COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **6**, No. 5, pp. 1137-1162

Commun. Comput. Phys.
November 2009

An Interface-Capturing Method for Resolving Compressible Two-Fluid Flows with General Equation of State

T. S. Lee, J. G. Zheng* and S. H. Winoto

Department of Mechanical Engineering, National University of Singapore, Singapore 119260.

Received 27 October 2008; Accepted (in revised version) 26 February 2009

Available online 14 May 2009

Abstract. In this study, a stable and robust interface-capturing method is developed to resolve inviscid, compressible two-fluid flows with general equation of state (EOS). The governing equations consist of mass conservation equation for each fluid, momentum and energy equations for mixture and an advection equation for volume fraction of one fluid component. Assumption of pressure equilibrium across an interface is used to close the model system. MUSCL-Hancock scheme is extended to construct input states for Riemann problems, whose solutions are calculated using generalized HLLC approximate Riemann solver. Adaptive mesh refinement (AMR) capability is built into hydrodynamic code. The resulting method has some advantages. First, it is very stable and robust, as the advection equation is handled properly. Second, general equation of state can model more materials than simple EOSs such as ideal and stiffened gas EOSs for example. In addition, AMR enables us to properly resolve flow features at disparate scales. Finally, this method is quite simple, time-efficient and easy to implement.

AMS subject classifications: 35L65, 65M50, 65M99, 76T99

Key words: MUSCL-Hancock scheme, adaptive mesh refinement, compressible two-fluid flows, general equation of state.

1 Introduction

Numerical simulation of compressible multi-fluid flows has attracted much attention during the past few decades because they widely exist in nature and industrial applications. However, direct application of existing high-resolution methods such as Godunov-type and WENO schemes to these flows gives rise to unphysical pressure oscillations on

*Corresponding author. *Email addresses:* mpeleets@nus.edu.sg (T. S. Lee), g0500385@nus.edu.sg (J. G. Zheng), mpewinot@nus.edu.sg (S. H. Winoto)

material interfaces. This problem was solved by Abgrall by assuming pressure equilibrium across a material interface [2]. Since then, much progress has been made by following the work of Abgrall, refer to [3,8,13–17,23,25,26], etc. All these methods including the one developed in this study are based on volume fraction formulation and belong to the interface-capturing family. In other words, the interfaces are not tracked accurately, but allowed to diffuse numerically over a few grid cells. The fluids are marked by volume fraction of one component. In transition layers between two components, value of volume fraction lies between 0 and 1 and a mixture or artificial equation of state (EOS) needs to be introduced to model the mixture. Compared with other methods like interface-tracking, volume of fluid, and level-set methods, the interface-capturing method is easier to implement but less accurate. The interface-tracking can accurately track the material interfaces without introducing numerical diffusion. The volume of fluid reconstructs the interfaces using complicated interpolation schemes. Although accurate, the two methods will be very complex when applied to three-dimensional problems and cases with drastic topological changes. The level-set method is simple but not discretely conservative. The present interface-capturing method is less accurate as it may admit excessive numerical diffusion on the interfaces. However, we can use adaptive mesh refinement (AMR) to improve resolution. This method can be easily applied to three-dimensional problems and those with complex EOS. See [3, 8] and references cited therein for comparison of different methods.

Although much research has been devoted to resolution of multi-fluid flows, few studies have been done on fluids modeled by general or complex EOS. In this study, we concentrate on extension of MUSCL-Hancock scheme, HLLC Riemann solver and AMR to treatment of flows with Mie-Grüneisen EOS, which includes cases of stiffened gas, van der Waals, Jones-Wilkins-Lee EOSs, etc. These EOSs can model a wide range of real materials in practice, but they also introduce difficulties into numerical algorithm. In our method, mass of each fluid is conserved because material parameters in the mixture EOS depend on the partial densities. However, only the momentum and energy equations for the mixture are included. The material interfaces are identified using the volume fraction, which obeys an advection equation. Physically, the volume fraction is passively advected at local speed. Nevertheless, some of methods in the literature update value of the volume fraction according to the velocity field at previous time step. This, however, can lead to unphysical partial densities and failure of calculations under certain situations. To rectify this problem, in this study, the modified HLLC Riemann solver developed in [8] is generalized to construct numerical fluxes, particularly those of the advection equation. Since these fluxes are constructed using solutions of Riemann problems and information of the velocity field at present time step has been taken into account, the volume fraction is advected properly. The resulting scheme is stabler than others. In addition, HLLC is less expensive computationally than other algorithms like two-shock solver where iteration is needed in finding solution, and therefore, our method is time-efficient. In this study, MUSCL-Hancock scheme is extended to construct input states for the Riemann problems. This scheme with minmod limiter proves to be very robust and easy to imple-

ment.

In this study, physical effects including surface tension, viscosity and heat transfer on the material interfaces are ignored as the simplified model equations are used. However, our method can be generalized to problems where viscosity should be considered. In fact, using the operator splitting [1, 4], the inviscid and viscous parts of governing equations can be handled separately in each time step. The former is solved using the present method and the latter is dealt with using an explicit central difference scheme.

Typically, solution of a compressible flow contains various flow features including steep gradients such as shocks and contact discontinuities for example and complex smooth structures. But many methods use an uniform grid and to capture fine flow structures, a very fine grid is needed. In this study, we employ AMR to sufficiently resolve these features at disparate scales and to reduce computational cost. The AMR algorithm used here is based on logically Cartesian block-structured grid [11]. As many restrictions are imposed, the resulting algorithm is easier to implement and data organization as well as programming are also simplified significantly.

This paper is organized as follows. In Section 2, the governing equations and EOSs are presented. In Section 3, the numerical method is discussed in detail. Issues related to AMR are addressed in Section 4. This method is validated on a set of test problems in Section 5. Finally, conclusion is given in Section 6.

2 Governing equations

The governing equations for inviscid, compressible two-fluid flows read

$$\begin{aligned}
&\frac{\partial}{\partial t}(Y^{(1)}\rho^{(1)})+\nabla\cdot(Y^{(1)}\rho^{(1)}\mathbf{v})=0,\\
&\frac{\partial}{\partial t}(Y^{(2)}\rho^{(2)})+\nabla\cdot(Y^{(2)}\rho^{(2)}\mathbf{v})=0,\\
&\frac{\partial(\rho\mathbf{v})}{\partial t}+\nabla\cdot(\rho\mathbf{v}\mathbf{v}+pI)=0,\\
&\frac{\partial(\rho E)}{\partial t}+\nabla\cdot[(\rho E+p)\mathbf{v}]=0,\\
&\frac{\partial Y^{(1)}}{\partial t}+\mathbf{v}\cdot\nabla Y^{(1)}=0,
\end{aligned} \tag{2.1}$$

where $\rho^{(1)}$ and $\rho^{(2)}$ are the partial densities of fluids 1 and 2, respectively; p is the pressure, $\mathbf{v}$ is the vector of velocity and $E=e+\frac{1}{2}\mathbf{v}\cdot\mathbf{v}$ is the total energy per unit mass with e being the internal energy; $Y^{(1)}$, the volume fraction of fluid 1, lies in the interval $[0,1]$ and satisfies relation $Y^{(1)}+Y^{(2)}=1$. The total density, momentum and energy of the mixture

are defined as

$$\rho=\sum_{k=1}^{2}Y^{(k)}\rho^{(k)},\quad \rho\mathbf{v}=\sum_{k=1}^{2}Y^{(k)}\rho^{(k)}\mathbf{v}^{(k)}, \tag{2.2a}$$

$$\rho E=\sum_{k=1}^{2}\left(Y^{(k)}\rho^{(k)}e^{(k)}+\frac{1}{2}Y^{(k)}\rho^{(k)}(\mathbf{v}^{(k)})^2\right). \tag{2.2b}$$

In [3], Allaire et al. proved consistency, hyperbolicity and existence of a mathematic entropy of the model system (2.1). For simplicity, this procedure is not repeated here.

Next, a mixture EOS needs to be derived to close (2.1). Here, each of the fluids is modeled by Mie-Grüneisen EOS,

$$p(\rho,e)=\Gamma(\rho)[\rho e-\rho e_{\text{ref}}(\rho)]+p_{\text{ref}}(\rho). \tag{2.3}$$

This EOS is referred to as general EOS as it can produce the following six types of EOSs:

(1) Ideal gas EOS

$$\begin{cases}\Gamma(\rho)=\gamma-1,\\ p_{\text{ref}}(\rho)=0,\\ e_{\text{ref}}(\rho)=0;\end{cases} \tag{2.4}$$

(2) Stiffened gas EOS

$$\begin{cases}\Gamma(\rho)=\gamma-1,\\ p_{\text{ref}}(\rho)=-\gamma\pi,\\ e_{\text{ref}}(\rho)=0;\end{cases} \tag{2.5}$$

(3) van der Waals EOS

$$\begin{cases}\Gamma(\rho)=\dfrac{\gamma-1}{1-b\rho},\\ p_{\text{ref}}(\rho)=-a\rho^2,\\ e_{\text{ref}}(\rho)=-a\rho;\end{cases} \tag{2.6}$$

(4) Jones-Wilkins-Lee EOS

$$\begin{cases}\Gamma(\rho)=\Gamma_0,\\ e_{\text{ref}}(\rho)=\dfrac{\mathcal{A}}{\mathcal{R}_1\rho_0}\exp\left(\dfrac{-\mathcal{R}_1\rho_0}{\rho}\right)+\dfrac{\mathcal{B}}{\mathcal{R}_2\rho_0}\exp\left(\dfrac{-\mathcal{R}_2\rho_0}{\rho}\right)-e_0,\\ p_{\text{ref}}(\rho)=\mathcal{A}\exp\left(\dfrac{-\mathcal{R}_1\rho_0}{\rho}\right)+\mathcal{B}\exp\left(\dfrac{-\mathcal{R}_2\rho_0}{\rho}\right);\end{cases} \tag{2.7}$$

