COMMUNICATIONS IN COMPUTATIONAL PHYSICS
Vol. **3**, No. 2, pp. 479-504

Commun. Comput. Phys.
February 2008

Simulations of Compressible Two-Medium Flow by Runge-Kutta Discontinuous Galerkin Methods with the Ghost Fluid Method

Jianxian Qiu[1,*], Tiegang Liu[2,3] and Boo Cheong Khoo[4]

[1] *Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, China.*
[2] *Institute of High Performance Computing, ♯01−01 The Capricorn, Singapore Science Park II, Singapore 117528, Singapore.*
[3] *Department of Mathematics, Beijing University of Aeronautics and Astronautics, Beijing, China.*
[4] *Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore./ Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore.*

Received 21 March 2007; Accepted (in revised version) 31 May 2007

Communicated by Chi-Wang Shu

Available online 23 October 2007

Abstract. The original ghost fluid method (GFM) developed in [13] and the modified GFM (MGFM) in [26] have provided a simple and yet flexible way to treat two-medium flow problems. The original GFM and MGFM make the material interface "invisible" during computations and the calculations are carried out as for a single medium such that its extension to multi-dimensions becomes fairly straightforward. The Runge-Kutta discontinuous Galerkin (RKDG) method for solving hyperbolic conservation laws is a high order accurate finite element method employing the useful features from high resolution finite volume schemes, such as the exact or approximate Riemann solvers, TVD Runge-Kutta time discretizations, and limiters. In this paper, we investigate using RKDG finite element methods for two-medium flow simulations in one and two dimensions in which the moving material interfaces is treated via non-conservative methods based on the original GFM and MGFM. Numerical results for both gas-gas and gas-water flows are provided to show the characteristic behaviors of these combinations.

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

Key words: Runge-Kutta discontinuous Galerkin method, WENO scheme, ghost fluid method, approximate Riemann problem solver.

*Corresponding author. *Email addresses:* jxqiu@nju.edu.cn (J. Qiu), liutg@ihpc.a-star.edu.sg (T. Liu), mpekbc@nus.edu.sg (B. C. Khoo)

1 Introduction

In this paper, we investigate and compare between the RKDG finite element methods employed for two-medium flow simulations in which the different non-conservative methods based on the original ghost fluid method (GFM) [13] and the modified GFM (MGFM) [26] are used to treat the moving material interface.

The Runge-Kutta discontinuous Galerkin (RKDG) method [8–12], for solving hyperbolic conservation laws given as

$$\begin{cases} U_t+\nabla\cdot F(U)=0 \\ U(x,0)=U_0(x), \end{cases} \tag{1.1}$$

is a high order finite element method employing the useful features from high resolution finite volume schemes, such as the exact or approximate Riemann solvers, TVD Runge-Kutta time discretizations [35, 36], and total variation bounded (TVB) limiters [34] or weighted essential non-oscillatory (WENO) type limiter [31, 32]. RKDG methods have been widely applied and perform very well to solve for single-medium compressible flow problems.

On the other hand, a relatively dominant difficulty for simulating compressible two-medium flow is the treatment of the moving material interfaces and their immediate vicinities. There can arise severe nonphysical oscillations in the vicinity of the material interface especially in the presence of shock and large density ratio even when a well-established numerical method for single-medium flow is applied directly to the multi-medium flow. As such, there are numerous works published in the literature on how to overcome this difficulty [1, 2, 4, 7, 20, 22, 24].

In the fairly recent times, the ghost fluid method (GFM) as proposed by Fedkiw et al. [13] has provided an alternative and yet flexible way to treat the two-medium flow. The main characteristic of the GFM is its simplicity, ease of extension to multi-dimensions and maintenance of a sharp interface without smearing. Essentially, the GFM makes the interface "invisible" during calculations by defining ghost cells and ghost fluids, and the ensuring computations are then carried out as for a single-medium manner via solving two respective single-medium GFM Riemann problems. As such, its extension to multi-dimensions becomes fairly straightforward. Also, since only single-fluid flux formulations are required for the GFM, the said method can be employed for any two fluids of vastly different EOS (equation of state), compressible-incompressible or viscous-inviscid two-fluid flow [5]. There is subsequently developed variations of the original GFM [14] with other applications as can be found in [2, 21].

However, it is precisely the manner of treatment of the single medium across the interface in the GFM that may cause numerical inaccuracy in the presence of a strong shock wave interacting with the interface [26]; this is especially so if such wave interaction with the interface is not taken into account properly in the definition of ghost fluid states. This happens because the dynamics of shock refraction at the material interface and the resultant interfacial status are highly dependent on the material properties on both sides

of the interface. Therefore, it is imperative that reasonable ghost fluid states have to be specially formulated taking into account the influence of both the material properties and wave interaction with the interface. This has resulted in a modified GFM (MGFM) [26]. The predicted ghost fluid status is obtained via solving implicitly the two non-linear characteristic equations interacting and applicable at the interface [24,25]. In fact, certain conditions called Condition I and Condition II have to be satisfied for the ghost fluid state in order to ensure that the two GFM Riemann problems provides the correct solution in the respective real fluids [23]. Those techniques developed in [24,25] will also be used to calculate the flow interface state in the multi-medium RKDG algorithm as proposed in this work.

It may be mentioned, generally, algorithms proposed for solving two-medium compressible flow comprise of two parts. The first part is the method for solving the single-medium flow, while the other relates to the treatment of the interface of the two fluids. In [5,13,14], Fedkiw and his coworkers employed the 3rd order ENO methods together with the GFM for treating the interface, while the Godunov-type or MUSCL (monotone upwind schemes for conservation laws) are used in [1,2,4,23–26]. Using GFM to treat a moving interface, the final algorithm/method is usually nonconservative. The preliminary analysis on the conservative errors for various existing GFM algorithms may be found in [26,37]. It was concluded that the conservative errors caused by the MGFM-based algorithm [26] is well-limited in comparison to the GFM-based algorithm [13]. Recently, efforts have also been made to develop a conservative GFM [3,15]; this includes the incorporation of RKDG method with a conservative treatment of the material interface via ghost nodes/ghost fluids for two-medium flow in one dimension [30]. As we are aware, however, a practical conservative GFM algorithm has yet to be developed.

As we are aware, there is much less work related to a DG method applied to multi-medium compressible flow. In literature, the basic scheme for solving compressible flow field is usually a finite difference scheme when the GFM is applied to treat the material interface. Although the basic procedure is similar to that of a finite difference scheme, some special cares are required when a DG method is coupled to the GFM applied to multi-medium flow. Comparison to the finite difference method, one major advantage of DG is its compactness. More specifically, the number of ghost cells required for finite difference method is depended on the order of accuracy of the scheme, the number of ghost cells for the DG method is only two cells and independent on the order of DG (for 1D). Another advantage of using DG method is that higher order accuracy is easily gained in the region of smooth solution in comparison to finite difference method. In this paper, our intention is, therefore, to investigate the use of RKDG finite element methods coupled to the GFM technique for two-medium flow simulations. Similar to the above-mentioned algorithms developed for multi-medium compressible flows, the present methods proposed also comprise of two parts. One is the usual RKDG algorithm for solving the flow field away from the material interface; the other is the newly developed techniques for treating the moving interface based on the original GFM [13] and the MGFM [26].

In Section 2, we first briefly review the usual RKDG method over a fixed and regu-

lar mesh system, and then we describe in some details the non-conservative approaches based on both the GFM and MGFM techniques for treating the moving material interface. Extensive numerical results are presented in Section 3 to illustrate and compare the behaviors of the RKDG methods incorporating the different techniques for treating the moving interface based on either the original GFM or MGFM. Concluding remarks are given in Section 4.

2 Implementation of RKDG methods for two-medium flow simulations

In this section we describe in some details the construction and implementation of RKDG methods for two-medium inviscid compressible flow simulations.

We consider both the one dimensional Euler equation (1.1) with

$$U=(\rho,\rho v,E)^T,\quad F(U)=(\rho v,\rho v^2+p,v(E+p))^T,$$

and the two dimensional Euler equation (1.1) with

$$\begin{aligned}&U=(\rho,\rho u,\rho v,E)^T,\quad F(U)=(F_1(U),F_2(U)),\\&F_1(U)=(\rho u,\rho u^2+p,\rho uv,u(E+p))^T,\quad F_2(U)=(\rho v,\rho uv,\rho v^2+p,v(E+p))^T.\end{aligned}$$

Here ρ is the density, (u,v) is the velocity components in the respective x_1 and x_2 directions, $m=\rho u$ and $n=\rho v$ are the moments, E is the total energy, p is the pressure, which is related to the total energy by $E=\rho e+\frac{1}{2}\rho(u^2+v^2)$ (where e is the specific internal energy per mass). For closure of system, the equation of state (EOS) is required. The $\gamma-$law used for gases is given as

$$\rho e=p/(\gamma-1) \tag{2.1}$$

and Tait EOS used for the water medium [6, 13, 24] is expressed as

$$\rho e=(p+N\overline{B})/(N-1), \tag{2.2}$$

where $\overline{B}=B-A$, $N=7.15, A=1.0\text{E}5$ Pa, $B=3.31\text{E}8$ Pa and $\rho_0=1000.0\text{kg}/\text{m}^3$.

