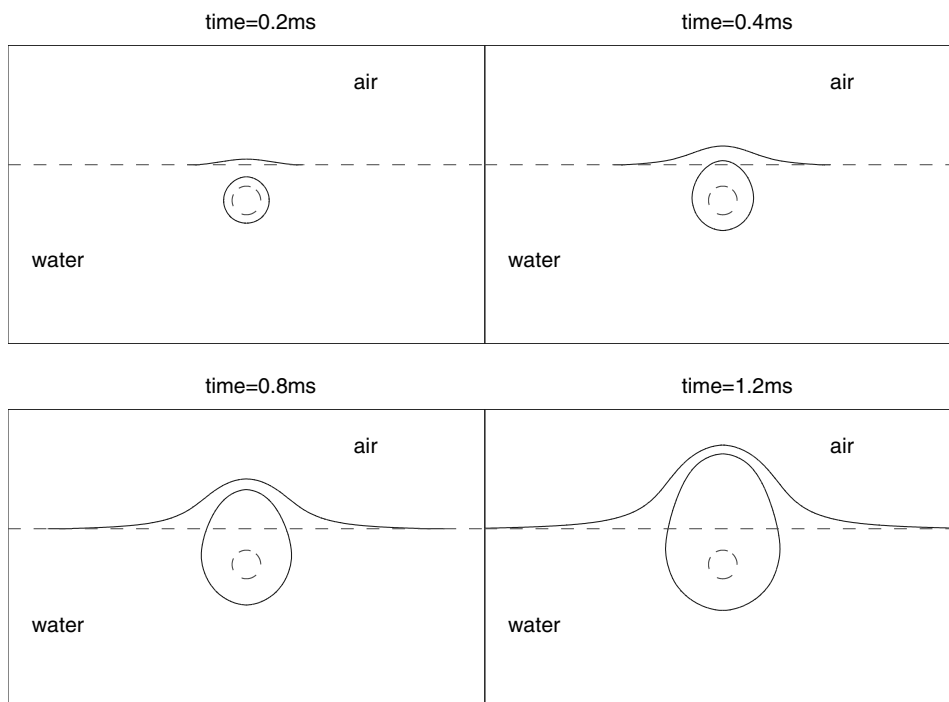


Fig. 10. Approximate locations of the tracked interface for the run shown in Fig. 9, where the dashed line in each subplot is the initial location of the horizontal and circular bubble interfaces at time $t = 0$.

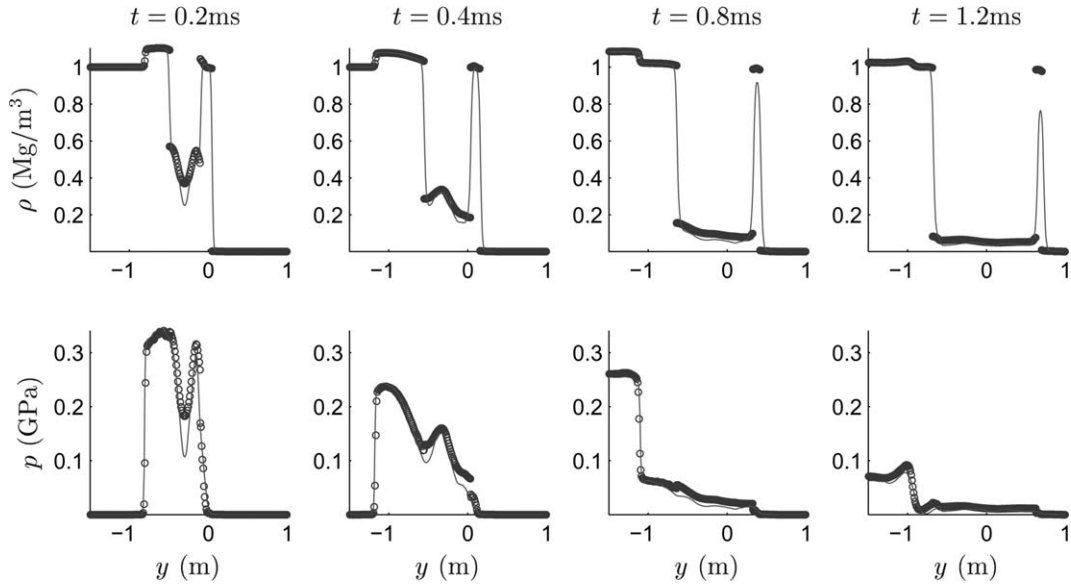


Fig. 11. Cross-sectional plots of the results for the run shown in Fig. 9 along $x = 0$, where the solid lines are the results obtained using the interface capturing method, and the dotted points are the tracking results.

