Journal of Computational Physics 228 (2009) 4012–4037

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A front-tracking/ghost-fluid method for fluid interfaces in compressible flows

Hiroshi Terashima *, Grétar Tryggvason *

Mechanical Engineering Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609-2280, United States

ARTICLE INFO

Article history:
Received 10 March 2008
Received in revised form 9 February 2009
Accepted 15 February 2009
Available online 1 March 2009

Keywords:
Compressible multiphase flow
Front-tracking method
Ghost-fluid method
Fluid interface
Compressible gas–water flow

ABSTRACT

A front-tracking/ghost-fluid method is introduced for simulations of fluid interfaces in compressible flows. The new method captures fluid interfaces using explicit front-tracking and defines interface conditions with the ghost-fluid method. Several examples of multiphase flow simulations, including a shock–bubble interaction, the Richtmyer–Meshkov instability, the Rayleigh–Taylor instability, the collapse of an air bubble in water and the breakup of a water drop in air, using the Euler or the Navier–Stokes equations, are performed in order to demonstrate the accuracy and capability of the new method. The computational results are compared with experiments and earlier computational studies. The results show that the new method can simulate interface dynamics accurately, including the effect of surface tension. Results for compressible gas–water systems show that the new method can be used for simulations of fluid interface with large density differences.

1. Introduction

The dynamics of interfaces separating different fluids in compressible flows is of interest in several scientific fields as diverse as astrophysics and geophysics. It is also of significant importance in many engineering applications. An in-depth understanding of multiphase flow in supersonic combustion is, for example, desirable for the development and effective operation of supersonic ramjet engine, and many other combustion systems involving fuel drops in high-speed airflow, where liquid jets must be atomized efficiently [1,2].

For computing compressible multifluids, early algorithms have treated discontinuous material interfaces with the γ (the ratio of specific heats)-model [10], the mass fraction model [12,50], or a level-set function [7] in order to identify each fluid, coupled to the Euler equations. These algorithms, usually based on classical shock-capturing methods, however, have suffered from unphysical oscillations developed at material interfaces. Abgrall [12] identified the cause of those spurious oscillations and proposed a quasi-conservative method based on the mass fraction model. This has been extended to more general equation of state [13]. Karni [10] also proposed a remedy to avoid pressure oscillations using a non-conservative scheme based on the primitive variables. Successful applications of Karni's method can be seen, for example, in computations of the interaction of a shock wave with a helium bubble. This problem has now become a classical benchmark for multidimensional compressible multifluids algorithms. Karni's approach was modified later in [11] to capture strong shocks, using the pressure evolution equation and the level-set equation.

A different approach for compressible multifluids simulations has been pursued by Glimm and co-authors [3,4], who have been using explicit marker points for capturing waves (such as contact discontinuity). The Glimm's front-tracking method, where selected waves are explicitly represented in the discrete form of the solution, Riemann solutions are constructed near

* Corresponding authors. Tel.: +1 508 831 6803 (H. Terashima), Tel.: +1 508 831 5759; fax: +1 508 831 5680 (G. Tryggvason).
E-mail addresses: htera@wpi.edu (H. Terashima), gretar@wpi.edu (G. Tryggvason).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.02.023

the fluid interface to propagate the solution in the normal and tangential directions using ghost nodes. For recent applications see the simulations of the three-dimensional Rayleigh–Taylor instability described in [5]. Similar idea of using Riemann solutions was used by Cocchi and Saurel [6]. Their algorithm consists of a predictor step and a corrector step. In the corrector step, a Riemann problem is solved at material interfaces in order to correct the solutions from the predictor step (where a standard Godunov type scheme is used) that generally generates numerical diffusion or spurious oscillations. More recently, following the strategy of using level-set function and adaptive mesh refinement, Nourgaliev et al. [9] have also attempted to avoid numerical oscillations near fluid interfaces by solving a suitable Riemann problem for interface cells.

Ghost nodes have more recently been used in the ghost-fluid method of Fedkiw and collaborators [8]. Fedkiw's ghost-fluid method is a level set based algorithm, which treats the interface as a moving boundary. By extrapolating the value of the discontinuous variables, such as entropy, across the fluid interface when solving the governing equations, one can reduce smearing of discontinuous variables such as density and other material properties. The method can also prevent pressure oscillations since a ghost fluid is set to be thermodynamically identical to the corresponding real fluid. Fedkiw's ghost-fluid method is easy to implement, robust, and can easily handle different fluids with different equations of state. Recently, however, it has been reported that pressure oscillations occur for flows with high density or pressure ratios and improved versions of Fedkiw's ghost-fluid method have therefore been proposed [15,24] to deal with such cases. In [15], for example, Riemann solutions are imposed for the ghost nodes in order to determine accurate interface status.

As other approach for compressible multifluids simulations, Chang and Liou [32] have recently developed a stratified flow model, which is capable of incorporating compressible gases and liquids.

So far the ghost-fluid method has only been used with the level-set methods to track the interface. It can, however, be used with other techniques for following the interface motion and in this paper we combine Fedkiw's ghost-fluid method with a front-tracking method. The front-tracking method originally developed by Tryggvason and co-workers [22,23] is used here. In this method, fluid interfaces are explicitly tracked by connected marker points. This method has been successfully applied to many multiphase flow problems, but so far all applications have been limited to incompressible flows. We therefore extend Tryggvason's front-tracking method here to solve the compressible flow equations. The concept of the ghost-fluid method, in which each fluid is solved separately, can match with the spirit of the front-tracking method, in which each fluid is distinguished sharply by explicit marker points. Another extension of Tryggvason's front-tracking method, using Fedkiw's ghost-fluid method to handle compressible flows, can be found in a recent study of Hao and Prosperetti [26]. There, the compressible fluid was limited to a simple gas model whereas here we use the full compressible Euler or Navier–Stokes equations both for the fluids.

In this paper, we limit our development and applications to two-dimensional flow fields. The proposed numerical techniques, however, can be extended to three dimensions in the same way, although the coding for restructuring front will be more complicated [23].

2. Governing equations

The governing equations for flow fields are the two-dimensional compressible Navier–Stokes equations written in Cartesian coordinate

$$\partial_t \boldsymbol{Q} + \partial_x \boldsymbol{E} + \partial_y \boldsymbol{F} = Re'^{-1}\left(\partial_x \boldsymbol{E}_v + \partial_y \boldsymbol{F}_v\right) + Fr^{2,\prime-1}\boldsymbol{G} + We'^{-1}\boldsymbol{S}. \tag{1}$$

Here, $\boldsymbol{Q}$ is the vector of conserved variables, $\boldsymbol{E}$ and $\boldsymbol{F}$ are the convective flux vectors and $\boldsymbol{E}_v$ and $\boldsymbol{F}_v$ are the flux vectors for the viscous terms:

$$\boldsymbol{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix}, \quad \boldsymbol{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ (E+p)u \end{bmatrix}, \quad \boldsymbol{F} = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ (E+p)v \end{bmatrix}, \quad \boldsymbol{E}_v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xx}u + \tau_{xy}v - q_x \end{bmatrix}, \quad \boldsymbol{F}_v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yx}u + \tau_{yy}v - q_y \end{bmatrix}, \tag{2}$$

where τ_{ij} are the shear stresses and q_i is the heat conduction. $\boldsymbol{G}$ is a body force (in most cases gravity). $\boldsymbol{S}$ is a source term for the surface tension, included in the momentum equations as a singular body force per unit volume:

$$\boldsymbol{S} = -\sigma \int_{\delta s} \kappa \boldsymbol{n} \delta(\boldsymbol{x} - \boldsymbol{x}_f) ds. \tag{3}$$

Here, σ is the surface tension, κ is twice the mean curvature, and $\boldsymbol{n}$ is a unit surface normal vector. The contribution of the surface tension is limited to fluid interfaces, as indicated by the delta function, δ. In the argument of δ, $\boldsymbol{x}$ is a point at which the equations are evaluated, and $\boldsymbol{x}_f$ is a point on the interfaces. Thus, in this way the surface tension effect is smeared in this study, but one may implement the surface tension effect in a sharp way using pressure difference done in [51], which will be more consistent with our method.

In addition to Eq. (1), the stiffened gas equation of state [13] is used:

$$p = (\gamma - 1)\rho e - \gamma \Pi. \tag{4}$$

Here p denotes the static pressure, ρ the density, and e the internal energy. The constants γ and Π are characteristic parameters of material. This equation reduces to the ideal gas equation of state when γ represents the ratio of the specific heats and Π is zero. The stiffened gas equation of state is a reasonable approximation for gases, liquids, and solid under high pressure conditions.

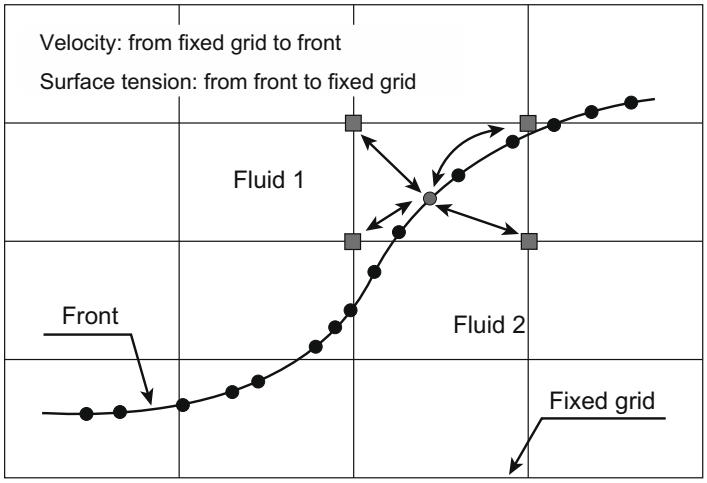


Fig. 1. Schematic view of front-tracking approach.

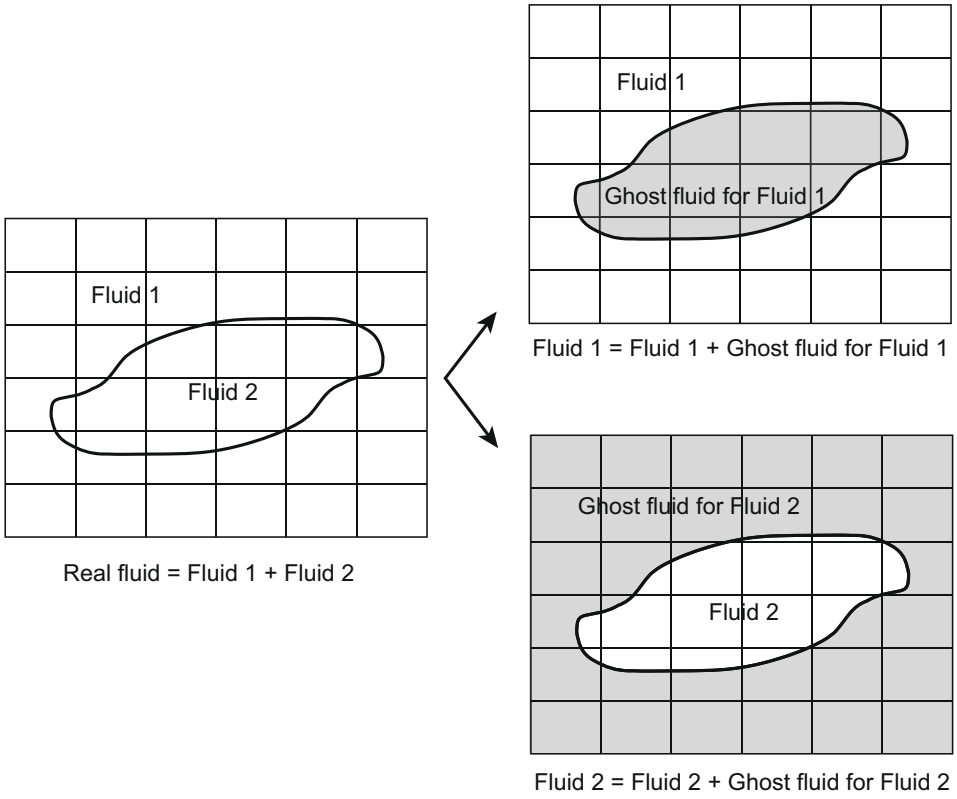


Fig. 2. Schematic of ghost-fluid method.

