Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 222 (2007) 353–373

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Runge–Kutta discontinuous Galerkin methods for compressible two-medium flow simulations: One-dimensional case

Jianxian Qiu [a,b,*,1], Tiegang Liu [c], Boo Cheong Khoo [b,d,2]

[a] *Department of Mathematics, Nanjing University, 22 Hankou Road, Nanjing, Jiangsu 210093, PR China*
[b] *Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore*
[c] *Institute of High Performance Computing, ♯01-01 The Capricorn, Singapore Science Park II, Singapore 117528, Singapore*
[d] *Singapore-MIT Alliance, 4 Engineering Drive 3, National University of Singapore, Singapore 117576, Singapore*

Received 24 November 2005; received in revised form 4 July 2006; accepted 25 July 2006
Available online 7 September 2006

Abstract

The Runge–Kutta discontinuous Galerkin (RKDG) method for solving hyperbolic conservation laws is a high order finite element method, which utilizes 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 the RKDG finite element method for compressible two-medium flow simulation with conservative treatment of the moving material interfaces. Numerical results for both gas–gas and gas–water flows in one-dimension are provided to demonstrate the characteristic behavior of this approach.

AMS: 65M60; 65M99; 35L65

Keywords: Runge–Kutta discontinuous Galerkin method; WENO scheme; Ghost fluid method; Approximate Riemann problem solver

1. Introduction

In this paper, we investigate using the Runge–Kutta discontinuous Galerkin (RKDG) finite element method for compressible two-medium flow simulation in one-dimensional case.

[*] Corresponding author. Address: Department of Mathematics, Nanjing University, 22 Hankou Road, Nanjing, Jiangsu 210093, PR China.

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

[1] Research partially supported by NNSFC Grant 10371118, Nanjing University Talent Development Foundation and NUS Research Project R-265-000-118-112.

[2] Research partially supported by NUS Research Project R-265-000-118-112.

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.07.023

The first discontinuous Galerkin (DG) method was introduced in 1973 by Reed and Hill [28], in the framework of neutron transport (steady-state linear hyperbolic equations). A major development of the DG method was then carried out by Cockburn et al. in a series of papers [11,10,9,8,12], in which they established a framework to easily solve the *nonlinear* time dependent hyperbolic conservation laws:

$$\begin{cases} u_t + \nabla \cdot f(u) = 0 \\ u(x,0) = u_0(x) \end{cases} \tag{1.1}$$

using an explicit, nonlinearly stable high order Runge–Kutta time discretizations [30] and DG discretization in space with exact or approximate Riemann solvers as interface fluxes and limiters such as the total variation bounded (TVB) limiters [29] or weighted essential non-oscillatory (WENO) type limiters [26,27], to achieve non-oscillatory properties for strong shocks. These schemes are termed RKDG methods. RKDG methods have been widely applied and performed very well to solve the single-medium compressible flow.

A relatively dominant difficulty for simulating compressible two-medium flow is the treatment of moving material interfaces and their vicinities. Nonphysical oscillations usually occur in the vicinity of the material interface when a well-established numerical method for single-medium flow is directly applied to multi-medium flow. In the literature there are some methods developed to overcome this difficulty [19,17,1,7,21,2,4].

The ghost fluid method (GFM) developed by Fedkiw et al. [13] has provided an attractive and flexible way to treat the two-medium flow. The main appealing features of the GFM are its simplicity, easy extension to multi-dimensions and maintenance of a sharp interface without smearing. The GFM makes the interface ‘‘invisible’’ during computations by defining ghost cells and ghost fluids, and the 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. Since only single-fluid flux formulations are required to make GFM workable, the GFM is easily employed for two fluids of vastly different types such as a compressible-incompressible or viscous-inviscid two-fluid flow [5]. Variants of the original GFM [14] and other applications can also be found in [2,18]. Recently, efforts have also been made to develop a conservative GFM as found in [15,3].

On the other hand, it is precisely the manner of treatment of the single medium across the interface in the GFM that may cause numerical inaccuracy when there is a strong shock wave interacting with the interface [23]; this is especially so if such wave interaction with the interface is not taken into account properly in the definition of the ghost fluid state. This situation arises because the pattern of shock refraction at a material interface and the resultant interfacial status depend highly on material properties on both sides of the interface. As such, reasonable ghost fluid states have to be formulated to take into account the influence of both material properties and wave interaction with the interface. This has led to the development of a modified GFM (MGFM) with a predicted ghost fluid status by Liu et al. [23] via implicitly solving two non-linear characteristic equations interacting and applicable at the interface [21,22]. In fact, it has been found that Conditions have to be satisfied for the ghost fluid state in order that the two GFM Riemann provides the correct solution in the respective real fluids [20]. Those techniques developed in [21,22] will also be used to calculate the flow interface state in the multi-medium RKDG algorithm as proposed in this work.

In general, algorithms proposed for solving two-medium compressible flow consist of two parts. One is the method for solving the single-medium flow and the other is to treat the interface of the two fluids. In [13,5,14], the 3rd order ENO methods are used together with the GFM for treating the interface, while the Godunov-type or MUSCL (monotone upwind schemes for conservation laws) are used in [1,2,21,22,4,23,20].

In this paper our intent is to investigate using the RKDG finite element method for multi-medium flow simulations in one-dimensional case. Similar to the above-mentioned algorithms developed for multi-medium compressible flows, the present method to be proposed also consists of two parts. One is the usual RKDG algorithm applicable for the flow field away from the material interface; the other is the newly developed DG technique for treating the moving interface. In Section 2, we first briefly review the usual RKDG method over a fixed and regular mesh system, and then we describe in detail the extension of the DG to treat the moving material interfaces conservatively. Extensive numerical results are presented in Section 3 to illustrate the characteristic behavior of the RKDG method presented in Section 2. Concluding remarks are given in Section 4.

2. Implementation of RKDG methods for two-medium flow simulations

In this section, we describe in detail the construction and implementation of RKDG methods for two-medium inviscid compressible flow simulations. We consider the one-dimensional Euler equation

$$u_t + f(u)_x = 0, \tag{2.1}$$

with

$$u = (\rho, m, E)^{\mathrm{T}}, \quad f(u) = (m, mv + p, v(E+p))^{\mathrm{T}}.$$

Here ρ is the density, v is the velocity, $m = \rho v$ is the moment, E is the total energy and p is the pressure which is related to the total energy by $E = \rho e + \frac{1}{2}\rho v^2$, where e is the specific internal energy per unit mass. For closure of the system, the equation of state (EOS) is required. The γ-law used for gases is given as

$$\rho e = p/(\gamma - 1) \tag{2.2}$$

and Tait EOS used for the water medium [6,13,21] is expressed as:

$$\rho e = (p + N\overline{B})/(N-1), \tag{2.3}$$

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

2.1. Description of RKDG methods

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 $I_i = \left[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\right]$, the cell centers by $x_i = \frac{1}{2}\left(x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}}\right)$ and the cell sizes by $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$, $h = \inf_i \Delta x_i$. 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 $\leqslant k$ on the cell I_i. We adopt a local orthogonal basis over I_i, $\{\phi_l^{(i)}(x), l = 0, 1, \ldots, k\}$, namely the scaled Legendre polynomials:

$$\phi_0^{(i)}(x) = 1, \quad \phi_1^{(i)}(x) = \frac{x - x_i}{\Delta x_i/2}, \quad \phi_2^{(i)}(x) = \left(\frac{x - x_i}{\Delta x_i/2}\right)^2 - \frac{1}{3}, \cdots$$

The numerical solution $u^h(x,t)$ of Eq. (2.1) in the test function space V_h^k can then be written as

$$u^h(x,t) = \sum_{l=0}^{k} u_i^{(l)}(t)\phi_l^{(i)}(x), \quad \text{for } x \in I_i \tag{2.4}$$

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

$$u_i^{(l)}(t) = \frac{1}{a_l}\int_{I_i} u^h(x,t)\phi_l^{(i)}(x)\,\mathrm{d}x, \quad l = 0, 1, \ldots, k,$$

where $a_l = \int_{I_i}(\phi_l^{(i)}(x))^2\,\mathrm{d}x$ 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 (2.4) into Eq. (2.1), and multiplying (2.1) with a test function $\phi_l^{(i)}(x)$, integrating over a cell I_i and integrating by parts, we can obtain the governing equations for the moments as

$$\frac{\mathrm{d}}{\mathrm{d}t}u_i^{(l)} + \frac{1}{a_l}\left(-\int_{I_i} f(u^h(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(i)}(x)\,\mathrm{d}x + f(u^h(x_{i+1/2},t))\phi_l^{(i)}(x_{i+1/2}) - f(u^h(x_{i-1/2},t))\phi_l^{(i)}(x_{i-1/2})\right) = 0,$$
$$l = 0, 1, \ldots, k. \tag{2.5}$$

Eq. (2.5) 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

$$\frac{\mathrm{d}}{\mathrm{d}t}u_i^{(l)} + \frac{1}{a_l}\left(-\int_{I_i} f(u^h(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(i)}(x)\,\mathrm{d}x + \hat{f}(u_{i+1/2}^-, u_{i+1/2}^+)\phi_l^{(i)}(x_{i+1/2}) - \hat{f}(u_{i-1/2}^-, u_{i-1/2}^+)\phi_l^{(i)}(x_{i-1/2})\right) = 0,$$
$$l = 0, 1, \ldots, k, \tag{2.6}$$

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

$$\hat{f}(u^-, u^+) = \frac{1}{2}[(f(u^-) + f(u^+)) - \alpha(u^+ - u^-)],$$

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 (2.6) can be computed either exactly or by a suitable numerical quadrature accurate to at least $\mathrm{O}(\Delta x_i^{k+l+2})$. In this paper, we use three and four point Gauss–Lobatto quadrature for P^1 and P^2 cases, respectively.

