Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **9**, No. 1, pp. 73-91

DOI: 10.4208/aamm.2015.m1070
February 2017

Runge-Kutta Discontinuous Galerkin Method with Front Tracking Method for Solving the Compressible Two-Medium Flow on Unstructured Meshes

Haitian Lu[1], Jun Zhu[2], Chunwu Wang[2] and Ning Zhao[1,*]

[1] *College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China*
[2] *College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China*

Received 28 April 2015; Accepted (in revised version) 18 December 2015

Abstract. In this paper, we extend using the Runge-Kutta discontinuous Galerkin method together with the front tracking method to simulate the compressible two-medium flow on unstructured meshes. A Riemann problem is constructed in the normal direction in the material interfacial region, with the goal of obtaining a compact, robust and efficient procedure to track the explicit sharp interface precisely. Extensive numerical tests including the gas-gas and gas-liquid flows are provided to show the proposed methodologies possess the capability of enhancing the resolutions nearby the discontinuities inside of the single medium flow and the interfacial vicinities of the two-medium flow in many occasions.

AMS subject classifications: 65M60, 65M99, 35L65

Key words: Runge-Kutta discontinuous Galerkin method, front tracking method, two-medium flow, Riemann problem, unstructured mesh.

1 Introduction

The algorithms for simulating the compressible two-medium flow are usually consisted of two parts: one is to solve the single-medium flow precisely and the other is to treat the material interface accurately. On the one hand, in recent years, the discontinuous Galerkin (DG) method has been a research hotspot in the simulations of the single medium flow. The original DG method was introduced by Reed and Hill [27] for solving the linear equations in the framework of neutron transport. A major development of

*Corresponding author.
Email: lhtgkzy@126.com (H. T. Lu), zhujun@nuaa.edu.cn (J. Zhu), wangcw@nuaa.edu.cn (C. W. Wang), nzhao2000@hotmail.com (N. Zhao)

the DG method was carried out by Cockburn and Shu in a series of papers [5–9]. They employed the total variation diminishing (TVD) high order Runge-Kutta time discretization [29] and DG discretization in space with exact or approximate Riemann solvers as interface fluxes and a total variation bounded (TVB) limiter [28] to achieve non-oscillatory property for strong shocks. These methods are termed as Runge-Kutta discontinuous Galerkin (RKDG) methods. On the other hand, a relatively dominant part is the treatment of the moving material interface and its vicinity for simulating compressible two-medium flow. Early algorithms have treated the material interfaces with the γ-based model [17], the mass fraction model [1,18] or a level set function [21,23]. These algorithms, based on shock capturing methods, always yield a numerical diffusion of contact discontinuities over several nodes. However, for the front tracking method [3,12–14], fluid interfaces are explicitly tracked by connected marker points and a sharp interface boundary is maintained during the computation. The ghost fluid method (GFM) introduced by Fedkiw et al. [2,10,11] presents a fairly simple and flexible way to treat multi-medium flows. However, when the pressure or the velocity experiences a large gradient across the interface, the GFM may cause numerical inaccuracy. Indeed, the ghost fluid states should consider the influence of both wave interaction and material properties on the interfacial evolution. This leads to the proposal of improved versions of GFM, for example, the real ghost fluid method (RGFM) [33]. The RGFM predicts the flow states for the real fluid nodes just next to the interface because wave interaction at the interface can propagate upward and downward simultaneously.

So far, there is less work related to a DG method coupled to the GFM [25,34] technique for two-medium flow simulations and the corresponding method to track the interface is almost on structured meshes. In this paper, our major intention is to investigate the performance of the RKDG method combined with the front tracking method to solve for the compressible two-medium flow on unstructured meshes. The RGFM is used to define the interface boundary conditions. Unlike interpolating from the fixed grids to obtain the interface velocity [32], we propose the method based on the Riemann problem constructed in the normal direction of the interface to determine the interface motion, and the corresponding Riemann solutions are also used directly to update the real fluid states in the RGFM. In the earlier works of RGFM [33] based on the uniform structured meshes, the cell center states were usually used as the initial conditions to the Riemann problem. However, it may cause some inaccuracies for the unstructured meshes since the mesh size may vary acutely near the material interface. Due to the explicit tracked interface in this paper, we can easily select the initial conditions to the Riemann problem in the same distances in the normal direction of the interface so that the geometrical influences of the triangular unstructured meshes are avoided.

Except the better adaptability with the complex boundary of the unstructured meshes, the main purposes to use the RKDG method in the simulation of the two-medium flow are in the following: firstly, the higher order accuracy can be obtained in smooth regions easily. Secondly, the initial conditions to the Riemann problem are obtained directly from the solution polynomials in the RKDG method, in contrast to the

interpolation used in the finite volume method. Thirdly, the normal vectors and the ghost fluid states on the ghost cells far from the interface are less accuracy in the front tracking method. However, the good compactness of the RKDG method makes it flexible in dealing with material interface since we only need to consider the ghost cells which have the common edge with the real fluid cells.

The paper is organized as follows. In Section 2 and Section 3, we describe the RKDG method with front tracking method for solving the two-medium flow on unstructured meshes and give the algorithms of advancing the material interface in detail. Extensive numerical examples are presented in Section 4 to verify the effectiveness of these procedures. Concluding remarks are then given in Section 5.

2 Equations and numerical method

2.1 Governing equations

We consider the two dimensional hyperbolic conservation laws

$$\frac{\partial \vec{U}}{\partial t}+\nabla\cdot\vec{F}(\vec{U})=0, \tag{2.1}$$

where $\vec{U}=[\rho,\rho u,\rho v,E]^T$, $\vec{F}(\vec{U})=[\vec{F}_1(\vec{U}),\vec{F}_2(\vec{U})]$, $\vec{F}_1(\vec{U})=[\rho u,\rho u^2+p,\rho uv,(E+p)u]^T$, $\vec{F}_2(\vec{U})=[\rho v,\rho uv,\rho v^2+p,(E+p)v]^T$. Here, ρ is the density, u and v are the velocities of different directions, p is the pressure, E is the total energy per unit volume. The total energy is given as

$$E=\rho e+\rho(u^2+v^2)/2, \tag{2.2}$$

where e is the internal energy per unit mass.

For closure of the system, the equation of state (EOS) is required. The EOS for gas or liquid medium can be written uniformly as

$$p=(\gamma-1)\rho e-\gamma B, \tag{2.3}$$

where γ and B are treated as fluid constants, and will be specified in Section 4.

2.2 The implementation of the RKDG method

In this section, we will briefly describe the RKDG method on unstructured meshes [9]. The coupling with the interface treatment will be presented in the next section.