(5) Cochran-Chan EOS

$$\begin{cases} \Gamma(\rho)=\Gamma_0, \\ e_{\text{ref}}(\rho)=\dfrac{-\mathcal{A}}{\rho_0(1-\varepsilon_1)}\left[\left(\dfrac{\rho_0}{\rho}\right)^{1-\varepsilon_1}-1\right]+\dfrac{\mathcal{B}}{\rho_0(1-\varepsilon_2)}\left[\left(\dfrac{\rho_0}{\rho}\right)^{1-\varepsilon_2}-1\right]-e_0, \\ p_{\text{ref}}(\rho)=\mathcal{A}\left(\dfrac{\rho_0}{\rho}\right)^{-\varepsilon_1}-\mathcal{B}\left(\dfrac{\rho_0}{\rho}\right)^{-\varepsilon_2}; \end{cases} \tag{2.8}$$

(6) Shock-Wave EOS

$$\begin{cases} \Gamma(\rho)=\Gamma_0\left(\dfrac{V}{V_0}\right)^{\alpha}, \quad V=\dfrac{1}{\rho}, \quad V_0=\dfrac{1}{\rho_0}, \\ p_{\text{ref}}(\rho)=p_0+\dfrac{c_0^2(V_0-V)}{[V_0-s(V_0-V)]^2}, \\ e_{\text{ref}}(\rho)=e_0+\dfrac{1}{2}[p_{\text{ref}}(\rho)+p_0](V_0-V). \end{cases} \tag{2.9}$$

These EOSs can model a wide range of real materials. Stiffened gas EOS, for example, is suitable for gas, liquid and solid under high pressure; van der Waals EOS takes into account real-gas effect because molecular cohesive forces and size of molecules are considered; Cochran-Chan EOS can also characterize solid under strong shock, but is more accurate than stiffened gas EOS. For more details of these EOSs, the reader is referred to [3, 14, 16].

It is well known that the challenging problem in simulation of multi-fluid flows is to eliminate pressure oscillations on the material interfaces. Cause of these oscillations was identified first by Abgrall [2]. The basic idea of Abgrall is to assume that there is no pressure jump across a material interface separating two fluids, i.e.,

$$p^{(1)}=p^{(2)}=p.$$

This assumption is key to oscillation-free schemes design and also adopted here.

It is easy to verify that Mie-Grüneisen EOS (2.3) can be rewritten as

$$p(\rho,\rho e)=(\gamma(\rho)-1)\rho e-\gamma(\rho)\pi(\rho), \tag{2.10}$$

where

$$\gamma(\rho)=\Gamma(\rho)+1, \quad \pi(\rho)=\frac{\Gamma(\rho)\rho e_{\text{ref}}(\rho)-p_{\text{ref}}(\rho)}{\Gamma(\rho)+1}.$$

It is also assumed that the velocity does not vary across a material interface, namely $\mathbf{v}^{(1)}=\mathbf{v}^{(2)}=\mathbf{v}$. Then, according to Eq. (2.2b), the internal energy of the mixture is given by

$$\rho e=\sum_k Y^{(k)}\rho^{(k)}e^{(k)}=\sum_k Y^{(k)}\frac{p^{(k)}+\gamma^{(k)}(\rho^{(k)})\pi^{(k)}(\rho^{(k)})}{\gamma^{(k)}(\rho^{(k)})-1}. \tag{2.11}$$

If the pressure of the mixture is related to its density and internal energy through a mixture EOS $p=(\gamma-1)\rho e-\gamma\pi$, then through using $p^{(1)}=p^{(2)}=p$, Eq. (2.11) can be split into the following two equalities,

$$\frac{1}{\gamma-1}=\sum_k \frac{\Upsilon^{(k)}}{\gamma^{(k)}(\rho^{(k)})-1}, \quad \frac{\gamma\pi}{\gamma-1}=\sum_k \Upsilon^{(k)}\frac{\gamma^{(k)}(\rho^{(k)})\pi^{(k)}(\rho^{(k)})}{\gamma^{(k)}(\rho^{(k)})-1}. \tag{2.12}$$

In calculations, we first evolve governing equations (2.1), and then values of $\rho^{(1)}$, $\rho^{(2)}$, $\mathbf{v}$, e and $\Upsilon^{(1)}$ follow. Next, γ and π in above two equalities are computed. Finally, solution to the pressure of the mixture is found through $p=(\gamma-1)\rho e-\gamma\pi$. At this point, the model system (2.1) is complete with the mixture EOS.

A quantity frequently used is the sound speed of the mixture, which can be derived in a straightforward manner according to its definition. In the case of Mie-Grüneisen EOS, the sound speed is given by

$$c^2=(\gamma-1)\sum_k \frac{z^{(k)}(c^{(k)})^2}{\gamma^{(k)}-1}, \tag{2.13}$$

where $z^{(k)}$ is mass fraction of fluid k and defined as $z^{(k)}=(\Upsilon^{(k)}\rho^{(k)})/(\sum\Upsilon^{(k)}\rho^{(k)})$.

3 Numerical method

In this paper, the classical MUSCL-Hancock scheme is extended to construct input states for Riemann problems. MUSCL scheme is the second-order extension of Godunov's first order upwind method. It is introduced by van Leer by representing variables distributions as linear functions in each grid cell [21]. MUSCL scheme is simplified significantly by Hancock and the resulting algorithm is called MUSCL-Hancock scheme, which is of predictor-corrector type [22]. Here, we reconstruct the primitive variables rather than conserved variables so as to maintain pressure equilibrium. HLLC Riemann solver is generalized to calculate numerical fluxes including those for the advection equation. As the volume fraction is advected properly, the present scheme is stabler than some of other methods. Multi-dimensional problems are handled using Strang splitting.

3.1 Variables reconstruction

We only concentrate on one-dimensional case since multi-dimensional problems can be treated dimension by dimension using Strang splitting [18]. The augmented one-dimensional governing equations in x direction can be expressed in terms of primitive variables as

$$\mathbf{W}_t+\mathrm{A}\mathbf{W}_x=0, \tag{3.1}$$

where $\mathbf{W}=(Y^{(1)}\rho^{(1)},Y^{(2)}\rho^{(2)},u,v,w,p,Y^{(1)})$ is the vector of primitive variables. The Jacobian matrix of the system takes the form

$$\mathrm{A}=\begin{pmatrix} u & 0 & Y^{(1)}\rho^{(1)} & 0 & 0 & 0 & 0 \\ 0 & u & Y^{(2)}\rho^{(2)} & 0 & 0 & 0 & 0 \\ 0 & 0 & u & 0 & 0 & \frac{1}{\rho} & 0 \\ 0 & 0 & 0 & u & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & u & 0 & 0 \\ 0 & 0 & \rho c^2 & 0 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & u \end{pmatrix}, \tag{3.2}$$

where c is the sound speed defined in Eq. (2.13).

In each cell i, $I_i=[x_{i-1/2},x_{i+1/2}]$, the reconstructed primitive variables distributions are linear functions,

$$\mathbf{W}_i(x)=\mathbf{W}_i^n+\frac{(x-x_i)}{\Delta x}\Delta_i,\quad x\in[x_{i-1/2},x_{i+1/2}], \tag{3.3}$$

where $\Delta x=x_{i+1/2}-x_{i-1/2}$ and Δ_i is a chosen slope vector

$$\Delta_i=\mathrm{minmod}(\mathbf{W}_i-\mathbf{W}_{i-1},\mathbf{W}_{i+1}-\mathbf{W}_i), \tag{3.4}$$

where minmod limiter is defined as $\mathrm{minmod}(x,y)=\frac{1}{2}(\mathrm{sign}(x)+\mathrm{sign}(y))\min(|x|,|y|)$. The interface values of $\mathbf{W}_i(x)$,

$$\mathbf{W}_{i-1/2,R}=\mathbf{W}_i^n-\frac{1}{2}\Delta_i,\quad \mathbf{W}_{i+1/2,L}=\mathbf{W}_i^n+\frac{1}{2}\Delta_i. \tag{3.5}$$

Next, the solution is advanced by the half time step

$$\mathbf{W}_i^{n+1/2}=\mathbf{W}_i^n-\frac{\Delta t}{2\Delta x}\mathrm{A}\Delta_i. \tag{3.6}$$

This is called predictor step. Then, using the old gradient Δ_i, the interface values in cell i are given by

$$\tilde{\mathbf{W}}_{i-1/2,R}=\mathbf{W}_i^n-\frac{1}{2}(I+\frac{\Delta t}{\Delta x}\mathrm{A})\Delta_i,\quad \tilde{\mathbf{W}}_{i+1/2,L}=\mathbf{W}_i^n+\frac{1}{2}(I-\frac{\Delta t}{\Delta x}\mathrm{A})\Delta_i, \tag{3.7}$$

where I is the identity matrix of size 7. These interface values are the required input states for the Riemann problems.

3.2 HLLC Riemann solver

HLLC solver is one of the most popular approximate Riemann solvers [20]. It computes the numerical fluxes directly, and therefore is time-efficient. Here, we generalize the

modified HLLC solver of Johnsen and Colonius in [8] to flows governed by equations (2.1) with complex EOSs (2.4)-(2.9).