The semi-discrete scheme (2.6) 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 [30]:

$$\begin{aligned}
u^{(1)} &= u^n + \Delta t L(u^n),\\
u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}),\\
u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}).
\end{aligned} \tag{2.7}$$

If there are strong discontinuities in the solution, the scheme (2.7) generates significant oscillations and even nonlinear instability. To avoid such difficulties, typically a slope limiter is used after each Runge–Kutta inner stage (or after the complete Runge–Kutta time step) to control the numerical oscillations. The limiter adopted in [10] is described below in some detail. We denote

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

Here

$$\tilde{u}_i = \sum_{l=1}^{k} u_i^{(l)}\phi_l^{(i)}(x_{i+1/2}), \quad \tilde{\tilde{u}}_i = -\sum_{l=1}^{k} u_i^{(l)}\phi_l^{(i)}(x_{i-1/2}).$$

$\tilde{u}_i$ and $\tilde{\tilde{u}}_i$ are modified by either the standard minmod limiter [16]

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

where m is given by

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

or the TVB modified minmod function [29]

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

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

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] for details.

Let $x = x^*(t^n)$ be the interface of two different fluids at $t = t^n$, and $x^*(t^n) \in I_j$; then we evolve (2.6) for Fluid I from $i = 1$ to $j - 2$ and for Fluid II from $i = j + 2$ to N, respectively. The computation for cells next to the interface will be described in detail in the next subsection.

2.2. Description of treatment of interface cell

We use the level set method to compute the location of the interface $x^*(t^{n+1})$ at $t = t^{n+1}$ based on the data at $t = t^n$. We will not repeat the level set technique here. One may refer to [25,24,13,21] for the details of level set methods.

Next, we present the method to obtain the solution for the cells next to the interface based on conservation laws. To avoid too small spatial step size, we merge cells I_{j-1}, I_j and I_{j+1} to form two new interface cells occupied by Fluid I and Fluid II, respectively, namely, cell $[x_{j-3/2}, x^*(t)]$ occupied by Fluid I and cell $[x^*(t), x_{j+3/2}]$ occupied by Fluid II. We can construct a DG algorithm for Fluid I over interval $[x_{j-3/2}, x^*(t)]$ and for Fluid II over interval $[x^*(t), x_{j+3/2}]$ by the following formulae, respectively

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{j-3/2}}^{x^*(t)} u(x,t)\phi_l^{(j)}(x)\,\mathrm{d}x &= \int_{x_{j-3/2}}^{x^*(t)} \frac{\partial u(x,t)}{\partial t}\phi_l^{(j)}(x)\,\mathrm{d}x + \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\phi_l^{(j)}(x^*(t))\\
&= -\int_{x_{j-3/2}}^{x^*(t)} \frac{\partial f(u(x,t))}{\partial x}\phi_l^{(j)}(x)\,\mathrm{d}x + \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\phi_l^{(j)}(x^*(t))\\
&= -\left[f(u(x^*(t),t)) - \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\right]\phi_l^{(j)}(x^*(t)) + f(u(x_{j-3/2},t))\phi_l^{(j)}(x_{j-3/2})\\
&\quad + \int_{x_{j-3/2}}^{x^*(t)} f(u(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(j)}(x)\,\mathrm{d}x, \quad l = 0,1,\ldots,k. \end{aligned}\tag{2.10}$$

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{x^*(t)}^{x_{j+3/2}} u(x,t)\phi_l^{(j)}(x)\,\mathrm{d}x &= \int_{x^*(t)}^{x_{j+3/2}} \frac{\partial u(x,t)}{\partial t}\phi_l^{(j)}(x)\,\mathrm{d}x - \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\phi_l^{(j)}(x^*(t))\\
&= -\int_{x^*(t)}^{x_{j+3/2}} \frac{\partial f(u(x,t))}{\partial x}\phi_l^{(j)}(x)\,\mathrm{d}x - \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\phi_l^{(j)}(x^*(t))\\
&= \left[f(u(x^*(t),t)) - \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)\right]\phi_l^{(j)}(x^*(t)) - f(u(x_{j+3/2},t))\phi_l^{(j)}(x_{j+3/2})\\
&\quad + \int_{x^*(t)}^{x_{j+3/2}} f(u(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(j)}(x)\,\mathrm{d}x, \quad l = 0,1,\ldots,k. \end{aligned}\tag{2.11}$$

In order to discretize Eqs. (2.10) and (2.11), we assume the numerical solution of $u(x,t)$ over the interface cell $[x_{j-3/2}, x^*(t)]$ for Fluid I as $u_{\mathrm{I}}^h(x,t) = \sum_{m=0}^k u_{\mathrm{I}}^{(m)}(t)\phi_m^{(j)}(x)$, and over the interface cell $[x^*(t), x_{j+3/2}]$ for Fluid II as $u_{\mathrm{II}}^h(x,t) = \sum_{m=0}^k u_{\mathrm{II}}^{(m)}(t)\phi_m^{(j)}(x)$. Here the base functions $\{\phi_l^{(j)}\}$ are extended to be valid also over cell $[x_{j-3/2}, x_{j+3/2}]$. This results in the non-orthogonality of base functions $\{\phi_l^{(j)}\}$ in the interval $[x_{j-3/2}, x^*(t)]$ and in $[x^*(t), x_{j+3/2}]$. Substituting $u_{\mathrm{I}}^h(x,t) = \sum_{m=0}^k u_{\mathrm{I}}^{(m)}(t)\phi_m^{(j)}(x)$ into (2.10), and $u_{\mathrm{II}}^h(x,t) = \sum_{m=0}^k u_{\mathrm{II}}^{(m)}(t)\phi_m^{(j)}(x)$ into (2.11), we obtain:

$$\begin{aligned}\frac{\mathrm{d}\bar{u}_I^{(l)}}{\mathrm{d}t} &= -f^*(u_{\mathrm{I}}(x^*(t),t))\phi_l^{(j)}(x^*(t)) + f(u_{\mathrm{I}}^h(x_{j-3/2},t))\phi_l^{(j)}(x_{j-3/2}) + \int_{x_{j-3/2}}^{x^*(t)} f(u_I^h(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(j)}(x)\,\mathrm{d}x,\\
&\quad l = 0,1,\ldots,k.\end{aligned}\tag{2.12}$$

and

$$\begin{aligned}\frac{\mathrm{d}\bar{u}_{\mathrm{II}}^{(l)}}{\mathrm{d}t} &= -f(u_{\mathrm{II}}^h(x_{j+3/2},t))\phi_l^{(j)}(x_{j+3/2}) + f^*(u_{\mathrm{II}}(x^*(t),t))\phi_l^{(j)}(x^*(t)) + \int_{x^*(t)}^{x_{j+3/2}} f(u_{\mathrm{II}}^h(x,t))\frac{\mathrm{d}}{\mathrm{d}x}\phi_l^{(j)}(x)\,\mathrm{d}x,\\
&\quad l = 0,1,\ldots,k.\end{aligned}\tag{2.13}$$

respectively. Here, $\bar{u}_{\mathrm{I}}^{(l)} = \int_{x_{j-3/2}}^{x^*(t)} u_{\mathrm{I}}^h(x,t)\phi_l^{(j)}(x)\,\mathrm{d}x$, $\bar{u}_{\mathrm{II}}^{(l)} = \int_{x^*(t)}^{x_{j+3/2}} u_{\mathrm{II}}^h(x,t)\phi_l^{(j)}(x)\,\mathrm{d}x$, and $f^*(u(x^*(t),t)) = f(u(x^*(t),t)) - \frac{\mathrm{d}x^*(t)}{\mathrm{d}t}u(x^*(t),t)$. The ODE system (2.12) and (2.13) can be numerically solved using (2.7) once we know how to evaluate the right hand sides of both (2.12) and (2.13).

Below are listed the details of evaluating the right hand sides of (2.12) and (2.13).

1. We replace the fluxes $f(u_{\mathrm{I}}^h(x_{j-3/2},t))$ and $f(u_{\mathrm{II}}^h(x_{j+3/2},t))$ with numerical fluxes (here, the Lax–Friedrichs flux) $\hat{f}(u_{j-3/2}^-, u_{j-3/2}^+)$ and $\hat{f}(u_{j+3/2}^-, u_{j+3/2}^+)$, which are the same as those in (2.6) computed in the single medium for Fluid I and for Fluid II, respectively.
2. We solve a multi-medium Riemann problem defined at the interface with status $u^-(x^*(t),t)$ on the left of the interface for Fluid I and $u^+(x^*(t),t)$ on the right for Fluid II to obtain numerical flux at the interface of two fluids. For example, for Euler equation (2.1), we adopted the Riemann problem solver developed in [21,23] to solve the multi-medium Riemann problem to get pressure $p(x^*(t))$ and velocity $v(x^*(t))$ at the interface which are assumed to be continuous across the interface. In Euler equation, we have

$$\frac{\mathrm{d}x^*(t)}{\mathrm{d}t} = v(x^*(t),t).$$

Then the numerical flux at the interface of two fluids can be written as:

$$\hat{f}^* = (0, p(x^*(t)), p(x^*(t))v(x^*(t)))^{\mathrm{T}}.$$

This flux at the interface is used for computation in both interface cells for Fluid I and Fluid II.

3. The spatial integral terms on the right hand sides of (2.12) and (2.13) are computed by the three and four point Gauss–Lobatto quadrature for P^1 and P^2 cases, respectively.

Remark. The formula of treatment of interface cells is conservative. Summing the numerical formulae (2.12) and (2.13) for $l = 0$ we can obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_{j-3/2}}^{x_{j+3/2}} u(x,t)\,\mathrm{d}x = \frac{\mathrm{d}\bar{u}_{\mathrm{I}}^{(0)}}{\mathrm{d}t} + \frac{\mathrm{d}\bar{u}_{\mathrm{II}}^{(0)}}{\mathrm{d}t} = \hat{f}(u_{j-3/2}^-, u_{j-3/2}^+) - \hat{f}(u_{j+3/2}^-, u_{j+3/2}^+). \tag{2.14}$$

It is clear that the formula (2.14) is conservative.

The distance of the interface travelling at every time step is less than CFL · h, where CFL is the CFL number. After each time step, there are three possible locations for the interface: case 1 where $x^*(t^{n+1}) \in I_j$, case 2 where $x^*(t^{n+1}) \in I_{j+1}$ and case 3 where $x^*(t^{n+1}) \in I_{j-1}$.

For case 1, the interface is still in the same cell as the previous time step, there is no additional treatment made to the interface cells.

