
Volume 49, issue 1, October 2011
ISSN 0045-7930
ELSEVIER
An International Journal
computers
&
fluids
Editor-in-Chief:
MICHEL DEVILLE
Associate Editor:
JEAN-JACQUES CHATTOT

Computers & Fluids 49 (2011) 247–257

Contents lists available at ScienceDirect

Computers & Fluids

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

# A finite difference real ghost fluid method on moving meshes with corner-transport upwind interpolation

Yan Ding [a,*], Li Yang [b], Li Yuan [c]

[a] School of Mathematics and Information Science, Henan Polytechnic University, Jiaozuo 454150, China
[b] School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo 454150, China
[c] LSEC and ICMSEC, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

*Article history:*
Received 27 January 2011
Received in revised form 12 May 2011
Accepted 12 June 2011
Available online 6 July 2011

*Keywords:*
Real ghost fluid method
Moving meshes
Corner-transport upwind method
High resolution correction

ABSTRACT

The real ghost fluid method (RGFM) [Wang CW, Liu TG, Khoo BC. A real-ghost fluid method for the simulation of multi-medium compressible flow. SIAM J Sci Comput 2006;28:278–302] has been shown to be more robust than previous versions of GFM for simulating multi-medium flow problems with large density and pressure jumps. In this paper, a finite difference RGFM is combined with adaptive moving meshes for one- and two-dimensional problems. A high resolution corner-transport upwind (CTU) method is used to interpolate approximate solutions from old quadrilateral meshes to new ones. Unlike the dimensional splitting interpolation, the CTU method takes into account the transport across corner points, which is physically more sensible. Several one- and two-dimensional examples with large density and pressure jumps are computed. The results show the present moving mesh method can effectively reduce the conservative errors produced by GFM and can increase the computational efficiency.

## 1. Introduction

In compressible multi-medium flow problems, the equations of state (EOS) are different and some physical quantities are discontinuous across the material interface. This poses challenging problems for numerical simulation. The key point in modeling such problems is how to correctly track the interface location and how to simulate the interaction of different fluids. Generally, there are two different approaches to deal with compressible multi-medium flows. One is the front capturing in which the fluid interface is treated as steep gradients to be resolved over a few grid cells, the other is the front tracking in which the fluid interface is explicitly tracked as an internal moving boundary. By comparison, the front capturing is more simple and easy to code, but physical quantities near the interface may be smeared and unphysical oscillations usually occur in naive implementation of high-resolution conservative schemes, while the front tracking can maintain the interface states well but is rather complicated in multi-dimension cases.

Level set method as developed by Osher and Sethian [1,2] has become one of the most widely used front capturing methods. It represents the interface as the zero level set of a signed distance function, and the interface location is “implicitly” captured by solving the level set equation which propagates the level set with the fluid velocity. A critical step in the level set method is to preserve the signed distance property of the level set function in the vicinity of the interface. For this, reinitialization is often used [3]. However, it is costly and often fail to recover signed distance function exactly. Meanwhile, other techniques for maintaining the signed distance property were also developed [4,5]. Recently, while developing the real ghost fluid method (RGFM), Wang et al. [6] proposed an approach which avoids reinitialization by using a constructed extension velocity to convect the level set function.

The ghost fluid method (GFM) as proposed by Fedkiw et al. [7], is an effective method for simulating multi-medium flows in Eulerian schemes that is able to maintain a Heaviside profile of the density with no numerical smearing. It treats the solution of the multi-medium problem as solutions of two single-medium problems. In doing so, GFM defines two types of fluids, i.e., real fluid and ghost fluid, at each grid point. The ghost fluid takes the pressure, normal velocity component from the local fluid it coincides, and obtains the entropy and tangential velocity component by extrapolating from the other side of the interface where corresponding real fluid exists. This definition of ghost fluid status as given in [7] (called OGFM in this study) was found not robust when applied to gas–water flows [8]. Therefore, a new definition of ghost fluid status was proposed especially for treating gas–water flows in [8] (NGFM). Specifically, the ghost fluid pressure is obtained from the less stiff gas and the ghost fluid normal velocity component is obtained from the more stiff water, while the ghost fluid entropy and tangential velocity component are obtained in the same way as in OGFM.

---

* Corresponding author. Tel.: +86 0391 3987791.
*E-mail addresses:* dingyan@lsec.cc.ac.cn (Y. Ding), yangli@hpu.edu.cn (L. Yang), lyuan@lsec.cc.ac.cn (L. Yuan).

0045-7930/$ - see front matter Crown Copyright © 2011 Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2011.06.004

Many compressible multi-medium flow problems have the situation where a strong shock wave interacts with the material interface. Since the pattern of this interaction at the interface and the resultant interfacial states are nonlinearly affected by the flow material properties on both sides of the interface, this implies that the definition of ghost fluid states by simply duplicating from local real fluid is not sufficient to take into account such interacting effects. In fact, because GFM provides solutions via essentially solving two single-medium Riemann problems, some conditions have to be satisfied in defining the ghost fluid states to ensure that the two single-medium Riemann problems give the correct solutions at the respective real fluid sides. This has led to the development of a modified ghost fluid method (MGFM) by Liu et al. [9]. In this version, a local multi-medium Riemann problem is defined and solved approximately to predict the interfacial status, and the predicted interfacial status are then used to define the ghost fluid status. MGFM has been shown to be more robust and can be applied to various EOS. However, in MGFM and earlier versions of GFM, the focus is on how to define the ghost fluid states while the real pressure and normal velocity at the real fluid nodes neighboring the interface are unchanged and only the density is modified via the simple isobaric fix of [7] which takes reference entropy from a nearby grid point. As a result, in impedance matching problems, non-physical reflection always occurs at the interface and can not be completely removed even with the use of MGFM. In the more recent work of Wang et al. [6], the interfacial states at the real fluid side next to the interface are completely replaced by corresponding approximate Riemann problem solutions. This slightly modified version is called real ghost fluid method (RGFM) [6]. It retains all the good properties of MGFM and can further suppress the non-physical reflection for impedance matching problems. In [10], Liang et al. made comparison among the OGFM and several modified versions of GFM. Those comparisons show that the RGFM (as well as the same technique MGFM) can treat higher pressure and higher density situations such as gas–water interface (pressure ratio 8000:1 in [9]) and shock impinging the fluid interface [9] more robust than the OGFM.

In essence, GFMs are nonconservative because they solve two single-fluid problems instead of the original two-fluid problem and the numerical flux is no longer single-valued at the interface. Among them, MGFM and RGFM have smaller conservative errors. Some efforts have also been made to develop a conservative GFM [11], however, a practical conservative GFM has yet to be developed. On the other hand, the adaptive moving mesh method has been proved to be a very effective way of reducing numerical errors and increasing the resolution of flow field where singular or nearly singular solutions and large gradient variations in flow variables exist. There have been many important progresses in adaptive moving mesh methods for partial differential equations, including the mesh-redistribution approach based on the variation principles [12–14]; moving mesh PDE methods [15,16]; moving mesh methods based on the harmonic mapping [17,18]; and adaptive elliptic grid generation equations [19,20], etc. Recently, an adaptive moving mesh finite volume RGFM [21] was developed for two-medium compressible flows. It was found that the mesh clustering in the vicinity of the material interface can effectively reduce conservative errors produced by RGFM.

In this paper, we couple RGFM proposed by Wang et al. [6] with the adaptive mesh redistribution method developed by Tang et al. [22,23] in the finite difference framework. For adaptive mesh redistribution method, the authors of Ref. [21] mentioned a particular difficulty in the mesh redistribution process for multi-medium flows, i.e., some mesh points may cross over the interface in the iteration. They suggested that after one loop of mesh-redistribution iteration, the crossing point and its neighbor point that borders the interface are pulled back to positions at the previous iteration. However, we will show in the remark of Section 3.2.2 that crossing over the interface is not a problem. Another common difficulty is to accurately interpolate or remap flow solutions and level set function from old mesh onto new mesh after each mesh redistribution iteration. In this study, we apply the high-resolution corner-transport upwind (CTU) method [24,25] to interpolation. The CTU method includes the proper transport across corner points, and was shown more accurate and stable [25]. This method is compared with traditional dimensional splitting conservative interpolation [23,21] in 2D examples.

The paper is organized as follows. In Section 2, the governing equations for multi-medium flows in generalized curvilinear coordinates, the stiffened gas EOS, and the level set equation are given. In Section 3, the RGFM is introduced, then the adaptive moving mesh method and CTU interpolation method which utilize RGFM is given in detail. The complete solution procedure is outlined in Section 4. Numerical experiments are carried out in Section 5 to demonstrate the effectiveness of the present adaptive method. Conclusions are made in Section 6.

