Commun. Comput. Phys.
doi: 10.4208/cicp.260314.310315a

Vol. **19**, No. 1, pp. 124-142
January 2016

A Front Tracking Method for the Simulation of Compressible Multimedium Flows

Haitian Lu[1], Ning Zhao[1,*] and Donghong Wang[2]

[1] *College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China.*
[2] *College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China.*

Communicated by Chi-Wang Shu

Received 26 March 2014; Accepted (in revised version) 31 March 2015

Abstract. A front tracking method combined with the real ghost fluid method (RGFM) is proposed for simulations of fluid interfaces in two-dimensional compressible flows. In this paper the Riemann problem is constructed along the normal direction of interface and the corresponding Riemann solutions are used to track fluid interfaces. The interface boundary conditions are defined by the RGFM, and the fluid interfaces are explicitly tracked by several connected marker points. The Riemann solutions are also used directly to update the flow states on both sides of the interface in the RGFM. In order to validate the accuracy and capacity of the new method, extensive numerical tests including the bubble advection, the Sod tube, the shock-bubble interaction, the Richtmyer-Meshkov instability and the gas-water interface, are simulated by using the Euler equations. The computational results are also compared with earlier computational studies and it shows good agreements including the compressible gas-water system with large density differences.

AMS subject classifications: 35L65, 76T10, 76T99

PACS: 47.10.ab, 47.11.-j, 47.55.Ca

Key words: Front tracking method, real ghost fluid method, compressible flows, Riemann problem, fluid interfaces.

1 Introduction

The dynamics of multimedium flow has become a research hotspot nowadays for its significant applications in many engineering fields. The investigation of the multimedium

*Corresponding author. *Email addresses:* lhtgkzy@126.com (H. Lu), zhaoam@nuaa.edu.cn (N. Zhao), wangdonghong@nuaa.edu.cn (D. Wang)

flow behaviors in the micro- and nanochannels [25], for example, can be applied to colloidal and interfacial systems, such as bioassays, microreactors, emulsification and encapsulation, etc.

As for the computation of multimedium flows, a relatively dominant part is the treatment of moving material interface and its vicinity. Among the numerical simulations used to compute the contact discontinuities based on Euler framework, two basic approaches may be distinguished. One is the shock capturing method and the other is the front tracking method. Early algorithms have treated the material interfaces with the γ-based model [14], the mass fraction model [1, 15], the volume of fluid [2] or a level set function [19, 21]. These algorithms, based on shock capturing methods, always yield a numerical diffusion of contact discontinuities over several nodes. However, for the front tracking method [8–10], fluid interfaces are explicitly tracked by connected marker points and a sharp interface boundary is maintained during the computation. In this method, a Riemann problem is constructed near the fluid interfaces to handle the sharp jump across the interface. Similar idea based on Riemann solutions can also be found in the study of Cocchi et al. [5]. In their algorithm, a Riemann problem is solved at material interfaces in the corrector step and the Riemann solutions are used to correct the results from the predictor step which generally generates spurious oscillations or numerical diffusion. The method proposed by Tryggvason et al. [28] is in fact a hybrid between front capturing and front tracking since a stationary regular grid is used for fluid flow and the interface is tracked by a separate grid of lower dimension.

The basic ideas of ghost nodes have more recently been used in the ghost fluid method (GFM) introduced by Fedkiw et al. [4, 6, 7]. In the GFM, the discontinuous physical variables, such as entropy, are extrapolated across the interface which can efficiently prevent pressure oscillations and reduce smearing of discontinuous variables. The GFM is robust and can easily handle fluids with different material properties. However, when the pressure or the velocity experiences a large gradient across the interface, the GFM cannot work well. Indeed, the ghost fluid states should consider the influence of both wave interaction and material properties on the interfacial evolution. This leads to the proposal of improved versions of GFM, for example, the modified ghost fluid method (MGFM) and the real ghost fluid method (RGFM). In the MGFM, Riemann problem is defined and solved approximately to predict interface states [17, 18]. The predicted interfacial states are then used to define the ghost fluid states. The RGFM described in [30] further predicts the flow states for the real fluid nodes just next to the interface because wave interaction at the interface can propagate upward and downward simultaneously. The RGFM enables a better imposition of interface boundary conditions and less conservative errors, especially for those critical problems of shock impendence matching.

The purpose of this paper is to simulate the multimedium flows in a more accurate and simple way. As is well known, the way to track the interface and define the interface boundary conditions plays the critical role in the numerical simulations. With legitimate concerns over these factors, we combine a front tracking method with RGFM (RGFM-FT) in this paper. Unlike the front tracking method in [29] where the marker points are

moved with fluid velocities interpolated from fixed grid, here we propose the method based on the Riemann problem constructed near the interface to determine the motion of interface. In addition, the new method is easy in implementation where the flow states updating in the RGFM [30] are all done by the Riemann solutions obtained at the marker point. The organization of this paper is as follows: in Section 2 and Section 3, we give the algorithms of advancing the material interface procedure in detail and describe the implementation in RGFM with the Riemann solutions. Extensive numerical examples for simulating the multimedium flow problems are presented in Section 4 to verify the effectiveness of these procedures. Concluding remarks are then given in Section 5.

2 Governing equations

The governing equations are the two-dimensional compressible Euler equations written in Cartesian coordinate:

$$\frac{\partial \vec{U}}{\partial t}+\nabla\cdot\vec{F}(\vec{U})=0, \tag{2.1}$$

where $\vec{U}=[\rho,\rho u,\rho v,E]^T$, $\vec{F}(\vec{U})=[\vec{F_1}(\vec{U}),\vec{F_2}(\vec{U})]$, $\vec{F_1}(\vec{U})=[\rho u,\rho u^2+p,\rho uv,(E+p)u]^T$, $\vec{F_2}(\vec{U})=[\rho v,\rho uv,\rho v^2+p,(E+p)v]^T$. Here ρ is the density, u and v are the velocities, p is the pressure, E is the total energy per unit volume. The total energy is given as:

$$E=\rho e+\rho(u^2+v^2)/2, \tag{2.2}$$

where e is the internal energy per unit mass.

For closure of the system, the equation of state (EOS) is required. The EOS for gas or liquid medium can be written uniformly as:

$$p=(\gamma-1)\rho e-\gamma B, \tag{2.3}$$

where γ and B are treated as fluid constants, and will be specified in Section 4.

3 Numerical procedures

3.1 Solving the compressible Euler equations

The motion of the fluid interfaces and the determination of Euler equations are treated separately. For solving the compressible Euler equations, the spatial discretization is treated by the fifth-order weighted essentially non-oscillatory with Lax-Friedrichs flux scheme (WENO-LF) [13]. The time integration is solved by a third-order TVD Runge-Kutta scheme [24] and the time step Δt which should satisfy the CFL condition is:

$$\Delta t=\min_{(i,j)}\left\{\frac{CFL}{((|u|+a)/\Delta x+(|v|+a)/\Delta y)}\right\}, \tag{3.1}$$

where Δx and Δy are the grid spacing, a is the sound speed and (i,j) represents all the grid points in the flow field.