We first construct numerical fluxes for all the equations in (2.1) except for the last one. Define intermediate states of the conserved variables as

$$\mathbf{U}_{*k}=\left(\frac{s_k-u_k}{s_k-s_*}\right)\begin{pmatrix}\rho_k^{(1)}Y_k^{(1)}\\ \rho_k^{(2)}Y_k^{(2)}\\ \rho_k s_*\\ \rho_k v_k\\ \rho_k w_k\\ \rho_k E_k+\rho_k(s_*-u_k)\left(s_*+\dfrac{p_k}{\rho_k(s_k-u_k)}\right)\end{pmatrix}, \tag{3.8}$$

where $k=L,R$. s_L and s_R are the minimum and maximum signal velocities in solution of the Riemann problem at a cell interface, and are defined as

$$s_L=\min(u_L-c_L,u_R-c_R),\quad s_R=\max(u_L+c_L,u_R+c_R).$$

The speed of intermediate wave, s_*, is calculated as

$$s_*=\frac{p_R-p_L+\rho_L u_L(s_L-u_L)-\rho_R u_R(s_R-u_R)}{\rho_L(s_L-u_L)-\rho_R(s_R-u_R)}. \tag{3.9}$$

The numerical fluxes corresponding to $\mathbf{U}_{*L}$ and $\mathbf{U}_{*R}$ are given by

$$\mathbf{F}_{*L}=\mathbf{F}_L+s_L(\mathbf{U}_{*L}-\mathbf{U}_L),\quad \mathbf{F}_{*R}=\mathbf{F}_R+s_R(\mathbf{U}_{*R}-\mathbf{U}_R). \tag{3.10}$$

Then, according to locations of these waves, the numerical fluxes can be determined,

$$\mathbf{F}=\begin{cases}\mathbf{F}_L & \text{if}\quad 0\le s_L,\\ \mathbf{F}_{*L} & \text{if}\quad s_L\le 0\le s_*,\\ \mathbf{F}_{*R} & \text{if}\quad s_*\le 0\le s_R,\\ \mathbf{F}_R & \text{if}\quad 0\ge s_R.\end{cases} \tag{3.11}$$

See [20] for details of this procedure. Next, we proceed to construct fluxes for the advection equation in (2.1) by following [8]. Rewrite the advection equation as

$$\frac{\partial Y^{(1)}}{\partial t}+\frac{\partial}{\partial x}(Y^{(1)}u)-Y^{(1)}\frac{\partial u}{\partial x}=0. \tag{3.12}$$

Denoting flux $Y^{(1)}u$ by h and integrating the above equation over cell i produce,

$$\frac{d(Y^{(1)})_i}{dt}=-\frac{1}{\Delta x}(h_{i+1/2}-h_{i-1/2})+\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}}Y^{(1)}\frac{\partial u}{\partial x}dx. \tag{3.13}$$

This equation can be approximated as

$$(Y^{(1)})_i^{n+1} = (Y^{(1)})_i^n - \frac{\Delta t}{\Delta x}(h_{i+1/2} - h_{i-1/2}) + \frac{\Delta t}{\Delta x}(Y^{(1)})_i^n (u_{i+1/2} - u_{i-1/2}), \tag{3.14}$$

where fluxes $h_{i\pm1/2}$ can be constructed in exactly the same manner as $\mathbf{F}$. $u_{i+1/2}$ is computed as,

$$u_{i+1/2} = \begin{cases} u_L, & \text{if} \quad s_L \geq 0, \\ u_L + s_L[(s_L - u_L)/(s_L - s_*) - 1], & \text{if} \quad s_L \leq 0 \leq s_*, \\ u_R + s_R[(s_R - u_R)/(s_R - s_*) - 1], & \text{if} \quad s_* \leq 0 \leq s_R, \\ u_R, & \text{if} \quad s_R \leq 0. \end{cases} \tag{3.15}$$

The other conserved variables are updated to the new time level t^{n+1} using the following scheme,

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}(\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}). \tag{3.16}$$

There are some advection schemes available. For example, Saurel and Abgrall [13,14] has applied the HLL approximate Riemann solver to multi-fluid flows and derived a discretization of the advection equation in (2.1). Although effective, HLL solver ignores contact and shear waves which are crucial to interface capturing. Some methods based on Roe Riemann solver were developed, but the resulting schemes are complicated.

Numerical tests show the present method based on HLLC solver is stabler than some methods in the literature under certain situations. The model system (2.1) can be derived using the theory of multi-phase flows. To use this system in whole computational domain, a pure fluid contained in a control volume is modeled as the mixture of this fluid and a small amount of the other component. For example, in initializing the solution, in regions containing fluid 1 only, the volume fractions of the two fluids are set to be $1-\epsilon$ and ϵ, respectively. Typically, ϵ is 10^{-8} or smaller. At each time step, we calculate the partial densities $\rho^{(1)}$ and $\rho^{(2)}$ by dividing $Y^{(1)}\rho^{(1)}$ and $Y^{(2)}\rho^{(2)}$ by the volume fractions $Y^{(1)}$ and $Y^{(2)}$, respectively. However, sometimes, this will lead to some numerical problems.

Take a one-dimensional, two-phase shock tube problem as an example. Initially, both fluids are stationary throughout the domain and the material interface is located at $x_{i-1/2}$ with fluid 1 on the left and fluid 2 on the right. Suppose that after the interface breaking, the fluids move to the right. After the first time step, values of all the variables in cell i must change. However, some of the methods advect the volume fraction $Y^{(1)}$ using the velocities on the previous time step. As a result, after the first time step, $Y^{(1)}$ and $Y^{(2)}$ do not vary as the velocities are zero initially. In cell i, dividing the updated value of $Y^{(1)}\rho^{(1)}$ by $Y^{(1)}$ produces an unphysical value of $\rho^{(1)}$, which is so large that the calculation fails immediately. In contrast, $Y^{(1)}$ is advected properly through (3.14), since numerical fluxes $h_{i\pm1/2}$ and $u_{i\pm1/2}$ are based on the Riemann problem solution which accounts for information on the present velocity field. The resulting value of $\rho^{(1)}$ is correct. Therefore, our method is stabler.

3.3 Oscillation-free property of the present method

It should be guaranteed that the discretized form of governing equations (2.1), i.e., Eqs. (3.16) and (3.14), can also maintain the pressure equilibrium across a material interface, which is a basic requirement of oscillation-free schemes design [2, 13]. As in [3], consider a one-dimensional interface advection problem under uniform pressure and velocity throughout the domain. The initial conditions are as follows: the interface is located at $x_{i-1/2}$; the state variables on the left side read

$$\mathbf{W}_L=(\rho^{(1)},\rho^{(2)},e^{(1)},e^{(2)},p,u,Y^{(1)})_L=(\rho_0^{(1)},\rho_0^{(2)},e_0^{(1)},e_0^{(2)},p_0,u_0,Y_L^{(1)})$$

and those on the right side read

$$\mathbf{W}_R=(\rho^{(1)},\rho^{(2)},e^{(1)},e^{(2)},p,u,Y^{(1)})_R=(\rho_0^{(1)},\rho_0^{(2)},e_0^{(1)},e_0^{(2)},p_0,u_0,Y_R^{(1)})$$

with $u_0>0$ and $Y_L^{(1)}\neq Y_R^{(1)}$. All the variables are continuous except for the volume fraction $Y^{(1)}$.

We compute the solution in cell i after a time step, as the interface must be in this cell at new time level. Denote the solution by $\bar{\mathbf{W}}$. The objective is to find $\bar{u}=u_0$, $\bar{p}=p_0$. It can be verified that at $x_{i-1/2}$, the speed of intermediate wave in (3.9) is $s_*=u_0>0$ and variables in (3.8) are $\mathbf{U}_{*k}=\mathbf{U}_k$. Then, from Eqs. (3.10) and (3.11), we calculate the numerical flux

$$\mathbf{F}_{i-1/2}=\mathbf{F}(\mathbf{U}_L). \tag{3.17}$$

Obviously, $h_{i-1/2}$ in (3.14) is given as $h_{i-1/2}=Y_L^{(1)}u_0$. Substituting $s_*=u_L=u_0>0$ into (3.15) gives

$$u_{i-1/2}=u_0. \tag{3.18}$$

Similarly, at $x_{i+1/2}$ we have $\mathbf{F}_{i+1/2}=\mathbf{F}(\mathbf{U}_R)$, $h_{i+1/2}=Y_R^{(1)}u_0$ and $u_{i+1/2}=u_0$. Substituting these fluxes into Eqs. (3.16) and (3.14) and performing some algebraic manipulations produce,

$$\begin{aligned}
&\bar{Y}^{(k)}\bar{\rho}^{(k)}=u_0\lambda Y_L^{(k)}\rho_L^{(k)}+(1-u_0\lambda)Y_R^{(k)}\rho_R^{(k)},\quad k=1,2,\\
&\bar{\rho}\bar{u}=(u_0\lambda\rho_L+(1-u_0\lambda)\rho_R)u_0,\\
&\bar{\rho}\bar{e}=u_0\lambda\rho_L e_L+(1-u_0\lambda)\rho_R e_R,\\
&\bar{Y}^{(k)}=u_0\lambda Y_L^{(k)}+(1-u_0\lambda)Y_R^{(k)},\quad k=1,2,
\end{aligned} \tag{3.19}$$

where $\lambda=\Delta t/\Delta x$. From the first and last equations in (3.19), we find $\bar{\rho}^{(k)}=\rho_0^{(k)}$. Similarly, we can obtain $\bar{u}=u_0$. Rewrite the third equation as,

$$\begin{aligned}
\bar{\rho}\bar{e}&=u_0\lambda\rho_L e_L+(1-u_0\lambda)\rho_R e_R\\
&=u_0\lambda\sum Y_L^{(k)}\rho_L^{(k)}e_L^{(k)}+(1-u_0\lambda)\sum Y_R^{(k)}\rho_R^{(k)}e_R^{(k)}\\
&=\sum(u_0\lambda Y_L^{(k)}+(1-u_0\lambda)Y_R^{(k)})\rho_0^{(k)}e_0^{(k)}=\sum\bar{Y}^{(k)}\rho_0^{(k)}e_0^{(k)}.
\end{aligned} \tag{3.20}$$

On the other hand, $\bar{\rho}\bar{e}$ is defined as $\bar{\rho}\bar{e}=\sum\bar{Y}^{(k)}\bar{\rho}^{(k)}\bar{e}^{(k)}$. This means

$$\sum\bar{Y}^{(k)}\rho_0^{(k)}e_0^{(k)}=\sum\bar{Y}^{(k)}\bar{\rho}^{(k)}\bar{e}^{(k)}. \tag{3.21}$$

Expressing the internal energy in terms of the pressure and density, we find that $\bar{p}=p_0$ is the unique solution of the above equation.

Since $\bar{p}=p_0$, there are no the pressure oscillations on the interface. Also, $\bar{p}=p_0$ holds in other cells. Therefore, our method is free of oscillations.