2.1 Description of RKDG method

Given a partition consisting of cells $\triangle_j$ (intervals in 1D, triangles or quadrilaterals in 2D, etc.), a semi-discrete discontinuous Galerkin method for solving the conservation law (1.1) is obtained by multiplying (1.1) with a test function $\Phi(x)$, integrating over a cell $\triangle_j$, and integrating by parts:

$$\frac{d}{dt}\int_{\triangle_j}U(x,t)\Phi(x)dx-\int_{\triangle_j}F(U)\cdot\nabla\Phi dx+\int_{\partial\triangle_j}F(U)\cdot n\Phi ds=0 \tag{2.3}$$

where n is the outward unit normal of the cell boundary $\partial\triangle_j$. We seek a piecewise polynomial U in $\mathbb{P}^k$ (of degree at most k, k could actually change from cell to cell, but for simplicity we assume it is a constant over the whole cells), such that (2.3) holds for any test function Φ also in $\mathbb{P}^k$. The boundary integral in (2.3) is typically discretized by a Gaussian quadrature of sufficiently high order of accuracy

$$\int_{\partial\triangle_j} F\cdot n\Phi ds \approx |\partial\triangle_j| \sum_{k=1}^{q} \omega_k F(U(G_k,t))\cdot n\Phi(G_k)$$

and $F(U(G_k,t))\cdot n$ is replaced by a monotone numerical flux (approximate or exact Riemann solvers in the system case). For example, one could use the simple Lax-Friedrichs flux, which is given by

$$F(U(G_k,t))\cdot n \approx \frac{1}{2}\left[\left(F(U^-(G_k,t))+F(U^+(G_k,t))\right)\cdot n-\alpha\left(U^+(G_k,t)-U^-(G_k,t)\right)\right],$$

where α is taken as an upper bound for $|F'(U)\cdot n|$ in the scalar case, or the absolute value of eigenvalues of the Jacobian in the n direction for the system case, and U^- and U^+ are the values of U inside the cell $\triangle_j$ and outside the cell $\triangle_j$ (inside the neighboring cell) at the Gaussian point G_k. The idea of using such a numerical flux is borrowed from finite volume methodology. The test function Φ in the boundary integral in (2.3) is taken from inside the cell $\triangle_j$. The volume integral $\int_{\triangle_j} F(U)\cdot\nabla\Phi dx$ in (2.3) is either computed exactly or by a numerical quadrature with sufficient accuracy, see [8,10] for details. The semidiscrete scheme (2.3), written as

$$U_t = L(U),$$

is then discretized in time by a total variation diminishing (TVD) Runge-Kutta method [35], for example the third order version given by

$$\begin{aligned} U^{(1)} &= U^n+\Delta t L(U^n),\\ U^{(2)} &= \frac{3}{4}U^n+\frac{1}{4}U^{(1)}+\frac{1}{4}\Delta t L(U^{(1)}),\\ U^{n+1} &= \frac{1}{3}U^n+\frac{2}{3}U^{(2)}+\frac{2}{3}\Delta t L(U^{(2)}). \end{aligned} \tag{2.4}$$

In the following, we describe RKDG method in detail for one dimensional case. The computational domain is divided into N cells with boundary points $a=x_{\frac{1}{2}}<x_{\frac{3}{2}}<\cdots<x_{N+\frac{1}{2}}=b$. We denote the cells by $I_i=[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}]$, the cell centers by $x_i=\frac{1}{2}(x_{i-\frac{1}{2}}+x_{i+\frac{1}{2}})$ and the cell sizes by $\Delta x_i=x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}$, $h=\sup_i\Delta x_i$. The solution as well as the test function space is given by $V_h^k=\{p:p|_{I_i}\in P^k(I_i)\}$, where $P^k(I_i)$ is the space of polynomials of degree $\leq k$ on the cell I_i. We adopt a local orthogonal basis over I_i, $\{\Phi_l^{(i)}(x), l=0,1,\cdots,k\}$, namely the scaled Legendre polynomials:

$$\Phi_0^{(i)}(x)=1,\quad \Phi_1^{(i)}(x)=\frac{x-x_i}{\Delta x_i/2},\quad \Phi_2^{(i)}(x)=\left(\frac{x-x_i}{\Delta x_i/2}\right)^2-\frac{1}{3},\quad \cdots$$

The numerical solution $U^h(x,t)$ of Eq. (1.1) in the test function space V_h^k can then be written as

$$U^h(x,t)=\sum_{l=0}^{k}U_i^{(l)}(t)\Phi_l^{(i)}(x),\qquad \text{for } x\in I_i \tag{2.5}$$

and the degrees of freedom $U_i^{(l)}(t)$ are the moments defined by

$$U_i^{(l)}(t)=\frac{1}{a_l}\int_{I_i}U^h(x,t)\Phi_l^{(i)}(x)dx,\qquad l=0,1,\cdots,k$$

where $a_l=\int_{I_i}(\Phi_l^{(i)}(x))^2dx$ are the normalization constants since the basis is not orthonormal. In order to determine the approximate solution, we need to evolve the moments $U_i^{(l)}$. By substituting (2.5) into Eq. (1.1), we can obtain the governing equations for the moments as

$$\begin{aligned}&\frac{d}{dt}U_i^{(l)}+\frac{1}{a_l}\bigg(-\int_{I_i}F(U^h(x,t))\frac{d}{dx}\Phi_l^{(i)}(x)dx+F(U^h(x_{i+1/2},t))\Phi_l^{(i)}(x_{i+1/2})\\&-F(U^h(x_{i-1/2},t))\Phi_l^{(i)}(x_{i-1/2})\bigg)=0,\qquad l=0,1,\cdots,k.\end{aligned} \tag{2.6}$$

Eq. (2.6) has to be solved approximately. In order to enforce the entropy condition, the flux $F(U^h(x_{i+1/2},t))$ is usually approximated using a monotone numerical flux $\hat{F}(U_{i+1/2}^-,U_{i+1/2}^+)$, resulting in a semi-discretization scheme as

$$\begin{aligned}&\frac{d}{dt}U_i^{(l)}+\frac{1}{a_l}\bigg(-\int_{I_i}F(U^h(x,t))\frac{d}{dx}\Phi_l^{(i)}(x)dx+\hat{F}(U_{i+1/2}^-,U_{i+1/2}^+)\Phi_l^{(i)}(x_{i+1/2})\\&-\hat{F}(U_{i-1/2}^-,U_{i-1/2}^+)\Phi_l^{(i)}(x_{i-1/2})\bigg)=0,\qquad l=0,1,\cdots,k,\end{aligned} \tag{2.7}$$

where $U_{i+1/2}^{\pm}=U^h(x_{i+1/2}^{\pm},t)$ are the left and right limits of the discontinuous solution U^h at the cell interface $x_{i+1/2}$, $\hat{F}(U^-,U^+)$ is a monotone flux (non-decreasing in the first argument and non-increasing in the second argument) for the scalar case and an exact or approximate Riemann solver for the system case. In this work, the simple Lax-Friedrichs flux is used, which is given as

$$\hat{F}(U^-,U^+)=\frac{1}{2}\left[\left(F(U^-)+F(U^+)\right)-\alpha\left(U^+-U^-\right)\right],$$

where α is taken as an upper bound for the absolute value of eigenvalues of the Jacobian for the system case. The integral term in (2.7) can be computed either exactly or by a suitable numerical quadrature accurate to at least $\mathcal{O}(h^{k+l+2})$. In this paper we use three and four point Gauss-Lobatto quadrature for P^1 and P^2 cases, respectively.

The semi-discrete scheme (2.7) can be generalized as

$$U_t=L(U),$$

which is discretized in time by a Runge-Kutta time discretization, e.g. the third order version (2.4).

If there are strong discontinuities in solution, the scheme (2.3)-(2.4) generates significant oscillations and even nonlinear instability. To avoid such difficulties, typically a slope limiter is used after each Runge-Kutta inner stage to control the numerical oscillations. The limiter adopted in [10] is described below in some detail. We denote

$$U^-_{i+1/2}=U^{(0)}_i+\tilde{U}_i,\quad U^+_{i-1/2}=U^{(0)}_i-\tilde{\tilde{U}}_i.$$

Here

$$\tilde{U}_i=\sum_{l=1}^{k}U^{(l)}_i\Phi^{(i)}_l(x_{i+1/2}),\quad \tilde{\tilde{U}}_i=-\sum_{l=1}^{k}U^{(l)}_i\Phi^{(i)}_l(x_{i-1/2}).$$

$\tilde{U}_i$ and $\tilde{\tilde{U}}_i$ are modified by either the standard minmod limiter [18]

$$\tilde{U}^{(mod)}_i=m(\tilde{U}_i,\Delta_+U^{(0)}_i,\Delta_-U^{(0)}_i),\quad \tilde{\tilde{U}}^{(mod)}_i=m(\tilde{\tilde{U}}_i,\Delta_+U^{(0)}_i,\Delta_-U^{(0)}_i),$$

where m is given by

$$m(a_1,a_2,\cdots,a_n)=\begin{cases} s\cdot\min_{1\le j\le n}|a_j|, & \text{if } sign(a_1)=sign(a_2)=\cdots=sign(a_n)=s,\\ 0, & \text{otherwise,}\end{cases}\tag{2.8}$$

or the TVB modified minmod function [34]

$$\tilde{m}(a_1,a_2,\cdots,a_n)=\begin{cases} a_1, & \text{if } |a_1|\le Mh^2,\\ m(a_1,a_2,\cdots,a_n), & \text{otherwise,}\end{cases}\tag{2.9}$$

where $M>0$ is a constant dependent on solution of the problem.

For two dimensional case, the limiter adopted in [8,12] is performed on the two first moment using the differences of means.

For the system cases, in order to achieve better qualities at the price of more complicated computations, the limiter is always used with a local characteristic field decomposition; see, e.g., [9,12] for details.

2.2 Description of treatment of interface cell with ghost fluid methods

In this section, we describe in detail the interfacial treatment via the technique of the ghost fluid method coupled to the DG method.

2.2.1 Outline of a GFM-based algorithm

In a GFM-based algorithm, the Level Set technique is usually employed to capture the moving interface. A band of 2 to 5 grid points as ghost cells is defined in the vicinity of the interface. At the ghost cells, ghost fluid and real fluid co-exist. Once the ghost fluid nodes and ghost fluid status are defined for each medium, one employs one's favorite

pure medium numerical scheme/solver (in this work, the DG method) to solve for each medium covering both the real fluid and ghost fluid grid nodes/cells. By combining the solution for each medium according to the new interface location, one obtains the overall solution valid for the whole computational domain at the new time step.

2.2.2 Outline of a GFM-based DG algorithm

Although the basic procedure is similar to that of a finite difference scheme, there are some differences and special cares required when the DG method is coupled to the GFM applied to simulate multi-medium flow. Unlike the finite difference method, where only a nodal value for the ghost fluid is required in each ghost cell, a solution function has to be defined in each ghost fluid cell in order that the DG method can be applied across the interface. In addition, unlike the finite difference method, where the number of ghost cells required is dependant on the order of accuracy of the scheme, the number of ghost cells for the DG method is only two cells (for 1D). Assuming that the interface is located in cell I_j at time $t=t^n$ (see Fig. 1 for illustration) and the solution at $t=t^n$ is known, we want to compute for the flow field at the next time step of $t=t^{n+1}$. Below are the steps for the 1D GFM-based DG algorithm:

Step 1: Calculate the time stepsize, which satisfies the stability condition over the whole computational domain.