3.2 Tracking fluid interfaces

As indicated in Fig. 1, the medium 1 and the medium 2 are separated by the fluid interface at time t^n. The marker points are the intersections of the interface and the grid lines. $\vec{N}$ is the normal vector and $\vec{T}$ is the tangential vector, respectively. The orientation of normal vector $\vec{N}$ is from the medium 1 to the medium 2. Point A and point B are in different mediums, obtained by the distance $\triangle n$ from the marker point P. Here:

$$\triangle n = \left[\left(\frac{N_x}{\triangle x} \right)^2 + \left(\frac{N_y}{\triangle y} \right)^2 \right]^{-\frac{1}{2}}, \tag{3.2}$$

where N_x and N_y are the components of unit normal vector at the marker point P. Point A and point B are all in the normal direction of the marker point P. Here, the density, the normal velocity and the pressure at point A and point B are denoted by the statevectors: $\vec{W}_A = [\rho^A, u_N^A, p^A]^T$ and $\vec{W}_B = [\rho^B, u_N^B, p^B]^T$, which can be obtained by the inverse distance weighting interpolation from the fixed grids in the same medium nearby:

$$\psi = \left(\sum_{i=1}^{m} \frac{\psi_i}{r_i} \right) \Big/ \left(\sum_{i=1}^{m} \frac{1}{r_i} \right), \tag{3.3}$$

where ψ_i is the density, the normal velocity or the pressure and r_i is the distance. The inverse distance weighting interpolation has no special requirements for the number of interpolation points, and therefore it is very suitable for this problem. The Riemann problem can be constructed if $\vec{W}_A$ and $\vec{W}_B$ are taken as the initial conditions and an approximate Riemann problem solver (ARPS) based on a two shock structure can be applied to obtain the Riemann solutions [16]. As the tangential velocity of the marker point P is not determined accurately from the Riemann problem, the method in [5] is used. Suppose that the tangential velocity is equal to the velocity of one of the two mediums at time t^n.

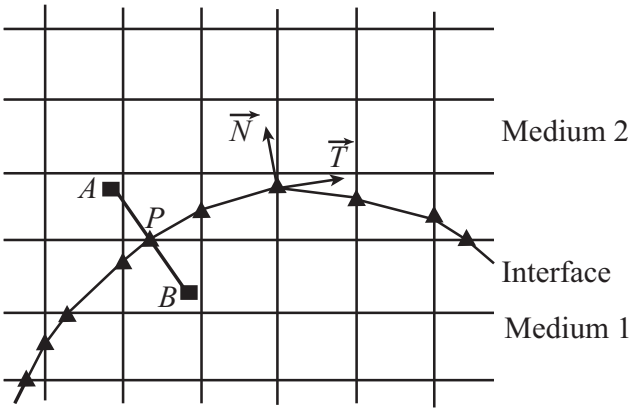


Figure 1: Construction of the Riemann problem at the marker point.

The choice of the tangential velocity depends on the sign of the normal velocity of the marker point P and is defined as:

$$v_T^I = \begin{cases} v_T^B, & u_N^I \geq 0, \\ v_T^A, & u_N^I < 0, \end{cases} \tag{3.4}$$

where u_N^I and v_T^I are the normal and tangential velocities of the marker point P, v_T^A and v_T^B are the tangential velocities of point A and point B.

Once the velocity of each marker point has been solved, its new position is updated simultaneously with the compressible flow equations by the same time integration scheme:

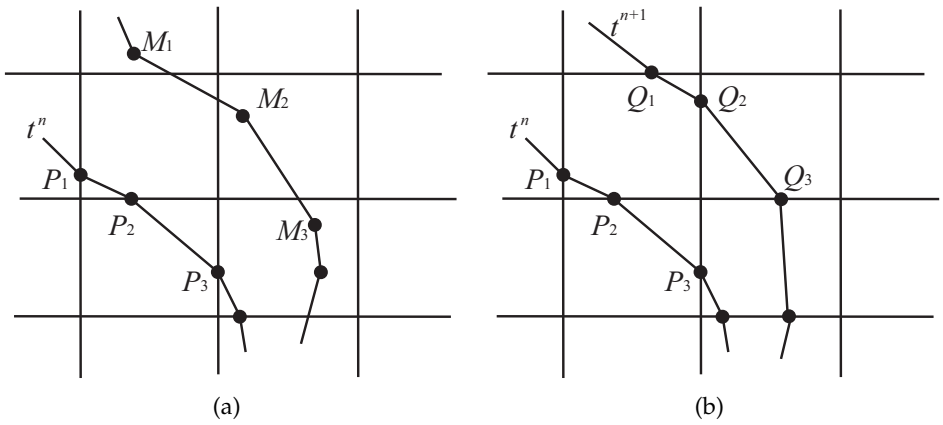
$$\vec{x}_f^{(1)} = \vec{x}_f^n + \Delta t \cdot \vec{v}_f(\vec{x}_f^n), \tag{3.5a}$$

$$\vec{x}_f^{(2)} = \frac{3}{4}\vec{x}_f^n + \frac{1}{4}\vec{x}_f^{(1)} + \frac{1}{4}\Delta t \cdot \vec{v}_f(\vec{x}_f^{(1)}), \tag{3.5b}$$

$$\vec{x}_f^{n+1} = \frac{1}{3}\vec{x}_f^n + \frac{2}{3}\vec{x}_f^{(2)} + \frac{2}{3}\Delta t \cdot \vec{v}_f(\vec{x}_f^{(2)}), \tag{3.5c}$$

where $\vec{x}_f^n$ and $\vec{x}_f^{n+1}$ are the positions of the interface at time t^n and t^{n+1}, respectively. $\vec{v}_f$ is the interface velocity and Δt is the time step in (3.1).

Here we briefly illustrate the process of interface reconstruction at time t^{n+1}. As shown in Fig. 2, points P_1, P_2 and P_3 are three marker points at time t^n and are moved to points M_1, M_2 and M_3 according to (3.5) respectively. Connect points M_1, M_2 and M_3 by the piecewise linear curves, which generates the intersections Q_1, Q_2 and Q_3 with the grid lines. Now points Q_1, Q_2 and Q_3 are the new marker points at time t^{n+1} and the interface is represented by connecting these new marker points. More details about interface reconstruction can be found in [9].



(a) (b)

Figure 2: Interface reconstruction at time t^{n+1}.

3.3 Determining the ghost fluid region

Since we solve the Euler equations in each fluid domain independently, it is required that we be able to identify fluid domain which the fixed grid points may fall within near the interface. In this paper we adopt the idea of indicator function which was originally used for evaluating one-field material properties in the front tracking implementation [28,29].