All variables are normalized by the freestream density ρ_∞, the freestream speed of sound a_∞, and a characteristic length L. The Reynolds number Re', the Froude number Fr', and the Weber number We' in Eq. (1) are therefore defined respectively as:

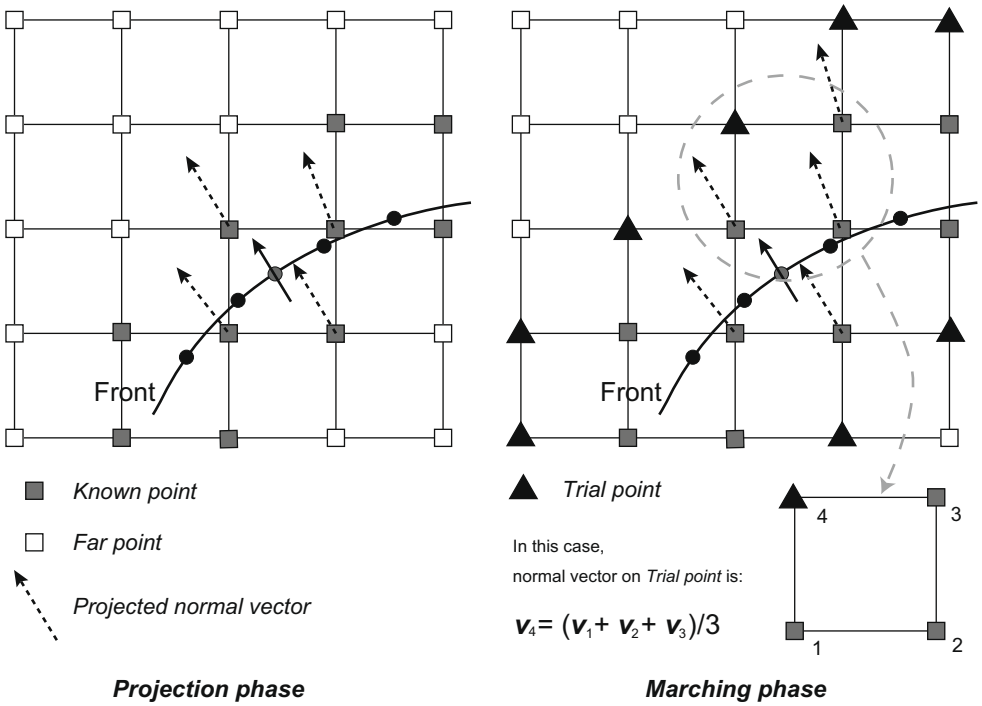


Fig. 3. Update procedure of surface normal vector.

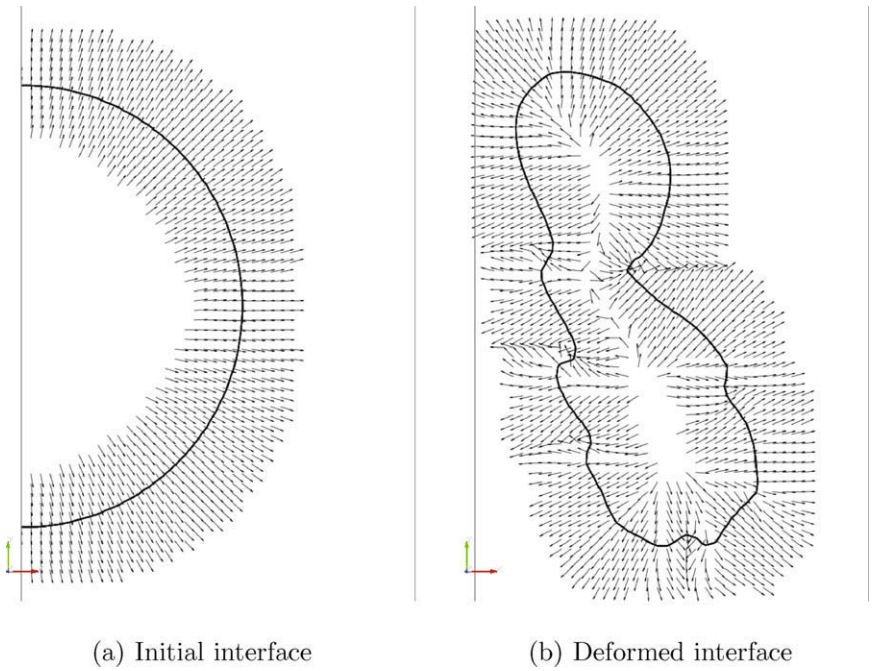


Fig. 4. An example of surface normal vector fields with a simple algebraic procedure.

$$Re' = \frac{\rho_\infty V_\infty L}{\mu_\infty}\frac{1}{M_\infty} = \frac{Re}{M_\infty}, \tag{5}$$

$$Fr^{2\prime} = \frac{V^2}{Lg_\infty}\frac{1}{M_\infty^2} = \frac{Fr^2}{M_\infty^2}, \tag{6}$$

$$We' = \frac{\rho_\infty V_\infty^2 L}{\sigma}\frac{1}{M_\infty^2} = \frac{We}{M_\infty^2}, \tag{7}$$

where $V_\infty, M_\infty, g_\infty$ and μ_∞ represent the freestream velocity, Mach number, gravity, and viscosity.

3. Numerical procedures

In this paper, we propose a new computational technique to simulate the motion of fluid interfaces in compressible high-speed flows, consisting of a front-tracking method coupled with the ghost-fluid method. The compressible Euler or Navier–Stokes equations are solved on a regular structured grid with uniform grid spacing.

3.1. Solving the compressible Navier–Stokes equations

The numerical fluxes are found by the Simple High-Resolution Upwind Scheme (SHUS) [16], which belongs to the family of Advection Upstream Splitting Method (AUSM)-type [17,18] schemes. Higher-order spatial accuracy is obtained using the Monotone Upstream-centered Schemes for Conservation Law (MUSCL) [19,20] with a primitive variable interpolation and minmod limiter. The viscous terms are evaluated by central differencing. The time integration is done with a third-order TVD Runge–Kutta scheme [21]. These numerical techniques have been used successfully for many compressible flow simulations, particularly in the transonic or supersonic flow regimes.

3.2. Tracking fluid interfaces

In the front-tracking method introduced by Unverdi and Tryggvason [22] the fluid interfaces are represented by marker points, which move with the fluid velocities, interpolated from the fixed grid. A schematic is shown in Fig. 1. The position of the marker points is updated simultaneously with the compressible flow equations, using the same time integration scheme:

$$\begin{aligned}
\boldsymbol{x}_f^{(1)} &= \boldsymbol{x}_f^n + \Delta t\, \boldsymbol{v}_f(\boldsymbol{x}_f^n),\\
\boldsymbol{x}_f^{(2)} &= \frac{3}{4}\boldsymbol{x}_f^n + \frac{1}{4}\boldsymbol{x}_f^{(1)} + \frac{1}{4}\Delta t\, \boldsymbol{v}_f(\boldsymbol{x}_f^{(1)}),\\
\boldsymbol{x}_f^{n+1} &= \frac{1}{3}\boldsymbol{x}_f^n + \frac{2}{3}\boldsymbol{x}_f^{(2)} + \frac{2}{3}\Delta t\, \boldsymbol{v}_f(\boldsymbol{x}_f^{(2)}).
\end{aligned} \tag{8}$$

Here $\boldsymbol{x}_f$ is the front position at a time step, $\boldsymbol{v}_f$ is the front velocity vector, and Δt is the time step size. The velocity interpolation from the fixed grid points to the marker points is done using bilinear interpolation (area-weighting).

The marker points are connected to form a front, which is used to construct the property fields on the fixed grid at each time step. Throughout the computations, marker points are dynamically added to the front, or deleted, in order to avoid excessive separation or clustering of neighboring marker points. This process ensures that there are approximately two to five marker points per cell of the fixed grid. The marker points are also used for the computations of the surface tension,

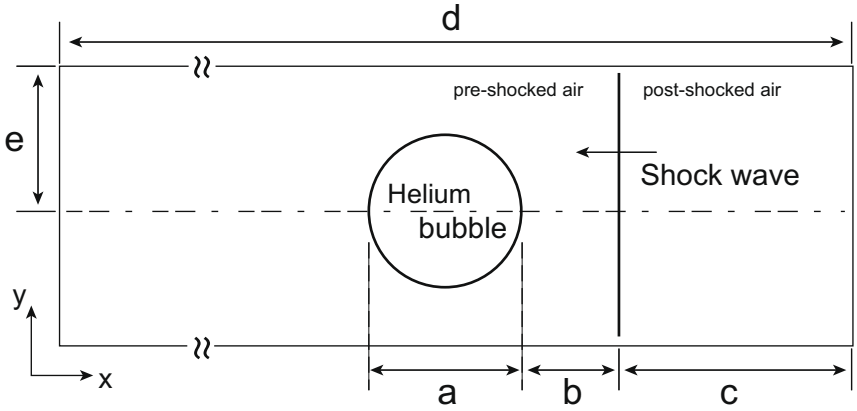
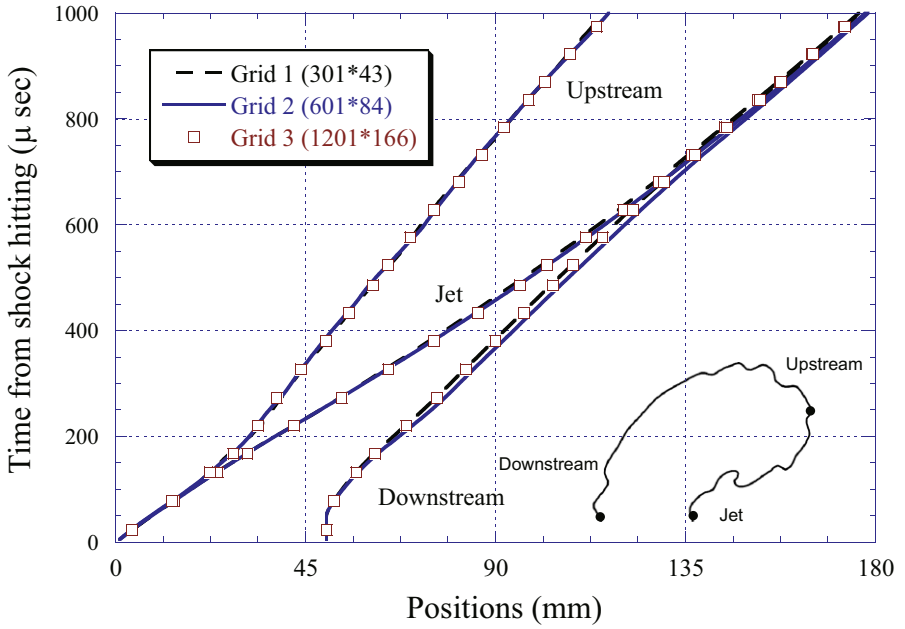
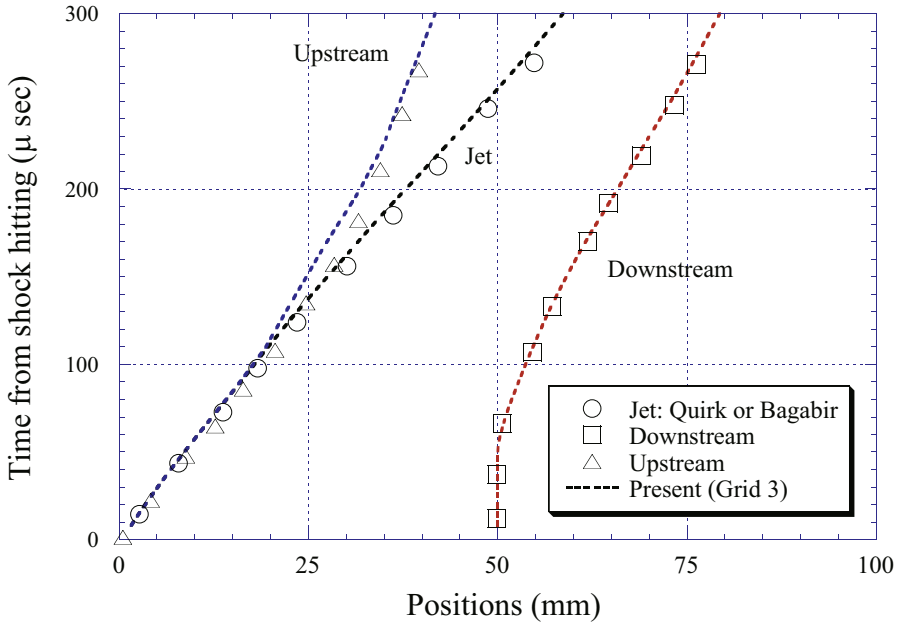


Fig. 5. A schematic of computational domain (not to scale).



(a) Grid refinement tests (whole process)



(b) Comparison with other results at initial stage

Fig. 6. Space–time diagrams for three characteristic interface points.

which is transferred to the fixed grid points in a similar way to the velocity interpolation. Further details and applications of the front-tracking method can be found in Tryggvason et al. [23].