and outside the MORB, we have the state variables in the preshock region

$$(\rho, u, v, p, Y) = (9961 \text{ kg/m}^3, 0, 0, 0, 0)$$

and the state variables in the postshock region

$$(\rho, u, v, p, Y) = (11,042 \text{ kg/m}^3, 543 \text{ m/s}, 0, 3 \times 10^{10} \text{ Pa}, 0).$$

As in [34,37,54], to model the thermodynamic behavior of solid media such as MORB and molybdenum, we use (2) together with the explicit expressions for Γ , p_{ref} , and e_{ref} as follows:

$$\begin{aligned} \Gamma(\rho) &= \Gamma_0 \left(\frac{\rho_0}{\rho} \right)^\alpha, \\ p_{\text{ref}}(\rho) &= p_0 + \frac{\rho_0 c_0^2 \left(1 - \frac{\rho_0}{\rho} \right)}{\left[1 - s \left(1 - \frac{\rho_0}{\rho} \right) \right]^2}, \\ e_{\text{ref}}(\rho) &= e_0 + \frac{1}{2\rho_0} \left(1 - \frac{\rho_0}{\rho} \right) (p_0 + p_{\text{ref}}(\rho)). \end{aligned} \quad (20)$$

We note that in above $\Gamma_0 = \gamma_0 - 1$ represents a reference Mie–Grüneisen coefficient at $\rho = \rho_0$, $\alpha \in [0, 1]$ is a dimensionless parameter, c_0 denotes the zero-pressure isentropic speed of sound, and s is a dimensionless parameter which is related to the pressure derivative of the isentropic bulk modulus $K_S = \rho(\partial p / \partial \rho)|_S$ by $(\partial K_S / \partial p)|_S = 4s - 1$ (cf. [49]). With the zero initial reference state for p_0 and e_0 , typical sets of material quantities for the molybdenum are

$$(\rho_0, c_0, s, \Gamma_0, \alpha) = (9961 \text{ kg/m}^3, 4770 \text{ m/s}, 1.43, 2.56, 1),$$