## 2. Governing equations

Suppose $\Omega_c$ be a two-dimensional computational domain with the coordinate system $(\xi,\eta)$, and $\Omega_p$ a physical domain with the Cartesian coordinate system $(x,y)$, where $x = x(\xi,\eta)$, $y = y(\xi,\eta)$ is a coordinate transformation from $\Omega_c$ to $\Omega_p$, which is established once the physical grid is created. The Euler equations for two-fluid compressible flows take the following form in generalized coordinate system $(\xi,\eta)$

$$\frac{\partial \widehat{\mathbf{U}}}{\partial t} + \frac{\partial \widehat{\mathbf{F}}_1}{\partial \xi} + \frac{\partial \widehat{\mathbf{F}}_2}{\partial \eta} = 0, \tag{1}$$

where

$$\begin{aligned}
\widehat{\mathbf{U}} &= J\mathbf{U},\\
\widehat{\mathbf{F}}_1 &= J(\xi_x \mathbf{F}_1 + \xi_y \mathbf{F}_2),\\
\widehat{\mathbf{F}}_2 &= J(\eta_x \mathbf{F}_1 + \eta_y \mathbf{F}_2),
\end{aligned}$$

and

$$\begin{aligned}
J &= x_\xi y_\eta - x_\eta y_\xi,\\
\mathbf{U} &= (\rho, \rho u, \rho v, E)^{\mathrm{T}},\\
\mathbf{F_1}(\mathbf{U}) &= (\rho u, \rho u^2 + p, \rho u v, u(E+p))^{\mathrm{T}},\\
\mathbf{F_2}(\mathbf{U}) &= (\rho v, \rho u v, \rho v^2 + p, v(E+p))^{\mathrm{T}}.
\end{aligned}$$

Here $J$ is the Jacobian determinant of the coordinate transformation, $\rho$ is the density, $u$ and $v$ are the velocity components in the respective $x$ and $y$ direction, $p$ is the pressure, and $E = \rho e + 0.5\rho(u^2 + v^2)$ is the total energy per unit volume.

For closure of the system (1), the equation of state (EOS) is required. In this paper, we use the following stiffened gas EOS

$$p = (\gamma - 1)\rho e - \gamma\pi \tag{2}$$

to model gas and water medium uniformly, where $\gamma$ and $\pi$ are fluid constants to be specified. The speed of sound $c$ and entropy $S$ can be computed by

$$\begin{aligned}
c &= \sqrt{\gamma \frac{p+\pi}{\rho}},\\
S &= \frac{p+\pi}{\rho^\gamma}.
\end{aligned}$$

In order to track the moving material interface, we adopt the level set method proposed in [1,?,?,?]. The level set equation can be written as

$$\phi_t + \mathbf{u}\cdot\nabla\phi = \phi_t + \left(\mathbf{u}\cdot\frac{\nabla\phi}{|\nabla\phi|}\right)|\nabla\phi| \triangleq \phi_t + u_n|\nabla\phi| = 0. \tag{3}$$

$\phi$is usually initialized as a signed distance function, and $u_n$ is the velocity field along the normal $\mathbf{n} = (n_x, n_y)^{\mathrm{T}}$ of $\phi$. As discussed in [6], $\phi$'s signed distance property can be maintained well as long as a suitably redefined normal velocity field is used in place of $u_n$ in (3). This velocity is called extension velocity $u_n^c$, and there are several ways of defining extension velocity in literature. We will discuss the construction of $u_n^c$ at the end of Section 3.1.2.

Eq. (3) can be rewritten as

$$\phi_t + u_n\mathbf{n}\cdot\nabla\phi = 0.$$

With $u_n = u_n^c$, it can be transformed to the coordinate system $(\xi, \eta)$ as following

$$\frac{\partial\phi}{\partial t} + u_n^c(n_x\ \xi_x + n_y\ \xi_y)\frac{\partial\phi}{\partial\xi} + u_n^c(n_x\ \eta_x + n_y\ \eta_y)\frac{\partial\phi}{\partial\eta} = 0, \tag{4}$$

where the metric terms are computed by central differences.

## 3. Numerical schemes

The present adaptive ghost fluid finite difference method contains two independent ingredients: The solvers for the Euler equations and for the level set equation and the adaptive mesh redistribution method. Th flow variables and the level set function as defined on the mesh points are solved for on a fixed mesh for a given physical time, then the mesh is redistributed by iterating an grid generation equation and simultaneously remap the numerical solutions onto the new mesh by using the interpolation method.

### 3.1. Real ghost fluid method

#### 3.1.1. One-dimensional case

To specify the ghost fluid states, we need to assign the values for density, velocity and pressure. Suppose that the fluid interface lies between $x_i$ and $x_{i+1}$ as depicted in Fig. 1. A Riemann problem is defined at the fluid interface with initial condition $\mathbf{U}_{\mathrm{L}} = \mathbf{U}_{i-1}$ and $\mathbf{U}_{\mathrm{R}} = \mathbf{U}_{i+2}$ (rather than $\mathbf{U}_i$ and $\mathbf{U}_{i+1}$ since they are thought to be polluted by numerical errors from the interface [7]) and solved with the two-shock approximate Riemann solver. Denote the resultant interfacial pressure and velocity as $p_{\mathrm{I}}$ and $u_{\mathrm{I}}$, and the densities on the left and right sides of the fluid interface as $\rho_{\mathrm{IL}}$ and $\rho_{\mathrm{IR}}$, respectively. For defining the ghost fluid states and implementing isobaric fix for medium I, $\rho_{\mathrm{IL}}$, $u_{\mathrm{I}}$, $p_{\mathrm{I}}$ are assigned to the ghost fluid points from $i + 1$ to $i + 2$ and to the real fluid point $i$. Similarly, for medium II, $\rho_{\mathrm{IR}}$, $u_{\mathrm{I}}$, $p_{\mathrm{I}}$ are assigned to the ghost fluid points from $i - 1$ to $i$ and to the real fluid point $i + 1$. After defining the ghost fluid states, we then solve (1) for each sing-medium by using the second-order HLLC scheme [26], and then solve the Hamilton–Jacobi type level set Eq. (4) by using a fifth-order WENO scheme [27] with approximate normal velocity field $u_n^c = u_{\mathrm{I}}$. The final solution $\mathbf{U}$ is obtained by combining the solutions of the two single-medium problems according to the sign of the new level set function at each mesh point.

#### 3.1.2. Two-dimensional case

In two-dimensional case there are two velocity components, and the main difficulty to implement RGFM is how to define a local Riemann problem at the interface. In virtue of the level set function, we can define its unit normal and normal velocity component at each grid point as

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|}, \tag{5}$$

$$u_n = (u, v)^{\mathrm{T}}\cdot\mathbf{n}, \tag{6}$$

Suppose medium I occupy the domain $\phi < 0$ and II the domain $\phi > 0$ (see Fig. 2). Since $\phi$ is a signed distance function, we can define a narrow band $|\phi| < 1.5\ \max(\Delta x, \Delta y)$. The procedure of defining the Riemann problem and ghost fluid for medium I is as follows. For each grid point A in the half band belonging to medium I, we search for a corresponding grid point B in the half band belonging to medium II, such that the angle made by the respective normals $\mathbf{n}_{\mathrm{A}}$ and $\mathbf{n}_{\mathrm{B}}$ is the minimum. Then a local 1D Riemann problem with the initial condition

$$\mathbf{U}_n(\mathbf{x}; 0) = \begin{cases} \mathbf{U}_n^{\mathrm{A}} & \text{if} \quad \phi < 0 \\ \mathbf{U}_n^{\mathrm{B}} & \text{if} \quad \phi > 0 \end{cases}$$

is defined, where $\mathbf{U}_n^{\mathrm{A}}$ and $\mathbf{U}_n^{\mathrm{B}}$ are the projections of the true fluid states at points A and B onto their normal directions, respectively, i.e., $\mathbf{U}_n^{\mathrm{A}} = \left(\rho^{\mathrm{A}}, \rho^{\mathrm{A}}u_n^{\mathrm{A}}, p^{\mathrm{A}}\right)^{\mathrm{T}}$ and $\mathbf{U}_n^{\mathrm{B}} = \left(\rho^{\mathrm{B}}, \rho^{\mathrm{B}}u_n^{\mathrm{B}}, p^{\mathrm{B}}\right)^{\mathrm{T}}$.

Solving this multi-medium Riemann problem by using two-shock wave approximate Riemann solver, we obtain the entropy $S_{\mathrm{IL}}$ and $S_{\mathrm{IR}}$ at the side $\phi < 0$ and the side $\phi > 0$ of the interface, respectively, the normal velocity component $u_{\mathrm{I}}$ and the pressure $p_{\mathrm{I}}$. To define the ghost fluid states, we first redefine the entropy, normal velocity component and pressure at the point A to be $S_{\mathrm{IL}}$, $u_{\mathrm{I}}$, $p_{\mathrm{I}}$. After redefining all the real points next to the interface, the ghost fluid states in the region $\phi > 0$ are obtained by solving the convection equation

$$I_\tau + \mathbf{n}\cdot\nabla I = 0 \tag{7}$$

until steady state is reached. Here $I$ is a column vector consisting of normal velocity, tangential velocity, pressure, and density or entropy. Eq. (7) is first transformed into computational domain, and then a first-order upwind scheme is used to discretize the spatial derivatives.

