Commun. Comput. Phys.
doi: 10.4208/cicp.201209.161010a

Vol. **10**, No. 4, pp. 785-806
October 2011

A Conservative Modification to the Ghost Fluid Method for Compressible Multiphase Flows

Wei Liu[1], Li Yuan[1] and Chi-Wang Shu[2,*]

[1] *Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing 100190, China.*
[2] *Division of Applied Mathematics, Brown University, Providence RI 02912, USA.*

Received 20 December 2009; Accepted (in revised version) 16 October 2010

Communicated by Boo Cheong Khoo

Available online 13 June 2011

Abstract. A conservative modification to the ghost fluid method (GFM) is developed for compressible multiphase flows. The motivation is to eliminate or reduce the conservation error of the GFM without affecting its performance. We track the conservative variables near the material interface and use this information to modify the numerical solution for an interfacing cell when the interface has passed the cell. The modification procedure can be used on the GFM with any base schemes. In this paper we use the fifth order finite difference WENO scheme for the spatial discretization and the third order TVD Runge-Kutta method for the time discretization. The level set method is used to capture the interface. Numerical experiments show that the method is at least mass and momentum conservative and is in general comparable in numerical resolution with the original GFM.

AMS subject classifications: 76M20, 76T99

Key words: WENO scheme, ghost fluid method (GFM), mass conservation, multiphase flow.

1 Introduction

Compressible multiphase flow problems are of great interest in applications, including the study of the stability of shock-interface interaction, underwater explosion and many others. Many modern Eulerian schemes exist for single phase flows. However, when solving multiphase flows, an unmodulated conservative shock capturing scheme may easily generate nonphysical oscillations near the material interfaces due to the smeared

[*] Corresponding author. *Email addresses:* liuwei@lsec.cc.ac.cn (W. Liu), lyuan@lsec.cc.ac.cn (L. Yuan), shu@dam.brown.edu (C.-W. Shu)

density and radical change in the equation of state (EOS) across the interface. Therefore, a special treatment of the material interface is often necessary [1,2]. There are two approaches to handle this problem: the front tracking method and the front capturing method. In the front tracking method, the interface is tracked as an internal moving boundary and a non-smeared interface can be materialized [8,9]. Front capturing method is easier to implement, usually the interface is implicitly tracked by using a level set function equation [3,20,21,24] or other representative function equation.

The ghost fluid method (GFM), developed by Fedkiw et al. [6,7], is a flexible way to treat compressible two-phase flows. The GFM captures the material interface by solving the level set equation and treats the interface as a boundary that separates a real fluid on one side and its corresponding ghost fluid on the other side. Both ghost fluid and real fluid exist at the grid cells and the problem near the interface essentially becomes two single-fluid problems. With properly defined ghost fluid, numerical oscillations are generally eliminated. The GFM is simple, easy to extend to multi-dimensions and can yield a sharp interface with little smearing. It can be used for two fluids of vastly different EOS. There are subsequently developed variants of the GFM which are capable of treating more extreme situations and finding wider applications [4,5,14–17,22,28].

One major drawback of the GFM is that it is a non-conservative method. The motivation of this paper is to reduce its conservation errors. We make a modification of the algorithm to obtain at least mass and momentum conservation. The original GFM uses the ghost fluid as the numerical solution for an interfacing cell at one side of the interface when the interface has left the cell. The main idea of our method is to track the conservative variables near the interface and use them to replace the numerical solution when the interface has moved away from the cell. The present method is still a GFM since the modification does not occur at every time step and does not affect the actual performance of the GFM in the interface cells. The modification procedure can be used on the GFM with any base scheme. In this paper we use the fifth order finite difference WENO scheme [12] for the spatial discretization and the third order TVD Runge-Kutta method [26] for the time discretization.

In Section 2, we introduce the governing equation and the equation of state. The ghost fluid method is also reviewed in this section. In Section 3, our numerical method and the algorithm are presented. Numerical examples including one dimensional and two dimensional test cases are given in Section 4. Section 5 contains concluding remarks.

2 The ghost fluid method

2.1 Governing equations

We consider both the one and two dimensional Euler equations. The two dimensional version is given as

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2+p \\ \rho uv \\ (E+p)u \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2+p \\ (E+p)v \end{pmatrix}_y = 0, \tag{2.1}$$

where ρ is the density, u and v are velocity components in the x and y directions, p is the pressure and E is the total energy (the sum of the internal energy and the kinetic energy), given by

$$E=\rho e+\frac{1}{2}\rho(u^2+v^2). \tag{2.2}$$

For a closure of the system, we use the γ-law as the equation of state.

$$\rho e = p/(\gamma-1). \tag{2.3}$$

The one-dimensional Euler equations can be obtained by setting $v=0$ and dropping the third term in Eq. (2.1).

2.2 Level set equation

We use the level set equation [3,20,21,24] to track the moving fluid interface

$$\phi_t+u\phi_x+v\phi_y=0. \tag{2.4}$$

The fluid interface is tracked as the zero level set of ϕ, which starts as the signed distance function. We use a fifth order finite difference WENO method [11] with the third order TVD Runge-Kutta time discretization to solve the level set equation (2.4). In order to keep the level set function to be (approximately) a signed distance function to control the quality of the zero level set and to define the ghost fluid values via extrapolation in the normal (to the interface) direction, we need to reinitialize it by marching

$$\phi_\tau+S(\phi_0)\left(\sqrt{\phi_x^2+\phi_y^2}-1\right)=0 \tag{2.5}$$

through the pseudo-time τ, where

$$S(\phi)=\frac{\phi}{\sqrt{\phi^2+(\Delta x)^2}} \tag{2.6}$$

is the approximate sign function. Details of this procedure can be found in [19,27]. Eq. (2.4) is solved independently from and concurrently with the Euler equations (2.1), using the velocity field obtained from the Euler equations.

2.3 The GFM algorithm

Given a level set function, we obtain two separate domains for the two fluids corresponding to the sign of the level set function, with the zero level set being the fluid interface. The GFM defines a ghost cell at every point in the computational domain (One can also define ghost points only near the interface to form the so-called narrow-banded GFM). Then at every point, we have two fluids: one is the real fluid and the other is the ghost density, momentum and energy for the other fluid. The ghost fluid values are used by the algorithm only near the interface.

Since the pressure and normal velocity are continuous across an interface without the surface tension, GFM sets the pressure and normal velocity of the ghost fluid equal to the real fluid values. Combined with the technique of isobaric fix [6,7], the density of the ghost fluid is defined by using one sided extrapolation of the entropy into the other side of the interface, as shown in Fig. 1.

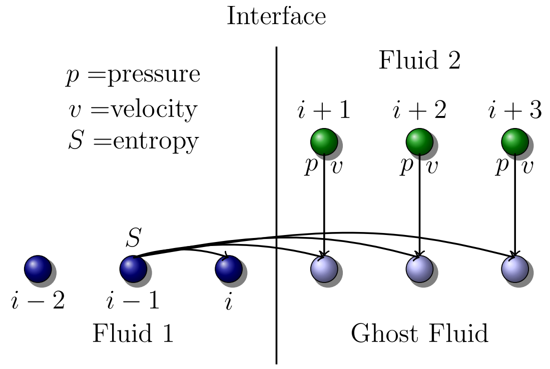

Figure 1: Ghost fluid method.

To be more specific, in the one-dimensional case, in order to solve Fluid 1 with the fifth order WENO scheme, we need to construct the ghost fluid at the nodes $i+1$, $i+2$, $i+3$ for Fluid 1, see Fig. 1. The pressure and velocity at the nodes $i+1$, $i+2$, $i+3$ are taken as those from Fluid 2. To define the value of density, the entropy of Fluid 1 at the node $i-1$ is used to perform a constant extrapolation. The isobaric fix technique is also used here by changing the value of entropy at the node i to be equal to the entropy at the node $i-1$. Similar procedure is used for solving Fluid 2.

In the multi-dimensional case, GFM defines the pressure and normal velocity component of the ghost fluid in the same way as in the one dimensional case. In order to apply the isobaric fix technique and define ghost fluid, one needs to solve a partial differential equation for a constant extrapolation in the normal direction

$$I_\tau \pm \frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla I = 0, \tag{2.7}$$

where ϕ is the level set function, I is the isobaric fix variable (which is entropy and velocity here), and τ is the pseudo-time. The sign $+$ in Eq. (2.7) is used for extrapolating entropy and velocity from the region where $\phi<0$ to the region where $\phi>0$. In this process, in order to apply the isobaric fix, we evolve Eq. (2.7) in pseudo time τ for the entropy and velocity while keeping its value in the region $\phi<-1.5\Delta x$ unchanged. Effectively, this is to extrapolate the value of entropy from the region $\phi<-1.5\Delta x$ to the region $\phi>-1.5\Delta x$. The sign $-$ in Eq. (2.7) is used for the extrapolation in the other direction. Note that we only need to solve Eq. (2.7) for several pseudo-time steps.

