Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 6385–6409
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An adaptive ghost fluid finite volume method for compressible gas–water simulations

Chunwu Wang [a], Huazhong Tang [b,*], Tiegang Liu [c]

[a] College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, PR China
[b] LMAM and CAPT, School of Mathematical Sciences, Peking University, Beijing 100871, PR China
[c] Department of Mathematics, Beijing University of Aeronautics and Astronautics, Beijing 100083, PR China

Received 27 February 2007; received in revised form 28 February 2008; accepted 10 March 2008
Available online 16 March 2008

Abstract

An adaptive ghost fluid finite volume method is developed for one- and two-dimensional compressible multi-medium flows in this work. It couples the real ghost fluid method (GFM) [C.W. Wang, T.G. Liu, B.C. Khoo, A real-ghost fluid method for the simulation of multi-medium compressible flow, SIAM J. Sci. Comput. 28 (2006) 278–302] and the adaptive moving mesh method [H.Z. Tang, T. Tang. Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515; H.Z. Tang, T. Tang, P.W. Zhang, An adaptive mesh redistribution method for non-linear Hamilton–Jacobi equations in two- and three-dimensions, J. Comput. Phys. 188 (2003) 543–572], and thus combines their advantages. This work shows that the local mesh clustering in the vicinity of the material interface can effectively reduce both numerical and conservative errors caused by the GFM around the material interface and other discontinuities. Besides the improvement of flow field resolution, the adaptive GFM also largely increases the computational efficiency. Several numerical experiments are conducted to demonstrate robustness and efficiency of the current method. They include several 1D and 2D gas–water flow problems, involving a large density gradient at the material interface and strong shock-interface interactions. The results show that our algorithm can capture the shock waves and the material interface accurately, and is stable and robust even for solutions with large density and pressure gradients.

Keywords: Finite volume method; Ghost fluid method; Moving mesh method; Level-set equation; Approximate Riemann solver; Gas–liquid Riemann problem

1. Introduction

The hydrodynamics of multi-medium and multi-phase compressible fluids such as gas and water is of great interest in a wide range of physical flows. Such multi-medium fluid flows give rise to challenging problems in both theory and numerical simulation. In general, conservative Eulerian algorithms such as those based on

* Corresponding author. Tel.: +86 10 62757018; fax: +86 10 62751801.
E-mail addresses: wangcw@nuaa.edu.cn (C. Wang), hztang@math.pku.edu.cn (H. Tang), liutg@buaa.edu.cn (T. Liu).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.03.005

high-resolution conservative schemes like TVD (total variation diminishing) or ENO (essentially non-oscillatory) schemes [46–48] perform very well when applied to single-medium compressible flows. However, when such an algorithm is employed to solve multi-medium or multi-phase compressible flows, numerical inaccuracies usually occur at the material interfaces due to "the loss of pressure-invariance property in discretization" [25,53]. To overcome this difficulty, many techniques and methods have been developed in the past two decades with an even increasing interest, see [1,2,23–25,27,28,33,20,40] for more information. Among them, some of the resultant algorithms [1,2,24,25,28] may not maintain the conservation property in the process of enforcing states at the moving material interface so as to suppress any undesired numerical oscillations.

The ghost fluid method (GFM) developed by Fedkiw et al. [18,19] has provided another flexible way to treat the multi-medium flows. The main appealing features of the GFM are its simplicity, easy extension to multi-dimensions and maintenance of a sharp-interface without smearing. The GFM makes the interface "invisible" during computations and the computations are carried out as for a single-medium manner such that its extension to multi-dimensions becomes fairly straightforward. Since only single-fluid flux formulations are required to make a GFM workable, the GFM is easily employed for capturing the boundary conditions of the Poisson equation [36] and two media of different coordinate system such as Eulerian–Lagrangian coupling [17,6] and for two fluids of vastly different types such as a compressible–incompressible [17] or viscous-inviscid two-fluid flow [9]. Recently, efforts have also been made to develop a conservative GFM as done in [39,20]. However, a practical conservative GFM has yet to be developed as far as we are aware.

On the other hand, when there is a strong shock wave interacting with the material interface, it is precisely the manner of treatment of the single-medium across the fluid interface in the GFM that may cause numerical inaccuracy [35]; this is especially so if such wave interaction with the fluid interface is not taken into account properly in the definition of the ghost fluid states. This situation arises because the pattern of shock refraction at the interface and the resultant interfacial states depend highly on material properties on both sides of the interface. As such, reasonable ghost fluid states have to be formulated to take into account the influence of both material properties and wave interaction with the interface. This has led to the development of a modified GFM (MGFM) with a predicted ghost fluid state by Liu et al. [35] via implicitly solving two non-linear characteristic equations interacting and applicable at the interface [33,34]. Defining and solving a Riemann problem is also the key part in the front track method [12]. In fact, it has been found that conditions have to be satisfied for the ghost fluid states in order that the two GFM Riemann problems provide the correct solution in the respective real fluids [31]. The MGFM has been shown to be robust and efficient when applied to gas–gas or gas–liquid compressible flows [35,31]. Those techniques in [33,34] have also been successfully used to the compressible multi-medium RKDG algorithm developed very recently in [43] and flow-structure coupling [32].

On applying the original GFM [18] to a shock impedance matching (-like) problem [31,35], it was found that special difficulties are encountered due to the inability of accurately imposing impedance matching conditions at the interface. As a result, non-physical reflection always occurs at the interface, where there should be no occurrence of reflection physically. Such non-physical reflection can be suppressed quite a bit but cannot be completely removed by the MGFM. To overcome these difficulties, Wang et al. [55] proposed to modify the fluid states at the real fluid points just next to the interface to better impose the interface boundary conditions. The resultant method was called real GFM. Besides keeping all the good properties of the MGFM, the real GFM can further suppress the non-physical reflection for the shock impedance matching (-like) problems. It was also shown [55] that the conservative errors caused by the real GFM is the smallest among all the existing GFMs.

Strictly, both the MGFM and the real GFM are non-conservative although the conservative errors caused by them have been shown to be well-limited, especially, for the latter. It was observed that the major conservative errors are caused in the very earlier stage (the very earlier 5–10 computational steps) of the numerical decomposition of the interface singularity [35,55] and are limited locally only in the vicinity of the interface. Because of these, the conservative errors may further be suppressed using mesh adaptivity. One of the motivations of this work is, thus, to bring down the conservative errors further by locally clustering mesh points.

The adaptive moving mesh methods have been proved to be a very effective way of reducing numerical errors and increasing the resolution of flow field for complex fluid dynamics, where singular or nearly singular solutions can develop dynamically in fairly localized regions of shock waves, boundary layers, and detonation waves etc. Numerically investigating these phenomena requires extremely fine meshes over a small portion of

the physical domain to resolve the large solution variations. Successful implementation of an adaptive strategy can increase accuracy of the numerical approximations and decrease the computational cost. Up to now, there have been many important progresses in adaptive moving mesh methods for partial differential equations, including the mesh-redistribution approach based on the variational principle of Winslow [57], Brackbill [8], Ren and Wang [44]; moving finite element methods of Millers [38], and Davis and Flaherty [13]; moving mesh PDEs methods of Russell et al. [10,21,22], and Ceniceros and Hou [11]; and moving mesh methods based on the harmonic mapping of Dvinsky [16], and Li et al. [29,30,14,15]. Computational costs of moving mesh methods can be efficiently saved with locally varying time steps [49].

Although the adaptive moving mesh methods have shown their effectiveness, there is still less work done on their extension and applications to multi-medium compressible flow, especially, with a large density ratio and strong shock wave interaction at the material interface. With the recent development and advancement of the ghost fluid method, it is possible to apply them to such complex flow conditions. The main objective of this paper is, therefore, to develop an adaptive ghost fluid finite volume method for two-medium compressible flows, where the adaptive moving mesh method developed in [51,52] as well as [50] is coupled to the real GFM recently proposed in [55]. To implement this method, two key techniques will be developed in this work. One is to accurately interpolate the solution in the vicinity of the interface during the mesh iterative redistribution. The other is to prevent the interface shift from one-medium to another during the mesh-redistribution. One will find that the local mesh clustering, due to mesh adaptivity in the vicinity of the material interface, can further and effectively reduce possible errors produced by the real GFM around the material interface. Furthermore, the proposed method is stable and robust even for large density and pressure gradients. It should be noted that the techniques developed in this work can also be applied to implement other GFMs coupled with the adaptive moving mesh.

Another widely used technique to improve the solution resolution is the adaptive mesh refinement (AMR) technique, in which the mesh is locally refined or coarsened. Such technique has successfully been coupled to the GFM or similar methods in [40,5,37]. The adaptive mesh technique used in this work is implemented via redistributing or moving the mesh points to a small portion of the physical domain in order to resolve the large solution variations, but does not change the total number of the mesh points.

The paper is organized as follows. In Section 2, the governing equations for flow, the EOS for gases and water, and the level set equation for the interface are provided. To avoid or prevent the appearance of hugely distorted level contours caused by the (potentially) large flow velocity gradient, an extension of the velocity field is constructed for solving the level set equation. The adaptive GFM for the governing system is developed in Section 3, in which the real GFM [55] is first employed to define the interface boundary conditions in the PDE evolution and the solution interpolation, and then the computations are carried out in the respective single-medium by using a high-resolution MUSCL-type finite volume scheme. Full solution procedure will be outlined in Section 4. Numerical experiments are carried out in Section 5 to validate the robustness and efficiency of the proposed adaptive approach. Finally, we conclude this work in Section 6.

2. Governing equations

We shall limit our study in two-dimensional cases. We use Ω_p to denote a two-dimensional (physical) domain with the Cartesian coordinate system $\boldsymbol{x} = (x, y)$, filled with two compressible fluids separated by a free moving interface. One fluid is air while the other is another gas or water. The two-dimensional system for a compressible fluid can be written as follows:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = 0, \tag{1}$$

where

$$\begin{aligned}
&\boldsymbol{U} = [\rho, \rho u, \rho v, E],\\
&\boldsymbol{F}(\boldsymbol{U}) = [\rho u, \rho u^2 + p, \rho uv, (E+p)u]^{\mathrm{T}},\\
&\boldsymbol{G}(\boldsymbol{U}) = [\rho v, \rho uv, \rho v^2 + p, (E+p)v]^{\mathrm{T}}.
\end{aligned}$$

Here ρ is the density, u and v are the velocity components in the respective x and y directions, p is the pressure and E is the total energy per unit volume. The total energy is the sum of internal energy and kinetic energy

$$E = \rho e + \frac{1}{2}\rho(u^2 + v^2), \tag{2}$$

where e is the internal energy per unit mass.

For closure of the system (1), the equation of state (EOS) is required. In the present work, our interest is centered on the compressible gas and water media. The EOS for gases or water medium can be written uniformly as

$$p = (\gamma - 1)\rho e - \gamma B \tag{3}$$

where γ and B are treated as fluid constants, and will be specified in Section 5.

2.1. Level set equation

To track the moving fluid interface, we employ the level set technique [45,42,3]. The level set equation can be written as

$$\frac{\partial \phi}{\partial t} + \boldsymbol{u} \cdot \nabla \phi = \frac{\partial \phi}{\partial t} + \left(\boldsymbol{u} \cdot \frac{\nabla \phi}{|\nabla \phi|} \right) |\nabla \phi| = 0. \tag{4}$$

In general $\phi(\boldsymbol{x}, t)$ starts off as a signed distance function. For complex interface evolution, it has been found that a higher-order accurate discretization can better help to maintain the accurate position of the interface [18]. In this work, the high-order accurate Hamilton–Jacobi solvers [52] will be used to track the interface together with the third-order Runge–Kutta time discretization [46].

As discussed in [55], the possible disadvantage of using the level set technique is that solving Eq. (4) may lead to severely distorted contours when the large gradient appears in the velocity field near the interface. To remedy this difficulty, a redefinition of the velocity field for (4) is called forth via the extension velocity [4]. There are several ways of constructing the extension velocity in literatures. In this work, the normal velocity at the interface, which is obtained from the Riemann problem solver, is extended to all the nodes in the fluid field as the extension velocity. By solving Eq. (4) with the extension velocity, the uniformly distributed level set contour and accurate interface position can be obtained as the Riemann problem solver provides accurate interface moving velocity. In particular, the signed distance function is maintained using the present extension velocity technique for the one-dimensional problems.

3. Numerical scheme

In this section, we shall develop our adaptive ghost fluid finite volume method, which couples the adaptive moving mesh method developed in [51,52] and the real GFM proposed in [55]. The basic idea of the adaptive GFM can be summarized as follows:

(1) Suppose a physical mesh is given on which the approximations to the solutions $\boldsymbol{U}$ and the level set function ϕ over cells and on cell vertexes are obtained, respectively.
(2) Redistribute the mesh by iterating an elliptic grid generator and simultaneously remap the approximate solutions $\boldsymbol{U}$ onto the new mesh by using a high-resolution conservative formula together with real GFM. And the level set function ϕ is redefined as the distance between the mesh point and the interface in one dimension, but is updated by using a non-conservative second-order interpolation formula in two dimensions.
(3) Advance $\boldsymbol{U}$ and ϕ with a physical time step size Δt by solving the governing equations as well as the level set equation with the extension velocity.