Consider the triangular cell K_0, replace the exact solution $\vec{U}$ by the numerical solution $\vec{U}^h$, the numerical solution as well as the test function space is given by $V_h^k=\{p:p|_{K_0}\in$

$P^k(K_0)\}$, where $P^k(K_0)$ is the space of polynomials with degree no greater than k on the cell K_0. We use the expression for the numerical solution $\vec{U}^h(x,y,t)$ inside the cell K_0,

$$\vec{U}^h_{K_0}(x,y,t)=\sum_{i=1}^{N}\vec{U}^{(i)}_{K_0}(t)\phi_i(x,y) \quad \text{for } (x,y)\in K_0. \tag{2.4}$$

For the P^1 case, the degrees of freedom $\vec{U}^{(i)}_{K_0}(t)$ are the values of the numerical solution at the midpoints of edges, and the basis function $\phi_i(x,y)$ is the linear function which takes the value 1 at the midpoint of the i-th edge, and the value 0 at the midpoints of the other two edges. For the P^2 case, the degrees of freedom $\vec{U}^{(i)}_{K_0}(t)$ are the values of the numerical solution at the three midpoints of edges and the three vertices. The basis function $\phi_i(x,y)$ is the quadratic function which takes the value 1 at the point i of the six points mentioned above, and the value 0 at the other five points.

By multiplying (2.1) with the test function, integrating over the cell K_0 by parts, and substituting (2.4) into (2.1), we can obtain the governing Galerkin weak form equations for the degrees of freedom as

$$\begin{aligned} M\frac{dU_{K_0}(t)}{dt}=&-\sum_{e\in\partial K_0}\int_e \overrightarrow{F}(\overrightarrow{U}^h(x,y,t))\cdot\overrightarrow{n}_{e,K_0}\phi(x,y)d\Gamma \\ &+\int_{K_0}\overrightarrow{F}(\overrightarrow{U}^h(x,y,t))\cdot grad\phi(x,y)dxdy, \end{aligned} \tag{2.5}$$

where

$$M_{N\times N}=\Big(\int_{K_0}\phi_i(x,y)\phi_j(x,y)dxdy\Big)_{ij}$$

is the mass matrix (see [9] for details), $U_{K_0}(t)=[\overrightarrow{U}^{(1)}_{K_0},\overrightarrow{U}^{(2)}_{K_0},\cdots,\overrightarrow{U}^{(N)}_{K_0}]^T$, $\phi(x,y)=[\phi_1(x,y),\phi_2(x,y),\cdots,\phi_N(x,y)]^T$ and $\overrightarrow{n}_{e,K_0}$ is the outward unit normal to edge e of cell K_0. The integrals are approximated by a Gaussian quadrature formula with sufficient accuracy as

$$\int_e\vec{F}(\vec{U}^h(x,y,t))\cdot\overrightarrow{n}_{e,K_0}\phi(x,y)d\Gamma\approx\sum_{l=1}^{L}\omega_l\vec{F}(\vec{U}^h(x_{el},y_{el},t))\cdot\overrightarrow{n}_{e,K_0}\phi(x_{el},y_{el})|e|, \tag{2.6a}$$

$$\int_{K_0}\vec{F}(\vec{U}^h(x,y,t))\cdot grad\phi(x,y)dxdy\approx\sum_{s=1}^{S}\bar{\omega}_s\vec{F}(\vec{U}^h(x_{K_0s},y_{K_0s},t))\cdot grad\phi(x_{K_0s},y_{K_0s})|K_0|. \tag{2.6b}$$

See [5,9] for details of the numerical quadrature formulas. The flux $\vec{F}(\vec{U}^h(x_{el},y_{el},t))\cdot\overrightarrow{n}_{e,K_0}$ in the boundary integral is replaced by the simple Lax-Friedrichs flux

$$\begin{aligned} \vec{F}(\vec{U}^h(x_{el},y_{el},t))\cdot\overrightarrow{n}_{e,K_0}\approx&\frac{1}{2}[(\vec{F}(\vec{U}^+(x_{el},y_{el},t))+\vec{F}(\vec{U}^-(x_{el},y_{el},t)))\cdot\overrightarrow{n}_{e,K_0} \\ &-\alpha(\vec{U}^+(x_{el},y_{el},t)-\vec{U}^-(x_{el},y_{el},t))], \end{aligned} \tag{2.7}$$

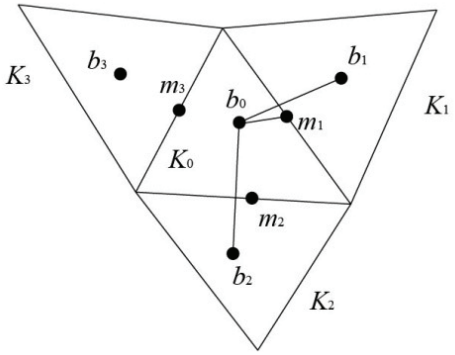


Figure 1: Illustration of slope limiter.

where α is the biggest absolute value of the eigenvalues of the Jacobian matrix in the $\overrightarrow{n}_{e,K_0}$ direction, and $\vec{U}^{\pm}(x_{el},y_{el},t)$ refer to the values of $\vec{U}^h$ in the neighboring cell and in the current cell K_0, respectively.

The semi-discrete scheme (2.5) can be generalized as

$$\overrightarrow{U}_t = L(\overrightarrow{U}), \tag{2.8}$$

which is discretized by a third-order TVD Runge-Kutta time discretization [29]

$$\overrightarrow{U}^{(1)} = \overrightarrow{U}^n + \Delta t L(\overrightarrow{U}^n), \tag{2.9a}$$

$$\overrightarrow{U}^{(2)} = \frac{3}{4}\overrightarrow{U}^n + \frac{1}{4}\overrightarrow{U}^{(1)} + \frac{1}{4}\Delta t L(\overrightarrow{U}^{(1)}), \tag{2.9b}$$

$$\overrightarrow{U}^{n+1} = \frac{1}{3}\overrightarrow{U}^n + \frac{2}{3}\overrightarrow{U}^{(2)} + \frac{2}{3}\Delta t L(\overrightarrow{U}^{(2)}). \tag{2.9c}$$