The ghost velocity is defined by setting its normal component to be the local normal velocity and its tangential component to be the extrapolated velocity vector projected onto the local tangential direction [6]. After defining the ghost fluid at every point in the computational domain, we can solve each fluid separately. That is, we solve two single fluid problems at all points (in effect the values of the ghost fluid are needed only near the interface), and then update the level set function. Finally, using the sign of the level set function we can decide which is the real solution at each point.

The GFM is very simple to implement and can be used on any type of EOS. The method is also easy to implement in multi-dimensions.

3 Conservative modification to GFM

3.1 Conservative error near the interface

The GFM is not a conservative method because it solves two single fluid problems instead of the two-phase problem and the numerical flux is no longer single-valued at the interface. In order to see more clearly when and how the conservation error occurs, we implement the GFM to a shock tube problem and track the total mass conservation error.

We use the well-known shock tube Riemann problem of Sod, with two different gases. The initial condition is

$$(\rho,v,p,\gamma)=\begin{cases} (1.0\text{ kg/m}^3,0\text{ m/s},1\times10^5\text{ Pa},1.4), & \text{for } x\le0.5, \\ (0.125\text{ kg/m}^3,0\text{ m/s},1\times10^4\text{ Pa},1.2), & \text{for } x>0.5. \end{cases} \tag{3.1}$$

We monitor the total mass conservation error in the computational domain at every time step and plot the result in Fig. 2.

The conservation error consists of two parts. The first part is from the non-uniqueness of the numerical flux for the cell which contains the interface, due to the GFM algorithm. The GFM treats the problem as two single-fluid problems separately at every time step and the numerical flux at the cell boundary near the interface is not unique because it is computed from two separate single-fluid problems. This error occurs at every time step. The second part of the error is from the ghost fluid replacement. When the interface moves out of the cell, GFM directly sets the computational values of the cell to be that of the ghost fluid defined before. This part of the error does not occur at every time step — it occurs only at the time step when the material interface moves out of the interfacing cell. From Fig. 2 we can see, for this problem, these two parts of the errors have opposite

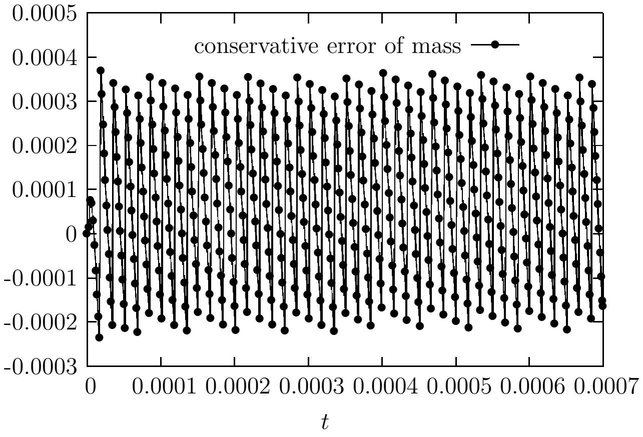

Figure 2: Mass conservation error of the two-phase Sod's shock tube problem using GFM.

signs and they counteract with each other, resulting in a total conservation error which oscillates around zero and does not become big.

Since we aim for a slight modification to the GFM, we do not want to do anything to the first part of the conservation error. The idea is to change the ghost fluid replacement procedure to make the second part of the error cancel with the first part. Notice that the second part of the error does not occur at every time step, hence our modification is not activated at every time step. Also, this modification does not affect the actual performance of a GFM in the interfacing cells since it is applied only after the interface has moved out of the interfacing cell. The computational method is thus still a GFM essentially.

3.2 Conservative modification procedure

We would like to change the second part of the error by using a more accurate value of the conservative variables rather than directly using the values of the ghost fluid when the interface moves out of a cell. We define a unique numerical flux from the two single-fluid problems at each cell boundary near the interface, and use these unique fluxes to update and obtain values of the conservative variables which are usually distinct from values of either fluids. We denote these values as from "Fluid e". When the interface moves out of a cell, we will use the values of "Fluid e" to be the numerical solution. When we track the conservation errors, we also use the values of "Fluid e" as the numerical solution for the cell which contains the interface. It should be noted that these values of the conservative variables of "Fluid e" are however *not* immediately used during the GFM evolution.

In the one dimensional case, the modification procedure is to track the more accurate values of the conservative variables in "Fluid e" for the cell which contains the interface. If the interface does not move out of this cell during the current time step, we define the left numerical flux as that of Fluid 1 (which is located to the left of the interface) and the right numerical flux as that of Fluid 2. The single valued numerical flux is calculated more carefully when the interface moves across the cell boundary during the current time step.

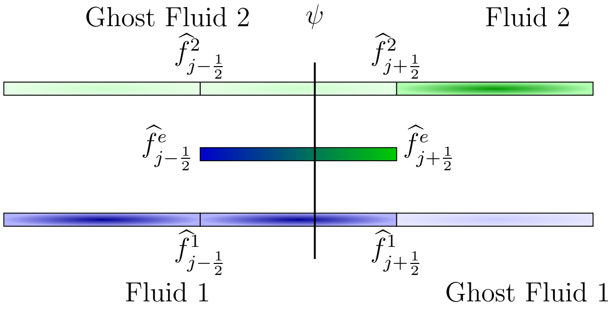
Ghost Fluid 2
ψ
Fluid 2
$\widehat{f}^2_{j-\frac{1}{2}}$
$\widehat{f}^2_{j+\frac{1}{2}}$
$\widehat{f}^e_{j-\frac{1}{2}}$
$\widehat{f}^e_{j+\frac{1}{2}}$
$\widehat{f}^1_{j-\frac{1}{2}}$
$\widehat{f}^1_{j+\frac{1}{2}}$
Fluid 1
Ghost Fluid 1

Figure 3: Conservative modification.

Specifically, we would like to track more accurately the values of the conservative variables in the cell which contains the interface, denoted by I_j in Fig. 3. The $\psi(t)$ is the interface position. If the interface position $\psi(t)$ does not cross the cell boundary from t^n to t^{n+1}, we can define a unique numerical flux from the two single-phase problems as follows:

$$\widehat{f}^e_{j-\frac{1}{2}}=\widehat{f}^1_{j-\frac{1}{2}}, \qquad \widehat{f}^e_{j+\frac{1}{2}}=\widehat{f}^2_{j+\frac{1}{2}}. \tag{3.2}$$

On the other hand, if the interface position $\psi(t)$ does cross the cell boundary $x_{j+\frac{1}{2}}$ at the instant t^* for $t^n<t^*<t^{n+1}$. We use linear interpolation to obtain this instant when the interface passes by the cell boundary, so

$$\begin{aligned} t^* &= \frac{\psi(t^*)-\psi(t^n)}{\psi(t^{n+1})-\psi(t^n)}\Delta t+t^n \\ &= \frac{x_{i+\frac{1}{2}}-\psi(t^n)}{\psi(t^{n+1})-\psi(t^n)}\Delta t+t^n. \end{aligned} \tag{3.3}$$

Then we define the numerical flux at $x_{j+\frac{1}{2}}$ as

$$\begin{aligned} \widehat{f}^e_{j+\frac{1}{2}} &= \frac{(t^*-t^n)\widehat{f}^2_{j+\frac{1}{2}}+(t^{n+1}-t^*)\widehat{f}^1_{j+\frac{1}{2}}}{t^{n+1}-t^n} \\ &= \frac{(x_{j+\frac{1}{2}}-\psi(t^n))\widehat{f}^2_{j+\frac{1}{2}}+(\psi(t^{n+1})-x_{j+\frac{1}{2}})\widehat{f}^1_{j+\frac{1}{2}}}{\psi(t^{n+1})-\psi(t^n)}. \end{aligned} \tag{3.4}$$

These single valued fluxes $\widehat{f}^e_{j+\frac{1}{2}}$ are used to obtain (presumably more accurate) values of the conservative variables, denoted as "Fluid e", which may be distinct from those of either fluid. In actual implementation, we only need to track the values of "Fluid e" in several cells near the interface. Since the unique numerical flux $\widehat{f}^e_{j+\frac{1}{2}}$ is defined as one of the numerical fluxes of single fluid problems (defined by the GFM) away from the interface, accuracy is not affected.