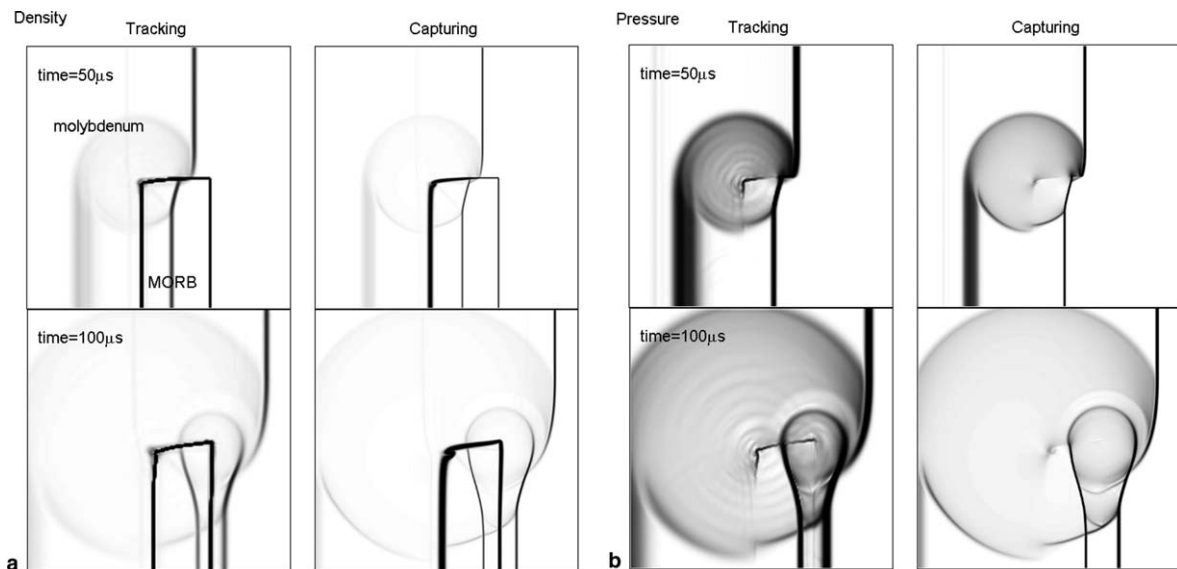
and for the MORB liquid are

$$(\rho_0, c_0, s, \Gamma_0, \alpha) = (2660 \text{ kg/m}^3, 2100 \text{ m/s}, 1.68, 1.18, 1).$$

It should be mentioned that, in contrast to [Example 5.1](#), here the rectangular (MORB–molybdenum) interface is accelerated by a shock wave coming from the heavy-fluid to the light-fluid region, and it is known that the early stage of the resulting wave pattern after the interaction would consist of a transmitted shock wave, an interface, and a reflected rarefaction wave (cf. [\[4,17\]](#)). To find the approximate solution of the current two-fluid problem with (20) in the Mie–Grüneisen equation of state (2), we use a fluid-mixture type of the model equations developed in [\[54\]](#) as the basis in our algorithm for numerical discretization.

[Fig. 12](#) shows the schlieren-type images of the density and pressure at two different times $t = 50$ and $100 \mu\text{s}$, where the computations are done using the tracking and non-tracking version of the methods with a uniform 200×200 grid and the solid wall boundary condition on the bottom, and the non-reflecting boundary condition on the remaining sides. From the displayed graphs of the density in [Fig. 12a](#), it is easy to observe the improved resolution of the solution near the interface, when our volume tracking method is employed in the test. In addition to that, from the pressure plots in [Fig. 12b](#), we find reasonable structure of the diffraction of a shock wave by the MORB liquid, and there is no spurious oscillation near the interface also, see [\[37\]](#) for a similar test but is done in cylindrical coordinates with a stronger incident shock than the one used here. The cross-section of the results for the same run along line $y = 0.4 \text{ m}$ is drawn in [Fig. 13](#), giving some quantitative information about the density and pressure of these two solutions at the selected times. [Fig. 14](#) gives the approximate location of the tracked interface for the volume tracking run, noticing the robustness of our algorithm to deal with the interface with real materials and when the interface is aligned with the grid cell boundary initially.

Example 5.4. Finally, as an example to show how our algorithm performs for tracking free surfaces, we are interested in a flying aluminum plate impact problem studied in [\[18,64\]](#), but is without the inclusion of the elastic–plastic condition for the solid aluminum. The computational domain we take for this problem is a 30 mm by 40 mm rectangular region in which on the right of a vertical interface at $x = 24 \text{ mm}$ we have the stationary aluminum target plate, while on the left of the domain we have an aluminum flyer which measures 5 mm by 12 mm on the center, traveling in free space with speed $u = 400 \text{ m/s}$ in the x -direction, see [Fig. 15](#) for an illustration of the setup. Initially, we assume that both the flying and target aluminum are in an uncompressed state, and are modeled by the same equation of state as MORB



[Fig. 12](#). Numerical results for a planar Mach 1.163 shock wave in molybdenum interacting with an encapsulated MORB liquid. Schlieren-type images are shown for (a) the density, and (b) the pressure at two different times $t = 50$ and $100 \mu\text{s}$ obtained using both the tracking and non-tracking version of the methods with a 200×200 grid.