Similar procedure can be applied to define the Riemann problem and ghost fluid states for medium II. The corresponding convection equation is

$$I_\tau - \mathbf{n}\cdot\nabla I = 0, \tag{8}$$

which is solved in the ghost fluid region $\phi < 0$ until steady state is reached.

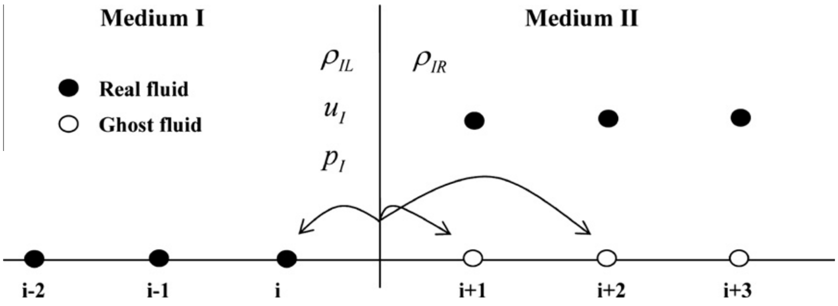


**Fig. 1.** Defining ghost fluid states and implement isobaric fix in the 1D RGFM.

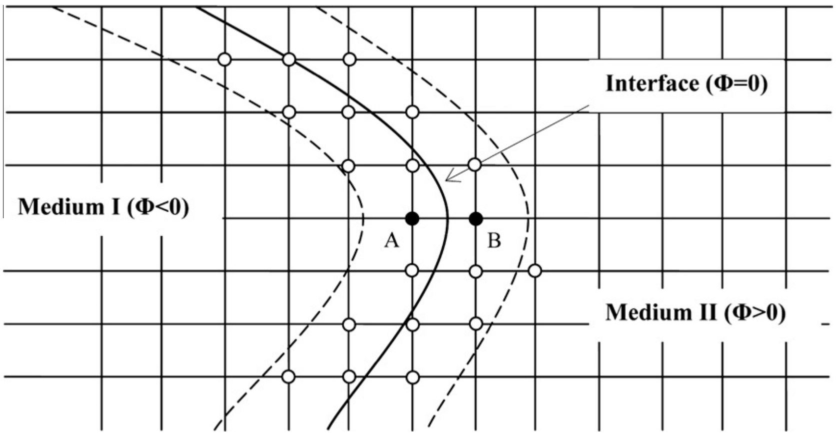


**Fig. 2.** Defining the Riemann problem for mesh points in the banded region $|\phi| < 1.5\ \max(\Delta x, \Delta y)$ in the 2D RGFM.

The above procedure of defining and solving Riemann problems gives the normal velocity $u_{\mathrm{I}}$ for all the grid points in the banded region, which is assigned to the extension velocity $u_n^c$. Outside this region, $u_n^c$ can be obtained by solving the convection Eq. (7) for the region $\phi > 1.5\ \max(\Delta x, \Delta y)$, and (8) for the region $\phi < -1.5\ \max(\Delta x, \Delta y)$, respectively.

### 3.2. Adaptive mesh redistribution

#### 3.2.1. Mesh movement

Let $(x, y)$ and $(\xi, \eta)$ be the physical and computational coordinates, respectively, and $\mathbf{x} \equiv (x, y) = (x(\xi, \eta),\ y(\xi, \eta))$ be the mesh map in two dimensions. We use a simple form of the Euler–Lagrange equation derived by Ceniceros and Hou [16]

$$\begin{aligned} &(\omega x_\xi)_\xi + (\omega x_\eta)_\eta = 0 \\ &(\omega y_\xi)_\xi + (\omega y_\eta)_\eta = 0 \end{aligned} \tag{9}$$

to provide the coordinate transformation $x = x(\xi, \eta),\ y = y(\xi, \eta)$, where $\omega$is the monitor function which generally depends on the flow solution **U** and level set $\phi$. Eq. (9) can be discretized on uniform mesh in the computational domain $\Omega_c$ and solved with the Gauss–Seidel type iteration

$$\begin{aligned} &\omega^{[v]}_{i+\frac{1}{2},j}\left(\mathbf{x}^{[v]}_{i+1,j} - \mathbf{x}^{[v+1]}_{i,j}\right) - \omega^{[v]}_{i-\frac{1}{2},j}\left(\mathbf{x}^{[v+1]}_{i,j} - \mathbf{x}^{[v+1]}_{i-1,j}\right) \\ &+ \omega^{[v]}_{i,j+\frac{1}{2}}\left(\mathbf{x}^{[v]}_{i,j+1} - \mathbf{x}^{[v+1]}_{i,j}\right) - \omega^{[v]}_{i,j-\frac{1}{2}}\left(\mathbf{x}^{[v+1]}_{i,j} - \mathbf{x}^{[v+1]}_{i,j-1}\right) = 0, \end{aligned} \tag{10}$$

where $\omega^{[v]}_{i+\frac{1}{2},j} = \frac{1}{2}\left(\omega^{[v]}_{i+1,j} + \omega^{[v]}_{i,j}\right)$ and $v$ is the index of iteration. In practical computation, it is common to use some spatial smoothing on the monitor function $\omega$ to obtain smoother meshes. We apply the following low-pass filter for the spatial smoothing

$$\begin{aligned} \omega_{i,j} \leftarrow \frac{1}{16}[&4\omega_{i,j} + 2(\omega_{i+1,j} + \omega_{i-1,j} + \omega_{i,j+1} + \omega_{i,j-1}) + (\omega_{i+1,j+1} \\ &+ \omega_{i+1,j-1} + \omega_{i-1,j+1} + \omega_{i-1,j-1})]. \end{aligned} \tag{11}$$

It is noted that each iteration is followed by the interpolation procedure in 3.2.2 for solutions **U** and $\phi$. The updated solutions $\mathbf{U}^{[v+1]}_{ij}$ and $\phi^{[v+1]}_{ij}$ are used to obtain $\omega^{[v+1]}_{ij}$. The 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.

#### 3.2.2. The solution interpolation

In each mesh iterative step, once the new mesh points are updated, we need to interpolate the solutions **U** and $\phi$ from the old mesh $\mathbf{x}^{[v]}_{i,j}$ to the new mesh $\mathbf{x}^{[v+1]}_{i,j}$. In this paper, to better approximate multidimensional effects, the CTU method with high resolution correction terms as introduced in [25] is used. The scheme is

$$\begin{aligned} q\left(\mathbf{x}^{[v+1]}_{i,j}\right) = q\left(\mathbf{x}^{[v]}_{i,j}\right) &- \left[\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{+}\Delta q_{i-\frac{1}{2},j} + \left(c^{\xi}_{i+\frac{1}{2},j}\right)^{-}\Delta q_{i+\frac{1}{2},j}\right] \\ &- \left[\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{+}\Delta q_{i,j-\frac{1}{2}} + \left(c^{\eta}_{i,j+\frac{1}{2}}\right)^{-}\Delta q_{i,j+\frac{1}{2}}\right] \\ &- \left(\tilde{f}_{i+\frac{1}{2},j} - \tilde{f}_{i-\frac{1}{2},j}\right) - \left(\tilde{g}_{i,j+\frac{1}{2}} - \tilde{g}_{i,j-\frac{1}{2}}\right) \end{aligned} \tag{12}$$

where $q$ represents **U** or $\phi$,

$$\begin{aligned} &c^{\xi}_{i+\frac{1}{2},j} = \frac{1}{2}\left(c^{\xi}_{i,j} + c^{\xi}_{i+1,j}\right), \quad c^{\eta}_{i,j+\frac{1}{2}} = \frac{1}{2}\left(c^{\eta}_{i,j} + c^{\eta}_{i,j+1}\right), \\ &\left(c^{\xi}_{i+\frac{1}{2},j}\right)^{\pm} = \frac{1}{2}\left(c^{\xi}_{i+\frac{1}{2},j} \pm \left|c^{\xi}_{i+\frac{1}{2},j}\right|\right), \quad \left(c^{\eta}_{i,j+\frac{1}{2}}\right)^{\pm} = \frac{1}{2}\left(c^{\eta}_{i,j+\frac{1}{2}} \pm \left|c^{\eta}_{i,j+\frac{1}{2}}\right|\right), \\ &\Delta q_{i-\frac{1}{2},j} = q_{i,j} - q_{i-1,j}, \quad \Delta q_{i,j-\frac{1}{2}} = q_{i,j} - q_{i,j-1}, \end{aligned}$$