In the above method, two simple but key techniques will be developed. One is to accurately interpolate the solution in the vicinity of the interface during the iterative mesh-redistribution. The other is to prevent the interface shift from one-medium to another during the mesh-redistribution.

3.1. PDE solver

In this subsection, we shall first describe Step 3 mentioned above, namely the numerical discretization for (1) and (4) in the physical domain. The mesh motion and solution interpolation parts will be provided in the next subsection. The system (1) is solved by using a finite volume version of the real GFM [55], and the level set Eq. (4) is approximated by using the high-resolution Hamilton–Jacobi solvers [52]. In the real GFM, a local multi-medium Riemann problem across the fluid interface is defined and solved to predict the interfacial states over the real and ghost fluid cells next to the fluid interface, and then the system (1) is independently solved by using the high-resolution MUSCL-type shock-capturing method for each single-medium.

3.1.1. 1D case

Assume that the physical domain $\Omega_p = [a, b]$ is covered by a (non-uniform) mesh $\{x_i \mid 0 \leqslant i \leqslant N+1\}$, and the fluid interface x_I lies between $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$ as depicted in Fig. 1, where $x_{i+\frac{1}{2}}$ denotes the centroid coordinates of the cell $A_{i+\frac{1}{2}} = [x_i, x_{i+1}]$. For medium 1, $A_{i-\frac{1}{2}}$ is named the real fluid cell, $A_{i+\frac{1}{2}}$ and $A_{i+\frac{3}{2}}$ are two ghost fluid cells, whereas $A_{i+\frac{1}{2}}$, $A_{i-\frac{1}{2}}$ and $A_{i-\frac{3}{2}}$ are a real fluid cell and two ghost fluid cells for medium 2, respectively. The fluid variables $\boldsymbol{U}$ are all approximated over the cell by their cell average values

$$\boldsymbol{U}_{i+\frac{1}{2}}(t) = \frac{1}{\left|A_{i+\frac{1}{2}}\right|} \int_{A_{i+\frac{1}{2}}} \boldsymbol{U}(x,t)\,\mathrm{d}x, \tag{5}$$

and the level set function ϕ is approximated on the cell vertex x_i, where $|A_{i+\frac{1}{2}}|$ denotes the area of the cell $A_{i+\frac{1}{2}}$.

In 1D real GFM, the local Riemann problem across $x_I \in (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ is constructed with initial constant states: $\boldsymbol{U}_L = \boldsymbol{U}_{i-\frac{1}{2}}$ and $\boldsymbol{U}_R = \boldsymbol{U}_{i+\frac{1}{2}}$, and solved by the Riemann solver [35,33] to obtain the intermediate states: p_I (pressure), u_I (velocity), and ρ_{IL} and ρ_{IR} representing the densities on the left and right sides of the fluid interface, respectively. Then those intermediate states are taken as newly defined cell averages over the above-mentioned real and ghost fluid cells, such as $\{A_{i-\frac{1}{2}}, A_{i+\frac{1}{2}}, A_{i+\frac{3}{2}}\}$ formedium 1 and $\{A_{i+\frac{1}{2}}, A_{i-\frac{1}{2}}, A_{i-\frac{3}{2}}\}$ for medium 2, that is to say, for medium 1, we impose the interface boundary conditions as $\boldsymbol{U}^{BC}_{i-\frac{1}{2}} = (\rho_{IL}, \rho_{IL}u_I, E_{IL})^T$, $\boldsymbol{U}^{BC}_{i+\frac{1}{2}} = (\rho_{IL}, \rho_{IL}u_I, E_{IL})^T$, and $\boldsymbol{U}^{BC}_{i+\frac{3}{2}} = (\rho_{IL}, \rho_{IL}u_I, E_{IL})^T$.

After defining the ghost fluid states, we then solve (1) for the respective media including the range of ghost fluid cells by using the second-order MUSCL-type scheme as given in [51]. The level set Eq. (4) is replaced by

$$\phi_t + \tilde{u}\phi_x = 0, \quad x \in \Omega_p,$$

and then solved by using the high-resolution 1D Hamilton–Jacobi solver [52], where $\tilde{u} := u_I$ is the so-called extension velocity. The final solution $\boldsymbol{U}$ is obtained by combining the respective solution for each fluid according to the new interface location obtained, as it is done for a conventional GFM algorithm.

Remark 1. In 1D case, the extension velocity is constant at each time level. If $x_I = x_{i+\frac{1}{2}}$, we may approximately choose $\boldsymbol{U}_L = \boldsymbol{U}_{i+\frac{1}{2}}$ and $\boldsymbol{U}_R = \boldsymbol{U}_{i+\frac{3}{2}}$ as the initial data of the local multi-medium Riemann problem.

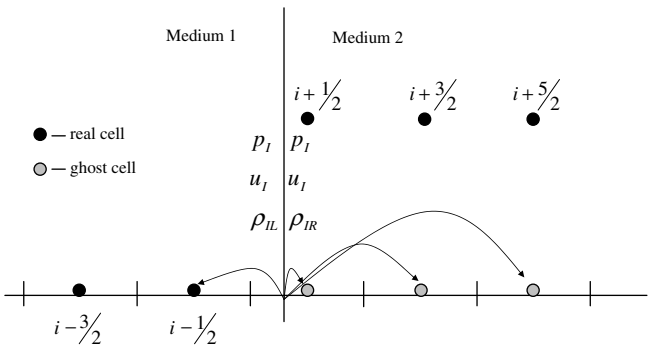


Fig. 1. Defining the real and ghost fluid states in the 1D real GFM. $\rho_{IR}, \rho_{IL}, p_I$, and u_I are the intermediate states of the local Riemann problem across the fluid interface x_I.

3.1.2. 2D case

Give a partition of the physical domain Ω_p, $\{A_{i+\frac{1}{2},j+\frac{1}{2}} | i,j \in \mathbb{Z}\}$, where $A_{i+\frac{1}{2},j+\frac{1}{2}}$ is a general quadrangle with four corners $\boldsymbol{x}_{i,j}$, $\boldsymbol{x}_{i+1,j}$, $\boldsymbol{x}_{i+1,j+1}$, and $\boldsymbol{x}_{i,j+1}$. Assume that the fluid interface $\boldsymbol{x}_I$ is a known curve as depicted in Fig. 2, where the fluid at the left (or right) hand side of the interface denotes medium 1 (or medium 2), and the cells A (or B) are just bordering the fluid interface in medium 1 (or medium 2). Thus, for medium 1, the cell A is a real cell, and the cell B is considered as a ghost cell.

The fluid variables $\boldsymbol{U}$ are again approximated on cell centers by their cell average values

$$\boldsymbol{U}_{i+\frac{1}{2},j+\frac{1}{2}}(t) = \frac{1}{\left|A_{i+\frac{1}{2},j+\frac{1}{2}}\right|} \int_{A_{i+\frac{1}{2},j+\frac{1}{2}}} \boldsymbol{U}(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}, \tag{6}$$

and the level set function ϕ is approximated at the cell corner points $\boldsymbol{x}_{i,j}$, where $|A_{i+\frac{1}{2},j+\frac{1}{2}}|$ denotes the area of the cell $A_{i+\frac{1}{2},j+\frac{1}{2}}$.

Using the approximate level set function $\phi_{i,j}$, we define the unit "normal" vector over each cell as

$$\boldsymbol{n}_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{\nabla\phi_{i+\frac{1}{2},j+\frac{1}{2}}}{|\nabla\phi_{i+\frac{1}{2},j+\frac{1}{2}}|}, \tag{7}$$

and the normal velocity over each cell (next to the interface)

$$(u_n)_{i+\frac{1}{2},j+\frac{1}{2}} = \boldsymbol{u}_{i+\frac{1}{2},j+\frac{1}{2}} \cdot \boldsymbol{n}_{i+\frac{1}{2},j+\frac{1}{2}}, \tag{8}$$

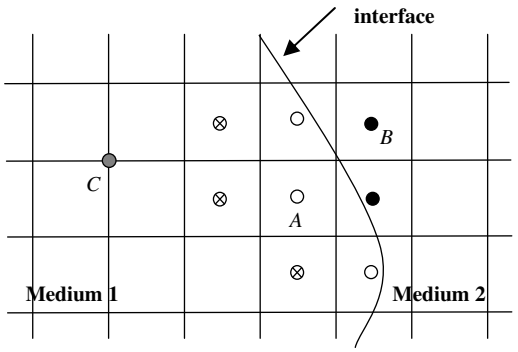
where

$$\phi_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(\phi_{i,j} + \phi_{i+1,j} + \phi_{i,j+1} + \phi_{i+1,j+1}).$$

We begin to construct the local Riemann problem across the fluid interface $\boldsymbol{x}_I$ to impose the interface boundary conditions. For the sake of clarity, all subscripts related to the mesh indexes such as i, j will be omitted in the following. For each cell A in *medium 1* (or B in medium 2) bordering the interface, we search for a corresponding cell B in medium 2 (or A in medium 1) locally such that the angle made by the respective normals $\boldsymbol{n}^A$ and $\boldsymbol{n}^B$ defined in (7) is the minimum. If the cell A and cell B satisfy the above need, we call them *a neighboring cell pair* with respect to the fluid interface. For each neighboring cell pair, a local 1D Riemann problem is constructed with initial data

$$\boldsymbol{V}(\boldsymbol{x}, t=0) = \begin{cases} \boldsymbol{V}^A, & \boldsymbol{x} \text{ is at the left hand side of } \boldsymbol{x}_I, \\ \boldsymbol{V}^B, & \boldsymbol{x} \text{ is at the right hand side of } \boldsymbol{x}_I, \end{cases} \tag{9}$$

and solved to provide corresponding intermediate states: ρ_{IL}, ρ_{IR}, $(u_n)_I$, and p_I, where $\boldsymbol{V}^A = [\rho^A, u_n^A, p^A]^T$, $\boldsymbol{V}^B = [\rho^B, u_n^B, p^B]^T$. As soon as the intermediate states are available, they are used to redefine the real fluid



⊗ — Cell far away the interface in medium 1, ○— Cell bordering the interface in medium 1,

● — Cellbordering the interface in medium 2.

Fig. 2. Updating the real and ghost nodes in the 2D real GFM.

states, i.e. over those real fluid cells (e.g. the cells A for medium 1) next to the fluid interface. The ghost fluid states are then defined by locally solving the convection equation [18]:

$$\boldsymbol{W}_t \pm \boldsymbol{n} \cdot \nabla \boldsymbol{W} = 0, \tag{10}$$

where $\boldsymbol{W}$ is a column vector consisting of the normal velocity, the tangential velocity, the pressure, and the density or entropy. This is done by using the high-resolution Hamilton–Jacobi solver [52] and fixing the values of $\boldsymbol{W}$ in the real fluid region where $\phi < 0$ for medium 1 ($\phi > 0$ for medium 2) until the steady state is reached for all ghost fluid cells.

Based on the above interface boundary conditions, the system (1) can be independently solved in medium 1 (or medium 2) by using the second-order MUSCL-type finite volume scheme as given in [51] in order to advance the fluid variable $\boldsymbol{U}$. The level set Eq. (4) is replaced by

$$\phi_t + \tilde{u}_n^C |\nabla \phi| = 0,$$

at each mesh point C as depicted in Fig. 2, and then solved by using the high-resolution Hamilton–Jacobi solver [52], where $\tilde{u}_n^C$ denotes the extension velocity defined at the mesh point C. For the mesh points C bordering the fluid interface, the extension velocity is directly defined by the intermediate velocity states of the Riemann solution, see [4]. For the mesh points away from the fluid interface, we fix the extension velocity on those mesh points next to the fluid interface and then solve (10) with a reverse convection direction to get $\tilde{u}_n^C$.

Remark 2. In 2D case, the fluid interface may be seriously winding so that the redefinition of the real and ghost fluid states becomes more difficult than that in 1D case. In the current implementation, a local multi-medium Riemann problem is appropriately constructed and solved to redefine the real fluid states next to the fluid interface, and then they are extrapolated to the associated ghost cells accordingly. The present technique is very simple and works efficiently as the interface status is predicted correctly. Generally, the computed level set function may not be exactly a signed distance from the interface using the extension velocity. However, we found that the level set contours keep quite uniform in practice. In our computations, the re-initialization is never used. The re-initialization is actually inevitable if the interface has large distortion, topological changes, or evolved for a long time. And it is fairly easy to do accurately for unstructured meshes, see e.g. [26].

3.2. The mesh-redistribution

In this subsection, we describe Step 2 mentioned above.