Since we are using the finite difference WENO method for the spatial discretization, this modification procedure can be extended to higher dimension straightforwardly. In the two dimensional case, the conservative modification is done dimension by dimension. Similar to the one dimensional case, the interface position in the x-direction is known at each time step by the level set function. We define the crossing instant t_x^* by using linear interpolation and then define the unique numerical flux in the x-direction in the same way as in the one-dimensional case. The procedure is likewise for the y-direction.

3.3 Algorithm

For spatial discretization, we use the fifth order finite difference WENO method in [12]. Then the semidiscrete scheme is written as

$$U_t = L(U) \tag{3.5}$$

which is discretized in time by the third order TVD Runge-Kutta method [25,26]

$$U^{(1)} = U^n + \Delta t L(U^n), \tag{3.6a}$$

$$U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)}), \tag{3.6b}$$

$$U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}). \tag{3.6c}$$

Assuming that the numerical solution at $t = t^n$ is known, and we would like to compute the solution at the next time step $t = t^{n+1}$, the algorithm is as follows:

1. Compute the time step size based on the CFL condition.
2. Solve the level set equation (2.4) and the re-initialization equation (2.5) by using the fifth order finite difference WENO scheme to obtain the new interface location.
3. Define the ghost fluid at every cell using the isobaric fix described in Section 2.3.
4. Solve the flow field for Fluid 1 using the WENO scheme.
5. Solve the flow field for Fluid 2 using the WENO Scheme.
6. Compute the single valued numerical flux described in Section 3.2 and the modified flow field for "Fluid e".
7. Obtain the numerical solution according to the new location of the interface, using the results of Steps 4 and 5.
8. Repeat Steps 3 through 7 for each stage of the Runge-Kutta time discretization given in (3.6).

9. Using the flow field for "Fluid e" obtained in Step 6 to replace the numerical solution in the cells which do not contain the interface.

Remark 3.1. The difference between our algorithm and the original GFM is the additional Steps 6 and 9. Note that we do not use the flow field for "Fluid e" obtained in Step 6 immediately. These "conservative modifications" get activated at Step 9 when the interface has just moved out of the relevant cell, so the modification does not change the GFM algorithm at the interface, hence retaining the advantages of GFM such as eliminating spurious oscillations near the interface.

Remark 3.2. In the one dimensional case, Step 9 does not need to be carried out at every time step. It is needed only when the interface moves into a new cell.

4 Numerical examples

In the following numerical examples, the "fully conservative modification" refers to the numerical method which is conservative in mass, momentum and energy, while the "partially conservative modification" means that the numerical method for this example is conservative in mass and momentum but not in energy (that is, our conservative modification is not applied to the energy equation), to retain better stability of the original GFM. The conservation error of the total mass at t_n is defined as the difference between total mass at t_n and the initial total mass (t_0) by adding the inflow mass and taking out the outflow mass from t_0 to t_n

$$error(t_n)=\sum_i \rho_i^n \Delta x-\sum_i \rho_i^0 \Delta x-\sum_{m=0}^{n-1}(\rho_l^m u_l^m-\rho_r^m u_r^m)\Delta t_m, \tag{4.1}$$

where ρ_l^m and u_l^m are the inflow density and velocity on the inflow boundary of the domain at time t_m, and ρ_r^m and u_r^m are the outflow density and velocity on the outflow boundary of the domain at time t_m.

In the two dimensional case, the conservation error is defined in a similar way.

Example 4.1. This is the well-known shock tube problem of Sod, with two different species. We solve this Riemann problem with a 1m domain and the following initial conditions:

$$(\rho,v,p,\gamma)=\begin{cases}(1.0\,\text{kg/m}^3, 0\,\text{m/s}, 1\times 10^5\,\text{Pa}, 1.4), & \text{for } x\le 0.5,\\ (0.125\,\text{kg/m}^3, 0\,\text{m/s}, 1\times 10^4\,\text{Pa}, 1.2), & \text{for } x>0.5.\end{cases} \tag{4.2}$$

We run the original GFM and the partial conservative modification to a final time of 0.0007s, using a uniform 200 cells mesh. The results are plotted on top of the exact solution in Fig. 4. There are slight overshoots near the interface in our partial conservative numerical results. However, comparing with traditional conservative methods, the

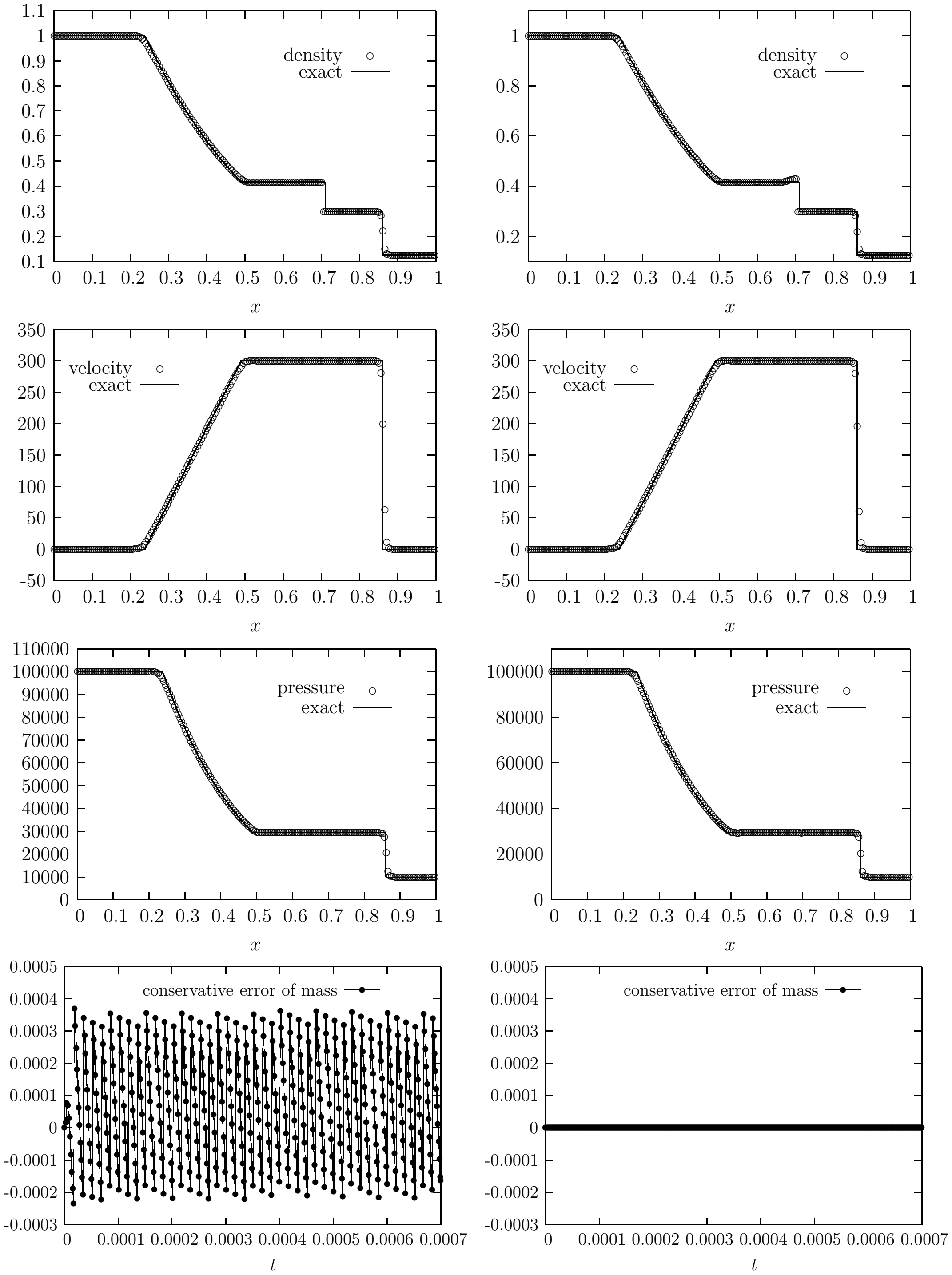

Figure 4: Example 4.1. Left: Result of GFM; Right: Result of GFM with partial conservative modification. CFL=0.4.

overshoots here are more localized and much milder. For stable interface flows, the overshoots will also become smaller under mesh refinement. From the last picture in Fig. 4, we can see our modification is conservative for mass. The conservative error of the momentum is also the same, which we do not plot here.