For case 2, the interface has crossed the cell boundary $x = x_{j+1/2}$ as time progresses from t^n to t^{n+1}. Cell I_{j-1} will then be computed as a normal cell using the usual RKDG algorithm at the next time step, while cells I_j, I_{j+1} and I_{j+2} will be treated as interface cells and calculated using the newly developed formula (2.12) and (2.13). We first split the interface cell $[x_{j-3/2}, x^*(t^{n+1})]$ for Fluid I into a normal cell I_{j-1} and a new interface cell $[x_{j-1/2}, x^*(t^{n+1})]$. Then the solution, $u_{\mathrm{I}}^h(x,t^{n+1}) = \sum_{m=0}^k u_{\mathrm{I}}^{(m)}(t^{n+1})\phi_m^{(j)}(x), \quad x \in [x_{j-3/2}, x^*(t^{n+1})]$, is mapped to the cell I_{j-1} to obtain $u_{j-1}^h(x,t^{n+1}) = \sum_{m=0}^k u_{j-1}^{(m)}(t^{n+1})\phi_m^{(j-1)}(x), \quad x \in [x_{j-3/2}, x_{j-1/2}]$ and to the new interface cell $[x_{j-1/2}, x^*(t^{n+1})]$ to obtain $u_{\mathrm{I,new}}^h(x,t^{n+1}) = \sum_{m=0}^k u_{\mathrm{I,new}}^{(m)}(t^{n+1})\phi_m^{(j+1)}(x)$, $x \in [x_{j-1/2}, x^*(t^{n+1})]$. More specifically, the moments $u_{j-1}^{(l)}(t^{n+1})$ and $u_{\mathrm{I,new}}^{(m)}(t^{n+1})$ are obtained using the following expressions, respectively, as

$$(u_{j-1}^{(0)}, \cdots, u_{j-1}^{(k)})^{\mathrm{T}} = M^{-1}(b_0, \cdots, b_k)^{\mathrm{T}}, \quad M = (m_{\mathrm{st}})_{(k+1)\times(k+1)},$$

$$m_{\mathrm{st}} = \int_{I_{j-1}} \phi_s^{(j-1)}(x)\phi_t^{(j-1)}(x)\,\mathrm{d}x, \quad b_s = \int_{I_{j-1}} u_{\mathrm{I}}^h(x,t^{n+1})\phi_s^{(j-1)}(x)\,\mathrm{d}x, \quad s,t = 0,\ldots,k,$$

and

$$(u_{\mathrm{I,new}}^{(0)}, \cdots, u_{\mathrm{I,new}}^{(k)})^{\mathrm{T}} = M^{-1}(b_0, \cdots, b_k)^{\mathrm{T}}, \quad M = (m_{\mathrm{st}})_{(k+1)\times(k+1)},$$
$$m_{\mathrm{st}} = \int_{x_{j-1/2}}^{x^*(t^{n+1})} \phi_s^{(j+1)}(x)\phi_t^{(j+1)}(x)\,\mathrm{d}x, \quad b_s = \int_{x_{j-1/2}}^{x^*(t^{n+1})} u_{\mathrm{I}}^h(x, t^{n+1})\phi_s^{(j+1)}(x)\,\mathrm{d}x, \quad s,t = 0,\ldots,k.$$

At the same time, we merge the old interface cell $[x^*(t^{n+1}), x_{j+3/2}]$ (where the solution is $u_{\mathrm{II}}^h(x,t^{n+1}) = \sum_{m=0}^k u_{\mathrm{II}}^{(m)}(t^{n+1})\phi_m^{(j)}(x))$ with the normal cell $[x_{j+3/2}, x_{j+5/2}]$ (over which the solution is $u_{j+2}^h(x,t^{n+1}) = \sum_{m=0}^k u_{j+2}^{(m)}(t^{n+1})\phi_m^{(j+2)}(x))$, into a new interface cell $[x^*(t^{n+1}), x_{j+5/2}]$ for Fluid II and conservatively construct the solution over it as

$$u_{\mathrm{II,new}}^h(x,t) = \sum_{m=0}^k u_{\mathrm{II,new}}^{(m)}(t^{n+1})\phi_m^{(j+1)}(x), \quad x \in [x^*(t^{n+1}), x_{j+5/2}].$$

Here,

$$(u_{\mathrm{II,new}}^{(0)}, \cdots, u_{\mathrm{II,new}}^{(k)})^{\mathrm{T}} = M^{-1}(b_0, \cdots, b_k)^{\mathrm{T}},$$
$$M = (m_{\mathrm{st}})_{(k+1)\times(k+1)}, m_{\mathrm{st}} = \int_{x^*(t^{n+1})}^{x_{j+5/2}} \phi_s^{(j+1)}(x)\phi_t^{(j+1)}(x)\,\mathrm{d}x,$$
$$b_s = \int_{x^*(t^{n+1})}^{x_{j+3/2}} u_{\mathrm{II}}^h(x,t^{n+1})\phi_s^{(j+1)}(x)\,\mathrm{d}x + \int_{x_{j+3/2}}^{x_{j+5/2}} u_{j+2}^h(x,t^{n+1})\phi_s^{(j+1)}(x)\,\mathrm{d}x, \quad s,t = 0,\ldots,k.$$

For case 3, the interface has crossed the cell boundary $x = x_{j-1/2}$ as time progresses from t^n to t^{n+1}. Cell I_{j+1} will be recovered as a normal cell using the usual RKDG algorithm at the next step of computation, while cells I_{j-2}, I_{j-1} and I_j will be merged and calculated using the newly developed formula (2.12) and (2.13). We split the interface cell $[x^*(t^{n+1}), x_{j+3/2}]$ for Fluid II into a normal cell I_{j+1} and a new interface cell $[x^*(t^{n+1}), x_{j+1/2}]$ and then merge the interface cell $[x_{j-3/2}, x^*(t^{n+1})]$ with cell $[x_{j-5/2}, x_{j-3/2}]$ to form a new interface cell $[x_{j-5/2}, x^*(t^{n+1})]$ for Fluid I. The solution over the recovered cell I_{j+1} and the new interface cell can be obtained similarly as for case 2.

3. Numerical results

In this section, we present the results of our numerical experiments for the RKDG schemes described in section 2 for both single-medium and two-medium flows. In this paper, the CFL numbers are taken as 0.3 and 0.18 for RKDG2 ($k = 1$)and RKDG3 ($k = 2$), respectively. Unfortunately, the TVB limiter constant M is dependent on the problem, there is no an automatic switching which works well for various situations. As it was pointed out in [9], the resolution of solution is dependent on the choice of constant M; and sometimes, the case of $k = 1$ may give better resolution to shock or contact discontinuous than the case of $k = 2$; this can also be seen in our numerical tests. We have experimented and settled on the following procedure: instead of using a constant M, we use two constants as denoted by M_1 for the interface cells limiting procedure and the other M_2 for the limiting procedure for cells away from the interface, and choose M_1 and M_2 by trial and error in this paper.

3.1. Numerical tests for single-medium flows

In this subsection, we first test the accuracy of the schemes on linear scalar problems, nonlinear scalar problems and nonlinear systems. We only show the results of nonlinear scalar and nonlinear system problems to save space. We define the standard L_1 and L_∞ error norms by sampling the errors at 20 equally spaced points inside each cell, emulating the L_1 and L_∞ norms of the error function which is defined everywhere.

Example 3.1. We solve the following nonlinear scalar Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{3.1}$$

with the initial condition $u(x,0) = 0.5 + \sin(\pi x)$, with 2-periodic boundary conditions. The initial artificial interface is located at $x = 0.5$, and the artificial interface is moving at randomly perturbing the length within 20% of cell length at every time step because of the restriction of the CFL condition.

When $t = 0.5/\pi$ the solution is still smooth, and the errors and numerical orders of accuracy are shown in Tables 3.1 and 3.2. We can see that both the schemes for $k = 1$ (second-order) and $k = 2$ (third-order) essentially achieve their designed order of accuracy.

Example 3.2. We solve the following nonlinear system of Euler equations

$$u_t + f(u)_x = 0 \tag{3.2}$$

with

$$u = (\rho, \rho v, E)^{\mathrm{T}}, \quad f(u) = (\rho v, \rho v^2 + p, v(E + p))^{\mathrm{T}}.$$

Here E is the total energy given by $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho v^2$ with $\gamma = 1.4$. The initial condition is set to be $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, with 2-periodic boundary conditions. The exact solution is $\rho(x,t) = 1 + 0.2\sin(\pi(x - t))$, $v = 1$, $p = 1$. The initial artificial interface is located at $x = 0.5$, and the artificial interface is again moving at randomly perturbing the length within 20% of cell length at every time step. We compute the solution up to $t = 2$. The errors and numerical orders of accuracy are shown in Tables 3.3 and 3.4. We can also see that both the schemes for $k = 1$ (second-order) and $k = 2$ (third-order) have achieved their designed order of accuracy.

Example 3.3. We solve the same nonlinear Burgers Eq. (3.1) as in Example 3.1 with the same initial condition $u(x,0) = 0.5 + \sin(\pi x)$, except that we now plot the results at $t = 1.5/\pi$ when a shock has already appeared in the solution. The initial artificial interface is located at $x = 0.5$, and the artificial interface is moving at randomly perturbing the length within 20% of cell length at every time step. In Fig. 3.1, the solutions for $k = 1$ (left) and $k = 2$ (right) with $N = 80$ grid points are shown. The solid line is the exact solution. We can see that the schemes give good shock transitions even when limiter is not applied.

Table 3.1
$u_t + \left(\frac{u^2}{2}\right)_x = 0$, $u(x,0) = 0.5 + \sin(\pi x)$, with periodic boundary conditions, $t = 0.5/\pi$

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	1.82E − 02		1.02E − 01	
20	4.84E − 03	1.91	3.79E − 02	1.42
40	1.20E − 03	2.01	1.10E − 02	1.79
80	3.11E − 04	1.95	2.90E − 03	1.92
160	7.83E − 05	1.99	7.47E − 04	1.96
320	1.95E − 05	2.00	1.89E − 04	1.98

L_1 and L_∞ errors, $k = 1$.

Table 3.2
$u_t + \left(\frac{u^2}{2}\right)_x = 0$, $u(x,0) = 0.5 + \sin(\pi x)$, with periodic boundary conditions, $t = 0.5/\pi$

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	7.85E − 03		6.15E − 02	
20	3.25E − 04	4.59	4.61E − 03	3.74
40	3.67E − 05	3.15	7.61E − 04	2.60
80	4.34E − 06	3.08	1.07E − 04	2.84
160	5.06E − 07	3.10	1.46E − 05	2.86
320	6.39E − 08	2.99	1.92E − 06	2.93

L_1 and L_∞ errors, $k = 2$.

