The discontinuous Petrov–Galerkin method for one-dimensional compressible Euler equations in the Lagrangian coordinate*

Zhao Guo-Zhong(赵国忠)[a)†], Yu Xi-Jun(蔚喜军)[b)], and Guo Peng-Yun(郭鹏云)[a)]

[a)] *Faculty of Mathematics, Baotou Teachers' College, Baotou* 014030, *China*

[b)] *Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing* 100088, *China*

(Received 13 September 2012; revised manuscript received 18 November 2012)

In this paper, a Petrov–Galerkin scheme named the Runge–Kutta control volume (RKCV) discontinuous finite element method is constructed to solve the one-dimensional compressible Euler equations in the Lagrangian coordinate. Its advantages include preservation of the local conservation and a high resolution. Compared with the Runge–Kutta discontinuous Galerkin (RKDG) method, the RKCV method is easier to implement. Moreover, the advantages of the RKCV and the Lagrangian methods are combined in the new method. Several numerical examples are given to illustrate the accuracy and the reliability of the algorithm.

Keywords: compressible Euler equations, Runge–Kutta control volume discontinuous finite element method, Lagrangian coordinate

PACS: 02.70.Dh, 02.60.Lj **DOI:** 10.1088/1674-1056/22/5/050206

1. Introduction

In the past several decades, the study of analytical and numerical methods for solving nonlinear partial differential equations has enjoyed a period of intense activity. Many efficient numerical methods have been developed, such as the finite-element method,[1] the finite-volume method,[2] and the boundary-element method.[3] Even though those methods are very effective for solving various kinds of partial differential equations, they have their own limits. For example, both fixed and moving mesh finite-element methods can not solve many engineering problems, such as large deformation and crack growth, even with the re-meshing and re-mapping techniques. To avoid those problems, in recent years, meshless techniques have attracted much attention. In a meshless method, a set of scattered nodes is used instead of meshing the domain of the problem. The meshless method is a new and interesting numerical technique which can solve many engineering problems with a minimum of meshing or no meshing at all.[4–17] Some meshless methods have been developed, such as the element-free Galerkin method,[4–10] the meshless reproducing kernel particle method,[11–15] and the complex variable element-free Galerkin method.[16,17] The above-mentioned meshless methods have been used for finding the numerical solutions to the equal-width wave equation, the two-dimensional potential problem, the steady-state heat conduction problem, and the regularized long wave equations. Furthermore, the above-mentioned methods have also been used intensively in numerical simulations of fluid flow problems.

In the numerical simulations of a fluid flow, there are two basic viewpoints. The first one is Lagrangian, in which the mesh moves with the local fluid velocity. The second one is Eulerian, in which the fluid flows through a grid fixed in space. Each framework has its own advantages and disadvantages. In the past several years, many Lagrangian type schemes have been constructed.[18–26] Hirt *et al.* developed a classical numerical method called the arbitrary Lagrangian–Eulerian (ALE) method,[27] which combines the best properties of the Eulerian and the Lagrangian methods. In general, the ALE method consists of three phases, a lagrangian phase, a rezoning phase, and a re-mapping phase. In this paper, we will develop a new high-order and high-resolution numerical method for compressible Euler equations in the Lagrangian coordinate.

The discontinuous Galerkin (DG) method was proposed in 1973 by Reed and Hill for the numerical simulation of the neutron transport equation.[28] A major development of the DG method was accomplished by Cockburn and Shu in a series of papers.[29–33] The scheme is termed the RKDG method. The method has some attractive properties.[34,35] Firstly, it can be designed for any order of accuracy in space and time. Secondly, it can handle complicated geometry and boundary conditions easily. Finally, it is extremely local in data communications and flexible to adapt. The majority of the previous studies are based on the Eulerian framework.[36–40]

Based on the generalized difference method[41] or the control volume finite-element method[42,43] and the RKDG fi-

*Project supported by the National Natural Science Foundation of China (Grant Nos. 11261035 and 11171038), the Science Research Foundation of the Institute of Higher Education of Inner Mongolia Autonomous Region, China (Grant No. NJZZ12198), and the Natural Science Foundation of Inner Mongolia Autonomous Region, China (Grant No. 2012MS0102).

†Corresponding author. E-mail: zhaoguozhongbttc@sina.com

© 2013 Chinese Physical Society and IOP Publishing Ltd

nite element method, Chen and Yu proposed a new high-order and high-resolution method called the Runge–Kutta control volume (RKCV) discontinuous finite-element method. The method combines most advantages of the methods we just mentioned. The RKCV method can easily handle a discontinuous solution and complicated boundary conditions. It has the property of preserving the local conservation, and it is simpler than the RKDG method. It has been used to solve one- and two-dimensional hyperbolic conservation laws in the Eulerian coordinate.[44,45]

Due to the fixed mesh on the Lagrangian coordinate, some modern high-order Eulerian methods can be used to discretize the conservation laws written in the Lagrangian coordinate. We have constructed the RKDG finite-element method for solving one- and two-dimensional compressible Euler equations in the Lagrangian coordinate.[23,24] The algorithm behaved very well, even for multi-medium fluid simulations. The cell centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics was accomplished by François Vilar.[46] Jia and Zhang developed a high-order discontinuous Galerkin spectral finite-element method for Lagrangian gas dynamics in two dimensions.[47] In this paper, the RKCV method for one-dimensional in-viscid compressible Euler equations in the Lagrangian coordinate is developed. The main motivation of this work is to combine the advantages of the Lagrangian and the RKCV methods to construct an applicable algorithm for gas dynamic simulations. Some numerical examples are given to verify the reliability and the efficiency of the algorithm. The paper is arranged as follows. In Section 2, the RKCV method in the Lagrangian coordinate will be described in detail for one-dimensional gas dynamic equations. Some numerical examples will be given in Section 3. In the last section, we will give the conclusion.