3.2.1. Mesh motion based on an iteration procedure

For simplicity, we mainly illustrate the idea of the mesh motion for 1D case. The extension to higher dimensions is straightforward, see [51]. Let x and ξ be the physical and logical or computational coordinates, respectively, which are (without loss of generality) assumed to be in $\Omega_p = [a, b]$ and $\Omega_l = [0, 1]$. We use the conventional 1D equidistribution principle

$$\omega(\boldsymbol{U}) \frac{\partial x}{\partial \xi} = \text{Const.}$$

or equivalently

$$\frac{\partial}{\partial \xi} \left\{ \omega(\boldsymbol{U}) \frac{\partial x}{\partial \xi} \right\} = 0 \tag{11}$$

to provide the coordinate transformation $x = x(\xi)$, where ω is the monitor function, which in general depends on the underlying solutions $\boldsymbol{U}$ and ϕ to be adapted.

Assume a (fixed) uniform mesh on the logical domain Ω_l is given by $\xi_i = i/(N-1)$, $i = 0, 1, 2, \ldots, N+1$. Eq. (11) can be solved by a Jacobian type iteration

$$\begin{cases} \omega\left(\boldsymbol{U}_{i+\frac{1}{2}}^{[v]}\right)\left(x_{i+1}^{[v]} - x_i^{[v+1]}\right) - \omega\left(\boldsymbol{U}_{i-\frac{1}{2}}^{[v]}\right)\left(x_i^{[v+1]} - x_{i-1}^{[v]}\right) = 0, \\ x_0 = a, \quad x_{N+1} = b, \quad i = 1, 2, \ldots, N. \end{cases} \tag{12}$$

Since the sign of the level set function is used to identify where a fluid is located, it is crucial to prevent the interface from shifting during the mesh-redistribution. This is done by preventing the mesh across the interface. Once one loop of the above iteration is finished, we check whether the mesh point in medium 1 as depicted in Fig. 1 has moved into the region of medium 2 or vice versa, before updating the solutions $\boldsymbol{U}$ and ϕ by a suitable way to be provided in subsection 3.2.2. For example, assume that $x_{\mathrm{I}} \in [x_{i-\frac{1}{2}}^{[v]}, x_{i+\frac{1}{2}}^{[v]})$ but $x_{i-1}^{[v+1]} < x_{\mathrm{I}} \leqslant x_{i-\frac{1}{2}}^{[v+1]}$ after one loop of the above mesh iteration, we set $x_i^{[v+1]}$ equal to x_{I} so that the final mesh $\left\{x_i^{[v+1]}\right\}$ will not move across the fluid interface. An alternative is to set $x_{i-l}^{[v+1]}$ equal to $x_{i-l}^{[v]}$, $l = 0, 1$. The final mesh $\left\{x_i^{[v+1]}\right\}$ also keeps the monotonic order of $\left\{x_i^{[v]}\right\}$ if the mesh $\left\{x_i^{[v+1]}\right\}$ obtained by (12) is order-preserving.

When the updated approximations $\boldsymbol{U}_{i+\frac{1}{2}}^{[v+1]}$ and $\phi_i^{[v+1]}$ are obtained, the iteration (12) can be employed again to improve the equality of the mesh. This leads to an iteration procedure on the mesh motion and solution interpolation. The iteration determines progressively better values of the new mesh locations and the approximate solutions. The total iteration is continued until there is no significant change in the calculated new meshes from one iteration to the next. Typically about 3–5 cycles of the mesh iteration are required. In most computations in this work we use 5 iterations at each time level.

In the practical computations, it is common to use some temporal or spatial smoothing on the monitor function ω to obtain smoother meshes. This work applies the following low pass filter to smooth the monitor

$$\omega_{i+\frac{1}{2}} \leftarrow \frac{1}{4}\left(\omega_{i+\frac{3}{2}} + 2\omega_{i+\frac{1}{2}} + \omega_{i-\frac{1}{2}}\right).$$

A 2D version of the above iteration formula is given by

$$\omega_{i+\frac{1}{2},j}^{[v]}\left(\boldsymbol{x}_{i+1,j}^{[v]} - \boldsymbol{x}_{i,j}^{[v+1]}\right) - \omega_{i-\frac{1}{2},j}^{[v]}\left(\boldsymbol{x}_{i,j}^{[v+1]} - \boldsymbol{x}_{i-1,j}^{[v]}\right) + \omega_{i,j+\frac{1}{2}}^{[v]}\left(\boldsymbol{x}_{i,j+1}^{[v]} - \boldsymbol{x}_{i,j}^{[v+1]}\right) - \omega_{i,j-\frac{1}{2}}^{[v]}\left(\boldsymbol{x}_{i,j}^{[v+1]} - \boldsymbol{x}_{i,j-1}^{[v]}\right) = 0, \tag{13}$$

where $\omega_{i+\frac{1}{2},j} = \frac{1}{2}(\omega_{i+1,j} + \omega_{i,j}), \omega_{i,j+\frac{1}{2}} = \frac{1}{2}(\omega_{i,j+1} + \omega_{i,j})$, and $\boldsymbol{x} = (x, y)$ denotes the physical coordinates. Similarly, after one loop of the above mesh iteration, we should avoid that any mesh point in medium 1 as depicted in Fig. 2 moves into the region of medium 2 or vice versa. The above 1D technique can be extended to two dimensions in order to prevent the 2D mesh across the fluid interface. For example, assume that $\boldsymbol{x}_{i+\frac{1}{2},j+\frac{1}{2}}^{[v]}$ is at the left hand side of the fluid interface $\boldsymbol{x}_{\mathrm{I}}$. If after one loop of the above mesh iteration (13) $\boldsymbol{x}_{i+\frac{1}{2},j+\frac{1}{2}}^{[v+1]}$ is at the right hand side of the fluid interface $\boldsymbol{x}_{\mathrm{I}}$, we then set $\boldsymbol{x}_{i+l,j+p}^{[v+1]}$ equal to $\boldsymbol{x}_{i+l,j+p}^{[v]}$, $l, p = 0, 1$.

Remark 3. The aim of avoiding the mesh across the interface is to simplify the interpolation of the fluid variables. It is worth noting that the fluid interface does not move physically in the iterative mesh-redistribution.

3.2.2. The solution interpolation

After each mesh iterative step, we need to update the approximate solutions $\boldsymbol{U}$ and ϕ on the new mesh $\{\boldsymbol{x}_{i,j}^{[v+1]}\}$ from the old mesh $\{\boldsymbol{x}_{i,j}^{[v]}\}$.

In one dimension, the level set function can be easily obtained in each step of mesh-redistribution/iteration due to the known interface location. In two dimensions, to construct the level set distance function is more complex over non-uniform mesh although the interface location at the new time step is known by advancing the level set equation over the old mesh in advance. To overcome this difficulty, the 2D level set function is updated by using the second-order accurate non-conservative interpolation introduced in [52]:

$$\begin{aligned}\phi_{i,j}^{[v+1]} = \phi_{i,j}^{[v]} &- \frac{1}{2}\left(c_{i,j}^{\xi}\left(v_{i+0,j}^{[v]} + v_{i-0,j}^{[v]}\right) - \left|c_{i,j}^{\xi} r\right|\left(v_{i+0,j}^{[v]} - v_{i-0,j}^{[v]}\right)\right) \\ &- \frac{1}{2}\left(c_{i,j}^{\eta}\left(w_{i,j+0}^{[v]} + w_{i,j-0}^{[v]}\right) - \left|c_{i,j}^{\eta}\right|\left(w_{i,j+0}^{[v]} - w_{i,j-0}^{[v]}\right)\right),\end{aligned} \tag{14}$$

where

$$(c^{\xi})_{i,j} = \frac{1}{J_{i,j}}\left[x_{\eta}\left(y^{[v]} - y^{[v+1]}\right) - y_{\eta}\left(x^{[v]} - x^{[v+1]}\right)\right]_{i,j}, \quad (c^{\eta})_{i,j} = \frac{1}{J_{i,j}}\left[y_{\xi}\left(x^{[v]} - x^{[v+1]}\right) - x_{\xi}\left(y^{[v]} - y^{[v+1]}\right)\right]_{i,j},$$

and

$$v_{i+0,j} = \Delta_\xi \phi_{i,j} - \frac{1}{2}\mathrm{vLL}(\Delta_\xi \phi_{i+1,j} - \Delta_\xi \phi_{i,j}, \Delta_\xi \phi_{i,j} - \Delta_\xi \phi_{i-1,j}),$$

$$v_{i-0,j} = \Delta_\xi \phi_{i-1,j} + \frac{1}{2}\mathrm{vLL}(\Delta_\xi \phi_{i,j} - \Delta_\xi \phi_{i-1,j}, \Delta_\xi \phi_{i-1,j} - \Delta_\xi \phi_{i-2,j}),$$

$$w_{i,j+0} = \Delta_\eta \phi_{i,j} - \frac{1}{2}\mathrm{vLL}(\Delta_\eta \phi_{i,j+1} - \Delta_\eta \phi_{i,j}, \Delta_\eta \phi_{i,j} - \Delta_\eta \phi_{i,j-1}),$$

$$w_{i,j-0} = \Delta_\eta \phi_{i,j-1} + \frac{1}{2}\mathrm{vLL}(\Delta_\eta \phi_{i,j} - \Delta_\eta \phi_{i,j-1}, \Delta_\eta \phi_{i,j-1} - \Delta_\eta \phi_{i,j-2}).$$

Here we have assumed $\Delta\xi = \Delta\eta = 1$ and defined $\Delta_\xi \phi_{i,j} = \phi_{i+1,j} - \phi_{i,j}$, $\Delta_\eta \phi_{i,j} = \phi_{i,j+1} - \phi_{i,j}$, and $J_{i,j} = (x_\xi y_\eta - x_\eta y_\xi)_{i,j}$. The function $\mathrm{vLL}(a,b)$ denotes van Leer's limiter [54], defined by

$$\mathrm{vLL}(a,b) = (\mathrm{sign}(a) + \mathrm{sign}(b))\frac{|ab|}{|a| + |b| + \varepsilon}, \tag{15}$$

where the parameter ε, $0 < \varepsilon \ll 1$, is used to avoid that the denominator becomes zero.

In theory, the interpolated interface $\{\phi^{[v+1]} = 0\}$ should be identical to $\{\phi^{[v]} = 0\}$. In 1D cases, it can be exactly guaranteed, due to redefinition of ϕ as the signed distance. In 2D cases, although the interpolation used in this work does not guarantee the above property, its error is limited to second-order. In practical computations, we may locally correct the interpolated level set function as follows: Assume that $x_{i,j}^{[v]}$ is in the region of medium 1 where $\phi < 0$. If the interpolated level set function $\phi_{i,j}^{[v+1]} > 0$, then we replace its value with $\phi_{i,j}^{[v]}$.

For single-medium flow, the high-resolution conservative interpolation formulas provided in [51] can be used to update the approximate solutions $\boldsymbol{U}$. However, when they are directly applied to solve multi-medium or multi-phase compressible flows, numerical inaccuracies will occur at the material interfaces due to the mesh-redistribution. To overcome this difficulty, we first use the real GFM technique to specify the real and ghost fluid states next to the fluid interface, and then remap or interpolate the fluid variables $\boldsymbol{U}$ onto the new mesh $\{\boldsymbol{x}^{[v+1]}\}$ for the respective medium 1 and medium 2 on both sides of the fluid interface by using the high-resolution conservative interpolation formulas [51] including the ghost cell states.

Remark 4. The GFM used for solution interpolation is the same as used in the PDE solver described in Section 3. When the conservative variables in the vicinity of the interface are deployed by using the real GFM, the conservation errors are introduced so that the above interpolation of the fluid variables $\boldsymbol{U}$ is not fully conservative in the physical domain Ω_p. Even so, it is still very important because the locally conservative interpolation of the conservative variables [51] for each single-medium flow minimizes the conservation errors well.