Table 3.3
Euler equations

N	L_1 error	L_1 order	L_∞ error	L_∞ order
10	8.09E − 03		2.09E − 02	
20	1.17E − 03	2.79	4.04E − 03	2.37
40	2.52E − 04	2.21	8.08E − 04	2.32
80	5.72E − 05	2.14	2.13E − 04	1.92
160	1.33E − 05	2.10	4.99E − 05	2.10
320	3.11E − 06	2.10	1.08E − 05	2.21

$\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, periodic boundary conditions, $t = 2$. L_1 and L_∞ errors of density ρ, $k = 1$.

Table 3.4
Euler equations

N	L_1 error	L_1 order	L_∞ error	L_∞order
10	6.34E − 04		2.88E − 03	
20	7.83E − 05	3.02	6.14E − 04	2.23
40	7.95E − 06	3.30	5.76E − 05	3.41
80	9.42E − 07	3.08	7.23E − 06	2.99
160	1.14E − 07	3.05	8.27E − 07	3.13
320	1.38E − 08	3.04	1.19E − 07	2.79

$\rho(x,0) = 1 + 0.2\sin(\pi x)$, $v(x,0) = 1$, $p(x,0) = 1$, periodic boundary conditions, $t = 2$. L_1 and L_∞ errors of density ρ, $k = 2$.

Example 3.4. We solve the one-dimensional nonlinear system of Euler equations (3.2). We use the following Riemann initial condition for the Lax problem:

$$(\rho, v, p) = (0.445, 0.698, 3.528) \quad \text{for } x \leqslant 0; \quad (\rho, v, p) = (0.5, 0, 0.571) \quad \text{for } x > 0.$$

The computed density ρ, velocity v and pressure p are plotted at $t = 1.3$ against the exact solution. In Figs. 3.2–3.4, the solutions of RKDG using $N = 200$ cells are shown. In this case, we choose the TVB limiter constant $M_1 = 0.1$, $M_2 = 50$ and $M_1 = 0.1$, $M_2 = 200$ for $k = 1$ and $k = 2$, respectively. We can see that the resolution of the contact discontinuity by RKDG is good for both the $k = 1$ and the $k = 2$ cases.

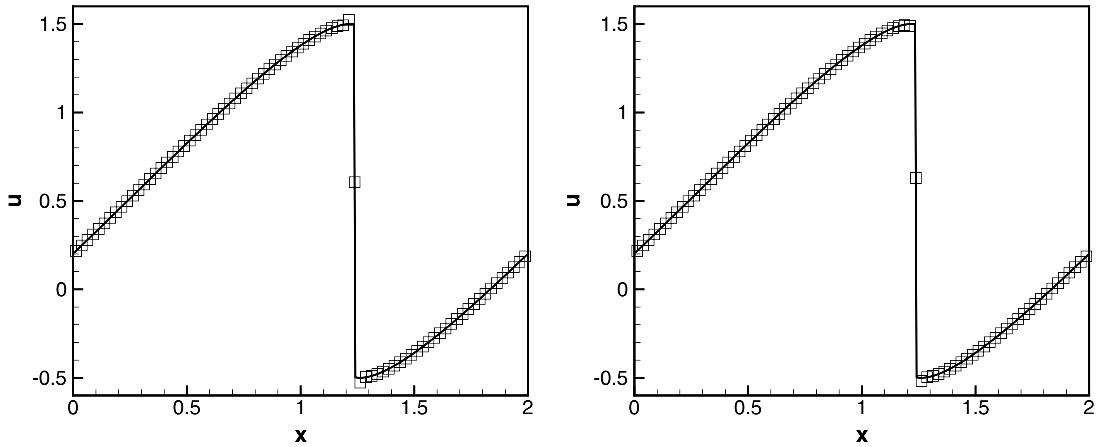

Fig. 3.1. Burgers equation. $u(x,0) = 0.5 + \sin(\pi x)$, $t = 1.5/\pi$, $N = 80$ points. Solid line: exact solution; squares: computed solution. Left: $k = 1$; right: $k = 2$.

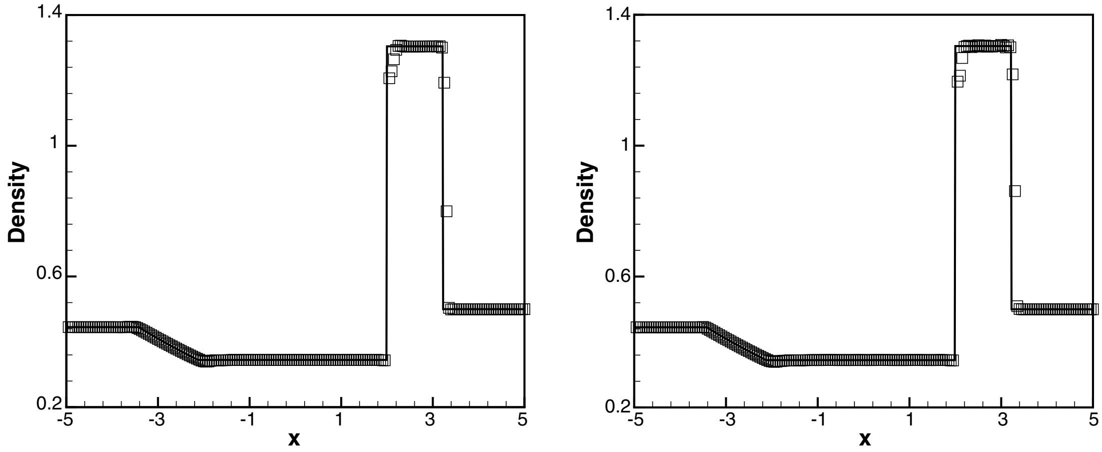

Fig. 3.2. Lax problem. $t = 1.3$, 200 cells. Density. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

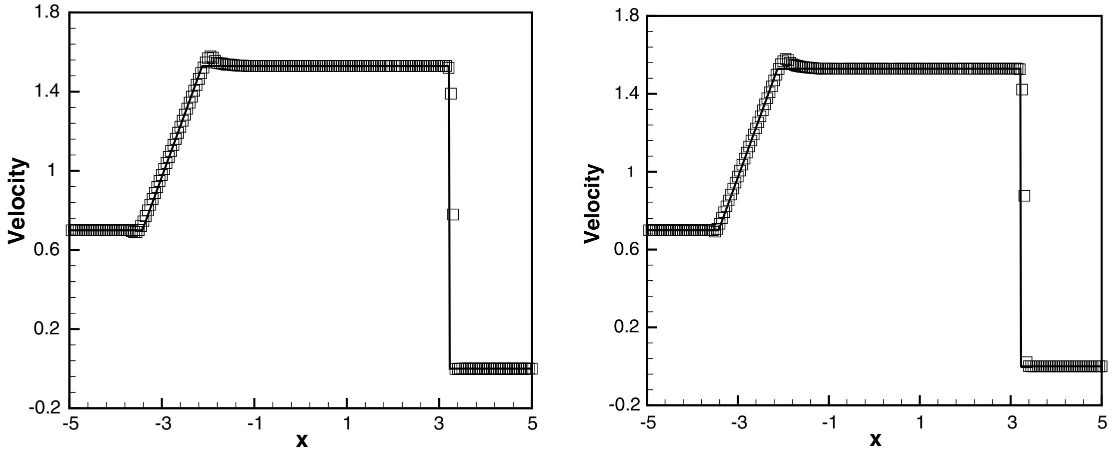

Fig. 3.3. Lax problem. $t = 1.3$, 200 cells. Velocity. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

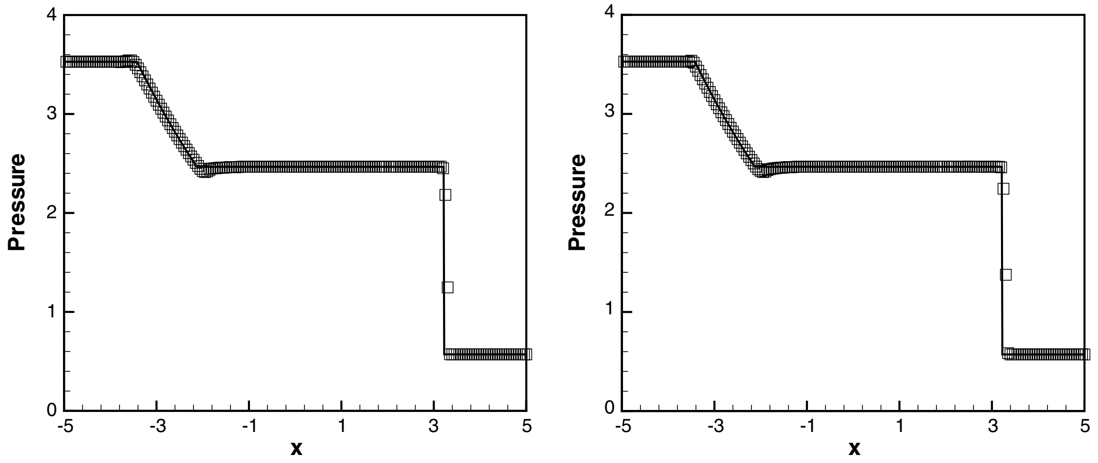

Fig. 3.4. Lax problem. $t = 1.3$, 200 cells. Pressure. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

3.2. Numerical tests for two-medium flows

In this section, we present the results of numerical experiments for various two-medium flows by the RKDG schemes described in Section 2. In the present numerical experiments, 200 cells are used. Units for density, velocity, pressure, length and time are kg/m^3, m/s, Pa, m and s, respectively.

Example 3.5. This is an air–helium shock tube problem taken from [13], with the initial conditions as

$$(\rho, v, p, \gamma) = (1, 0, 1 \times 10^5, 1.4) \quad \text{for } x \leqslant 0.5; \quad (\rho, v, p, \gamma) = (0.125, 0, 1 \times 10^4, 1.2) \quad \text{for } x > 0.5.$$

The computed density ρ, velocity v and pressure p by RKDG schemes are plotted at $t = 0.0007$ against the exact solution in Figs. 3.5–3.7. In this case, we choose the TVB limiter constant $M_1 = 0.1$, $M_2 = 100$ and $M_1 = 0.1$, $M_2 = 10{,}000$ for $k = 1$ and $k = 2$, respectively.