and

$$\begin{aligned} c^{\xi}_{i,j} &= [\nabla\xi \cdot (\mathbf{x}^{[v]} - \mathbf{x}^{[v+1]})]_{i,j} = \frac{1}{J_{i,j}}[y_\eta(x^{[v]} - x^{[v+1]}) - x_\eta(y^{[v]} - y^{[v+1]})]_{i,j}, \\ c^{\eta}_{i,j} &= [\nabla\eta \cdot (\mathbf{x}^{[v]} - \mathbf{x}^{[v+1]})]_{i,j} = \frac{1}{J_{i,j}}[x_\xi(y^{[v]} - y^{[v+1]}) - y_\xi(x^{[v]} - x^{[v+1]})]_{i,j} \end{aligned}$$

are the moving speed of mesh points, $J_{i,j} = (x_\xi y_\eta - x_\eta y_\xi)_{i,j}$ is the Jacobian determinant of coordinate transformation. The terms in the square brackets in RHS of (12) are the donor-cell upwind contributions, while the terms in circle brackets are the CTU corrections plus the high-resolution corrections

$$\begin{aligned} \tilde{f}_{i-\frac{1}{2},j} = &-\frac{1}{2}\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{-}\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{+}\Delta q_{i,j-\frac{1}{2}} \\ &-\frac{1}{2}\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{-}\left(c^{\eta}_{i,j+\frac{1}{2}}\right)^{-}\Delta q_{i,j+\frac{1}{2}} \\ &-\frac{1}{2}\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{+}\left(c^{\eta}_{i-1,j-\frac{1}{2}}\right)^{+}\Delta q_{i-1,j-\frac{1}{2}} \\ &-\frac{1}{2}\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{+}\left(c^{\eta}_{i-1,j+\frac{1}{2}}\right)^{-}\Delta q_{i-1,j+\frac{1}{2}} \\ &+\frac{1}{2}\left|c^{\xi}_{i-\frac{1}{2},j}\right|\left(1 - \left|c^{\xi}_{i-\frac{1}{2},j}\right|\right)\mathrm{minmod}\left(\Delta q_{i-\frac{1}{2},j}, \Delta q_{i-\frac{3}{2},j}\right) \\ \tilde{g}_{i,j-\frac{1}{2}} = &-\frac{1}{2}\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{-}\left(c^{\xi}_{i-\frac{1}{2},j}\right)^{+}\Delta q_{i-\frac{1}{2},j} \\ &-\frac{1}{2}\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{-}\left(c^{\xi}_{i+\frac{1}{2},j}\right)^{-}\Delta q_{i+\frac{1}{2},j} \\ &-\frac{1}{2}\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{+}\left(c^{\xi}_{i-\frac{1}{2},j-1}\right)^{+}\Delta q_{i-\frac{1}{2},j-1} \\ &-\frac{1}{2}\left(c^{\eta}_{i,j-\frac{1}{2}}\right)^{-}\left(c^{\xi}_{i+\frac{1}{2},j-1}\right)^{-}\Delta q_{i+\frac{1}{2},j-1} \\ &+\frac{1}{2}\left|c^{\eta}_{i,j-\frac{1}{2}}\right|\left(1 - \left|c^{\eta}_{i,j-\frac{1}{2}}\right|\right)\mathrm{minmod}\left(\Delta q_{i,j-\frac{1}{2}}, \Delta q_{i,j-\frac{3}{2}}\right) \end{aligned}$$

with the slope limiter

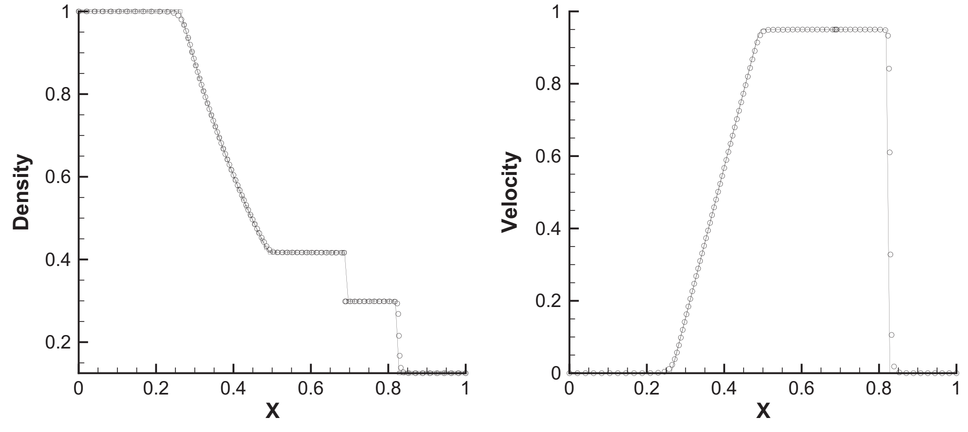
$$\text{minmod}(a,b)=\begin{cases}0, & \text{if } \text{sign}(a)\neq\text{sign}(b),\\ \text{sign}(a)\min(|a|,|b|), & \text{if } \text{sign}(a)=\text{sign}(b).\end{cases}$$

The first four terms in $\tilde{f}_{i-1/2,j}$ or $\tilde{g}_{i,j-1/2}$ are the CTU corrections and the last term is the second-order high-resolution correction. In [21], the interpolation formula is a second-order MUSCL method in dimension splitting form, while the current CTU method takes account of flow direction more fully and is more stable since it has a larger numerical domain of dependence [25].

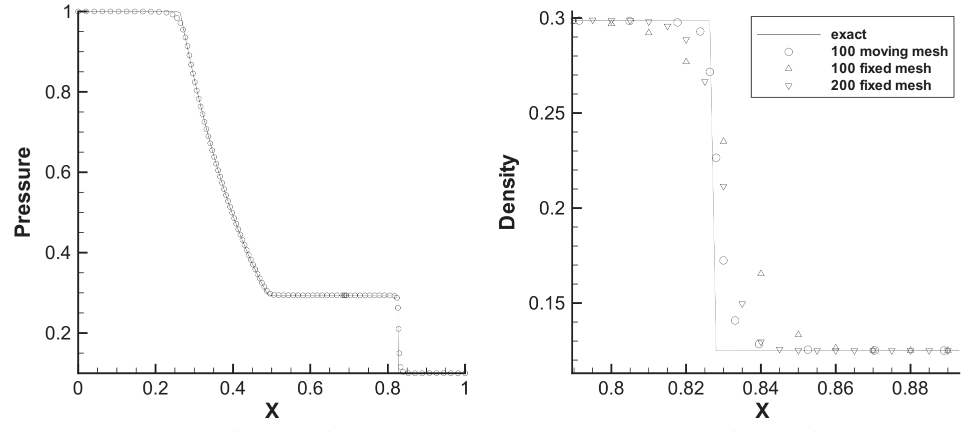
**Remark 1.** After each mesh iterative step, we can do the RGFM process, and interpolate the solutions **U** for each medium and the level set function $\phi$ onto the new mesh, then decide the fluid state and the equation of state (EOS) for each grid point based on the sign of the updated $\phi$. Since the interpolation for **U** based on RGFM does not produce extra difficulty, we need not consider whether or not some points may cross the interface. In fact, this allows the mesh to move more freely so as to provide better mesh adaptivity than in [21].

## 4. Solution procedure of RGFM on moving mesh

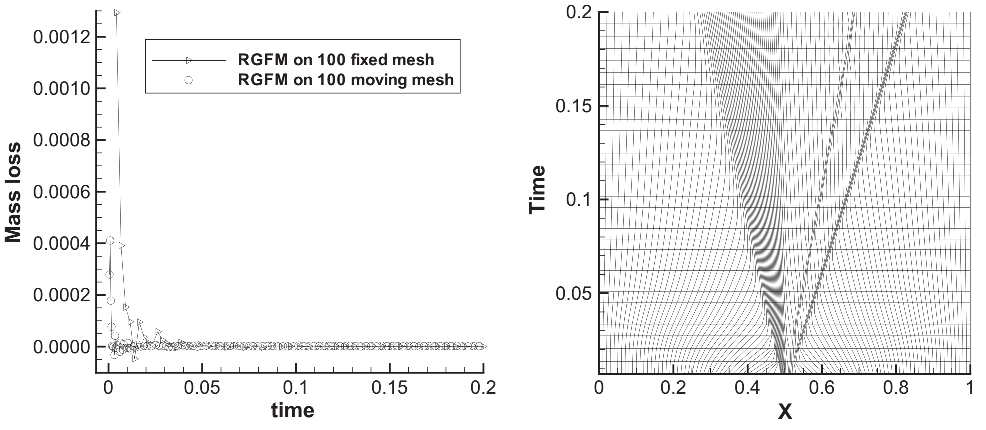
The whole solution procedure consists of two parts: one is the evolution of the fluid equations and the level set equation, the other is the mesh moving procedure. We use a second-order HLLC



A (Denstiy)    B (Velocity)



C (Pressure)    D (Density)



E (mass loss)    F (Mesh trajectory)

