Commun. Comput. Phys.
doi: 10.4208/cicp.210313.181213s

Vol. **15**, No. 4, pp. 1184-1206
April 2014

A Runge Kutta Discontinuous Galerkin Method for Lagrangian Compressible Euler Equations in Two-Dimensions

Zhenzhen Li[1,2], Xijun Yu[3,*], Jiang Zhu[4] and Zupeng Jia[3]

[1] *School of Mathematical Sciences, University of Science and Technology of China, Hefei 230026, P.R. China.*
[2] *Graduate School, China Academy of Engineering Physics, Beijing 100088, P.R. China.*
[3] *Laboratory of Computational Physics, Institute of Applied Physics and Computational Mathematics, Beijing 100088, P.R. China.*
[4] *National Laboratory for Scientific Computing, LNCC/MCTI, Avenida Getúlio Vargas 333, 25651-075 Petrópolis, RJ, Brazil.*

Received 21 March 2013; Accepted (in revised version) 18 December 2013

Communicated by Kun Xu

Available online 21 January 2014

Abstract. This paper presents a new Lagrangian type scheme for solving the Euler equations of compressible gas dynamics. In this new scheme the system of equations is discretized by Runge-Kutta Discontinuous Galerkin (RKDG) method, and the mesh moves with the fluid flow. The scheme is conservative for the mass, momentum and total energy and maintains second-order accuracy. The scheme avoids solving the geometrical part and has free parameters. Results of some numerical tests are presented to demonstrate the accuracy and the non-oscillatory property of the scheme.

AMS subject classifications: 65M60, 76M10

Key words: Lagrangian type scheme, compressible Euler equations, RKDG method, conservative scheme.

1 Introduction

In determining a numerical method for the multi-dimensional fluid flow, there are two typical choices. One is the Lagrangian framework in which the mesh is embedded in the

[*] Corresponding author. *Email addresses:* lyzhen@mail.ustc.edu.cn (Z. Z. Li), yuxj@iapcm.ac.cn (X. J. Yu), jiang@lncc.br (J. Zhu), zpjia@iapcm.ac.cn (Z. P. Jia)

fluid and moves with it. And the other is the Eulerian framework in which the mesh is treated as a fixed reference frame when the fluid moves. More generally, the grid points may be moved in some arbitrarily specified way that it is called the Arbitrary Lagrangian-Eulerian method (ALE). Most ALE algorithms consist of three steps. One is the Lagrangian step in which the solution and the grid are updated. The second step is a rezoning mesh in which the grid nodes are moved to a more optimal position and the third is remapping values in which the Lagrangian solution values are transferred to the new grid. In the numerical simulations of multi-material compressible fluid flows, the difficulty is how to handle the moving medium interface. Since the Lagrangian method can calculate fluid interface clearly, we present a new Lagrangian type scheme for solving the Euler equations of compressible gas dynamics in this paper.

An important point of constructing a Lagrangian discrete scheme is to decide where to locate degrees of freedom. It is generally separated into two kinds. One is the staggered scheme in which the velocity is defined at the nodes while the other variables are located inside the cells. This type of scheme was first introduced by von Neumann and Richtmyerfor in [2] for one-dimensional flows. The two-dimensional scheme was extended by Wilkins in [23] based on an internal energy formulation. The scheme was not conservative and admitted numerical spurious modes. In spite of these drawbacks, the scheme has been widely used for many years. Of course, many improvements have been made in order to solve the previous problems. Caramana and Shashkov constructed a staggered scheme which ensures the conservation of total energy in [1]. In [3], based on a mimetic finite difference scheme, Campbell and Shashkov improved the discretization of artificial viscosity so that the staggered Lagrangian scheme is an accurate and robust method. The other is the cell-centered scheme in which all variables are defined inside the cells. Here we concentrate our interests in the cell-centered schemes. This is because the staggered Lagrangian schemes use different control volumes for primary variables, it is very difficult to construct coherent high-order schemes for all these variables. On the contrary, the cell-centered Lagrangian schemes use only one control volume for primary variables, thus it is possible to construct coherent high-order schemes for all these variables.

In general, there are two kinds of the cell-centered Lagrangian schemes.

The first is that the mesh moves with the flow velocity (e.g. [4–8]). In [4], Cheng and Shu developed a class of Lagrangian cell-centered schemes on quadrilateral meshes. Their schemes are based on the finite volume method and achieve a higher order accuracy by using the high-order ENO reconstruction. The schemes are conservative for the density, momentum and total energy, are essentially non-oscillatory, have free parameters, and can maintain formal high order accuracy both in space and time. But the ENO reconstruction requires the information from the surrounding cells. Therefore, this method has less compactness for high order schemes. In [6], Maire et al. developed a new Lagrangian cell-centered scheme. In their scheme the vertex velocities and numerical fluxes through the cell interfaces are not computed independently as usual but in a consistent manner. The scheme feature is the introduction of four pressures on each edge, two for each node on each side of the edge. In the limit of a one-dimensional flow computed by their two-dimensional solver, the scheme recovers the classical Godunov approximate Riemann solver. This scheme is first order accurate.

The second is that the mesh is fixed in the Lagrangian space (e.g. [9–12, 19]). Després and Mazeran proposed a new and canonical way of writing the equations of gas dynamics in a fully Lagrangian form in [19]. They showed that the physical part is symmetrizable and that the weak hyperbolicity is due to shear contact discontinuities. Based on this formulation, a new conservative and entropy consistent Lagrangian scheme in two-dimension was obtained by using the finite volume method. The node velocity is computed in a coherent manner with the face fluxes. However, it appears that the scheme leads to a nodal velocity depending on the cell aspect ratio in the case of one-dimensional flows. In [9], Hui et al. introduced a unified coordinate system. In this system the flow variables are considered to be functions of time and of some permanent identification of pseudo-particles which move with velocity hU, where U is the velocity of fluid particles, and h is a parameter. This turns out to be a unified description, ranging from Eulerian when $h=0$ to Lagrangian when $h=1$, and the freedom in choosing h makes it possible to avoid the disadvantages of excessive diffusion across slip lines in the Eulerian description and of severe grid deformation in the Lagrangian description. But this method has to choose a proper h in every test case. In [10], Jia and Zhang solved a fully Lagrangian form of the gas dynamics equations which contains the physical part and the geometrical part. They discretized this system in the Lagrangian space by discontinuous Galerkin method using a spectral basis. The feature is that the vertex velocities and numerical fluxes are computed by using the method in [6]. In [12], Zhao and Yu also used discontinuous Galerkin method to solve the gas dynamics equations which contains the physical part and the geometrical part, and they used the Lax-Friedriches flux in their DG scheme. The schemes in [10] and [12] can maintain high-order accuracy both in space and time and has free parameter, but their system has eight equations and it is weakly hyperbolic system.

On the basis of previous studies, in this paper we give a new Lagrangian scheme for solving the compressible Euler equations. In this new scheme the system of equations is discretized by the RKDG finite element method. The mesh moves with the fluid flow and the scheme uses the Lax-Friedrichs and the HLLC numerical flux. In order to control spurious numerical oscillation near the discontinuity, we use a slope limiter in the scheme. The new scheme is conservative for the mass, momentum and total energy. The scheme avoids solving the geometrical part, and has no parameters to be tuned for individual test cases. Several two-dimensional numerical examples show that our scheme can achieve uniformly second-order accuracy on moving and distorted Lagrangian meshes. Moreover, it is essentially non-oscillatory.

The paper is organized as follows. In Section 2 we recall the gas dynamics equations written in the Lagrangian form, and give their weak form. In Section 3 we use the RKDG method to solve this weak form. In Section 4 we validate our new scheme with several test cases which demonstrate the non-oscillatory property and the accuracy. Finally, we give concluding remarks in Section 5.

2 Derivation of the Euler equations in the Lagrangian formalism

In this section we recall briefly how compressible Euler equations are written in the Lagrangian formalism which follow the approach developed in [6], and give its weak form.