4. Solution procedure

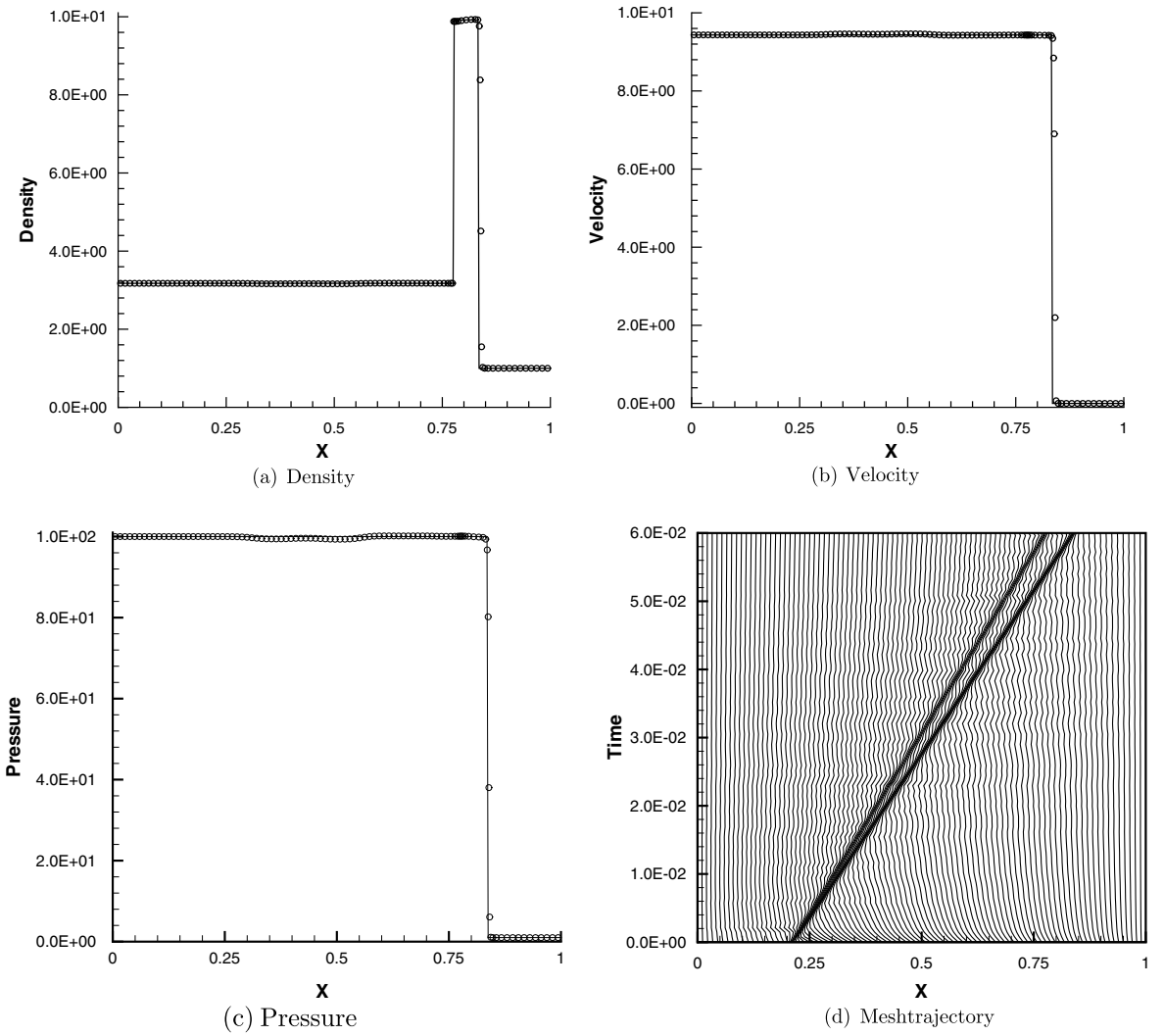
Our solution procedure is formed by two independent parts: evolution of the governing equations of fluid flows and the level set equation and an iterative mesh-redistribution. The first part is both a high-resolution MUSCL-type shock-capturing method for each single-medium flow and a high-resolution Hamilton–Jacobi solver on fixed but non-uniform meshes, together with the interface boundary conditions defined by using the real GFM, see Section 3. In each iteration of the second part, the mesh points are first redistributed by the relaxed iteration method (12) or (13), and then the conservative variables $\boldsymbol{U}$ are updated for each single-medium flow on the resulting new meshes by using the locally conservative interpolation formulas [51] and the interface boundary conditions defined by using the real GFM, whereas the level set function is remapped by using high-resolution Hamilton–Jacobi solvers [52]. The solution procedure can be illustrated by the following flowchart:

Step 1 Give an initial partition of the physical domain Ω_p and the logical domain Ω_l, denoted by $\{\boldsymbol{x}_{i,j}^0\}$ and $\{\boldsymbol{\xi}_{i,j}\}$ respectively. Compute the cell average of the conservative variables $\boldsymbol{U}$ denoted by $\boldsymbol{U}^0_{i+\frac{1}{2},j+\frac{1}{2}}$ and the node value of the level set function $\phi_{i,j}^0$.

Step 2 Set $\bar{\boldsymbol{x}}_{i,j}^{[0]} := \boldsymbol{x}_{i,j}^0$. For $v = 0, 1, 2, \ldots, \mu - 1$, do the following:

(1) Move mesh points $\bar{\boldsymbol{x}}^{[v]}$ to $\bar{\boldsymbol{x}}^{[v+1]}$.
(2) Redefine the initial value of $\boldsymbol{U}^0_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\phi^0_{i,j}$ on the new mesh $\{\bar{\boldsymbol{x}}^{[v+1]}_{i,j}\}$.
(3) Set $\boldsymbol{x}^0_{i,j} := \bar{\boldsymbol{x}}^{[v+1]}_{i,j}$

Step 3 For $n = 0, 1, \ldots$, advance the solutions $\boldsymbol{U}$ and ϕ in time by doing the following steps:
(1) Define the interface boundary conditions and ghost fluid states by using the real GFM.
(2) Evolve the level set equation with the extension velocity to obtain the numerical approximation $\phi^{n+1}_{i,j}$ at the time level $t = t_{n+1}$; and for each single-medium flow, evolve the governing equations of fluid flows by using the high-resolution finite volume shock-capturing method to obtain the numerical approximation $\boldsymbol{U}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$ at the time level $t = t_{n+1}$.
(3) Set $\boldsymbol{x}^{[0]}_{i,j} := \boldsymbol{x}^n_{i,j}$, $\phi^{[0]}_{i,j} := \phi^{n+1}_{i,j}$, $\boldsymbol{U}^{[0]}_{i+\frac{1}{2},j+\frac{1}{2}} := \boldsymbol{U}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}}$. For $v = 0, 1, 2, ..., \mu - 1$, iteratively redistribute the mesh by doing the following:
   (a) Move mesh points $\boldsymbol{x}^{[v]}$ to $\boldsymbol{x}^{[v+1]}$.



(a) Density (b) Velocity (c) Pressure (d) Meshtrajectory

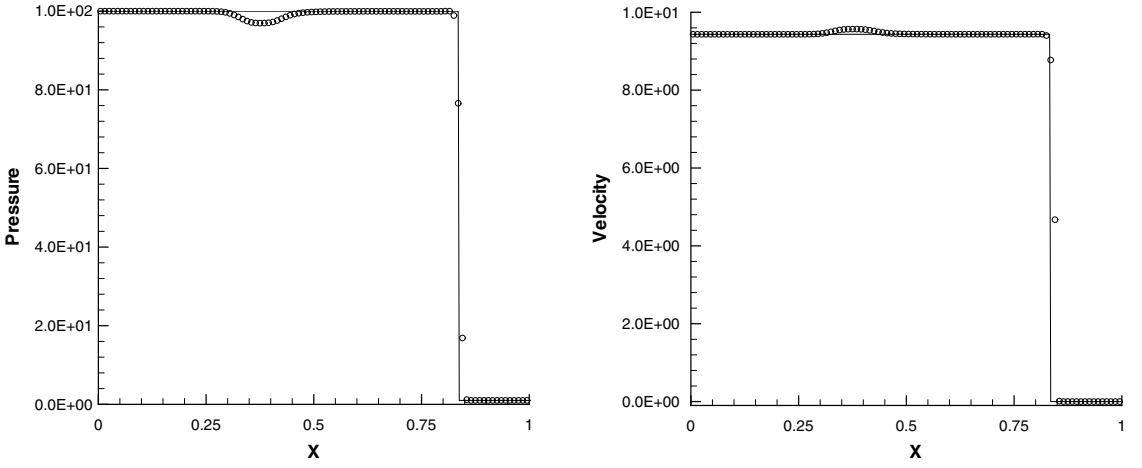
Fig. 3. Example 1: The numerical results ("○") at time $t = 0.06$ calculated by the proposed adaptive GFM method. The solid line denotes the exact solution.

(b) Define the interface boundary conditions by using the real GFM.
(c) For each single-medium flow, remap the conservative variables $\boldsymbol{U}^{[v+1]}_{i+\frac{1}{2},j+\frac{1}{2}}$ onto the new mesh $\{\boldsymbol{x}^{[v+1]}_{i,j}\}$.
(d) Interpolate or redefine the level set function $\phi^{[v+1]}_{i,j}$.

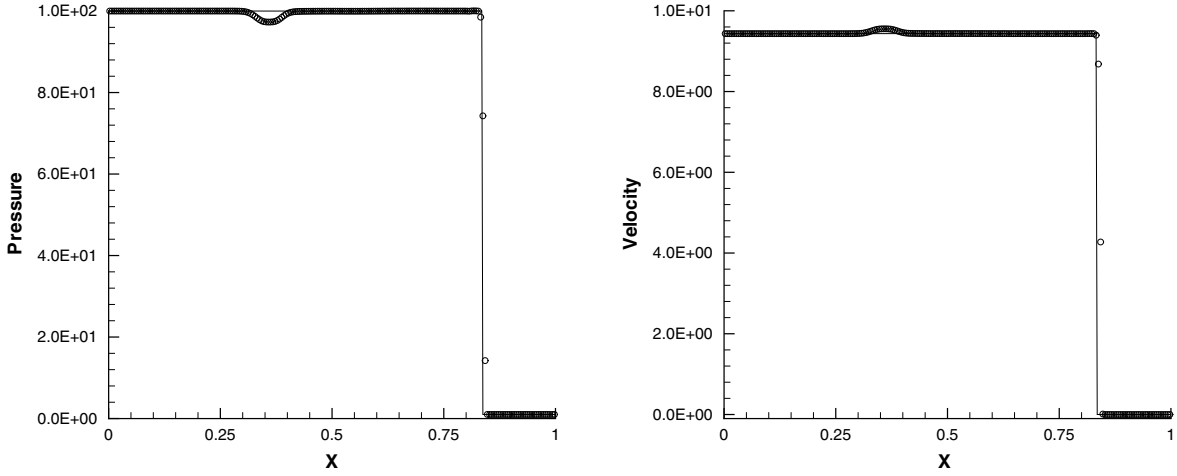
(4) Set $\boldsymbol{x}^{n+1}_{i,j} := \boldsymbol{x}^{[\mu]}_{i,j}$, $\phi^{n+1}_{i,j} := \phi^{[\mu]}_{i,j}$, $\boldsymbol{U}^{n+1}_{i+\frac{1}{2},j+\frac{1}{2}} := \boldsymbol{U}^{[\mu]}_{i+\frac{1}{2},j+\frac{1}{2}}$.

Step 4 If $t_{n+1} < T$, then go to **Step 3**; otherwise output the computed results and stop.

Remark 5. During the iterative mesh-redistribution, the interface boundary conditions for $v > 0$ may be simply taken as those for $v = 0$ so that the Riemann problems across the interface are only solved once at the beginning of the iteration, since they are costly.



(a) Pressure and velocity obtained by 100 cells



(b) Pressure and velocity obtained by 200 cells

Fig. 4. Example 1: The numerical results ("∘") at time $t = 0.06$ calculated by the real GFM method on uniform meshes. The solid line denotes the exact solution.

5. Numerical experiments

In this section, the adaptive real GFM method developed above is used to solve several one- and two-dimensional problems in order to validate the capability and efficiency in treating the moving fluid interface and its vicinity of two fluids and resolving strong shock wave etc.

5.1. One-dimensional examples

Four 1D examples will be considered in this subsection. All of them have been used by several authors to test various numerical schemes. Throughout our 1D computations, the physical domain Ω_p is taken as $[0,1]$ and divided into 100 cells. The CFL number is taken as 0.45. The approximate Riemann solver described in [33,31] is employed to supply the intermediate states. The purpose of using an approximate Riemann solver here is purely for the future application in real UNDEX (underwater explosion), where the EOS for water is usually a polynomial EOS [56]. The monitor function ω is taken as

$$\omega = \sqrt{1 + \alpha\left(\frac{p_\xi}{\max\{|p_\xi|\}}\right)^2 + \beta\left(\frac{\rho_\xi}{\max\{|\rho_\xi|\}}\right)^2}, \tag{16}$$

where α and β are two non-negative constants and taken as 60 and 80, respectively, unless otherwise stated.