Example 4.2. This is a problem of a shock wave refracting at an air-helium interface with a reflected rarefaction wave. For this test, we use a 1m domain. A right going shock is located at 0.05 and an interface is at 0.5. The left, middle, and right states are defined as:

$$\begin{aligned}&(\rho,v,p,\gamma)\\ =&\left\{\begin{array}{llllll}(1.3333\,\mathrm{kg/m^3}, & 0.3535\sqrt{10^5}\,\mathrm{m/s}, & 1.5\times 10^5\,\mathrm{Pa}, & 1.4), & \text{for} & x\le 0.05,\\ (\quad 1.0\,\mathrm{kg/m^3}, & 0\,\mathrm{m/s}, & 1\times 10^5\,\mathrm{Pa}, & 1.4), & \text{for} & 0.05\le x\le 0.5,\\ (0.1379\,\mathrm{kg/m^3}, & 0\,\mathrm{m/s}, & 1\times 10^5\,\mathrm{Pa}, & 1.67), & \text{for} & x>0.5.\end{array}\right.\end{aligned} \tag{4.3}$$

We run the original GFM and the fully conservative modification to a final time of 0.0012s, using a uniform 200 cells mesh. The result with the GFM method from [6] and our modified scheme are shown in the Fig. 5. Again, there are some overshoots for the fully conservative modification, especially in density, near the material interface, but these overshoots are localized and limited in their strength. In fact, the fully conservative modification is in better agreement with the exact solution for the shock, as can be seen clearly in the plots of velocity and pressure.

Example 4.3. This example is similar to Example 4.2, except that here we increase the strength of the shock. For this test, we use a 1m domain. A right going shock is located at 0.05 and an interface is at 0.5. The left, middle, and right states are defined as:

$$\begin{aligned}&(\rho,v,p,\gamma)\\ =&\left\{\begin{array}{llllll}(4.3333\,\mathrm{kg/m^3}, & 3.2817\sqrt{10^5}\,\mathrm{m/s}, & 1.5\times 10^6\,\mathrm{Pa}, & 1.4), & \text{for} & x\le 0.05,\\ (\quad 1.0\,\mathrm{kg/m^3}, & 0\,\mathrm{m/s}, & 1\times 10^5\,\mathrm{Pa}, & 1.4), & \text{for} & 0.05\le x\le 0.5,\\ (0.1379\,\mathrm{kg/m^3}, & 0\,\mathrm{m/s}, & 1\times 10^5\,\mathrm{Pa}, & 1.67), & \text{for} & x>0.5.\end{array}\right.\end{aligned} \tag{4.4}$$

We again run the original GFM and the fully conservative modification to a final time of 0.0005s, using a uniform 200 cells mesh. The results with the GFM from [6] and the conservative results with our method are shown in Fig. 6. The comparison is similar to that for the previous example.

Example 4.4. This is a gas-water shock tube problem. We solve this Riemann problem with a 1m domain and the following initial conditions:

$$(\rho,v,p,\gamma)=\left\{\begin{array}{ll}(1270\,\mathrm{kg/m^3},0\,\mathrm{m/s},8\times 10^8\,\mathrm{Pa},1.4), & \text{for}\ \ x\le 0.5,\\ (1000\,\mathrm{kg/m^3},0\,\mathrm{m/s},1\times 10^5\,\mathrm{Pa},7.15), & \text{for}\ \ x>0.5.\end{array}\right. \tag{4.5}$$

We run the code to a final time of 0.00016s, using a uniform 200 cells mesh. In this example, in order to minimize the adverse effects of the conservative modification, we perform

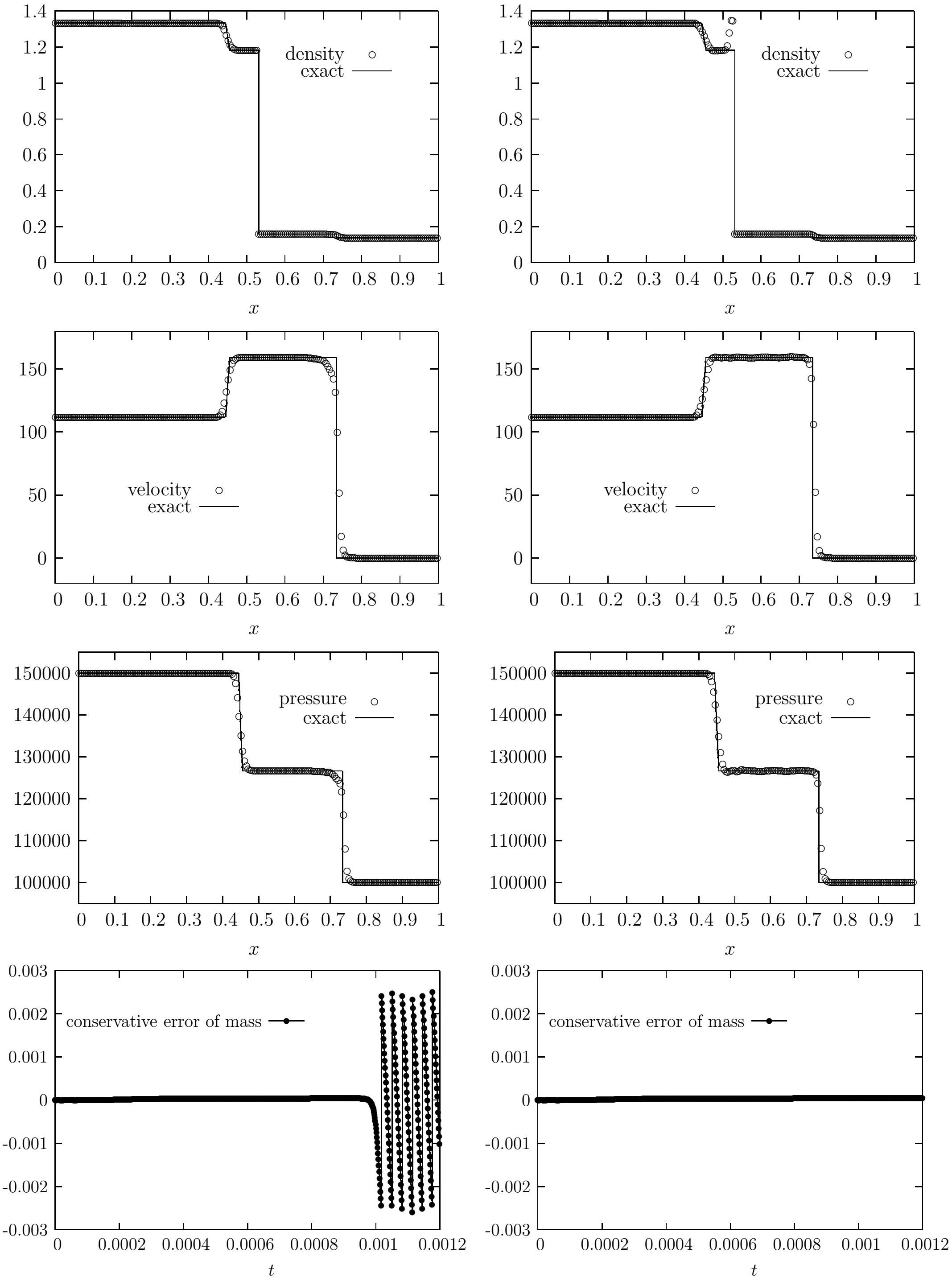

Figure 5: Example 4.2. Left: Result of GFM; Right: GFM with fully conservative modification. CFL=0.4.