Let D be a region filled with an inviscid fluid. Considering a fluid particle, its initial position is the point M, and at time $t>0$ it is moving through the point m. The coordinates of point M are denoted by (X,Y) and are named Lagrangian coordinates. The coordinates of point m are denoted by (x,y) and are named Eulerian coordinates. The Eulerian coordinates are obtained from the trajectory equations

$$\frac{dx}{dt}=u(t,x,y), \quad x(0,X,Y)=X, \tag{2.1a}$$

$$\frac{dy}{dt}=v(t,x,y), \quad y(0,X,Y)=Y, \tag{2.1b}$$

where $V=(u,v)$ is the fluid velocity. If the velocity field is smooth enough, there exists a unique smooth solution $(x(t,X,Y),y(t,X,Y))$. The map is defined as

$$\chi_t:(X,Y)\rightarrow(x,y), \tag{2.2}$$

where (x,y) is the unique solution of (2.1). With fixed time $t>0$, this map advances each fluid particle from its position at time $t=0$ to its position at time t. If Ω denotes a region in D, then $\chi_t(\Omega)=W$ is the volume Ω moving with the fluid. We assume that for each $t>0$, χ_t is invertible. The Jacobian of the map is

$$J(X,Y,t)=\begin{vmatrix} \dfrac{\partial x}{\partial X} & \dfrac{\partial y}{\partial X} \\ \dfrac{\partial x}{\partial Y} & \dfrac{\partial y}{\partial Y} \end{vmatrix}. \tag{2.3}$$

Since $J(X,Y,0)=1$, and χ_t is invertible, for each $t>0$, we have $J(X,Y,t)>0$. We know that the time differentiation of (2.3) is

$$\frac{dJ}{dt}-J\nabla\cdot V=0, \tag{2.4}$$

where $\nabla\cdot$ denotes the divergence operator, namely, $\nabla\cdot V=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}$.

Let us consider a flow variable $\varphi:\varphi\equiv\varphi(x,y,t)$ depending on the Eulerian coordinates. The time differentiation of φ is

$$\frac{d\varphi}{dt}=\frac{\partial\varphi}{\partial t}+\frac{dx}{dt}\frac{\partial\varphi}{\partial x}+\frac{dy}{dt}\frac{\partial\varphi}{\partial y}.$$

By using (2.1) we have

$$\frac{d\varphi}{dt}=\frac{\partial\varphi}{\partial t}+\nabla\varphi\cdot V, \tag{2.5}$$

where ∇ denotes the gradient operator, namely $\nabla\varphi=(\frac{\partial\varphi}{\partial x},\frac{\partial\varphi}{\partial y})^T$. Finally, by combining (2.4) and (2.5) we can have

$$\frac{d}{dt}(\varphi J)=J\Big[\frac{\partial\varphi}{\partial t}+\nabla\cdot(\varphi V)\Big]. \tag{2.6}$$

The system of compressible Euler equations (see [20]) is

$$\begin{cases} \frac{\partial\rho}{\partial t}+\nabla\cdot(\rho V)=0,\\ \frac{\partial}{\partial t}(\rho V)+\nabla\cdot(\rho V\otimes V)+\nabla p=0,\\ \frac{\partial}{\partial t}(\rho E)+\nabla\cdot(\rho EV)+\nabla\cdot(pV)=0, \end{cases} \tag{2.7}$$

where ρ is the density, p is the press, and E is the specific total energy. We denote the specific internal energy by $\varepsilon=E-\frac{1}{2}\|V\|^2$. The thermodynamic closure of (2.7) is given by the equation of state $p\equiv p(\rho,\varepsilon)$.

Now, by using (2.6) for the conservative variables $\varphi=\rho,\rho V,\rho E$ and substituting them into (2.7), we get the compressible Euler equations in Lagrangian form as

$$\begin{cases} \frac{d}{dt}(\rho J)=0,\\ \frac{d}{dt}(\rho JV)+J\nabla p=0,\\ \frac{d}{dt}(\rho JE)+J\nabla\cdot(pV)=0. \end{cases} \tag{2.8}$$

We notice that (2.8) is only a semi-Lagrangian formula since the gradient and divergence operate on variables which depend on Eulerian coordinates. In order to write (2.8) in a full Lagrangian way, one has to express the gradient and divergence operators with Lagrangian coordinates using the map χ_t.

In order to use the discontinuous Galerkin (DG) method to perform the space discretization in the next section, we give the weak form of system (2.8). That is, both sides of Eq. (2.8) are multiplied by the function $\psi(X,Y)\in L_2(\Omega)$ and integrated on the Lagrangian domain Ω:

$$\begin{cases} \frac{d}{dt}\int_\Omega \rho J\psi d\Omega=0,\\ \frac{d}{dt}\int_\Omega \rho JV\psi d\Omega+\int_\Omega J\nabla p\psi d\Omega=0,\\ \frac{d}{dt}\int_\Omega \rho JE\psi d\Omega+\int_\Omega J\nabla\cdot(pV)\psi d\Omega=0. \end{cases} \tag{2.9}$$

Since χ_t is invertible, we have

$$\psi(X,Y)=\psi(\chi_t^{-1}(x,y))=\psi\cdot\chi_t^{-1}(x,y),$$

and $\psi(X,Y)\in L_2(\Omega)$, χ_t^{-1} is invertible, so

$$\psi\cdot\chi_t^{-1}(x,y)\in L_2(W).$$

We set $\psi\cdot\chi_t^{-1}(x,y)=\varphi(x,y)$. By knowing that $\chi_t(\Omega)=W$ and $J\cdot d\Omega=dW$, we can get

$$\begin{cases} \frac{d}{dt}\int_W \rho\varphi dW=0, \\ \frac{d}{dt}\int_W \rho V\varphi dW+\int_W \nabla p\varphi dW=0, \\ \frac{d}{dt}\int_W \rho E\varphi dW+\int_W \nabla\cdot(pV)\varphi dW=0, \end{cases} \tag{2.10}$$

where $V=(u,v)$ is the velocity. We call that (2.10) is the weak form of (2.8). We notice that $\frac{d}{dt}\psi(X,Y)=0$, Eq. (2.10) are obtained by performing a mapping transformation on (2.9).

We set $\varphi=1$ and use Green's formula, then (2.10) becomes

$$\begin{cases} \frac{d}{dt}\int_W \rho dW=0, \\ \frac{d}{dt}\int_W \rho V dW+\int_{\partial W} p\mathbf{n}dl=0, \\ \frac{d}{dt}\int_W \rho E dW+\int_{\partial W} pV\cdot\mathbf{n}dl=0, \end{cases} \tag{2.11}$$

where $\mathbf{n}$ is the unit outward normal vector of ∂W. This system (2.11) represents the conservation of mass, momentum and energy so that the system (2.10) also satisfies the conservation of mass, momentum and energy.

3 RKDG method

3.1 Spatial discretization

In this section, we present a spatial discretization of (2.10) by using the DG method.

Note that $W_{t=0}=\Omega$, Ω is discretized into $M\times N$ computational cells. As time progresses, we use the velocity at the vertex to get the new fluid region W and the new partition $\{W_{ij},\ i=1,\cdots,M,\ j=1,\cdots,N\}$. The four vertices of cell W_{ij} are $\{(x_{i,j},y_{i,j}),\ (x_{i+1,j},y_{i+1,j}),\ (x_{i+1,j+1},y_{i+1,j+1}),\ (x_{i,j+1},y_{i,j+1})\}$.