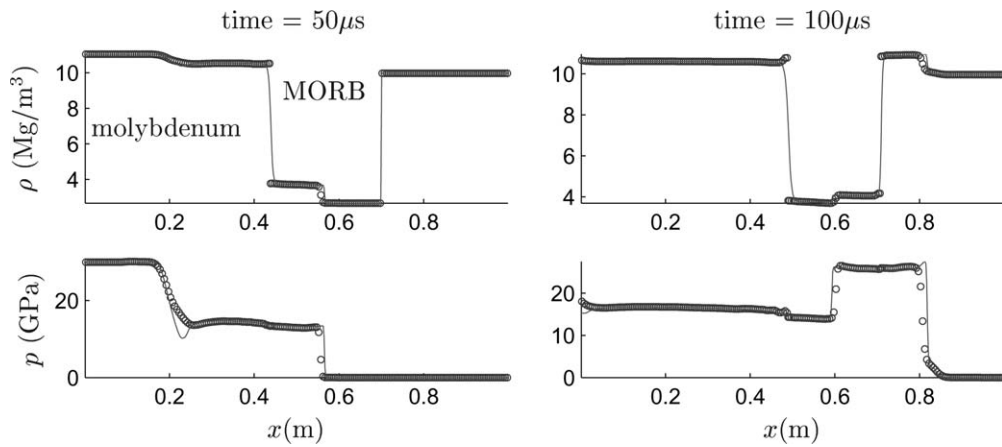


Fig. 13. Cross-sectional plots of the results for the run shown in Fig. 12 along the line $y = 0.4$ m, where the solid lines are the results obtained using the interface capturing method, and the dotted points are the tracking results.

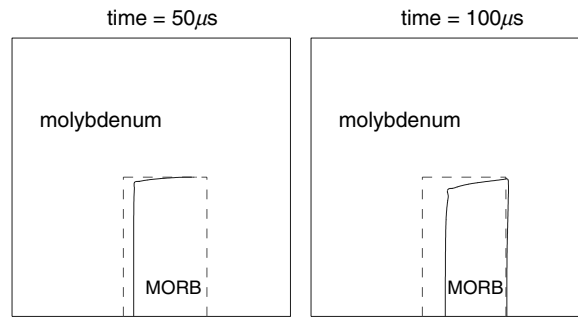


Fig. 14. Approximate locations of the tracked interfaces for the run shown in Fig. 12, where the dashed line in each subplot is the initial location of the MORB-molybdenum interface at time $t = 0$.

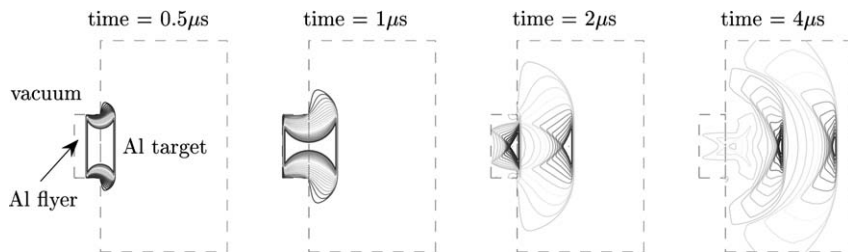


Fig. 15. Volume tracking results for a flying aluminum-plate problem with impact velocity 400 m/s. Contour plots of the pressure are shown at four different times $t = 0.5, 1, 2$, and $4\mu\text{s}$ using a 600×800 grid. The dashed lines in each subplot are the initial location of aluminum (flyer or target) plates at time $t = 0$.

and molybdenum considered before. In this case, the material quantities of an aluminum under the zero initial reference state for p_0 and e_0 take the following values (cf. [37]):

$$(\rho_0, c_0, s, \Gamma_0, \alpha) = (2785 \text{ kg/m}^3, 5238 \text{ m/s}, 1.338, 3, 1).$$

Since to the best of our knowledge there is not any well-established Eulerian-based “free-surface” (or called vacuum-boundary) capturing method available that can be used for the current application, it is only possible to carry out the computation using some sort of interface tracking for the free surface. In this case, it is customary to model the free space as a vacuum region with zero density and pressure inside the domain, and evolve the free surface with respect to time according to a speed of the vacuum boundary determined by the so-called vacuum Riemann problem, see [56,60–62], for example.