Example 1. The first example is to simulate a gaseous shock in impedance-matching medium, in which a right-moving shock on the left side of the interface impacts on a gas–gas interface [35]. The shock strength is 100 and the initial location of the gaseous shock is the same as the interface location at $x_I = 0.2$. The states on the left and right sides of the interface are defined respectively as follows:

$$(\rho, u, p, \gamma, B) = \begin{cases} (0.82369077, 0, 1, 1.667, 0), & 0 \leqslant x < x_I, \\ (1, 0, 1, 1.2, 0), & x_I < x \leqslant 1. \end{cases}$$

Based on the analysis on the interaction of the shock reflection at the interface, the gaseous shock is transmitted to the right side of the interface and there is no reflection of any kind. However, the calculation with the original GFM [18], the MGFM [35], the new GFM [17] shows the obvious presence of non-physical reflection, due to the inability of accurately imposing the shock impedance matching (like) condition at the interface. The real GFM [55] can improve the calculation even compared to the MGFM. We refer the readers to [55] for a

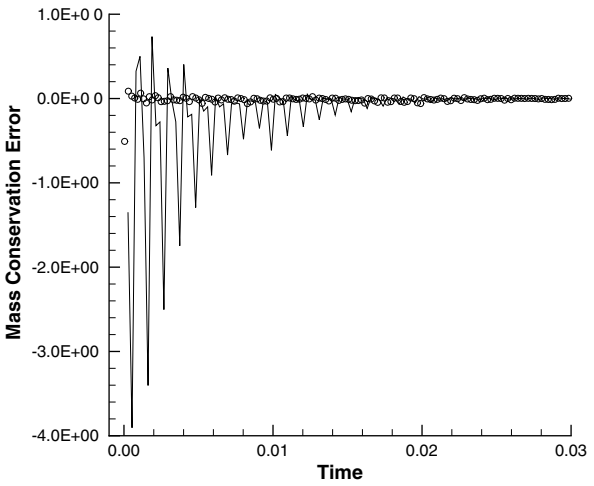


Fig. 5. Example 1: The comparison of total mass conservation errors caused by the adaptive real GFM method ("○") and the real GFM method on the uniform mesh of 200 cells (the solid line).

detailed comparison of various GFMs. Theoretically as discussed at length in [35], such non-physical reflection is indeed very difficult to remove completely, even though the very fine mesh is used. Shown in Figs. 3 and 4 are the respective results obtained by the present adaptive GFM and the real GFM on uniform meshes at time $t = 0.06$, where we have taken $\alpha = 100$ and $\beta = 80$. The non-physical reflection observed in Fig. 4 has been suppressed almost completely by the adaptive GFM. Fig. 5 shows the time-history record of total mass conservation errors for both the present adaptive GFM and the real GFM on the uniform mesh. Here the conservative errors are measured by (3.7) in [35]. Obviously, the adaptive GFM decreases the conservative error further in comparison with the real GFM results on the uniform mesh.

Example 2. We solve a shock tube problem with a gas–water interface, which is identical to Case 4 in [31]. The states on the left and right sides of the initial interface are

$$(\rho, u, p, \gamma, B) = \begin{cases} (0.001, 80, 1, 1.4, 0), & 0 \leqslant x < 0.3, \\ (1, 0, 1, 7.15, 3309), & 0.3 < x \leqslant 1. \end{cases}$$

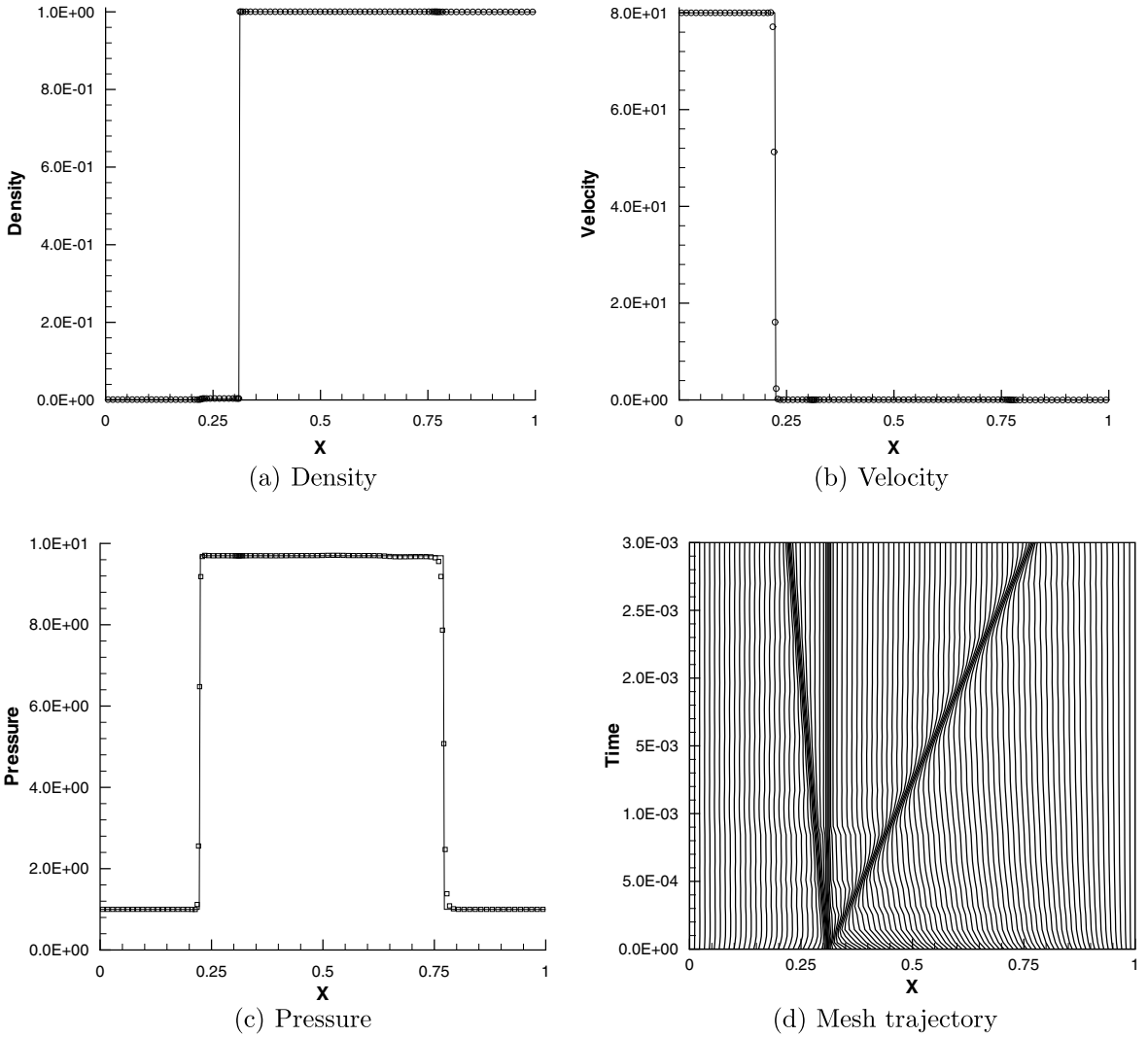


Fig. 6. Example 2: The numerical results ("∘") at time $t = 0.003$ calculated by the proposed adaptive GFM method. The solid line denotes the exact solution.

The parameters α and β in (16) are taken as 100 and 80, respectively.

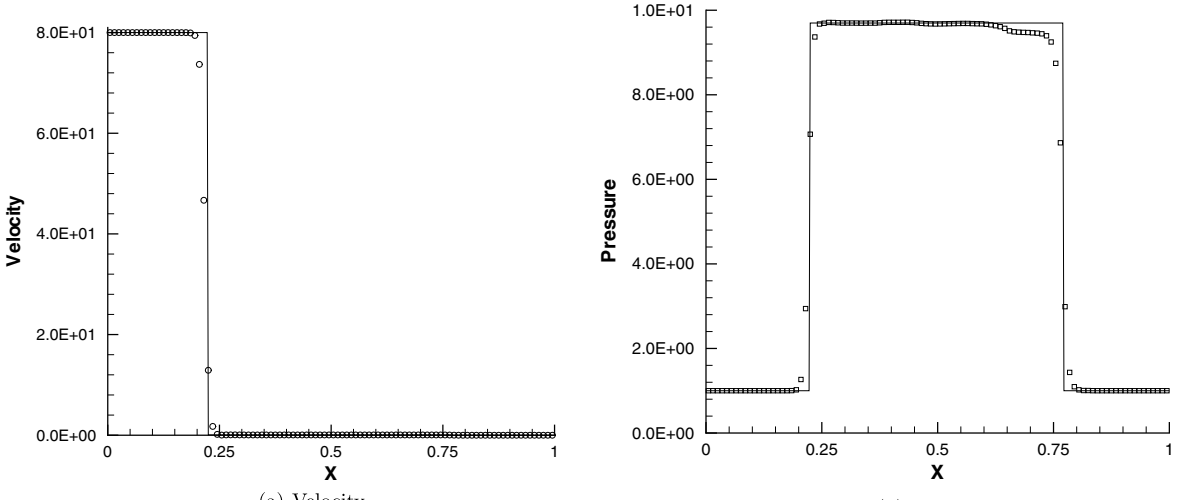
Due to the large difference between the initial gas density and the water density as well as severe oscillations, the original MUSCL-based GFM is unable to provide meaningful results. There are also some small but limited oscillations behind the transmitted shock using the MGFM, even though the overall trend and location of the shock front in the water region are not much affected [31]. Fig. 6 shows the computed solutions at time $t = 0.003$ and the trajectory of the grid points with the adaptive GFM method. For comparison, Fig. 7 gives corresponding results obtained by the real GFM method on a uniform mesh of 100 cells. It is found that both the reflected and transmitted shock waves agree well with the theoretical results with a sharper shock wave front obtained via the adaptive GFM method. Both the locations of the interface and shock front are also predicted accurately. The undesirable effects in Fig. 7 have completely removed by the adaptive GFM method.

Example 3. This example is to solve another shock tube problem with a gas–water interface. The initial interface lies at the middle of the domain $\Omega_p = [0, 1]$. The left side of the interface is filled with gas, while the right is taken by water. In our computations, the initial states on the left and right sides of the interface are taken as

$$(\rho, u, p, \gamma, B) = \begin{cases} (1.27, 0, 8000, 1.4, 0), & 0 \leqslant x < 0.5, \\ (1, 0, 1, 7.15, 3309), & 0.5 < x \leqslant 1. \end{cases}$$

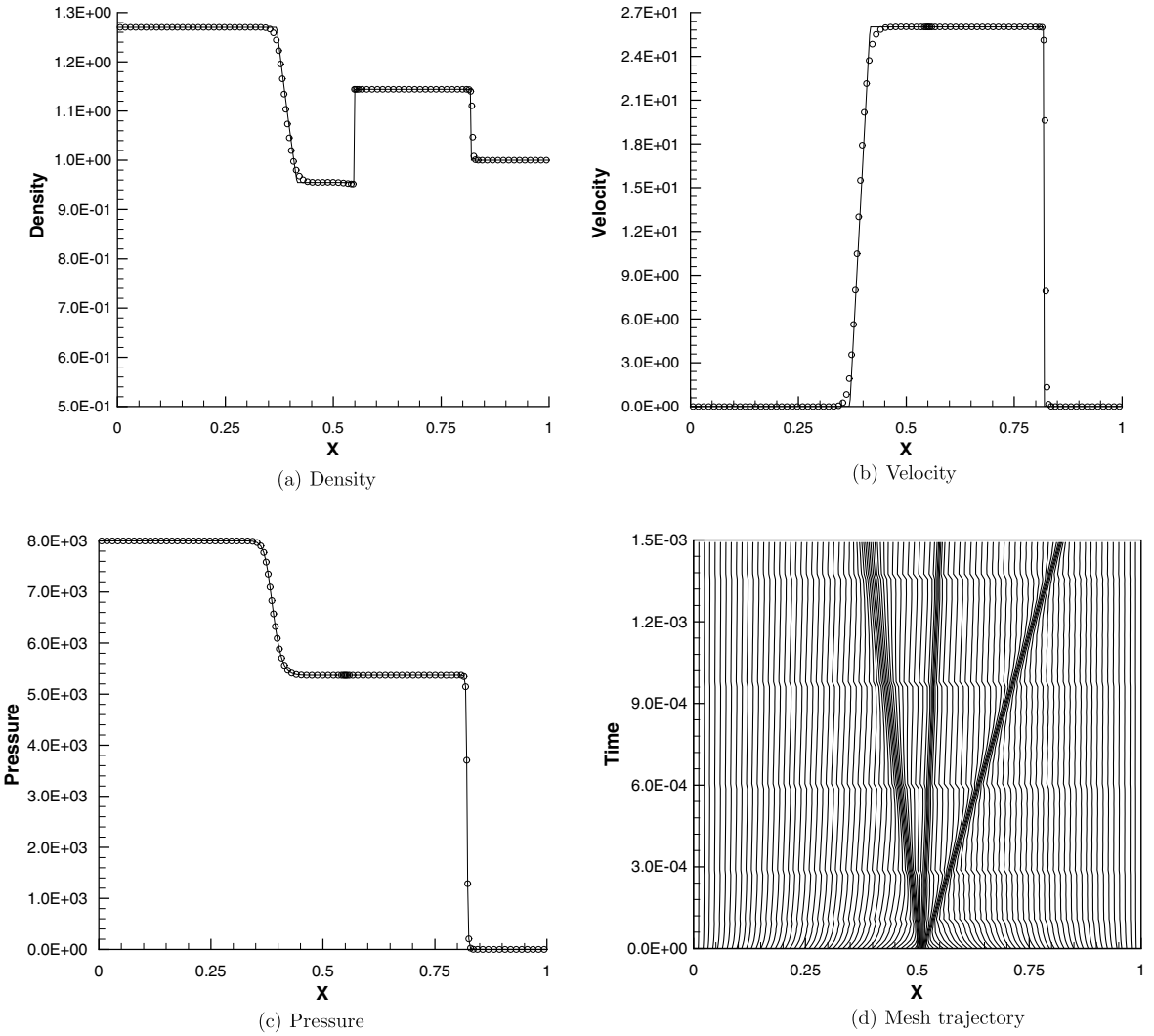
Fig. 8 shows that the numerical results at $t = 0.0015$ calculated by the proposed adaptive GFM method. The mesh trajectory is given in Fig. 8(d). We observe that the initial discontinuity is decomposed into a left-moving rarefaction wave and a right-moving shock wave; the fluid interface is exactly captured, and the rarefaction waves are also captured very well (see Fig. 9).