If there are strong discontinuities in solution, spurious oscillations may occur. Thus, a local slope limiter is used to overcome such difficulty. Consider the triangles in Fig. 1, where m_i is the midpoint of the edges on the boundary of K_0 for $i=1,2,3$, and b_i denotes the barycenter of the triangles K_i for $i=0,1,2,3$. Since we have

$$\begin{aligned} x_{m_1} - x_{b_0} &= \alpha_1(x_{b_1} - x_{b_0}) + \alpha_2(x_{b_2} - x_{b_0}), \\ y_{m_1} - y_{b_0} &= \alpha_1(y_{b_1} - y_{b_0}) + \alpha_2(y_{b_2} - y_{b_0}), \end{aligned}$$

for nonnegative coefficients α_1, α_2 and

$$\overline{U}_{K_i} = \frac{1}{K_i}\int_{K_i} \overrightarrow{U}^h dxdy, \quad i=0,1,2,3,$$

we denote

$$\overline{\overrightarrow{U}}^h(m_1,K_0) \equiv \overrightarrow{U}^h_{K_0}(m_1) - \overline{U}_{K_0}, \tag{2.10a}$$

$$\Delta\overline{U}(m_1,K_0) \equiv \alpha_1(\overline{U}_{K_1} - \overline{U}_{K_0}) + \alpha_2(\overline{U}_{K_2} - \overline{U}_{K_0}). \tag{2.10b}$$

It is similar for the other two midpoints of the cell K_0. Consider a piecewise linear function $\vec{U}^h$,

$$\vec{U}^h_{K_0}(x,y,t)=\sum_{i=1}^{3}\vec{U}^h_{K_0}(m_i)\phi_i(x,y)=\overline{U}_{K_0}+\sum_{i=1}^{3}\overline{\overline{U}}^h(m_i,K_0)\phi_i(x,y)\quad\text{for }(x,y)\in K_0,\tag{2.11}$$

we first compute the quantities:

$$\Delta_i=\overline{m}(\overline{\overline{U}}^h(m_i,K_0),\nu\Delta\overline{U}(m_i,K_0)),\tag{2.12}$$

here we take $\nu=1.5$ and $\overline{m}$ is a modified minmod function

$$\overline{m}(a_1,a_2)=\begin{cases}a_1, & \text{if } |a_1|\le M|K_0|,\\ s\cdot\min(|a_1|,|a_2|), & \text{if } s=sign(a_1)=sign(a_2),\\ 0, & \text{otherwise,}\end{cases}\tag{2.13}$$

where M is the TVB limiter constant. Then if $\sum_{i=1}^{3}\Delta_i=0$, we simply set

$$\vec{U}^h_{K_0}(x,y,t)=\overline{U}_{K_0}+\sum_{i=1}^{3}\Delta_i\phi_i(x,y).\tag{2.14}$$

If $\sum_{i=1}^{3}\Delta_i\neq 0$, we compute

$$pos=\sum_{i=1}^{3}\max(0,\Delta_i),\qquad neg=\sum_{i=1}^{3}\max(0,-\Delta_i),$$
$$\theta^+=\min\Big(1,\frac{neg}{pos}\Big),\qquad \theta^-=\min\Big(1,\frac{pos}{neg}\Big),$$

then

$$\vec{U}^h_{K_0}(x,y,t)=\overline{U}_{K_0}+\sum_{i=1}^{3}\overline{\Delta}_i\phi_i(x,y),$$

where

$$\overline{\Delta}_i=\theta^+\max(0,\Delta_i)-\theta^-\max(0,-\Delta_i).\tag{2.15}$$

For the system cases, the limiter is always used with a local characteristic field decomposition in the $\overrightarrow{b_0b_i}$, $(i=1,2,3)$ direction. See, e.g., [9] for details.

3 The interface treatment

In this section we describe the method to track the interface and determine the ghost fluid region in detail. Moreover, the RGFM [33] is described here briefly when coupled to the RKDG method.

3.1 Tracking fluid interface

As indicated in Fig. 2, medium 1 and medium 2 are separated by the fluid interface at time t^n. The marker points are intersections of the interface with the grid lines. $\vec{N}_P=(N_{Px},N_{Py})$ is the unit normal vector of marker point $P(x_P,y_P)$ and its orientation is always from medium 1 to medium 2. Point $A(x_A,y_A)$ and point $B(x_B,y_B)$ are in different mediums, obtained by the same distance h from the marker point P (h can be the maximum circumscribed diameter of triangular meshes)

$$x_A=x_P+h\cdot N_{Px},y_A=y_P+h\cdot N_{Py},$$
$$x_B=x_P-h\cdot N_{Px},y_B=y_P-h\cdot N_{Py}.$$

The state vectors at point A and point B can be obtained directly from the solution polynomials in the RKDG method

$$\vec{U}_A=\sum_{i=1}^{N}\vec{U}_{K_A}^{(i)}(t)\phi_i(x_A,y_A), \tag{3.1a}$$

$$\vec{U}_B=\sum_{i=1}^{N}\vec{U}_{K_B}^{(i)}(t)\phi_i(x_B,y_B), \tag{3.1b}$$

where K_A and K_B are the grid cells containing point A and point B. The density, the normal velocity and the pressure at point A and point B are calculated, denoted by $\overrightarrow{W}_A=[\rho^A,u_N^A,p^A]^T$ and $\overrightarrow{W}_B=[\rho^B,u_N^B,p^B]^T$. Therefore we can construct a Riemann problem in the normal direction of marker point P with the initial conditions

$$\overrightarrow{W}_0=\begin{cases}\overrightarrow{W}_B,\\ \overrightarrow{W}_A.\end{cases} \tag{3.2}$$

An approximate Riemann problem solver (ARPS) based on a two shock structure [19] is applied to solve this Riemann problem. The Riemann solutions are denoted by $\overrightarrow{W}_P=[\rho_L^I,\rho_R^I,u_N^I,p^I]^T$, where the superscript "I" refers to the interface, and the subscript "L" and "R" denote the left and right side of the interface, respectively. As the tangential velocity of marker point P is not determined from the Riemann problem, the method proposed by Cocchi et al. [4] is used

$$v_T^I=\begin{cases}v_T^B, & \text{if } u_N^I\geq 0,\\ v_T^A, & \text{otherwise,}\end{cases} \tag{3.3}$$

where u_N^I and v_T^I are the normal and tangential velocities of marker point P, v_T^A and v_T^B are the tangential velocities of point A and point B. Once the velocity of each marker point has been solved, its new position is updated simultaneously with the governing

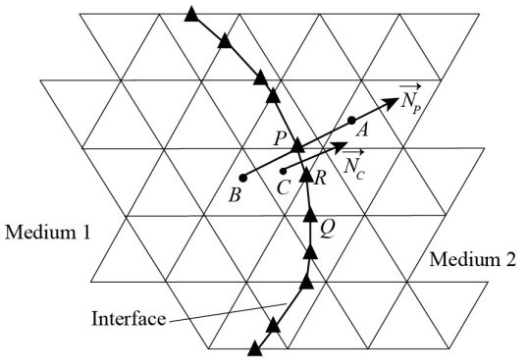


Figure 2: Construction of the Riemann problem at the marker point P.

equations by the same time integration scheme

$$\vec{x}_f^{(1)} = \vec{x}_f^n + \Delta t \cdot \overrightarrow{v}_f(\vec{x}_f^n), \tag{3.4a}$$

$$\vec{x}_f^{(2)} = \frac{3}{4}\vec{x}_f^n + \frac{1}{4}\vec{x}_f^{(1)} + \frac{1}{4}\Delta t \cdot \overrightarrow{v}_f(\vec{x}_f^{(1)}), \tag{3.4b}$$

$$\vec{x}_f^{n+1} = \frac{1}{3}\vec{x}_f^n + \frac{2}{3}\vec{x}_f^{(2)} + \frac{2}{3}\Delta t \cdot \overrightarrow{v}_f(\vec{x}_f^{(2)}), \tag{3.4c}$$

where $\vec{x}_f^n$ and $\vec{x}_f^{n+1}$ are the positions of marker points at time t^n and t^{n+1}, respectively. $\vec{v}_f$ is the interface velocity vector, and Δt is the time step.