We define a finite-element space consisting of piecewise polynomials

$$V_h^k=\Big\{\varphi_h\in L_2(W): \varphi_h|_{W_{ij}}\in P_k(W_{ij}); 1\le i\le M, 1\le j\le N\Big\},$$

where $P_k(W_{ij})$ denotes a set of polynomials of degree up to k defined on the cell W_{ij}. We use the DG method to solve (2.10) as follows: find the approximate function $U_h\in V_h^k$ such that, for all test functions $\varphi_h\in V_h^k$ and all $1\le i\le M, 1\le j\le N$, we have

$$\begin{cases} \frac{d}{dt}\int_{W_{ij}} \rho_h\varphi_h dW_{ij}=0, \\ \frac{d}{dt}\int_{W_{ij}} (\rho V)_h\varphi_h dW_{ij}+\int_{W_{ij}} \nabla p_h\varphi_h dW_{ij}=0, \\ \frac{d}{dt}\int_{W_{ij}} (\rho E)_h\varphi_h dW_{ij}+\int_{W_{ij}} \nabla\cdot(pV)_h\varphi_h dW_{ij}=0. \end{cases} \tag{3.1}$$

where $U_h=(\rho_h,(\rho V)_h,(\rho E)_h)^t=(\rho_h,(\rho u)_h,(\rho v)_h,(\rho E)_h)^t$, and we set

$$F(U)=(0,p,0,pu)^t, \quad G(U)=(0,0,p,pv)^t.$$

Then (3.1) can be rewritten as

$$\frac{d}{dt}\int_{W_{ij}} U_h\varphi_h dW_{ij}+\int_{W_{ij}}\nabla\cdot(F(U_h),G(U_h))\varphi_h dW_{ij}=0. \tag{3.2}$$

By using Green's formula we get

$$\frac{d}{dt}\int_{W_{ij}} U_h\varphi_h dW_{ij}=\int_{W_{ij}}(F(U_h),G(U_h))\cdot\nabla\varphi_h dw_{ij}-\int_{\partial W_{ij}}\widehat{(F,G)\cdot\mathbf{n}}\varphi_h dl, \tag{3.3}$$

where $\mathbf{n}=(n_x,n_y)$ is the unit outward normal vector of ∂W_{ij}, $\widehat{(F,G)\cdot\mathbf{n}}$ is the numerical flux, which is a single-valued function defined at the cell interfaces. Here we set $\widehat{H}=\widehat{(F,G)\cdot\mathbf{n}}$.

In this paper, we use the following two typical numerical fluxes:

(1) The L-F (Lax-Friedrichs) flux

$$\widehat{H}_{W_{ij}}=\widehat{H}_{W_{ij}}(U_h^-,U_h^+)=\frac{1}{2}[(F(U_h^-),G(U_h^-))\cdot\mathbf{n}+(F(U_h^+),G(U_h^+))\cdot\mathbf{n}-\alpha_{ij}(U_h^+-U_h^-)],$$

where $U_h^\pm$ are the values of U_h inside and outside the cell W_{ij}, and α_{ij} is taken as an upper bound for the eigenvalues of the Jacobian. Here in the Lagrangian formulation, $\alpha_{ij}=\max(c_{ij}^+,c_{ij}^-)$, $c_{ij}^\pm$ are the values of the sound speed inside and outside the cell.

(2) The HLLC (Harten-Lax-van Leer contact wave) flux.

We refer to [14] for the details of the HLLC flux. This flux has been successfully used to compute the compressible Euler equations in the Lagrangian framework [4].

3.2 The basis functions

Let the Taylor basis be a set of basis functions of V_h^k. For the P_1 case, assuming that u_h is a component of U_h, we use the following expression for the approximate solution

$$u_h(x,y,t)=\sum_{l=1,3}u_{ijl}(t)B_{lij}(x,y),\quad (x,y)\in W_{ij}, \tag{3.4}$$

where the basis functions are

$$B_{1ij}(x,y)=1,\quad B_{2ij}(x,y)=\frac{x-x_c}{\Delta x},\quad B_{3ij}(x,y)=\frac{y-y_c}{\Delta y}.$$

Here (x_c,y_c) is the centroid of the cell W_{ij}. $\Delta x=0.5(x_{\max}-x_{\min})$, $\Delta y=0.5(y_{\max}-y_{\min})$, and $x_{\max}$, $x_{\min}$, $y_{\max}$ and $y_{\min}$ are the maximum and minimum coordinates in the cell W_{ij} in x-, and y-direction, respectively.

Because for all test functions $\varphi_h\in V_h^k$, (3.3) is right. We substitute (3.4) into (3.3), and set $\varphi_h=B_{mij}$, $m=1,2,3$, then we have

$$\frac{d}{dt}\mathbf{M_{ij}}\cdot\mathbf{u_{hij}}=L(U_h), \tag{3.5}$$

where $\mathbf{u_{hij}}=(u_{ij1},u_{ij2},u_{ij3})$, $\mathbf{M_{ij}}=\int_{W_{ij}}B_{lij}B_{mij}dW_{ij}$ $(l=1,2,3;\ m=1,2,3)$ denotes the mass matrix, and L is a discrete spatial operator.

Notice that $P_k(W_{ij})$ and B_{lij} $(l=1,2,3)$ are defined in the cell W_{ij}. As time progresses, W_{ij} is updated, so $P_k(W_{ij})$ and B_{lij} $(l=1,2,3)$ should be updated at every time step. The detail is described in Section 3.4.

3.3 The determination of the vertex velocity

In the Lagrangian formula, the grid moves with the fluid velocity, thus we would need to know the velocity at the vertex to move the grid. Since the velocity is a deduced quantity, we would need to obtain it from the conserved variables. In the following we describe how to determine the vertex velocity in our scheme, which follows the approach in [4].

Considering a vertex (i,j) shared by four edges, we give these four edges a serial number $k=1,2,3,4$, and define the direction of each edge to be the direction of the incremental index i or j, for example the direction of the edge with two endpoints $(i-1,j)$ and (i,j) is from $(i-1,j)$ to (i,j). For the piecewise-constant approximation, we use the cell-averaged solution as left and right variables $(\rho^{k\pm},(\rho u)^{k\pm},(\rho v)^{k\pm},(\rho E)^{k\pm})$ at the point (i,j) of the edge k. For the linear piecewise polynomial approximation, we use the expansion formulation (3.4) of left and right cells to compute the values $(\rho^{k\pm},(\rho u)^{k\pm},(\rho v)^{k\pm},(\rho E)^{k\pm})$ at the point (i,j) of the edge k. Then at the point (i,j) of each edge k we can obtain the left value of velocity $(u^{k-},v^{k-})=((\rho u)^{k-}/\rho^{k-},(\rho v)^{k-}/\rho^{k-})$ and the right value of velocity $(u^{k+},v^{k+})=((\rho u)^{k+}/\rho^{k+},(\rho v)^{k+}/\rho^{k+})$.

We then split the left and right velocities into normal and tangential components along the edge k. Let (n_x^k,n_y^k) be the clockwise unit normal of the edge k, w_t^{k-} and w_t^{k+} be their tangential components, and w_n^{k-} and w_n^{k+} be their normal components. Then we define the tangential velocity of the vertex (i,j) along the edge k

$$w_t^k=\frac{1}{2}(w_t^{k-}+w_t^{k+}),\quad k=1,2,3,4. \tag{3.6}$$

As to the normal velocity, we define it as the Roe average of the normal velocities from its two sides

$$w_n^k=\frac{\sqrt{\rho^-}w_n^{k-}+\sqrt{\rho^+}w_n^{k+}}{\sqrt{\rho^-}+\sqrt{\rho^+}},\quad k=1,2,3,4. \tag{3.7}$$

Thus by the formulas (3.6) and (3.7), we can get four x-velocities and y-velocities at the vertex (i,j) which have the following form

$$w_x^k=w_n^kn_x^k-w_t^kn_y^k,\quad w_y^k=w_n^kn_y^k+w_t^kn_x^k,\quad k=1,2,3,4. \tag{3.8}$$

Finally, the velocity at the vertex (i,j) is obtained as follows

$$u_{i,j}=\frac{1}{4}(w_x^1+w_x^2+w_x^3+w_x^4),\quad v_{i,j}=\frac{1}{4}(w_y^1+w_y^2+w_y^3+w_y^4). \tag{3.9}$$

3.4 Time discretization

The time marching for the semi-discrete scheme (3.5) is implemented by a class of Runge-Kutta (RK) type methods [21]. Since the mesh changes with the time advancing in the Lagrangian simulation, the velocity at the position of each vertex and the size of each cell needs to be updated at each RK stage. In this section the variables with the superscripts n and $n+1$ represent the values of the corresponding variables at the nth and $n+1$th time steps, respectively.