Example 4. The final 1D example is about a strong shock impacting on a gas–water interface [35]. Initially, a strong shock is located at the same position as the interface, which lies at $x = 0.5$. The shock is defined as $\gamma_S = 1.4$, $\rho_S = 0.00596521$, $u_S = 911.8821$, $p_S = 1000$, and the states on the left and right sides of the interface are defined respectively as $\gamma_L = 1.4$, $\gamma_R = 7.15$; $\rho_L = 0.001$, $\rho_R = 1$; $p_L = 1$, $p_R = 1$; $u_L = u_R = 0$ and $B_L = 0$, $B_R = 3309$. Fig. 10 shows the numerical results with the adaptive GFM algorithm at time $t = 0.0007$. Fig. 10(d) shows the trajectory of the grid points. It is found that both the reflected and transmitted shock waves agree well with the theoretical results and a sharper shock wave is obtained with the adaptive mesh method. The interface and shock front are also predicted accurately. One may also observe that the results



(a) Velocity (b) Pressure

Fig. 7. Same as Fig. 6, except for an uniform mesh with 100 cells.



(a) Density (b) Velocity

(c) Pressure (d) Mesh trajectory

Fig. 8. Example 3: The numerical results ("○") at time $t = 0.0015$ calculated by using the proposed adaptive GFM method. The solid line denotes the exact solution.

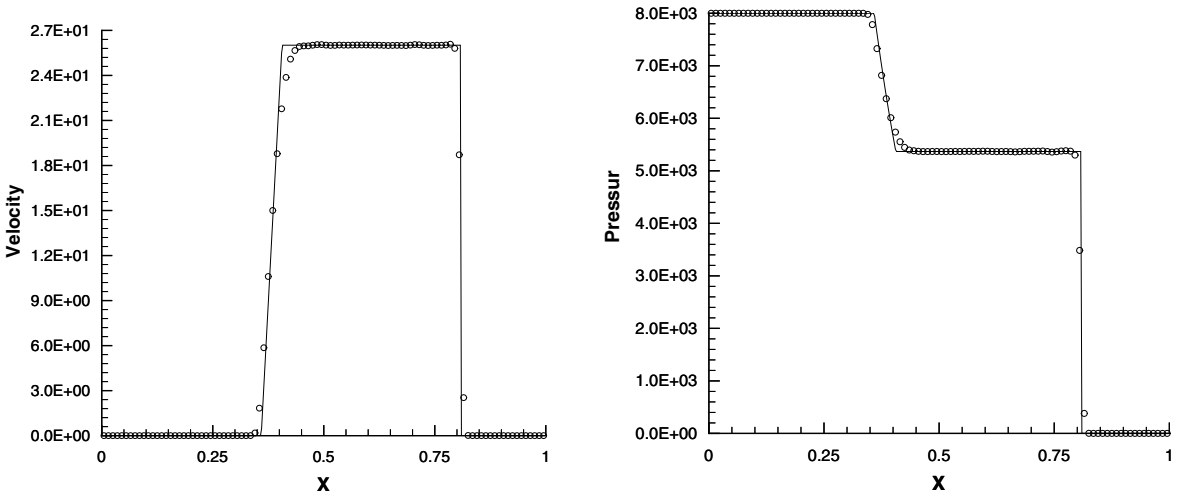
by the adaptive GFM method gain much improvement in comparison to the results by the real GFM on the uniform mesh as shown in Fig. 11.

5.2. Two-dimensional examples

Three 2D examples will be considered in this subsection. Throughout our computations, the CFL number is taken as 0.24. The approximate Riemann solver described in [33,31] is similarly employed to supply the intermediate states.

Example 5. This example is to investigate a high-pressure air cavity expanding in water. Initially, a high-pressure, cylindrical air cavity is at the center of the physical domain $\Omega_p = [0,4] \times [0,4]$ with a diameter of 1. The initial states inside and outside the cylindrical cavity are taken as

$$(\rho, u, v, p, \gamma, B) = \begin{cases} (1.27, 0, 0, 8000, 1.4, 0), & \text{inside the cylinder,} \\ (1, 0, 0, 1, 7.15, 3309), & \text{outside the cylinder.} \end{cases}$$



(a) Velocity (b) Pressure

Fig. 9. Same as Fig. 8, except for a uniform mesh of 100 cells.

The physical domain Ω_p is divided into 100×100 mesh cells, and the monitor function ω is chosen as

$$\omega = \sqrt{1 + \alpha\left(\frac{\rho_\xi^2 + \rho_\eta^2}{\max\{\rho_\xi^2 + \rho_\eta^2\}}\right) + \beta\left(\frac{p_\xi^2 + p_\eta^2}{\max\{p_\xi^2 + p_\eta^2\}}\right)}, \tag{17}$$

with $\alpha = 180$ and $\beta = 160$.

Fig. 12 gives numerical results obtained at time $t = 0.0045$, which show that a water shock is formed running towards the boundary of the domain and an expansive wave is running towards the center of the domain. We see that the solutions are almost cylindrically symmetric, and the level set function is a good approximation of the distance function. Fig. 12(d) gives a comparison of the density distributions along $y = 2$ obtained by using the adaptive GFM method ('$\circ$') and the real GFM method on an uniform mesh of 200×200 mesh cells. It shows that the present adaptive GFM algorithm can capture the discontinuities and interface accurately with far few mesh points.

Example 6. This example is about a strong water shock impacting on a cylindrical air bubble in water. It has been numerically and experimentally investigated by several authors, see e.g. [7]. Here, we consider the similar setup of the initial and boundary conditions as in [7], but they are in a non-dimensional form and only the upper half of the problem is simulated.

In our computations, the physical domain Ω_p is taken as $[0, 15] \times [0, 6]$, and divided into 150×60 mesh cells; the lower boundary is the symmetry plane, the non-reflecting conditions are specified on the upper and right boundaries, while the left boundary conditions are taken as the post-shock values (the non-reflecting) before (after) the rarefaction wave reaches to the left boundary.

Initially, a cylindrical air bubble with a diameter of 6 is immersed in water, and centered at $(6, 0)$, see the schematic diagram shown in Fig. 13. Inside the cylindrical air bubble, the physical variables $(\rho, u, v, p, \gamma, B)$ are taken as $(0.001, 0, 0, 1, 1.4, 0)$. A strong water shock wave neighbors initially the air bubble with pressure strength of 19,000 and strikes the air bubble. The pre- and post-shock states of the incident water shock wave are

$$(\rho, u, v, p, \gamma, B) = \begin{cases} (1.313345, 67.3267, 0, 19000, 7, 3309), & x < 3, \\ (1, 0, 0, 1, 7, 3309), & x > 3, \end{cases}$$

where the post-shock values are obtained by using the Rankine-Hugoniot condition. It means that the initial water shock wave is propagating from the left to the right, and will strike the air bubble.

Figs. 14 and 15 present the adaptive meshes and the pressure contours at several moments, which are obtained by using the proposed adaptive GFM method. In order to resolve the transmitted weak shock wave in the air bubble, the monitor function ω has been chosen as

$$\omega = \sqrt{1 + \alpha\left(\frac{\rho_\xi^2 + \rho_\eta^2}{\max\{\rho_\xi^2 + \rho_\eta^2\}}\right) + \beta\left(\frac{u_\xi^2 + v_\eta^2}{\max\{u_\xi^2 + v_\eta^2\}}\right)}, \tag{18}$$

with $\alpha = 120$ and $\beta = 160$. The results show that the initial incident water shock first struck and deformed the left wall of the air bubble, and then a relatively weak shock wave running rightward was transmitted into the air. A strong expansion fan moving leftward was produced in the water. The air shock wave propagated more slowly than the incident water shock, and was decoupled from the incident shock after $t = 0.02$. At $t = 0.025$ a water jet was formed and running to the right along the symmetry plane $y = 6$. At $t = 0.031$, the water jet reached the right wall of the air bubble, resulting in bubble collapse. Due to bubble collapse, an intense blast wave in the surrounding water is produced at $t = 0.032$. The air bubble resembles a tear-drop and the air

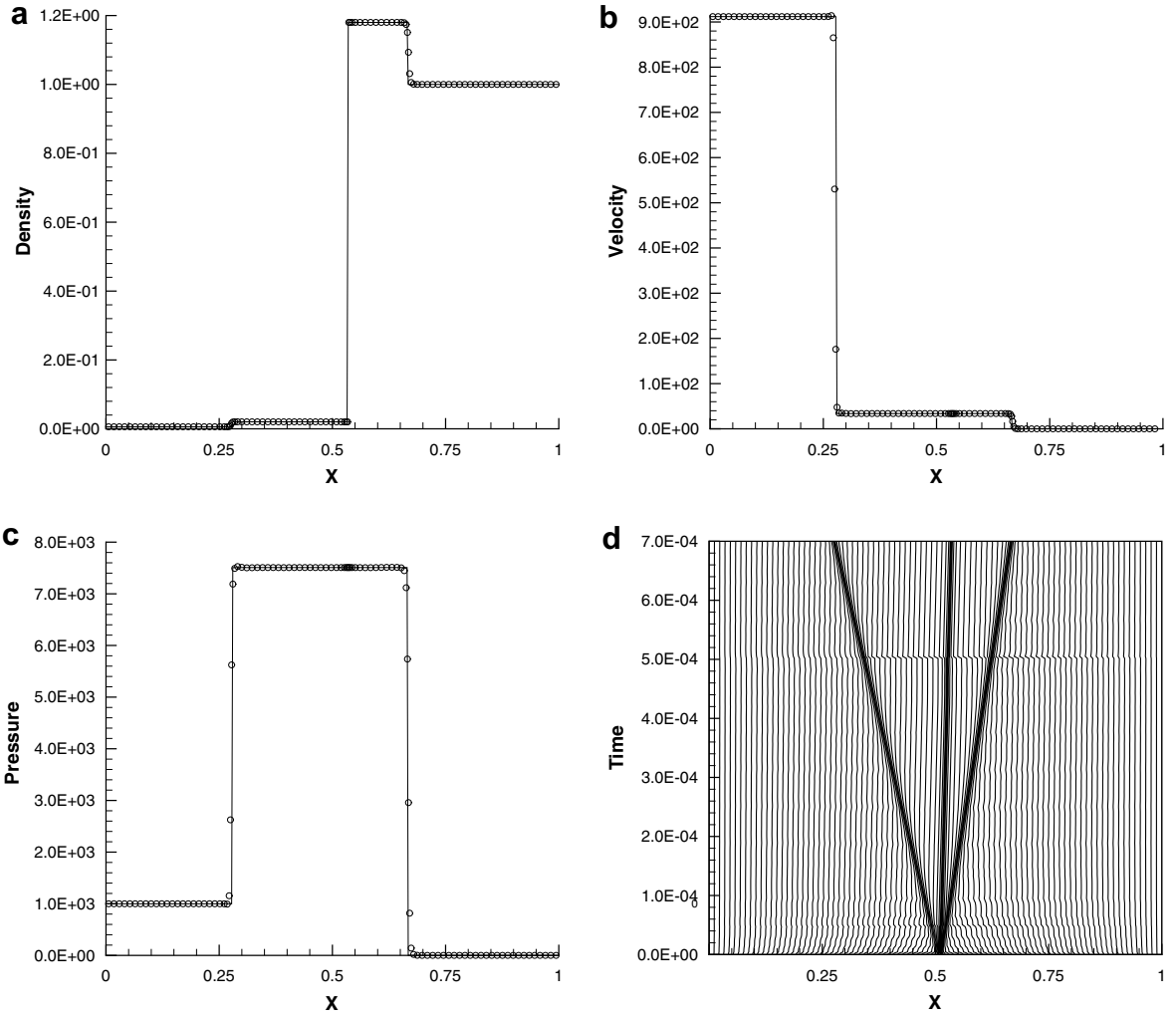
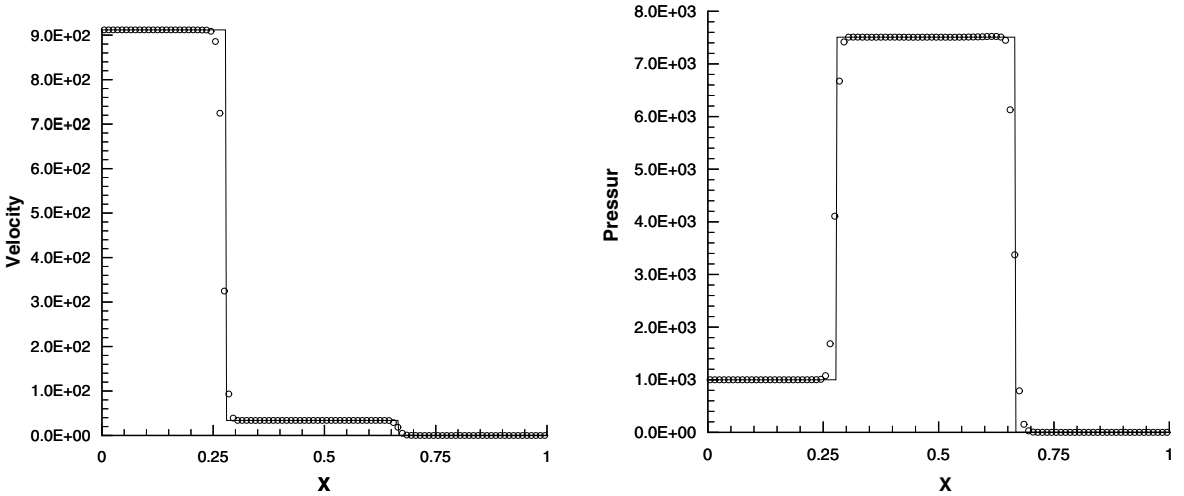


Fig. 10. Example 4: The numerical results ("○") at $t = 0.0007$ obtained by using the proposed adaptive GFM algorithm. The solid line denotes the exact solution.