Reconstructing the interface on triangular meshes is similar with uniform structured meshes. The robust grid based method is used here: we connect the marker points at the new positions and compute the intersections with the grid lines. These intersections are the new marker points at time t^{n+1} after reordering them. More details about reconstructing the interface methodology, see, e.g., [13].

3.2 Determining the ghost fluid region and normal vector

Since the Euler equations are solved in each fluid domain independently, it is required that we be able to identify the ghost fluid region near the interface. In many papers, this is accomplished by using the Heaviside function and solving a Poisson equation [15,31]. As is known, the front tracking method, compared to the level set method, should be more dependent on the interface geometrical information. Here we use a simpler method to determine the ghost fluid region. Consider the grid point C in Fig. 2. The normal vector of each marker point is always set to direct from medium 1 to medium 2. We first calculate the distance between the grid point C and the marker points nearby. The marker point P is selected if the minimum distance is obtained there. Next, we calculate the sign of the inner-product $\overrightarrow{r}_{PC} \cdot \overrightarrow{N}_P$, where $\overrightarrow{r}_{PC}$ is the position vector directing from point P to point C. If the sign is positive, then the grid point C is in the ghost fluid region of medium 1.

Otherwise, it is in the ghost fluid region of medium 2. It is similar for other grid points near the interface. The normal vector on the fixed grid cells near the interface is solved by area-weighting [32]. The normal vector of the grid cells far away from the interface need not to be computed due to the good compactness of the RKDG method.

3.3 Real ghost fluid method

In the RGFM [33], a Riemann problem is defined at the interface to update the flow states at the real fluid cells adjacent to the interface and the ghost fluid states are extrapolated by solving an advection equation. However, since the Riemann problem has been constructed when tracking fluid interface in Section 3.1, the Riemann solutions can be used directly. As indicated in Fig. 2, points R, P, Q are the marker points near the grid point C. $\overrightarrow{N}_P$ is the normal vector of marker point P and $\overrightarrow{N}_C$ is the normal vector of grid cell C. The flow states at the cell C can be updated by the marker point nearby. The marker point P is selected if the angle between $\overrightarrow{N}_P$ and $\overrightarrow{N}_C$ is the minimum compared with other marker points. Since in general the derivatives of physical solutions are discontinuous across the interface, we project the flow states at the marker point P (ρ_L^I, u_N^I, p^I) to obtain the average values in cell C while the tangential velocity in cell C remains unchanged. It is similar to update the flow states in other real fluid cells adjacent to the interface.

In order to extrapolate the updated flow states into the ghost fluid region, the following advection equation is used as

$$\frac{\partial \phi}{\partial t} \pm \overrightarrow{N} \cdot \nabla \phi = 0, \tag{3.5}$$

where ϕ represents the density, the normal velocity, the tangential velocity and the pressure in the ghost fluid region, $\overrightarrow{N}$ is the unit normal vector on the ghost cells. As indicated in Fig. 2, the "$+$" sign is used in (3.5) if the interface boundary conditions of medium 1 are to determine and the "$-$" sign is used in (3.5) if the interface boundary conditions of medium 2 are to determine. This advection equation is solved by iterating in fictitious time τ at each time step. Consider the compactness of the RKDG method, we only need to take 2$\sim$3 grid cells across the interface as the ghost fluid region. Once the advection equation has been solved in the ghost fluid region, we project the flow states to obtain the average ghost fluid values, and then the interface boundary conditions can be obtained.

3.4 Summary of the solution procedures

The implementation procedures are shown by the flowchart in Fig. 3.

4 Numerical results

In this section, several two dimensional compressible two-medium flow problems are simulated. The RKDG method combined with the front tracking method is named as

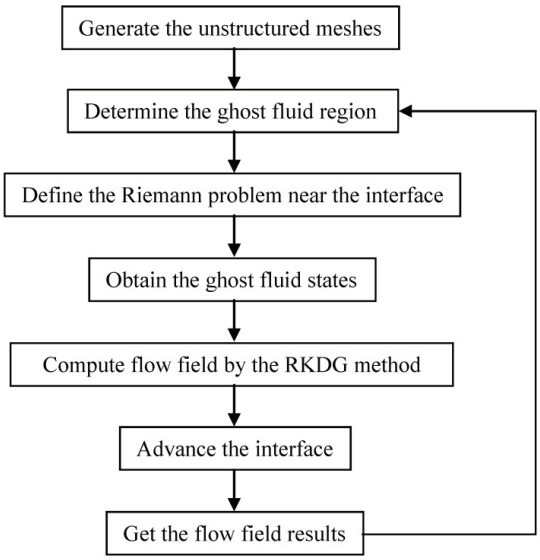


Figure 3: Flow chart for implementation procedures.