For the first-order spatial discretization, the corresponding time discretization uses the first-order forward Euler step. Now the basis function is $B_{ij}=1$. So

$$\mathbf{M_{ij}}=\int_{W_{ij}}1\cdot 1dW_{ij}=S_{ij},$$

where S_{ij} is the area of cell W_{ij}. We know the value of each variable at the nth time step. By using the vertex velocity we get

$$x_{ij}^{n+1}=x_{ij}^{n}+u_{ij}^{n}\Delta t^{n},\quad y_{ij}^{n+1}=y_{ij}^{n}+v_{ij}^{n}\Delta t^{n},\quad i=1,\cdots,M+1,\ \ j=1,\cdots,N+1,$$

thus the four vertices of cell W_{ij}^{n+1} are $\{(x_{i,j}^{n+1},y_{i,j}^{n+1}),\ (x_{i+1,j}^{n+1},y_{i+1,j}^{n+1}),\ (x_{i+1,j+1}^{n+1},y_{i+1,j+1}^{n+1}),\ (x_{i,j+1}^{n+1},y_{i,j+1}^{n+1})\}$. Then we can get

$$\begin{aligned}\mathbf{M_{ij}^{n+1}}&=\int_{W_{ij}^{n+1}}1\cdot 1dW_{ij}^{n+1}\\&=S_{ij}^{n+1}\\&=\frac{1}{2}[(x_{i+1,j+1}^{n+1}-x_{i,j}^{n+1})(y_{i,j+1}^{n+1}-y_{i+1,j}^{n+1})-(x_{i,j+1}^{n+1}-x_{i+1,j}^{n+1})(y_{i+1,j+1}^{n+1}-y_{i,j}^{n+1})].\end{aligned}$$

Therefore, the first-order time discretization of (3.5) is

$$\mathbf{M_{ij}^{n+1}}\cdot\mathbf{u_{hij}^{n+1}}=\mathbf{M_{ij}^{n}}\cdot\mathbf{u_{hij}^{n}}+\Delta t^{n}L(U_h^n).\tag{3.10}$$

For the second-order spatial discretization, the corresponding time discretization uses the second-order RK method. We know the value of each variable at the nth time step. The time marching is as follows.

Step 1: Using the method in Section 3.3, we get the vertex velocity $(u_{ij}^{(1)},v_{ij}^{(1)})$, then

$$x_{ij}^{(1)}=x_{ij}^{n}+u_{ij}^{(1)}\Delta t^{n},\quad y_{ij}^{(1)}=y_{ij}^{n}+v_{ij}^{(1)}\Delta t^{n};\quad i=1,\cdots,M+1,\ \ j=1,\cdots,N+1.$$

Through the four vertices $\{(x_{i,j}^{(1)},y_{i,j}^{(1)}),\ (x_{i+1,j}^{(1)},y_{i+1,j}^{(1)}),\ (x_{i+1,j+1}^{(1)},y_{i+1,j+1}^{(1)}),\ (x_{i,j+1}^{(1)},y_{i,j+1}^{(1)})\}$ of $W_{ij}^{(1)}$, we can get $(x_c^{(1)},y_c^{(1)})$, $\Delta x^{(1)}$ and $\Delta y^{(1)}$. Then

$$B_{1ij}^{(1)}(x,y)=1,\quad B_{2ij}^{(1)}(x,y)=\frac{x-x_c^{(1)}}{\Delta x^{(1)}},\quad B_{3ij}^{(1)}(x,y)=\frac{y-y_c^{(1)}}{\Delta y^{(1)}},$$

$$
\begin{aligned}
&\mathbf{M}^{(1)}_{\mathbf{ij}}=\int_{W^{(1)}_{ij}} B^{(1)}_{lij}B^{(1)}_{mij}dW^{(1)}_{ij},\\
&\mathbf{M}^{(1)}_{\mathbf{ij}}\cdot\mathbf{u}^{(1)}_{\mathbf{hij}}=\mathbf{M}^{\mathbf{n}}_{\mathbf{ij}}\cdot\mathbf{u}^{\mathbf{n}}_{\mathbf{hij}}+\Delta t^n L(U^n_h).
\end{aligned}
$$

Step 2: Using variables obtained from Step 1 and the method in Section 3.3, we get the vertex velocity (u^n_{ij},v^n_{ij}), then

$$x^{n+1}_{ij}=\frac{1}{2}x^n_{ij}+\frac{1}{2}(x^{(1)}_{ij}+u^n_{ij}\Delta t^n),\quad y^{n+1}_{ij}=\frac{1}{2}y^n_{ij}+\frac{1}{2}(y^{(1)}_{ij}+v^n_{ij}\Delta t^n),\quad i=1,\cdots,M+1,\ \ j=1,\cdots,N+1.$$

Through the four vertices $\{(x^{n+1}_{i,j},y^{n+1}_{i,j}),\ (x^{n+1}_{i+1,j},y^{n+1}_{i+1,j}),\ (x^{n+1}_{i+1,j+1},y^{n+1}_{i+1,j+1}),\ (x^{n+1}_{i,j+1},y^{n+1}_{i,j+1})\}$ of W^{n+1}_{ij}, we can get (x^{n+1}_c,y^{n+1}_c), Δx^{n+1} and Δy^{n+1}. Then

$$
\begin{aligned}
&B^{n+1}_{1ij}(x,y)=1,\quad B^{n+1}_{2ij}(x,y)=\frac{x-x^{n+1}_c}{\Delta x^{n+1}},\quad B^{n+1}_{3ij}(x,y)=\frac{y-y^{n+1}_c}{\Delta y^{n+1}},\\
&\mathbf{M}^{\mathbf{n+1}}_{\mathbf{ij}}=\int_{W_{ij}} B^{n+1}_{lij}B^{n+1}_{mij}dW_{ij},\quad (l=1,2,3;\ m=1,2,3),\\
&\mathbf{M}^{\mathbf{n+1}}_{\mathbf{ij}}\cdot\mathbf{u}^{\mathbf{n+1}}_{\mathbf{hij}}=\frac{1}{2}\mathbf{M}^{\mathbf{n}}_{\mathbf{ij}}\cdot\mathbf{u}^{\mathbf{n}}_{\mathbf{hij}}+\frac{1}{2}(\mathbf{M}^{(1)}_{\mathbf{ij}}\cdot\mathbf{u}^{(1)}_{\mathbf{hij}}+\Delta tL(U^{(1)}_h)).
\end{aligned}
$$

The time step Δt^n is chosen as follows

$$\Delta t^n=\lambda\cdot\min_{i=1,\cdots,M;j=1,\cdots,N}(\Delta l^n_{ij}/c^n_{ij}), \tag{3.11}$$

where Δl^n_{ij} is the shortest edge length of the cell W_{ij}, and c^n_{ij} is the sound speed within this cell. The Courant number λ in the following tests is set to be 0.1 unless otherwise stated.

3.5 Boundary conditions

Usually boundary conditions are expected to provide numerical fluxes at the boundaries. We add some fictitious cells adjacent to the boundary real cells, and give the values of variables in the fictitious cells according to the boundary conditions, then we can compute the boundary fluxes as done for the interior cells. In this section, we prescribe fictitious data values in the fictitious cells, and the variables with the superscripts of f and r represent the values of the corresponding variables in fictitious cells and the real cells, respectively. Here we only consider two types of boundaries: prescribed normal velocity and transmissive boundaries.

Prescribed normal velocity. Let w^*_n be the value of the prescribed normal velocity along the boundary edge. Let $(w^f_n,w^f_t),(w^r_n,w^r_t)$ be the normal and tangential velocity along the boundary edge for the fictitious cell and the real cell, respectively. Then we set

$$w^f_n=2w^*_n-w^r_n,\quad w^f_t=w^r_t.$$

Thus we can get the x-velocity and the y-velocity:

$$u_x^f = w_n^f n_x - w_t^f n_y, \quad u_y^f = w_n^f n_y + w_t^f n_x,$$

where (n_x, n_y) is the unit normal vector of the fictitious cell. We also set

$$\rho^f = \rho^r, \quad e^f = e^r,$$

where ρ^f and ρ^r are the density, e^f and e^r are the specific internal energy. Then we can get the specific total energy

$$E^f = e^f + \frac{1}{2} u_x^f u_x^f + \frac{1}{2} u_y^f u_y^f.$$

Thus we get the conserved variables $(\rho^f, \rho^f u_x^f, \rho^f u_y^f, \rho^f E^f)$.