3.3. Ghost-fluid method

In incompressible multiphase flow simulations, the densities are assumed to be constant in each fluid and can therefore simply be updated using the front location. The correct density jump at fluid interfaces is therefore easily guaranteed. The

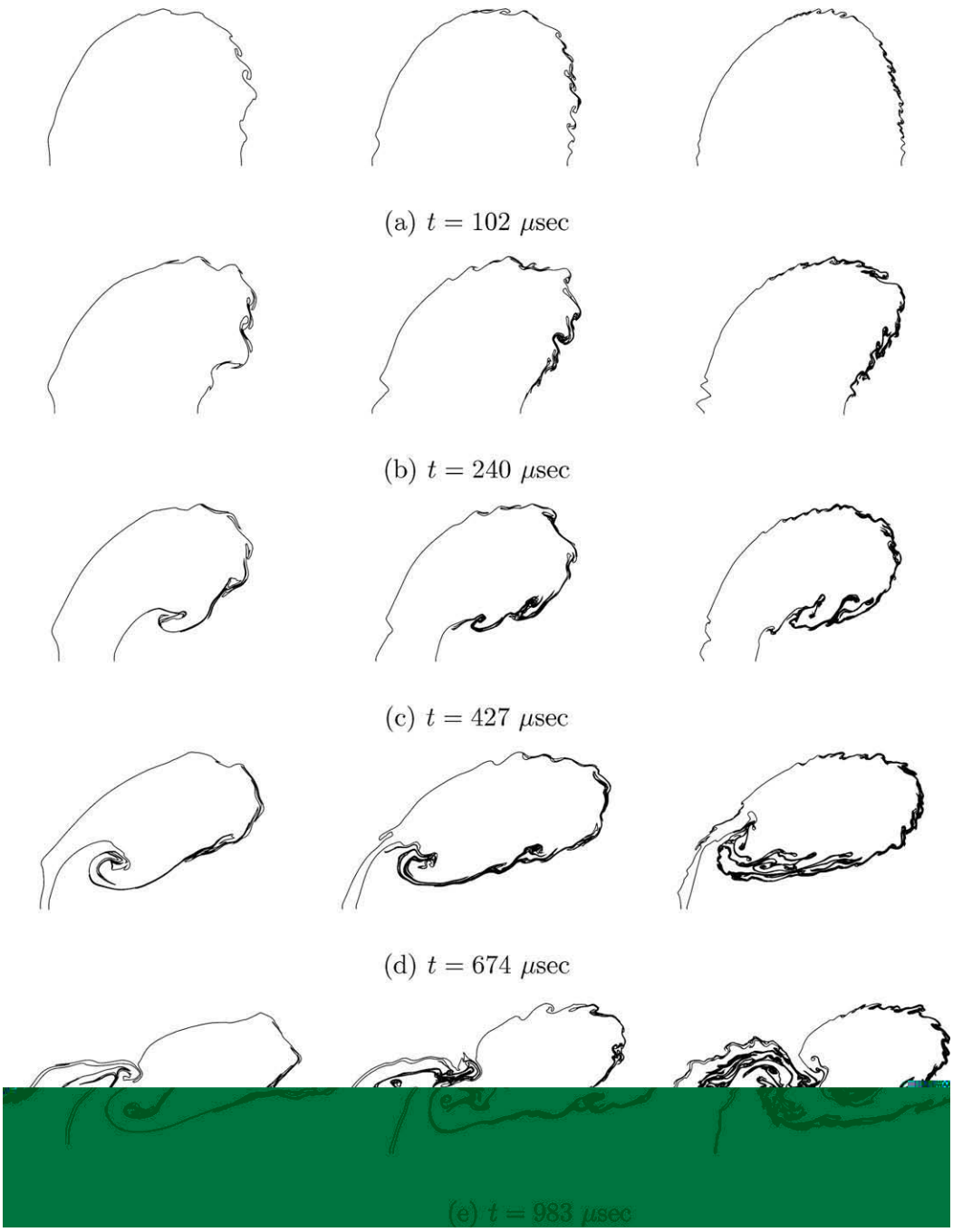


Fig. 7. Grid refinement test (left: 301×43, middle: 601×84, right: 1201×166).

velocity and the pressure are then determined using the updated density field. In compressible flows, on the other hand, the density usually varies in each fluid and must be updated by solving the governing equations.

The ghost-fluid method of Fedkiw et al. [8] offers a way to capture fluid interfaces in compressible flows, avoiding unphysical oscillations and minimizing the smearing of discontinuous variables such as entropy. As indicated in Fig. 2, the ghost-fluid method first defines each fluid domain with the corresponding ghost-fluid region, and the governing equations with the corresponding equation of state in each fluid domain are then solved independently. Finally the solutions in both domains are combined.

In the ghost-fluid regions, as explained by Fedkiw et al. [8,24], discontinuous variables across a fluid interface are given using a one-sided extrapolation and continuous variables are copied from the real fluid on a node by node basis. In the case

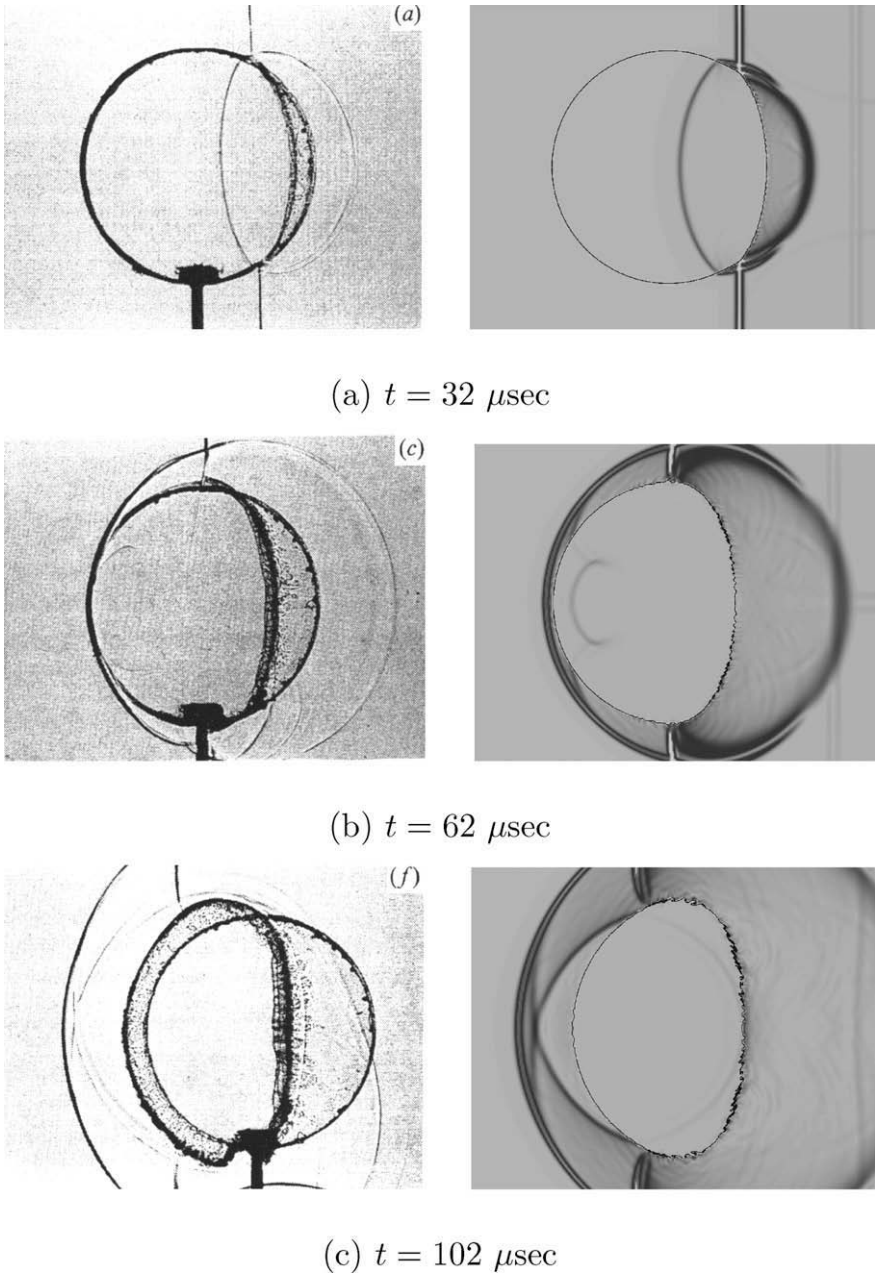


Fig. 8. Comparison between computation (Mach number contours and schlieren images) and experiment (shadowgraphs). Experimental images © Cambridge University Press. Reprinted with permission of the Cambridge University Press 1987.

of the Euler equations, since inviscid fluids are usually assumed to slip at a contact interface, the entropy and the tangential velocity are extrapolated into the ghost-fluid regions and the pressure and the normal velocity are copied from the real fluid. In the case of viscous flows, only the entropy is extrapolated. Other variables are copied. In this way, the ghost-fluid method captures boundary conditions at fluid interfaces implicitly and avoids numerical oscillations at fluid interfaces by extrapolating thermodynamically identical states into ghost-fluid regions.

For large density differences across an interface, such as for gas–water flow, Fedkiw [24] suggested an alternative way of constructing the ghost-fluid states. In addition to the procedure described above, the velocity in the ghost-fluid region for the water and the pressure in the ghost-fluid region for the gas are extrapolated from their real fluid regions, respectively. Although this approach may not be universally effective, as suggested by [14,15], we use it for our gas–water simulation later in this paper.

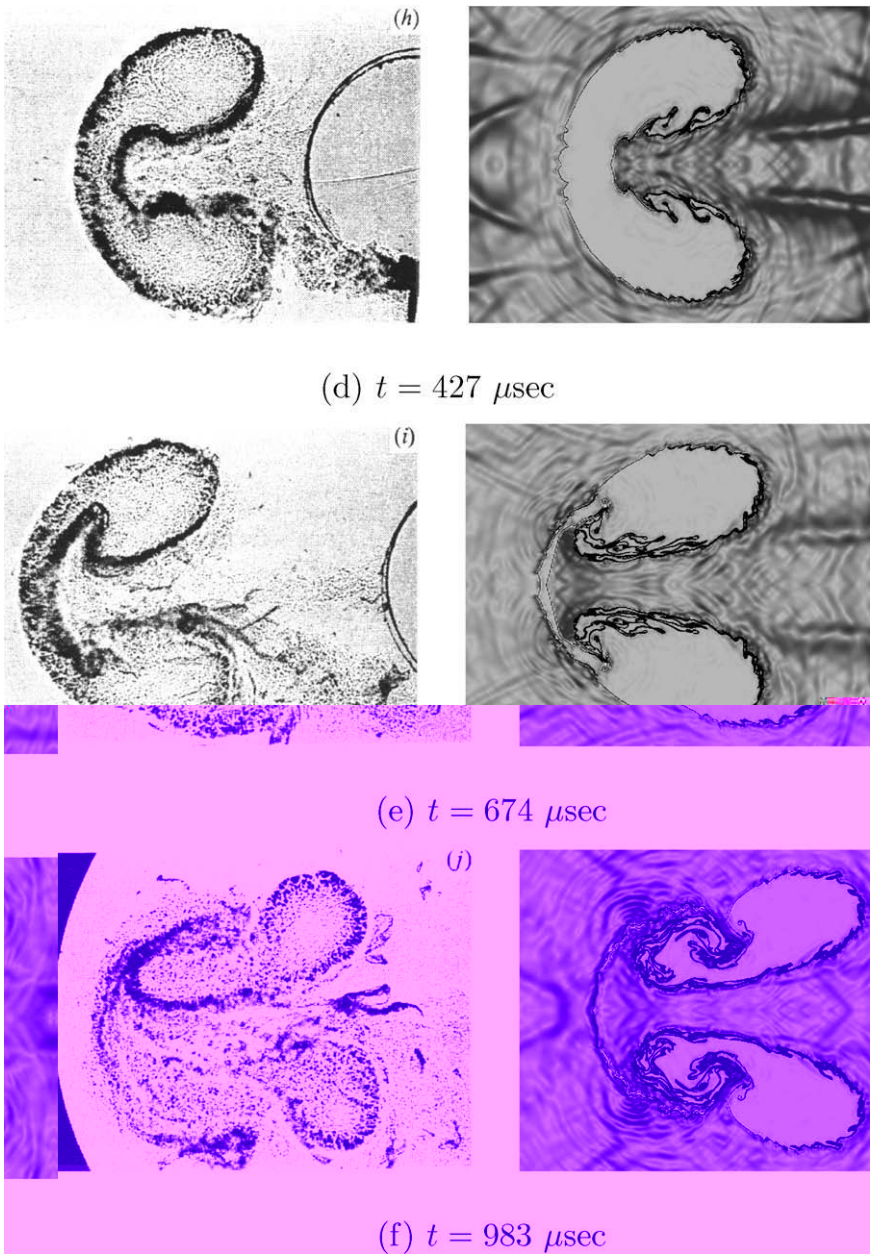


Fig. 8 (*continued*)

In order to extrapolate the discontinuous variables into the ghost-fluid regions, we use the following advection equation,

$$\frac{\partial \phi}{\partial \tau} + n_x \frac{\partial \phi}{\partial x} + n_y \frac{\partial \phi}{\partial y} = 0, \tag{9}$$

where ϕ represents a scalar variable, such as the entropy or a velocity component. n_x and n_y are the components of a unit surface normal vector. This advection equation is solved in fictitious time τ at each time step using a first-order upwind scheme for the spatial variables and the third-order TVD Runge–Kutta method for the time integration. The time step size is selected to be almost the same as, or smaller than, the one for the solution of the governing equations. We take 5–7 grid cells across the interface as the ghost-fluid regions. In this study, the isobaric fix introduced in Fedkiw's original ghost-fluid method is not applied because the influence is considered to be small.

3.4. Determining the ghost-fluid region

Since the fluid interfaces generally move at every time step, it is necessary to continuously reset the ghost-fluid regions. One way to do that is to use the signed distance between each grid point and the interface, as is done in the level-set method. However, the calculation of the signed distance at every time step can be time-consuming, although the number of the grid points to be updated are limited. The convection of the signed distance function would be another possible choice, but may be subject to accumulation of numerical errors. Neither approach is, furthermore, in the "spirit" of the front-tracking method and since we only need an indicator that identify each fluid, it is not necessary to construct the signed distance function.