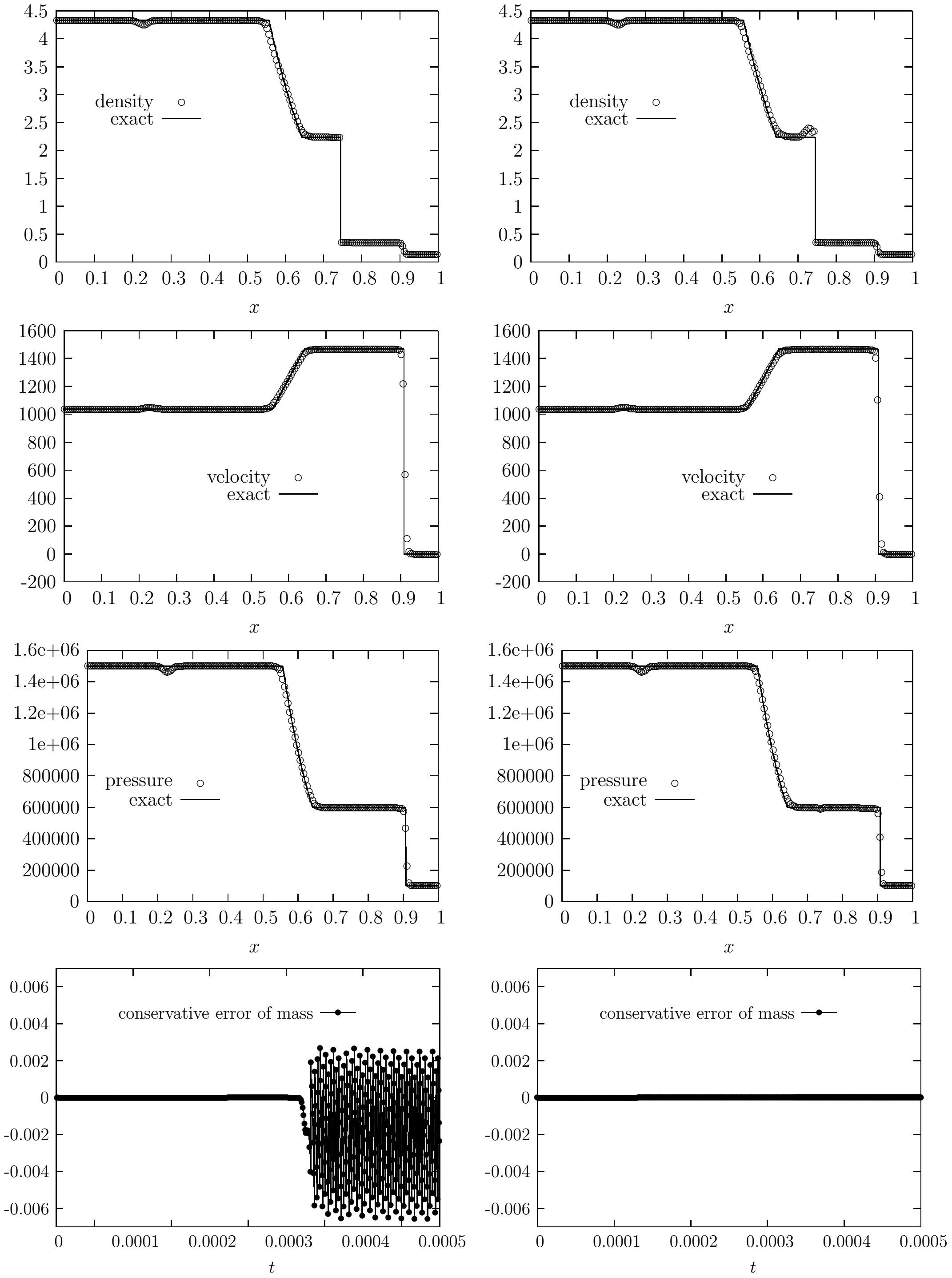

Figure 6: Example 4.3. Left: Result of GFM; Right: GFM with fully conservative modification. CFL=0.4.

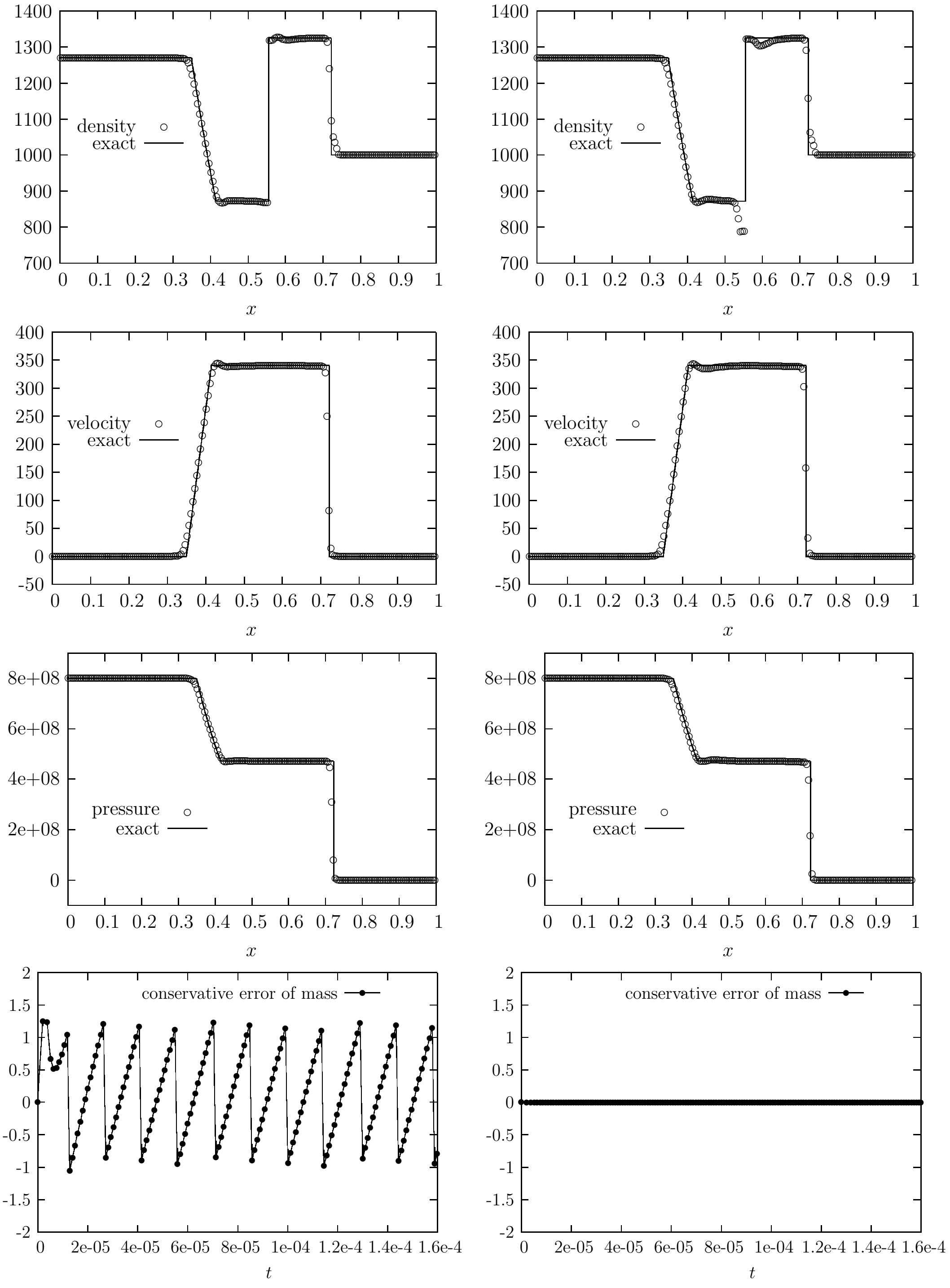

Figure 7: Example 4.4. Left: Result of GFM; Right: GFM with partially conservative modification. CFL=0.4.

only partially the conservative modification procedure to density and momentum, but not to energy. As a result, our modified schemes are conservative in mass and momentum but not conservative in energy. The results with the GFM from [6] and the partially conservative results with our method are shown in Fig. 7. The comparison is similar to those in previous examples.

Example 4.5. This is another gas-water shock tube problem with stronger pressure jumps. We solve this Riemann problem with a 1m domain and the following initial conditions:

$$(\rho,v,p,\gamma)=\begin{cases} (1630\ \mathrm{kg/m^3},0\ \mathrm{m/s},7.81\times10^9\ \mathrm{Pa},1.4), & \text{for } x\le 0.5, \\ (1000\ \mathrm{kg/m^3},0\ \mathrm{m/s},1\times10^5\ \mathrm{Pa},7.15), & \text{for } x>0.5. \end{cases} \tag{4.6}$$

We run the code to a final time of 0.00016s, using a uniform 200 cells mesh. Similar to the previous example, we perform only partially the conservative modification procedure to density and momentum. The results with the GFM from [6] and the partially conservative results with our method are shown in Fig. 8. We again observe quite satisfactory results for our partially conservative scheme.