Transmissive boundaries. For a transmissive boundary we give the boundary condition as follows:

$$\rho^f = \rho^r, \quad u_x^f = u_x^r, \quad u_y^f = u_y^r, \quad e^f = e^r.$$

3.6 The local slope limiting

In the case of piecewise-constant approximations, the artificial viscosity is introduced by the numerical flux which is enough to keep the stability. But for high-order piecewise polynomial approximations, the influence of the numerical fluxes can not be enough to guarantee the absence of spurious oscillations of the scheme. To enhance the stability of the scheme and eliminate possible spurious oscillations in the approximate solution, we introduce a local slope limiting developed in [21].

The slope limiting is performed on u_{ij2} and u_{ij3} in (3.4). By using the differences of the means. For a scalar equation, u_{ij2} would be limited by

$$\tilde{u}_{ij2} = \bar{m}(u_{ij2}, \bar{u}_{i+1,j} - \bar{u}_{i,j}, \bar{u}_{i,j} - \bar{u}_{i-1,j}), \tag{3.12}$$

where the function $\bar{m}$ is the TVB corrected *minmod* function defined by

$$\bar{m}(a_1, a_2, \cdots, a_n) = \begin{cases} a_1, & \text{if } |a_1| \le H \cdot \Delta x^2, \\ m(a_1, a_2, \cdots, a_n), & \text{else,} \end{cases} \tag{3.13}$$

with the *minmod* function

$$m(a_1, a_2, \cdots, a_n) = \begin{cases} s \cdot \min\limits_i |a_i|, & \text{if } s = sign(a_1) = \cdots = sign(a_n), \\ 0, & \text{else.} \end{cases}$$

For an estimate of the TVB constant H in (3.13), see [15]. Similarly, u_{ij3} is limited by

$$\tilde{u}_{ij3} = \bar{m}(u_{ij3}, \bar{u}_{i,j+1} - \bar{u}_{i,j}, \bar{u}_{i,j} - \bar{u}_{i,j-1}), \tag{3.14}$$

just with a change of Δx to Δy in (3.13).

For the system, when we obtain the physical variables and the moving control volume W^{n+1} at the time $t=t^{n+1}$, we perform the limiting in the local characteristic variables at this fixed time $t=t^{n+1}$ and this fixed fluid region W^{n+1} (see [21]). Here we just give a simple description.

The Euler equations (2.7) can be rewritten as

$$U_t+f(U)_x+g(U)_y=0,$$

where

$$\begin{aligned}f(U)&=(\rho u,\rho u^2+p,\rho uv,(\rho E+p)u)^t,\\ g(U)&=(\rho v,\rho uv,\rho v^2+p,(\rho E+p)v)^t.\end{aligned}$$

To limit the vector U_{ij2} in the element W_{ij}^{n+1}, we proceed as follows:

(1) Find the matrix R and its inverse R^{-1} which diagonalize the Jacobian evaluated at the mean in the element W_{ij}^{n+1} in the x-direction,

$$R^{-1}\frac{\partial f(\bar{U}_{ij})}{\partial U}R=\Lambda,$$

where Λ is a diagonal matrix containing the eigenvalues of the Jacobian.

(2) Transform all quantities needed for the limiting, i.e., the three vectors U_{ij2}, $\bar{U}_{i+1,j}-\bar{U}_{i,j}$, and $\bar{U}_{i,j}-\bar{U}_{i-1,j}$, to the characteristic fields. This is achieved by left-multiplying these three vectors by R^{-1}.

(3) Apply the scalar limiter (3.12) to each of the components of the transformed vectors.

(4). The result is transformed back to the original space by left multiplying R on the left.

4 Numerical results

It is much more difficult to simulate a 2D problem than to simulate a 1D one in the Lagrangian framework, mainly because of the mesh distortion in multi-dimensions. In this section, we present several tests in order to validate our numerical scheme. We use the L-F flux and the HLLC flux in our scheme. Their results are similar, so we mainly show the results obtained by using L-F flux.

4.1 Accuracy test

First we compute two problems to check the accuracy of our schemes.

Example 4.1 (The two-dimensional low-density flow problem)**.** The initial condition is set to be

$$\begin{aligned}&\rho_0(x,y,0)=1+0.99\cdot\sin(2\pi(x+y)),\quad u_0(x,y,0)=1,\\&v_0(x,y,0)=1,\quad p_0(x,y,0)=1.\end{aligned}$$

The domain is taken as $[0,1]\times[0,1]$, and periodic boundary conditions in both directions are used. The exact solution is

$$\begin{aligned}&\rho(x,y,t)=1+0.99\cdot\sin(2\pi((x+y-2t))),\quad u(x,y,t)=1,\\&v(x,y,t)=1,\quad p(x,y,t)=1.\end{aligned}$$

The errors and the convergence order of our first, second-order Lagrangian schemes with L-F flux at $t=0.1$ are listed in Tables 1 and 2, respectively. Through the convergence results, we can see the desired first and second-order accuracy.

Table 1: The density results of first-order scheme for the low-density problem.

P^0	L_1		L_2		L_∞	
Grids	error	order	error	order	error	order
20×20	0.15808		0.17768		0.31690	
40×40	8.599E-002	0.8784	9.917E-002	0.8413	0.201695	0.6518
80×80	4.5288E-002	0.9251	5.4712E-002	0.8581	0.126958	0.6678
160×160	2.3411E-002	0.9519	3.0040E-002	0.8649	7.9326E-002	0.6784

Table 2: The density results of second-order scheme for the low-density problem.

P^1	L_1		L_2		L_∞	
Grids	error	order	error	order	error	order
20×20	8.7791E-003		1.1768E-002		3.2857E-002	
40×40	1.9441E-003	2.1750	3.1439E-003	1.9043	1.1317E-002	1.5377
80×80	4.5108E-004	2.1076	7.8545E-004	2.0010	3.4939E-003	1.6955
160×160	9.8273E-005	2.1985	1.7019E-004	2.2063	8.9011E-004	1.9728

Example 4.2 (The two-dimensional periodic vortex problem)**.** The periodic vortex problem is described as follows: The mean flow is $\rho=1$, $p=1$ and $(u,v)=(1,1)$ (diagonal flow). We add to this mean flow an isentropic vortex perturbation in (u,v) and the temperature $T=\frac{p}{\rho}$, no perturbation in the entropy $S=\frac{p}{\rho^\gamma}$, that is

$$\begin{aligned}&(\delta u,\delta v)=\frac{\varepsilon}{2\pi}e^{0.5(1-r^2)}(-\bar{y},\bar{x}),\\&\delta T=-\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{(1-r^2)},\quad \delta S=0,\end{aligned}$$

Table 3: The density results of first-order scheme for the vortex problem.

P^0	L_1		L_2		L_∞	
Grids	error	order	error	order	error	order
20×20	1.2260E-002		3.2825E-002		0.2838	
40×40	7.4237E-003	0.7237	2.0169E-002	0.7026	0.1831	0.6323
80×80	4.1550E-003	0.8373	1.1298E-002	0.8359	0.1006	0.8640
160×160	2.2127E-003	0.9089	6.0107E-003	0.9105	5.3096E-002	0.9219

Table 4: The density results of second-order scheme for the vortex problem.

P^1	L_1		L_2		L_∞	
Grids	error	order	error	order	error	order
20×20	3.3967E-003		8.3551E-003		8.2688E-002	
40×40	7.8493E-004	2.1135	1.8629E-003	2.1650	2.3617E-002	1.8078
80×80	1.8907E-004	2.0536	4.4571E-004	2.0634	5.5510E-003	2.0890
160×160	4.6594E-005	2.0207	1.0968E-004	2.0227	1.3550E-003	2.0343

where $(\bar{x},\bar{y})=(x-5,y-5)$, $r^2=\bar{x}^2+\bar{y}^2$, and the vortex strength is $\varepsilon=5$.