Step 2: Obtain the new location of the interface via solving the governing equation of the Level Set function Φ with re-initialization using the fifth order WENO scheme, if necessary. (Readers can refer to [13, 24, 28, 29] for the detail of level set methods and we shall not repeat it here.)

Step 3: Compute the flow field for Fluid I. To do so, cells I_{j+1} and I_{j+2} are defined as the ghost fluid cells for Fluid I (see Fig. 1), and then the ghost fluid solution functions are assigned to these ghost fluid cells. The way of defining these functions will be specified according to the specified GFM implementation to be given in the sections below. Once this is done, one applies the DG method for solving for Fluid I from cell 1 to cell I_{j+1}.

Step 4: Compute the flow field for Fluid II. Similarly, cells I_{j-1} and I_{j-2} are defined as the ghost fluid cells for Fluid II (see Fig. 2), and then the ghost fluid solution functions are assigned to these ghost fluid cells. Finally, one applies the DG method to get the solution for Fluid II from cell I_{j-1} to the last cell.

Step 5: Obtain the final solution according to the new location of the interface. For cells fully immersed in Fluid I (Fluid II) at $t=t^{n+1}$, the solution of Fluid I (Fluid II) is taken as the final solution in those cells at the new time step. For the cell where interface is located, either solution of Fluid I or Fluid II can be taken as the final solution at $t=t^{n+1}$ for this cell. For purpose of plotting, if the interfacial cell is mainly occupied by Fluid I, the solution of Fluid I is taken as the final solution, otherwise, the solution of Fluid II is taken as the final solution at $t=t^{n+1}$ for this cell.

Step 6: Repeat Step 2 to Step 5 for each sub-step of the Runge-Kutta temporal discretization given in 2.4.

Step 7: Check if the stop criterion is satisfied. If it is not, repeat Step 1 to Step 6.

Obviously, by using the GFM technique in the normal direction of the interface, the GFM-based DG algorithm can be easily implemented for multi-dimensions. Here we briefly state its extension to two-dimensions in general. First, we define a rectangular computational domain $[I_{1K},I_{2K}]\times[J_{1K},J_{2K}]$ and an identification matrix S_K for each medium (the K^{th} medium), where $S_K(i,j)=1$ if grid point (not grid cell) (i,j) is taken by the K^{th} medium, otherwise it is set to 0. We carry out the DG computation for each medium respectively, and denote the result by U_K^{n+1} after defining the ghost fluid states for each ghost fluid cell. Then the final solution at the new time step is given by $U^{n+1}=\Sigma S_K^{n+1}U_K^{n+1}$, where S_K^{n+1} is the new identification matrix of the K^{th} medium, which is obtained in advance by the level set technique. One can define a computational domain for each medium that includes boundary points and grid points in the interfacial regions associated with this medium within a band of 2 to 4 grid points defined by $\Phi<\varepsilon$. Here Φ is the level set distance function and ε is set to be about $2\max[\Delta x_1,\Delta x_2]$; Δx_1 and Δx_2 are spatial step sizes in the respective x_1 and x_2 directions.

Now we briefly summarize the programming procedures for the 2D GFM-based DG algorithm. We assume the flow variables at $t=t^n$ are known, and the following steps are taken to obtain quantities at the next time step.

Step 1: Calculate the time stepsize, which satisfies the stability condition over the whole computational domain.

Step 2: Advance the level set function Φ to the next time step and update the identification matrix S_K for each medium over the total domain.

Step 3. Define the computational domain for each medium.

Step 4. Map the previous (sub-step) flow field in the K^{th} medium computational domain to the working space; define ghost fluid states for each ghost fluid cell according to the specific GFM implementation to be given in the sections below.

Step 5: Use the DG solver to obtain U_K^{n+1} and update $U^{n+1}=U^{n+1}+S_K^{n+1}U_K^{n+1}$. ($U^{n+1}$ is initialized to zero at the beginning.)

Step 6: Repeat of Steps 4 and 5 from the first medium to the last medium, and then proceed to the next time sub-step of the RKDG.

Step 7: Check if the stop criterion is satisfied. If it is not, repeat Step 1 to Step 6.

Next, we shall describe the approach based on the original GFM technique [13] and the MGFM [26] for treating the interface cell I_j and its vicinity. In order to advance the solution for Fluid I, we have to construct the solution function (i.e. the moments) in cells I_{j+1} and I_{j+2} for Fluid I and the solution function for Fluid II in cells I_{j-2} and I_{j-1} as well.

2.2.3 The technique based on the original GFM [13]

In the original GFM, to define the ghost fluid state for Fluid I, the pressure and velocity (for Fluid II) in cells I_{j+1} and I_{j+2} are directly taken as the ghost fluid pressure and velocity. To define the ghost fluid density, the entropy of Fluid I at the center of cell I_{j-1} is used. In other words, the entropy of the ghost fluid is assumed to be equal to that of Fluid I at the center of cell I_{j-1}. With the entropy defined, the ghost fluid density can be obtained. We then project the flow states to the base function space in cells I_{j+1} and I_{j+2},

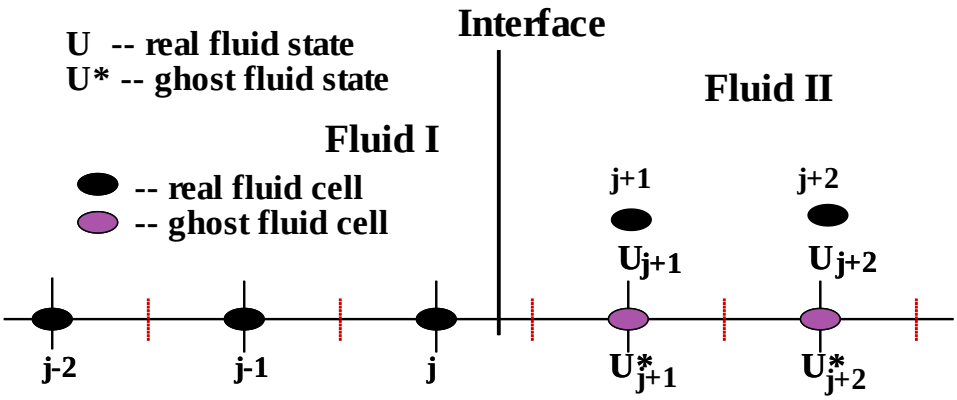

Figure 1: Define ghost fluid for Fluid I.

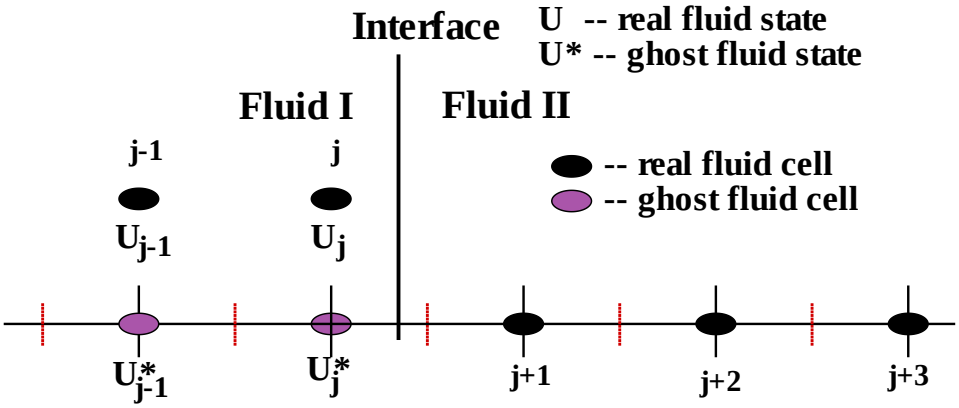

Figure 2: Define ghost fluid for Fluid II.

respectively, to obtain the zero-order moments. Because the derivatives of physical solution are discontinuous across the interface in general, the higher-order moments are set to be zero. Similarly, the ghost fluid states for Fluid II can be defined. With the definitions of ghost fluid states, we carry out computation for each fluid across the interface using the DG method. The final solution can then be obtained according to the new interface location as described in Section 2.2.2.

To apply the original GFM-based DG method to multi-dimensional two medium flow, we define the ghost fluid states for each fluid by following exactly the same way as given in [13] to define the normal velocity, tangential velocity, pressure and entropy in the ghost cells. We then project the flow states to the base function space and set the higher-order moments to be zero to construct the solution function in the ghost fluid cells. Once this is done, the DG is used to advance the solution for each fluid to the next time step. By combining the solution for each fluid together according to the new location of the interface, the final solution is determined. This algorithm is named RKDG-GFM for convenience of discussion in this work.

2.2.4 The technique based on the MGFM [26]

In the MGFM-based algorithm, to define the ghost fluid states, a multi-medium Riemann problem is defined and exactly or approximately solved to predict the interface status. The predicted interface status is then employed to define the ghost fluid states for each fluid. For one dimensional problems, the fluid states in cell I_{j-1} and cell I_{j+1} are used to define the multi-medium Riemann problem. The defined Riemann problem is then approximately solved using the approximate Riemann problem solver developed in [24, 26]. The obtained interface state for Fluid I is applied to replace the flow state in cell I_j and also to define the ghost fluid states in cell I_{j+1} and cell I_{j+2} for Fluid I. We then project the flow states to the base function space in cells I_j, I_{j+1} and I_{j+2}, respectively, to obtain the zero-order moments and set the higher-order moments to be zero at the same time. Similarly, the ghost fluid states for Fluid II can be defined using the predicted interface state for Fluid II. Once the ghost fluid states are defined, the remaining steps are the same as the original GFM-based DG algorithm.

To extend the MGFM-based DG method to multi-dimensional two medium flow, we first define a multi-medium Riemann problem in the normal direction of the interface by following exactly the same way as described in [37]. The defined Riemann problem is also solved using the approximate Riemann problem solver developed in [24, 26]. The predicted interface normal velocity, pressure and density are then used to replace the flow normal velocity, pressure and density at the cell of interface located and propagated to the ghost fluid cells using the technique developed in [13]. The tangential velocity in the ghost cells can be defined by following the same way as given in [13]. The rest of steps is the same as the original GFM-based algorithm. The MGFM-based DG algorithm is named RKDG-MGFM for convenience of discussion in this work.