Example 4.6. This is the case 1 in [17]. The problem simulate a strong shock impacting on a gas-gas interface. The interface is initially located at $x=0.4$ and the shock is at $x=0.3$. The initial condition is:

$$(\rho,v,p,\gamma)=\begin{cases} (0.3884,\ 27.1123\sqrt{(}10^5),\ 1.0\times10^7,\ \frac{5}{3}), & \text{for } x\le 0.3, \\ (0.1,\ 0,\ 1\times10^5,\ \frac{5}{3}), & \text{for } 0.3<x<0.4, \\ (1,\ 0,\ 1\times10^5,\ 1.4), & \text{for } x>0.4. \end{cases} \tag{4.7}$$

For this example and the next example, when doing modification, we do not only modify the interfacing cell but also the cells near the interface at the inflow side. We run the code to a final time of $t=0.0001$, using a uniform 200 cells mesh. Similar to the previous example, we perform only partially the conservative modification procedure to density and momentum. The results with the GFM from [6] and the partially conservative results with our method are shown in Fig. 9. Again, our partially conservative results are quite satisfactory.

Example 4.7. This is another gas-gas shock tube problem with stronger shock impacting on the interface with a critical condition. It is the case 2 in [17]. The initial condition are chosen so that there are no reflected wave at the interface. The initial interface is at $x=0.2$. We solve this Riemann problem with the following initial conditions:

$$(\rho,v,p,\gamma)=\begin{cases} (3.176,\ 9.435,\ 100,\ \frac{5}{3}), & \text{for } x\le 0.2, \\ (1.0,\ 0,\ 1.0,\ 1.2), & \text{for } x>0.2. \end{cases} \tag{4.8}$$

We run the code to a final time of 0.05, using a uniform 200 cells mesh. We perform only partially the conservative modification procedure to density and momentum. The results with the GFM from [6] and the partially conservative results with our method are shown in Fig. 10. Our partially conservative results are found to be satisfactory once again.

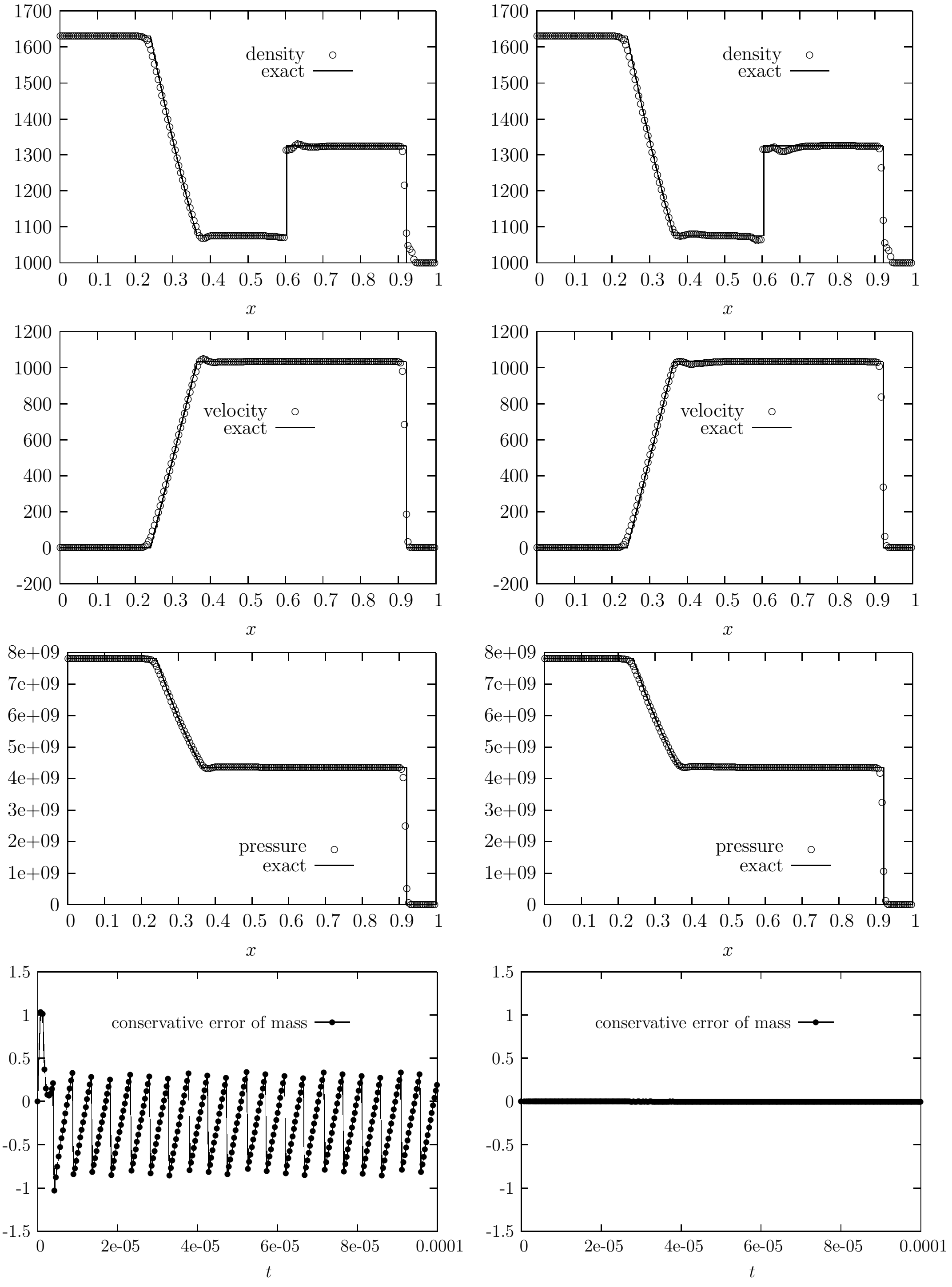

Figure 8: Example 4.5. Left: Result of GFM; Right: GFM with partially conservative modification. CFL=0.4.

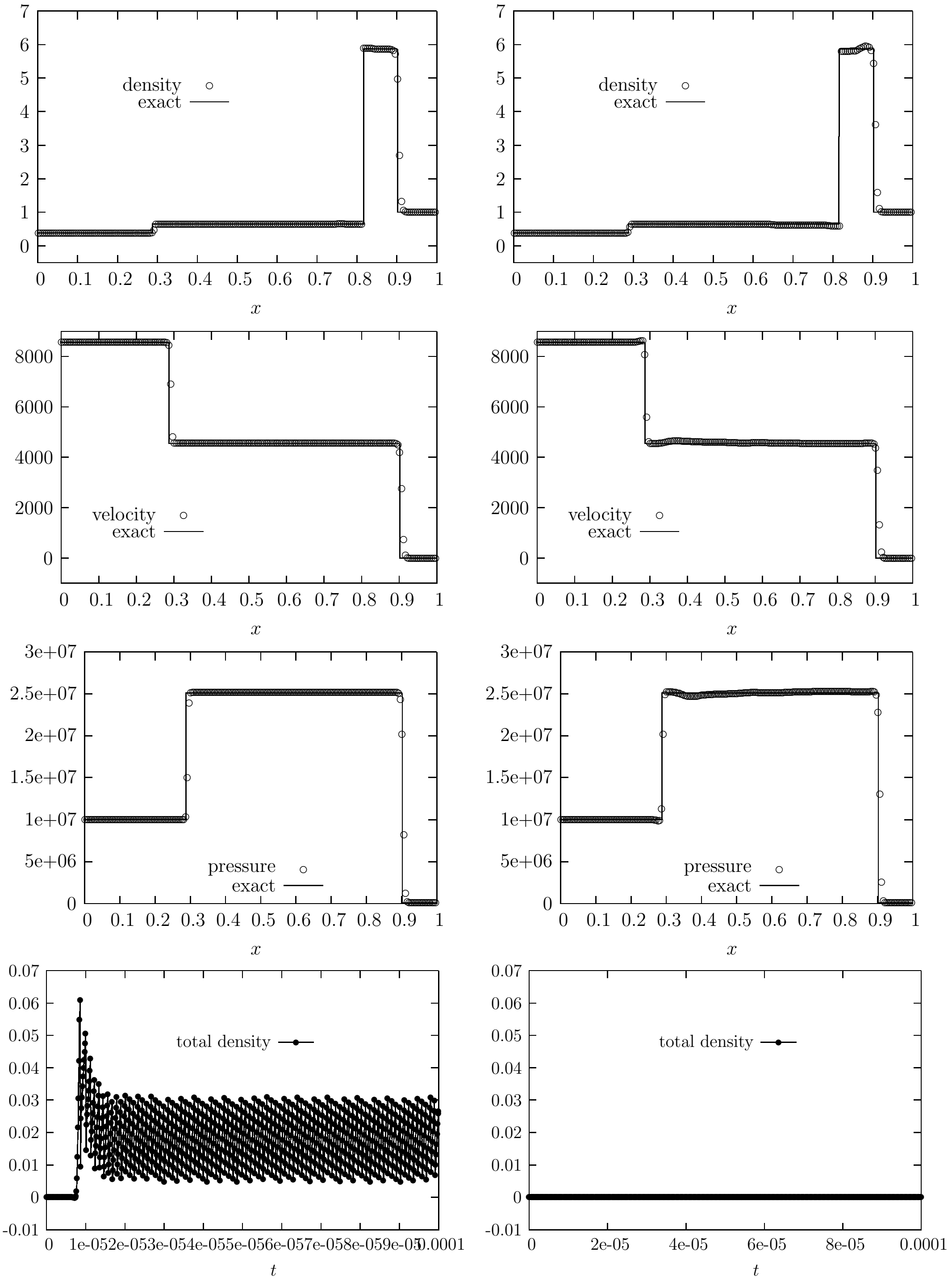

Figure 9: Example 4.6. Left: Result of GFM; Right: GFM with partially conservative modification. CFL=0.4.