In order to set the ghost-fluid regions, we use the Heaviside function and the method used in the original front-tracking method [23] to determine the marker function. This is also similar to the approach of Hao and Prosperetti [26]. Since the front marks the jump in the Heaviside function and this jump is translated into a steep gradient on the fixed grid, the gradient of the Heaviside function can be expressed as

$$\nabla H(\boldsymbol{x}) = \int \Delta H(\boldsymbol{x}) \boldsymbol{n} \delta(\boldsymbol{x} - \boldsymbol{x}_f) ds, \tag{10}$$

where $H(\boldsymbol{x})$ is the Heaviside function and $\Delta H(\boldsymbol{x})$ should be equal to 1.0. The discrete version, for the fixed grid, is

$$(\nabla H(\boldsymbol{x}))_{j,k} = \sum_l \Delta H(\boldsymbol{x}) w^l_{j,k} \boldsymbol{n}_l ds_l, \tag{11}$$

where j and k identify the nodes on the fixed grid. l is an index for the marker elements of the front and $w^l_{j,k}$ is the weight for grid point (j,k) with respect to element l. Here we use area-weighting for $w^l_{j,k}$. Finally, by taking the divergence of the gradient, Eq. (11) results in a numerical approximation to the Laplacian:

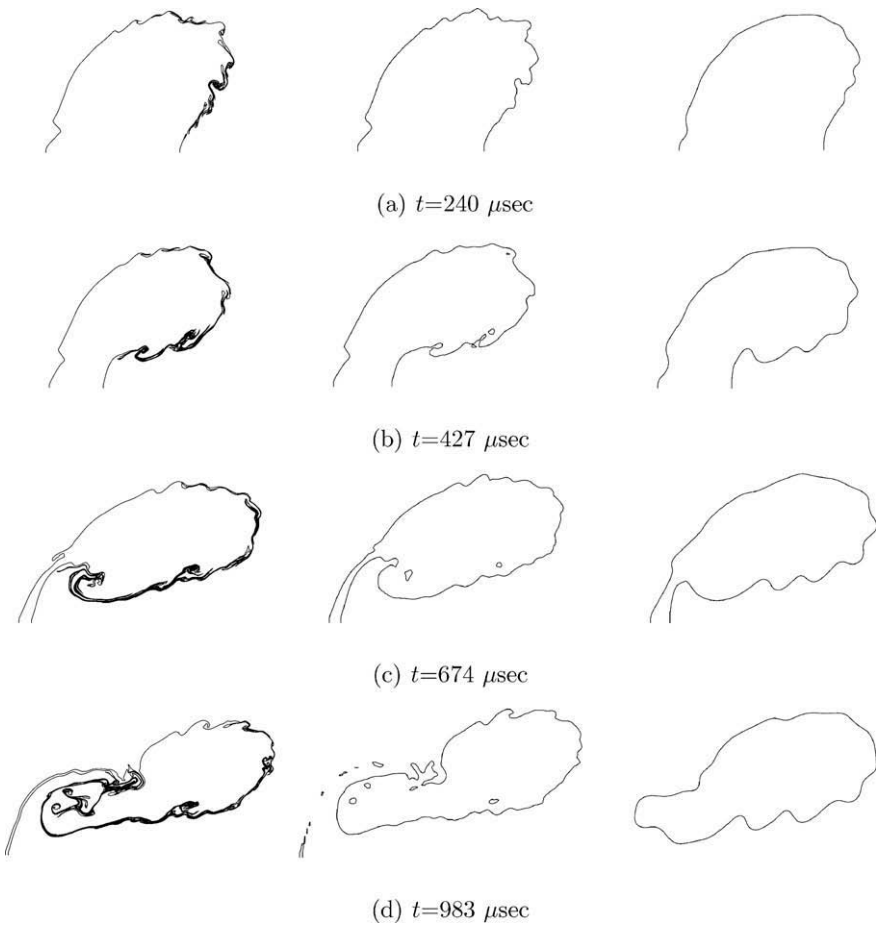


Fig. 9. Comparison between front-tracking/ghost-fluid and level-set/ghost-fluid methods (left: marker points, middle: indicator function, right: level-set).

$$\nabla^2 H(\boldsymbol{x}) = \nabla \cdot (\nabla H(\boldsymbol{x})). \tag{12}$$

In the actual implementation, we solve Eq. (12) by iterating only on a few points close to the fluid interface for a rapid convergence. The ghost-fluid regions on each side are then determined using the sign of the Heaviside function. For example, at a grid point (j,k), we calculate the sign $I(j,k) = H(\boldsymbol{x})_{j,k} \cdot H(\boldsymbol{x})_{m,n}$ over $m = j - iband \sim j + iband$ and $n = k - iband \sim k + iband$. "*iband*" is an arbitrary band number for ghost-fluid region. We usually take $iband = 5$—7. If at least one $I(j,k)$ takes negative value, the grid point (j,k) is considered to be a point in a ghost-fluid region. Finally, if $H(\boldsymbol{x})_{j,k} < 0$, the grid points (j,k) is recognized in a point in a ghost-fluid region for a real fluid region defined by $H(\boldsymbol{x})_{j,k} > 0$ and vice versa.

It should be noted that unlike Hao and Prosperetti [26] who used $H(\boldsymbol{x})$ also for the calculation of the normal surface vector, we use $H(\boldsymbol{x})$ only to decide which fluid the fixed grid point belong to. As described in the next section, the normal surface vector is provided directly from the marker points.

3.5. Extrapolating the surface normal vector

One drawback of the front-tracking method, compared to the level-set method, is that the front-tracking method does not provide any information about the signed distance from the interface, in the whole domain. Thus, as already discussed in the previous section, distinguishing each fluid at every time step involves a separate operation.

The generation of a surface normal vector in the ghost-fluid regions, which is required in order to solve the advection equation, Eq. (9) poses another challenge. In the level-set method, the surface normal vector can be obtained easily everywhere by taking the gradient of the signed distance function. In contrast, in the front-tracking method this information is only available at the marker points. Therefore, we need to propagate the surface normal vector from the maker points into the ghost-fluid region on the fixed grid. To do so, we use a simple algebraic method because only the direction of the surface normal vector needs to be propagated. The method used here is similar to the fast marching method [27] although no partial differential equations need to be solved.

First, as shown in Fig. 3, the surface normal vectors, given at the marker points, are projected onto the fixed grid points using area-weighting. These grid points are marked as *Known points*. The surface normal vectors on the *Known points* are fixed. The surface normal vectors on other grid points are taken to be zero and these grid points are marked as *Far points*. Then, as in the fast marching method, the grid points next to the *Known points* are identified as *Trial points*. On a *Trial point*, possible values can be provided from some of the neighboring *Known points*. If we have three *Known points* to update the value on a *Trial point*, then we take the average of the three values on the *Known points* as the new value on the *Trial point*. A *Trial point* which have received a definite value is upgraded to a *Known point*. The procedure is iterated over all the grid points in the ghost-fluid region until there are no *Far point* left.

This algebraic method is quite robust compared to when an advection equation is solved to march the values out from the front. The most important thing is that we use the accurate surface normal vector calculated from the marker points directly,

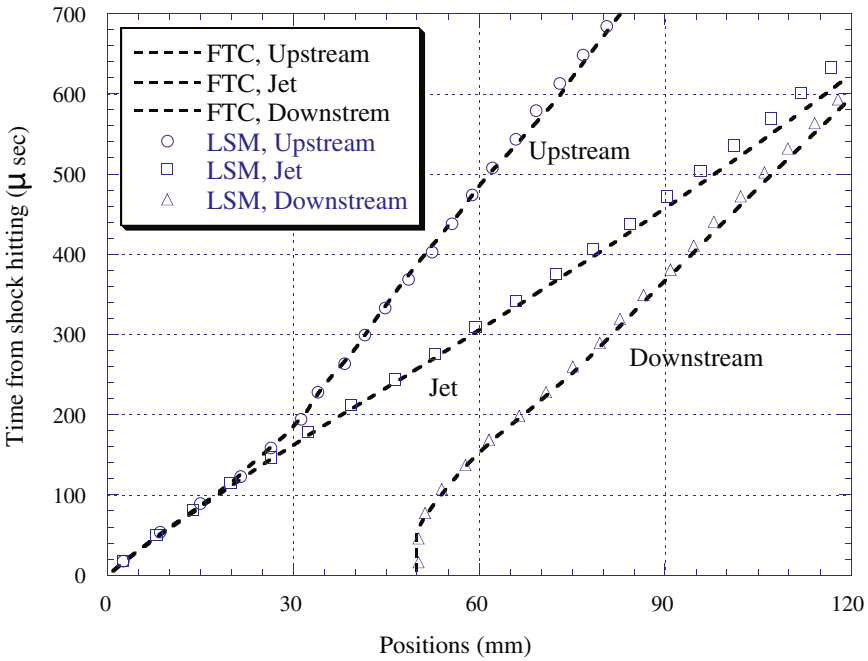


Fig. 10. Comparison with a level-set method in space–time diagrams.

in contrast to obtaining it from the marker function. This means that the method described above makes maximum use of the explicit tracking. Fig. 4 shows an example of the surface normal vector fields obtained by the procedure described above. Even for large interface deformations, the method generates a well-behaved surface normal vector fields.

3.6. Numerical algorithm

The proposed method consists of the following steps:

(1) Update the flow field using the Euler or the Navier–Stokes equations with the proper boundary conditions.
(2) Obtain the velocity of the front marker points and then move the front to a new position using Eq. (8).
(3) Generate the ghost-fluid regions using Eq. (12).

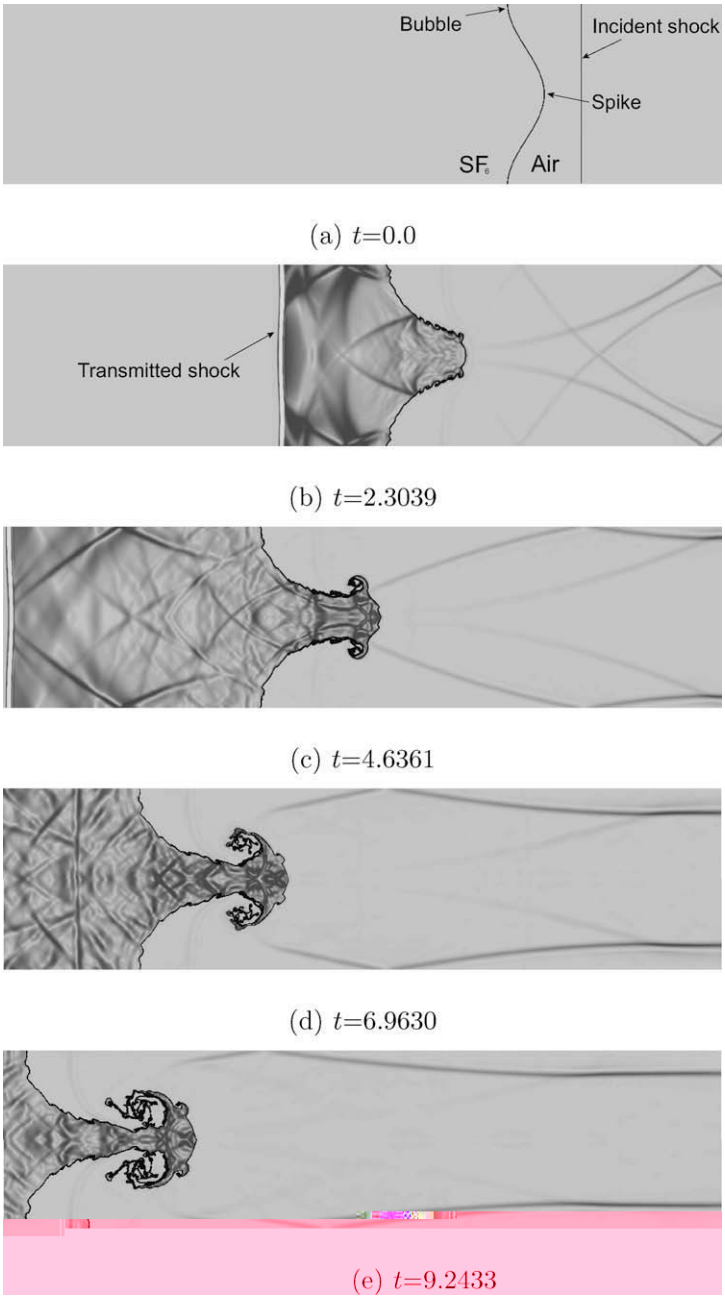


Fig. 11. Richtmyer–Meshkov instability with schlieren-typed image.

(4) Extrapolate the surface normal vector into the ghost-fluid regions.
(5) Solve Eq. (9) to set the boundary conditions in the ghost-fluid regions.
(6) Return to the first step.

These steps are performed three times per one physical time step with the third-order TVD Runge–Kutta method.

4. Results and discussion

To demonstrate the capability and applicability of the proposed front-tracking/ghost-fluid method, in this section we describe several numerical simulations of compressible multiphase flows. It is noted that the CFL numbers used in each case are not the minimum CFL numbers taken for stable computations, although in some cases those might be thought to be quite small.