3.4 Extension to multiple dimensions

This method can be easily extended to multiple dimensions using the Strang splitting [18,20]. A three-dimensional problem, for example, can be handled as follows,

$$\begin{aligned}\mathbf{U}^{n+1}&=L_z(\Delta t)L_y(\Delta t)L_x(\Delta t)\mathbf{U}^n,\\ \mathbf{U}^{n+2}&=L_x(\Delta t)L_y(\Delta t)L_z(\Delta t)\mathbf{U}^{n+1},\end{aligned} \tag{3.22}$$

where L_x,L_y,L_z denote the one-dimensional operators in the x, y and z directions, respectively. This scheme is seconder-order accurate in time.

3.5 Boundary conditions

Three types of boundary conditions are employed in simulations: reflecting, non-reflecting and periodic boundary conditions. Fig. 1 shows a domain discretized into N grid cells in x direction, i.e., cells 1 to N. With MUSCL scheme, two fictitious cells are needed at each boundary. One needs to determine the primitive variables, $\mathbf{W}=(Y^{(1)}\rho^{(1)},Y^{(2)}\rho^{(2)},u,v,w,p,Y^{(1)})$, in these fictitious cells at the beginning of each time step.

Figure 1: Boundary conditions. At each boundary, there are two fictitious cells.

Reflecting boundary represents a solid wall. In cell $N+1$, all the variables are given as $\mathbf{W}_{N+1}=\mathbf{W}_N$ except that the normal velocity is reflected, $u_{N+1}=-u_N$. The variables in cell $N+2$ can be calculated from those in cell $N-1$ by taking the same procedure. Similarly, data in cells 0 and 1 can be determined. These conditions guarantee there is no flow across the boundaries. The non-reflecting boundary conditions allow waves to exit the boundaries and have no any effects on the waves. The data in fictitious cells are given by $\mathbf{W}_{-1,0}=\mathbf{W}_1$ and $\mathbf{W}_{N+1,N+2}=\mathbf{W}_N$. For periodic boundary conditions, $\mathbf{W}_0=\mathbf{W}_N$, $\mathbf{W}_{-1}=\mathbf{W}_{N-1}$, $\mathbf{W}_{N+1}=\mathbf{W}_1$ and $\mathbf{W}_{N+2}=\mathbf{W}_2$. For details of these boundary conditions, the reader is referred to [20].

4 Adaptive mesh refinement

The algorithm implemented in this study is categorized into the structured adaptive mesh refinement family which was first proposed by Berger et al. [5,6]. Although flexible and memory-efficient, the algorithm of Berger et al. results in a very complex code. This approach was significantly simplified by MacNeice et al. [11] and the resulting AMR tool, PARAMESH, is employed here due to its simplicity. It is based on the logically Cartesian block-structured grid, which implies that a physical grid must be converted into a Cartesian grid. The computational domain is covered with a hierarchy of grid blocks having the identical logical structure. Contrary to the algorithm of Berger et al., the present approach does not allow the grid blocks to overlap each other, to be rotated around the coordinate axes and to be merged with other blocks at the same refinement level. These restrictions imposed on the algorithm make the programming easier and the resulting code simpler [11]. In the following, we present details of this algorithm.

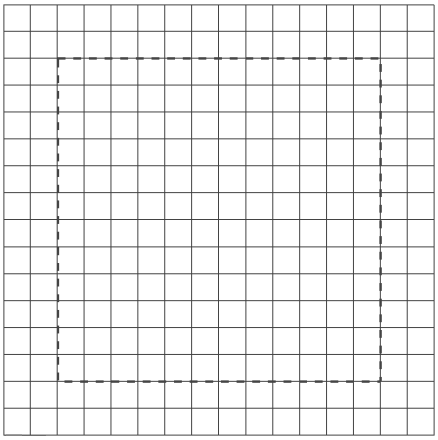

Figure 2: The structure of a grid block. Each block has 12×12 interior cells bounded by dashed line and 2 guard cells on each side.

Fig. 5(d) illustrates a computational domain with outlines of child blocks overlaid. A grid block is sketched in Fig. 2. It has 12×12 cells surrounded by 2 guard cells on each side. In the refinement process, a block is cut in half in the x and y directions, respectively, producing four children at a higher refinement level. Each of them has 12×12 cells, but the grid spacing in each direction is half that of its parent. After the refinement, the solution on the child blocks is prolonged from their parent using a quadratic interpolation. The guard cells of a block are filled with data from its neighbor blocks or user-defined boundary conditions, depending on its physical position. Obviously, a given point may be contained in a set of blocks at different refinement levels and among them, the one at the highest level is referred to as leaf block. Only the solution on the leaf blocks is updated in calculations. It is also required that jump in refinement level between two adjacent blocks is not larger than 1.

Criteria are needed to determine where and whether the refinement-derefinement procedure should be performed. They must be able to detect sharp structures before they enter a certain block. Here, the criterion proposed by Sun and Takayama [19] is employed. Define two error indicators

$$\phi_i = \frac{|\rho_{i-1,j} - 2\rho_{i,j} + \rho_{i+1,j}|}{\alpha\rho_{i,j} + |\rho_{i+1,j} - \rho_{i-1,j}|}, \quad \phi_j = \frac{|\rho_{i,j-1} - 2\rho_{i,j} + \rho_{i,j+1}|}{\alpha\rho_{i,j} + |\rho_{i,j+1} - \rho_{i,j-1}|}, \tag{4.1}$$

where $\alpha = 0.03$. Define $E_{ij} = \mathrm{Max}(\phi_i, \phi_j)$. For each block, calculate E_{ij} at each point and find the maximum. If the maximum is larger than 0.06, then the block is marked for refinement; otherwise if it is smaller than 0.05, the block is marked for derefinement.

Another criterion is also used to properly refine regions containing rarefaction in one-dimensional problems below [7, 10, 24]. It is required to calculate the difference approximation to a normalized second derivative error norm. In one dimension, the approximation is given as

$$E_i = \frac{|u_{i+2} - 2u_i + u_{i-2}|}{|u_{i+2} - u_i| + |u_i - u_{i-2}| + \epsilon[|u_{i+2}| + 2|u_i| + |u_{i-2}|]}. \tag{4.2}$$

Here, u represents a flow variable; ϵ is an adjustable parameter set to be 0.01. The maximum of E_i is calculated for each block.

We also enforce flux conservation. If there is a jump in refinement level between two adjacent blocks, the flux density entering or leaving a coarse cell interface on the less refined block should be equal to the average of those across two fine cells interfaces on the more refined block, as illustrated in Fig. 3.

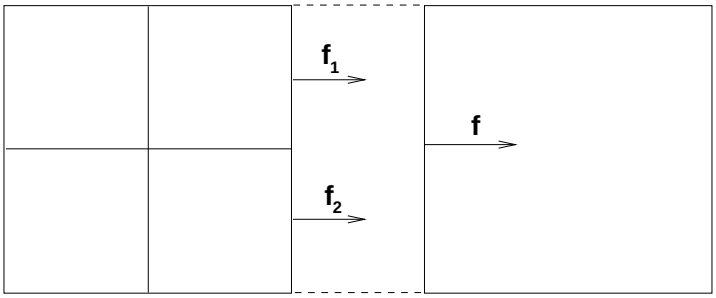

Figure 3: Flux conservation at a jump in refinement. In calculations, $\mathrm{f} = 0.5(\mathrm{f}_1 + \mathrm{f}_2)$.

Finally, the implementation of the hydrodynamic algorithm in conjunction with AMR is as follows:

1. Generate a set of grid blocks covering the computational domain and initialize the solution on this grid. Perform the refinement-derefinement process a few times to get the desired grid resolution everywhere and then initialize the solution on the newly generated grid.

2. Update the solution on each leaf block independently of any of other blocks by calling hydrodynamic subroutines and then apply the flux conservation.

3. Calculate error indicators or norms for each block to determine whether it should marked for refinement or derefinement.

4. If a block is determined to be refined, it is cut in half in each direction, producing four child blocks with their solution prolonged from their parent. Otherwise, if the block is determined to be derefined, the corresponding restriction process is performed. Next,fill the guard cells of grid blocks.

5. If the present time is less than end time, return to step 2.

5 Numerical results

The present method is validated on a set of test problems. In all calculations below except for Example 5.1, the Courant-Friedrichs-Lewy (CFL) number is set to be 0.8.

Example 5.1. To check order of accuracy of our method in space and time, we perform a convergence study on a one-dimensional (1D) single-component density advection problem [12]. The computational domain is $[0,2]$ with periodic boundary conditions. The initial conditions read $\rho(x,0)=1+0.2\sin(\pi x)$, $p(x,0)=1$, $u(x,0)=1$. The exact solution of this problem is $\rho(x,t)=1+0.2\sin(\pi(x-t))$, $p(x,t)=1$, $u(x,t)=1$. As the solution is smooth, the minmod limiter is turned off. The fluid is modeled by ideal gas EOS with $\gamma=1.4$.

In validating spatial accuracy, the ratio of time step to grid size is fixed at $\Delta t/\Delta x=0.2$. We run this case to $t=2$ and calculate L_1 error in the density, which is defined as $L_1=\Delta x\sum_{i=1}^{N_k}|\rho_i-\rho^{\text{exact}}|$. In this case, five sets of grids are used, i.e., $N_k=20,40,80,160,320$. Log-log plot of L_1 error vs grid size is shown in Fig. 4(a). As expected, the corresponding line is parallel to the dashed line of slope 2. The slope of solid line is around 2.001, which proves that our method is second-order accurate in space.

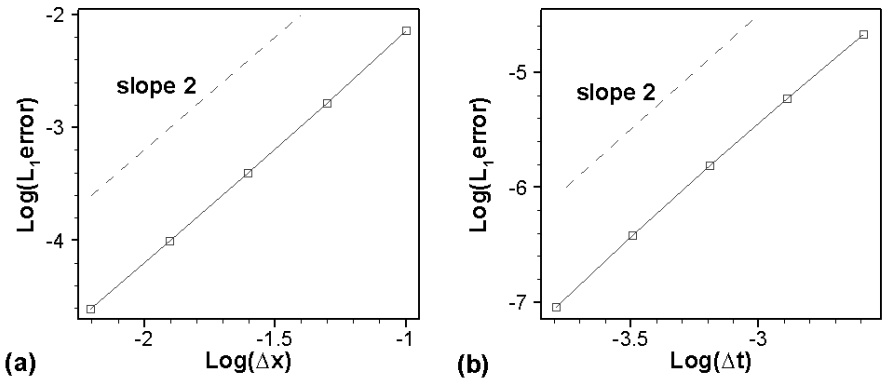

Figure 4: Log-Log plots of L_1 errors vs grid size (a) and time step (b).