3 Numerical results

In this section we present the results of numerical experiments for various two-medium flows by the RKDG-GFM and RKDG-MGFM schemes described in Section 2. From the results in one dimensional cases listed below, we can see that the RKDG-MGFM scheme is more effective than RKDG-GFM. It should be noted that the RKDG-GFM suffers similar difficulties as pointed out in [23, 26] when applied to a strong shock or jet impacting on the interface. One may refer to [23, 26] for the detailed discussion and analysis; we will not repeat the analysis here. The results by the RKDG-GFM schemes in two dimensional cases will also not be shown to save space. As expected, the DG-based MGFM algorithm works even better than a finite difference based MGFM algorithm (the MUSCL-based MGFM) [23, 26] in the region of smooth solution because a DG is usually easy to gain local high accuracy in such regions, the comparison between results for MUSCL-MGFM scheme and the second order RKDG-MGFM are shown in Example 3.1 and 3.5.

As it is well known that the TVB limiter constant M is dependent on the problem. We have experimented with several constants and have settled on the following procedure:

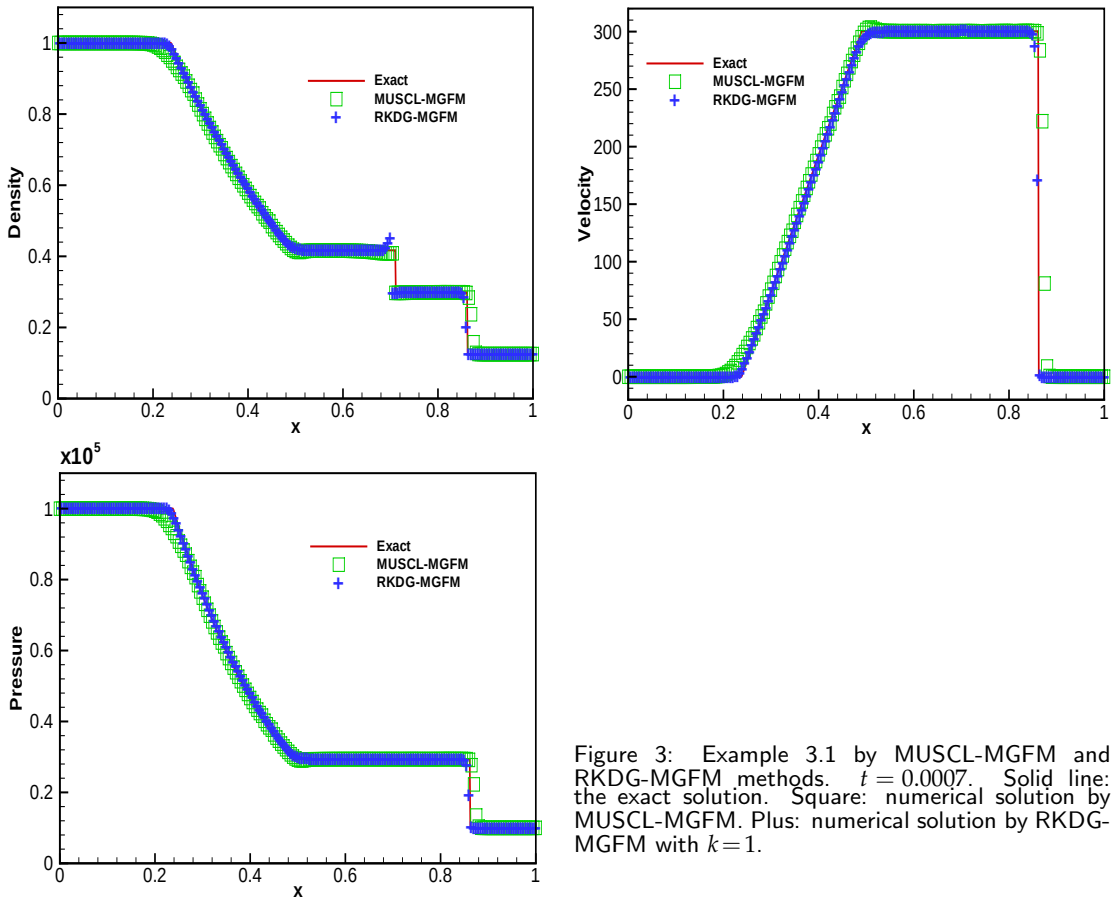

Figure 3: Example 3.1 by MUSCL-MGFM and RKDG-MGFM methods. $t = 0.0007$. Solid line: the exact solution. Square: numerical solution by MUSCL-MGFM. Plus: numerical solution by RKDG-MGFM with $k=1$.

instead of using a constant M, we use two constants as denoted by M_1 for Fluid I limiting procedure and the other M_2 for Fluid II. In the present numerical experiments, 200 cells are used and the CFL numbers are taken as 0.3 and 0.18 for $k=1$ and $k=2$, respectively. Units for density, velocity, pressure, length and time are kg/m^3, m/s, Pa, m and s, respectively.

Example 3.1. This is an air-helium shock tube problem taken from [13], with the initial conditions as

$$(\rho,v,p,\gamma)=(1,0,1\times10^5,1.4) \text{ for } x\leq 0.5; \quad (\rho,v,p,\gamma)=(0.125,0,1\times10^4,1.2) \text{ for } x>0.5.$$

The computed density ρ, velocity v and pressure p by the MUSCL-MGFM scheme and the second order RKDG-MGFM scheme with the TVB limiter constant $M_1=1$ and $M_2=10$ are plotted at $t=0.0007$ against the exact solution in Fig. 3.

The location of the material interface is captured correctly by both the MUSCL-MGFM and RKDG-MGFM schemes. The resolution of computed results are reasonable, and the

overall results by both schemes are comparable to analysis. The RKDG-MGFM works better than MUSCL-MGFM in the region of smooth solution and shock. The results obtained via the RKDG-MGFM scheme indicate perceptibly better concurrence with the analytical solution.

Example 3.2. This is a problem of a shock wave refracting at an air-helium interface with a reflected rarefaction wave. This example is also taken from [13]. The flow initial conditions are

$$(\rho,v,p,\gamma)=\begin{cases}(4.3333,3.2817\sqrt{10^5},1.5\times10^6,1.4), & \text{for } x\leq 0.05,\\ (1,0,1\times10^5,1.4), & \text{for } 0.05<x\leq 0.5,\\ (0.1379,0,1\times10^5,\frac{5}{3}), & \text{for } x>0.5.\end{cases}$$

The strength of the shock is $p_L/p_R=15$ and initially located at $x=0.05$, with the initial state ahead of the shock given as $(\rho,v,p)=(1,0,1\times10^5)$. The initial interface of air and helium is located at $x^*(0)=0.5$. The computed density ρ, velocity v and pressure p by both RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$ are plotted at $t=0.0005$ against the exact solution in Fig. 4. In this case, we choose the TVB limiter constant $M_1=1$ and $M_2=30$ for the case of $k=1$, and $M_1=1$ and $M_2=300$ for the case of $k=2$.

It is clear that the location of the material interface is captured correctly by both RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$. The computed results of density by the RKDG-GFM scheme with both $k=1$ and $k=2$ and RKDG-MGFM scheme with $k=1$ depict a little oscillatory feature in the neighborhood of the interface, whereas that from the RKDG-MGFM scheme with $k=2$ is essentially oscillatory free. Other than that, the overall computed results are in reasonably agreement with the analysis, except for a little hump detected in the fluid on the left, which is also found in [13]. Closer examination of the computed results reveals that the RKDG-MGFM scheme exhibits better agreement with the analysis.

Example 3.3. We consider the Euler equation (1.1) with the following Riemann initial conditions

$$(\rho,v,p,\gamma)=\begin{cases}(1.3333,0.3535\sqrt{10^5},1.5\times10^5,1.4), & \text{for } x\leq 0.05,\\ (1,0,1\times10^5,1.4), & \text{for } 0.05<x\leq 0.5,\\ (3.1538,0,1\times10^5,1.249), & \text{for } x>0.5.\end{cases}$$

The computed density ρ, velocity v and pressure p for both RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$ are plotted at $t=0.0017$ against the exact solution in Fig. 5. In this example, we choose the TVB limiter constant $M_1=1$ and $M_2=300$ for the case of $k=1$, and $M_1=1$ and $M_2=30000$ for the case of $k=2$.

We can see that the interface of the two-medium is located at the correct cell by both RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$. The computed results by both schemes are comparable to the analysis. The computed results for density are essentially oscillation free in the neighborhood of the interface for both $k=1$ and $k=2$