2. RKCV method in Lagrangian coordinate

The one-dimensional compressible Euler equations for an in-viscid compressible fluid in the Eulerian coordinate are[20,26]

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x = 0, \tag{1}$$

$$\boldsymbol{u}(x,0) = \boldsymbol{u}_0(x), \tag{2}$$

with

$$\boldsymbol{u} = \begin{pmatrix} \rho \\ \rho v \\ \rho E \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix}, \tag{3}$$

where ρ is the density, v is the velocity, E is the specific total energy, and p is the pressure. They are related by $E = p/[\rho(\gamma-1)] + v^2/2$.

By introducing the Lagrangian coordinate ξ given by $\xi = \int_{x_0(t)}^{x} \rho(z,t)\,\mathrm{d}z$, where $x_0(t)$ denotes the Eulerian coordinate of a given fluid particle, Eqs. (1) and (2) in the Lagrangian coordinate can be written as[20,26]

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_\xi = 0, \tag{4}$$

$$\boldsymbol{u}(\xi,0) = \boldsymbol{u}_0(\xi), \tag{5}$$

with

$$\boldsymbol{u} = \begin{pmatrix} \tau \\ v \\ E \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} -v \\ p \\ pv \end{pmatrix}, \tag{6}$$

where $\tau = 1/\rho$ is the specific volume. The relation between the Lagrangian and the Eulerian coordinates is

$$\frac{\mathrm{d}x(\xi,t)}{\mathrm{d}t} = v(\xi,t). \tag{7}$$

2.1. Spacial discretization

The computational domain $[a,b]$ is divided into N cells with boundary points $a = \xi_{\frac{1}{2}} < \xi_{\frac{3}{2}} < \cdots < \xi_{N+\frac{1}{2}} = b$. The cells are denoted by $I_j = [\xi_{j-\frac{1}{2}}, \xi_{j+\frac{1}{2}}]$, $1 \le j \le N$. Assume

$$\xi_j = (\xi_{j-1/2} + \xi_{j+1/2})/2,$$

$$\Delta\xi_j = \xi_{j+1/2} - \xi_{j-1/2}, \quad f^+_{j\pm\frac{1}{2}} = \lim_{\xi \to \xi^+_{j\pm\frac{1}{2}}} f(\xi),$$

$$f^-_{j\pm\frac{1}{2}} = \lim_{\xi \to \xi^-_{j\pm\frac{1}{2}}} f(\xi),\ 1 \le j \le N; \quad h = \max_{1\le j\le N} \Delta\xi_j.$$

The trial function space is defined as piecewise polynomials

$$V_h = V_h^k = \{v \in L^1(a,b) : v|_{I_j} \in P^k(I_j), j = 1,\ldots,N\}, \tag{8}$$

where $P^k(I_j)$ denotes a set of polynomials of degree up to k defined on cell I_j. In the following calculation, we let $k = 1$. The basis functions of V_h in each element I_j are chosen as

$$\varphi_j^{(0)} = 1, \quad \varphi_j^{(1)} = \xi - \xi_j.$$

Thus, the numerical solutions in each cell I_j can be expressed as

$$\boldsymbol{u}_h(\xi,t) = \sum_{l=0}^{1} \boldsymbol{u}_j^l(t)\varphi_j^{(l)}(\xi), \tag{9}$$

where the degrees of freedom $\boldsymbol{u}_j^0$ and $\boldsymbol{u}_j^1$ are the cell average values and the first-order derivatives, respectively.

The test function space W_h based on control volumes $I^{*+}_{j-1/2} = (\xi_{j-1/2}, \xi_j)$ and $I^{*-}_{j+1/2} = (\xi_j, \xi_{j+1/2})$ is a piecewise constant function space. The corresponding basis functions of W_h are

$$\psi_{j-1/2} = \begin{cases} 1, & \forall\xi \in I^{*+}_{j-1/2}, \\ 0, & \text{otherwise}, \end{cases} \quad \psi_{j+1/2} = \begin{cases} 1, & \forall\xi \in I^{*-}_{j+1/2}, \\ 0, & \text{otherwise}. \end{cases}$$

Multiplying Eq. (4) by test function $w_h \in W_h$ and integrating over I_j, we get

$$\int_{I_j} \boldsymbol{u}_t w_h\,\mathrm{d}\xi + \int_{I_j} \boldsymbol{f}(\boldsymbol{u})_\xi w_h\,\mathrm{d}\xi = 0. \tag{10}$$

After the test function being replaced by the basis function of W_h, $\boldsymbol{u}$ by $\boldsymbol{u}_h \in V_h$, the following formulas are obtained:

$$\int_{I^{*+}_{j-1/2}} (\boldsymbol{u}_h)_t \,\mathrm{d}\xi + \int_{I^{*+}_{j-1/2}} \boldsymbol{f}(\boldsymbol{u})_\xi \,\mathrm{d}\xi = 0, \tag{11}$$

$$\int_{I^{*-}_{j+1/2}} (\boldsymbol{u}_h)_t \,\mathrm{d}\xi + \int_{I^{*-}_{j+1/2}} \boldsymbol{f}(\boldsymbol{u})_\xi \,\mathrm{d}\xi = 0. \tag{12}$$

Using a simple formal integration by parts, we obtain the following formulas:

$$\int_{I^{*+}_{j-1/2}} (\boldsymbol{u}_h)_t \,\mathrm{d}\xi + \boldsymbol{f}_j - \hat{\boldsymbol{f}}_{j-1/2} = 0, \tag{13}$$

$$\int_{I^{*-}_{j+1/2}} (\boldsymbol{u}_h)_t \,\mathrm{d}\xi + \hat{\boldsymbol{f}}_{j+1/2} - \boldsymbol{f}_j = 0. \tag{14}$$