(a) Velocity (b) Pressure

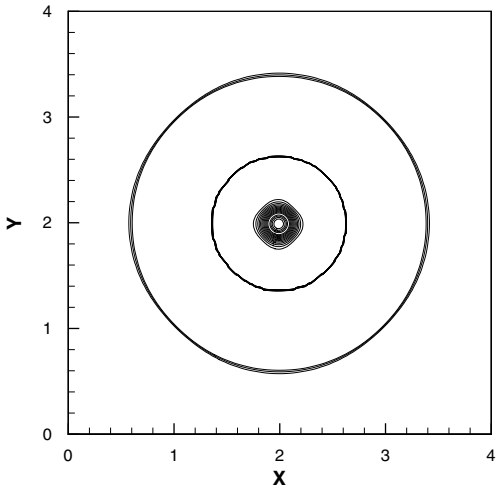
Fig. 11. Same as Fig. 10 except for a uniform mesh of 100 cells.

shock (now traveling to the left) has begun to interact with the lower left bubble wall and produces a Mach reflection, which is more apparent at $t = 0.033$. Final three plots in Figs. 14 and 15 show the subsequent evolution of the flow and the adaptive mesh. At $t = 0.037$, a compression wave propagates outwards from the air bubble. This appears to result from transmission of the air shock into the water. At the same time, the air bubble begins to draw into the vortex core due to baroclinicity.

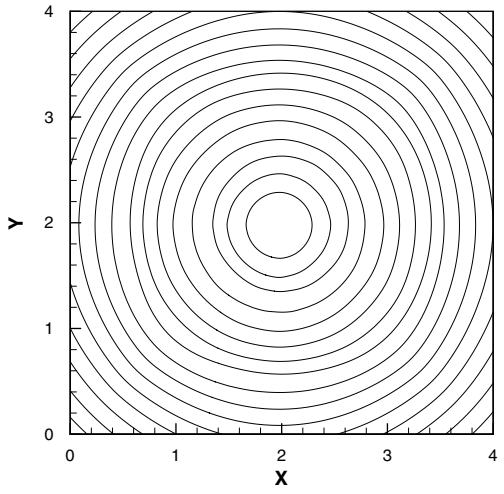
From Figs. 14 and 15, we see clearly that the bubble evolution is accurately tracked. In addition, both the strong water shock wave and weak air shock waves inside the bubble have been well resolved. Our results are well comparable to those obtained by the free-Lagrange code of Ball et al. [7], who used 5×10^4 cells for half of the domain Ω_p. Preventing mass loss is the biggest challenge in simulating this problem [40]. Fig. 16 shows the recorded total mass loss of the gas bubble with time. It is found that the biggest mass loss occurs at the moment of the bubble collapse with a loss of 13%. After the bubble collapse, some mass loss of the gas bubble is erroneously recovered numerically. For this problem, the total mass loss can also occur to a pure Lagrangian method due to topological change. With a relatively small amount of mesh points, the mass loss is still limited in about 10%.

Fig. 17 gives a comparison of the density distributions at $t = 0.012$ and 0.025 along the symmetric plane obtained by using the adaptive GFM method ('$\circ$') and the real GFM on a uniform mesh of 450×180 mesh cells, respectively. The results show that the adaptive GFM method can capture the discontinuities well and track the interface accurately with far fewer mesh points. To demonstrate the efficiency of the present adaptive GFM method, we estimate the CPU times of two algorithms on the Pentium-III with 2 Ghz under a Windows environment. The CPU times are 2702 (2860) s for the present adaptive GFM method and 7330 (7753) s for the original real GFM method on a uniform mesh of 450×180 cells, respectively, when the solution is evolved from $t = 0$ to 0.012 (from $t = 0.012$ to 0.025). It means that about 63.14% of the cost is saved in the computation by using the adaptive GFM method. It is noted that we are comparing CPU times of two algorithms when the computational results are of (relatively) same quality. Our experiments have demonstrated that the distance function is maintained well, and the level set contours are almost uniformly distributed around the interface.

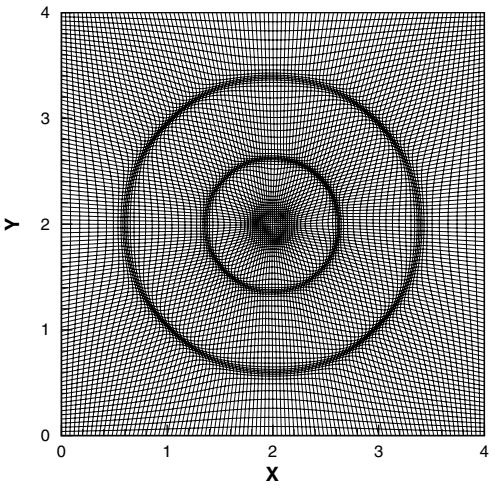
Example 7. The final 2D example is about a gas shock wave impacting on a cylindrical liquid mass. It has been numerically and experimentally investigated by several authors, see e.g. [40]. This problem is particularly difficult, because it involves a curved gas/liquid slow-fast interface, and associated with rather complex irregular shock refraction patterns. Here, we consider the similar setup of the initial and boundary conditions as in [40], but they are in a non-dimensional form and only the upper half of the problem is simulated. Initially, in the



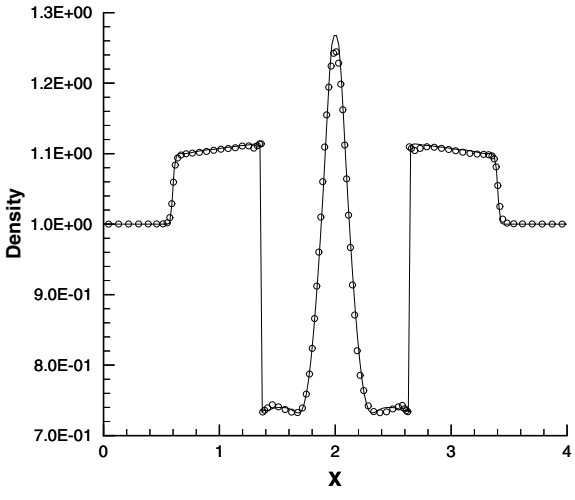
(a) Density



(b) Level set function



(c) Adaptive mesh



(d) Densities along $y = 2$

Fig. 12. Example 5: The numerical results at $t = 0.0045$ obtained by using the proposed adaptive GFM algorithm.

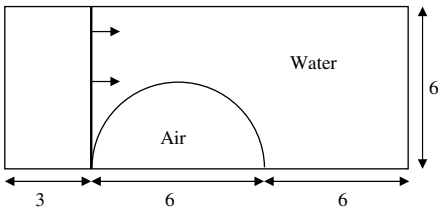
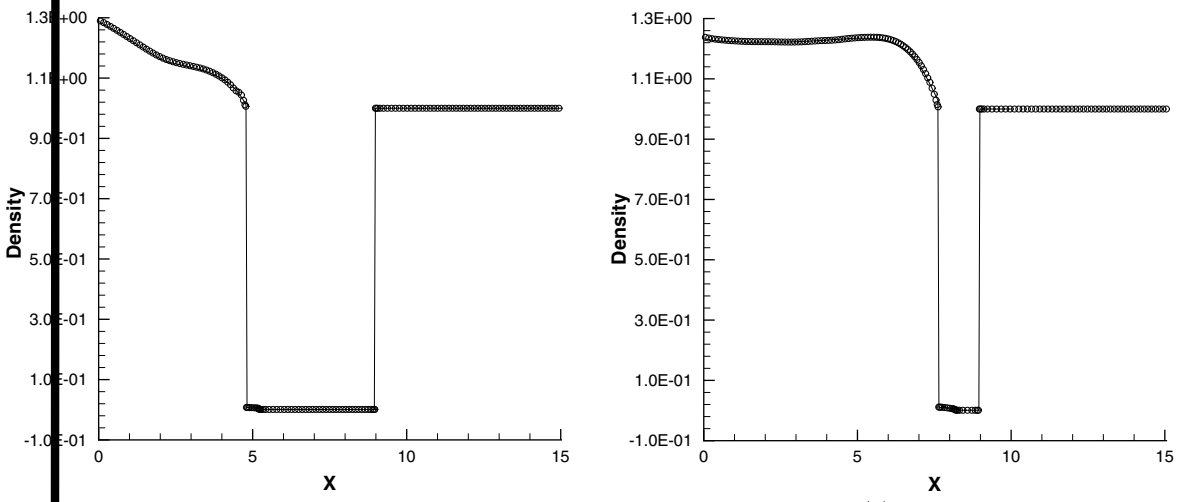


Fig. 13. The geometry of Example 6.

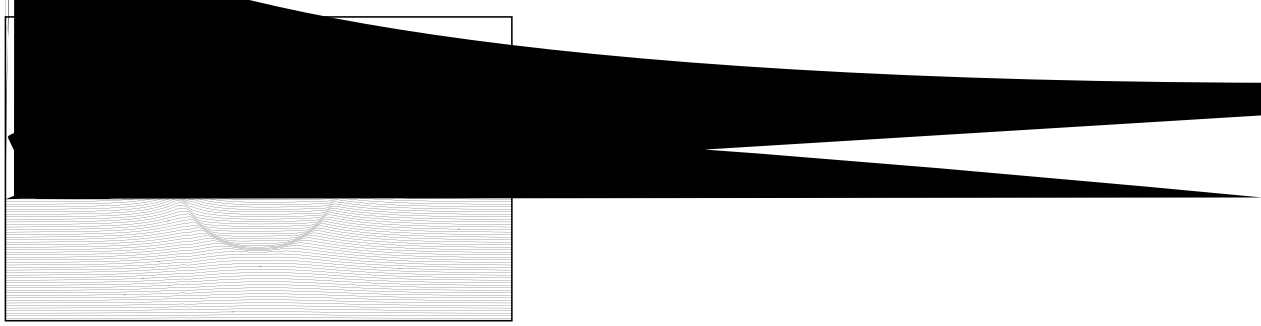
physical domain $\Omega_p = [0, 20] \times [0, 6]$, a cylindrical water drop with a diameter of 3.2 is surrounded by air, and centered at $(10, 0)$. Inside the cylindrical water drop, the physical variables $(\rho, u, v, p, \gamma, B)$ are taken as (1, 0,0, 1, 2.8, 3036). A right-moving gas shock wave neighbors initially the water drop and strikes the water drop. The pre- and post-shock states of the incident water shock wave are