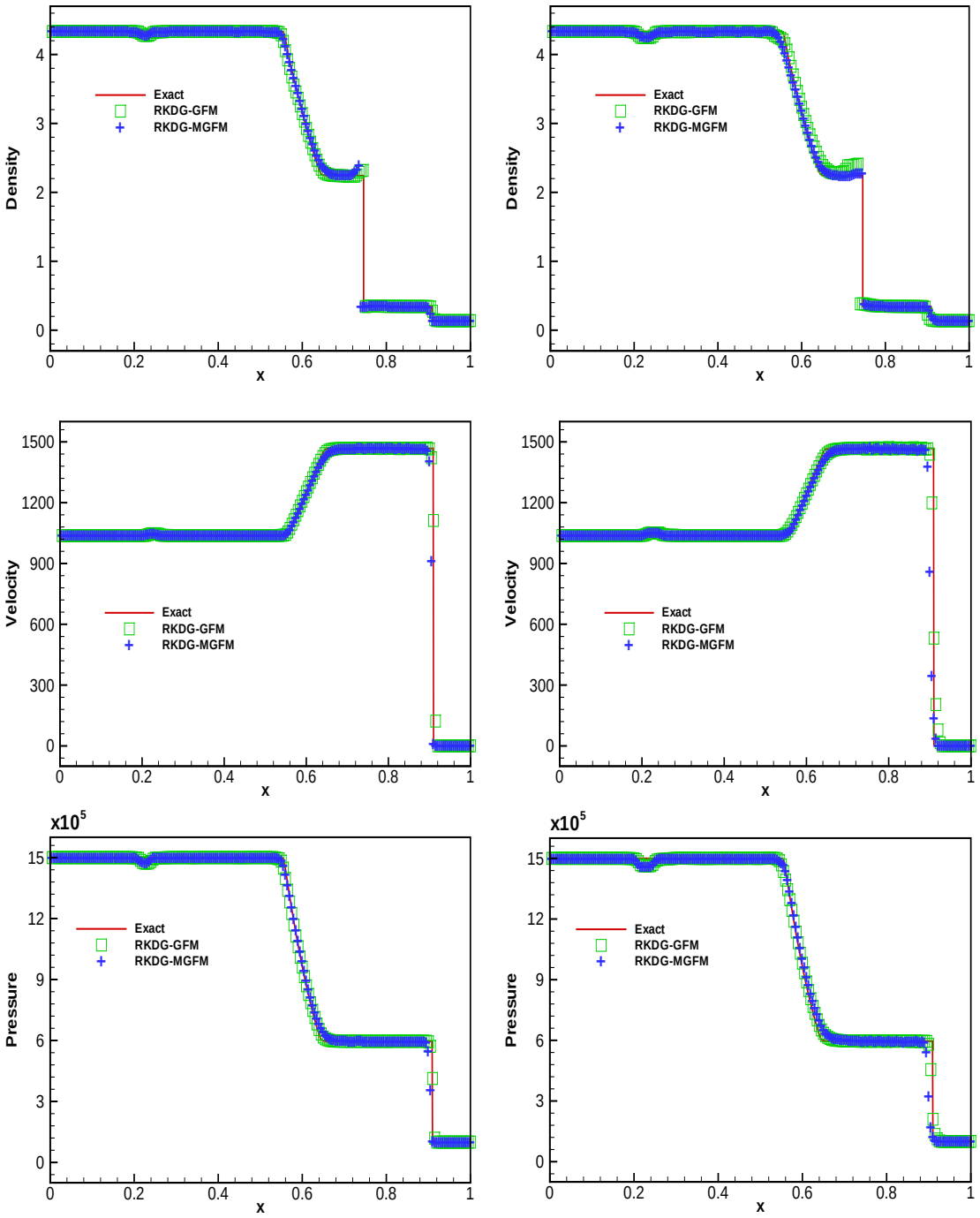

Figure 4: Example 3.2 by RKDG-GFM and RKDG-MGFM methods. $t=0.0005$. Solid line: the exact solution. Square: numerical solution by RKDG-GFM. Plus: numerical solution by RKDG-MGFM. Top: Density; Middle: Velocity; Bottom: Pressure. Left: $k=1$; Right: $k=2$.

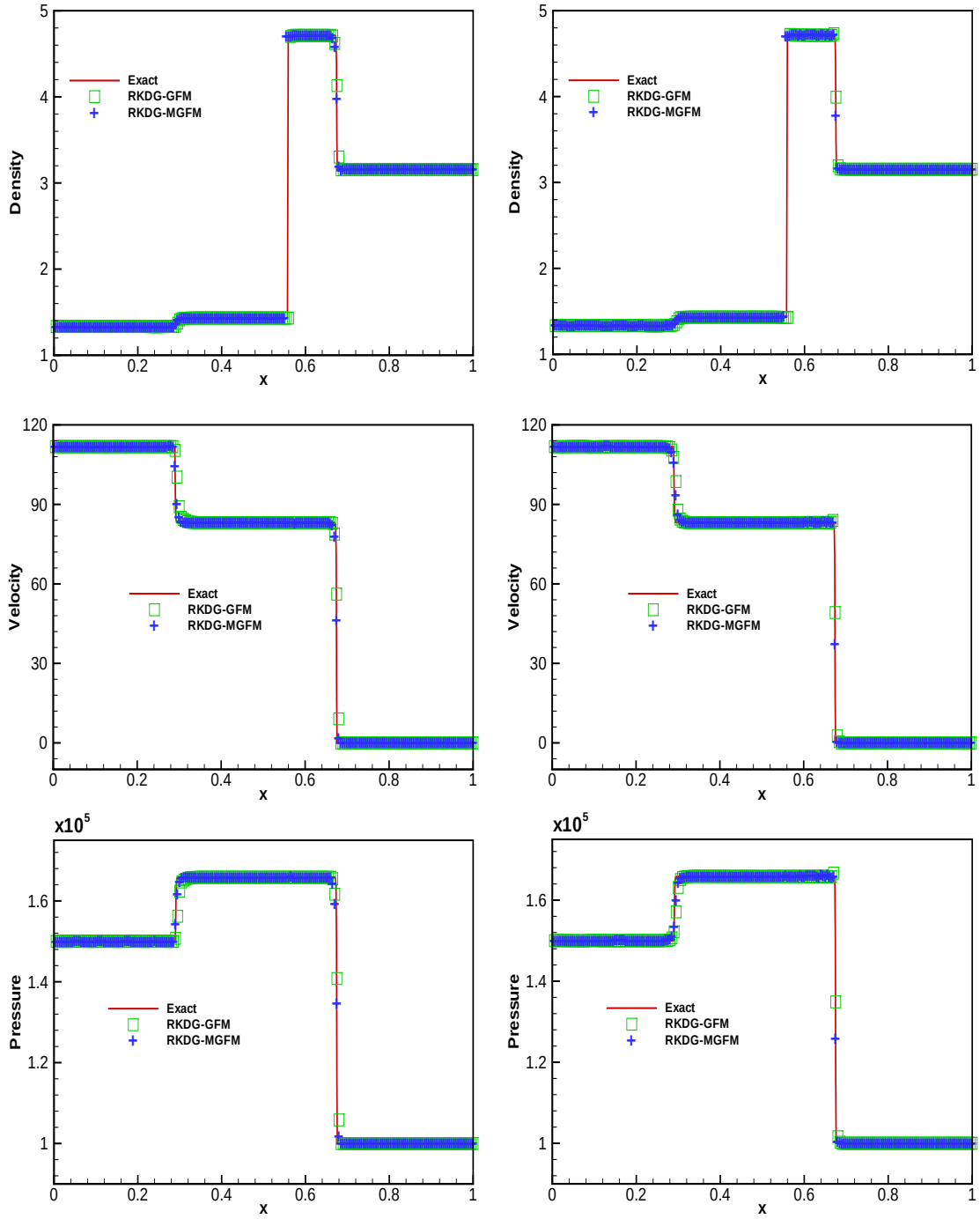

Figure 5: Example 3.3 by RKDG-GFM and RKDG-MGFM methods. $t=0.0017$. Solid line: the exact solution. Square: numerical solution by RKDG-GFM. Plus: numerical solution by RKDG-MGFM. Top: Density; Middle: Velocity; Bottom: Pressure. Left: $k=1$; Right: $k=2$.

cases. The computed results by the RKDG-GFM scheme are comparable to those by the RKDG-MGFM scheme.

Example 3.4. Next we consider a strong shock impacting on a gas-gas interface with the strength of right shock wave to $p_L/p_R=100$, with the following initial conditions

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

The computed density ρ, velocity v and pressure p by both RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$ are plotted at $t=0.0001$ against the exact solution in Fig. 6. In this example, we choose the TVB limiter constant $M_1=1.0$ and $M_2=0.1$ for the case of $k=1$ and $M_1=0.1$ and $M_2=0.001$ for the case of $k=2$.

This example is the Case 1 in [26], the intent of Liu et al. [26] is to show that the MUSCL with the original GFM does not work properly for the problem with a strong shock impacting on a gas-gas interface. From our calculations, it is evidently clear that the shock positions in both Fluid I and Fluid II are found located at the correct cell by the RKDG-MGFM scheme with $k=1$ and $k=2$. Similar to the results in [26], we found that the shock locations in both Fluid I and Fluid II by the RKDG-GFM scheme with $k=1$ and $k=2$ are located in the wrong cell, at least 2 cells away from the correct one. For this example, the RKDG-MGFM scheme work much better than the RKDG-GFM scheme.

Example 3.5. This is a gas-water shock tube problem with very high pressure in the gas medium. The initial condition are

$$(\rho,v,p,\gamma)=\begin{cases}(1270,0,8\times10^8,1.4), & \text{for } x\leq 0.5,\\ (1000,0,1\times10^5,7.15), & \text{for } x>0.5.\end{cases}$$

In this underwater explosion problem, the initial pressure in the gas is extremely high and thus, a very strong shock is generated in the water. The computed density ρ, velocity v and pressure p by the MUSCL-MGFM scheme and the second order RKDG-MGFM scheme with the TVB limiter constant $M_1=1$ and $M_2=1$ are plotted at $t=0.00016$ against the exact solution in Fig. 7.

The location of the material interface is captured correctly by both the MUSCL-MGFM and RKDG-MGFM schemes. The resolution of computed results are reasonable, and the overall results by both schemes are comparable to analysis. The RKDG-MGFM works better than MUSCL-MGFM in the region of smooth solution and shock. The results obtained via the RKDG-MGFM scheme indicate perceptibly better concurrence with the analytical solution.

Example 3.6. We greatly increase the energy of the explosive gaseous medium in Example 3.5. The initial condition becomes

$$(\rho,v,p,\gamma)=\begin{cases}(1630,0,7.81\times10^9,1.4), & \text{for } x\leq 0.5,\\ (1000,0,1\times10^5,7.15), & \text{for } x>0.5.\end{cases}$$