The location of the material interface is captured correctly by both the RKDG schemes with $k = 1$ and $k = 2$; the computed results are oscillatory free at the neighborhood of the interface by both the schemes and are very comparable to the analysis. We also can see that the resolution of computed results by RKDG schemes is better than that in [13].

Example 3.6. 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

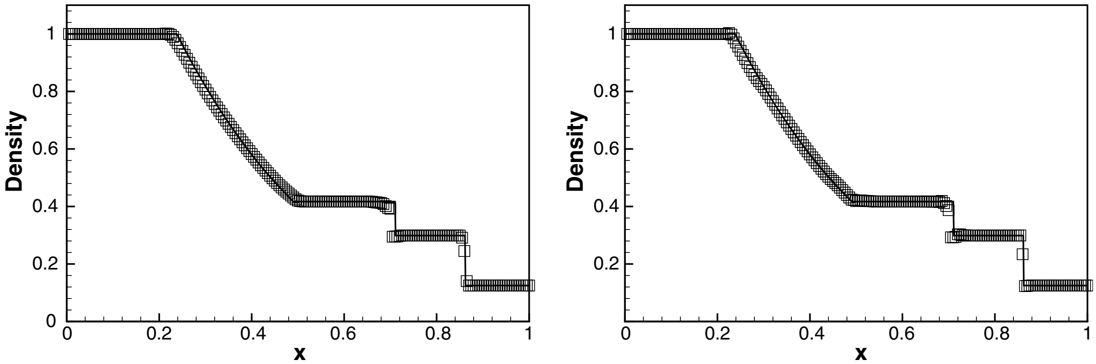

Fig. 3.5. Density for Example 3.5 by RKDG method. $t = 0.0007$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

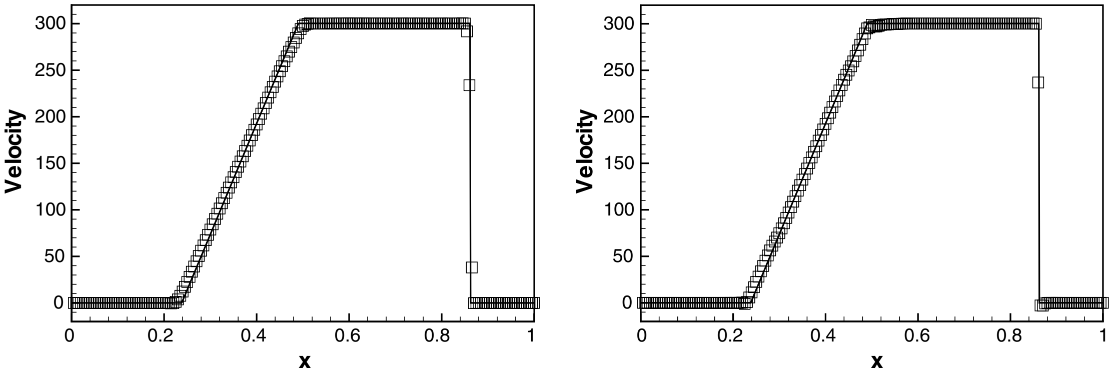

Fig. 3.6. Velocity for Example 3.5 by RKDG method. $t = 0.0007$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

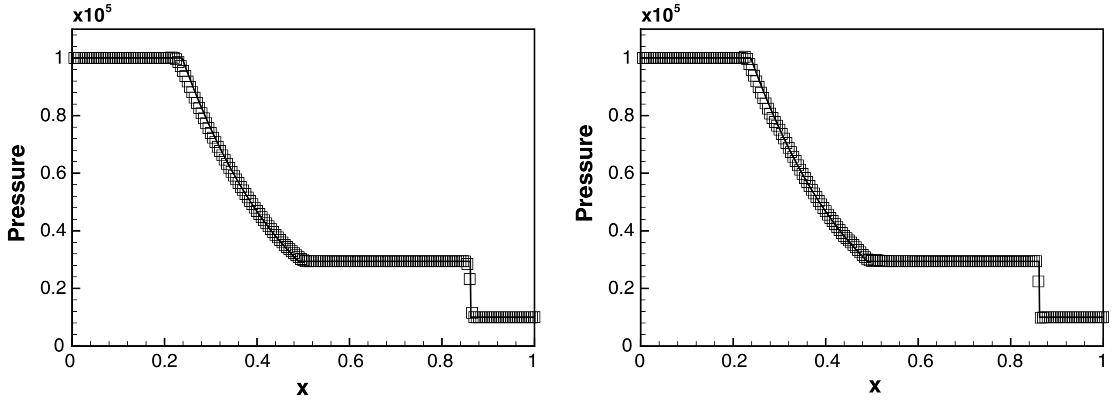

Fig. 3.7. Pressure for Example 3.5 by RKDG method. Solid line: the exact solution. Squares: numerical solution. $t = 0.0007$. Left: $k = 1$; right: $k = 2$.

$$(\rho, v, p, \gamma) = \begin{cases} (1.3333, 0.3535\sqrt{10^5}, 1.5 \times 10^5, 1.4) & \text{for } x \leqslant 0.05, \\ (1, 0, 1 \times 10^5, 1.4) & \text{for } 0.05 < x \leqslant 0.5, \\ (0.1379, 0, 1 \times 10^5, \frac{5}{3}) & \text{for } x > 0.5. \end{cases}$$

The strength of the shock is $p_L/p_R = 1.5$ and initially located at $x = 0.05$, with the initial state ahead of the shock as $(\rho, v, p) = (1, 0, 1 \times 10^5)$. 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 the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.0012$ against the exact solution in Figs. 3.8–3.10. In this case, we choose the TVB limiter constant $M_1 = 10$, $M_2 = 10$ and $M_1 = 10$, $M_2 = 30$ for $k = 1$ and $k = 2$, respectively.

We can also see that the material interface is located at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$. The computed results by both the schemes are very comparable to the analytical solution. The computed results are oscillatory free at the neighborhood of the interface for density. The resolution of computed results by the RKDG schemes is also much better than that in [13].

Example 3.7. We increase the strength of the right shock wave to $p_L/p_R = 15$ and the other initial conditions are unchanged as for Example 3.6. This example is also taken from [13]. We thus have the following initial conditions

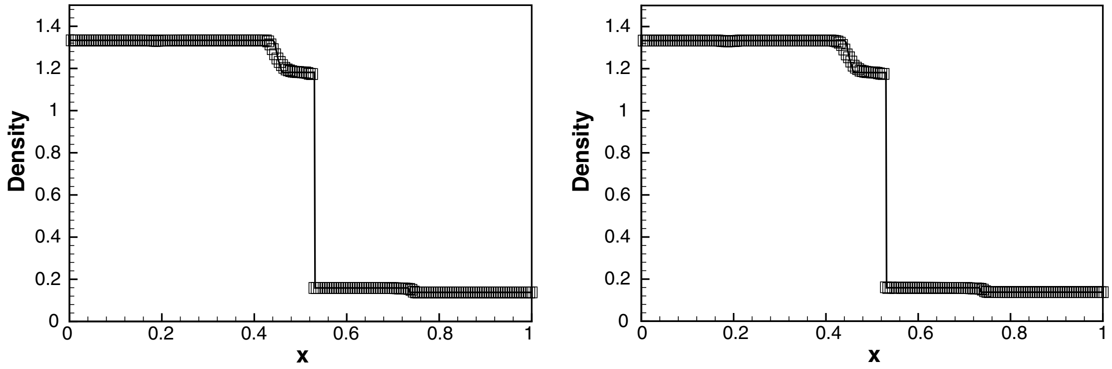

Fig. 3.8. Density for Example 3.6 by RKDG method. $t = 0.0012$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

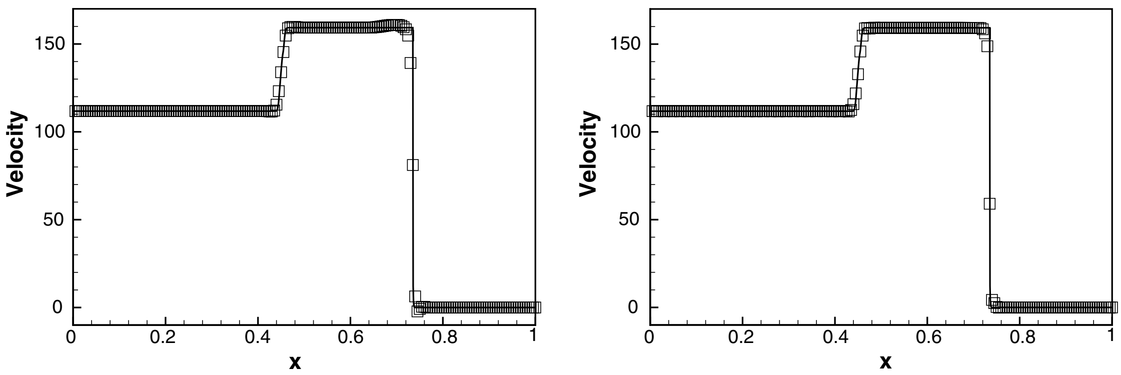

Fig. 3.9. Velocity for Example 3.6 by RKDG method. $t = 0.0012$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

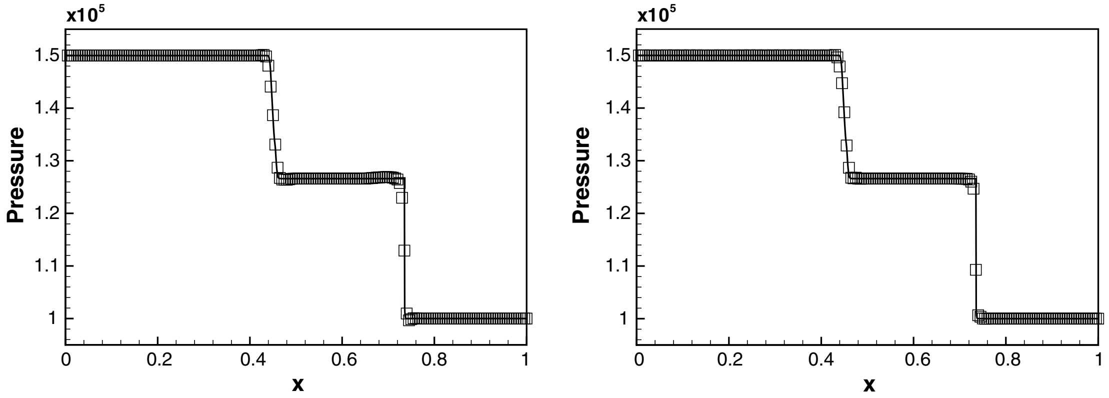

Fig. 3.10. Pressure for Example 3.6 by RKDG method. $t = 0.0012$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

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

The computed density ρ, velocity v and pressure p by both the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.0005$ against the exact solution in Figs. 3.11–3.13. In this case, we choose the TVB limiter constant $M_1 = 0.1$, $M_2 = 500$ and $M_1 = 0.1$, $M_2 = 1500$ for $k = 1$ and $k = 2$, respectively.