4.1. Shock–bubble interaction

We first simulate Hass and Sturtevant's experiment [28], in which a shock wave with Mach number of 1.22 hits a helium bubble. This study has been used by several authors [8,29–33] in order to demonstrate the performance of their numerical methods. A very detailed computational analysis was performed by Quirk and Karni [30], using an adaptive mesh refinement strategy, and Fedkiw et al. [8] used data from [28] to demonstrate the applicability of the ghost-fluid/level-set method.

Fig. 5 shows the computational set-up. The shock wave is initially located on the right side of the helium bubble. It propagates to the left and hits the helium bubble. For the two-dimensional cylindrical bubble, the geometrical parameters in Fig. 5 are:

$$a = 50\ \text{mm}, \quad b = 25\ \text{mm}, \quad c = 100\ \text{mm}, \quad d = 325\ \text{mm}, \quad e = 44.5\ \text{mm}.$$

Here we use the Euler equations, without surface tension, as has been done in past computations. We assumed that the flow field is symmetric about the center axis and so only the top half domain is computed. At the axis, the density, velocity in x (or vertical) direction, and pressure are extrapolated and velocity in y (or radical) direction is set to zero. On the left and right boundaries, all variables are simply extrapolated with zero gradient and the upper boundary is treated as slip-wall. The nondimensional initial conditions are given so that the Mach number of the incident shock wave is 1.22:

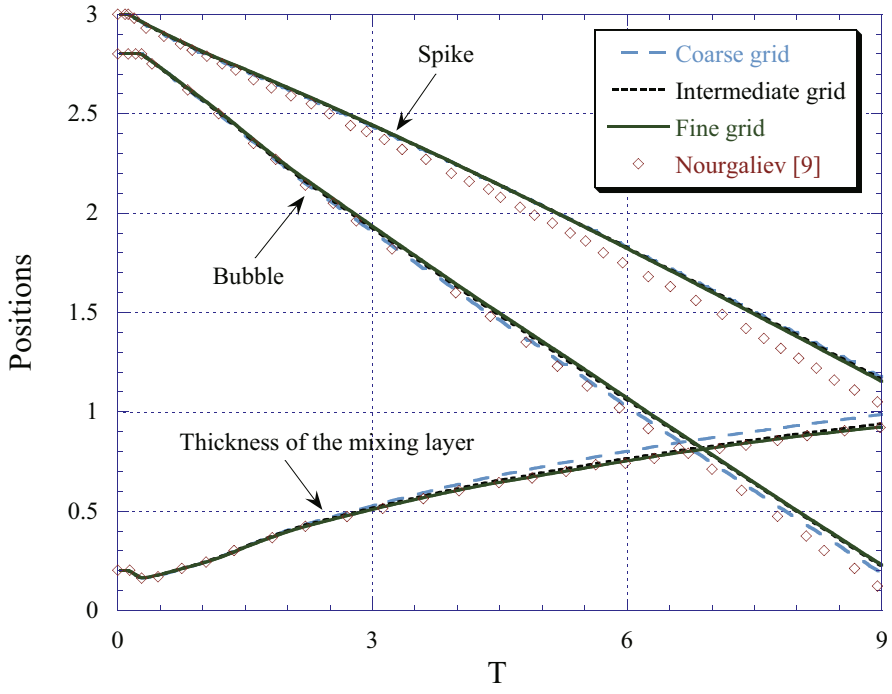


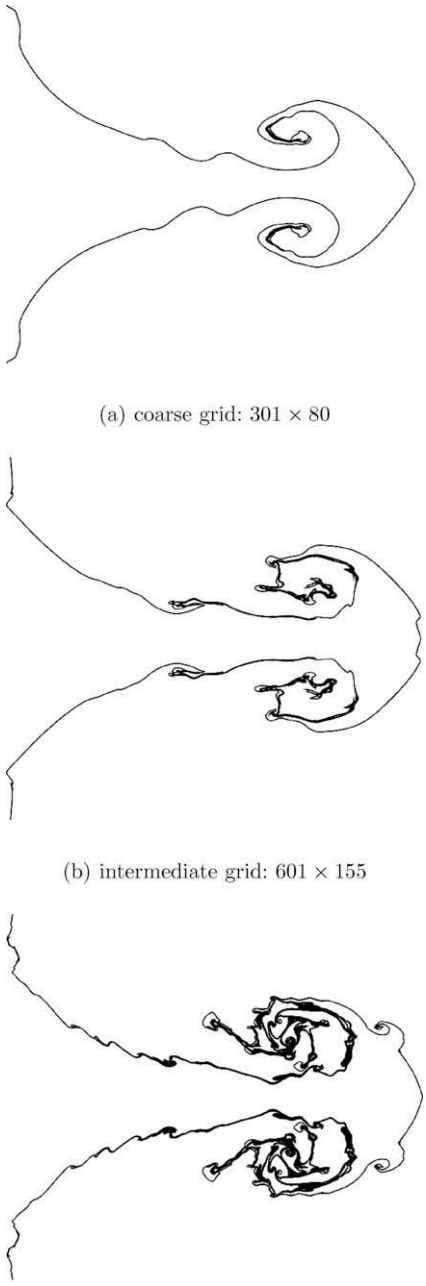
Fig. 12. Grid refinement tests and comparison with other study on time histories of three characteristic positions.

$$\bar{\rho} = 1, \quad \bar{u} = 0, \quad \bar{v} = 0, \quad \bar{p} = \frac{1}{\gamma_{air}}, \quad \gamma_{air} = 1.4, \quad \text{for pre-shocked air},$$

$$\bar{\rho} = 1.3764, \quad \bar{u} = -0.3336, \quad \bar{v} = 0, \quad \bar{p} = \frac{1.5698}{\gamma_{air}}, \quad \text{for post-shocked air},$$

$$\bar{\rho} = 0.1819, \quad \bar{u} = 0, \quad \bar{v} = 0, \quad \bar{p} = \frac{1}{\gamma_{air}}, \quad \gamma_{helium} = 1.648, \quad \text{for helium}.$$

Note that we use the speed of sound and the bubble diameter for the nondimensionalization. The time step size is selected such that the *CFL* number is 0.2.



(a) coarse grid: 301×80

(b) intermediate grid: 601×155

(c) fine grid: 1201×305

Fig. 13. Comparison of interfaces at $t = 9.24$, showing effect of grid resolution.

4.1.1. Grid refinement test

We first present grid refinement test using three different grids: coarse (301 × 43 mesh points), intermediate (601 × 84), and fine (1201 × 166). The initial bubble diameter is resolved with about 40, 80, and 160 cells, respectively.

Fig. 6(a) shows the space–time diagram for three characteristic points (Jet, Downstream, and Upstream shown in the figure) during the whole process, with the effect of grid resolution. The three grids give mostly identical results for the location of these points versus time. The centroid velocity of the helium bubble, found by taking the volume average of the velocity inside the bubble, was also identical on the three grids (not shown here). These grid refinement test, therefore, shows the new method gives a converged solution for the overall bubble motions (not for the small-scale interface structures). Fig. 6(b) presents a comparison of the space–time diagram with earlier computational results from [30,33] during the early stage (since the two results are almost identical, only one result is plotted in the figure). 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 computed on the three grids. Overall, the evolution of the bubble shape is similar on the three grids, despite the large interface deformation. The present results are in good qualitative agreement with the results reported in many earlier studies (see e.g. [30]) for the entire process. With the higher grid resolution, the shear instability on the fluid interface is enhanced and the interface structure becomes more complex. The emergence of small scales can be clearly seen as the grid is refined. In a detailed look of the results, some small difference can be found in the formation of the vortical structure due to the jet evolution. The jet rolls-up in the clockwise direction and then penetrates into the helium bubble from the top side, inducing another jet of ambient air.

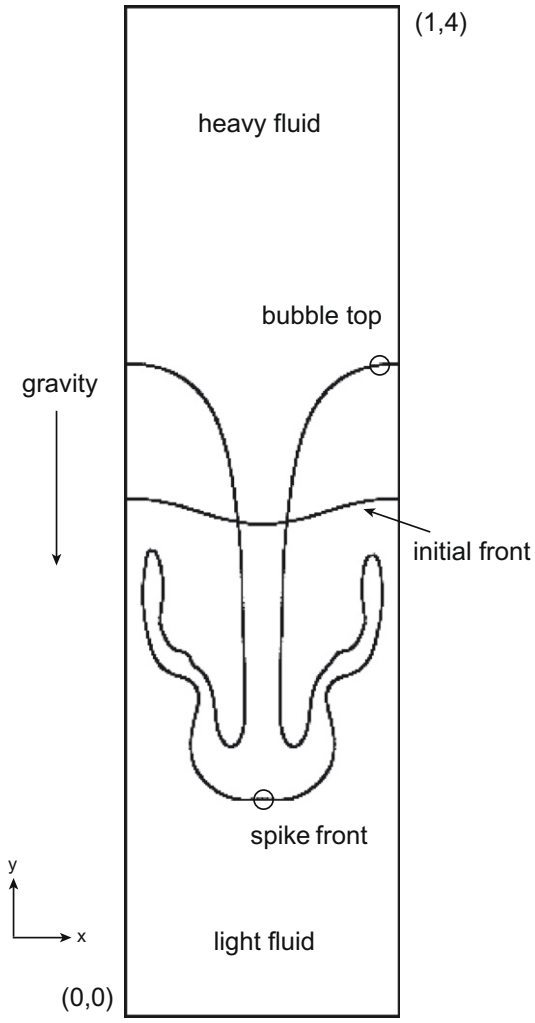
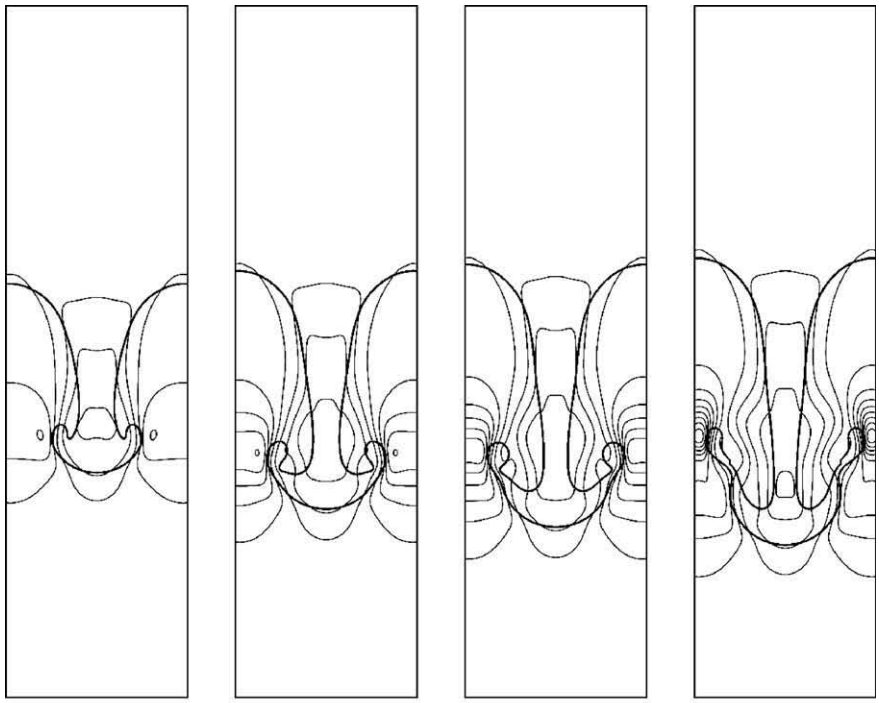


Fig. 14. Configuration of initial front and locations where data are taken (spike front and bubble top).



(a) compressible flow simulation

Fig. 15. Comparison between compressible and incompressible flow simulations on Rayleigh–Taylor instability. The times are 0.654, 0.756, 0.810, 0.865.

4.1.2. Comparison with the experiment

A sequence of shadow-photographs for the interaction of a shock wave with a helium bubble were obtained in the experimental study of Hass and Sturtevant [28]. In this section, we make a qualitative comparison with their results. Fig. 8 shows a sequence of experimental shadow images and the corresponding computational schlieren-type images. The schlieren-type image is created with a nonlinear shading function used in Quirk and Karni [30] in order to accentuate weak flow features. The shock wave inside the bubble moves faster than the one in the surrounding air due to the different speed of sound (Fig. 8(a) and (b)). The upstream side of the bubble becomes flat after the shock hits it and then the flow induced by the passage of the shock wave deforms the bubble from its center as shown in Fig. 8(c) and (d). The ambient fluid piercing the bubble (usually referred to as the jet) grows and rolls-up in a clockwise direction as mentioned before (Fig. 8(e) and (f)). Obviously, the general trend, including the bubble shape and the refracted shock wave, are quite similar in the experimental results.

4.1.3. Comparison with level-set method

In this section, we compare our results with results obtained using a level-set method [25]. In previous implementation of the ghost-fluid method level sets have generally been used to track the interface. In order to keep everything else in the code unchanged, we replaced the front-tracking part with our own implementation of a level-set method. We solved the advection equation for the level-set function and the reinitialization equation with a fifth-order Weighted Essentially Non-Oscillatory (WENO) scheme proposed by Jiang and Peng [34]. In the reinitialization equation, the modified sign function introduced by Peng et al. [35] was used. We iterated the reinitialization equation until convergence at every time step with a convergence criterion and a time step size suggested by Sussman et al. [36], although doing reinitialization at every time step may be excessive. We point out that improved level-set algorithms have been proposed by several researchers [37–39] and the results from the present level-set algorithms do not represent the capability of the latest level-set algorithms. They do, however, represent a fairly common implementation of the level-set method. Except for capturing the fluid interfaces with a level-set function, the numerical technique is the same as described in the previous section. The intermediate grid is used here.