The front marks the jump in the indicator function and this jump is translated into a steep gradient on the fixed grid. The indicator function $I(\vec{x})$, which is based on a Heaviside function and designed to be 1 if the grid position $\vec{x}$ is in one medium and -1 in the other medium at each side at time t^n, is defined as:

$$I(\vec{x})=1-2H(\vec{x}), \tag{3.6}$$

where $H(\vec{x})$ is the Heaviside function. If we calculate the gradient we find:

$$\nabla I(\vec{x})=-2\nabla H(\vec{x})=-2\int \Delta H(\vec{x})\vec{n}\delta(\vec{x}-\vec{x}_f)ds, \tag{3.7}$$

here, $\Delta H(\vec{x})$ should be 1.0. Upon taking the divergence of (3.7)) and replacing the δ distribution with a smooth approximation, we have the following Poisson equation in the discrete version for the fixed grid:

$$(\nabla^2 I(\vec{x}))_{i,j}=-2\nabla\cdot(\nabla H(\vec{x}))_{i,j}=-2\nabla\cdot\Big(\sum_l \omega_{i,j}^l \vec{n}_l \Delta s_l\Big), \tag{3.8}$$

where i and j identify the nodes on the fixed grid, l is the index for the marker elements of the interface, $\vec{n}_l$ is the unit normal vector of element l, Δs_l is the length of element l and $\omega_{i,j}^l$ is the weight of grid point (i,j) with respect to element l which should satisfy:

$$\sum_l \omega_{i,j}^l=1. \tag{3.9}$$

Since the weights have a finite support, there are a relatively small number of elements that can be used. For propagating from the geometric center of the element l which is denoted as $\vec{x}_p=(x_p,y_p)$, the weight for the grid point (i,j) is given by:

$$\omega_{i,j}(\vec{x}_p)=d(x_p-ih)d(y_p-jh), \tag{3.10}$$

here, h is the grid spacing and $d(r)$ is taken as [22]:

$$d(r)=\begin{cases} (1/2\xi h)[1+\cos(\pi r/\xi h)], & |r|<\xi h, \\ 0, & |r|>\xi h. \end{cases} \tag{3.11}$$

The Poisson equation (3.8) is solved by iterating only on a few grid points near the interface for a rapid convergence. The numerical solution of Poisson equation gives a continuous indicator function increasing smoothly from -1 to 1 over a distance of the order ξh

from the actual interface. In this paper, the parameter ξ is taken as 4. The use of this relatively large value of ξ increases the smoothing of the front and this smoothing permits an accurate localization of the grid points in their appropriate phase [11]. The unit surface normal vector $\vec{N}_g$ on the fixed grid cells can be determined immediately by the indicator function:

$$\vec{N}_g = \nabla I / |\nabla I|. \tag{3.12}$$

3.4 RGFM

In the RGFM [30], a Riemann problem is defined at the interface to update the flow states at the real grid cells next to the interface and the ghost fluid states are obtained by solving an advection equation. However, since the Riemann problem has been constructed when tracking the fluid interface in this paper, the Riemann solutions can be used directly. As indicated in Fig. 3, the medium 1 and the medium 2 are separated by the interface and points R, P, Q are three marker points. $\vec{N}_P$ is the normal vector of the marker point P and $\vec{N}_A$ is the surface normal vector of the grid cell A, respectively. The flow states at the grid cell A can be updated by the marker points nearby. The marker point P is selected if the angle between $\vec{N}_P$ and $\vec{N}_A$ is the minimum compared with the other marker points. If the Riemann solutions at marker point P are denoted as $[\rho_L^I, \rho_R^I, u_N^I, p^I]^T$, then the density, the normal velocity and the pressure at the grid cell A can be updated by ρ_L^I, u_N^I, p^I respectively while the tangential velocity at the grid cell A remains unchanged. It is similar to update the flow states in other real grid cells adjacent to the interface.

In order to extrapolate the updated flow states into the ghost fluid region, the following advection equation is used:

$$\frac{\partial \phi}{\partial t} \pm \vec{N}_g \cdot \nabla \phi = 0, \tag{3.13}$$

where ϕ represents the density, the normal velocity, the tangential velocity and the pressure in the ghost fluid region, $\vec{N}_g$ is the unit surface normal vector on the fixed grid cells (see (3.12)). As indicated in Fig. 3, the $+$ sign is used in (3.13) if the interface boundary conditions of medium 1 are to determine and the $-$ sign is used in (3.13) if the interface

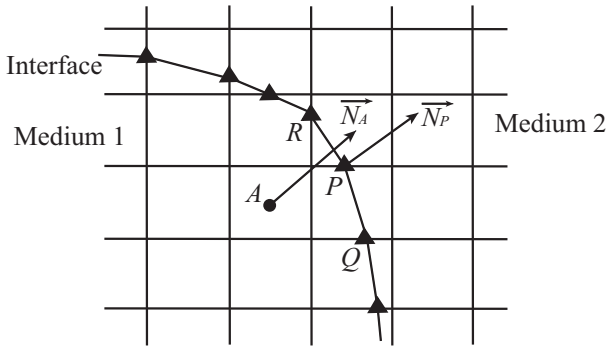


Figure 3: Updating the grid cells adjacent to the interface by the Riemann solutions.

boundary conditions of medium 2 are to determine. This advection equation is solved by iterating in fictitious time τ at each time step. The fictitious time step is chosen to be a little smaller than the one for the solution of the governing equations. To be time-saving throughout the computations, we only take 5-7 grid cells across the interface as the ghost fluid region. Once the advection equation has been solved in the ghost fluid region, the interface boundary conditions can be obtained.

3.5 Summary of the solution procedures

Here we briefly summarize the implementation procedures for the RGFM-FT method assuming that the flow states at time t^n have been obtained:

1. Generate the ghost fluid region and calculate the unit surface normal vector for the fixed grid cells.
2. Construct the Riemann problem to obtain the density, the velocity and the pressure of the marker points.
3. Define ghost fluid states for each ghost fluid cell according to the RGFM implementation.
4. Obtain the flow field by solving the Euler equations with proper boundary conditions.
5. Obtain the new location of the interface via the Riemann solutions of the marker points and get the final solutions according to the new location of the interface.
6. Return to the first step.

4 Numerical tests

In this section the RGFM-FT method is used to simulate several two-dimensional compressible, non-viscous multimedium flows in order to validate the accuracy and capability in treating the fluid interface. The compressible Euler equations are solved on uniform Cartesian grids.