**Fig. 3.** Example 1. Computational results at $t = 0.2$. The solid line in A–D denotes the analytic solution. E is the comparison of total mass conservation errors caused by the adaptive RGFM on 100 mesh points and the RGFM on 100 uniform mesh points. F is the time history of the mesh distribution.

scheme [26] to solve the Euler Eq. (1) and a fifth-order WENO scheme [27] to solve the level set Eq. (4). We apply the Gauss–Seidel iteration method (10) to move the mesh, and then apply the CTU method (12) to interpolate the conservative variables for each single medium with the help of RGFM, and the level set function from the old mesh to the new mesh. The complete algorithm procedure is illustrated as follows:

Step 1: Given the initial mesh $\left\{\tilde{\mathbf{x}}^0_{i,j}\right\}$ and initial values $\mathbf{U}^0_{i,j}$ and $\phi^0_{i,j}$. Let $\tilde{\mathbf{x}}^{[0]}_{i,j} = \tilde{\mathbf{x}}^0_{i,j}$. For $v$ = 0, 1, 2, …, $L - 1$, do (1)–(3):

**Table 1**
Comparison of errors of RGFM on moving mesh and fixed mesh for example 1.

| Mesh | Time steps | $L^2$ error of density | $L^2$ error of pressure |
|---|---|---|---|
| Fixed 100 grids | 50 | 0.0191 | 0.0170 |
| Fixed 200 grids | 100 | 0.0119 | 0.0095 |
| Fixed 300 grids | 150 | 0.0089 | 0.0061 |
| Moving 100 grids | 330 | 0.0095 | 0.0066 |

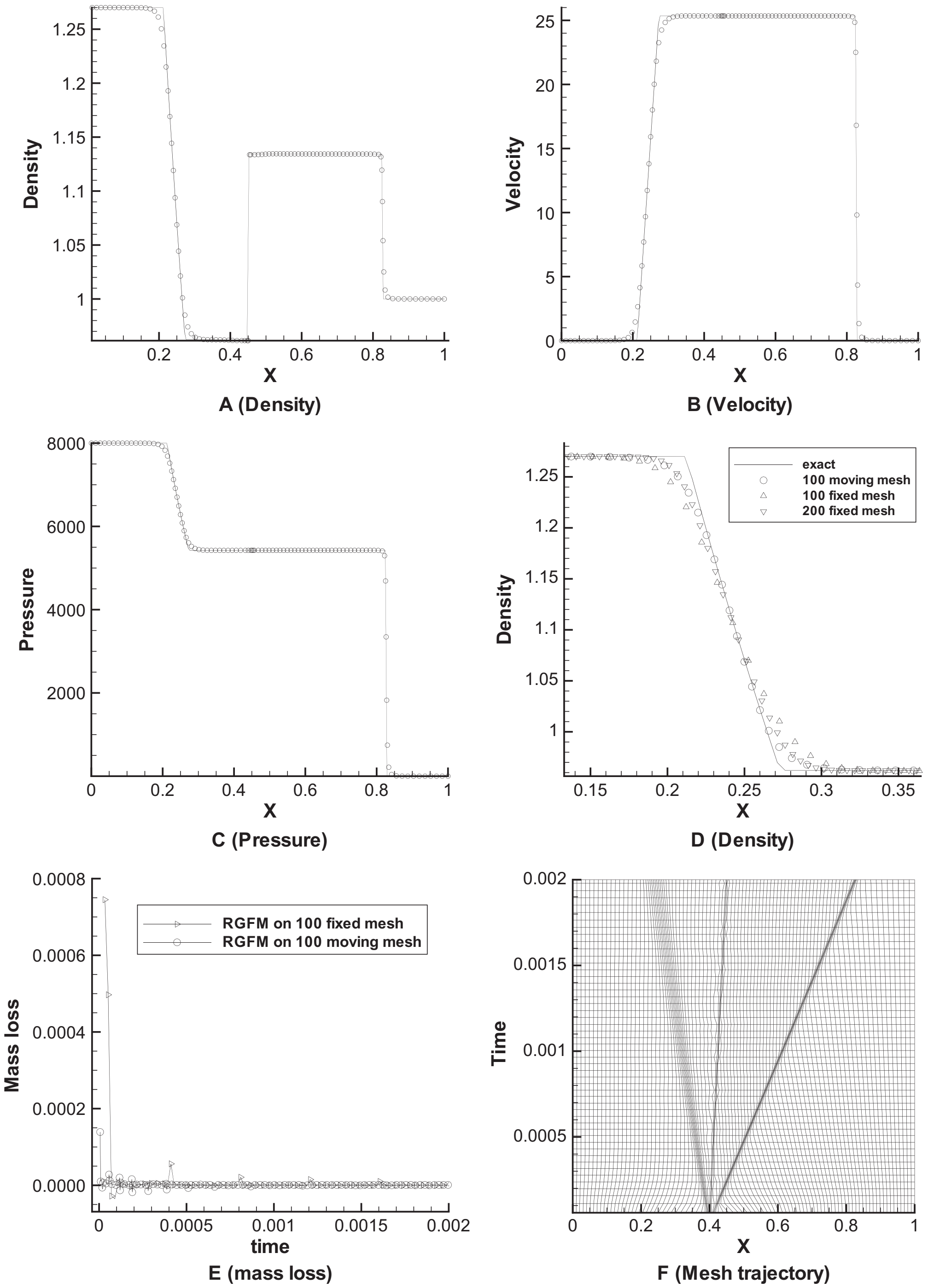


**Fig. 4.** Example 2. Computational results at $t$ = 0.002. The solid line in A–D denotes the analytic solution. D is the close-up of A around the rarefaction wave. E is the comparison of total mass conservation errors caused by the adaptive RGFM on 100 moving mesh points and the RGFM on 100 uniform mesh points. F is the time history of the mesh distribution.

**Table 2**
Comparison of errors of RGFM on moving mesh and fixed mesh for example 2.

| Mesh | Time steps | $L^2$ error of density | $L^2$ error of pressure |
|---|---|---|---|
| Fixed 100 grids | 70 | 0.0142 | 333.3388 |
| Fixed 200 grids | 130 | 0.0085 | 209.0767 |
| Fixed 300 grids | 200 | 0.0062 | 158.0870 |
| Fixed 400 grids | 260 | 0.0057 | 136.0303 |
| Moving 100 grids | 500 | 0.0058 | 135.0900 |

(1) Move mesh points $\tilde{\mathbf{x}}_{i,j}^{[\nu]}$ to $\tilde{\mathbf{x}}_{i,j}^{[\nu+1]}$.
(2) Redefine the initial values of $\mathbf{U}_{i,j}^{0}, \phi_{i,j}^{0}$ on the new mesh $\tilde{\mathbf{x}}_{i,j}^{[\nu+1]}$ according to initial condition.
(3) Let $\mathbf{x}_{i,j}^{0} = \tilde{\mathbf{x}}_{i,j}^{[\nu+1]}$.

Step 2: For physical time level $n = 1, 2, \ldots$, do (1)–(4):

(1) Solve the governing equations:
  (a) Define the ghost fluid states for each medium by using RGFM.
  (b) Solve the Euler Eq. (1) for each single medium to obtain the solution $\mathbf{U}_{i,j}^{n+1}$, and solve the level set Eq. (4) to obtain $\phi_{i,j}^{n+1}$ at new time level $t = t_{n+1}$.

(2) Redistribute the mesh iteratively: Let $\mathbf{x}_{i,j}^{[0]} = \mathbf{x}_{i,j}^{n}$, $\phi_{i,j}^{[0]} = \phi_{i,j}^{n+1}$, $\mathbf{U}_{i,j}^{[0]} = \mathbf{U}_{i,j}^{n+1}$. For $\nu = 0, 1, \ldots, L-1$, do (a)–(c):
  (a) Move mesh points $\mathbf{x}_{i,j}^{[\nu]}$ to $\mathbf{x}_{i,j}^{[\nu+1]}$.
  (b) Interpolate the level set function $\phi_{i,j}^{[\nu]}$ and the conservative variables $\mathbf{U}_{i,j}^{[\nu]}$ of each single medium from the old mesh $\mathbf{x}_{i,j}^{[\nu]}$ to the new mesh $\mathbf{x}_{i,j}^{[\nu+1]}$.

(3) Let $\mathbf{x}_{i,j}^{n+1} = \mathbf{x}_{i,j}^{[L]}, \phi_{i,j}^{n+1} = \phi_{i,j}^{[L]}, \mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^{[L]}$.
(4) If $t^{n+1} < T$, then go to (1) of step 2; otherwise output the final results and stop.

The maximal iteration number is set to be $L = 5$.