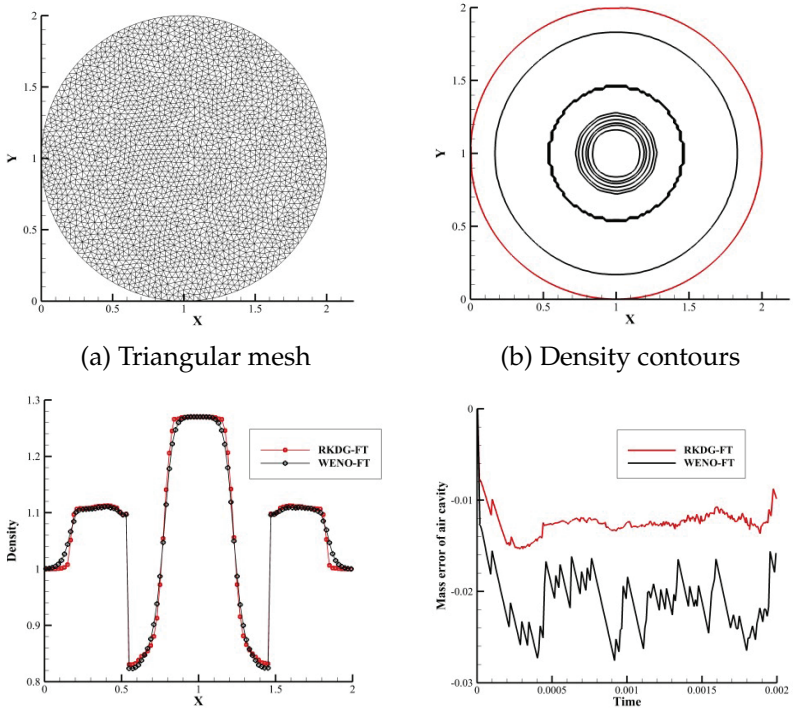
RKDG-FT method for convenience. The TVB limiter constant is taken as $M=0.1$. The software POINTWISE is used to generate the unstructured meshes.

Case 1: Sod shock tube

This first example is a simple sod shock tube problem and is used to investigate a high pressure air cavity expanding in water. We artificially design a circular computational domain centered at $(1,1)$ with a diameter of 2 in order to demonstrate the adaptability with the irregular domain boundary by the unstructured RKDG-FT method. Initially, a cylindrical air cavity is at the center of the domain with a diameter of 0.8. The initial conditions are taken as

$$(\rho,u,v,p,\gamma,B)=\begin{cases}(1.27,0,0,8000,1.4,0), & \text{inside the cylinder,}\\ (1,0,0,1,7.15,3309), & \text{outside the cylinder.}\end{cases} \tag{4.1}$$

The nonreflecting boundary condition is used at all the boundaries. The mesh we use is shown in Fig. 4(a). The computational domain is divided into 4746 mesh cells and the number of grid nodes is 2443. We use the P^2 RKDG method to compute this problem. The density contour at time $t=0.002$ is shown in Fig. 4(b) (the red line is the domain boundary). A shock is formed running towards the boundary of the domain and an expansive wave is running towards the center of the domain. Here we replace the RKDG method by the third order weighted essentially non-oscillatory (WENO) method and the WENO method combined with the front tracking method is named as WENO-FT method for convenience (see [20] for the implementation of the WENO-FT method in detail). We compare the density distributions along $y=1$ at time $t=0.002$ in Fig. 4(c). We can see that



(a) Triangular mesh (b) Density contours

(c) Density distribution along $y=1$ (d) Mass errors of air cavity

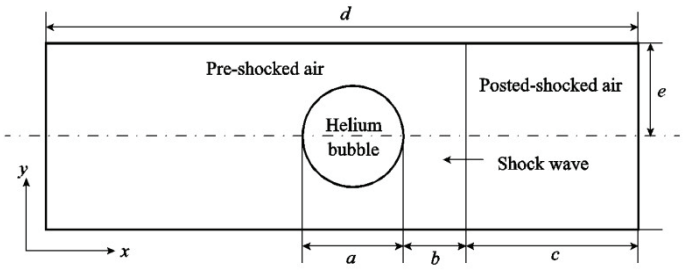
Figure 4: The numerical results for case 1.

these results are in a relative good agreement, including the interface position. However, better results are obtained with the RKDG-FT method, especially for the shock wave and expansive wave. The shock wave is sharper by the RKDG-FT method. Fig. 4(d) shows the mass errors of air cavity for these two methods. The mass error is defined by $E_m=(m_n-m_0)/m_0$, where m_n is the mass of air cavity at time t^n and m_0 is the initial mass of air cavity. It is found that there are mass losses of air cavity for both methods, however, the mass loss caused by the RKDG-FT method is smaller.

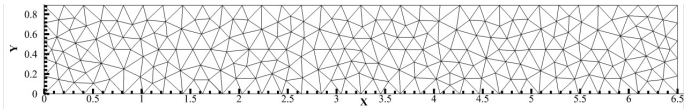
Case 2: Shock-bubble interaction

Shock wave interactions with bubbles have been studied experimentally and numerically by several authors [16,22,24,26,31]. The Hass and Sturtevant's experiment is simulated here. Fig. 5(a) shows the computational set-up and the geometrical parameters are: $a=50$mm, $b=25$mm, $c=100$mm, $d=325$mm, $e=44.5$mm. The cylindrical helium bubble is placed within the air. The shock wave propagates to the left and hits a helium bubble with a Mach number of 1.22.

Since the flow field is symmetric about the center axis, only the upper half domain is computed. On the left and right boundaries, nonreflecting boundary condition is used and the upper boundary is treated as slip-wall. The sound speed in the pre-shocked air

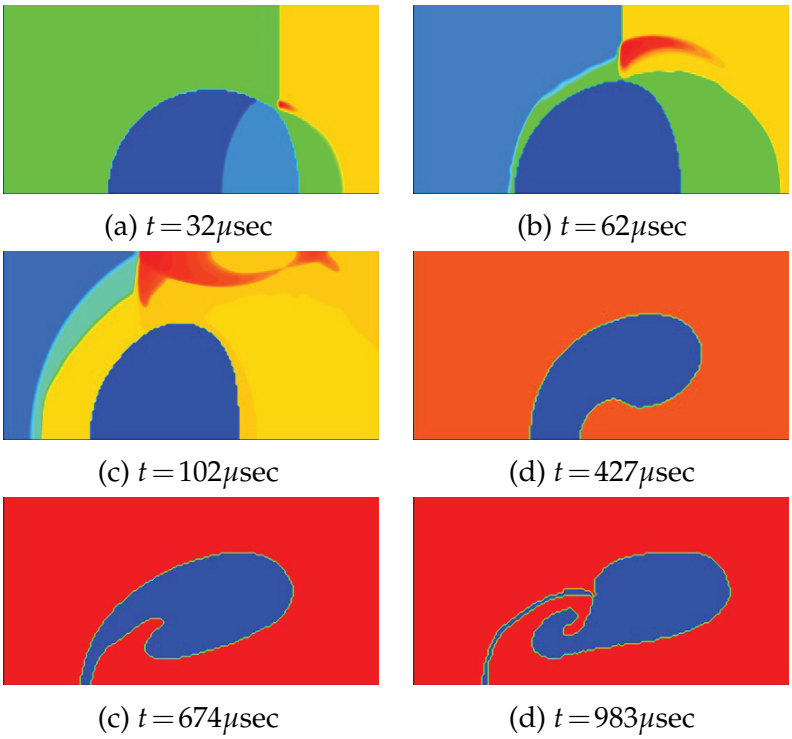


(a) A schematic of computational domain (not to scale)



(b) Sample mesh

Figure 5: Computational domain and sample mesh.