Here, $\hat{\boldsymbol{f}}_{j\pm1/2}$ are the numerical fluxes, which are single-valued functions defined at the cell interfaces and depend on the numerical solution $\boldsymbol{u}_h$ at both sides of the interface

$$\hat{\boldsymbol{f}}_{j\pm1/2} = \hat{\boldsymbol{f}}(\boldsymbol{u}_h(\xi^-_{j\pm\frac{1}{2}}, t), \boldsymbol{u}_h(\xi^+_{j\pm\frac{1}{2}}, t)).$$

In this paper, the numerical flux is taken as the Lax–Friedrichs (LF) flux

$$\begin{aligned}\hat{\boldsymbol{f}}^{\mathrm{LF}}_{j\pm1/2} = {} & \frac{1}{2}[\boldsymbol{f}(\boldsymbol{u}_h(\xi^-_{j\pm\frac{1}{2}})) + \boldsymbol{f}(\boldsymbol{u}_h(\xi^+_{j\pm\frac{1}{2}}, t)) \\ & - \alpha_{j\pm1/2}(\boldsymbol{u}_h(\xi^+_{j\pm\frac{1}{2}}, t) - \boldsymbol{u}_h(\xi^-_{j\pm\frac{1}{2}}, t))],\end{aligned} \tag{15}$$

with

$$\alpha_{j\pm1/2} \equiv \alpha = \max_j\{\rho^{\pm}_{j\pm1/2} c^{\pm}_{j\pm1/2}\},$$

where c represents the sound velocity and satisfies $c = \sqrt{\gamma p/\rho}$. By using numerical solution u_h (Eq. (9)), equations (13) and (14) can be rewritten as

$$\begin{aligned}&\int_{I^{*+}_{j-1/2}} \varphi^{(0)}_j \,\mathrm{d}\xi \frac{\mathrm{d}\boldsymbol{u}^0_j}{\mathrm{d}t} \\ &+ \int_{I^{*+}_{j-1/2}} \varphi^{(1)}_j \,\mathrm{d}\xi \frac{\mathrm{d}\boldsymbol{u}^1_j}{\mathrm{d}t} + \boldsymbol{f}_j - \hat{\boldsymbol{f}}_{j-1/2} = 0, \\ &\int_{I^{*-}_{j+1/2}} \varphi^{(0)}_j \,\mathrm{d}\xi \frac{\mathrm{d}\boldsymbol{u}^0_j}{\mathrm{d}t}\end{aligned} \tag{16}$$

$$+ \int_{I^{*-}_{j+1/2}} \varphi^{(1)}_j \,\mathrm{d}\xi \frac{\mathrm{d}\boldsymbol{u}^1_j}{\mathrm{d}t} + \hat{\boldsymbol{f}}_{j+1/2} - \boldsymbol{f}_j = 0. \tag{17}$$

After the spacial discretization, the ordinary differential equations (ODEs) about the undetermined coefficients $\boldsymbol{u}^l_j(t)$ $(l = 0, 1)$ are obtained

$$\frac{\mathrm{d}\boldsymbol{u}_h}{\mathrm{d}t} = L_h(\boldsymbol{u}_h), \quad \forall t \in [0, T]. \tag{18}$$

2.2. Temporal discretization

Let δt denote the time step. The above ODEs are solved by the total variation diminishing (TVD) Runge–Kutta method.[48] The second-order TVD Runge–Kutta formula is

$$\begin{aligned}&\boldsymbol{u}^{(1)} = \boldsymbol{u}^n + \delta t L_h(\boldsymbol{u}^n, t^n), \\ &\boldsymbol{u}^{n+1} = 0.5\boldsymbol{u}^n + 0.5\boldsymbol{u}^{(1)} + 0.5\delta t L_h(\boldsymbol{u}^{(1)}, t^n + \delta t).\end{aligned}$$

The numerical simulations are accomplished by using the linear element $(k = 1)$ combining with the second-order TVD Runge–Kutta method.

2.3. Slop limiting

In order to maintain the stability and the non-oscillating property of the RKCV method for solving the conservation laws with strong shocks, a limiter must be applied. The limiter adopted is described below in detail. We only give the limiter for the internal cells. The limiting process of the boundary cells can be implemented in a similar way. We define

$$\boldsymbol{u}^-_{j+\frac{1}{2}} = \boldsymbol{u}^0_j + \boldsymbol{u}^1_j \frac{\Delta\xi_j}{2}, \quad \boldsymbol{u}^+_{j-\frac{1}{2}} = \boldsymbol{u}^0_j - \boldsymbol{u}^1_j \frac{\Delta\xi_j}{2}.$$

Then the second degree of freedom $\boldsymbol{u}^1_j$ is modified by either the standard total variation diminishing minmod limiter

$$(\boldsymbol{u}^1_j)^{\mathrm{mod}} = m\left(\boldsymbol{u}^1_j, \frac{\boldsymbol{u}^0_{j+1} - \boldsymbol{u}^0_j}{(\Delta\xi_j + \Delta\xi_{j+1})/4}, \frac{\boldsymbol{u}^0_j - \boldsymbol{u}^0_{j-1}}{(\Delta\xi_j + \Delta\xi_{j-1})/4}\right), \tag{19}$$

where m is given by

$$m(a_1, a_2, \ldots, a_n) = \begin{cases} s\min_{1\le j\le n}|a_j|, & \text{if } \operatorname{sign}(a_1) = \operatorname{sign}(a_2) = \cdots = \operatorname{sign}(a_n) = s, \\ 0, & \text{otherwise,} \end{cases} \tag{20}$$

or the TVB modified minmod function

$$\tilde{m}(a_1, a_2, \ldots, a_n) = \begin{cases} a_1, & \text{if } |a_1| \le Mh^2, \\ m(a_1, a_2, \ldots, a_n), & \text{otherwise,} \end{cases} \tag{21}$$