Fig. 9 shows a comparison between the front tracking and the level-set methods at four different times. The overall results from the two approaches are almost identical, but the new front-tracking method provides more detailed interface structures due to shear instability and the jet evolution. These features are also reflected in the indicator function generated from the marker points as seen in Fig. 9. The fine structure of the interface remains in the indicator function, as compared with the level-set method on the same grid resolution.

A comparison of the time histories of the helium bubble shape is shown in Fig. 10. They are in good agreement for the three characteristic points, although slight differences appears at later time, especially in the jet evolution. The jet velocity in the front-tracking method becomes gradually higher a bit than that in the level-set method.

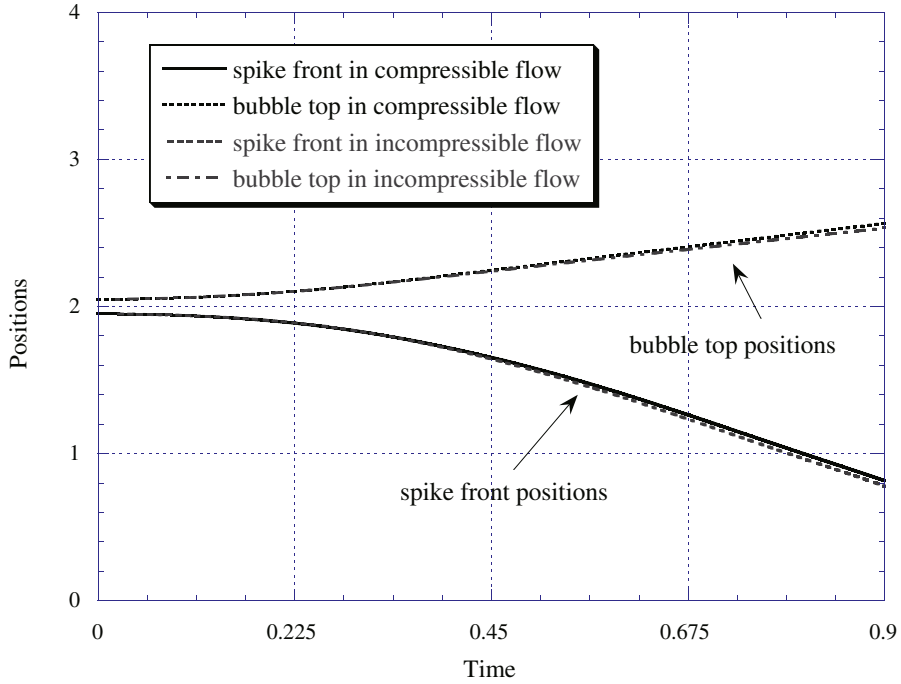
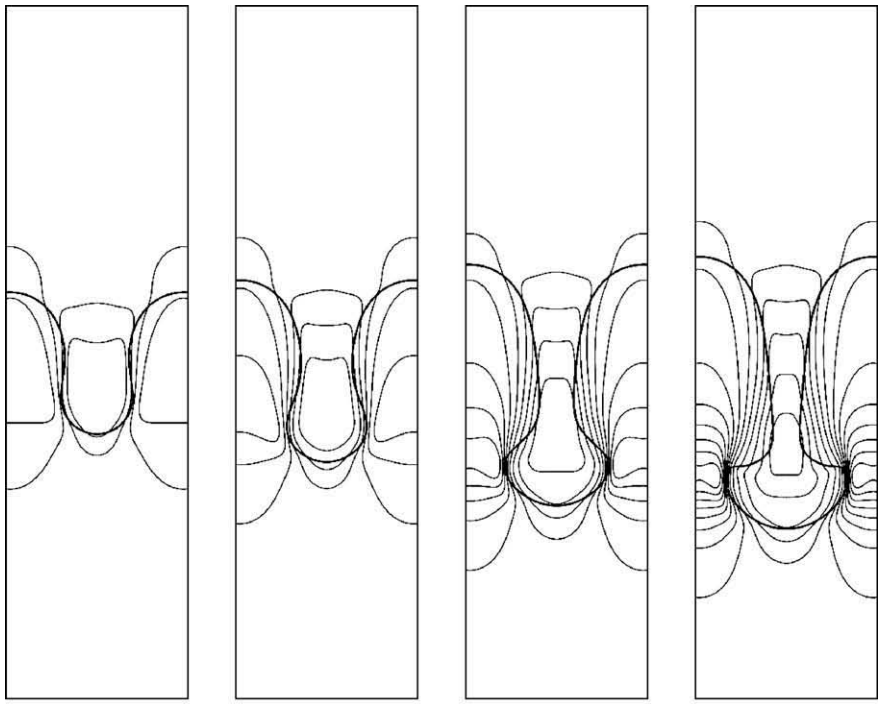
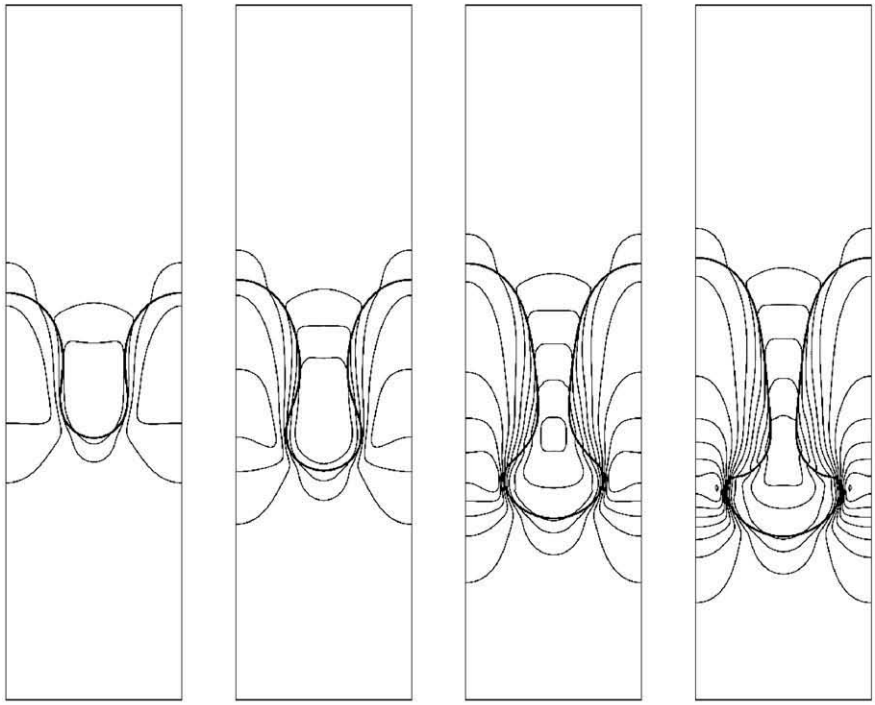


Fig. 16. Comparison of spike and bubble motions between compressible and incompressible flow simulations.



(a) compressible flow simulation



(b) incompressible flow simulation

Fig. 17. Effect of surface tension on Rayleigh–Taylor instability. The times are 0.77, 0.88, 1.03, 1.08.

the positions of the spike and bubble are a little bit different, the evolution of the thickness of the mixing layer is very similar).

4.3. Rayleigh–Taylor instability

To show an example where a body force, such as gravity, and surface tension are added to the Navier–Stokes equations we simulate the classic Rayleigh–Taylor instability. Many numerical simulations of the Rayleigh–Taylor instability have been performed (see e.g. [42–44]), most assuming incompressible flows.

The numerical conditions shown in Fig. 14 are taken from [43,44,46]. The computational domain is a 1×4 rectangular box discretized using 92×351 grid points. A heavy fluid, whose density is $\rho_1 = 1.225$, initially occupies the upper part and a light fluid, whose density is $\rho_2 = 0.1694$, occupies the lower part. The initial velocity field is zero. The Reynolds number based on gravity ($Re = \frac{\rho_1\sqrt{gL}L}{\mu}$) is 1225.19, where $g = 9.8$, $L = 1$, and $\mu = 0.00313$ are taken. The initial interface is created using $y(x) = 2.0 + 0.05\cos(2\pi x)$ to impose an initial perturbation. Slip-wall boundary conditions are applied on the upper and lower boundaries and the left and right boundaries are periodic. The time step size is calculated using $CFL = 0.6$.

We also did an incompressible flow simulation ourselves to make sure the conditions for both simulations are identical. The numerical method used for the incompressible flow simulation is described in [23] and has been fully validated through many multiphase flow simulations. For the parameters used here, the results should be nearly identical.

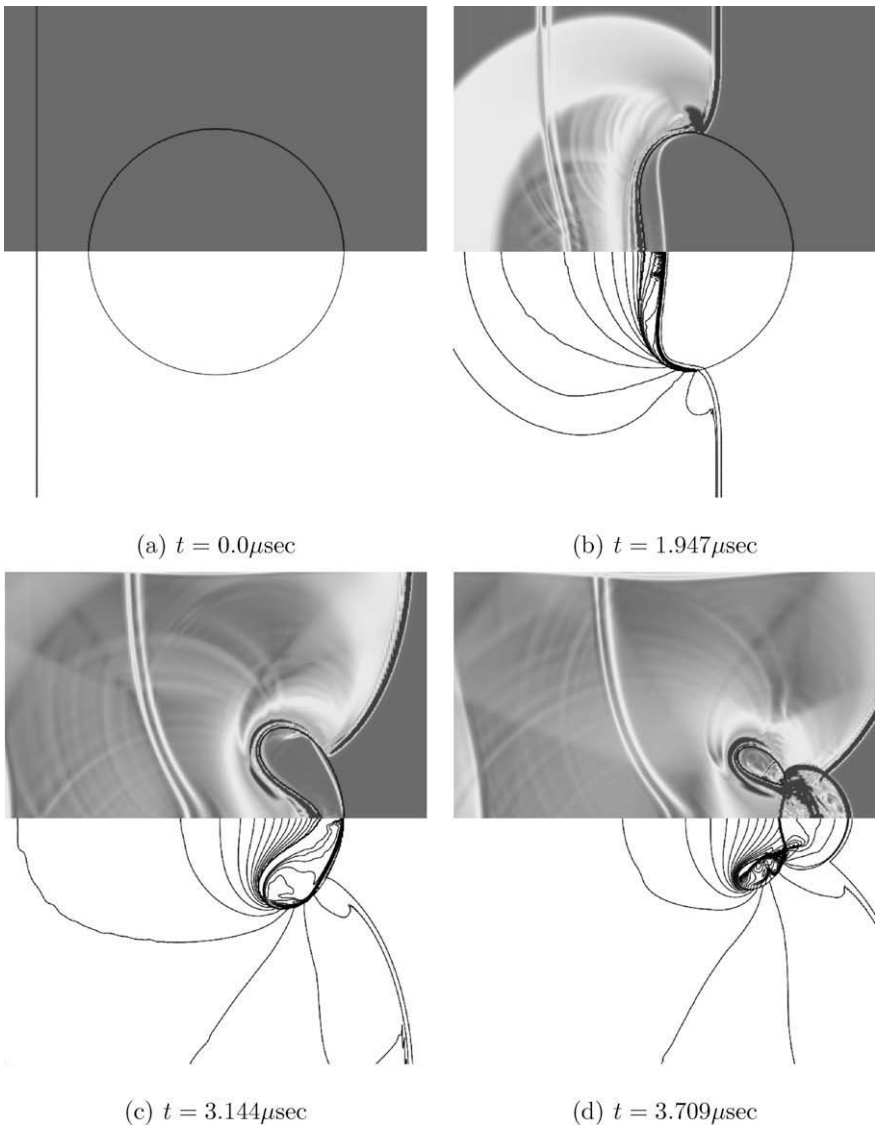


Fig. 19. Collapse of air bubble in water: density gradients image (upper) and Mach number contous (lower).

Fig. 15 shows the interface at four times, without surface tension, comparing the results of the new method for compressible flow with those of incompressible flow simulation. The contours show the magnitude of the vertical velocity. The heavy fluid falls into the light fluid due to gravity and the interface (the spike) forms a mushroom shaped end. The behavior is well captured by the new method (compressible flow simulation) and the results are in good agreement with those obtained by the incompressible flow simulation. The results also compare well with other numerical results [43,44,46].

Fig. 16 shows the locations of the tip of the spike and the top of the bubble versus time (the locations of the spike front and bubble top are indicated in Fig. 14). There is no difference in either the locations of the tip of the spike or the top of the bubble between the two simulations.

Fig. 17 illustrates the interface evolution when surface tension is included. The value of the surface tension is $\sigma = 0.1531$, which is taken to be slightly lower than the one corresponding to the most unstable wavelength [45,46]. Obviously the presence of the surface tension reduces the growth rate of the interface instability compared to the case without the surface tension. Fig. 18 shows the locations of the tip of the spike and the top of the bubble. The results without surface tension, from Fig. 16, are also plotted for comparison. It is clear that the main effect of the surface tension is to reduce the growth rate of the spike. As in the no surface tension case, there is no difference between compressible and incompressible flow simulations. These results also compare well with earlier results [46].