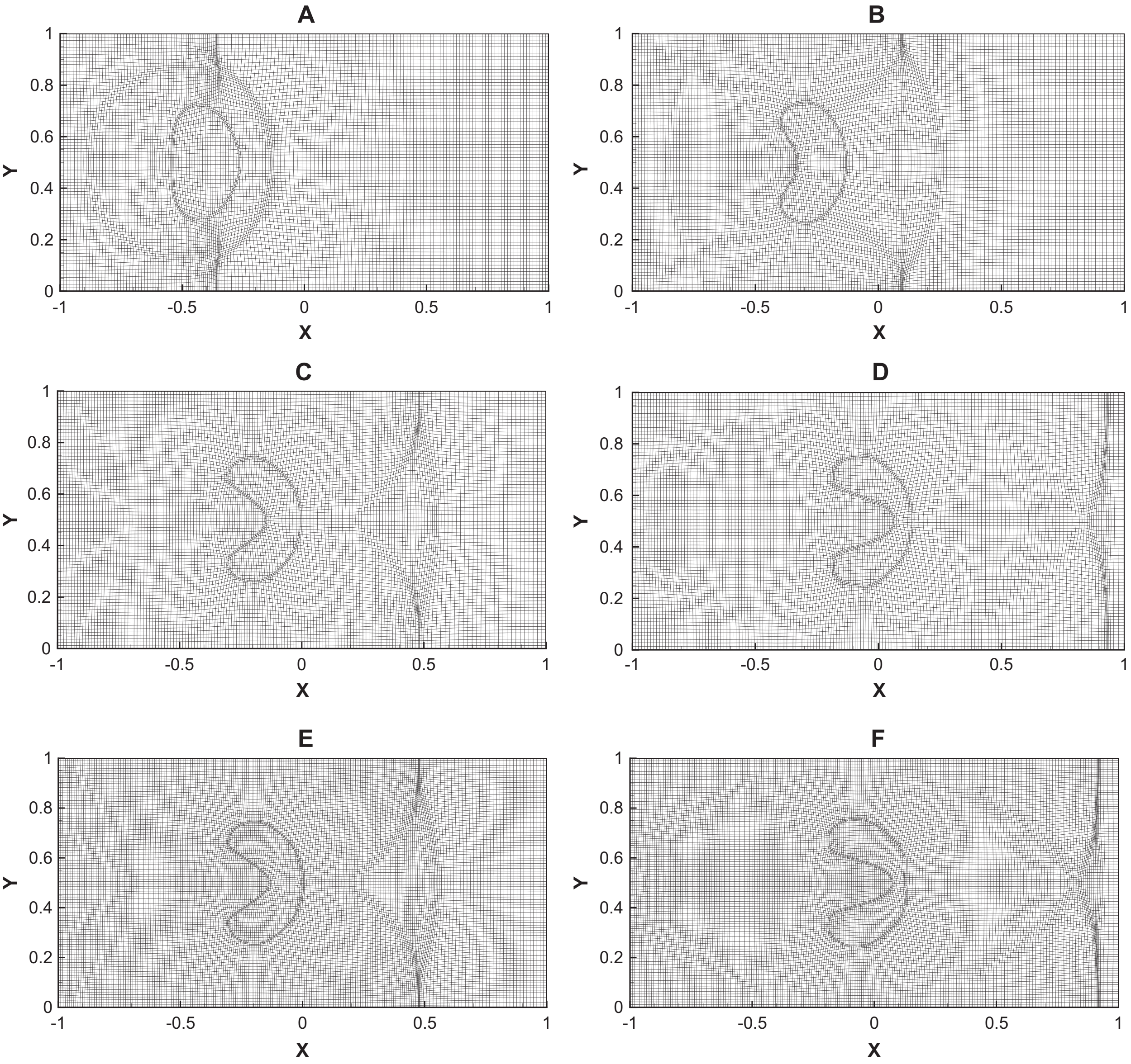


**Fig. 5.** Example 3. A, B, C, D are the adaptive meshes using the CTU interpolation at $t = 0.27, 0.56, 0.80, 1.10$, while E and F are that using the dimensional splitting interpolation at $t = 0.80, 1.10$.

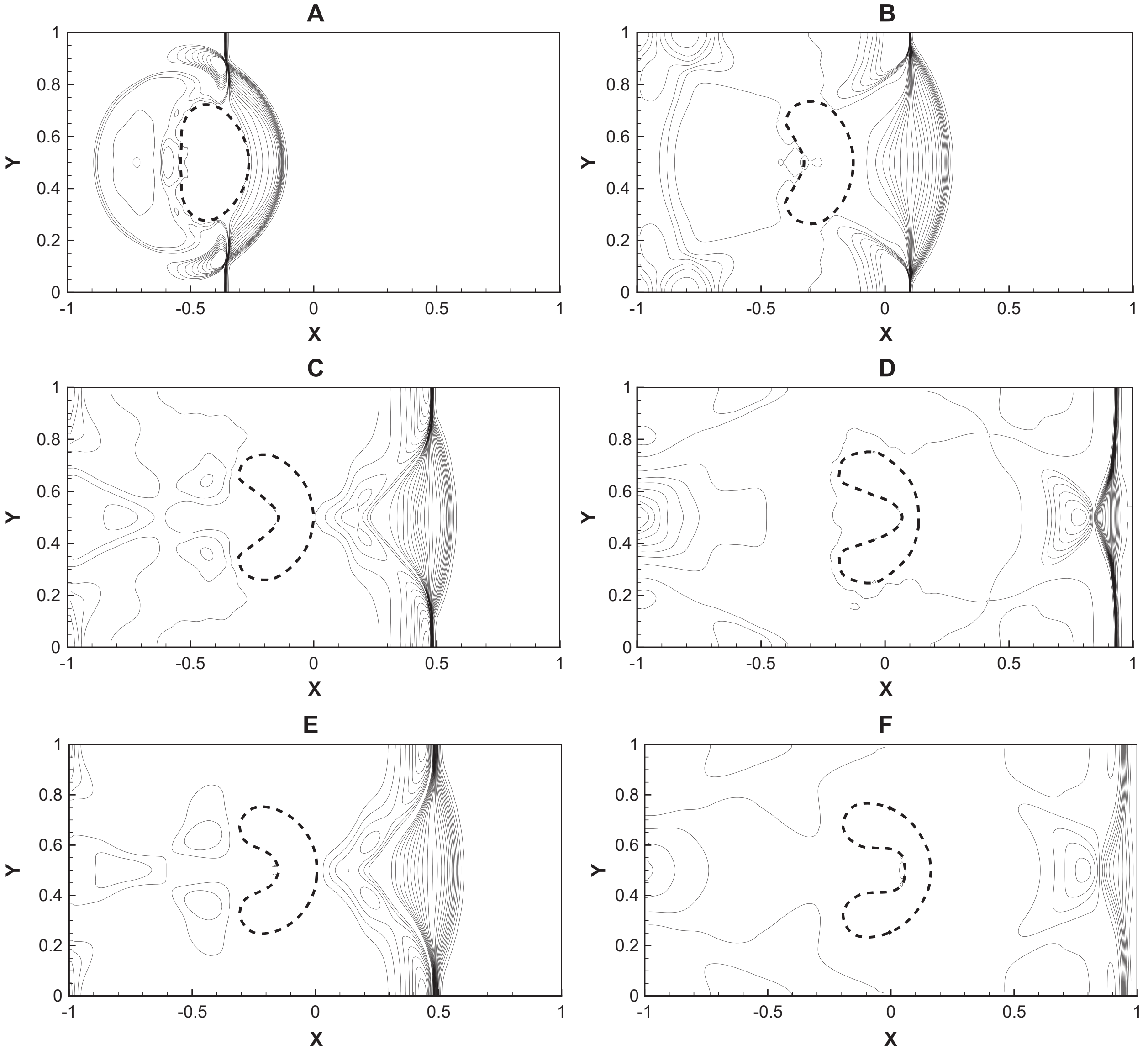


**Fig. 6.** Example 3. A, B, C, D are the pressure contours at $t = 0.27, 0.56, 0.80, 1.10$ as computed with the adaptive RGFM, and E and F are pressure contours at $t = 0.80$ and 1.10 as computed with fixed mesh RGFM. The dashed line denotes the bubble interface.

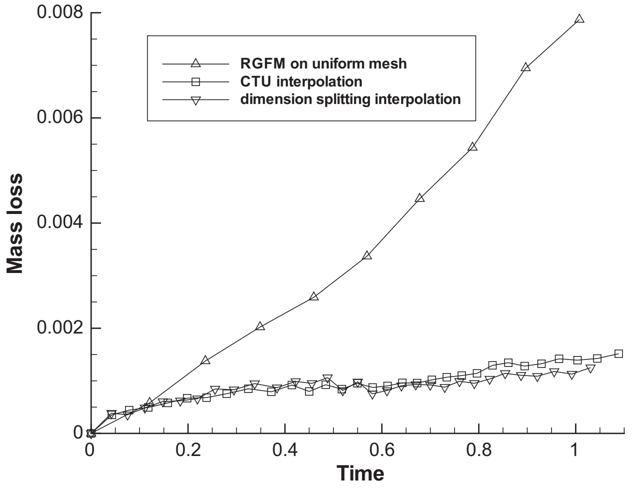


**Fig. 7.** Example 3. The recorded total mass losses of the gas bubble with time for the adaptive RGFM method with CTU interpolation, that with dimensional splitting interpolation, and the real GFM method on the uniform mesh.