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

After limiting, we obtain

$$\boldsymbol{u}^{-\mathrm{mod}}_{j+\frac{1}{2}} = \boldsymbol{u}^0_j + (\boldsymbol{u}^1_j)^{\mathrm{mod}} \frac{\Delta\xi_j}{2},$$

$$\boldsymbol{u}^{+\mathrm{mod}}_{j-\frac{1}{2}} = \boldsymbol{u}^0_j - (\boldsymbol{u}^1_j)^{\mathrm{mod}} \frac{\Delta\xi_j}{2}. \tag{22}$$

In this paper, a limiter based on the characteristic field decomposition is used in each time step. The Jacobin matrix of Eqs. (4)–(6) is

$$\boldsymbol{A} = \begin{pmatrix} 0 & -1 & 0 \\ -p/\tau & -(\gamma-1)v/\tau & (\gamma-1)/\tau \\ -pv/\tau & p - (\gamma-1)v^2/\tau & (\gamma-1)/\tau v \end{pmatrix}. \tag{23}$$

The matrix consisting of the right eigenvectors of the Jacobin matrix is

$$\boldsymbol{R}=\begin{pmatrix}1 & 1 & 1\\ \rho c & 0 & -\rho c\\ \rho cv-p & p/(\gamma-1) & -\rho cv-p\end{pmatrix}. \tag{24}$$

The matrix consisting of the left eigenvectors of the Jacobin matrix is

$$\boldsymbol{L}=\frac{\gamma-1}{2\gamma p}\begin{pmatrix}p/(\gamma-1) & v+\gamma p/(\gamma-1)\rho c & -1\\ 2p & -2v & 2\\ p/(\gamma-1) & v-\gamma p/(\gamma-1)\rho c & -1\end{pmatrix}. \tag{25}$$

The eigenvalues are $\lambda_1=-\rho c$, $\lambda_2=0$, and $\lambda_3=\rho c$. We give the following three steps for limiting vector $\boldsymbol{u}_j^1$. Firstly, transform the three vectors $\boldsymbol{u}_j^1$, $\boldsymbol{u}_{j+1}^0-\boldsymbol{u}_j^0$, and $\boldsymbol{u}_j^0-\boldsymbol{u}_{j-1}^0$ to the characteristic field by left-multiplying with matrix $\boldsymbol{L}$. Secondly, apply the scalar limiter to each component of the transformed vectors $\boldsymbol{L}\boldsymbol{u}_j^1$, $\boldsymbol{L}(\boldsymbol{u}_{j+1}^0-\boldsymbol{u}_j^0)$, and $\boldsymbol{L}(\boldsymbol{u}_j^0-\boldsymbol{u}_{j-1}^0)$. Finally, transform the limited version of the vector back to the original space by left-multiplying with matrix $\boldsymbol{R}$.

In order to identify the location of every fluid we are concerned with, the equation

$$\frac{\mathrm{d}x(\xi,t)}{\mathrm{d}t}=v(\xi,t) \tag{26}$$

should be discretized. Integrating Eq. (26) in time interval $[t^n,t^{n+1}]$, we can obtain

$$x(\xi,t^{n+1})=x(\xi,t^n)+\int_{t^n}^{t^{n+1}}v(\xi,t)\,\mathrm{d}t. \tag{27}$$

By using an appropriate numerical integration formula, the location of the fluid in time t^{n+1} can be obtained. In all the following numerical experiments, the formula

$$x(\xi,t^{n+1})=x(\xi,t^n)+\frac{\delta t}{2}(v(\xi,t^n)+v(\xi,t^{n+1})) \tag{28}$$

is used. If the fluid we concerned locates within the cells, then the fluid velocity is already known. If the fluid we are concerned with is located at the interface of the cells, the Roe average is used to derive the velocity

$$v_{j+\frac{1}{2}}=\frac{\sqrt{\rho_{i+1/2}^-}v_{i+1/2}^-+\sqrt{\rho_{i+1/2}^+}v_{i+1/2}^+}{\sqrt{\rho_{i+1/2}^-}+\sqrt{\rho_{i+1/2}^+}}. \tag{29}$$

3. Numerical results

In this section, some numerical experiments for one-dimensional inviscid compressible gas dynamic equations are performed. The Lagrangian computation and the ideal gas with $\gamma=1.4$ are used in the following tests.

Example 1 This numerical example is used for accuracy test. We solve the nonlinear system of Euler equations with the following initial conditions:

$$\rho(x,0)=1+0.2\sin(\pi x),\ u(x,0)=1,\ p(x,0)=1,$$

and 2-periodic boundary conditions. The exact solution is

$$\rho(x,t)=1+0.2\sin(\pi(x-t)),\ u(x,t)=1,\ p(x,t)=1.$$

The L^∞ and L^1 errors for ρ are defined as

$$\|e\|_{\mathrm{L}^\infty}=\max_{i=1}^{N}(|\rho(\xi_i,T)-\rho_h(\xi_i,T)|), \tag{30}$$

$$\|e\|_{\mathrm{L}^1}=\int_{\mathrm{DL}}(|\rho(\xi_i,T)-\rho_h(\xi_i,T)|)\,\mathrm{d}\xi, \tag{31}$$

where ρ and ρ_h are the corresponding exact and numerical solutions for density, and DL represents the computational domain in the Lagrangian coordinate. The numerical orders of accuracy are calculated as follows:

$$O_{\mathrm{L}^\infty}=\frac{\log\left(\frac{\|e\|_{\mathrm{L}^\infty,2h}}{\|e\|_{\mathrm{L}^\infty,h}}\right)}{\log 2},\ O_{\mathrm{L}^1}=\frac{\log\left(\frac{\|e\|_{\mathrm{L}^1,2h}}{\|e\|_{\mathrm{L}^1,h}}\right)}{\log 2}.$$

Firstly, we compute the solution up to $t=2$. The errors and the numerical orders of accuracy are shown in Table 1. We can see that the schemes we constructed behave very well. Furthermore, the numerical schemes can work for a bit longer time. Figure 1 gives the comparisons of numerical and exact solutions at time $t=100$ with two different meshes.