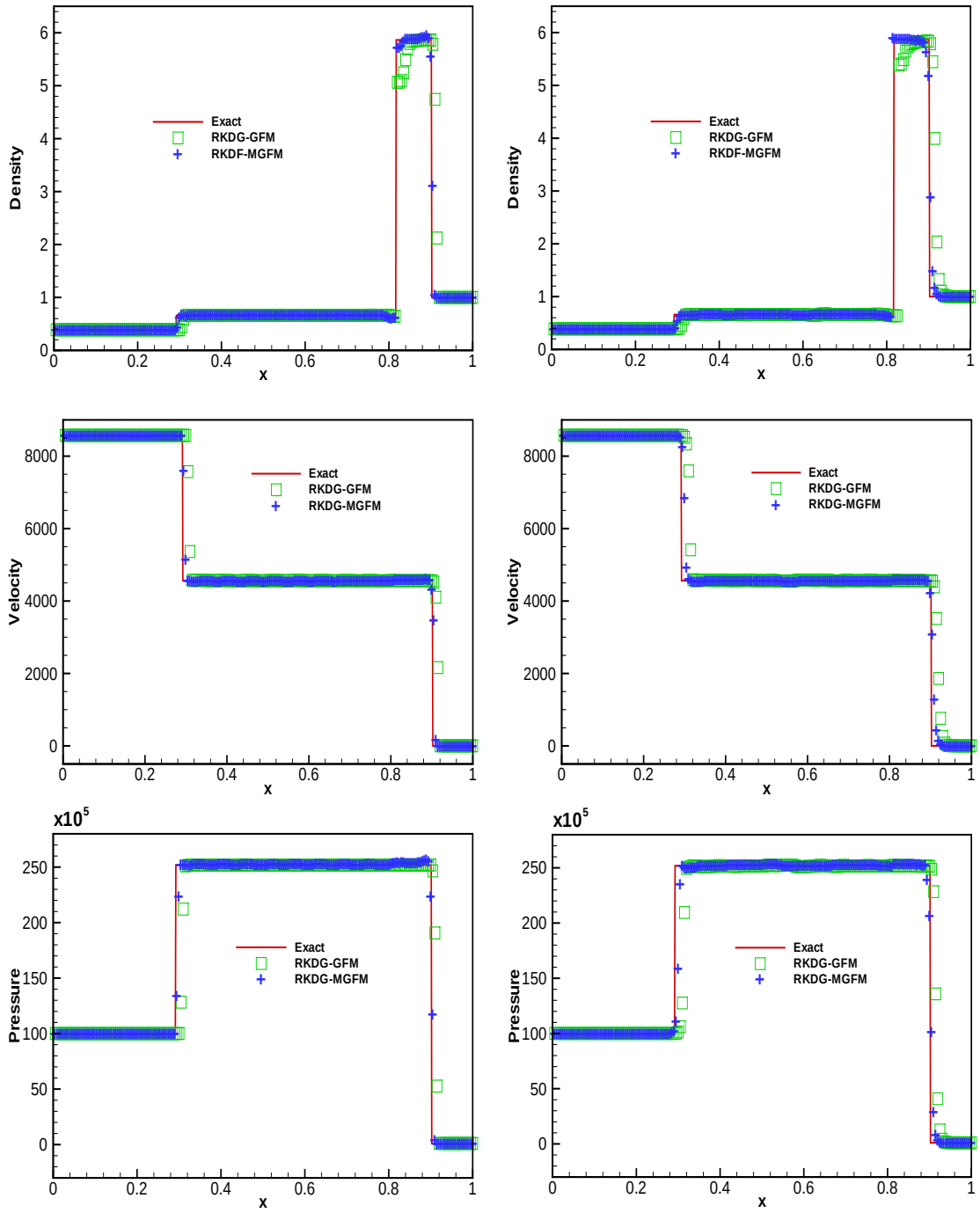

Figure 6: Example 3.4 by RKDG-GFM and RKDG-MGFM methods. $t=0.0001$. Solid line: the exact solution. Square: numerical solution by RKDG-GFM. Plus: numerical solution by RKDG-MGFM. Top: Density; Middle: Velocity; Bottom: Pressure. Left: $k=1$; Right: $k=2$.

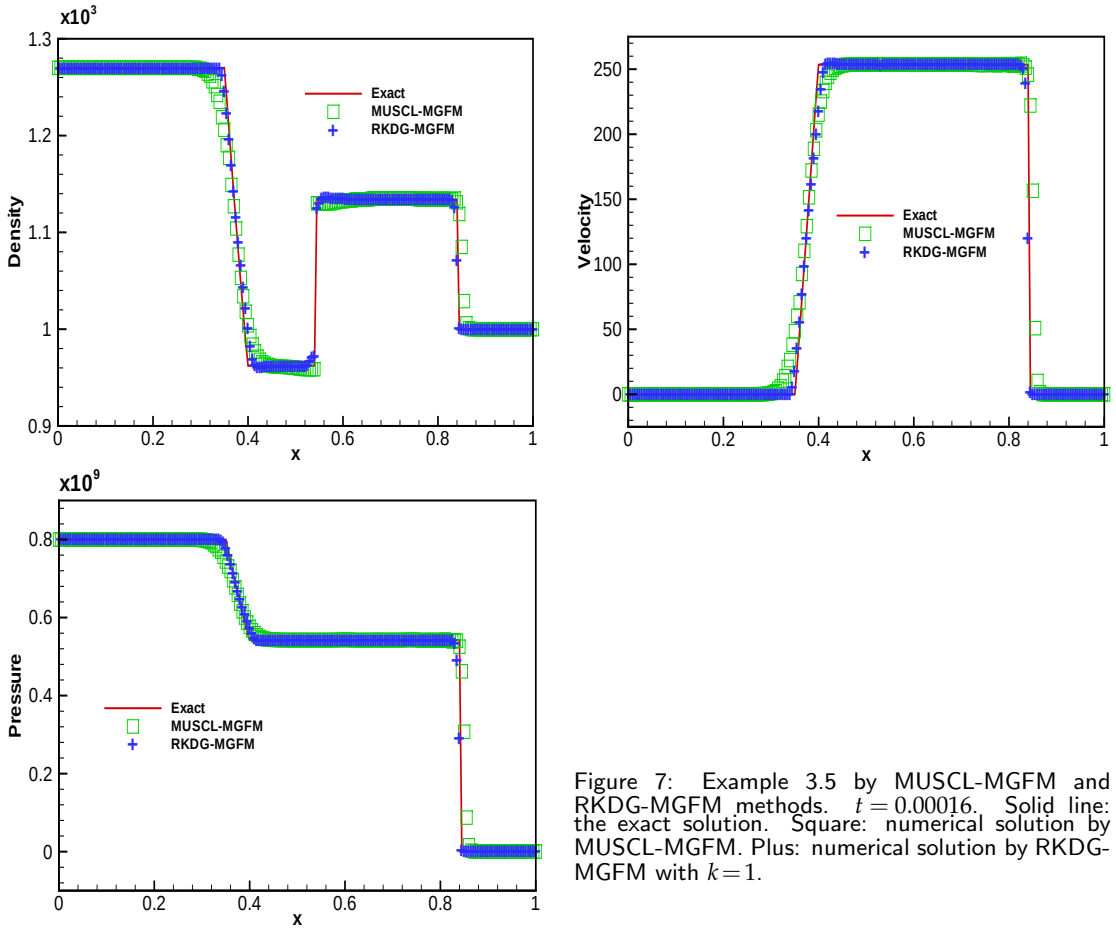

Figure 7: Example 3.5 by MUSCL-MGFM and RKDG-MGFM methods. $t = 0.00016$. Solid line: the exact solution. Square: numerical solution by MUSCL-MGFM. Plus: numerical solution by RKDG-MGFM with $k=1$.

The computed density ρ, velocity v and pressure p are plotted at $t=0.0001$ against the exact solution in Fig. 8. In this example, we choose the TVB limiter constant $M_1=1$ and $M_2=1$ for the case of $k=1$, and $M_1=4$ and $M_2=4$ for the case of $k=2$.

We can observe that the computed gas-water interface is found located at the correct cell by both the RKDG-GFM and RKDG-MGFM schemes with $k=1$ and $k=2$. The computed results of density by the RKDG-GFM schemes with $k=1$ and $k=2$ exhibit undershoot and overshoot before and after the interface of gas and water, while those by the RKDG-MGFM schemes with $k=1$ and $k=2$ are effectively oscillatory free. It is obvious that the computed results for density by the RKDG-MGFM schemes are much better than those by the RKDG-GFM schemes.

Example 3.7. We consider the two dimensional Euler equation (1.1).

We examine an initial Mach 1.22 air shock impacting on a helium bubble. The schematic for this problem are given in Fig. 9, where the upper and lower boundary conditions are nonreflective open boundaries. The left and right boundary conditions

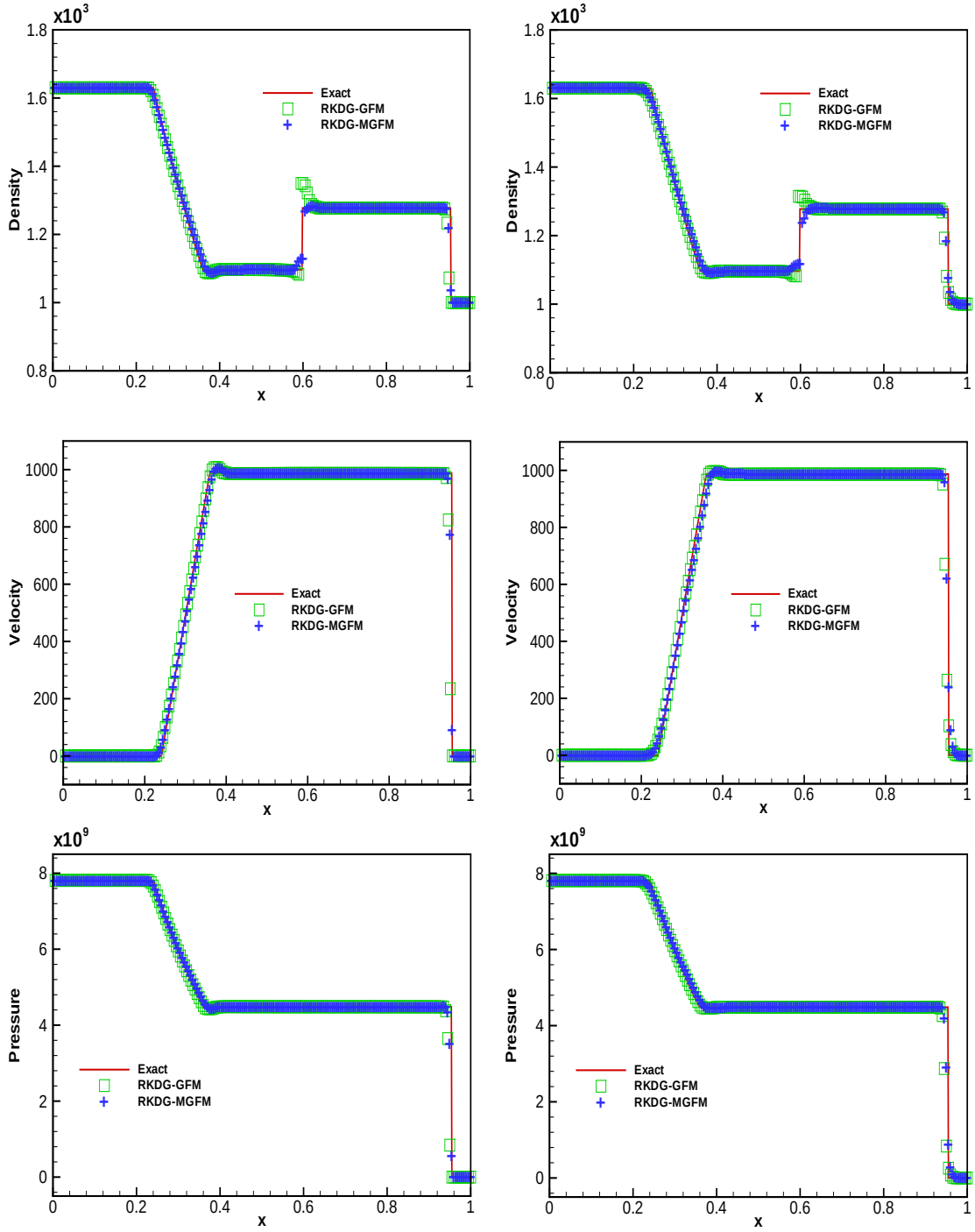

Figure 8: Example 3.6 by RKDG-GFM and RKDG-MGFM methods. $t=0.0001$. Solid line: the exact solution. Square: numerical solution by RKDG-GFM. Plus: numerical solution by RKDG-MGFM. Top: Density; Middle: Velocity; Bottom: Pressure. Left: $k=1$; Right: $k=2$.