## 5. Numerical examples

In this section, we give several 1D and 2D numerical examples to illustrate the efficiency of our adaptive RGFM method in decreasing numerical errors and the effectiveness of the high-resolution CTU interpolation. The first two examples are 1D two-medium Riemann problems, unless otherwise stated, the physical domain is taken as [0, 1] and is divided into 100 uniform cells, the CFL number is set to 0.8, and the monitor function in the mesh generation equation has the form

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

where $\alpha$ and $\beta$ are two free parameters for controlling the adaptivity. They are varied from case to case.

### *5.1. One-dimensional examples*

For 1D examples we will show the efficiency of the present adaptive RGFM and its capability to reduce conservative errors.

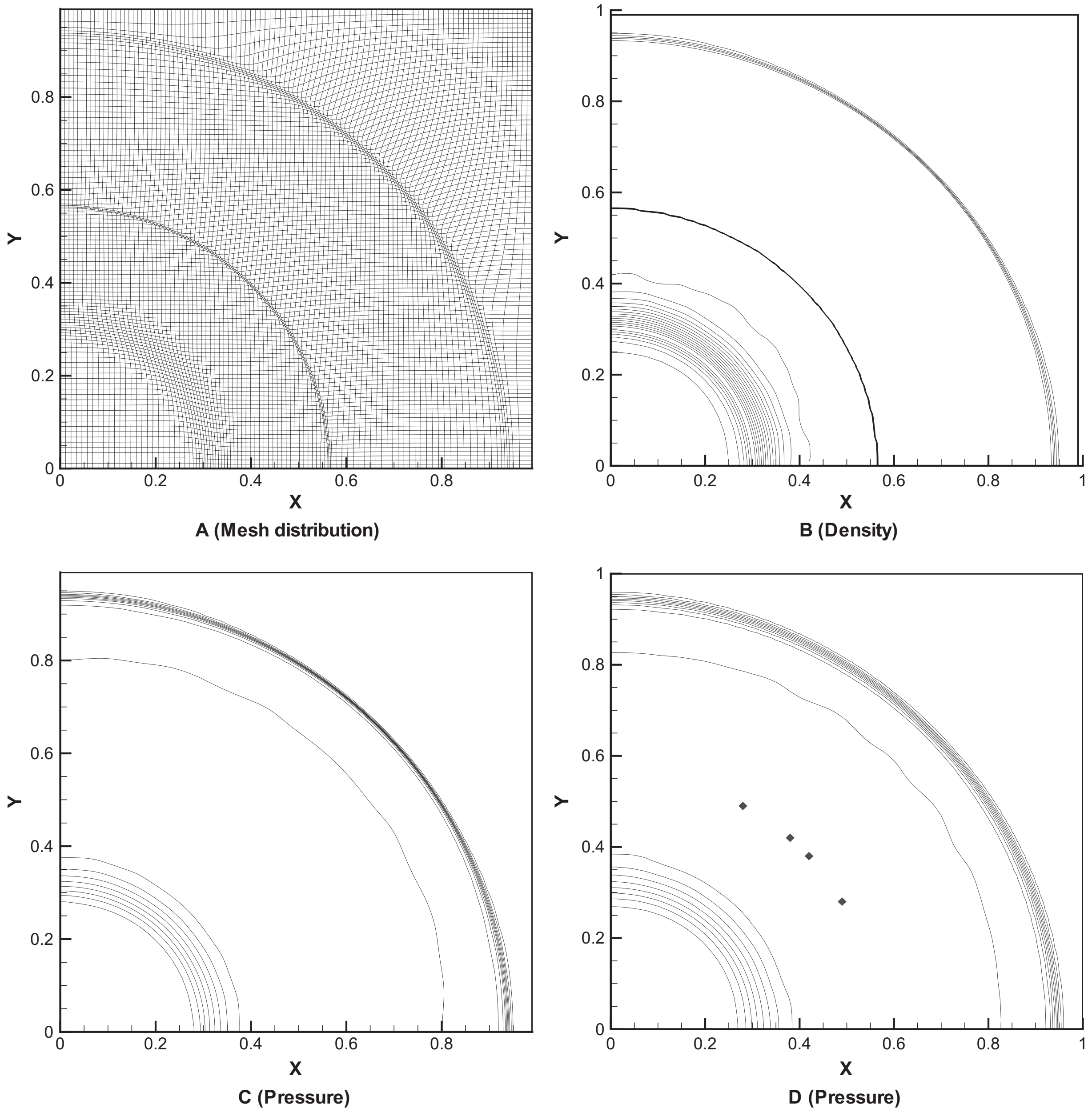
**Example 1. Gas–gas shock tube problem.** This problem is from reference [7] but we have chosen different dimensional scales. Initially, the interface is at $x = 0.5$, and the states on the left and right sides of the interface are

$$(\rho, u, p, \gamma, \pi) = \begin{cases} (1.0, 0, 1.0, 1.4, 0), & 0 \leqslant x < 0.5, \\ (0.125, 0, 0.1, 1.2, 0), & 0.5 \leqslant x \leqslant 1. \end{cases}$$

In the computation, the two free parameters are $\alpha = 100$, $\beta = 100$. The computational results at $t = 0.2$ are shown in Fig. 3, where frames A, B and C show comparison of the numerical profiles (symbol"$\circ$") for density, pressure and velocity with the exact solutions (solid line). It is seen that high resolution results for the left moving rarefaction wave, the contact discontinuity, and the right moving shock wave are obtained. Fig. 3D is the close-up view for density near the right moving shock, on which we plot the results of RGFM on 100 moving mesh points (symbol "$\circ$"), 100 fixed (symbol "$\Delta$"), and 200 fixed mesh points (symbol "$\nabla$"), respectively. It is observed that 100 moving mesh result is even more accurate than 200 fixed mesh result. Fig. 3E shows the time history of total mass conservation errors for both the adaptive RGFM and the RGFM on the uniform mesh. Here the conservative errors are measured by (3.7) in [9]. It is seen that the adaptive RGFM decreases the conservative error more quickly.

Fig. 3F shows the mesh moving history in time, where one can observe that the mesh points are automatically clustered near where large gradient variation exists. In Table 1, we present the comparison between $L^2$ errors of the RGFM on moving and fixed meshes. It is obvious that the results on 100 moving grids is nearly as accurate as that on 300 fixed grids. Although the adaptive RGFM costs double time steps to go through the time, it is still about 1.5 times more efficient due to fewer grids being used.

**Example 2. Gas–water shock tube problem.** This problem is taken from [9]. The initial interface lies at $x = 0.4$, and the states on the left and right side of the interface are



A (Mesh distribution)

B (Density)

C (Pressure)

D (Pressure)

**Fig. 8.** Example 4. A is the adaptive mesh, B and C are the contours of density and pressure on adaptive mesh, and D is the pressure contours on fixed mesh.

$$(\rho, u, p, \gamma, \pi) = \begin{cases} (1.27, 0, 8000.0, 1.4, 0), & 0 \leqslant x < 0.4, \\ (1.0, 0, 1.0, 7.15, 3309), & 0.4 \leqslant x \leqslant 1. \end{cases}$$

The two free parameters are $\alpha = 100$ and $\beta = 160$. The computational results at $t = 0.002$ are illustrated in Fig. 4. In A, B and C, the numerical density, pressure and velocity profiles of RGFM on 100 moving mesh points (symbol "o") are compared with the exact solutions, while in D, a close-up view for density is shown to compare results obtained on 100 moving meshes (symbol"$\circ$"), 100 uniform (symbol "$\Delta$") and 200 (symbol "$\nabla$") uniform meshes, respectively. It can be seen that RGFM results on 100 moving mesh points is even better than RGFM results on 200 uniform mesh points. Fig. 4E shows the time history of total mass conservation errors for both the adaptive RGFM and the fixed mesh RGFM. Again, it is seen that the the adaptive RGFM decreases the conservative error further. Fig. 4F shows the mesh trajectory, where the mesh points are more clustered near the shock and interface than near the rarefaction wave. In Table 2, we also give a comparison of the $L^2$ errors of RGFM on moving and fixed meshes. It is noted that the RGFM on 100 moving mesh is nearly as accurate as on 400 fixed meshes. After considering more time steps cost, the adaptive RGFM is still twice times more efficient than the RGFM only.