Table 1. Errors and numerical orders of accuracy of density ρ at $t=2$ in example 1.

Grids	$\|e\|_{\mathrm{L}^\infty}$	O_{L^∞}	$\|e\|_{\mathrm{L}^1}$	O_{L^1}
10	3.16909×10^{-2}		1.84083×10^{-2}	
20	1.92845×10^{-2}	0.71663	4.88476×10^{-3}	1.91400
40	1.04250×10^{-2}	0.88740	1.30948×10^{-3}	1.89929
80	5.39036×10^{-3}	0.95159	3.33241×10^{-4}	1.97436
160	2.73696×10^{-3}	0.97780	8.39284×10^{-5}	1.98933
320	1.37857×10^{-3}	0.98941	2.10565×10^{-5}	1.99490

Example 2 We consider the Riemann problems of the gas dynamics equation for a polytropic gas. The Sod shock wave problem is discussed in this example. The initial conditions are $(\rho_{\mathrm{L}},u_{\mathrm{L}},p_{\mathrm{L}})=(1.0,0,1.0)$, $(\rho_{\mathrm{R}},u_{\mathrm{R}},p_{\mathrm{R}})=(0.125,0,0.10)$. The initial computational domain is $[0,1]$, and the discontinuity locates at $x_0=0.5$. The transmissive boundary conditions are applied at both ends of the computational domain. The exact solution of this example consists of a left rarefaction wave, a right-travelling shock wave, and a stationary contact discontinuity. Figure 2 shows the numerical results with 200 initially uniform cells at $t=0.2$. There is no oscillation, the shock transition region is grasped sharply.

Example 3[49] We consider the Riemann problems of the gas dynamics equation for the polytropic gas. The Lax shock

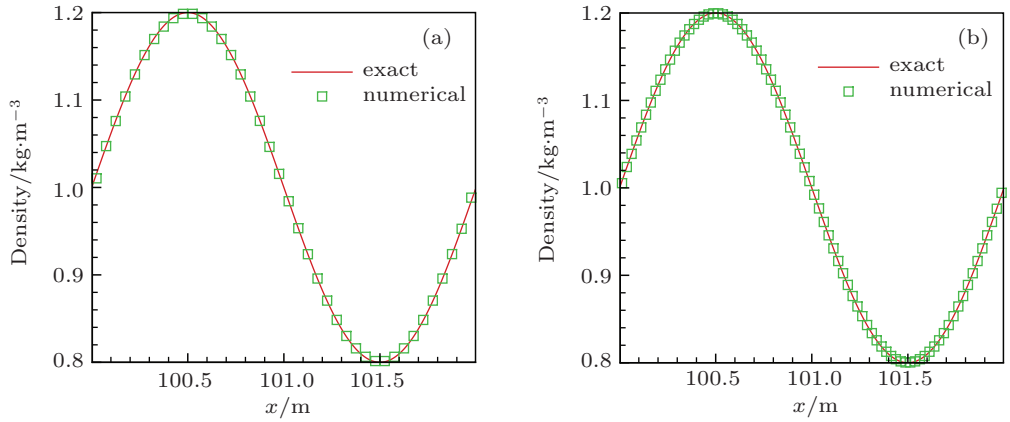


Fig. 1. (color online) Comparison between the RKCV method solution and the exact solution at time 100.0 with different grids ((a) 40 cells, (b) 80 cells) for example 1.

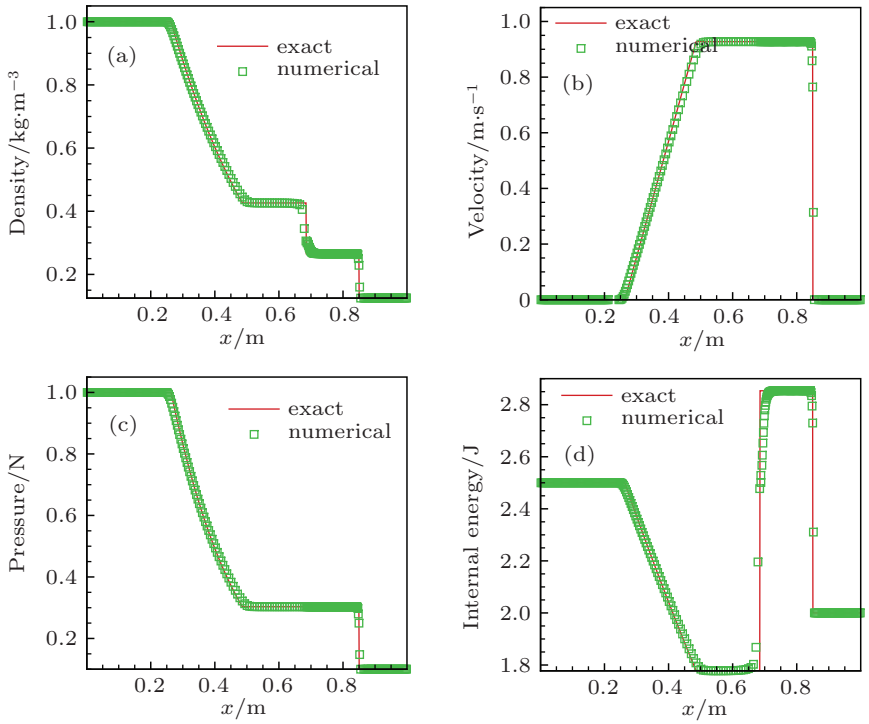


Fig. 2. (color online) Comparison between the RKDG solution and the exact solution at time 0.2 for the Sod problem in example 2: (a) density, (b) velocity, (c) pressure, (d) energy.

wave problem is discussed in this example. The initial conditions are $(\rho_{\rm L}, u_{\rm L}, p_{\rm L}) = (0.445, 0.698, 3.528)$, $(\rho_{\rm R}, u_{\rm R}, p_{\rm R}) = (0.5, 0, 0.571)$. The initial computational domain is $[0, 1]$, and the initial discontinuity locates at $x_0 = 1$. The transmissive boundary conditions are applied at both ends of the computational domain. Figure 3 shows the numerical results with 200 initially uniform cells at $t = 0.13$.