4.1 Bubble advection

This problem is linear advection of a helium bubble in air and is used to test the accuracy and conservation [7]. The computational domain is $[0,2]\times[0,2]$ and a cylindrical helium bubble with a diameter of 0.4 is centered at $(0.25,0.25)$. The conservative fluid variables are simply extrapolated with zero gradient on all the boundaries. The nondimensionalized initial conditions are:

$$(\rho,u,v,p,\gamma,B)=\begin{cases}(1,1,1,1,1.4,0), & \text{air},\\ (0.138,1,1,1,1.67,0), & \text{helium}.\end{cases} \tag{4.1}$$

Table 1: Numerical accuracy for helium advection in air.

Mesh size	E_d	order	E_{He}	order
1/10	8.82×10^{-2}	-	9.98×10^{-1}	-
1/20	7.63×10^{-3}	3.53	7.01×10^{-2}	3.83
1/40	5.04×10^{-4}	3.92	4.41×10^{-3}	3.99
1/80	3.26×10^{-5}	3.95	2.74×10^{-4}	4.01

Two discrete errors are measured here, namely the L1 error in the density field, E_d and the relative error in total mass of helium, E_{He}. The errors and order of accuracy are given in Table 1 for different mesh sizes at time $t=0.5$. The numerical results show that the errors in both the density field and total mass of helium almost converge at fourth order.

4.2 Sod shock tube

This simple example is taken from [31] and is used to investigate a high pressure air cavity expanding in water. Initially, a cylindrical air cavity is at the center of the computational domain $[0,4]\times[0,4]$ with a diameter of 1. The initial conditions are taken as:

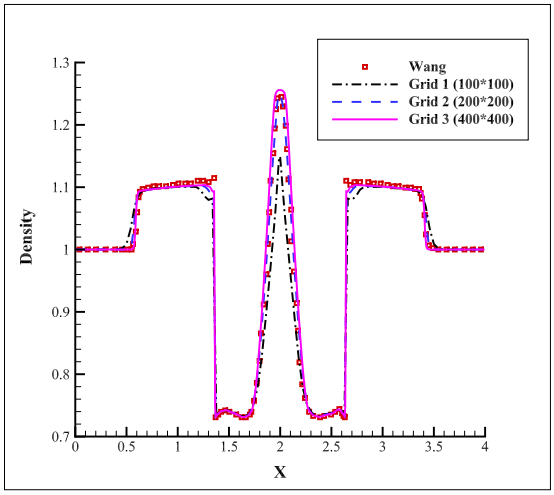
$$(\rho,u,v,p,\gamma,B)=\begin{cases}(1.27,0,0,8000,1.4,0), & \text{air},\\ (1,0,0,1,7.15,3309), & \text{water}.\end{cases} \tag{4.2}$$

The CFL number is 0.24 and the nonreflecting boundary condition is used at all the boundaries.

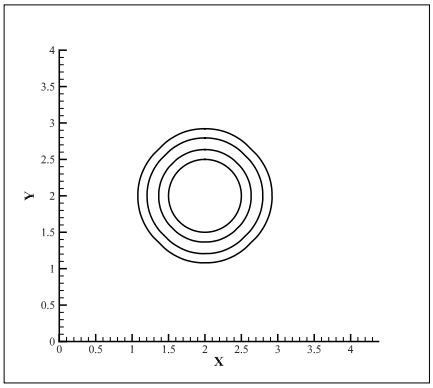
Three different grids are used here for grid refinement test: coarse (100×100 mesh points), intermediate (200×200) and fine (400×400). Fig. 4(a) gives the density distributions along $y=2$ at time $t=0.0045$, with the effect of grid resolution, and the results obtained by an adaptive ghost fluid finite volume method from Wang et al. [31] are also shown in order to make comparisons. We can see that when using coarse grids there is a density discrepancy near the center of computational domain, however, the intermediate and fine grids give mostly identical results, which show a converged solution of the new method. Fig. 4(b) shows the interfaces at time $t=0$, 0.0045, 0.00935, 0.0137 on the fine grid. These interfaces are a series of concentric circles and the shape of interface remains unchanged. The density contours at $t=0.0045$ on the fine grid are also presented in Fig. 4(c), where the interface, the shock wave and the expansive wave are shown clearly. A shock is formed running towards the boundary of the domain and an expansive wave is running towards the center of the domain.

4.3 Shock-bubble interaction

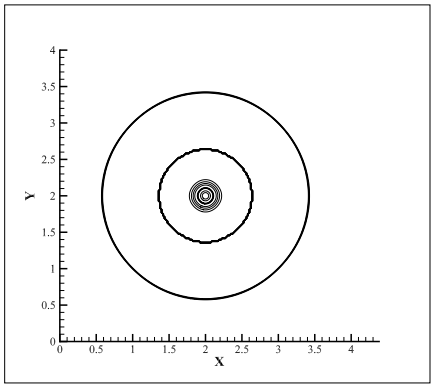
Shock wave interactions with bubbles have been studied experimentally by Hass and Sturtevant [12]. It also has been numerically investigated by several authors, see [20, 23, 27]. Hass and Sturtevant's experiment is simulated in this example, where a shock



(a) Comparison of density along $y=2$



(b) The interface evolution



(c) Density contours

Figure 4: The numerical results for the sod shock tube problem.

wave propagates to the left and hits a helium bubble with Mach number of 1.22. Fig. 5 shows the computational set-up and the geometrical parameters are: $a=50mm$, $b=25mm$, $c=100mm$, $d=325mm$, $e=44.5mm$. Since the flow field is symmetric about the center axis, only the bottom half domain is computed. At the axis, the velocity in y direction is set to zero and the density, the velocity in x direction and the pressure are all symmetric. On the left and right boundaries, all the variable are simply extrapolated with zero gradient and the lower boundary is treated as slip-wall. For this simulation the sound speed in the pre-shocked air and the bubble diameter are used for the nondimensionalization, and the CFL number is 0.2. The computational domain is divided into 650×89 mesh cells. The initial conditions are: $\rho=1$, $u=0$, $v=0$, $p=1/1.4$, $\gamma=1.4$, $B=0$, for pre-shocked air, $\rho=1.3764$, $u=-0.3336$, $v=0$, $p=1.5698/1.4$, $\gamma=1.4$, $B=0$, for post-shocked air, $\rho=0.1819$, $u=0$, $v=0$, $p=1/1.4$, $\gamma=1.648$, $B=0$, for helium.

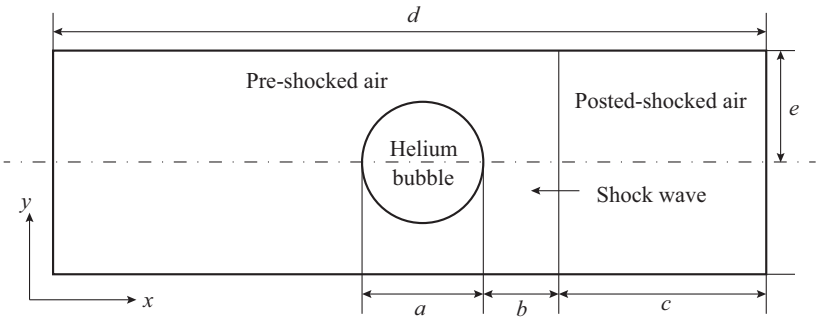


Figure 5: A schematic of computational domain (not to scale).