In validating temporal accuracy, we use a 200-cell grid and a sequence of time steps, $\Delta t = 2.58\times 10^{-3}/2^N$ with $N = 0-4$. In the limit of $\Delta t \to 0$, we obtain a numerical solution, which is close to the exact solution. Then, we calculate error in L_1 norm and plot it in Fig. 4(b). The slope is about 2. This shows our method is also second-order accurate in time.

Example 5.2. This example is on a 1D material interface advection [3,14]. But it is run on a two-dimensional (2D) grid with y velocity component set to be 0 initially. In the unit square of $[0,1]\times[0,1]$, a material interface is located at x=0.5, separating two materials under uniform pressure $p = 10^5$Pa and velocity $u = 1500$m/s throughout the computational domain. The material on the left of the interface is copper with density of $\rho = 8900\text{kg/m}^3$, while that on the right is a solid explosive with density of $\rho = 1840\text{kg/m}^3$. Both components are modeled by Cochran-Chan EOS, but their material parameters are different and listed below,

$$(\rho_0,\mathcal{A},\mathcal{B},\epsilon_1,\epsilon_2,\Gamma_0,e_0) = \begin{cases} 8900\text{kg/m}^3,145.67\text{GPa},147.75\text{GPa},2.99,1.99,2,117.9\text{kJ/kg}, \\ 1840\text{kg/m}^3,12.87\text{GPa},13.42\text{GPa},4.1,3.1,0.93,326.1\text{kJ/kg}. \end{cases} \tag{5.1}$$

Obviously, the interface should move to the right with the uniform pressure and velocity. The cross-sectional results along $y = 0.5$ at $t = 240\mu s$ are presented in Fig. 5, where dash-dot lines are solution with AMR while solid lines denote the solution with an uniform grid of 5000 cells. The pressure and velocity remain equilibrium throughout the domain, as expected. There are no any pressure oscillations on the interface, which proves that our method is free of oscillations. As stated earlier, in the present numerical model, it is assumed that there is no pressure jump across a material interface. However, if one uses the so-called isothermal assumption, i.e. all components in a fluid element share the same temperature, the serious pressure oscillations occur on the interface. This is because this assumption does not reflect the actual physics in the interface region, see [2,3,13,14].

In this run, a grid with 7 refinement levels is used with level 1 corresponding to a single grid block covering the domain, and the effective grid size at the highest level is 768×768. Outlines of grid blocks at end time are demonstrated in Fig. 5(d). It is observed that interface region is maximally refined and therefore the interface remains sharp, while elsewhere the grid is derefined as the solution is smooth. Fig. 5 presents a good agreement between two sets of results, which also shows validity of AMR. If the interface region is not locally refined, the interface will diffuse greatly and become wider.

It should be remarked that, in all 1D problems, i.e., Examples 5.2 to 5.6, the computational domain is set to be an unit square of $[0,1]\times[0,1]$. In each figure, the dash-dot and solid lines represent solutions obtained using AMR and a fine grid of 5000 cells, respectively. Block structure of AMR grid is overlaid in frame (d) of each figure.

Example 5.3. This case is concerned with a copper-explosive impact problem of Saurel and Abgrall [14]. The initial conditions are nearly identical to those in Example 5.2, except that the velocity on the right side is set to be 0.

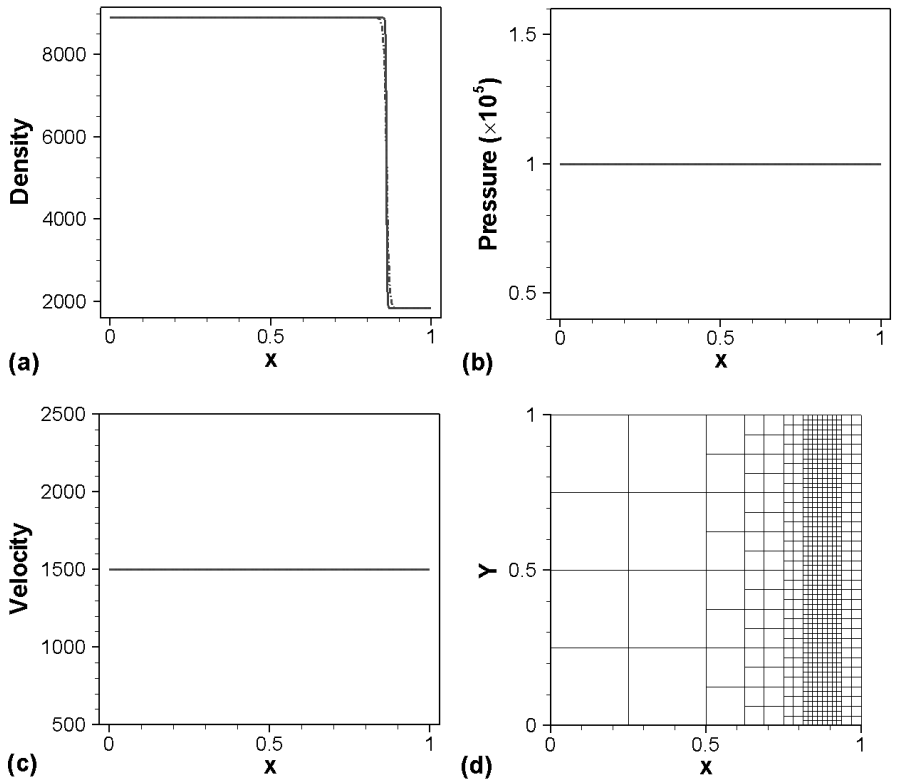

Figure 5: Cross-sectional results along $y=0.5$ for a copper-explosive interface advection problem $t=240\mu s$. The dash-dot lines are solution with AMR, while solid lines are solution with an uniform grid of 5000 cells. Frame (d) shows outlines of AMR blocks.

The solution consists of a rightward-moving shock, a leftward-moving shock and a material interface in between. The results of running this problem to $t=85\mu s$ are demonstrated in Fig. 6. As shown in Fig. 6, these nonlinear structures are all well resolved and remain sharp. The pressure oscillations are not present on the material interface. This run demonstrates that the present method can properly capture strong shock and material interface.

Again, 7 refinement levels are used. The regions surrounding two shocks and the material interface are maximally refined, while other regions are covered with grid blocks at lower levels. There is an excellent agreement between the solution with AMR and that with a fine grid.

Example 5.4. We study a two-component Riemann problem of Saurel and Abgrall [14]. The material on the left side of interface at $x=0.5$ is a detonation product with the pressure of 3.7×10^{10}Pa and density of 2485.37kg/m^3, while the material on the right side is copper which has the pressure of 10^5Pa and density of 8900kg/m^3. The two materials are both stationary initially, and modeled by Jones-Wilkins-Lee and Cochran-Chan EOSs,

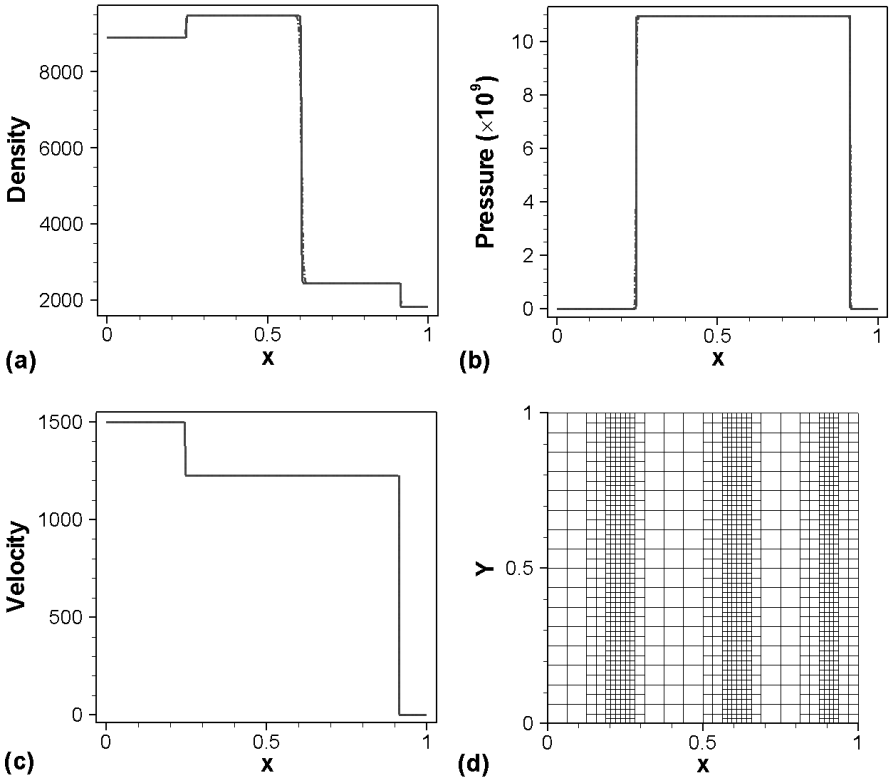

Figure 6: Results of a copper-explosive impact problem at $t=85\mu s$.

respectively, with material parameters given below,

$$\begin{aligned}&(\rho_0,\mathcal{A},\mathcal{B},\mathcal{R}_1,\mathcal{R}_2,\Gamma_0,e_0)_L\\=&(1840\text{kg/m}^3,854.5\text{GPa},20.5\text{GPa},4.6,1.35,0.25,8149.2\text{kJ/kg}),\\&(\rho_0,\mathcal{A},\mathcal{B},\epsilon_1,\epsilon_2,\Gamma_0,e_0)_R\\=&(8900\text{kg/m}^3,145.67\text{GPa},147.75\text{GPa},2.99,1.99,2,117.9\text{kJ/kg}).\end{aligned}\tag{5.2}$$

Because of the high pressure ratio up to 3.7×10^5 across the interface, there will be a strong shock in Copper, a rarefaction in detonation product and a material interface in between. As demonstrated in results at $t=73\mu s$ in Fig. 7, all these flow features are resolved well, which shows our code's ability to capture shock, material interface and to produce profile of rarefaction correctly. Also observed is the good agreement between two sets of solutions based on the fine grid and AMR grid with 7 refinement levels. For this Riemann problem with the initial stationary velocity field, our method is very stable, producing correct results, as discussed in Subsection 3.2.