Fig. 15 shows the contours of the pressure at four different times $t = 0.5, 1, 2$, and $4 \mu\text{s}$ obtained using our volume tracking method with a 600×800 grid. From the figure, when the flying plate first strikes the target plate, we observe clearly the occurrence of a transmitted shock moving rightward into the target as well as a reflected shock traveling leftward back into the flyer. Note that since the flyer and the target plates share common material properties, these shock waves are in fact identical in view of the basic structure of the solution. As the time proceeds, while the reflected shock wave in the flyer is approaching the rear of the flyer, resulting in a returned rarefaction wave subsequently, the rarefactions generated from the free surface–aluminum interfaces parallel to the x -axis are about to collide at the center line of the y -axis also. In a somewhat later time of the solution process, the rarefaction waves behind the aforementioned transmitted shock have begun to overtake the shock, yielding a significant reduction of the pressure near the shock front, see Fig. 16 for a cross-sectional plots of the results for the same run along the center line $y = 20 \text{ mm}$ for illustrations. If we now compare our results to the hydrodynamic solution appeared in [64] for the same problem, but is using a simplified Mie–Grüneisen equation of state and a FCT-based volume tracking scheme, we would find reasonable qualitative agreement of these two solutions.

The results for the approximate location of the tracked free surface are presented in Fig. 17 which clearly shows the viability of our volume tracking algorithm for this problem. It should be mentioned that if we use a higher impact velocity to the computation, say with $u = 2000 \text{ m/s}$, we obtain essentially the same global structure of the solution, but is with stronger shock waves and the more deformation of the back surface of the flying plate (not shown here).

To end this section, it should be mentioned that all the simulations done here were run on an AlphaServer DS20E under the Tru64 Unix operating system. As the whole, it is observed that, the CPU time that requires

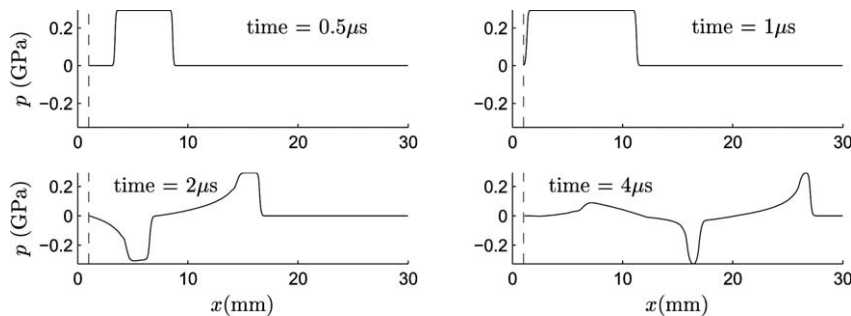


Fig. 16. Cross-sectional plots of the results for the run shown in Fig. 15 along the center line $y = 20 \text{ mm}$, where the dashed line in each subplot is the approximate location of the free surface.

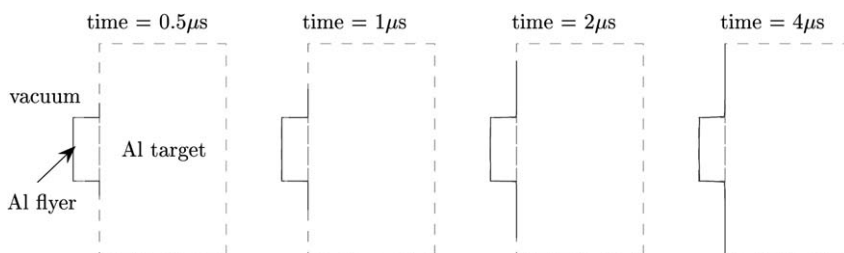


Fig. 17. Approximate locations of the tracked interfaces (solid lines) for the run shown in Fig. 15.

for carrying out the interface-tracking computations is about 16% more than the CPU time that requires for carrying out the interface-capturing computations. But since in the former case we have a better resolution on the structure of the interfaces than those in the latter case, this extra amount of the computing time is worth spending. We note that the computer programs (written in Fortran 77) used to obtain the results above are available from the author.