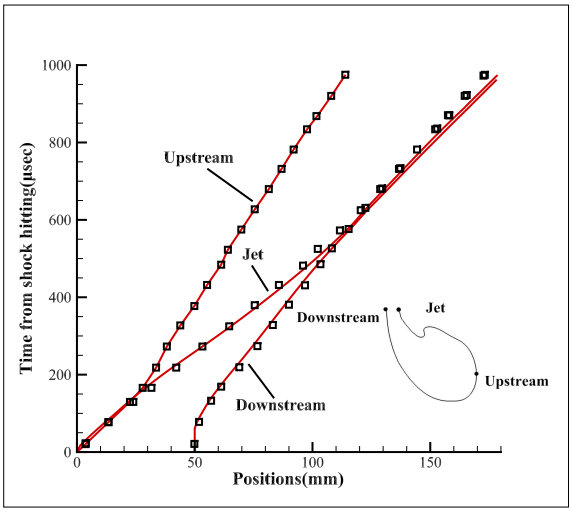
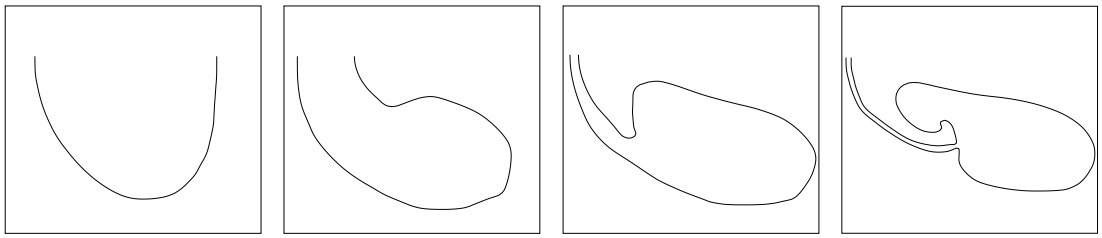
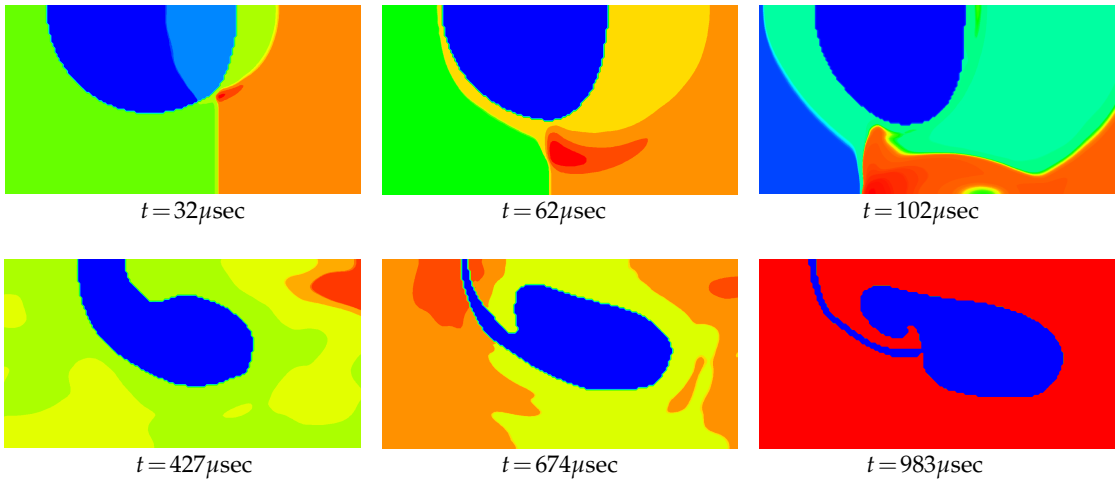


Figure 6: Space-time diagrams for three characteristic interface points.

Fig. 6 shows the space-time diagram for three characteristic points (Jet, Downstream, Upstream shown in the figure) compared with the earlier computational results ('○') from [27] during the early stage. The present results are in a relatively good agreement with those results, showing the accuracy of the new method. Fig. 7 shows the time histories of the helium bubble shape and the density field. The timing is the same as in [27] for easy comparison. We can see that the results, including the positions of shock wave, are in good qualitative agreement with the earlier results for the entire process. However, there are differences in the details of the interface, which is to be expected since the interface is unstable, and without some regularization there will be no unique or resolved answer to the Euler equations [7]. At the initial stage, the interface swept over by the shock wave is set into motion while the others remain at rest. During the contraction of the bubble, the length of the interface decreases and then the interface begins to fold down over itself. The ambient fluid piercing the bubble grows and rolls up in the anticlockwise



(a) The interfaces at different time



(b) The density field at different time

Figure 7: The evolution of interfaces and density field.

direction, thus an overpressure behind the bubble is formed to compress the bubble to the left part of the computational domain. The shock wave diffracts into two waves after it hits the helium bubble. Part of the shock wave is transmitted to the bubble with a stronger intensity than that of the incident shock along the normal direction of the interface. This shock wave moves faster than the one in the surrounding air due to the different sound speed and therefore a relative sliding of the helium on the air is observed, which indicates the importance of the tangential velocity and the approximation made on the determination of the tangential velocity in this paper is acceptable.

4.4 Richtmyer-Meshkov instability

We now present numerical results to the simulation of Richtmyer-Meshkov instability [3]. This example consists of two simulations with gas-gas and gas-liquid interfaces. It has been investigated by several authors to test and validate their methods before [20, 26, 27]. The first one is a gas-gas interface taken from [20]. Only the bottom half domain is