In Examples 5.4-5.6, the numerical solutions contain rarefaction. It is found that (4.1) cannot refine rarefaction regions sufficiently. Alternatively, we use criterion (4.2) with the

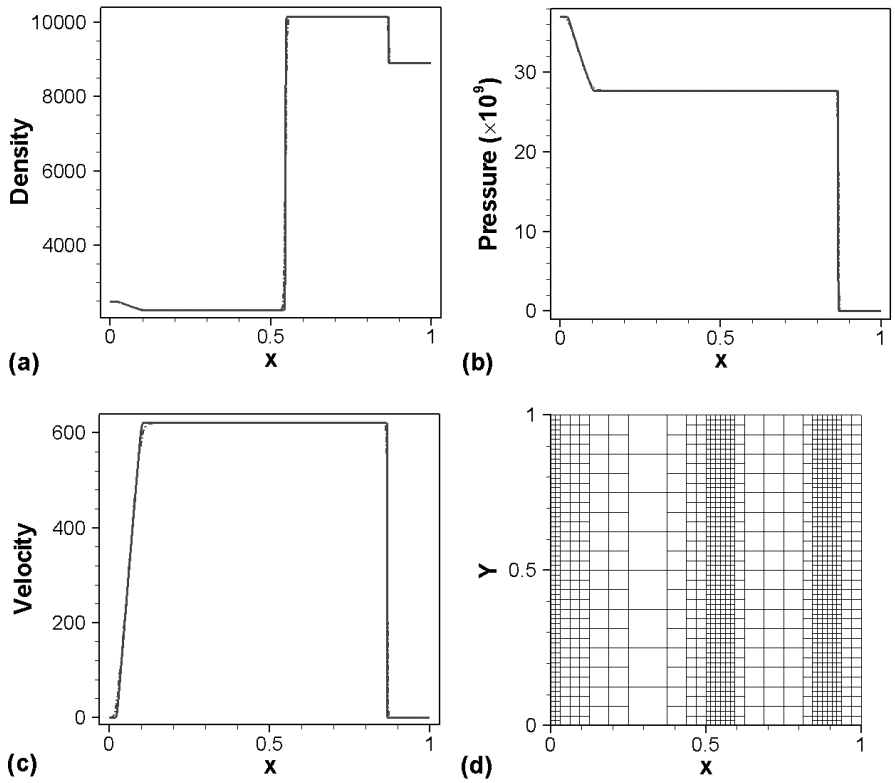

Figure 7: Results of a detonation product-copper Riemann problem at $t=73\mu s$.

momentum ρu and pressure p as two variables. As observed in Fig. 7, near head and tail of the rarefaction, the solution with AMR agrees quite well with that with the fine grid.

Example 5.5. Next, we consider a shock in molybdenum interacting with a molybdenum-MORB interface. Here, MORB is mid-ocean ridge basalt liquid. The initial conditions consist of a molybdenum-MORB interface at $x=0.6$ and a Mach 1.163 shock in molybdenum traveling to the right with its front located at $x=0.4$. The post-shock state reads $(\rho,u,p)=(11042\text{kg/m}^3,543\text{m/s},3\times10^{10}\text{Pa})$. The stationary molybdenum to the left of the interface has initial data $(\rho,u,p)=(9961\text{kg/m}^3,0,0)$, while state variables of MORB liquid are $(\rho,u,p)=(2260\text{kg/m}^3,0,0)$. Molybdenum and MORB are both modeled by shock wave EOS with the following material parameters,

$$(\rho_0,c_0,s,\Gamma_0,\alpha,e_0)=\begin{cases}(9961\text{kg/m}^3,4770\text{m/s},1.43,2.56,1,0\text{kJ/kg})_L,\\(2660\text{kg/m}^3,2100\text{m/s},1.68,1.18,1,0\text{kJ/kg})_R.\end{cases}\tag{5.3}$$

After shock interaction with the material interface, there are a transmitted shock in MORB, an interface separating two materials, and a reflected rarefaction in molybdenum. These structures are well captured in Fig. 8. Contact discontinuity and shock are resolved

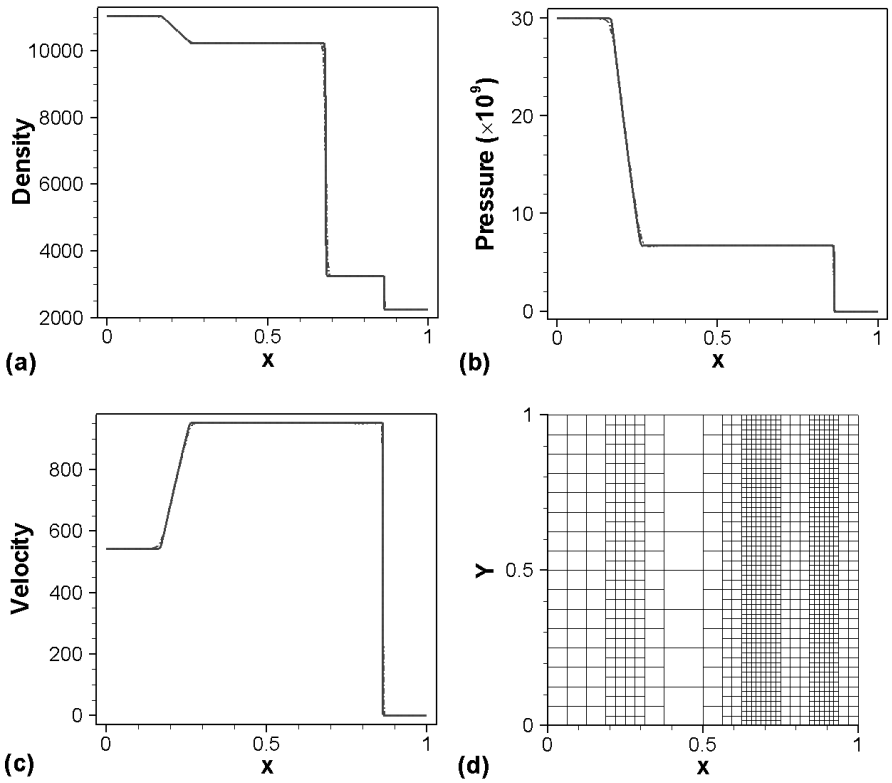

Figure 8: Results of the interaction between a shock in molybdenum and molybdenum-MORB interface at $t=120\mu s$.

sharply. This case demonstrates that our method can handle more complex 1D shock-interface interaction problem.

Example 5.6. As the last 1D case, we consider a liquid-gas shock tube problem to show that the present method can handle van der Waals and stiffened gas EOSs. At initial time, an interface is located at $x=0.7$. A liquid modeled by stiffened gas EOS is on the left of the interface with the state variables $(\rho,p;\gamma,\pi)=(10^3\text{kg/m}^3,10^9\text{Pa};4.4,6\times10^8\text{Pa})$. A van der Waals gas under atmospheric pressure is on the right of the interface and its state variables read, $(\rho,p;\gamma,a,b)=(50\text{kg/m}^3,10^5\text{Pa};1.4,5\text{Pam}^6/\text{kg},10^{-3}\text{m}^3/\text{kg})$. Both fluids are at rest.

Results at $t=240\mu s$ are presented in Fig. 9. All features of interest are resolved well. This problem developed by Shyue [15] is an extension of two-phase shock tube problem of Saurel and Abgrall [13] where gas is modeled by ideal gas EOS. In [13], small oscillations in density are present on the interface and attributed to stiffness of this problem arising from high pressure and density ratios across the interface. However, in the present solution, these oscillations are eliminated. In this case, 8 refinement levels are used and the grid size at the highest level is 1536×1536.

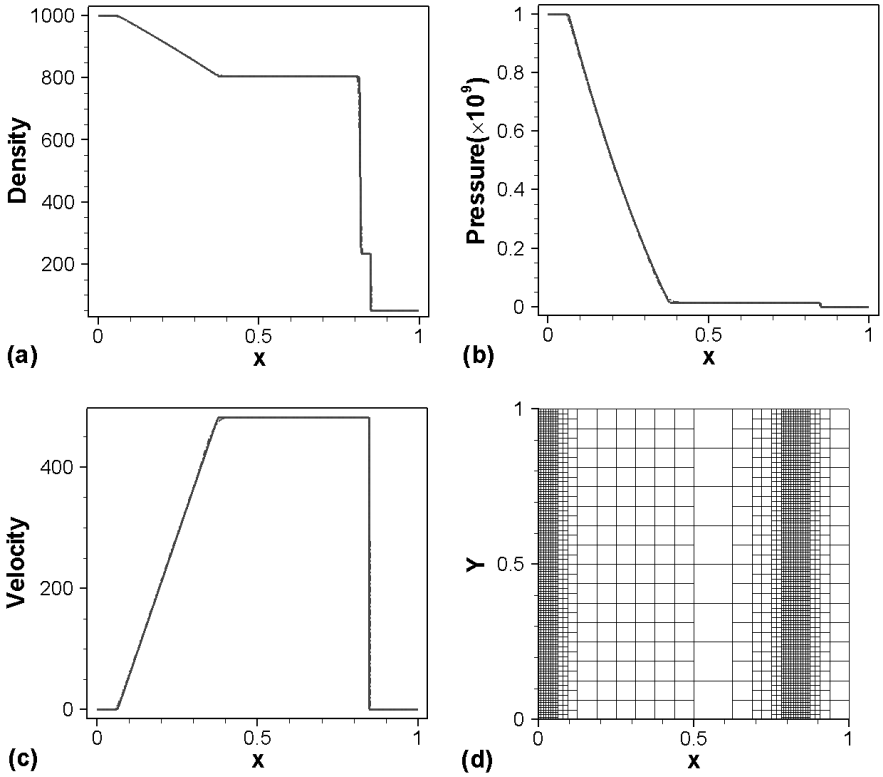

Figure 9: Results of a liquid-gas shock tube problem at $t=240\mu s$.