6. Conclusion

We have presented a volume-of-fluid interface tracking algorithm for the numerical simulation of compressible multicomponent problems in two space dimensions. The algorithm is based on a uniform Cartesian grid with some grid cells subdivided by tracked interfaces, approximately aligned with the material interfaces in the flow field. The new location of the interface is evolved according to a simple volume-moving procedure from the current time to the next which is known to be more robust than the use of a standard surface-moving procedure when the topology of the interface is complex, for instance. In the algorithm, the cell averages in the resulting slightly non-uniform grid is updated by employing a finite volume method based on a wave-propagation formulation, which is very easy to implement and can give stable and accurate results even in the presence of small cells and the use of a time step with respect to the uniform grid cells. Various numerical results are present in this paper that demonstrates the feasibility of the proposed volume tracking algorithm as applied to a reasonable class of compressible hydrodynamics problems with real materials characterized by a Mie–Grüneisen equation of state.

One of the ongoing works is to improve upon the method so that the elastic-plastic condition of the solid materials can be included in the simulation of practical impact problems, see [18,35,36,65,68] for some recent development in this direction. Besides that, the study of the capillary effects to the compressible multicomponent flow problems (cf. [41]) will be of great interests also.

Acknowledgment

This work was supported in part by National Science Council of Republic of China Grants NSC-89-2115-M-002-033, 90-2115-M-002-022, and 91-2115-M-002-016.

References

- [1] J.B. Bell, P. Colella, M.L. Welcome, Conservative front-tracking for inviscid compressible flow, *UCRL-JC-105251, preprint*, 1991. Present at AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June 24–26.
- [2] M.J. Berger, C. Helzel, R.J. LeVeque, H-box methods for the approximation of hyperbolic conservation laws on irregular grids, *SIAM J. Numer. Anal.* 41 (2003) 893–918.
- [3] R.P. Brent, *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, 1972.
- [4] T. Chang, L. Hsiao, *The Riemann Problem and Interaction of Waves in Gas Dynamics*, Longman Scientific & Technical, England, 1989.
- [5] I.-L. Chern, P. Colella, A conservative front tracking method for hyperbolic systems of conservation laws, *UCRL-97200, LLNL*, 1987.
- [6] I.-L. Chern, J. Glimm, O.A. McBryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics, *J. Comput. Phys.* 63 (1986) 83–110.
- [7] P. Colella, D.T. Graves, B.J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, *J. Comput. Phys.* 211 (2006) 347–366.
- [8] J.K. Dukowicz, Efficient volume computation for three-dimensional hexahedral cells, *J. Comput. Phys.* 74 (1988) 493–496.
- [9] J. Glimm, J. Grove, B. Lindquist, O.A. McBryan, G. Tryggvason, The bifurcation of tracked scalar waves, *SIAM J. Sci. Stat. Comput.* 9 (1) (1988) 61–79.
- [10] E. Godlewski, P.-A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, Applied Mathematical Science, vol. 118, Springer, Berlin, 1996.
- [11] J.E. Goodman, J. O'Rourke, *Handbook of Discrete and Computational Geometry*, CRC Press, New York, 1997.
- [12] J. Grove, R. Menikoff, The anomalous reflection of a shock wave at a material interface, *J. Fluid Mech.* 219 (1990) 313–336.
- [13] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, *J. Comput. Phys.* 152 (1999) 423–456.
- [14] J.-F. Haas, B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, *J. Fluid Mech.* 181 (1987) 41–76.
- [15] D.J.E. Harvie, D.F. Fletcher, A new volume of fluid advection algorithm: the stream scheme, *J. Comput. Phys.* 162 (2000) 1–32.