The interface is captured at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$. The computed results obtained are fairly comparable to the analysis. The computed results are oscillatory free at the neighborhood of the interface for density, but the density before the interface indicates some small degree of smearing for both $k = 1$ and $k = 2$ cases. The resolution of computed results by RKDG schemes is, however, still slightly better than that shown in [13].

Example 3.8. We consider the Euler equation (2.1) with the following Riemann initial conditions

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

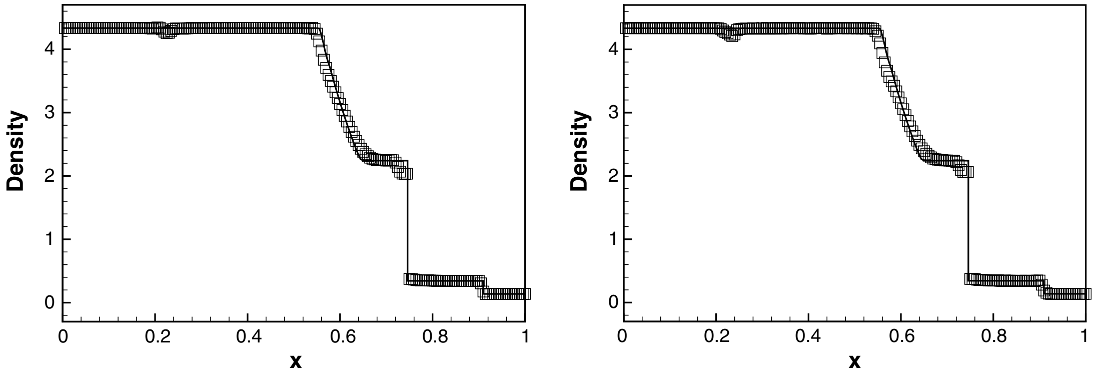

Fig. 3.11. Density for Example 3.7 by RKDG method. $t = 0.0005$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

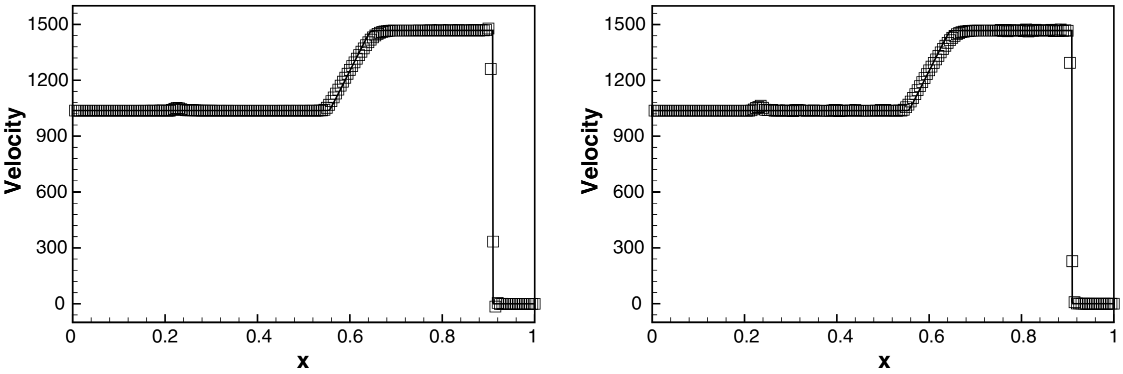

Fig. 3.12. Velocity for Example 3.7 by RKDG method. $t = 0.0005$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

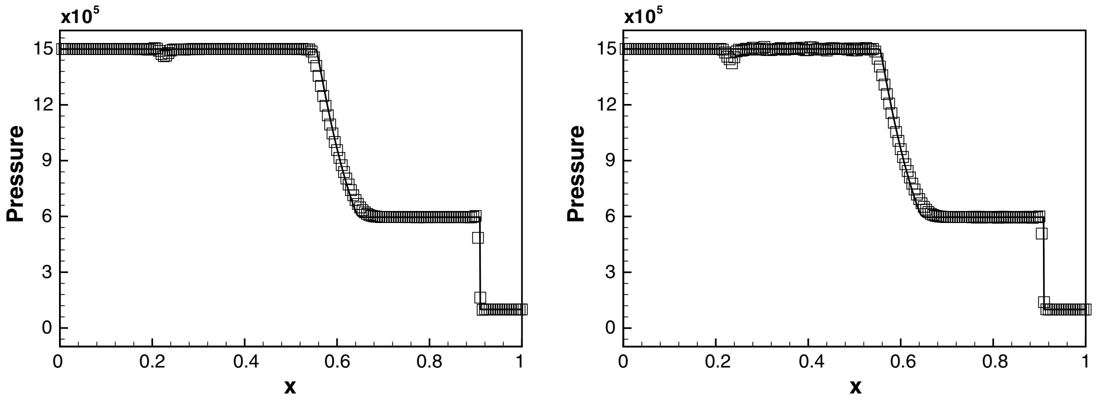

Fig. 3.13. Pressure for Example 3.7 by RKDG method. $t = 0.0005$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

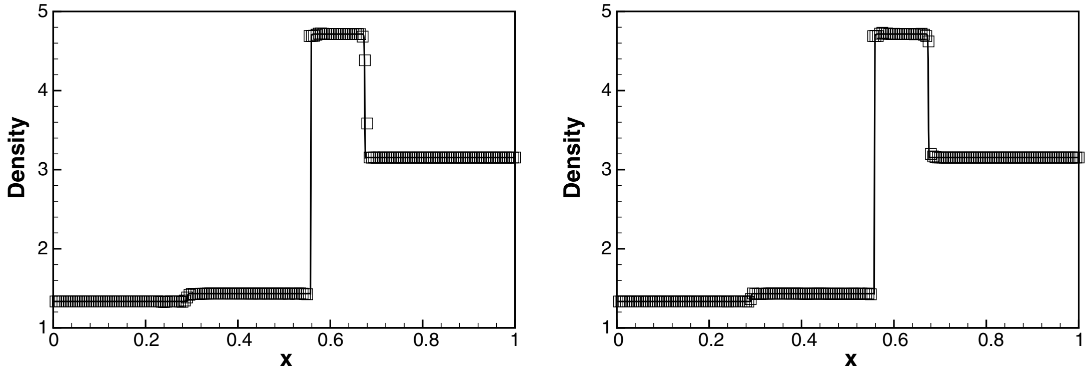

Fig. 3.14. Density for Example 3.8 by RKDG method. $t = 0.0017$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

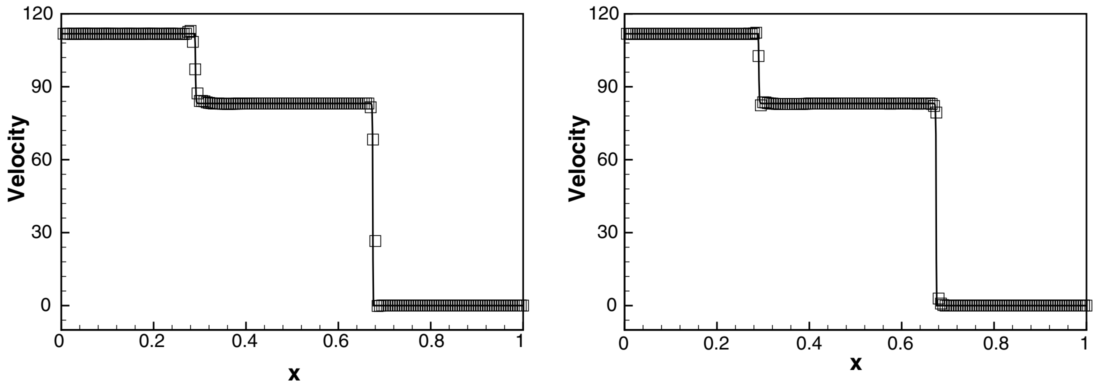

Fig. 3.15. Velocity for Example 3.8 by RKDG method. $t = 0.0017$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

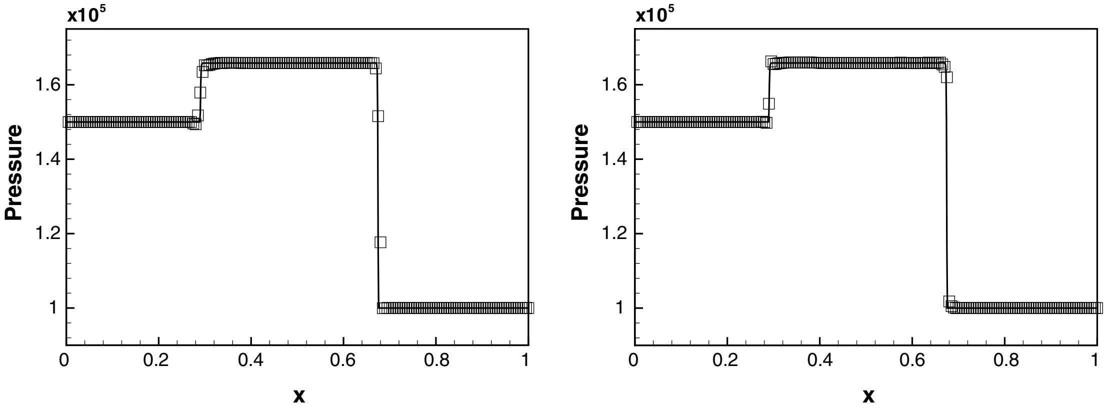

Fig. 3.16. Pressure for Example 3.8 by RKDG method. $t = 0.0017$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

This example is similar to Example 3.6, except that the initial status of the fluid on the right of the interface has been changed. The computed density ρ, velocity v and pressure p by both the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.0017$ against the exact solution in Figs. 3.14–3.16. In this case, we choose the TVB limiter constant $M_1 = 0.1$, $M_2 = 1000$ and $M_1 = 0.1$, $M_2 = 60{,}000$ for $k = 1$ and $k = 2$, respectively.