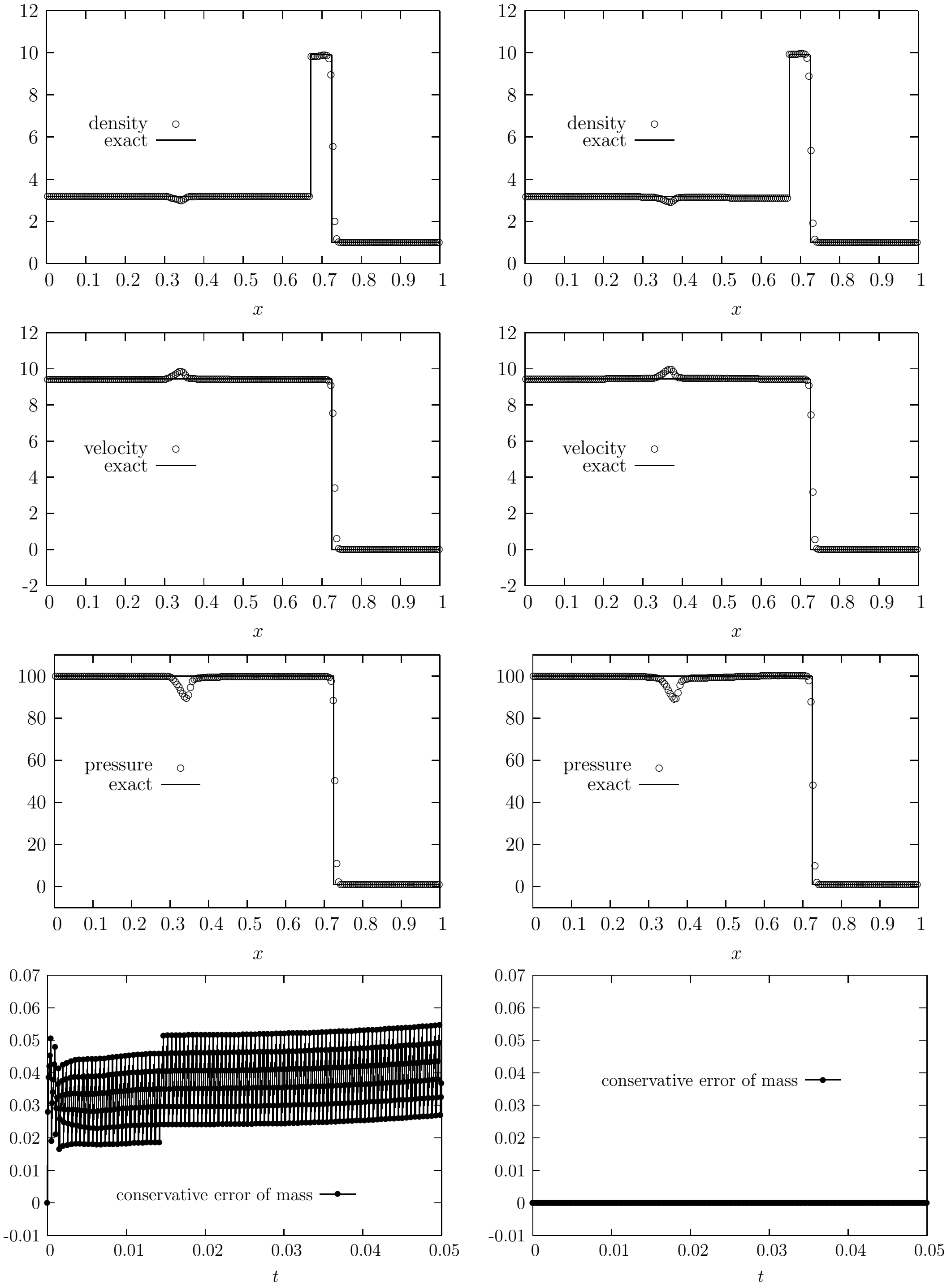

Figure 10: Example 4.7. Left: Result of GFM; Right: GFM with partially conservative modification. CFL=0.4.

Example 4.8. This is a two dimensional example, with an initial Mach 1.22 air shock impacting on a helium bubble. This problem is studied by Haas and Sturtevant experimentally in [10]. Several numerical studies have been undertaken for this example [6,13,18,23].

The computational domain is $[0,325]\times[-44.5,44.5]$, the nondimensionalized initial conditions are:

$$\begin{cases} x>225: & (\rho,u,v,p,\gamma)=(1.376363,-0.394728,0,1.5698,1.4), \\ x\leq 225: & (\rho,u,v,p,\gamma)=(1,0,0,1,1.4), \\ (x-175)^2+y^2\leq 625: & (\rho,u,v,p,\gamma)=(0.181875,0,0,1,1.648). \end{cases} \tag{4.9}$$

The upper and lower boundary conditions are implemented as nonreflective open boundaries. The left and right boundary conditions are the outflow and inflow. The level set function is initialized as $\phi=\sqrt{(x-175)^2+y^2}-25$ where $\phi\leq 0$ represents helium and $\phi>0$ represents the air. The post-shock air state is given for $x>225$.

We use a 640×160 uniform mesh to simulate this problem with a CFL number taken as 0.3. We run our partially conservative modified GFM with conservative mass and momentum to a time of 134.8, this corresponds to $674\mu s$ in the experimental result of [10]. The conservation error of the total mass is plotted in Fig. 11. We can see clearly that the original GFM has rather large conservation errors, while the modified GFM is basically mass conserved. The small deviation from zero for the modified GFM is most likely due to the error in our evaluation of mass inflow and outflow through the boundary of the physical domain. Fig. 12 compares the experimental result from [10] and our method at time $32\mu s$ (Fig. 12(a)), $52\mu s$ (Fig. 12)(b)), $62\mu s$ (Fig. 12(c)), $72\mu s$ (Fig. 12(d)), $82\mu s$ (Fig. 12(e)), $102\mu s$ (Fig. 12(f)), $245\mu s$ (Fig. 12(g)) and $427\mu s$ (Fig. 12(h)). We observe quite satisfactory computational results by our partially conservative modified GFM.

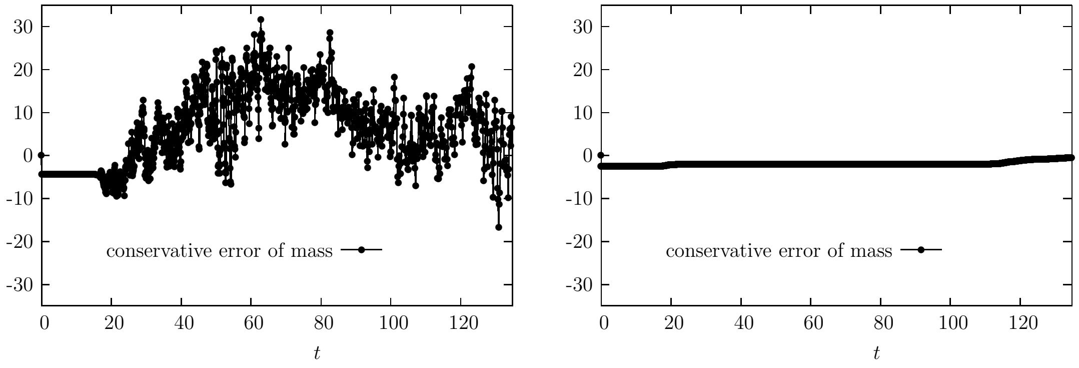

Figure 11: Example 4.8. Conservation errors of mass. Left: the GFM; Right: the partially conservative modified GFM.