The domain is taken as $[0,10]\times[0,10]$, and periodic boundary conditions in both directions are used. It can be readily verified that the Euler equations with the above initial conditions admit an exact solution which is convected with speed $(1,1)$ in the diagonal direction. The errors and the convergence order of our first, second-order Lagrangian schemes with L-F flux at $t=1.0$ are listed in Tables 3 and 4, respectively. The initial mesh and the mesh at $t=1.0$ are displayed in Fig. 1. Through the convergence results and Fig. 1, we can see the desired first and second-order accuracy on moving and distorted Lagrangian meshes.

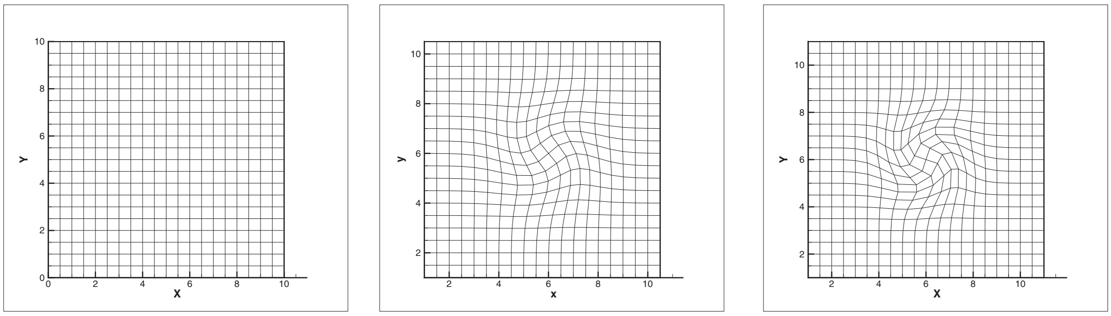

Figure 1: The mesh of the vortex problem. Left: initial mesh; Middle: the mesh of first-order accuracy at $t=1.0$; Right: the mesh of second-order accuracy at $t=1.0$.

4.2 Non-oscillatory tests

We compute five problems to check the non-oscillatory property of our scheme.

Example 4.3 (Two-dimensional Sod's shock tube problem [12])**.** This problem can assess the reasonableness and non-oscillatory property of our schemes. Its initial condition is as follows

$$
\begin{aligned}
&(\rho_L,u_L,v_L,p_L)=(1.0,0,0,1.0), \quad x\le 0.5;\\
&(\rho_R,u_R,v_R,p_R)=(0.125,0,0,0.1), \quad x>0.5.
\end{aligned}
$$

The domain is taken as $[0,1]\times[0,1]$. The problem can be interpreted as the two-dimension form of one-dimensional Sod's shock tube problem. Let nx and ny be the number of cells in the x- and y-directions. We set $nx=200$ and $ny=10$. In Fig. 2, we present the results by using the L-F flux at $t=0.2$, $y=0.5$, and we plot the piece-wise straight line connecting the two ends of each cell in x-direction for the density and pressure results of the second scheme. From the results we can see that the second-order scheme satisfy the non-oscillatory property.

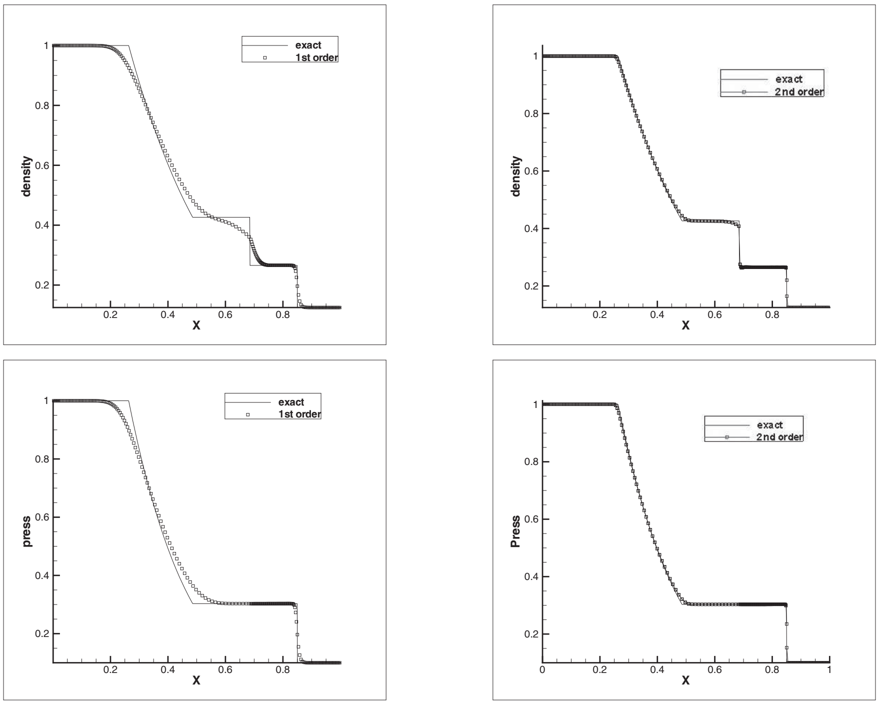

Figure 2: The results of the Sod problem. Left: first-order; Right: second-order. Top: density; Bottom: press.

Example 4.4 (The Saltzman problem [18])**.** This is a well known difficult test to validate the robustness of a Lagrangian scheme when the mesh is not aligned with the fluid flow.

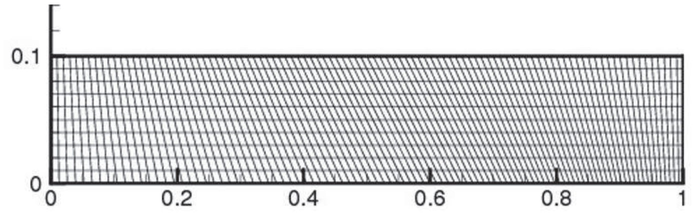

Figure 3: The initial computational mesh of the Saltzman problem.

The problem consists of a rectangular box whose left end is a piston. The piston moves into the box with a constant velocity of 1.0. The one-dimensional symmetry is broken by the initial mesh composed of 100×10 non-uniform cells defined as follows:

$$x(i,j)=(i-1)\Delta x+(11-j)\sin(0.01\pi(i-1))\Delta y,$$
$$y(i,j)=(j-1)\Delta y,$$

where $\Delta x=\Delta y=0.01$. The initial mesh is displayed in Fig. 3. Note that the initial mesh is deliberately distorted to set it as a more demanding test case. The working fluid is described by an ideal gas with $\gamma=5/3$. The initial conditions involve a stationary gas with a unity density and an internal energy of 10^{-4}. Reflective boundary conditions are used on the right, upper and lower boundaries. For this test case, it is necessary to first use a smaller Courant number in order to maintain stability. The Courant number λ is set to be 0.01 initially and returns to be 0.1 after $t=0.01$. Under such conditions, a first shock propagates toward the right with a velocity almost equal to 1.333; the post shock state is defined by a density of almost 4, a pressure of almost 1.333 and a velocity of almost 1. The results by using the HLLC flux are shown in Fig. 4 at time $t=0.75$. At this time, the shock