We can also see that the interface of the two-medium is located at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$. The computed results by both the schemes are comparable to the analysis. The computed results are oscillatory free at the neighborhood of the interface for density for both $k = 1$ and $k = 2$ cases. We also can see that the resolution of computed results by RKDG schemes is much better than that shown in [13].

Example 3.9. We increase the strength of right shock wave to $p_L/p_R = 15$, while the other initial conditions remain unchanged as for Example 3.8. We have the Riemann problems with the following initial conditions

$$(\rho, v, p, \gamma) = \begin{cases} (4.3333, 3.2817\sqrt{10^5}, 1.5 \times 10^6, 1.4) & \text{for } x \leqslant 0.05, \\ (1, 0, 1 \times 10^5, 1.4) & \text{for } 0.05 < x \leqslant 0.5, \\ (3.1538, 0, 1 \times 10^5, 1.249) & \text{for } x > 0.5. \end{cases}$$

The computed density ρ, velocity v and pressure p by both the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.0007$ against the exact solution in Figs. 3.17–3.19. In this case, we choose the TVB limiter constant $M_1 = 0.5$, $M_2 = 1000$ and $M_1 = 1.0$, $M_2 = 3000$ for $k = 1$ and $k = 2$, respectively.

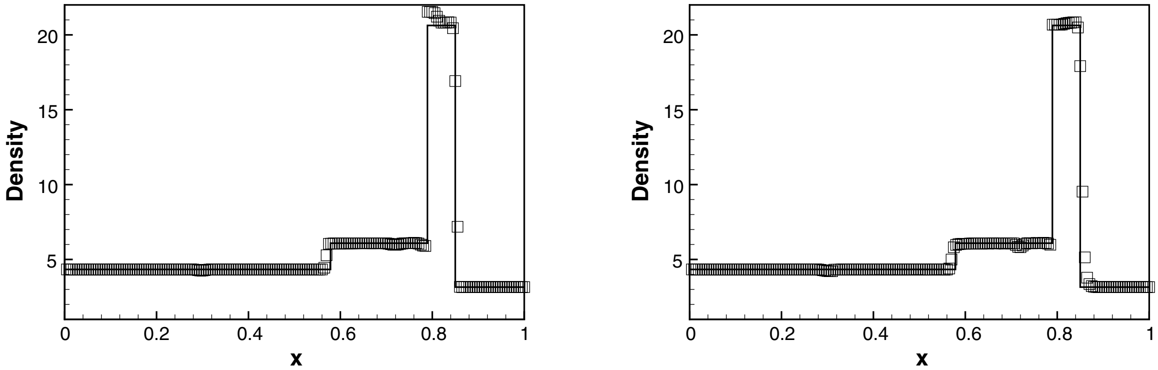

Fig. 3.17. Density for Example 3.9 by RKDG method. $t = 0.0007$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

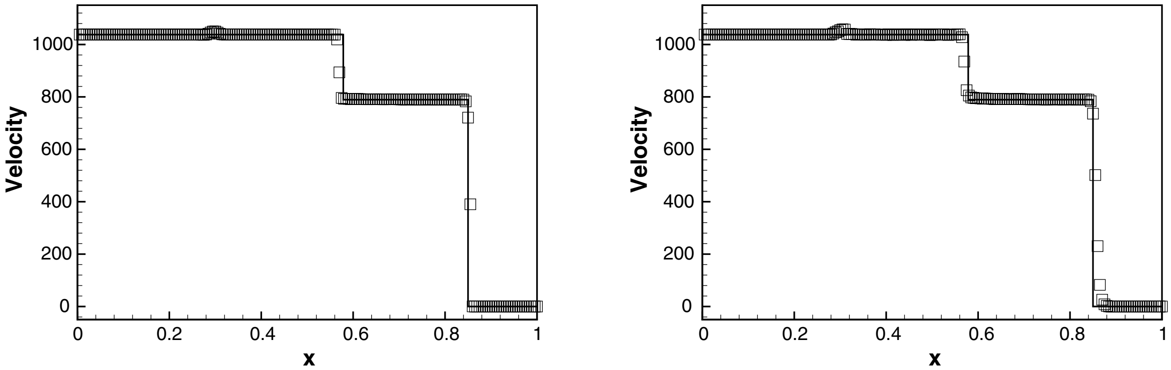

Fig. 3.18. Velocity for Example 3.9 by RKDG method. $t = 0.0007$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

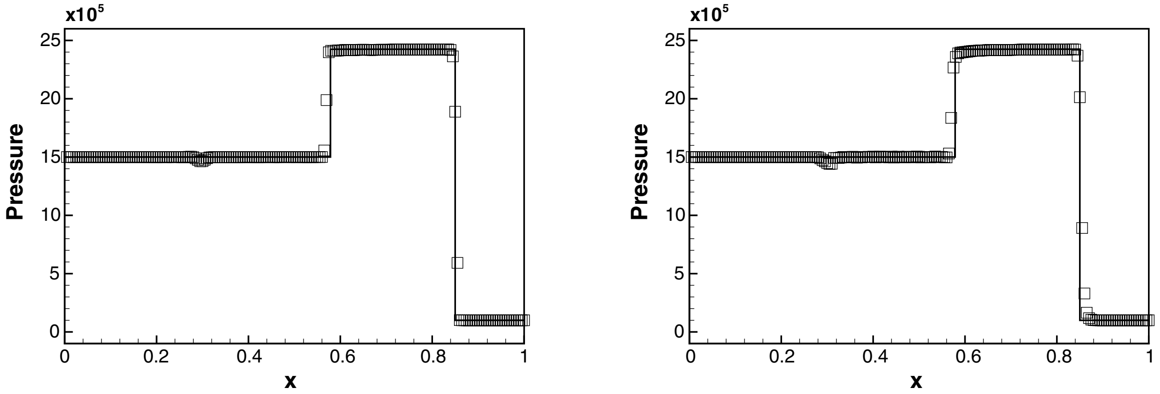

Fig. 3.19. Pressure for Example 3.9 by RKDG method. $t = 0.0007$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

We can also see that the material interface is located at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$, and the computed results by both the schemes are comparable to the analysis. The computed result depicts some overshoot behind the interface for the density for $k = 1$ case, and is oscillatory free at the neighborhood of the interface for the density for $k = 2$ case. But the resolution of shock computed by RKDG scheme with $k = 1$ is a little better than that by $k = 2$.

Example 3.10. This is a gas–water shock tube problem with very high pressure in the gaseous medium. The initial condition are

$$(\rho, v, p, \gamma) = \begin{cases} (1270, 0, 8 \times 10^8, 1.4) & \text{for } x \leqslant 0.5, \\ (1000, 0, 1 \times 10^5, 7.15) & \text{for } x > 0.5. \end{cases}$$

In this 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 both the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.00016$ against the exact solution in Figs. 3.20–3.22. In this case, we choose the TVB limiter constant $M_1 = 0.4$ and $M_2 = 10$ for both $k = 1$ and $k = 2$.

It is observed that the computed gas–water interface is located at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$, and the computed results by both the schemes are comparable to the analysis.

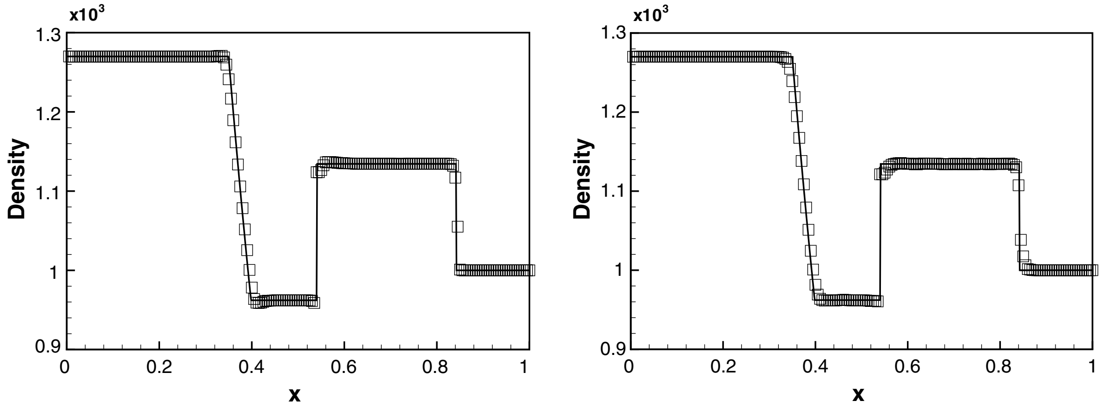

Fig. 3.20. Density for Example 3.10 by RKDG method. $t = 0.00016$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

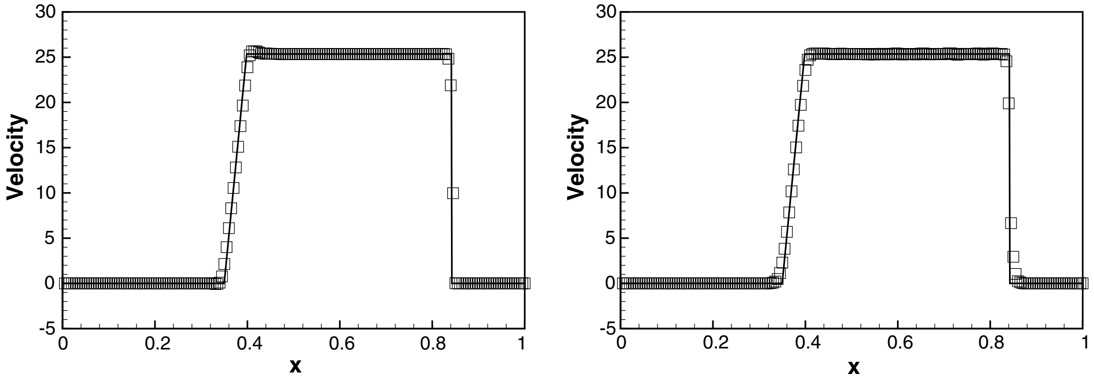

Fig. 3.21. Velocity for Example 3.10 by RKDG method. $t = 0.00016$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