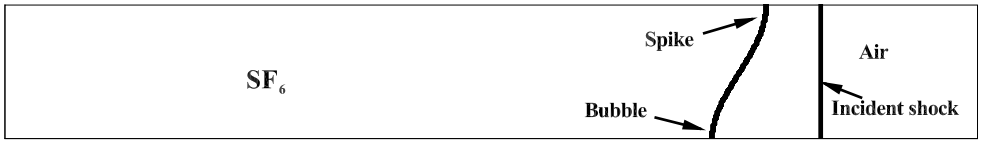
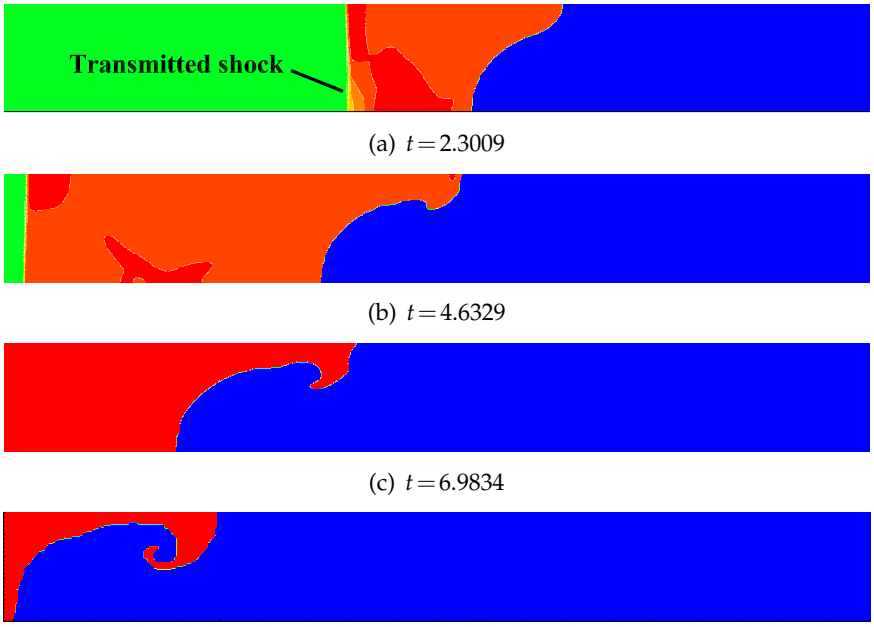


Figure 8: A schematic of flow field at $t=0$.

computed since the flow field is symmetric about the center axis. As indicated in Fig. 8, a computational domain of $[0,4]\times[0,0.5]$ is used and we take a single mode perturbation of an air-SF_6 interface that the initial location of the interface is represented by: $x=2.9-0.1\sin(2\pi(y+0.25))$, $0<y<0.5$. The initial conditions are: $\rho=5.04$, $u=0$, $v=0$, $p=1$, $\gamma=1.093$, $B=0$, for SF_6, $\rho=1$, $u=0$, $v=0$, $p=1$, $\gamma=1.4$, $B=0$, for pre-shocked air, $\rho=1.411$, $u=-0.39$, $v=0$, $p=1.628$, $\gamma=1.4$, $B=0$, for post-shocked air. To trigger the instability, at $x=3.2$ there is a planar Mach number 1.24 shock wave in air propagating from the right to the left of the interface. The interface is accelerated by a shock wave coming from the light fluid to the heavy fluid region. The upper and lower boundaries are taken to be periodic and the nonreflecting boundary condition is applied at the left and right boundaries. The CFL number is taken as 0.4 and a 1000×125 grid is used.

The flow evolution by plotting the density field is presented in Fig. 9 where the interface and the transmitted shock wave can be observed clearly. Due to the initial pertur-



(a) $t=2.3009$

(b) $t=4.6329$

(c) $t=6.9834$

(d) $t=9.2399$

Figure 9: Dynamics of the interface and density field.

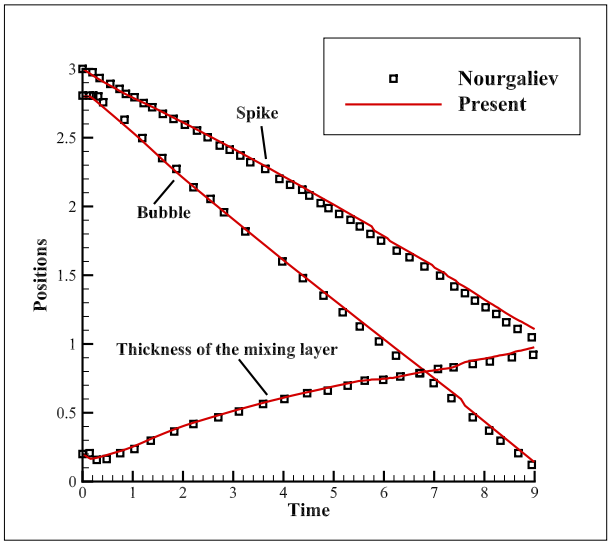


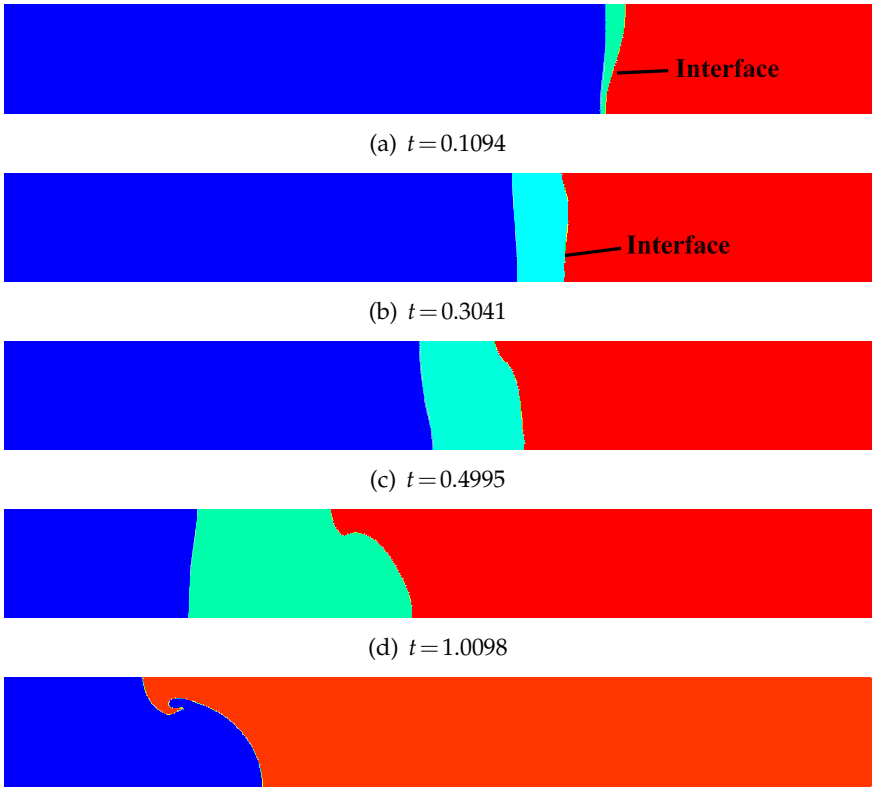
Figure 10: Comparison with other study on time histories of characteristic positions.

bation of the interface, there is a misalignment of density and pressure gradient, which results in a non-zero term $(\frac{1}{\rho^2}\nabla p\times\nabla\rho)$, and the vorticity is formed in the perturbed zone, as can be seen in Fig. 9(c) and Fig. 9(d). In order to validate the accuracy of the results, Fig. 10 presents the time evolution of the location of the spike, the leading edge of the bubble and the thickness of the mixing layer along with the results of Nourgaliev et al. [20]. The thickness of the mixing layer is defined as the distance between the spike and the bubble. It shows that these results are almost identical for the location of these characteristic points.