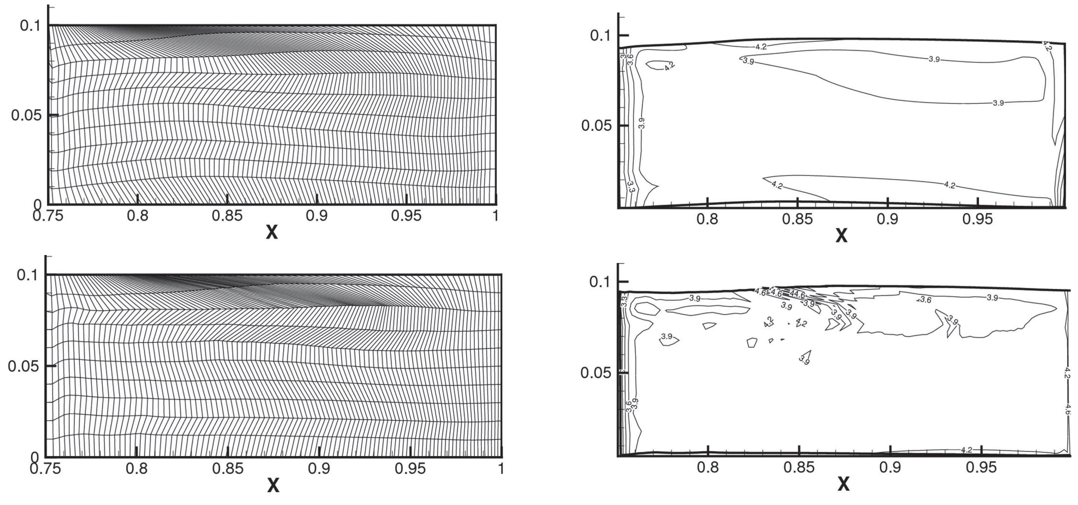

Figure 4: The results of the Saltzman problem at $t=0.75$. Left: mesh; Right: density contours. Top: first-order; Bottom: second-order.

wave hits the face $x=1$. We can observe that the numerical result of our second-order scheme is well except for the region near the up and bottom wall boundaries where the results are affected by the boundary conditions.

Example 4.5 (The Sedov blast wave problem [12]). We consider the propagation of a high intensity cylindrical shock wave generated by a strong explosion; see, for instance, [12]. The initial density is unity and the initial velocity is zero. The internal energy is zero except in the regional center where it has a value of 1. The domain is taken as $[-1.21,1.21]\times[-1.21,1.21]$, and the mesh is of 121×121 cells. All the boundary conditions are wall conditions. The analytical solution gives a shock at radius unity at time unity with a peak density of 6. Due to the symmetry of the problem, we only give the grid and the density contours of the first quadrant at top and middle of Fig. 5, respectively. But we display the result of the density in all the cells with respect to the cell center radius at the bottom of Fig. 5. In Fig. 6 we give the density profiles of the x axis of our second-order scheme and the same order scheme in [12]. From the direct comparison with the exact solution, we can clearly see that our second-order scheme is more precise than the scheme in [12].

Example 4.6 (Explosion problem [17]). The domain is taken as $[0,2]\times[0,2]$. Its initial condition is as follows

$$\begin{cases} (\rho_L,u_L,v_L,p_L)=(1.0,0,0,1.0), & (x-1)^2+(y-1)^2\leq 0.25, \\ (\rho_R,u_R,v_R,p_R)=(0.125,0,0,0.1), & (x-1)^2+(y-1)^2>0.25. \end{cases} \tag{4.1}$$

All the boundary conditions are wall conditions. We remark that in initializing the explosion problems, we modify the initial data on quadrilateral cells cutting the initial discontinuity, by assigning modified area-weighted values to the appropriate cells at the initial time $t=0$. This procedure avoids forming small amplitude waves created at early times by the staircase data. Fig. 7 shows the results of our scheme by using the L-F flux with 200×200 initially uniform cells at $t=0.25$. The up pictures in Fig. 9 show the results of the WAF (Weighted Average Flux) scheme developed in [17] with 200×200. In order to obtain the convergence of our scheme, we compute this problem by using the WAF scheme with 1000×1000 cells as the reference "exact" solution. Fig. 8 shows a comparison between the reference "exact" solution and our solutions with the three grids 100×100, 200×200 and 300×300 along the radial line which towards the origin $(1,1)$. We can observe the convergence of the numerical solutions toward the reference "exact" solution.

Example 4.7 (Implosion problem [17]). The domain is taken as $[0,2]\times[0,2]$. Its initial condition is as follows

$$\begin{cases} (\rho_L,u_L,v_L,p_L)=(0.125,0,0,0.1), & (x-1)^2+(y-1)^2\leq 0.25, \\ (\rho_R,u_R,v_R,p_R)=(1.0,0,0,1.0), & (x-1)^2+(y-1)^2>0.25. \end{cases} \tag{4.2}$$

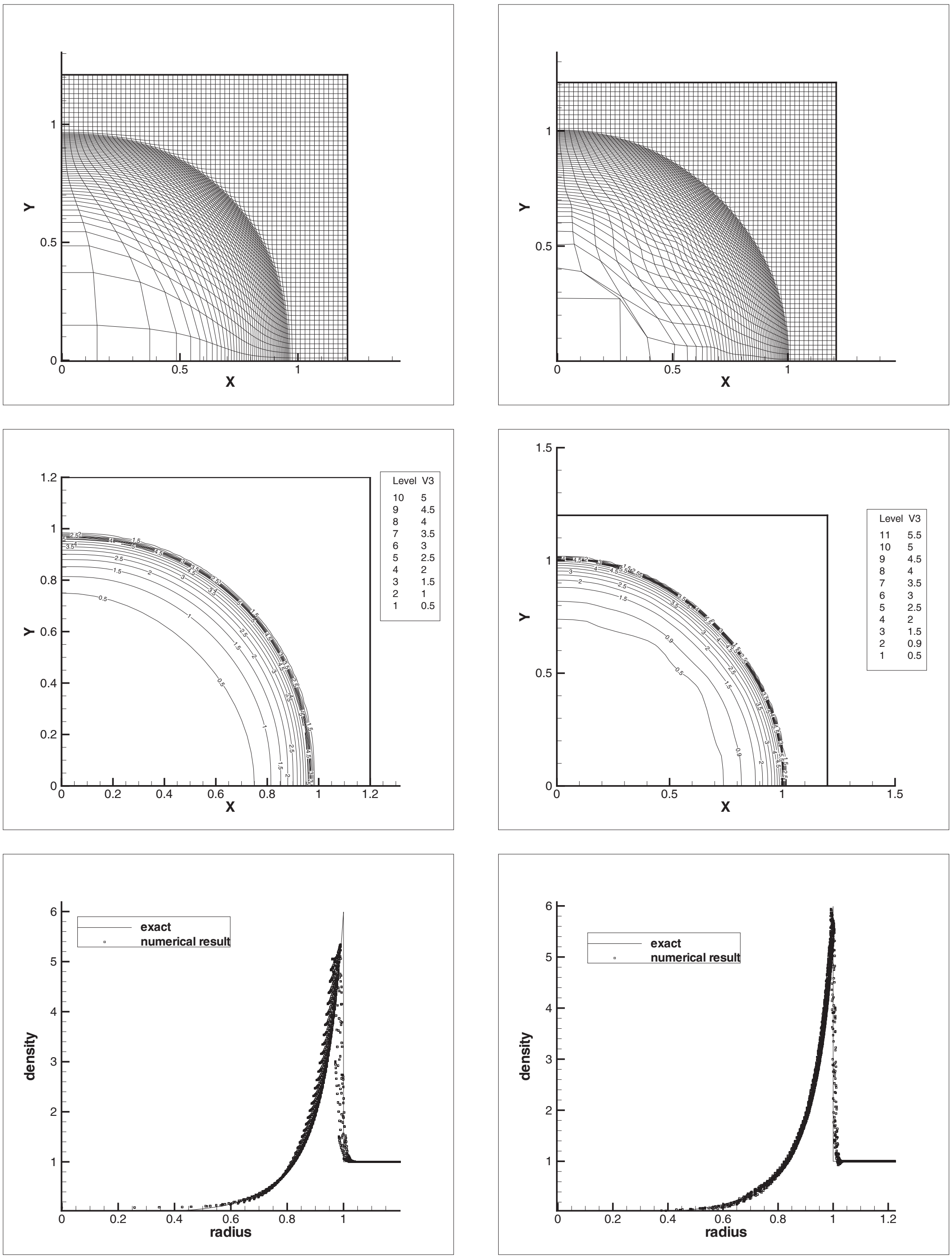

Figure 5: The results of the Sedov problem by using the L-F flux at t=1.0. Left: first-order; Right: second-order. Top: mesh; Middle: density contours; Bottom: density as a function of the radius in all the cells.