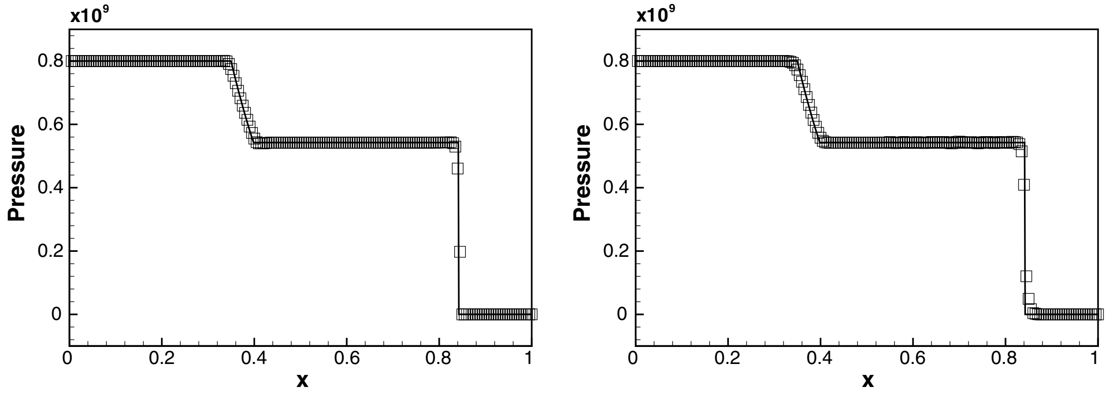

Fig. 3.22. Pressure for Example 3.10 by RKDG method. $t = 0.00016$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

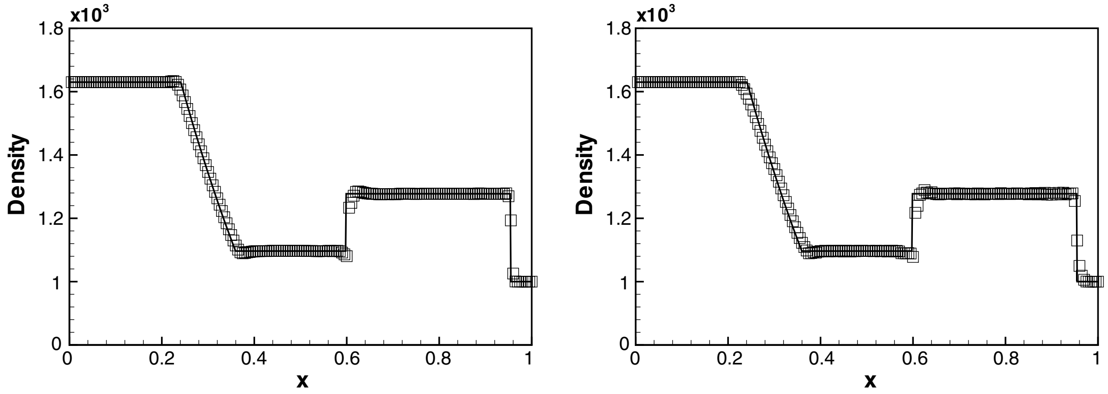

Fig. 3.23. Density for Example 3.11 by RKDG method. $t = 0.0001$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

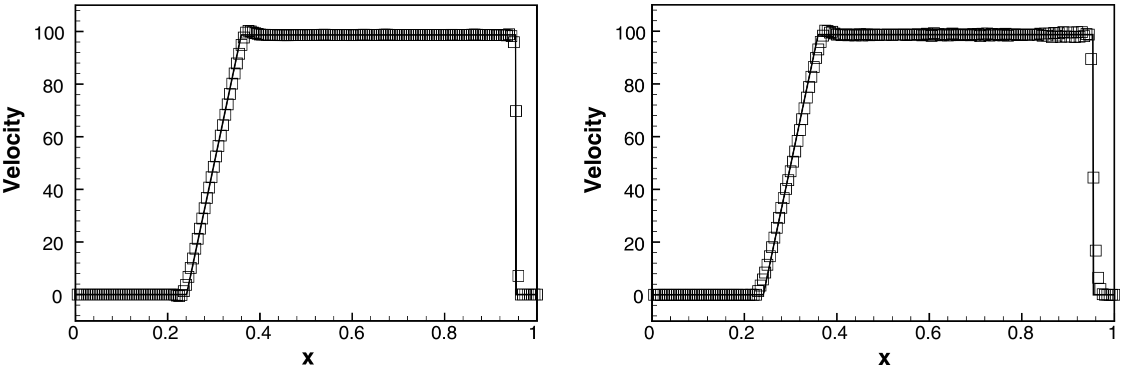

Fig. 3.24. Velocity for Example 3.11 by RKDG method. $t = 0.0001$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

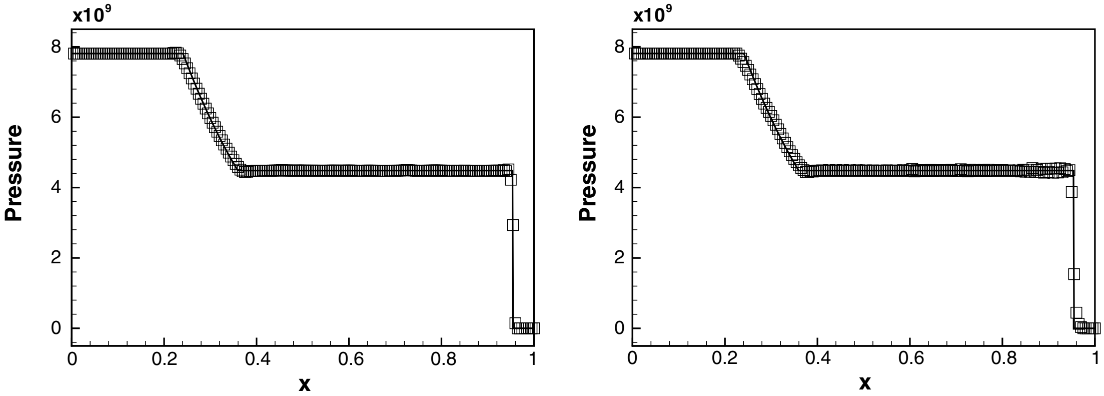

Fig. 3.25. Pressure for Example 3.11 by RKDG method. $t = 0.0001$. Solid line: the exact solution. Squares: numerical solution. Left: $k = 1$; right: $k = 2$.

There are a little but well limited oscillations in the region of the interface for the density for both the $k = 1$ and $k = 2$ cases.

Example 3.11. We increase the energy of the explosive gaseous medium in Example 3.10. The initial condition becomes

$$(\rho, v, p, \gamma) = \begin{cases} (1630, 0, 7.81 \times 10^9, 1.4) & \text{for } x \leqslant 0.5, \\ (1000, 0, 1 \times 10^5, 7.15) & \text{for } x > 0.5. \end{cases}$$

The computed density ρ, velocity v and pressure p by both the RKDG schemes with $k = 1$ and $k = 2$ are plotted at $t = 0.0001$ against the exact solution in Figs. 3.23–3.25. In this case, we choose the TVB limiter constant $M_1 = 0.4$, $M_2 = 300$ and $M_1 = 0.4$, $M_2 = 900$ for $k = 1$ and $k = 2$, respectively.

We can also observe that the computed gas–water interface is located at the correct cell by both the RKDG schemes with $k = 1$ and $k = 2$, and the computed results by both the schemes are comparable to the analysis. There are exhibited a little but still well limited oscillations in the region of the interface for the density for both the $k = 1$ and $k = 2$ cases.

4. Concluding remarks

We have investigated extensively using the discontinuous Galerkin (DG) finite element methods for two-medium flow simulations in one-dimension, having incorporated a conservative treatment of the moving material interface. Numerical results for both gas–gas and gas–water flow in one-dimension are provided to show the characteristic behavior and applicability of this procedure to a wide range of conditions. The proposed method has been found to be able to provide the correct interface location and still reasonable solution with well-limited oscillations in the interface region. Further research on the efficient implementation of these methods for multi-dimensional problems is ongoing.

Acknowledgements

The authors are indebted to Professor Chi-Wang Shu for his help. The authors also wish to thank the unnamed reviewers for their careful review works and helpful suggestions which led to an improved paper.

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, S. Karni, Computations of compressible multifluids, J. Comput. Phys. 169 (2001) 594–623.

[3] D.A. Bailey, P.K. Sweby, 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, 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, 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, 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, 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, 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, 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, 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, C.-W. Shu, The Runge–Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, Math. Model. Numer. Anal. (M^2AN) 25 (1991) 337–361.

[12] B. Cockburn, 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, 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, N. Zhao, Conservative front tracking and level set algorithms, Proc. Natl. Acad. Sci. 98 (2001) 14198–14201.

[16] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.

[17] S. Karni, Multi-component flow calculations by a consistent primitive algorithm, J. Comput. Phys. 112 (1994) 31–43.

[18] B. Koren, M.R. Lewis, E.H. van Brummelen, B. van Leer, Riemann-problem and level-set approaches for homentropic two-fluid flow computations, J. Comput. Phys. 181 (2002) 654–674.

[19] B. Larrouturou, How to preserve the mass fractions positivity when computing compressible multi-component flow, J. Comput. Phys. 95 (1991) 31–43.

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

[21] T.G. Liu, B.C. Khoo, K.S. Yeo, The simulation of compressible multi-medium flow. Part I: a new methodology with applications to1D gas–gas and gas–water cases, Comput. Fluids 30 (2001) 291–314.

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

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

[24] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463–502.

[25] S. Osher, A. Sethian, Fronts prohabiliting with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulation, J. Comput. Phys. 79 (1988) 12–49.

[26] J. Qiu, C.-W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput. 26 (2005) 907–929.

[27] J. Qiu, 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.

[28] W.H. Reed, T.R. Hill, Triangular mesh methods for neutron transport equation, Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[29] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comput. 49 (1987) 105–121.

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