The second one is a gas-liquid interface that is interacting with a planar Mach number 1.95 shock wave at $x=3.025$ initially in liquid. The computational domain, the grids and the initial location of the interface are the same with the first simulation of Richtmyer-Meshkov instability. The initial conditions are: ρ=1, u=0, v=0, p=1, γ=1.4, B=0, for air, ρ=5, u=0, v=0, p=1, γ=4, B=1, for pre-shocked liquid, ρ=7.093, u=−0.7288, v=0, p=10, γ=4, B=1, for post-shocked liquid. The density field at different time is shown in Fig. 11 where the interface and the transmitted shock wave can be seen clearly. The complex wave structure is once again present in this problem and is relatively well captured. To check the correctness of the computed solutions, Fig. 12 compares the distributions of density and pressure along line $y=0.5$ with the results ('$\circ$') obtained by using the γ-based model in [26]. Good agreement of the solutions is clearly observed.

4.5 Gas-water interface test

The final example is about a shock wave in air impacting on the cylindrical water droplet. It has been numerically and experimentally investigated by several authors [20, 27, 31].



(a) $t=0.1094$

(b) $t=0.3041$

(c) $t=0.4995$

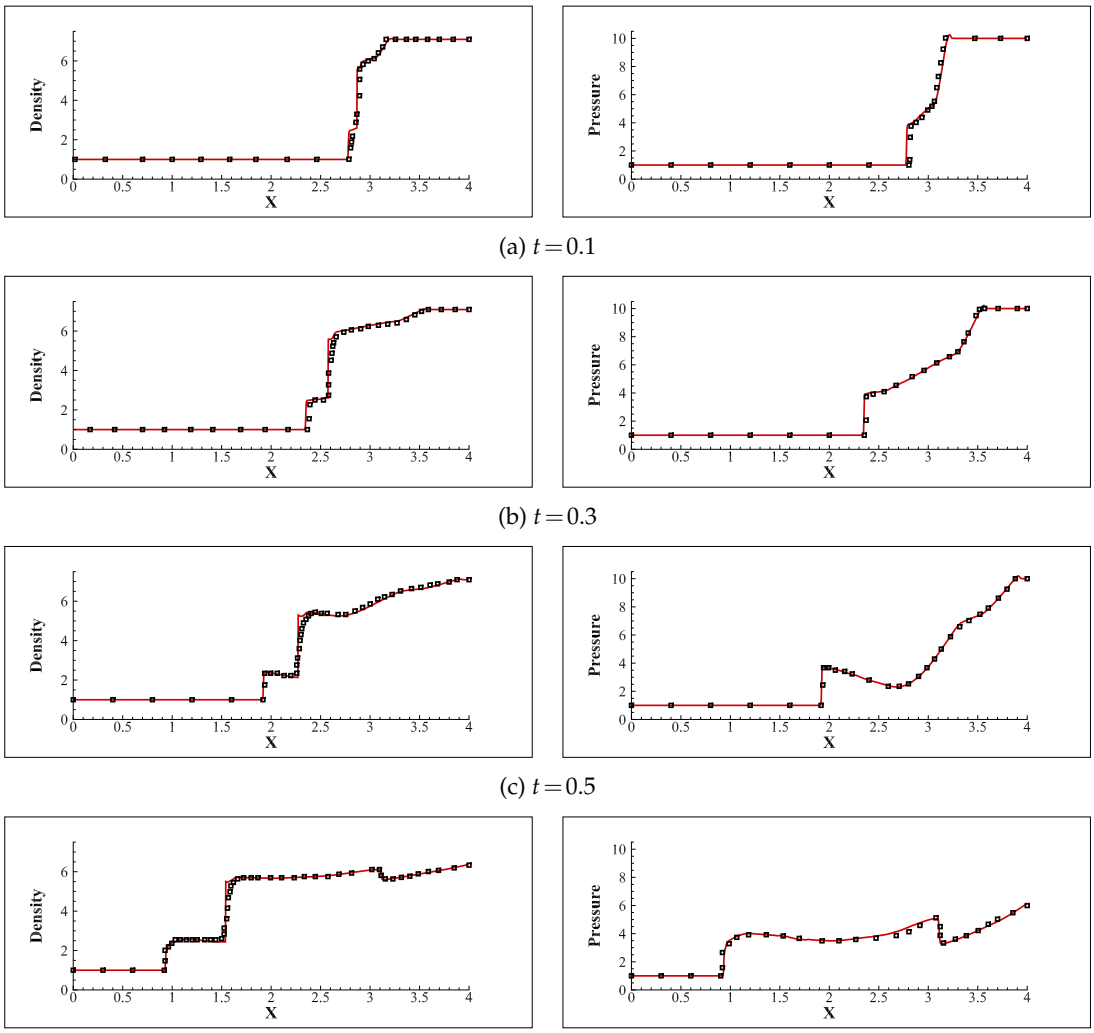
(d) $t=1.0098$

(e) $t=1.6349$

Figure 11: The density field at different time.

This is to test the method for multimedium flow with large density jump across the interface. It involves a curved interface and associated with rather complex irregular shock refraction patterns. Here we take the same initial and boundary conditions as in [31] and also only the bottom half of the problem is simulated. A right-moving shock wave with the Mach number 1.47 is initially placed at $x=6$. The computational domain is $[0,20]\times[0,6]$ and the cylindrical water droplet surrounded by the air is centered at $(10,6)$ with the radius of 3.2. The initial conditions are: $\rho=1$, $u=0$, $v=0$, $p=1$, $\gamma=2.8$, $B=3036$, for water droplet, ρ=0.001, u=0, v=0, p=1, γ=1.4, B=0, for pre-shocked air, ρ=0.001811, $u=24.624$, $v=0$, $p=2.35$, $\gamma=1.4$, $B=0$, for post-shocked air. The computational domain is divided into 200×60 mesh cells. The upper boundary is symmetric and nonreflecting boundary condition is applied on the other boundaries. The CFL number is taken as 0.24.

The early stage of the interaction of the shock wave with water droplet is shown in Fig. 13 with the interface and the pressure contours. When the incident shock hits the droplet, it is reflected as a shock, transmitting a pressure wave into the droplet. The results, including the positions of incident shock wave and transmitted shock wave, are in good agreement with those shown in [31].



(a) $t=0.1$

(b) $t=0.3$

(c) $t=0.5$

(d) $t=1$

Figure 12: Comparison of density (left) and pressure (right) along line $y=0.5$.