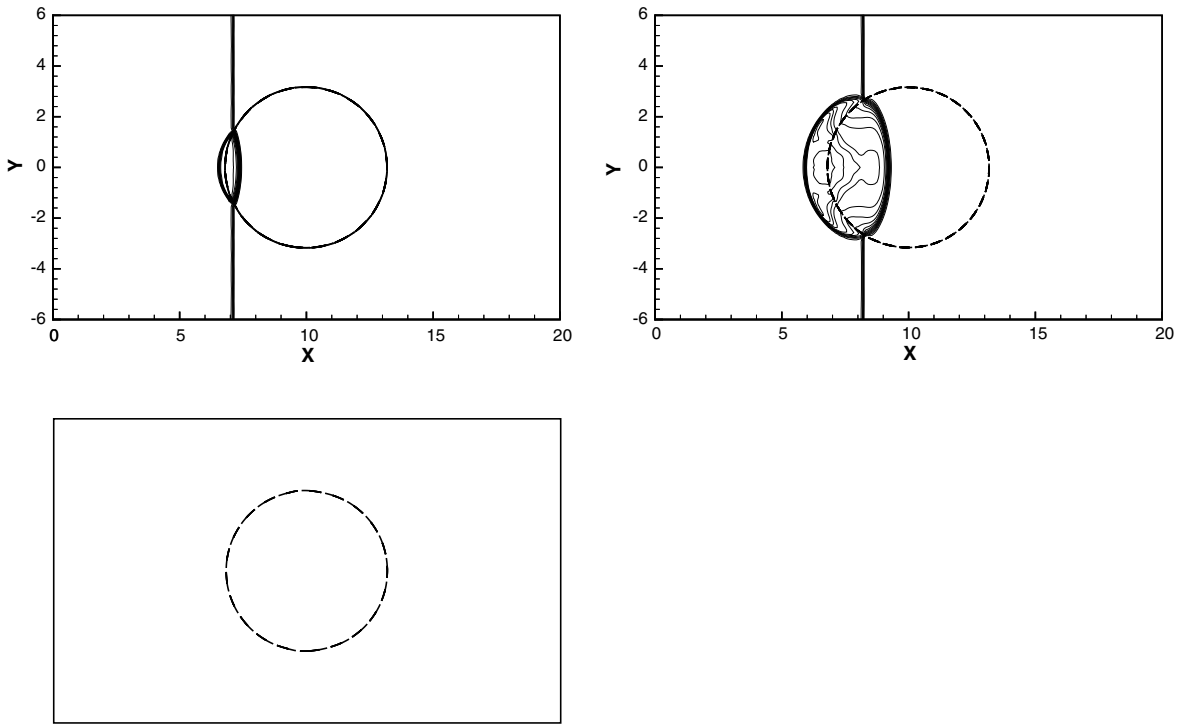
the drop surface causes the present algorithm down without special treatments due to the imposition of pressure continuous across the drop interface. We emphasize that the difficulties caused by the negative pressure are nothing to do with the present numerical method developed but comes purely from physical sides. As a matter of fact, such difficulties can be easily overcome by introducing a proper cavitation model as done in [58]. The discussion on cavitation modeling is, however, not the topic of the present work.



(a) $t = 0.012$ (b) $t = 0.025$

Fig. 17. Example 7: A comparison of the density alone the symmetrical plane. The symbol "$\circ$" and the solid line denote the densities obtained by the adaptive GFM method and the real GFM method on a uniform mesh, respectively.





6. Concluding remarks

In this paper we have developed an efficient adaptive ghost fluid finite volume method for one- and two-dimensional multi-medium flows. The proposed method couples the real ghost fluid method (GFM) [55] and the adaptive moving mesh method [51,52] and thus retains their advantages such as flexible treatment of the fluid interface in multi-medium flows and effective reduction in numerical errors as well as the improvement of the solution resolution while overcoming their weaknesses such as conservative errors of the GFM. It consists of two parts: the PDE evolution and the iterative mesh-redistribution. In the first part, the real ghost fluid method (real GFM) is applied to define the interface boundary conditions and ghost fluid states, the computation is then carried out for a single-medium problem such that the general schemes can be used to discretize the governing equations of fluid flows. In the second part, the mesh is first redistributed. To keep a sharp interface, the associated updating technique of the conservative fluid variables is implemented on both sides of the interface by using the real GFM, respectively. The level set function is interpolated by directly using the high-resolution Hamilton–Jacobi solver [52]. Technique to prevent interface shift during the mesh-redistribution was also developed.

The proposed method has been applied to several one- and two-dimensional gas–water flow problems involving large gradient density at the interface and strong shock-interface interaction. The numerical results have shown that (1) locally clustering mesh points in the vicinity of the material interface can effectively reduce numerical errors caused by the GFM around the material interface and other discontinuities; (2) the proposed method can capture the shock waves and the material interface accurately, and is stable and robust even for large density and pressure.

Acknowledgments

The authors would like to thank anonymous referees' valuable suggestions during the revision of the paper. The authors also thank an anonymous referee mentioning the work on the conservative implicit-cut-cell

algorithm [41]. Chunwu Wang was partially supported by the National Natural Science Foundation of China (No. 10601023) and the Institute for Mathematical Sciences of National University of Singapore. Huazhong Tang was partially supported by the National Basic Research Program under the Grant 2005CB321703, the National Natural Science Foundation of China (Nos. 10431050 and 10576001), SRF for ROCS, SEM, NCET, and Doctoral Program of Education Ministry of China (No. 20070001036).

References

[1] R. Abgrall, How to prevent oscillations in multicomponent flow calculations: a quasi conservative approach, J. Comput. Phys. 125 (1996) 150–160.
[2] R. Abgrall, S. Karni, Computations of compressible multifluids, J. Comput. Phys. 169 (2001) 594–623.
[3] D. Adalsteinsson, J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys. 118 (1995) 269–277.
[4] D. Adalsteinsson, J.A. Sethian, The fast construction of extension velocities in level set methods, J. Comput. Phys. 148 (1999) 2–22.
[5] M. Aivazis, W.A. Goddard, D. Meiron, M. Ortiz, J. Pool, J. Shepherd, A virtual test facility for simulating the dynamic response of materials, Comput. Sci. Eng. 2 (2000) 42–53.
[6] M. Arienti, P. Hung, E. Morano, J.E. Shepherd, A level set approach to Eulerian–Lagrangian coupling, J. Comput. Phys. 185 (2003) 213–251.
[7] G.J. Ball, B.P. Howell, T.G. Leighton, M.J. Schofield, Shock-induced collapse of a cylindrical air cavity in water: a free-Lagrange simulation, Shock Waves 10 (2000) 265–276.
[8] J.U. Brackbill, An adaptive grid with directional control, J. Comput. Phys. 108 (1993) 38–50.
[9] 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.
[10] W.M. Cao, W.Z. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.
[11] H.D. Ceniceros, T.Y. Hou, An efficient dynamicaly adaptive mesh for potentially singular solutions, J. Comput. Phys. 172 (2001) 609–639.
[12] I.L. Chern, J. Glimm, O. McBryan, B. Plohr, S. Yaniv, Front tracking for gas dynamics, J. Comput. Phys. 62 (1986) 83–110.
[13] S.F. Davis, J.E. Flaherty, An adaptive finite element method for initial-boundary value problems for partial differential equations, SIAM J. Sci. Stat. Comput. 3 (1982) 6–27.
[14] Y.N. Di, R. Li, T. Tang, P.W. Zhang, Moving mesh finite element methods for the incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 26 (2005) 1036–1056.
[15] Y. Di, P. Zhang, Moving mesh kinetic simulation for sheared rodlike polymers with high potential intensities, Commun. Comput. Phys. 1 (2006) 859–873.
[16] A.S. Dvinsky, Adaptive grid generation from harmonic maps on Riemannian manifolds, J. Comput. Phys. 95 (1991) 450–476.
[17] 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.
[18] 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) 457–492.
[19] R.P. Fedkiw, T. Aslam, S. Xu, The ghost fluid method for deflagration and detonation discontinuities, J. Comput. Phys. 154 (1999) 393–427.
[20] J. Glimm, X.L. Li, Y. Liu, N. Zhao, Conservative front tracking and level set algorithms, PNAS 98 (2001) 14198–14201.
[21] W. Huang, Y. Ren, R.D. Russell, Moving mesh methods based on moving mesh partial differential equations, J. Comput. Phys. 113 (1994) 279–290.
[22] W.Z. Huang, Mathematical principles of anisotropic mesh adaptation, Commun. Comput. Phys. 1 (2006) 276–310.
[23] E. Johnsen, T. Colonius, Implementation of WENO schemes in compressible multicomponent flow problems, J. Comput. Phys. 219 (2006) 715–732.
[24] S. Karni, Hybrid multifluid algorithms, SIAM J. Sci. Comput. 17 (1996) 1019–1039.
[25] S. Karni, Multicomponent flow calculations by a consistent primitive algorithm, J. Comput. Phys. 112 (1994) 31–43.
[26] B. Kim, Y. Liu, I. Llamas, J. Rossignac, Advections with significantly reduced dissipation and diffusion, IEEE Trans. Vis. Comput. Graphics 13 (2007) 135–144.
[27] B. Koren, M.R. Lewis, E.H. van Brummelen, B. van Leer, Riemann-problem and level-set approaches for homentropic two-fluid flow computations, J. Comput. Phys. 181 (2002) 654–674.
[28] B. Larouturou, How to preserve the mass fraction positive when computing compressible multi-component flows, J. Comput. Phys. 95 (1991) 59–84.
[29] R. Li, T. Tang, P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.
[30] R. Li, T. Tang, P.-W. Zhang, A moving mesh finite element algorithm for singular problems in two and three space dimensions, J. Comput. Phys. 177 (2002) 365–393.

[31] T.G. Liu, B.C. Khoo, C.W. Wang, The ghost fluid method for gas–water simulation, J. Comput. Phys. 204 (2005) 193–221.
[32] T.G. Liu, B.C. Khoo, W.F. Xie, The modified ghost fluid method as applied to extreme fluid-structure interaction in the presence of cavitation, Commun. Comput. Phys. 1 (2006) 898–919.
[33] T.G. Liu, B.C. Khoo, K.S. Yeo, The simulation of compressible multi-medium flow. Part I: a new methodology with applications to 1D gas–gas and gas–water cases, Comput. Fluids 30 (2001) 291–314.
[34] T.G. Liu, B.C. Khoo, K.S. Yeo, The simulation of compressible multi-medium flow. Part II: applications to 2D underwater shock refraction, Comput. Fluids 30 (2001) 315–337.
[35] T.G. Liu, B.C. Khoo, K.S. Yeo, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys. 190 (2003) 651–681.
[36] X.D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domain, J. Comput. Phys. 160 (2000) 151–178.
[37] F. Losasso, F. Gibou, R. Fedkiw, Simulating water and smoke with an octree data structure, in: Proceedings of ACM SIGGRAPH 2004, 2004, pp. 457–462.
[38] K. Miller, R.N. Miller, Moving finite element. I, SIAM J. Numer. Anal. 18 (1981) 1019–1032.
[39] D.Q. Nguyen, F. Gibou, R.P. Fedkiw, M. Kang, A fully conservative ghost fluid method and stiff detonation waves, in: The 12th International Detonation Symposium, San Diego, CA, 2002.
[40] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, Adaptive characteristics-based matching for compressible multifluid dynamics, J. Comput. Phys. 213 (2006) 500–529.
[41] R.R. Nourgaliev, T.G. Theofanous, H. Park, V. Mousseau, D. Knoll, Direct numerical simulation of interfacial flows: implicit sharp-interface method (I-SIM), AIAA 2008-1453, 46rd AIAA Aerospace Sciences Meeting and Exhibit, January 7–10, 2008, Reno, NV, USA.
[42] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.
[43] J.X. Qiu, T.G. Liu, B.C. Khoo, Runge–Kutta discontinuous Galerkin methods for compressible two-medium flow simulations: one-dimensional case, J. Comput. Phys. 222 (2007) 353–373.
[44] W.Q. Ren, X.P. Wang, An iterative grid redistribution method for singular problems in multiple dimensions, J. Comput. Phys. 159 (2000) 246–273.
[45] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
[46] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[47] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.
[48] C.W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker, S. Osher, High order ENO schemes applied to two and three dimensional compressible flow, Appl. Numer. Math. 9 (1992) 45–71.
[49] Z.J. Tan, Z.R. Zhang, T. Tang, Y.Q. Huang, Moving mesh methods with locally varying time steps, J. Comput. Phys. 200 (2004) 347–367.
[50] H.Z. Tang, A moving mesh method for the Euler flow calculations using a directional monitor function, Commun. Comput. Phys. 1 (2006) 656–676.
[51] H.Z. Tang, T. Tang, Moving mesh methods for one- and two-dimensional hyperbolic conservation laws, SIAM J. Numer. Anal. 41 (2003) 487–515.
[52] H.Z. Tang, T. Tang, P.W. Zhang, An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimensions, J. Comput. Phys. 188 (2003) 543–572.
[53] E.H. van Brummelen, B. Koren, A pressure-invariant conservative Godunov-type method for barotropic two-fluid flows, J. Comput. Phys. 185 (2003) 289–308.
[54] B. van Leer, Towards the ultimate conservative difference scheme V: a second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[55] C.W. Wang, T.G. Liu, B.C. Khoo, A real-ghost fluid method for the simulation of multi-medium compressible flow, SIAM J. Sci. Comput. 28 (2006) 278–302.
[56] A.B. Wardlaw, J.A. Luton, Fluid-structure interaction mechanisms for close-in explosions, Shock Vib. 7 (2000) 265–275.
[57] A. Winslow, Numerical solution of the quasi-linear Poisson equation, J. Comput. Phys. 1 (1967) 149–172.
[58] W.F. Xie, T.G. Liu, B.C. Khoo, The simulation of cavitating flows induced by underwater shock-free surface interaction, Appl. Numer. Math. 57 (2007) 734–745.