Example 5.7. As the first 2D example, this case deals with interaction of a shock in air with a helium bubble. We set up the simulation as follows. A helium bubble of radius 2cm is suspended in a rectangular domain of $[0,32]\times[0,8]\text{cm}^2$ with its center located at $(x,y)=(30,4)$cm. Initially, both helium and air are stationary under atmospheric pressure. Helium is lighter than air and their densities are $0.167\text{kg}/\text{m}^3$ and $1.29\text{kg}/\text{m}^3$, respectively. They are both modeled by ideal gas EOS with ratios of specific heat, $\gamma_{\text{helium}}=1.67$, $\gamma_{\text{air}}=1.4$. A Mach 1.2 shock is on the right of the bubble and is to collide with the bubble from right to left. On the left and right sides, non-reflecting boundary conditions are employed, while the reflecting boundary conditions are imposed on top and bottom.

Schlieren images of density field in Fig. 10 show time evolution of helium bubble. After the incident shock interacts with the bubble, there will be a transmitted shock in helium, which is faster than the incident one in air. At $t=50\mu s$, the transmitted shock has passed the bubble, see the first frame in Fig. 10. The doubly transmitted shock and incident one merge eventually, see frames at $400\mu s$ and $450\mu s$. Passage of the incident shock leads to baroclinic vorticity generation along the bubble interface. On a given point of the interface, density gradient is in the radial direction, while pressure gradient is approximately in the horizontal direction after shock passage. Misalignment of the density and

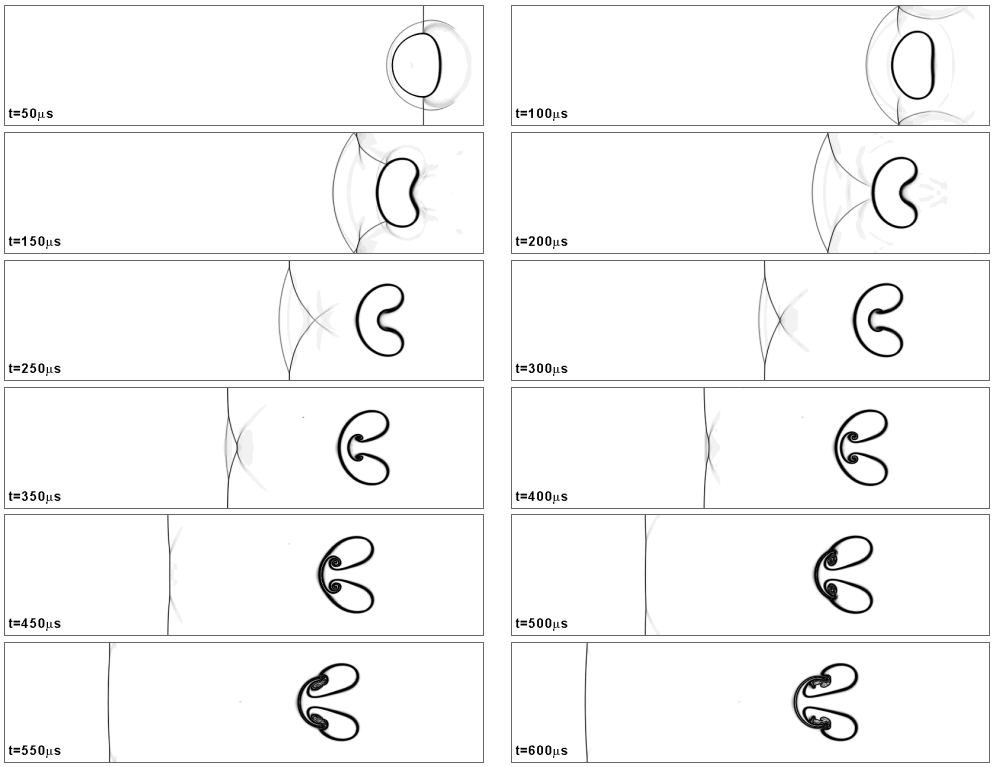

Figure 10: Schlieren images of density field for shock-helium bubble interaction.

pressure gradients gives rise to baroclinic vorticity, which is believed to drive the bubble distortion. In addition, a jet forms along centerline of the bubble and moves faster than the surrounding gas. This jet hits the left side of bubble and the bubble is eventually split into two vortex rings at late times. All these findings are consistent with those in [9]. It is observed that oscillations are not present on the interface. This simulation shows that the present method can successfully resolve problem involving complex interaction of shock, material interface and other structures.

In this case, a grid with 7 refinement levels is used. The equivalent grid size at the highest level is 3072×768. Use of AMR greatly reduces computational cost.

Example 5.8. We consider a model underwater explosion problem of Shyue [17]. The computational domain is a rectangle of $[-2,2]\times[-1.5,1.5]\text{m}^2$. An air-water interface is located at $y=0$. Air, modeled by ideal gas EOS, is above the interface with state variables $(\rho,p;\gamma)=(1.225\text{kg}/\text{m}^3,1.01325\times10^5\text{Pa};1.4)$. Water is modeled as a stiffened liquid and its initial data read $(\rho,p;\gamma,\pi)=(10^3\text{kg}/\text{m}^3,1.01325\times10^5\text{Pa};4.4,6\times10^8\text{Pa})$. In water, these is a pressurized air bubble of radius 0.12m with its center located at $(x,y)=(0,-0.3)\text{m}$. Air inside the bubble has density of $1250\text{kg}/\text{m}^3$ and pressure of 10^9Pa. Initially, both fluids are at rest. The reflecting boundary conditions are imposed on bottom of the domain, while non-reflecting boundary conditions are used on the remaining sides.

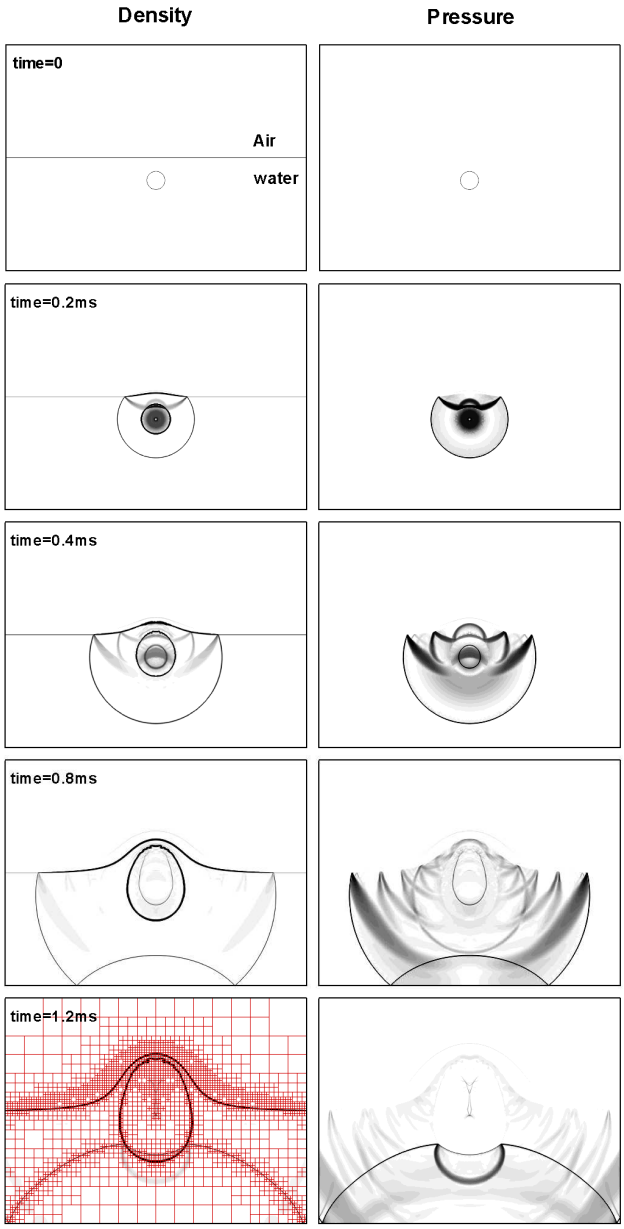

Figure 11: Schlieren images of density and pressure fields for the underwater explosion problem.