5 Concluding remarks

We have introduced a front tracking method combined with the RGFM for simulations of fluid interfaces in compressible flows. Fluid interface boundary conditions are defined with the RGFM and the motion of interface is obtained by solving a Riemann problem near the interface. This new method is more accurate and easier to implementation. Extensive numerical tests are performed to demonstrate the accuracy and capability of the new method. The numerical results of bubble advection indicate, by grid refinement tests, that the present method is convergent and stable. A Sod shock tube study shows

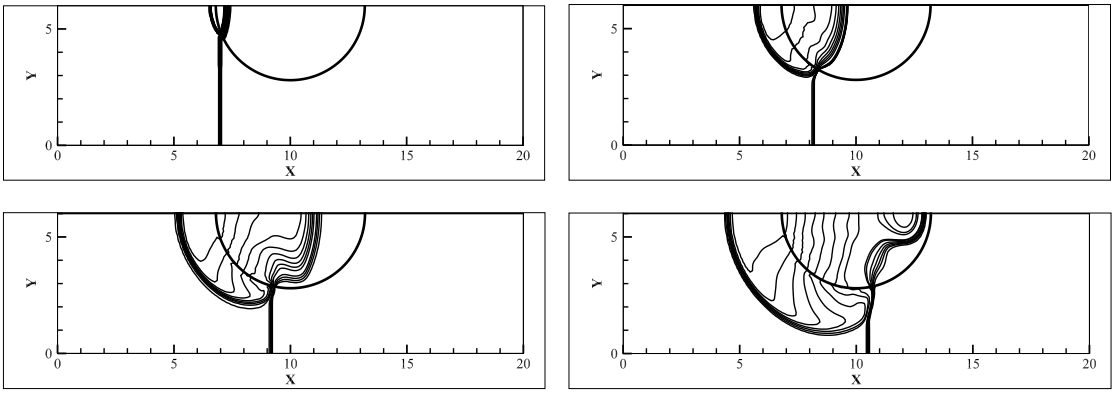


Figure 13: The pressure contours and the interface.

that the new method is able to handle the high pressure air expansion in water. In simulations of shock-bubble interaction, it shows that the bubble deformation and the motion of characteristic points compare well with earlier experimental observations and computational results. Simulations of Richtmyer-Meshkov instability demonstrate the capability of the method to track complicated interface caused by the passage of a shock wave. A simulation of compressible gas-water system shows that the new method can be applied to multimedium flows with large density jump across the interface. The results presented in this paper show that this new method can accurately simulate multimedium flows for a wide range of initial conditions.

Acknowledgments

All the authors are supported by NSFC grants 91130030 and 11432007. Additional support is provided by a project funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions. The authors would like to thank Professor Jie Wu for his useful suggestions.

References

[1] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: a quasi conservative approach, J. Comput. Phys., 125 (1996), 150-160.
[2] J. U. Brackbill, D. B. Kothe and C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys., 100 (1992), 335-354.
[3] M. Brouillette, The Richtmyer-Meshkov instability, Annu. Rev. Fluid Mech., 34 (2002), 445-468.
[4] R. Caiden, R. P. Fedkiw and C. Anderson, A numerical method for two-phase flow consisting of separate compressible and incompressible regions, J. Comput. Phys., 166 (2001), 1-27.

[5] J.-P. Cocchi and R. Saurel, A Riemann problem based method for the resolution of compressible multimaterial flows, J. Comput. Phys., 137 (1997), 265-298.
[6] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, J. Comput. Phys., 175 (2002), 200-224.
[7] R. P. Fedkiw, T. Aslam, B. Merriman and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys., 152 (1999), 457-492.
[8] J. Glimm, J. W. Grove, X. L. Li, K.-M Shyue, Y. Zeng and Q. Zhang, Three-dimensional front tracking, SIAM J. Sci. Comput., 19 (1998), 703-727.
[9] J. Glimm, J. W. Grove, X. L. Li and N. Zhao, Simple front tracking, Contemp. Math., 238 (1999), 133-149.
[10] J. Glimm, J. W. Grove, X. L. Li, W. Oh and D. H. Sharp, A critical analysis of Rayleigh-Taylor growth rates, J. Comput. Phys., 169 (2001), 652-677.
[11] Y. Hao and A. Prosperetti, A numerical method for three-dimensional gas-liquid flow computations, J. Comput. Phys., 196 (2004), 126-144.
[12] J. F. Hass and B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech., 181 (1987), 41-76.
[13] G.-S. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys., 126 (1996), 202-228.
[14] S. Karin, Multicomponent flow calculations by a consistent primitive algorithm, J. Comput. phys., 112 (1994), 31-43.
[15] B. Larrouturou, How to preserve the mass fractions positivity when computing compressible multi-component flows, J. Comput. Phys., 95 (1991), 59-84.
[16] T. G. Liu, B. C. Khoo and K. S. Yeo, The simulation of compressible multi-medium flow. I. A new methodology with test applications to 1D gas-gas and gas-water cases, Comput. & Fluids, 30 (2001), 291-314.
[17] T. G. Liu, B. C. Khoo and K. S. Yeo, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys., 190 (2003), 651-681.
[18] T. G. Liu, B. C. Khoo and C. W. Wang, The ghost fluid method for compressible gas-water simulation, J. Comput. Phys., 204 (2005), 193-221.
[19] W. Mulder, S. Osher and J. A. Sethian, Computing interface motion in compressible gas dynamics, J. Comput. Phys, 100 (1992), 209-228.
[20] R. R. Nourgaliev, T. N. Dinh and T. G. Theofanous, Adaptive characteristics-based matching for compressible multifluid dynamics, J. Comput. Phys., 213 (2006), 500-529.
[21] S. Osher and R. P. Fedkiw, Level set methods: An overview and some recent results, J. Comput. Phys., 169 (2001), 463-502.
[22] C. S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys., 25 (1977), 220-252.
[23] J. J. Quirk and S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech., 318 (1996), 129-163.
[24] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys., 77 (1988), 439-471.
[25] L. Shui, J.C.T. Eijkel and A. van den Berg, Multiphase flow in microfluidic systems-Control and applications of droplets and interfaces, Adv. Colloid Interface Sci., 133 (2007), 35-49.
[26] K.-M. Shyue, An efficient shock-capturing algorithm for compressible multicomponent problems, J. Comput. Phys., 142 (1998), 208-242.
[27] H. Terashima and G. Tryggvason, A front-tracking/ghost-fluid method for fluid interfaces

in compressible flows, J. Comput. Phys., 228 (2009), 4012-4037.
[28] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas and Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys., 169 (2001), 708-759.
[29] S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys., 100 (1992), 25-37.
[30] C. W. Wang, T. G. Liu and B. C. Khoo, A real ghost fluid method for the simulation of multimedium compressible flow, SIAM J. Sci. Comput., 28 (2006), 278-302.
[31] C. W. Wang, H. Z. Tang and T. G. Liu, An adaptive ghost fluid finite volume method for compressible gas-water simulations, J. Comput. Phys., 227 (2008), 6385-6409.