4.4. Gas–liquid interface tests

Our final example consists of two simulations of problems with a gas–liquid interface. One is a cylindrical air bubble hit by a shock wave in water and the other one is a cylindrical water droplet hit by a shock wave in gas. This is to test the method for multiphase flow with large density jump across the interface. Here we use the Euler equations and the constants in Eq. 4 are taken as

$$\gamma = 1.4, \quad \Pi = 0 \quad \text{for air},$$
$$\gamma = 4.4, \quad \Pi = 6 \times 10^8 \text{ [Pa]} \quad \text{for water}$$

for the simulation of a cylindrical air bubble in water. The computational domain is the one shown in Fig. 5. The various lengths are taken as:

$$a = 6.0 \text{ mm}, \quad b = 1.2 \text{ mm}, \quad c = 1.8 \text{ mm}, \quad d = 15.0 \text{ mm}, \quad e = 6.0 \text{ mm}.$$

The initial conditions used in earlier studies [47,14,9] were used:

$$\rho = 1000.0 \text{ [kg/m}^3\text{]}, \quad u = 0, \quad v = 0, \quad p = 1 \times 10^5 \text{ [Pa]} \quad \text{for pre-shocked water},$$
$$\rho = 1323.65 \text{ [kg/m}^3\text{]}, \quad u = -681.58 \text{ [m/s]}, \quad v = 0, \quad p = 1 \times 10^9 \text{ [Pa]} \quad \text{for post-shocked water},$$
$$\rho = 1.0 \text{ [kg/m}^3\text{]}, \quad u = 0, \quad v = 0, \quad p = 1 \times 10^5 \text{ [Pa]} \quad \text{for air bubble}.$$

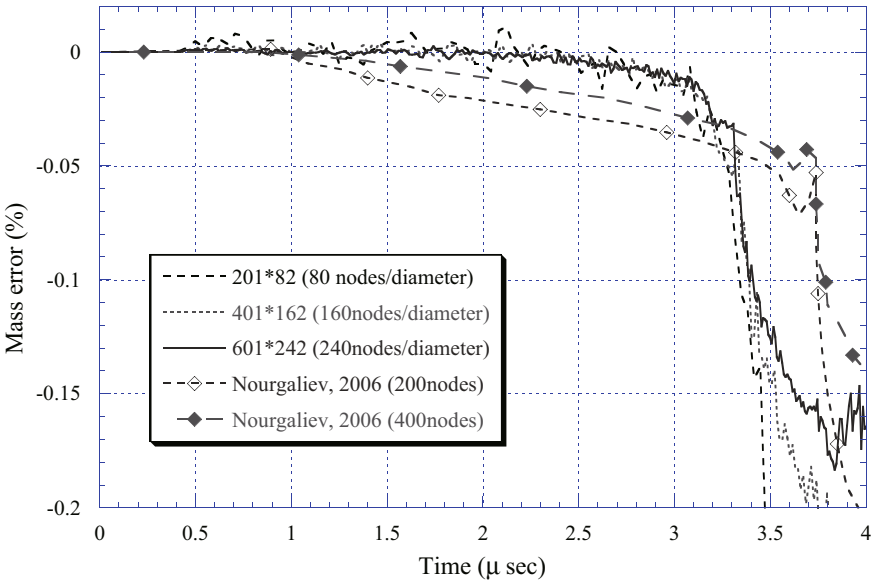


Fig. 20. Mass conservation error with effect of grid resolution and with a result from [9].

Only the upper half domain was considered by assuming a symmetry across the centerline. All variables on the left, right and top boundaries were extrapolated assuming a zero gradient. The number of the grid point is 601×242. The time step size was selected such that the *CFL* number is 0.2.

Fig. 19 shows the response of the air bubble to the shock wave passage at four times. The density gradients, generated as in [30], and the Mach number contours are shown. The overall bubble deformation and behavior is similar to that of the case simulated earlier (see Fig. 8) despite the much different density ratio. In the present case, however, the air bubble finally disappears completely. The result also agrees very well with earlier results [9,14,47], although the timing of the breakup is slightly different between the different references. No numerical oscillations near the interface were observed during the computations.

Fig. 20 shows the time histories for the mass conservation errors for the three grid resolutions (two coarser grids, 201×82 and 401×162, are added) along with results from [9]. The mass is calculated as $M(t) = \sum_{bubble} \rho \Delta V_{j,k}$ assuming that the density in each cell is constant (areas of cut cells are not accounted for) where $\Delta V_{j,k}$ is the volume of the computational cell having indices j and k. The mass conservation errors of the new method are within 1% for any grid resolutions, until the breakup occurs, showing good mass conservation properties for the new method, even on coarse grids. After the breakup ($t > 3$ μs), the mass conservation error depend strongly on the grid resolution.

The second example is the interaction of a shock wave in air with a cylindrical water droplet. This is an inverse-impedance problem of the first example. A wide computational domain is used in order to avoid boundary influences during the entire evolution of a droplet. The lengths are as:

$$a = 4.8 \text{ mm}, \quad b = 1.6 \text{ mm}, \quad c = 45.6 \text{ mm}, \quad d = 100.0 \text{ mm}, \quad e = 40.0 \text{ mm}.$$

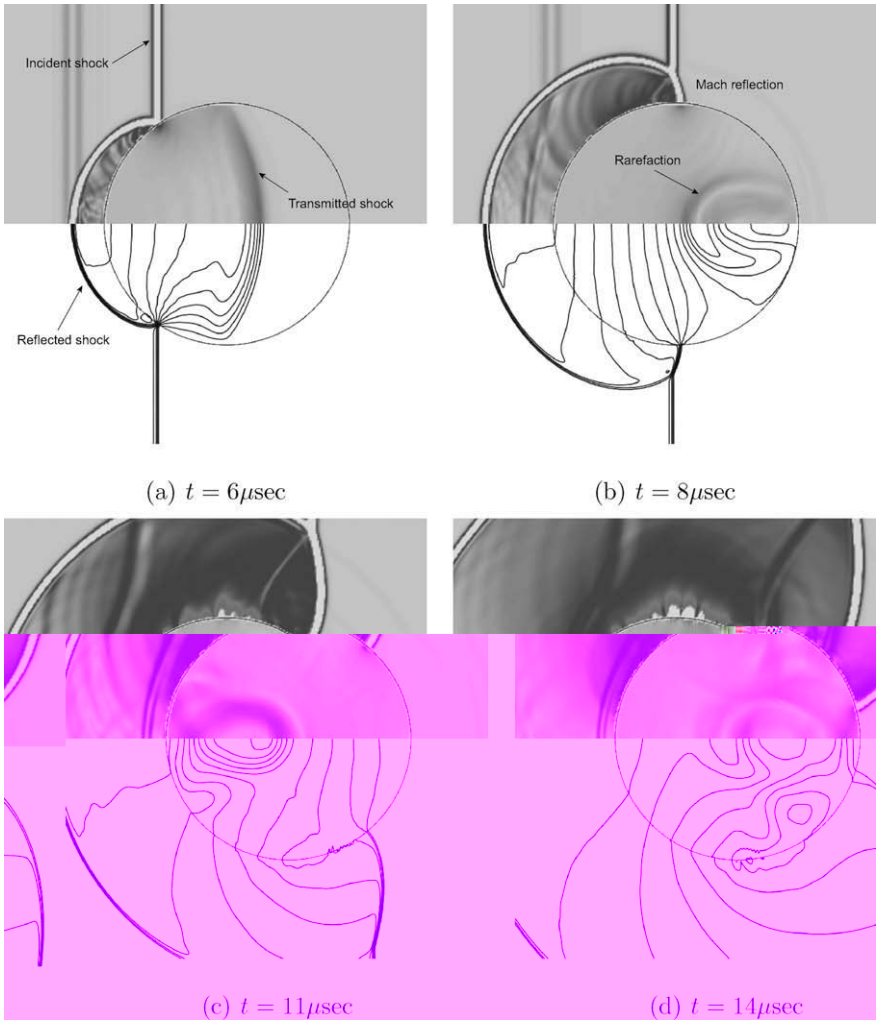


Fig. 21. Interaction of a shock wave in air with a water droplet (top: schlieren-typed image, bottom: pressure contours).

An initial condition with a planar incident $M_{sh} = 1.47$ shock wave used in the experiment in [48] and for earlier numerical studies [9,32,48,49] is used:

$$\rho = 1.0\ [\mathrm{kg/m^3}],\quad u = 0,\quad v = 0,\quad p = 1 \times 10^5\ [\mathrm{Pa}]\quad \text{for pre-shocked air},$$
$$\rho = 1.811\ [\mathrm{kg/m^3}],\quad u = -246.24\ [\mathrm{m/s}],\quad v = 0,\quad p = 2.35 \times 10^5\ [\mathrm{Pa}]\quad \text{for post-shocked air},$$
$$\rho = 1000.0\ [\mathrm{kg/m^3}],\quad u = 0,\quad v = 0,\quad p = 1 \times 10^5\ [\mathrm{Pa}]\quad \text{for water droplet}.$$

A detailed description at the early stage of the shock–droplet interaction can be found in [9]. Non-reflection boundary condition [41] is used for all boundaries. Two grids, 375×300 (24 nodes/diameter) and 751×602 (48 nodes/diameter), are used. These grid resolutions are not enough to capture many detailed flow features as seen in [9]. We have therefore re-simulated the evolution for the early stage ($t < 30\ \mu s$) using a smaller computational domain of size 32 mm $\times$ 32 mm and using a finer grid resolution of 1001×1001 grid points (200 nodes/diameter). The *CFL* number is 0.4.

The early stage of the interaction of a shock wave with a water droplet is shown in Fig. 21 with shlieren-typed image and pressure contours. The complex flow features, such as the reflected shock wave, the transmitted shock wave and the rarefaction waves, are captured relatively well with the new method on the grid resolution used here. Due to different parameters in the stiffened equation of state, there are small differences, however, in propagation speeds of the transmitted shock wave and the rarefaction wave inside the droplet between the present results and those in [9]. Since some small noise can be seen in the vicinity of the interface in Fig. 21(c) and (d), finer grid resolution is needed to improving the present solution and numerical stability as suggested in [9].

Fig. 22 presents a comparison of the unsteady drag coefficient with two other computational results, at the early stage of the shock–droplet interaction. The drag coefficient is defined as $C_d = \frac{F}{0.5\rho_g u_g^2 d}$, where F is the unsteady drag force, ρ_g and u_g are the density and velocity of gas phase, and d is the diameter of droplet. For comparison, another case (a $M_{sh} = 1.3$ shock wave and d = 6.4 mm droplet diameter) was also simulated and the results are plotted in the same figure. The present results are in good agreement with computational results of Igra and Takayama [48] and Chen [49] and for both Mach numbers.

Fig. 23 shows the entire breakup of a water droplet hit by a shock wave, for much longer time than examined in Figs. 21 and 22. After the shock passes, large recirculation regions are formed behind the drop. The pressure difference between the upstream facing part of the drop and the wake flattens the drop, eventually leading to thin ligaments pulled from the side. Although first stretched directly outward, eventually the ligaments are carried downstream by the flow, thus stripping water from the original drop. For the parameters used here, some fluid does, however, remain in the original drop. We emphasize

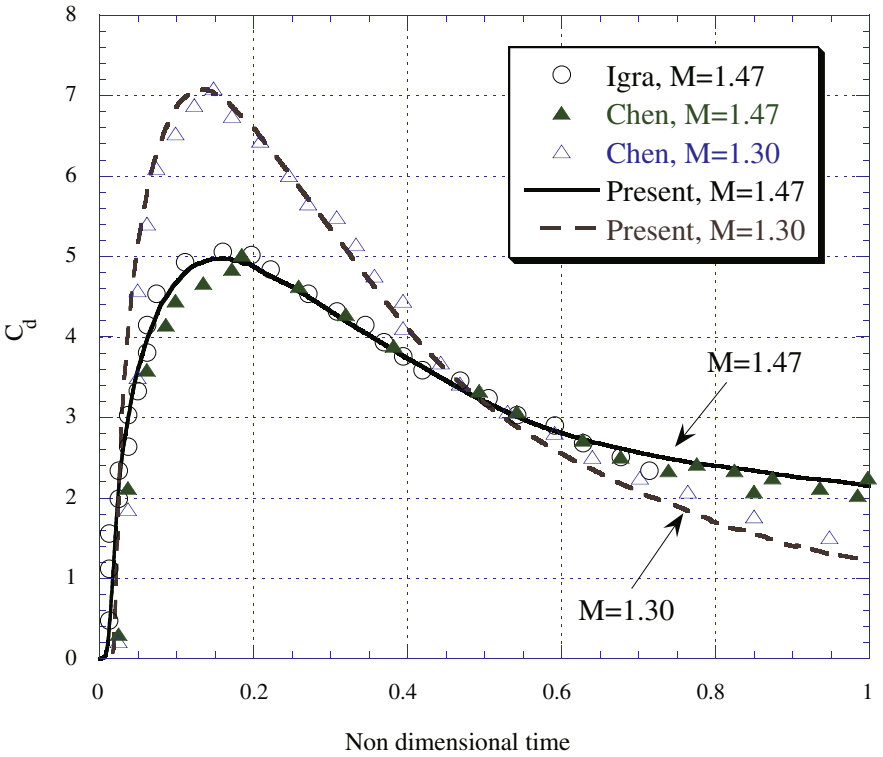


Fig. 22. Comparison of unsteady drag coefficient at early stage.