### 5.2. Two-dimensional examples

For following two examples, we will show that our adaptive mesh RGFM can increase the numerical accuracy for the interface and can effectively eliminate nonphysical oscillations. In addition, the CTU interpolation is compared with the dimensional splitting interpolation [23]. The monitor function in 2D examples takes the form

$$\omega(\mathbf{U}) = \sqrt{1 + \alpha\left(\frac{p_\xi^2 + p_\eta^2}{\max\{p_\xi^2 + p_\eta^2\}}\right) + \beta\left(\frac{\rho_\xi^2 + \rho_\eta^2}{\max\{\rho_\xi^2 + \rho_\eta^2\}}\right)}.$$

**Example 3. Planar air shock interacting with helium bubble.** This example is taken from reference [7] and is widely studied by many researchers. We do not list the lengthy initial conditions here. Initially, a Mach 1.22 planar shock in air locates at $x = -0.75$ and moves to the right in a physical domain $[-1,1] \times [0,1]$, and there is a helium bubble centered at $(-0.5, 0.5)$ with radius $r = 0.2$. The mesh point number used is $200 \times 100$, and the CFL number is set to 0.5. The two free parameters for grid adaptivity are $\alpha = 100$ and $\beta = 50$. In Fig. 5A–D show the moving mesh distributions using the CTU interpolation for four instants $t = 0.27$, 0.56, 0.80, 1.10, respectively, and frames E and F show that using the dimensional splitting interpolation for the last two instants. Small difference can be observed between the two interpolation methods. Fig. 6A–D show the pressure contours at above four instants and E and F show pressure contours at last two instants as computed by the fixed mesh RGFM. We can see that the moving mesh computation gives sharper profiles of the interface and there are no nonphysical oscillations. For simulating this flow problem, preventing mass loss is a big concern.

Fig. 7 shows the total mass loss of the helium bubble with time. Three computational results, obtained respectively by using adaptive RGFM with the CTU interpolation, that with the dimensional splitting interpolation, and the RGFM on fixed meshes, are compared. It is found that the adaptive RGFM results with CTU interpolation and the dimensional splitting interpolation have comparable mass losses, while the fixed mesh result has much larger mass loss.

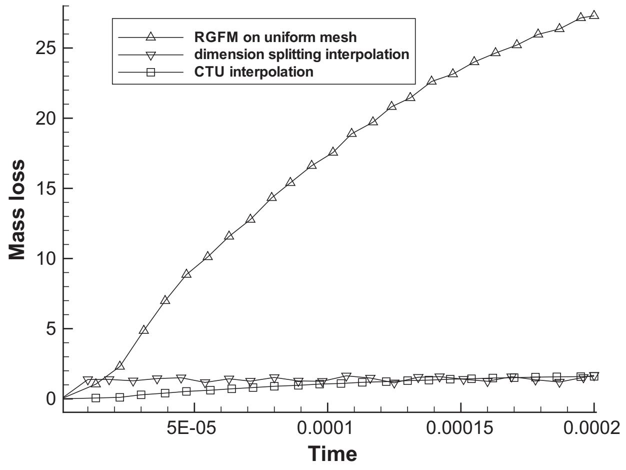


**Fig. 9.** Example 4. The recorded total mass losses of the gas bubble with time for the adaptive RGFM method with CTU interpolation, that with dimensional by dimension interpolation, and the real GFM method on the uniform mesh.

**Example 4. Two-dimensional underwater explosion.** This example is a 2D gas–water Riemann problem. The physical domain is $[0,1] \times [0,1]$ and is initially divided into $100 \times 100$ uniform meshes, the CFL number is taken as 0.5, the initial conditions are taken as

$$(\rho, u, v, p, \gamma, \pi) = \begin{cases} (1250, 0, 0, 10^9, 1.4, 0), & r < 0.5, \\ (1000, 0, 0, 10^5, 7.15, 3.309 \times 10^8), & r > 0.5. \end{cases}$$

The solution of this problem comprises of an inward moving rarefaction wave, a fast outward moving shock and a slowly outward moving contact discontinuity. We take the free parameters $\alpha = 60$ and $\beta = 100$. Fig. 8 shows the moving mesh (A), the density and pressure contours (B and C) as computed by the moving mesh RGFM and the pressure contour (D) as computed by the fixed mesh RGFM. It can be observed that by using moving mesh method, the resolutions near the interface and the shock wave are improved remarkably. The numerical abnormality near the interface in frame D is removed in frame C. We also compare the total mass loss of the gas bubble with time in Fig. 9. Again, it is found that the adaptive RGFM results with CTU interpolation and the dimensional splitting interpolation have comparable mass losses, while the fixed mesh result has much larger mass loss.

## 6. Conclusions

In this paper, we implement the real ghost fluid method in finite difference discretization on adaptive moving meshes for one- and two-dimensional problems. We show that the strategy used by previous investigators [21] to avoid grid points from crossing over the material interface is unnecessary. Thus, the adaptivity will be improved in simulating moving singularities. We also use a high-resolution corner-transport upwind method for interpolating solutions from old mesh to new one. This interpolation method should be more accurate and stable than previous dimension splitting interpolation method. The numerical examples demonstrate that the present moving mesh method is effective in increasing numerical resolution of two-fluid compressible flows and decreasing conservative errors caused by the ghost fluid method.

## Acknowledgments

This work is supported by Natural Science Foundation of China (10972230, 11021101) and State Key Program for Developing Basic Sciences (2010CB731505). Yan Ding is supported by State Key Laboratory of Scientific and Engineering Computing, Chinese Academy of Sciences.

## References

[1] Osher S, Sethian JA. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 1988;79:12–49.
[2] Sethian JA. Level set method for fluid interfaces and fast marching methods. Cambridge Unverstiy Press; 1999.
[3] Sussman M, Smereka P, Osher SJ. A level set method for computing solutions to incompressible two-phase flow. J Comput Phys 1994;114:146–59.
[4] Enright D, Fedkiw R, Ferziger J, Mitchell I. A hybrid particle level set method for improved interface capturing. J Comput Phys 2002;183:83–116.
[5] Sethian JA. Fast marching methods. SIAM Rev 1999;41(2):199–235.
[6] Wang CW, Liu TG, Khoo BC. A real ghost fluid method for the simulation of multimedium compressible flow. SIAM J Sci Comput 2006;28:278–302.
[7] Fedkiw RP, Aslam T, Merriman B, Osher S. A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). J Comput Phys 1999;152:457–92.
[8] Fedkiw RP. Coupling an Eulerian fluid calculation to a lagrangian solid calculation with the ghost fluid method. J Comput Phys 2002;175:200–24.
[9] Liu TG, Khoo BC, Yeo KS. Ghost fluid method for strong shock impacting on material interface. J Comput Phys 2003;190:651–81.
[10] Xu Liang, Liu Tiegang. Accuracies and conservation errors of various ghost fluid methods for multi-medium Riemann problem. J Comput Phys 2011;230:4975–90.
[11] Nguyen DQ, Gibou F, Fedkiw RP, Kang M. A fully conservative ghost fluid method and stiff detonation waves. In: The 12th international detonation symposium, San Diego, CA; 2002.
[12] Winslow A. Numerical solution of the quasi-linear Plisson equatiion. J Comput Phys 1967;1:149–72.
[13] Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. J Comput Phys 1982;46(3):342–68.
[14] Ren WQ, Wang XP. An iterative grid redistribution method for singular problems in multiple dimensions. J Comput Phys 2000;159:246–73.
[15] Huang W, Ren Y, Russell RD. Moving mesh methods based on moving mesh partial differential equations. J Comput Phys 1994;113:279–90.
[16] Ceniceros HD, Hou TY. An efficient dynamically adaptive mesh for potentially singular solutions. J Comput Phys 2001;172:609–39.
[17] Dvinsky AS. Adaptive grid generation from harmonic maps on Riemannian manifolds. J Comput Phys 1991;95:450–76.
[18] Li R, Tang T, Zhang P. Moving mesh method in multiple dimensions based on harmonic maps. J Comput Phys 2001;170:562–88.
[19] Eiseman PR. Adaptive grid generation. Comput Methods Appl Mech Eng 1987;64:321–76.
[20] Yuan L, Tang T. Resolving the shock-induced combustion by an adaptive mesh redistribution method. J Comput Phys 2007;224:587–600.
[21] Wang Chunwu, Tang Huazhong, Liu Tiegang. An adaptive ghost fluid finite volume method for compressible gas–water simulation. J Comput Phys 2008;227:6385–409.
[22] Tang H-Z, Tang Tao, Zhang Pingwen. An adaptive mesh redistribution method for nonlinear Hamilton–Jacobi equations in two- and three-dimension. J Comput Phys 2003;188:543–72.
[23] Tang Huazhong, Tang Tao. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. SIAM J Numer Anal 2003;41:487–515.
[24] Collela P. Multidimensional upwind methods for hyperbolic conservation laws. J Comput Phys 1990;87:171–200.
[25] Leveque Randall J. Finite-volume method for hyperbolic problems. Cambridge University Press; 2002.
[26] Toro Eleuterio F. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer; 1999.
[27] Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126:202–28.