Example 4[45] We consider the two interacting blast waves problem. The initial conditions are

$$\rho = 1, \quad u = 1, \quad p = \begin{cases} 10^3, & 0 < x < 0.1, \\ 10^{-2}, & 0.1 < x < 0.9, \\ 10^2, & 0.9 < x < 1.0. \end{cases}$$

The reflective boundary conditions are applied at $x = 0$ and $x = 1$. Figures 4 and 5 show the numerical results at $t = 0.038$ for different initial uniform cells. From the numerical results, we can see that the numerical results with 400 cells and 800 cells have no obvious difference.

Comparing the numerical solution to the exact solution of Woodward and Colella,[50] we find that all important features of the various interaction are shown, thus our solution can be considered as a converged one. Our numerical results can be compared with other numerical results in some references.[45,49]

Example 5 We consider the shock entropy wave interaction. To demonstrate the advantage of the higher-order methods, we use the Euler equations with initial conditions

$$(\rho_L, u_L, p_L) = (3.85714, 2.629369, 10.33333), \quad x < -4,$$
$$(\rho_R, u_R, p_R) = (1.0 + \varepsilon \sin(kx), 0, 1.0), \quad x \geq -4,$$

where ε and k are the amplitude and the wave number of the entropy wave, respectively. In our test, we take $\varepsilon = 0.2$ and $k = 5$. The final time is $t = 1.8$. This problem is very suitable for testing the advantage of a high-order scheme when the solution contains both shocks and complete smooth structures. The initial computational domain is $[-10, 5]$. The numerical results for density with different meshes are shown in Fig. 6.

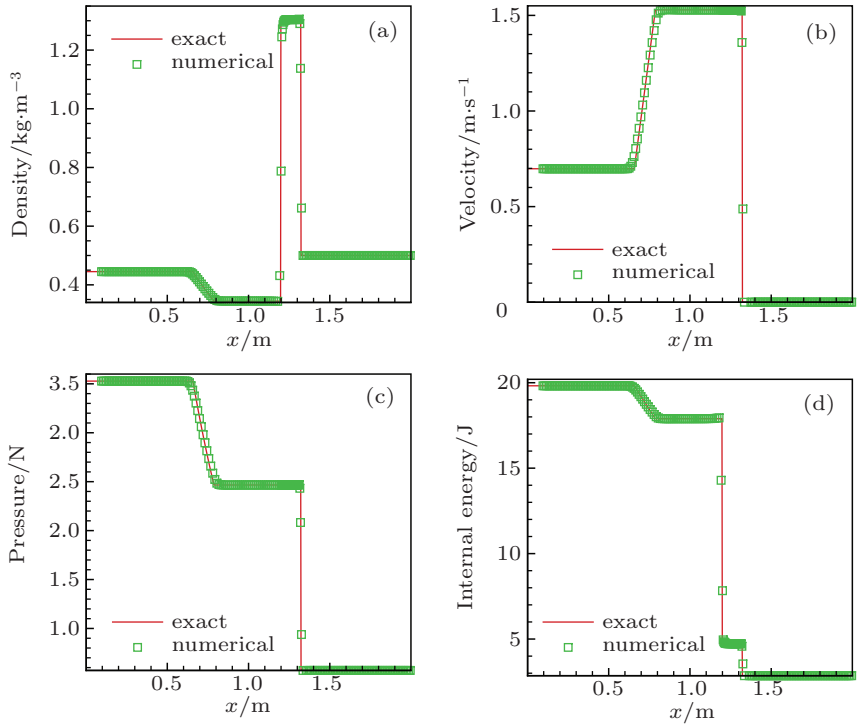


Fig. 3. (color online) Comparison between the RKDG solution and the exact solution at time 0.13 for the Lax problem in example 3: (a) density, (b) velocity, (c) pressure, (d) energy.

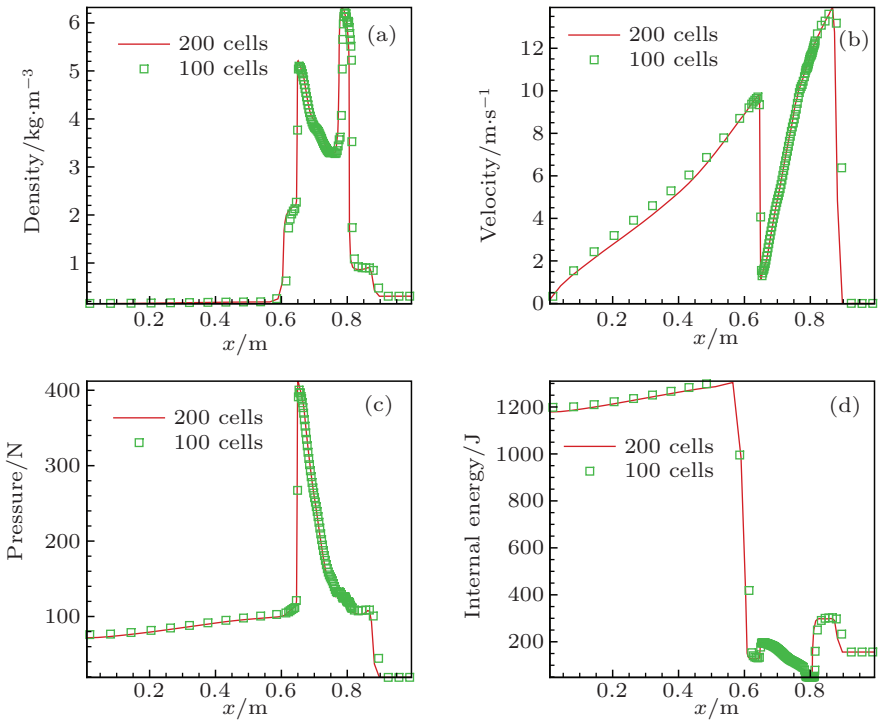


Fig. 4. (color online) The RKCV method solution at time 0.038 for two interacting blast waves problem with 100 cells and 200 cells in example 4: (a) density, (b) velocity, (c) pressure, (d) energy.