(a) $t=32\mu$sec (b) $t=62\mu$sec

(c) $t=102\mu$sec (d) $t=427\mu$sec

(c) $t=674\mu$sec (d) $t=983\mu$sec

Figure 6: The evolution of density field.

and the bubble diameter are used for the nondimensionalization. The non-dimensional initial conditions are: $\rho=1$, $u=0$, $v=0$, $p=1/1.4$, $\gamma=1.4$, $B=0$, for pre-shocked air,

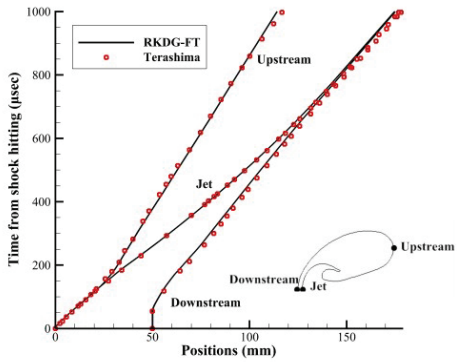


Figure 7: Space-time diagrams for three characteristic interface points.

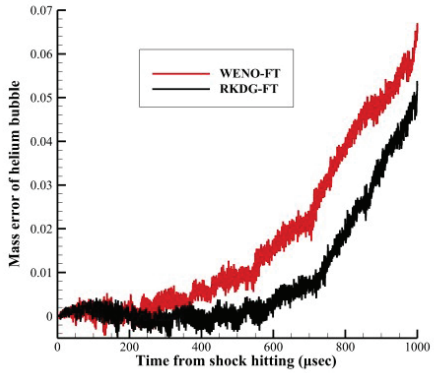


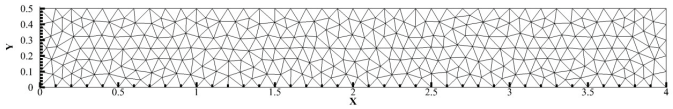
Figure 8: Comparison of mass errors of helium bubble.

$\rho=1.3764$, $u=-0.3336$, $v=0$, $p=1.5698/1.4$, $\gamma=1.4$, $B=0$, for post-shocked air, $\rho=0.1819$, $u=0$, $v=0$, $p=1/1.4$, $\gamma=1.648$, $B=0$, for helium. The computational domain is divided into 58364 mesh cells and the number of grid nodes is 29604. A sample mesh coarser than what is used in the actual computation is shown in Fig. 5(b). We use the P^2 RKDG method to compute this problem. Fig. 6 shows the time histories of density field. The timing is the same as in [16,31] for easy comparison. We can see that the general trend, including the bubble shape and the refracted shock wave, are in good qualitative agreement with the earlier results. Fig. 7 shows the space-time diagram for three characteristic points (Jet, Downstream, Upstream shown in the figure) compared with the earlier computational results from [31] during the early stage. In general, these results are in a relatively good agreement, showing the accuracy of the new method. In order to compare the RKDG-FT method with the WENO-FT method, the mass errors of helium bubble are calculated and presented in Fig. 8. It is found that the mass errors are limited within 7% before the helium bubble collapses for both methods. In general the mass error caused by the RKDG-FT method is smaller.

Case 3: Richtmyer-Meshkov instability

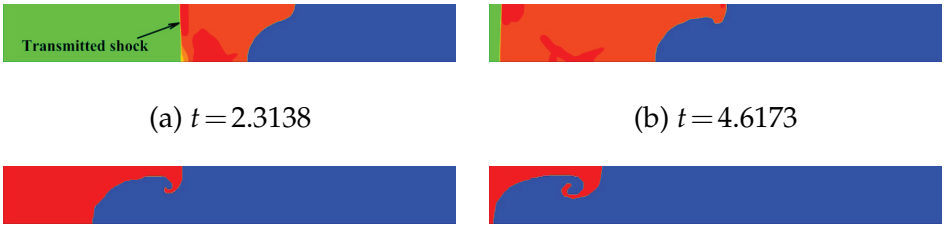
This example consists of two simulations with gas-gas and gas-liquid interfaces. It has been investigated by several authors to test and validate their methods before [22,30,31]. The first one is a gas-gas interface taken from [31]. Only the bottom half domain is computed since the flow field is symmetric about the center axis. As indicated in Fig. 9(a), a computational domain of $[0,4]\times[0,0.5]$ is used and we take a single mode perturbation of an air-SF_6 interface that the initial location of the interface is represented by: $x=2.9-0.1\sin(2\pi(y+0.25))$, $0<y<0.5$. The initial conditions are: $\rho=5.04$, $u=0$, $v=0$, $p=1$, $\gamma=1.093$, $B=0$, for SF_6, $\rho=1$, $u=0$, $v=0$, $p=1$, $\gamma=1.4$, $B=0$, for pre-shocked air, $\rho=1.411$, $u=-0.39$, $v=0$, $p=1.628$, $\gamma=1.4$, $B=0$, for post-shocked air. To trigger the instability, at $x=3.2$ there is a planar Mach number 1.24 shock wave in air propagating from the right to the left of the interface. The upper and lower boundaries are taken to be periodic and the nonreflecting boundary condition is applied at the left and right boundaries. The computational domain is divided into 105542 mesh cells and the number of grid nodes is 53357. A sample mesh coarser than what is used in the actual computation is shown in Fig. 9(b). We use the P^2 RKDG method to compute this problem. The flow evolution in the density field is presented in Fig. 10 where the interface and the transmitted shock wave can be observed clearly. Due to the initial perturbation of the interface, there is a misalignment of density and pressure gradient, which forms the vorticity in the

(a) A schematic of flow field at $t=0$



(b) Sample mesh. The mesh points on the boundary are uniformly distributed with cell length 1/10

Figure 9: Flow field and sample mesh.



(a) $t=2.3138$ (b) $t=4.6173$

(c) $t=6.9298$ (d) $t=9.2443$

Figure 10: Dynamics of the interface and density field.

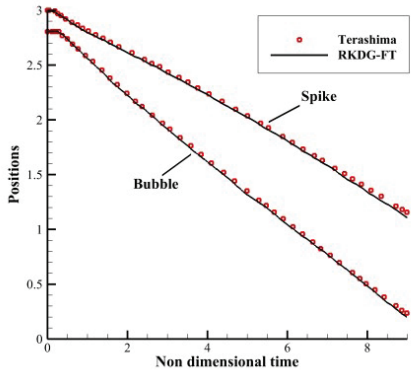


Figure 11: Comparison with other results on time histories of characteristic positions.