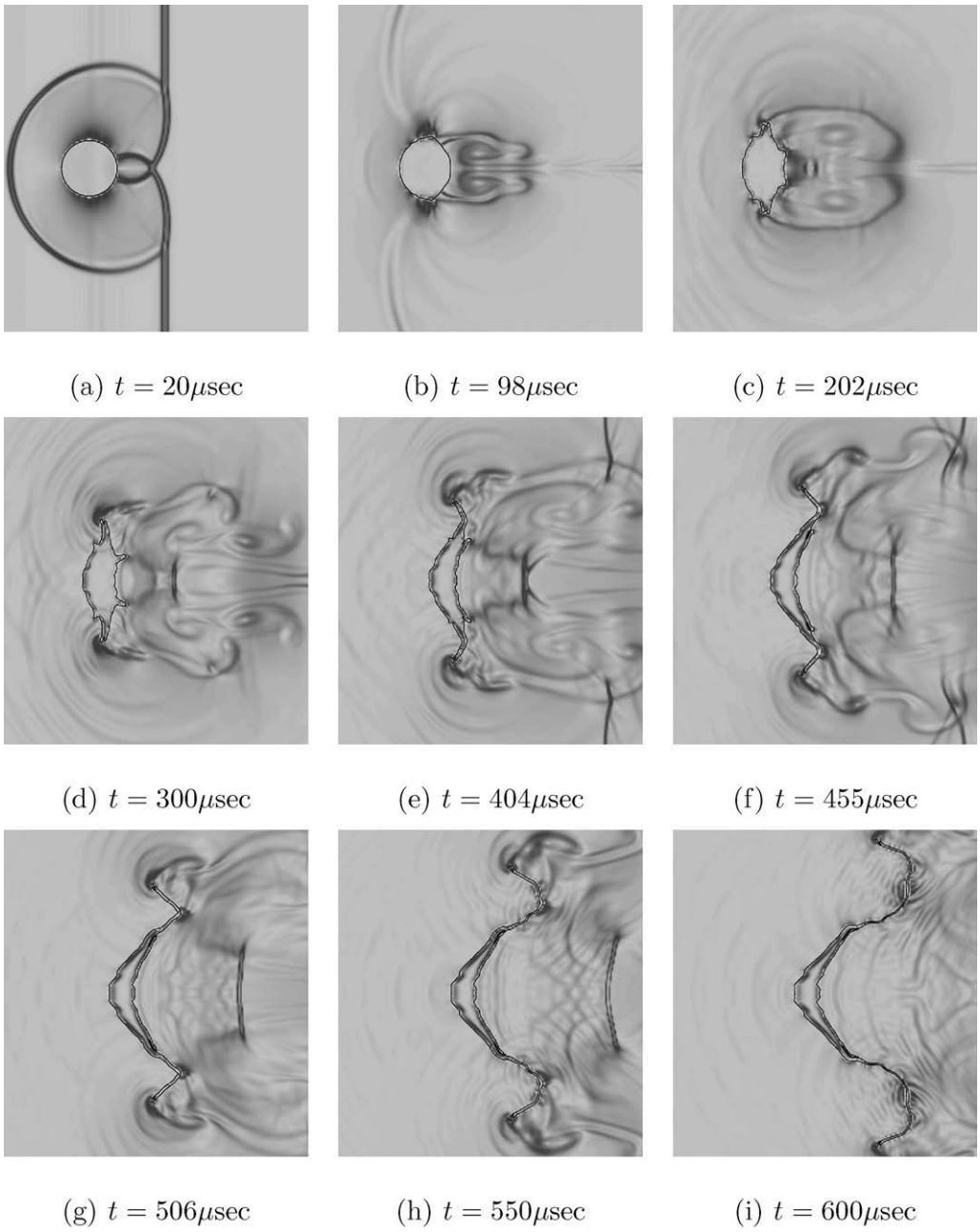


Fig. 23. Entire evolution of a water droplet in air due to shock hitting.

that this simulation is for a water drop in air, demonstrating the capability of the method to deal with large density differences.

Further simulations of the break up of a drop, including grid refinement tests and studies of the effect of surface tension, are currently in progress and further results will be reported in a later publication.

5. Conclusions

We have introduced a front-tracking/ghost-fluid method for simulations of fluid interfaces in compressible flows. The method captures fluid interface using connected marker points and defines fluid interface conditions with the ghost-fluid method, avoiding unphysical oscillations and minimizing the smearing of discontinuous variables. The implementation of the new procedure is relatively simple. We believe that the method can be extended to fully three-dimensional flow. A

few simulations have been performed in order to demonstrate the capabilities of the method. A shock–bubble interaction study showed, by grid refinement tests and comparisons of characteristic points with other results, that the method can capture the detailed small-scales bubble structure with good accuracy. The computational results for the bubble deformation agreed very well with experimental observations. A simulation of the Richtmyer–Meshkov instability with grid refinement tests demonstrated the capability of the method to follow complicated interface behavior caused by the passage of a shock wave, showing the grid convergence for the overall interface motions and the good agreement of the spike and bubble evolutions with the other computational result. In simulations of the Rayleigh–Taylor instability, it was shown that the method can handle the inclusion of surface tension, and that in the limit of slow motion, the results agree with those obtained by computations assuming incompressible flow. Simulations of compressible gas–water system showed that the new method can be applied to multiphase flows with large density jump across the interface. It was shown that the results for an air bubble hit by a shock wave in water are in good qualitative agreement with earlier results and that the new method has good mass conservation property, even for relatively low grid resolutions. The results for a water droplet with a shock wave in air showed that the main flow features at the early stage, including the unsteady drag coefficient, are good agreement with other earlier studies. The simulation of the entire evolution of the droplet indicated the method's applicability to explore dynamics of liquid droplets in compressible flows. The results presented demonstrate that the new method can accurately simulate multiphase flows for a wide range of density ratios and including the effect of surface tension.

Acknowledgment

This work is supported by the Basic Science Research Program of the Sumitomo Foundation.

References

[1] J. Yang, T. Kubota, E.E. Zukoski, Applications of shock-induced mixing to supersonic combustion, AIAA J. 31 (1993) 854–862.
[2] J.C. Hermanson, Dynamics of supersonic droplets of volatile liquids, AIAA J. 45 (3) (2007) 730–733.
[3] I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics, J. Comput. Phys. 62 (1986) 83–110.
[4] C.L. Gardner, J. Glimm, O. McBryan, R. menikoff, D. Sharp, Q. Zhang, The dynamics of bubble growth for Rayleigh–Taylor unstable interfaces, Phys. Fluids 31 (1988) 447–465.
[5] J. Glimm, J.W. Grove, X.L. Li, K. -H Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM J. Sci. Comput. 19 (3) (1998) 703–727.
[6] J.-P. Cocchi, R. Saurel, A Riemann problem based method for the resolution of compressible multimaterial flows, J. Comput. Phys. 137 (1997) 265–298.
[7] W. Mulder, S. Osher, J.A. Sethian, Computing interface motion in compressible gas dynamics, J. Comput. Phys. 100 (1992) 209–228.
[8] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–494.
[9] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, Adaptive characteristics-based matching for compressible multifluid dynamics, J. Comput. Phys. 213 (2006) 500–529.
[10] S. Karni, Multicomponent flow calculations by a consistent primitive algorithm, J. Comput. Phys. 112 (1994) 31–43.
[11] S. Karni, Hybrid multifluid algorithms, SIAM J. Sci. Comput. 17 (1) (1996) 1019–1039.
[12] R. Abgrall, How to prevent oscillation in multicomponent flow calculation, J. Comput. Phys. 125 (1996) 150–160.
[13] R. Saurel, R. Abgrall, A simple method for compressible multifluid flows, SIAM J. Sci. Comput. 21 (3) (1999) 1115–1145.
[14] X.Y. Hu, B.C. Khoo, An interface interaction method for compressible multifluids, J. Comput. Phys. 198 (2004) 35–64.
[15] T.-G. Liu, B.-C. Khoo, C.-W. Wang, The ghost fluid method for compressible gas–water simulation, J. Comput. Phys. 204 (2005) 193–221.
[16] E. Shima, T. Jounouchi, Role of CFD in aeronautical engineering (No. 14) – AUSM type upwind schemes, in: Proc. of the 14th NAL Symposium on Aircraft Computational Aerodynamics SP34, 1997, pp. 7–12.
[17] M.-S. Liou, C.J. Steffen, A new flux splitting scheme, J. Comput. Phys. 107 (1993) 23–39.
[18] M.-S. Liou, A Sequel to AUSM, Part II: AUSM(+)-up for all speeds, J. Comput. Phys. 214 (2006) 137–170.
[19] B. van Leer, Towards the ultimate conservation difference scheme. IV. A new approach to numerical convection, J. Comput. Phys. 23 (3) (1977) 276–299.
[20] B. van Leer, Towards the ultimate conservation difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 31 (1) (1979) 101–136.
[21] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (221) (1998) 73–85.
[22] S.O. Unverdi, G. Tryggvason, A front tracking method for viscous incompressible flows, J. Comput. Phys. 100 (1992) 25–37.
[23] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[24] 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.
[25] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithm based on Hamilton–Jacobi formulation, J. Comput. Phys. 79 (1988) 12–49.
[26] Y. Hao, A. Prosperetti, A numerical method for three-dimensional gas–liquid flow computations, J. Comput. Phys. 196 (2004) 126–144.
[27] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Nat. Acad. Sci. 93 (1996) 1591–1595.
[28] J.F. Hass, B. Sturtevant, Interactions of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech. 181 (1987) 41–76.
[29] J.M. Picone, J.P. Boris, Vorticity generation by shock propagation through bubbles in a gas, J. Fluid Mech. 189 (1988) 23–51.
[30] J.J. Quirk, S. Karni, On the dynamics of a shock–bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[31] E. Johnsen, T. Colonius, Compressible multicomponent flow calculations and shock–bubble interaction, in: Proc. of Sixth International Symposium on Cavitation, 2006, 11 p.
[32] C.-H. Chang, M.-S. Liou, A robust and accurate approach to computing compressible multiphase flow: stratified flow model and AUSM$^+$-up scheme, J. Comput. Phys. 225 (2007) 840–873.
[33] A. Bagabir, D. Drikakis, Mach number effects on shock–bubble interaction, Shock Waves 11 (2001) 209–218.
[34] G.-S. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (6) (2000) 2126–2143.
[35] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-based fast local level set method, J. Comput. Phys. 155 (1999) 410–438.
[36] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[37] D. Enright, R. Fedkiw, J. Ferziger, I. Mitchell, A hybrid particle level set method for improved interface capturing, J. Comput. Phys. 183 (2002) 83–116.

[38] V. Mihalef, D. Metaxas, M. Sussman, Textured liquids based on the marker level set, in: D. Cohen-Orr, P. Slavik (Eds.), Eurograhics, The Eurographics Association and Blackwell Publishing, Oxford, 2007.
[39] R.R. Nourgaliev, T.G. Theofanous, High-fidelity interface tracking in compressible flows: unlimited anchored adaptive level set, J. Comput. Phys. 224 (2007) 836–866.
[40] M. Brouillette, The Richtmyer–Meshkov instability, Annu. Rev. Fluid Mech. 34 (2002) 445–468.
[41] K.W. Thompson, Time dependent boundary conditions for hyperbolic systems, J. Comput. Phys. 68 (1987) 1–24.
[42] G. Tryggvason, S.O. Unverdi, Computations of three-dimensional Rayleigh–Taylor instability, Phys. Fluids 2 (1990) 656–659.
[43] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269–282.
[44] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Methods Fluids 30 (1999) 775–793.
[45] S. Chandrasekhar, Hydrodynamics and Hydromagnetic Stability, Oxford University Press, Oxford, 1961.
[46] A. Smolianski, Finite-Element/Level-Set/Operator-Splitting (FELSOS) approach for computing two-fluid unsteady flows with free moving interfaces, Int. J. Numer. Meth. Fluids 48 (3) (2005) 231–269.
[47] G.J. Ball, B.P. Howell, T.G. Leighton, M.J. Schofield, Shock-induced collapse of a cylinder air cavity in water: a free-Lagrange simulation, Shock Waves 10 (2000) 265–276.
[48] D. Igra, K. Takayama, Investigation of aerodynamic breakup of a cylindrical water droplet, Reports of the Institute of Fluid Science, Tohoku University, vol. 11 (3), 2001, pp. 123–134.
[49] H. Chen, Two-dimensional simulation of stripping breakup of a water droplet, AIAA J. 45 (5) (2008) 1135–1143.
[50] B. Larouturou, How to preserve the mass fraction positive when computing compressible multi-component flows, J. Comput. Phys. 95 (1991) 59–84.
[51] R. Caiden, R.P. Fedkiw, C. Anderson, A numerical method for two-phase flow consisting of separate compressible and incompressible regions, J. Comput. Phys. 166 (2001) 1–27.