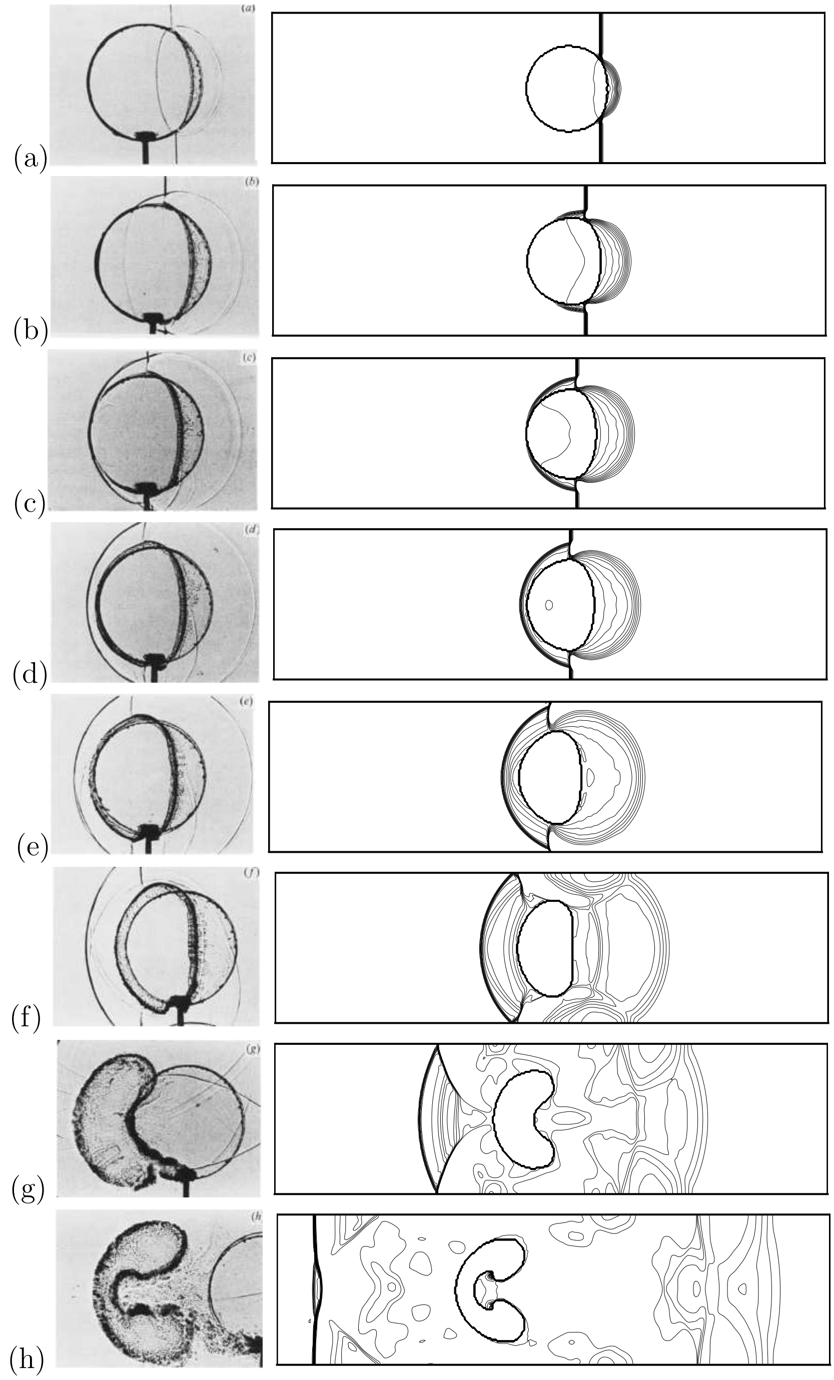

Figure 12: Example 4.8. Left: experimental result (Haas and Sturtevant [10]); Right: contour of density obtained by our partially conservative modified GFM. (a) $t=32\mu s$, (b) $t=52\mu s$, (c) $t=62\mu s$, (d) $t=72\mu s$, (e) $t=82\mu s$, (f) $t=102\mu s$, (g) $t=245\mu s$, (h) $t=427\mu s$.

5 Concluding remarks

In this paper, we have made a conservative modification to the Ghost Fluid Method by tracking the conservative variables near the interface and using them to replace the numerical solution when the interface has moved away from the cell. From the numerical examples, our method achieves at least mass and momentum conservation without seriously affecting its resolution adversely. In some numerical examples, our conservative modified method even exhibits better agreement with the exact solution.

Acknowledgments

The research of L. Yuan was supported by NSFC 10531080, 10972230, and 973 project 2005CB321703 and 2010CB731505. The research of C.-W. Shu was supported by ARO grant W911NF-08-1-0520 and NSF grant DMS-0809086.

References

[1] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: a quasi-conservative approach, J. Comput. Phys., 125 (1996), 150-160.

[2] R. Abgrall and S. Karni, Computations of compressible multifluids, J. Comput. Phys., 169 (2001), 594-623.

[3] D. Adalsteinsson and J.A. Sethian, A fast level set method for propagating interfaces, J. Comput. Phys., 118 (1995), 269-277.

[4] D.A. Bailey, P.K. Sweby and P. Glaister, A ghost fluid, moving finite volume plus continuous remap method for compressible Euler flow, Int. J. Numer. Methods Fluids, 47 (2005), 833-840.

[5] 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.

[6] R.P. Fedkiw, T. Aslam, B. Merriman and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method), J. Comput. Phys., 152 (1999), 457-492.

[7] R.P. Fedkiw, A. Marquina and B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, J. Comput. Phys., 148 (1999), 545-578.

[8] J. Glimm, J.M. Grove, X.L. Li and N. Zhao, Simple front tracking, in: G.Q. Chen, E. DiBebedetto (Eds.), Contemporary Mathematics, vol. 238, AMS, Providence, RI, 1999.

[9] J. Glimm, D. Marchesin and O. McBryan, A numerical method for two phase flow with an unstable interface, J. Comput. Phys., 39 (1981), 179-200.

[10] J.-F. Haas and B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech., 181 (1987), 41-76.

[11] G. Jiang and D.-P. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, SIAM J. Sci. Comput., 21 (2000), 2126-2143.

[12] G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys., 126 (1996), 202-228.

[13] E. Johnsen and T. Colonius, Implementation of WENO schemes in compressible multicomponent flow problems, J. Comput. Phys., 219 (2006), 715-732.

[14] T.G. Liu, B.C. Khoo and C.W. Wang, The ghost fluid method for compressible gas-water simulation, J. Comput. Phys., 204 (2005), 193-221.

[15] T.G. Liu, B.C. Khoo and 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, Computers & Fluids, 30 (2001), 291-314.

[16] T.G. Liu, B.C. Khoo and K.S. Yeo, The simulation of compressible multi-medium flow. Part II: applications to 2D underwater shock refraction, Computers & Fluids, 30 (2001), 315-337.

[17] T.G. Liu, B.C. Khoo and K.S. Yeo, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys., 190 (2003), 651-681.

[18] A. Marquina and P. Mulet, A flux-split algorithm applied to conservative models for multicomponent compressible flows, J. Comput. Phys., 185 (2003), 120-138.

[19] W. Mulder, S. Osher and J.A. Sethian, Computing interface motion in compressible gas dynamics, J. Comput. Phys., 100 (1992), 209-228.

[20] S. Osher, R.P. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, 2003.

[21] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on HamiltonJacobi formulations, J. Comput. Phys., 79 (1988), 12-49.

[22] J. Qiu, T. Liu and B.C. Khoo, Simulations of compressible two-medium flow by Runge-Kutta discontinuous Galerkin methods with the Ghost Fluid Method, Commun. Comput. Phys., 3 (2008), 479-504.

[23] J. Quirk and S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech., 318 (1996), 129-163.

[24] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.

[25] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput., 9 (1988), 1073-1084.

[26] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys., 77 (1988), 439-471.

[27] M. Sussman, P. Smereka and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys., 114 (1994), 146-159.

[28] C.W. Wang, T.G. Liu and B.C. Khoo, A real-ghost fluid method for the simulation of multi-medium compressible flow, SIAM J. Sci. Comput., 28 (2006), 278-302.