- [16] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* 39 (1981) 201–225.
- [17] R.L. Holmes, A Numerical Investigation of the Richtmyer–Meshkov Instability Using Front Tracking, Ph.D. thesis, SUNY at Stony Brook, August, 1994 (unpublished).
- [18] B.P. Howell, G.J. Ball, A free-Lagrange augmented Godunov method for the simulation of elastic–plastic solids, *J. Comput. Phys.* 175 (2002) 128–167.
- [19] J.M. Hyman, Numerical methods for tracking interfaces, *Physica D* 12 (1984) 396–407.
- [20] D.B. Kothe, W.J. Rider, S.J. Mosso, J.S. Brock, J.I. Hochstein, Volume tracking of interfaces having surface tension in two and three dimensions, *AIAA-1996-0859*, 1996, in: 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, January 15–18, 1996.
- [21] D.B. Kothe, M.W. Williams, K.L. Lam, D.R. Korzekwa, P.K. Tubesing, E.G. Puckett, A second order accurate, linearity-preserving volume tracking algorithm for free surface flow on 3-d unstructured meshes, in: Proceedings of the 3rd ASM/JSME Joint Fluids Engineering Conference, San Francisco, CA, July 18–22, 1999.
- [22] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, G. Zanetti, Modelling merging and fragmentation in multiphase flows with SURFER, *J. Comput. Phys.* 113 (1994) 134–147.
- [23] J.O. Langseth, R.J. LeVeque, A wave propagation method for three-dimensional hyperbolic conservation laws, *J. Comput. Phys.* 165 (2000) 126–166.
- [24] R.J. LeVeque, Large time step shock-capturing techniques for scalar conservation laws, *SIAM J. Numer. Anal.* 19 (6) (1982) 1091–1109.
- [25] R.J. LeVeque, A large time step generalization of Godunov’s method for systems of conservation laws, *SIAM J. Numer. Anal.* 22 (6) (1985) 1051–1073.
- [26] R.J. LeVeque, High resolution finite volume methods on arbitrary grids via wave propagation, *J. Comput. Phys.* 78 (1988) 36–63.
- [27] R.J. LeVeque, Hyperbolic conservation laws and numerical methods, in: Von Karman Institute for Fluid Dynamics, Lecture series on computational fluid dynamics 1990-03, 1990.
- [28] R.J. LeVeque, Wave propagation algorithms for multi-dimensional hyperbolic systems, *J. Comput. Phys.* 131 (1997) 327–353.
- [29] R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, New York, 2002.
- [30] R.J. LeVeque, K.-M. Shyue, One-dimensional front tracking based on high resolution wave propagation methods, *SIAM J. Sci. Comput.* 16 (1995) 348–377.
- [31] R.J. LeVeque, K.-M. Shyue, Two-dimensional front tracking based on high resolution wave propagation methods, *J. Comput. Phys.* 123 (1996) 354–368.
- [32] H. Luo, J.D. Baum, R. Löhner, On the computation of multi-material flows using ALE formulation, *J. Comput. Phys.* 194 (2004) 304–328.
- [33] A. Marquina, P. Mulet, A flux-split algorithm applied to conservative models for multicomponent compressible flows, *J. Comput. Phys.* 185 (2003) 120–138.
- [34] R.G. McQueen, S.P. Marsh, J.W. Taylor, J.N. Fritz, W.J. Carter, The equation of state of solids from shock wave studies, in: R. Kinslow (Ed.), *High Velocity Impact Phenomena*, Academic Press, San Diego, 1970, pp. 293–417.
- [35] G.H. Miller, P. Colella, A high-order Eulerian Godunov method for elastic–plastic flow in solids, *J. Comput. Phys.* 167 (2001) 131–176.
- [36] G.H. Miller, P. Colella, A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing, *J. Comput. Phys.* 183 (2002) 26–82.
- [37] G.H. Miller, E.G. Puckett, A high order Godunov method for multiple condensed phases, *J. Comput. Phys.* 128 (1996) 134–164.
- [38] E.S. Oran, J.P. Boris, *Numerical Simulation of Reactive Flow*, Elsevier, New York, 1987.
- [39] J. O’Rourke, *Computational Geometry in C*, Cambridge University Press, New York, 1994.
- [40] B.J. Parker, D.L. Youngs, Two and three dimensional Eulerian simulation of fluid flow with material interfaces, Atomic Weapons Establishment (AWE), Preprint No. 01/92, 1992.
- [41] G. Perigaud, R. Saurel, A compressible flow model with capillary effects, *J. Comput. Phys.* 209 (2005) 139–178.
- [42] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, *J. Comput. Phys.* 199 (2004) 465–502.
- [43] E.G. Puckett, A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction, in: Proceedings of the 4th International Conference on Computational Fluid Dynamics, Davis, CA, September 9–12, 1991.
- [44] J.J. Quirk, S. Karni, On the dynamics of a shock-bubble interaction, *J. Fluid Mech.* 318 (1996) 129–163.
- [45] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, *J. Comput. Phys.* 141 (1998) 112–152.
- [46] P.L. Roe, Approximate Riemann solvers, parameter vector, and difference scheme, *J. Comput. Phys.* 43 (1981) 357–372.
- [47] M. Rudman, Volume tracking methods for interfacial flow calculations, *Intl. J. Numer. Meth. Fluids* 24 (1997) 671–691.
- [48] M. Rudman, A volume-tracking method for incompressible multifluid flows with large density variations, *Intl. J. Numer. Meth. Fluids* 28 (1998) 357–378.
- [49] A.L. Ruoff, Linear shock-velocity-particle-velocity relationship, *J. Appl. Phys.* 38 (13) (1967) 4976–4980.
- [50] K. Shahbazi, M. Paraschivoiu, J. Mostaghimi, Second order accurate volume tracking based on remapping for triangular meshes, *J. Comput. Phys.* 188 (2003) 100–122.
- [51] K.-M. Shyue, Front tracking based on high-resolution wave propagation methods, in: W. Gautschi (Ed.), *Proceedings of the Symposium on Applied Mathematics*, vol. 48, American Mathematical Society, 1994, pp. 383–387.
- [52] K.-M. Shyue, An efficient shock-capturing algorithm for compressible multicomponent problems, *J. Comput. Phys.* 142 (1998) 208–242.