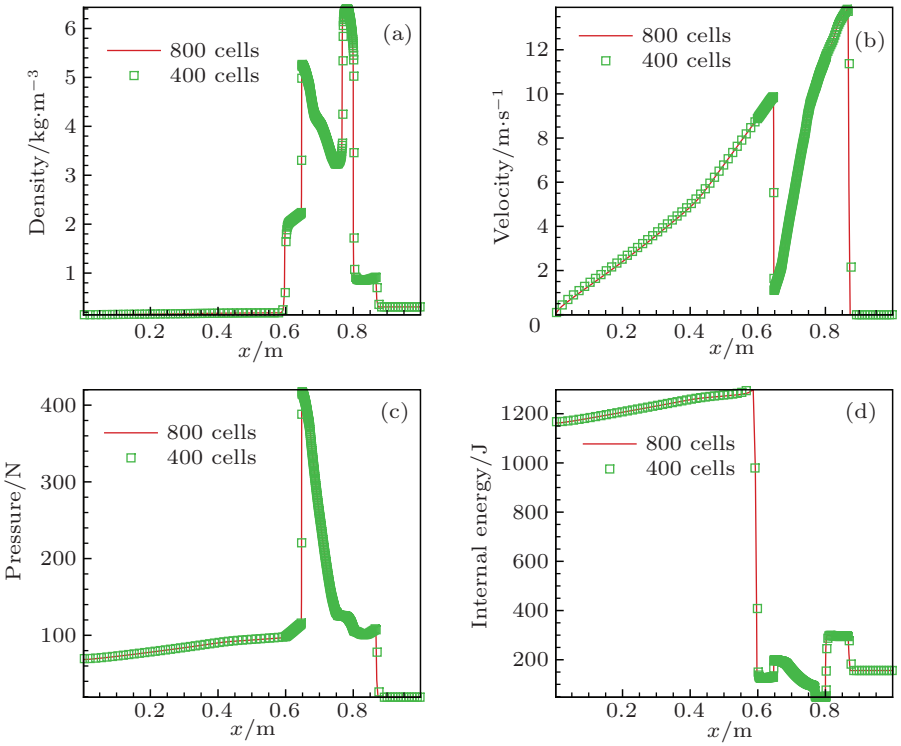


Fig. 5. (color online) The RKCV method solution at time 0.038 for two interacting blast waves problem with 400 cells and 800 cells in example 4: (a) density, (b) velocity, (c) pressure, (d) energy.

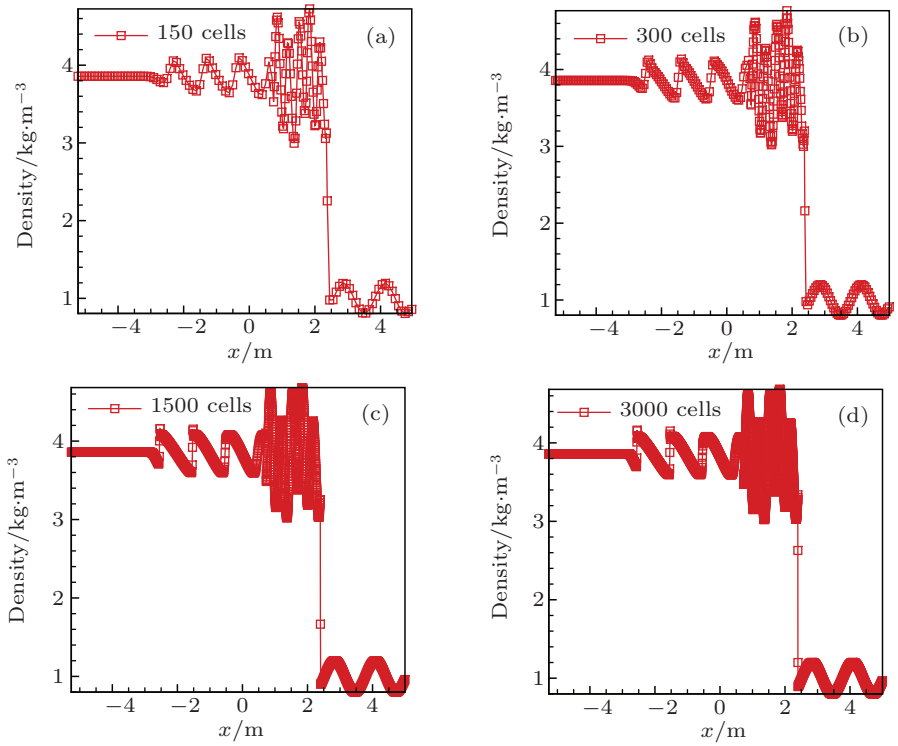


Fig. 6. (color online) Density at time 1.8 with different cells ((a) 50 cells, (b) 300 cells, (c) 1500 cells, (d) 3000 cells) and TVB limiter ($M = 50$) for the shock entropy wave interaction problem in example 5.

In order to compare the time consumed by using the RKDG and the RKCV methods, we perform the numerical simulations by using a fixed time step with different meshes. From the numerical results, we can see clearly that the RKDG method consumes more time than the RKCV method. The algorithm is implemented using the Fortran programm on a Vostro 220s Dell-PC. All computations are performed in double precision.

Table 2. Time consumed by using RKDG and RKCV methods for examples 1–5 with different time steps and meshes.