are the inflow and outflow, respectively. The nondimensionalized initial conditions are:

$$(\rho,u,v,p,\gamma)=\begin{cases}(1,0,0,1,1.4), & \text{for pre-shocked air,}\\ (1.3764,0.394,0,1.5698,1.4), & \text{for post-shocked air,}\\ (0.138,0,0,1,\frac{5}{3}), & \text{for helium.}\end{cases}$$

and the level set function $\phi=\sqrt{x^2+y^2}-1$, where $\phi\leq 0$ represents helium and $\phi>0$ represents the air. The post-shock air state is given for $x<-1.2$.

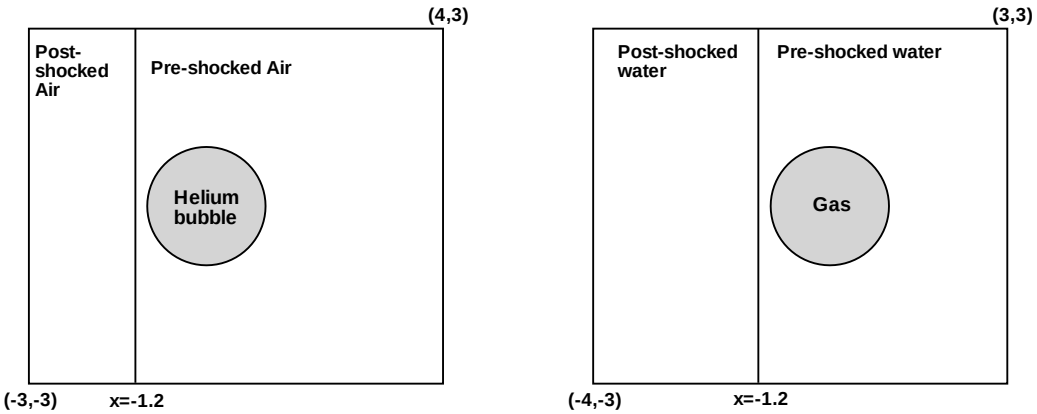

Figure 9: Physical domain for Example 3.7 (left) and Example 3.8 (right).

This problem has been experimentally investigated in [17] and numerically studied in detail in [19,33]. Very complex physics occurs in this problem especially after a re-entrant jet forms and impact the rear of the helium bubble. To capture the complex physics occurring in the late time stage, very fine mesh or adaptive mesh refinement technique has to be employed as done in [19,33]. In this work, our computation stopped before the formation of the re-entrant jet. The computed results was quite comparable to the results shown in [17] even over a relatively coarse mesh.

The computed density ρ with 280×240 uniform cells is plotted at $t=0.5$, $t=1.0$, $t=2.0$ and $t=4.0$ in Fig. 10. In this case, we choose the TVB limiter constant $M_1=0.1$ and $M_2=0.1$ for both the cases of $k=1$ and $k=2$.

When the incident shock hits the bubble, the incident shock refracts at the bubble surface. The incident shock is partly transmitted inside the helium bubble and partly reflected from the bubble surface and back into the air. As the sound speed in helium is faster than that in air, this causes the initial regular shock refraction soon becomes irregular and the bifurcation of the incident shock on the bubble surface, leading to the formation of a precursor to the incident shock as indicated by the experimental results and also the present numerical results at $t=0.5$ shown in Fig. 10. At $t=1.0$, the transmitted shock inside the bubble has interacted with the rear of the bubble and transmits into the air, while the incident shock just passed over the top of the bubble. With the transmission

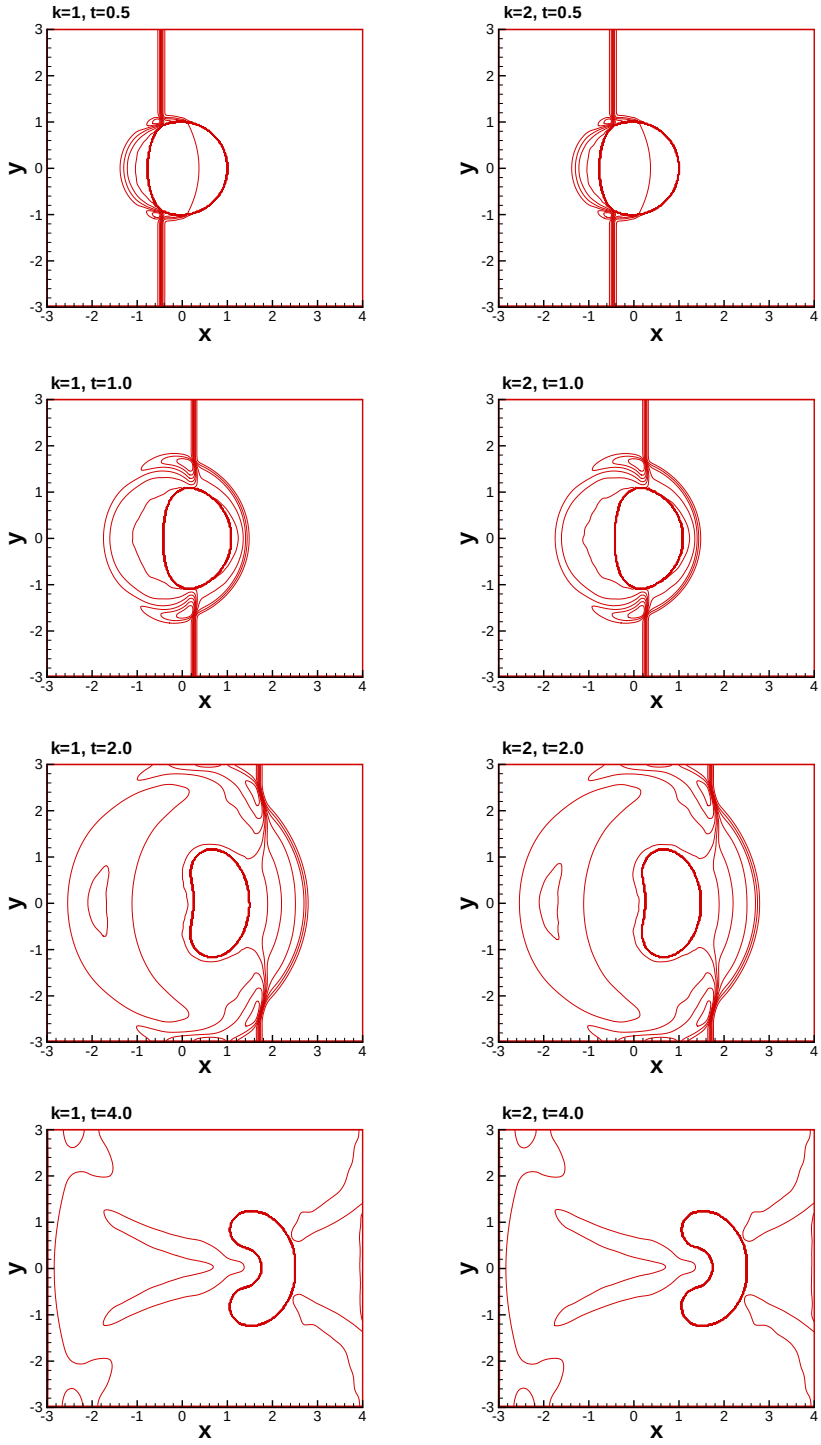

Figure 10: Example 3.7 by RKDG-MGFM method with 280×240 cells, 30 equally spaced density contours from 0.1 to 1.6. Left: $k=1$; Right: $k=2$. From top to bottom are $t=0.5$, $t=1.0$, $t=2.0$, $t=4.0$, respectively.

of the shock inside the bubble into the air, the whole bubble starts to moves towards the right. At $t=2.0$, the incident shock has passed over the whole bubble and the bubble has deformed and evolved into a kidney shape. After that, a re-entrant jet starts to form. At $t=4.0$, the re-entrant jet has been formed. As the re-entrant jet becomes stronger and stronger, the interface instability occurs. As time further goes, the re-entrant jet impacts the rear side of the bubble and bubble collapses. To capture all these later stages fine physics, the adaptive mesh refinement has to be applied. Our computation stopped at the nondimensional time $t=4.0$.

For this problem, the RKDG-MGFM schemes with $k=1$ and $k=2$ provide very similar results.

Example 3.8. In the final problem, we consider a planar underwater shock interacting with a gas bubble in an open domain. Similar problems have previously been studied in [16, 25, 27].

We examine an initial Mach 1.653 underwater shock wave makes impact on a gas bubble. The schematic for this problem are given in Fig. 9. The nondimensionalized initial conditions are:

$$(\rho,u,v,p,\gamma)=\begin{cases}(1000,0,0,1,7.15), & \text{for pre-shocked water,}\\ (1176.3333,1.1692,0,9120,7.15), & \text{for post-shocked water,}\\ (1,0,0,1,1.4), & \text{for gas.}\end{cases}$$

and the level set function $\phi=\sqrt{x^2+y^2}-1$, where $\phi\leq 0$ represents the gas and $\phi>0$ represents the water. The post-shock water state is given for $x<-1.2$, as shown in Fig. 9.