- [53] K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with van der Waals equation of state, *J. Comput. Phys.* 156 (1999) 43–88.
- [54] K.-M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with Mie–Grüneisen equation of state, *J. Comput. Phys.* 171 (2001) 678–707.
- [55] K.-M. Shyue, A simple volume tracking method for compressible two-phase flow, *J. Korean Astron. Soc.* 34 (2001) S237–S241.
- [56] K.-M. Shyue, A simple multifluid algorithm for elastic-plastic flow in solids: One-dimensional case (in preparation).
- [57] K.-M. Shyue, A wave-propagation based volume tracking method for compressible multicomponent flow in three space dimensions (in preparation).
- [58] K.-M. Shyue, *Front Tracking Methods based on Wave Propagation*, Ph.D. thesis, University of Washington, June, 1993 (unpublished). Available from: <http://www.math.ntu.edu.tw/~shyue/preprints/thesis.pdf>.
- [59] 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.
- [60] H.S. Tang, D. Huang, A second-order accurate capturing scheme for 1D inviscid flows of gas and water with vacuum zones, *J. Comput. Phys.* 128 (1996) 301–318.
- [61] H.S. Tang, F. Sotiropoulos, A second-order Godunov method for wave problems in coupled solid-water-gas systems, *J. Comput. Phys.* 151 (1999) 790–815.
- [62] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, second ed., Springer, Berlin, 1999.
- [63] L.N. Trefethen, D. Bau III, *Numerical Linear Algebra*, SIAM, Philadelphia, 1997.
- [64] M.B. Tyndall, *Numerical modelling of shock waves in solid materials*, Ph.D. thesis, Monash University, 1991 (unpublished).
- [65] H.S. Udaykumar, L. Tran, D.M. Belk, K.J. Vanden, An Eulerian method for computation of multimaterial impact with ENO shock-capturing and sharp interfaces, *J. Comput. Phys.* 186 (2003) 136–177.
- [66] A.B. Wardlaw Jr., H.U. Mair, Spherical solutions of an underwater explosion bubble, *Shock Vib.* 5 (1998) 89–102.
- [67] S.W.J. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, *J. Comput. Phys.* 160 (2000) 662–682.
- [68] M.L. Wilkins, *Computer Simulation of Dynamic Phenomena*, Springer, New York, 1999.
- [69] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, *J. Comput. Phys.* 169 (2001) 556–593.
- [70] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, in: K.W. Morton, M.J. Baines (Eds.), *Numerical Methods for Fluid Dynamics*, Academic Press, San Diego, 1982, pp. 273–285.
- [71] D.L. Youngs, An interface tracking code for a 3D Eulerian hydrodynamics code, Technical Report 44/92/35, AWRE, 1984.