Example	Time step	Mesh	Terminal time/s	RKDG/s	RKCV/s
Example 1	0.000100	200	2.000	13.1508843	12.2460785
Example 2	0.000100	400	0.200	2.52721620	2.38681530
Example 3	0.000100	400	0.130	1.63801050	1.56001000
Example 4	0.000001	800	0.038	97.4850249	94.6770069
Example 5	0.000100	300	1.800	17.7997141	16.5517061

4. Conclusion

In this paper, the RKCV method for solving the one-dimensional in-viscid compressible Euler equations in the Lagrangian coordinate is developed. Several numerical examples are given to demonstrate the accuracy and the reliability of the algorithm. It may be used in more complex fluid interaction simulations, especially for the multi-medium fluid simulation.

References

[1] Bai D M and Zhang L M 2011 *Commun. Nonlinear Sci. Numer. Simul.* **16** 1263
[2] Beljadid A, Mohammadian A and Qiblawey H 2012 *Computers & Fluids* **62** 64
[3] Steve Y and Pozrikidis C 1998 *Computers & Fluids* **27** 879
[4] Cheng R J and Cheng Y M 2011 *Chin. Phys. B* **20** 070206
[5] Wang J F, Sun F X and Cheng Y M 2012 *Chin. Phys. B* **21** 090204
[6] Cheng R J and Ge H X 2009 *Chin. Phys. B* **18** 4059
[7] Ge H X, Liu Y Q and Cheng R J 2012 *Chin. Phys. B* **21** 010206
[8] Wang J F, Sun F X and Cheng R J 2010 *Chin. Phys. B* **19** 060201
[9] Cheng R J and Cheng Y M 2008 *Acta Phys. Sin.* **57** 6037 (in Chinese)
[10] Ren H P, Cheng Y M and Zhang W 2009 *Chin. Phys. B* **18** 4065
[11] Cheng R J and Ge H X 2012 *Chin. Phys. B* **21** 100209
[12] Yu H M, Cheng R J and Ge H X 2010 *Chin. Phys. B* **19** 100512
[13] Cheng R J and Ge H X 2010 *Chin. Phys. B* **19** 090201
[14] Cheng R J and Ge H X 2012 *Chin. Phys. B* **21** 040203
[15] Chen L and Cheng Y M 2008 *Acta Phys. Sin.* **57** 6047 (in Chinese)
[16] Cheng Y M, Wang J F and Bai F N 2012 *Chin. Phys. B* **21** 090203
[17] Cheng Y M, Li R X and Peng M J 2012 *Chin. Phys. B* **21** 090205
[18] Després B and Mazeran C 2005 *Archive for Rational Machanics and Analysis* **178** 327
[19] Cheng J and Shu C W 2007 *Commun. Comput. Phys.* **227** 1008
[20] Loubère R, Ovadia J and Abgrall R 2004 *Int. J. Numer. Meth. Fluids* **44** 645
[21] Maire P H 2009 *J. Comput. Phys.* **228** 2391
[22] François V, Maire P H and Abgrall R 2011 *Computers & Fluids* **46** 498
[23] Zhao G Z and Yu X J 2012 *Acta Phys. Sin.* **61** 110208 (in Chinese)
[24] Zhao G Z, Yu X J and Zhang R P 2012 *Chin. J. Comput. Phys.* **29** 166 (in Chinese)
[25] Cheng J and Shu C W 2007 *J. Comput. Phys.* **227** 1567
[26] Shen Z J, Yan W and Lv G X 2010 *J. Comput. Phys.* **229** 4522
[27] Hirt C, Amsden A and Cook J 1974 *J. Comput. Phys.* **14** 227
[28] Reed W H and Hill T R 1973 *Triangular Mesh Methods for the Neutron Transport Equation* (Los Alamos Scientific Laboratory Report LA-UR-73-479)
[29] Cockburn B and Shu C W 1991 *Math. Model. Numer. Anal.* **25** 337
[30] Cockburn B and Shu C W 1989 *Math. Comp.* **52** 411
[31] Cockburn B, Lin S Y and Shu C W 1989 *J. Comput. Phys.* **84** 90
[32] Cockburn B, Hou S and Shu C W 1990 *Math. Comp.* **54** 545
[33] Cockburn B and Shu C W 1998 *J. Comput. Phys.* **141** 199
[34] Cockburn B and Shu C W 2001 *J. Sci. Comput.* **16** 173
[35] Chen R S and Yu X J 2006 *Chin. J. Comput. Phys.* **23** 43
[36] Qiu J X, Liu T G and Khoo B C 2008 *Commun. Comput. Phys.* **3** 479
[37] Qiu J X, Liu T G and Khoo B C 2007 *J. Comput. Phys.* **222** 353
[38] Zhang R P, Yu X J and Zhao G Z 2011 *Chin. Phys. B* **20** 110205
[39] Zhang R P, Yu X J and Feng T 2012 *Chin. Phys. B* **21** 030202
[40] Zhang R P and Zhang L W 2012 *Chin. Phys. B* **21** 090206
[41] Li R H, Chen Z and Wu W 2000 *Generalized Difference Methods for Differential Equations* (New York: Marcel Dekker)
[42] Baliga B R and Patankar S V 1980 *Numerical Heat Transfer* **3** 393
[43] Chen Z X 2006 *Networks and Heterogeneous Media* **1** 689
[44] Chen D W and Yu X J 2009 *Chin. J. Comput. Phys.* **26** 501
[45] Chen D W, Yu X J and Chen Z X 2011 *Int. J. Numer Meth. Fluids* **67** 711
[46] Francçois V 2012 *Computers & Fluids* **64** 64
[47] Jia Z P and Zhang S D 2011 *J. Comput. Phys.* **230** 2496
[48] Shu C W 1988 *SIAM J. Sci. Stat. Comput.* **9** 1073
[49] Toro E F 1997 *Riemann Solvers and Numerical Methods for Fluid Dynamics — a Practical Introduction* (2nd edn.) (Berlin: Springer)
[50] Woodward P and Colella P 1984 *J. Comput. Phys.* **54** 115