Due to high pressure ratio across the air bubble (about 10^4) and high density ratio across air-water interface (about 816), this problem is not easy to simulate. The schlieren images of the density and pressure of this run are illustrated in Fig. 11. Following breaking of air bubble, a strong shock is transmitted to water, moving radially outward and a rarefaction is generated inside the bubble. The radially-moving shock interaction with

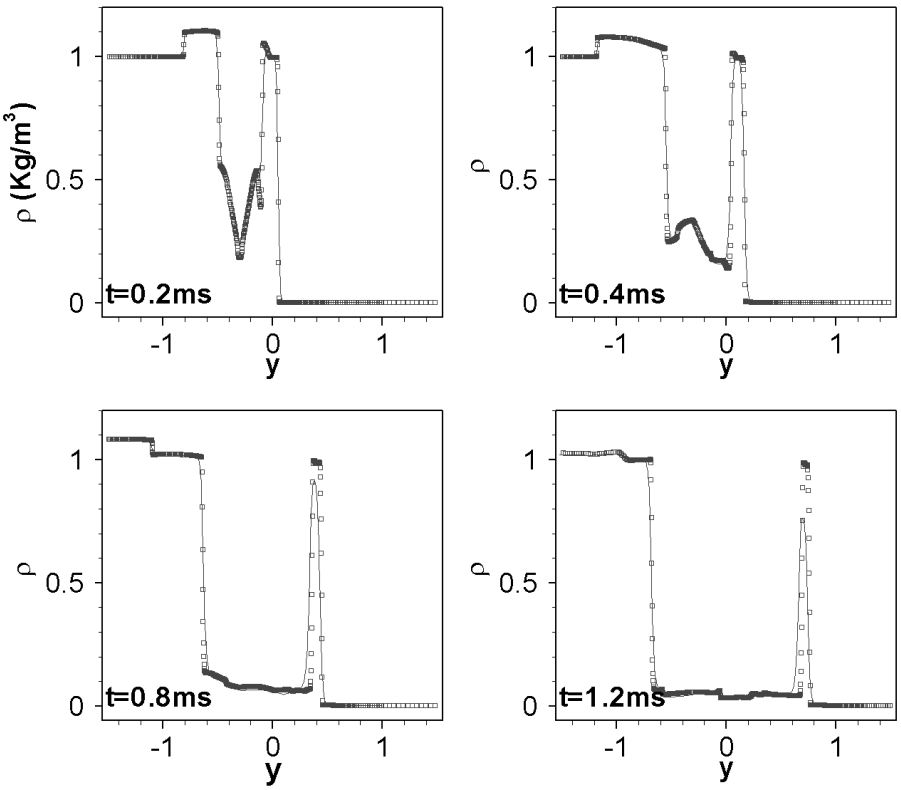

Figure 12: Density profiles along $x=0$ for the problem in Fig. 11. Squares and solid lines are obtained using the present and Shyue's methods, respectively.

the air-water surface, in turn, generates a transmitted shock in air and a reflected rarefaction in water, as shown in the second row of Fig. 11. The reflected rarefaction impacts upper side of the bubble, leading to acceleration of the bubble in the vertical direction and deformation of bubble shape, see the last three rows in Fig. 11. In addition, as the air bubble moves upwards, the air-water surface is also accelerated in the vertical direction and deforms. Fig. 12 provides a comparison of the density distributions along $x=0$ in the present results and those of Shyue [17]. It can be seen that the two sets of results are in good agreement. The differences in the right-most peaks in the last two frames of Fig. 12 stem from the fact the Shyue's method did not correctly resolve these flow features.

A grid with 6 refinement levels is used with the equivalent grid size of 1536×1152 at the highest level. Outlines of AMR blocks at $t=1.2ms$ are overlaid in Fig. 11. The material interfaces in the present results are sharper than those in [17], which is the consequence of use of AMR.

Example 5.9. Finally, we are concerned with interaction of a shock in molybdenum with a block of encapsulated MORB [16]. The computational domain is chosen to be an unit square. A Mach 1.163 rightward-moving shock is located at $x=0.3$ and to impact MORB

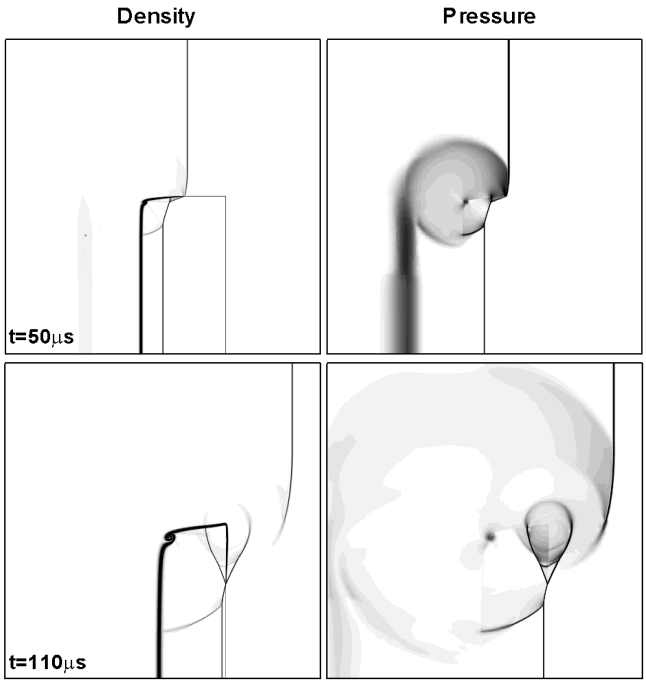

Figure 13: Schlieren images of density and pressure fields for shock interaction with a block of MORB.

contained in a rectangle of $[0.4,0.7]\times[0,0.5]$. Both materials are modeled by shock wave EOS. The initial conditions and material parameters are identical to those in Example 5.5. Reflecting boundary conditions are imposed on bottom and non-reflecting boundary conditions are used on the other three sides.

Schlieren images of density and pressure at two selected times are illustrated in Fig. 13. In density plot at $t=50\mu s$, we can observe the incident shock in molybdenum and transmitted shock in MORB with the former moving faster than the latter. By $t=110\mu s$, the transmitted shock has not passed the MORB block completely. The structure of diffraction of the shock by MORB is well captured in the pressure plots and there are no pressure oscillations on the interface. These findings are consistent with those in [16]. In this case, the highest refinement level in AMR is 8 and the equivalent grid size is 1536×1536. Therefore, the material interfaces are sharper than those in [16] obtained using a 200×200 grid. All these problems, i.e., Examples 5.1 to 5.9 can be simulated successfully, which proves that our method is very stable and robust.

6 Conclusion

We present a simple interface-capturing method for resolving compressible two-fluid flows with general EOS. Six types of equations of state are used and therefore a wide

range of real materials can be modeled. This is a remarkable advantage of our method. A key ingredient of this method is to generalize a modified HLLC Riemann solver to compute numerical fluxes including those of the advection equation. Based on the Riemann problem solution, the volume fraction is updated properly and the resulting scheme is very stable and robust under different situations. In addition, HLLC solver is time-efficient. MUSCL-Hancock scheme is extended to construct input states for Riemann problems. The popular AMR capability is built into hydrodynamic algorithm and thus various flow features at disparate scales can be resolved with appropriate grid resolution while computational cost is not increased significantly. Our method is easy to implement and numerical results show that it can be applied to many real problems.

References

[1] S. Abarbanel and D. Gottlieb, Optimal time splitting for two- and three-dimensional navier-stokes equations with mixed derivatives, J. Comput. Phys., 41 (1981), 1–33.
[2] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach, J. Comput. Phys., 125 (1996), 150–160.
[3] G. Allaire, S. Clerc and S. Kokh, A five-equation model for the simulation of interfaces between compressible fluids, J. Comput. Phys., 181 (2002), 577–616.
[4] K. R. Bates, N. Nikiforakis and D. Holder, Richtmyer-Meshkov instability induced by the interaction of a shock wave with a rectangular block of SF_6, Phys. Fluids, 19 (2007), 036101-1–036101-16.
[5] M. J. Berger and J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys., 53 (1984), 484–512.
[6] M. J. Berger and P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys., 82 (1989), 64–84.
[7] B. Fryxell et al., Flash: an adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes, ApJS, 131 (2000), 273–334.
[8] E. Johnsen and T. Colonius, Implementation of WENO schemes in compressible multicomponent flow problems, J. Comput. Phys., 219 (2006), 715–732.
[9] G. Layes and O. Le Métayer, Quantitative numerical and experimental studies of the shock accelerated heterogeneous bubbles motion, Phys. Fluids, 19 (2007), 042105(1)–13.
[10] R. Löhner, An adaptive finite element scheme for transient problems in CFD, Comput. Methods Appl. Mech. Engrg., 61 (1987), 323–338.
[11] P. MacNeice, K. M. Olson, C. Mobarry, R. D. Fainchtein and C. Packer, PARAMESH: A parallel adaptive mesh refinement community toolkit, Comput. Phys. Commun. , 126 (2000), 330–354.
[12] J. X. Qiu, B. C. Khoo and C. W. Shu, A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes, J. Comput. Phys., 212 (2006), 540–565.
[13] R. Saurel and R. Abgrall, A simple method for compressible multifluid flows, SIAMJ. Sci. Comput., 21(3) (1999), 1115–1145.
[14] R. Saurel and R. Abgrall, A multiphase Godunov method for compressible multifluid and multiphase flows, J. Comput. Phys., 150 (1999), 425–467.
[15] K. M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with van der waals equation of state, J. Comput. Phys., 156 (1999), 43–88.

[16] K. M. Shyue, A fluid-mixture type algorithm for compressible multicomponent flow with Mie-Grüneisen equation of state, J. Comput. Phys., 178 (2001), 678–707.
[17] K. M. Shyue, A wave-propagation based volume tracking method for compressible multicomponent flow in two space dimensions, J. Comput. Phys., 215 (2006), 219–244.
[18] G. Strang, On the construction and comparison of difference schemes, SIAMJ. Numer. Anal., 5(3) (1968), 506-517.
[19] M. Sun and K. Takayama, Conservative smoothing on an adaptive quadrilateral grid, J. Comput. Phys., 150 (1999), 143–180.
[20] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: A practical introduction, Springer-Verlag: Berlin Heidelberg, 1999.
[21] 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.
[22] B. van Leer, Upwind and high-resolution methods for compressible flow: from donor cell to residual-distribution scheme, Commun. Comput. Phys., 1(2) (2006), 192–206.
[23] J. Wackers and B. Koren, A fully conservative model for compressible two-fluid flow, Int. J. Numer. Meth. Fluids, 47 (2005), 1337–1343.
[24] W. Q. Zhang and A. I. MacFadyen, Ram: a relativistic adaptive mesh refinement hydrodynamics code, ApJS, 164 (2006), 255–279.
[25] J. G. Zheng, T. S. Lee and D. J. Ma, A piecewise parabolic method for barotropic two-fluid flows, Int. J. Mod. Phys. C, 18(3) (2007), 375–390.
[26] J. G. Zheng, T. S. Lee and S. H. Winoto, A piecewise parabolic method for barotropic and nonbarotropic two-fluid flows, Int. J. Numer. Methods Heat Fluid Flow, 18(6) (2008), 708–729.