Again, very complex physics occurs in this problem especially when a jet forms and bubble collapses in the very late stage. To capture, the later stage fine physics, very fine mesh or adaptive mesh has to be applied as done in [27]. Our computations stopped just before bubble collapse.

In the earlier stage, the shock refraction on the bubble surface is regular initially but transits into an irregular type after the incident shock past over a critical angle [16, 25]. The detailed physical analysis can be found in [25] especially for the earlier stage. For the detailed physics in the late time stage, one may refer to [27].

In our computation, 280×240 uniform cells was used. Fig. 11 show the computed density ρ at $t=0.06$, $t=0.19$, $t=0.357$, and $t=0.481$, respectively. In this case, we choose the TVB limiter constant $M_1=0.1$ and $M_2=0.1$ for both the cases of $k=1$ and $k=2$. The RKDG-MGFM schemes with $k=1$ and $k=2$ show similar results.

4 Concluding remarks

We have investigated extensively using the discontinuous Galerkin finite element methods for two-medium flow simulations in one and two dimensions via incorporating the ghost fluid method (GFM) and the modified ghost fluid method (MGFM) to treat the

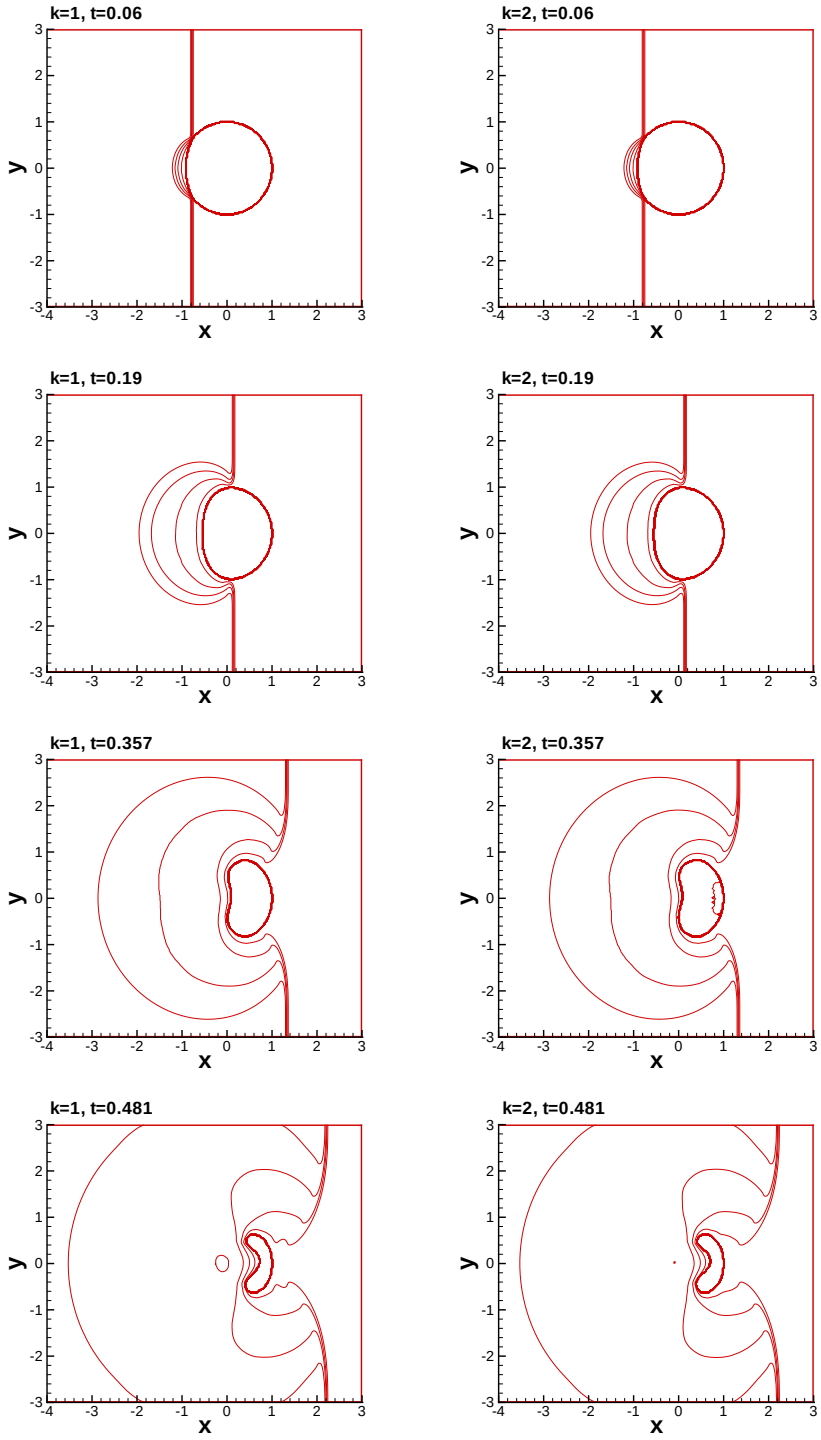

Figure 11: Example 3.8 by RKDG-MGFM method with 280×240 cells, 30 equally spaced density contours from 1.0 to 1200. Left: $k=1$; Right: $k=2$. From top to bottom are $t=0.06$, $t=0.19$, $t=0.357$, $t=0.481$, respectively.

moving material interfaces. Numerical results for both gas-gas and gas-water flows in one and two dimensions have been provided to illustrate the behavior of this procedure. Numerical results show that behavior of RKDG with the modified ghost fluid method is perceptibly better than that by the RKDG with the original ghost fluid method.

Acknowledgments

The research of the first author was supported by NSFC grant 10671091, Nanjing University Talent Development Foundation and SRF for ROCS, SEM. Additional support was provided by NUS Research Project R-265-000-118-112 while he was in residence at the Department of Mechanical Engineering, National University of Singapore, Singapore 119260.

References

[1] R. Abgrall, How to prevent pressure oscillations in multicomponent flow calculations: a quasi-conservative approach, J. Comput. Phys., 125 (1996), 150-160.
[2] R. Abgrall and S. Karni, Computations of compressible multifluids, J. Comput. Phys., 169 (2001), 594-623.
[3] D. A. Bailey, P. K. Sweby and P. Glaister, A ghost fluid, moving finite volume plus continuous remap method for compressible Euler flow, Int. J. Numer. Meth. Fluids, 47 (2005), 833-840.
[4] E. H. van Brummelen and B. Koren, A pressure-invariant conservative Godunov-type method for barotropic two-fluid flows, J. Comput. Phys., 185 (2003), 289-308.
[5] 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), 1-27.
[6] T.-J. Chen and C. H. Cooke, On the Riemann problem for liquid or gas-liquid media, Int. J. Numer. Meth. Fluids, 18 (1994), 529-541.
[7] J.-P. Cocchi and R. Saurel, A Riemann problem based method for the resolution of compressible multimaterial flows, J. Comput. Phys., 137 (1997), 265-298.
[8] 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), 545-581.
[9] 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), 90-113.
[10] 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), 411-435.
[11] 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), 337-361.
[12] B. Cockburn and C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys., 141 (1998), 199-224.

[13] R. P. Fedkiw, T. Aslam, B. Merriman and S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys., 152 (1999), 457-492.
[14] R. P. Fedkiw, Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method, J. Comput. Phys., 175 (2002), 200-224.
[15] J. Glimm, X. L. Li, Y. Liu and N. Zhao, Conservative front tracking and level set algorithms, P. Natl. Acad. Sci., 98 (2001), 14198-14201.
[16] J. Grove and R. Menikoff, Anomalous reflection of a shock wave at a fluid interface, J. Fluid Mech., 219 (1990), 313-336.
[17] J.-F. Haas and B. Sturtevant, Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities, J. Fluid Mech., 181 (1987), 41-76.
[18] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys., 49 (1983), 357-393.
[19] E. Johnsen and T. Colonius, Implementation of WENO schemes in compressible multicomponent flow problems, J. Comput. Phys., 219 (2006), 715-732.
[20] S. Karni, Multi-component flow calculations by a consistent primitive algorithm, J. Comput. Phys., 112 (1994), 31-43.
[21] B. Koren, M. R. Lewis, E. H. van Brummelen and B. van Leer, Riemann-problem and level-set approaches for homentropic two-fluid flow computations, J. Comput. Phys., 181 (2002), 654-674.
[22] B. Larrouturou, How to preserve the mass fractions positivity when computing compressible multi-component flow, J. Comput. Phys., 95 (1991), 31-43.
[23] T. G. Liu, B. C. Khoo and C. W. Wang, The ghost fluid method for compressible gas-water simulation, J. Comput. Phys., 204 (2005), 193-221.
[24] T. G. Liu, B. C. Khoo and K. S. Yeo, The simulation of compressible multi-medium flow. Part I: a new methodology with applications to 1D gas-gas and gas-water cases, Comput. Fluids,, 30 (2001), 291-314.
[25] T. G. Liu, B. C. Khoo and K. S. Yeo, The simulation of compressible multi-medium flow. Part II: applications to 2D underwater shock refraction, Comput. Fluids, 30 (2001), 315-337.
[26] T. G. Liu, B. C. Khoo and K. S. Yeo, Ghost fluid method for strong shock impacting on material interface, J. Comput. Phys., 190 (2003), 651-681.
[27] R. R. Nourgaliev, T. N. Dinh and T. G. Theofanous, Adaptive characteristic-based matching for compressible multifluid dynamics, J. Comput. Phys., 213 (2006), 500-529.
[28] S. Osher and R. P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys., 169 (2001), 463-502.
[29] S. Osher and A. Sethian, Fronts progahating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulation, J. Comput. Phys., 79 (1988), 12-49.
[30] J. Qiu, T. G. Liu and B. C. Khoo, Runge-Kutta discontinuous Galerkin methods for two-medium flow simulations: one-dimensional case, J. Comput. Phys., 222 (2007), 353-373.
[31] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput., 26 (2005), 907-929.
[32] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one dimensional case, J. Comput. Phys., 193 (2004), 115-135.
[33] J. J. Quirk and S. Karni, On the dynamics of a shock-bubble interaction, J. Fluid Mech., 318 (1996), 129-163.
[34] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comput., 49

(1987), 105-121.
[35] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput., 9 (1988), 1073-1084.
[36] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys., 83 (1989), 32-78.
[37] 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), 278-302.