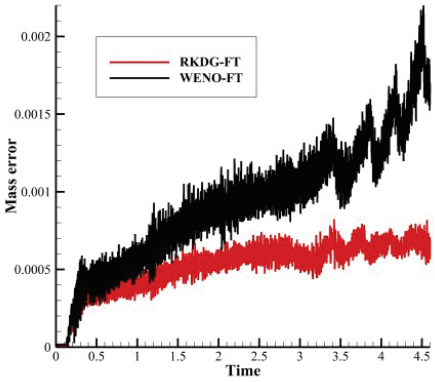
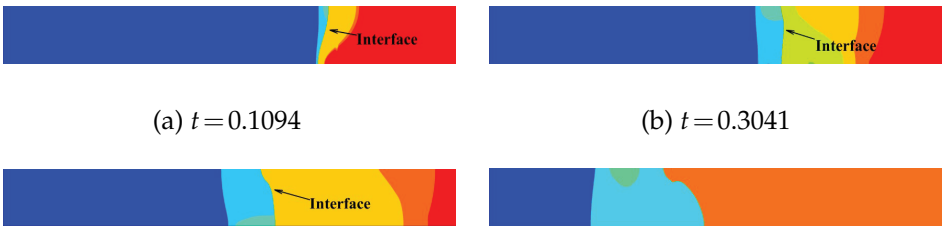


Figure 12: Comparison of mass errors of the SF_6 medium.

perturbed zone, as can be seen in Fig. 10(c) and Fig. 10(d). In order to validate the accuracy of the results, in Fig. 11 we present the time evolution of the location of the spike and the leading edge of the bubble along with the results of Terashima et al. [31]. It shows that these results are almost identical for the location of the characteristic points. Similar with the shock-bubble interaction problem, here we also compare the mass errors of the SF_6 medium in Fig. 12 before the shock wave transmits to the left boundary. It is found that the mass errors are similar for both methods at the initial stage. However, the error by the WENO-FT method increases quickly after the shock wave transmits into the SF_6 medium. The error curve for the RKDG-FT method is smoother and the corresponding mass error is much smaller.

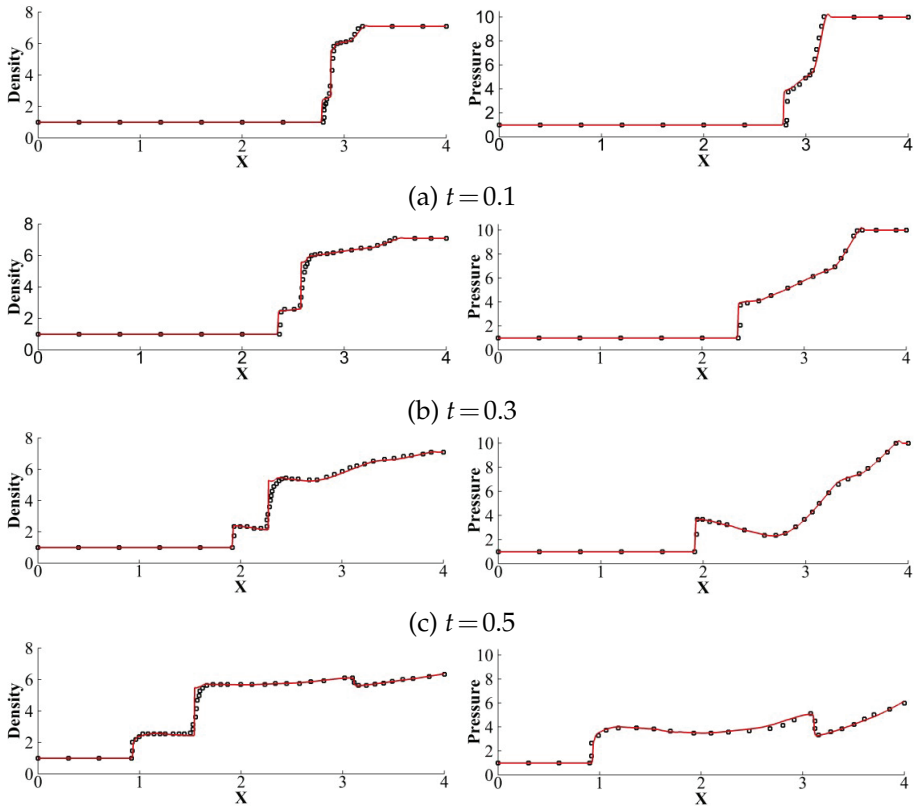
The second one is a gas-liquid interface that is interacting with a planar Mach number 1.95 shock wave at $x=3.025$ initially in liquid. The computational domain, the grids and the initial location of the interface are the same with the first simulation of Richtmyer-Meshkov instability. The initial conditions are: $\rho=1$, $u=0$, $v=0$, $p=1$, $\gamma=1.4$, $B=0$, for air, $\rho=5$, $u=0$, $v=0$, $p=1$, $\gamma=4$, $B=1$, for pre-shocked liquid, $\rho=7.093$, $u=-0.7288$, $v=0$,



(a) $t=0.1094$ (b) $t=0.3041$

(c) $t=0.4995$ (d) $t=0.9907$

Figure 13: The density field at different time.



(a) $t=0.1$

(b) $t=0.3$

(c) $t=0.5$

(d) $t=1.0$

Figure 14: Comparison of density and pressure along line $y=0.5$ at different time.

$p=10$, $\gamma=4$, $B=1$, for post-shocked liquid. We use the P^2 RKDG method to compute this example. The density field is shown in Fig. 13 and the complex wave structure is once again present and is relatively well captured. To check the correctness of the computed solutions, in Fig. 14 we compare the distributions of density and pressure along line $y=$

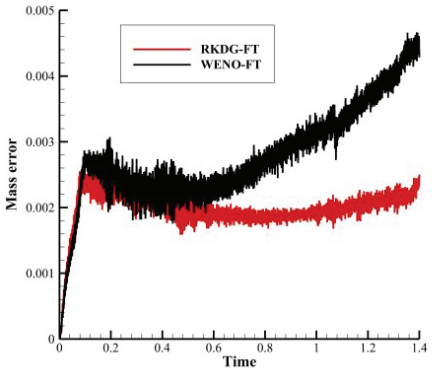


Figure 15: Comparison of mass errors of the air medium.

0.5 with the results ("$\circ$") obtained by using the γ-based model in [30]. Good agreement of the solutions is clearly observed. In Fig. 15 the mass errors of the air medium are calculated by the RKDG-FT method and the WENO-FT method, respectively. It shows that the mass error obtained by the WENO-FT method increases quickly as the shock wave transmits in the air medium. It demonstrates again that the RKDG-FT method has advantages in the mass conservation compared to the WENO-FT method.

5 Concluding remarks

In this paper, the RKDG method combined with the front tracking method is used to simulate the compressible two-medium flow on unstructured meshes. Due to the solution polynomial in the RKDG method, we can directly obtain the state vectors. The normal vectors and ghost fluid states far from the interface which is less accurate need not to be solved due to the good compactness of the RKDG method. Extensive numerical examples are provided to demonstrate the new procedures are compact, adaptive with the irregular domain boundary and the results compared with the earlier literatures are stable and robust subject to many varied initial conditions. The mass errors are calculated to make comparison between the RKDG-FT method and the WENO-FT method. It is found that in general the mass errors caused by the RKDG-FT method are smaller.

Acknowledgements

The research was supported by the National Basic Research Program of China ("973" Program) under grant No. 2014CB046200, NSFC grants 11432007, 11372005, 11271188. Additional support is provided by a project funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions.

References

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

[2] R. CAIDEN, R. P. FEDKIW AND C. ANDERSON, *A numerical method for two-phase flow consisting of separate compressible and incompressible regions,* J. Comput. Phys., 166 (2001), pp. 1–27.

[3] I.-L. CHERN, J. GLIMM, O. MCBRYAN, B. PLOHR AND S. YANIV, *Front tracking for gas dynamics,* J. Comput. Phys., 62 (1986), pp. 83–110.

[4] J.-P. COCCHI AND R. SAUREL, *A Riemann problem based method for the resolution of compressible multimaterial flows,* J. Comput. Phys., 137 (1997), pp. 265–298.

[5] B. COCKBURN, S. HOU AND C.-W. SHU, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case,* Math. Comput., 54 (1990), pp. 545–581.

[6] B. COCKBURN, S.-Y. LIN AND C.-W. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems,* J. Comput. Phys., 84 (1989), pp. 90–113.

[7] B. COCKBURN AND C.-W. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework,* Math. Comput., 52 (1989), pp. 411–435.

[8] B. COCKBURN AND C.-W. SHU, *The Runge-Kutta local projection $P1$-discontinuous Galerkin finite element method for scalar conservation laws,* Math. Model. Numer. Anal., 25 (1991), pp. 337–361.

[9] B. COCKBURN AND C.-W. SHU, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems,* J. Comput. Phys., 141 (1998), pp. 199–224.

[10] R. P. FEDKIW, *Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method,* J. Comput. Phys., 175 (2002), pp. 200–224.

[11] 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), pp. 457–492.

[12] J. GLIMM, J. W. GROVE, X. L. LI, K.-M. SHYUE, Y. ZENG AND Q. ZHANG, *Three-dimensional front tracking,* SIAM J. Sci. Comput., 19 (1998), pp. 703–727.

[13] J. GLIMM, J. W. GROVE, X. L. LI AND N. ZHAO, *Simple front tracking,* Contemp. Math., 238 (1999), pp. 133–149.

[14] J. GLIMM, J. W. GROVE, X. L. LI, W. OH AND D. H. SHARP, *A critical analysis of Rayleigh-Taylor growth rates,* J. Comput. Phys., 169 (2001), pp. 652–677.

[15] Y. HAO AND A. PROSPERETTI, *A numerical method for three-dimensional gas-liquid flow computations,* J. Comput. Phys., 196 (2004), pp. 126–144.

[16] J.-F. HASS AND B. STURTEVANT, *Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities,* J. Fluid Mech., 181 (1987), pp. 41–76.

[17] S. KARNI, *Multicomponent flow calculations by a consistent primitive algorithm,* J. Comput. phys., 112 (1994), pp. 31–43.

[18] B. LARROUTUROU, *How to preserve the mass fractions positivity when computing compressible multi-component flows,* J. Comput. Phys., 95 (1991), pp. 59–84.

[19] T. G. LIU, B. C. KHOO AND K. S. YEO, *The simulation of compressible multi-medium flow I: a new methodology with test applications to 1D gas-gas and gas-water cases,* Comput. Fluids, 30 (2001), pp. 291–314.

[20] H. LU, N. ZHAO AND D. WANG, *A front tracking method for the simulation of compressible*

multimedium flows, Commun. Comput. Phys., 19 (2016), pp. 124–142.
[21] W. MULDER, S. OSHER AND J. A. SETHIAN, *Computing interface motion in compressible gas dynamics,* J. Comput. Phys., 100 (1992), pp. 209–228.
[22] R. R. NOURGALIEV, T. N. DINH AND T. G. THEOFANOUS, *Adaptive characteristics-based matching for compressible multifluid dynamics,* J. Comput. Phys., 213 (2006), pp. 500–529.
[23] S. OSHER AND R. P. FEDKIW, *Level set methods: an overview and some recent results,* J. Comput. Phys., 169 (2001), pp. 463–502.
[24] J. M. PICONE AND J. P. BORIS, *Vorticity generation by shock propagation through bubbles in a gas,* J. Fluid Mech., 189 (1988), pp. 23–51.
[25] J. QIU, T. G. 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), pp. 479–504.
[26] J. J. QUIRK AND S. KARNI, *On the dynamics of a Shock-bubble interaction,* J. Fluid Mech., 318 (1996), pp. 129–163.
[27] W. H. REED AND T. R. HILL, *Triangular mesh methods for the neutron transport equation,* Los Alamos Scientific Laboratory Report LA-UR-73-479, 1973.
[28] C.-W. SHU, *TVB uniformly high-order schemes for conservation laws,* Math. Comput., 49 (1987), pp. 105–121.
[29] C.-W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock capturing schemes,* J. Comput. Phys., 77 (1988), pp. 439–471.
[30] K.-M. SHYUE, *An efficient shock-capturing algorithm for compressible multicomponent problems,* J. Comput. Phys., 142 (1998), pp. 208–242.
[31] H. TERASHIMA AND G. TRYGGVASON, *A front tracking/ghost fluid method for fluid interfaces in compressible flows,* J. Comput. Phys., 228 (2009), pp. 4012–4037.
[32] G. TRYGGVASON, B. BUNNER, A. ESMAEELI, D. JURIC, N. AL-RAWAHI, W. TAUBER, J. HAN, S. NAS AND Y.-J. JAN, *A front tracking method for the computations of multiphase flow,* J. Comput. Phys., 169 (2001), pp. 708–759.
[33] C. W. WANG, T. G. LIU AND B. C. KHOO, *A real ghost fluid method for the simulation of multimedium compressible flow,* SIAM J. Sci. Comput., 28 (2006), pp. 278–302.
[34] J. ZHU AND J. QIU, *Adaptive Runge-Kutta discontinuous Galerkin methods with modified ghost fluid method for simulating the compressible two-medium flow,* J. Math. Study, 47 (2014), pp. 250–273.