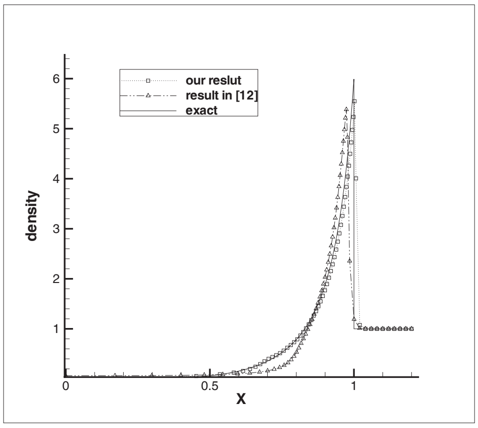

Figure 6: The density profiles along the x axis of our second-order scheme and that of the same order scheme in [12].

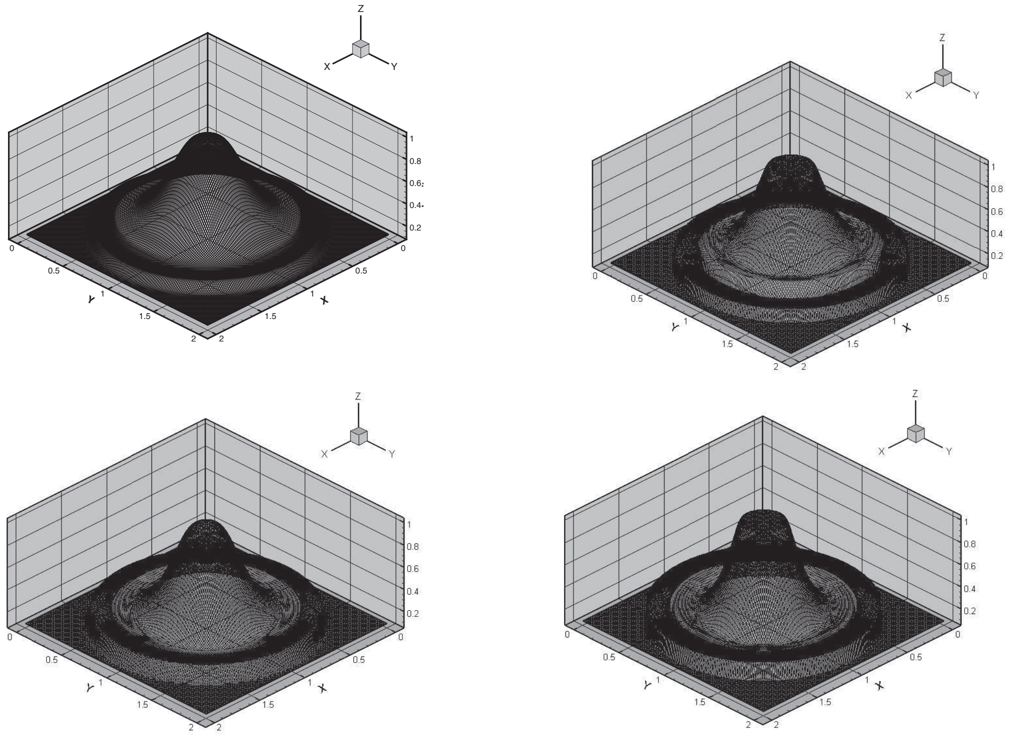

Figure 7: The results of our scheme for the explosion problem. Left: first-order; Right: second-order. Top: density; Bottom: press.

All the boundary conditions are wall conditions. We modify the initial data on quadrilateral cells cutting the initial discontinuity same as the method in Example 4.6. This problem is created by reversing the initial data of (4.1). Now shock focusing takes place

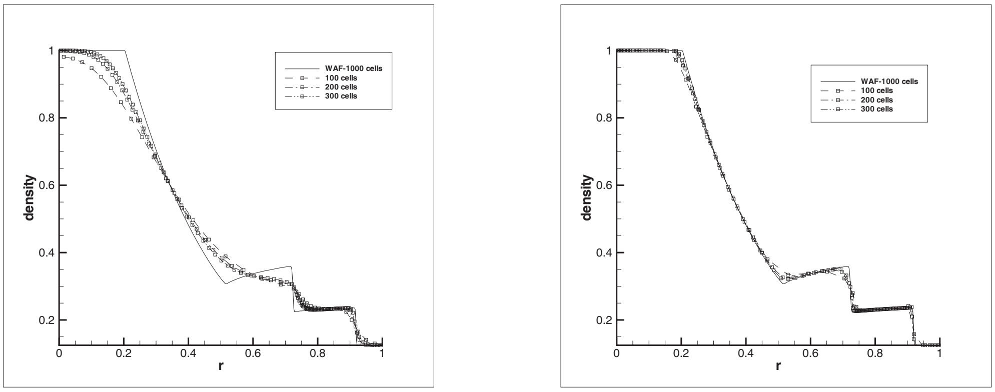

Figure 8: Comparison between the numerical solution and the reference "exact" solution of the explosion problem at $t=0.25$. Left: first-order; Right: second-order.

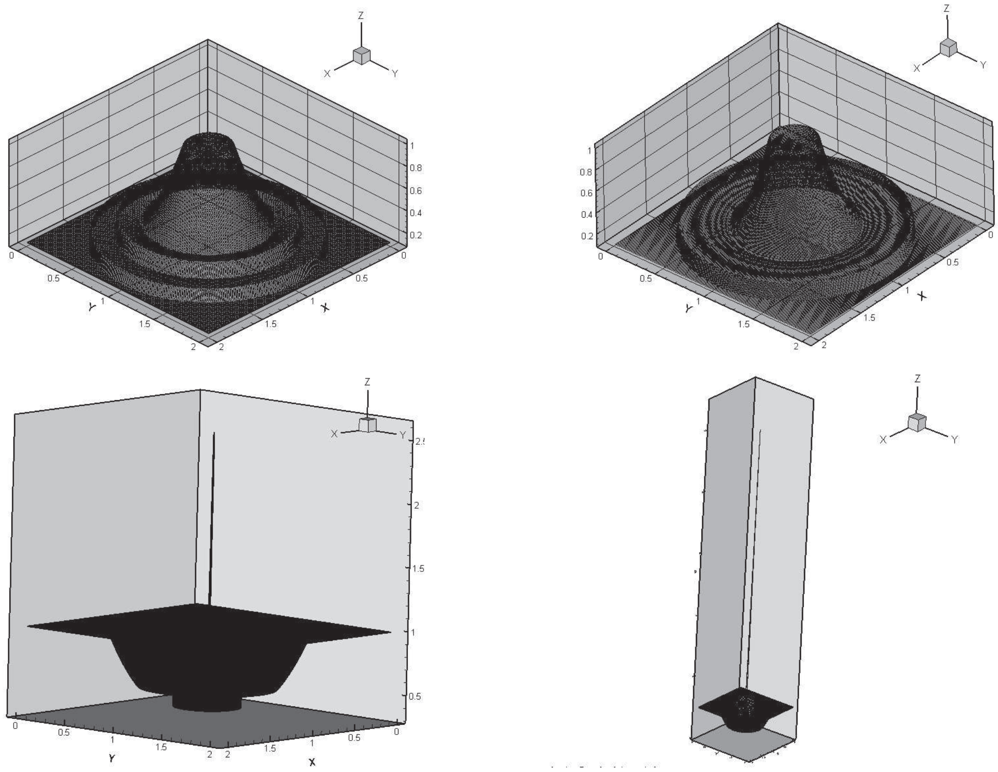

Figure 9: The results of the WAF scheme. Left: density; Right: press. Up: explosion problem. Down: implosion problem.

as part of the solution. Fig. 10 shows the results of our scheme by using the L-F flux with 200×200 initially uniform cells at $t=0.25$. The down pictures in Fig. 9 show the results of the WAF(Weighted Average Flux) scheme developed in [17] with 1000×1000 initially

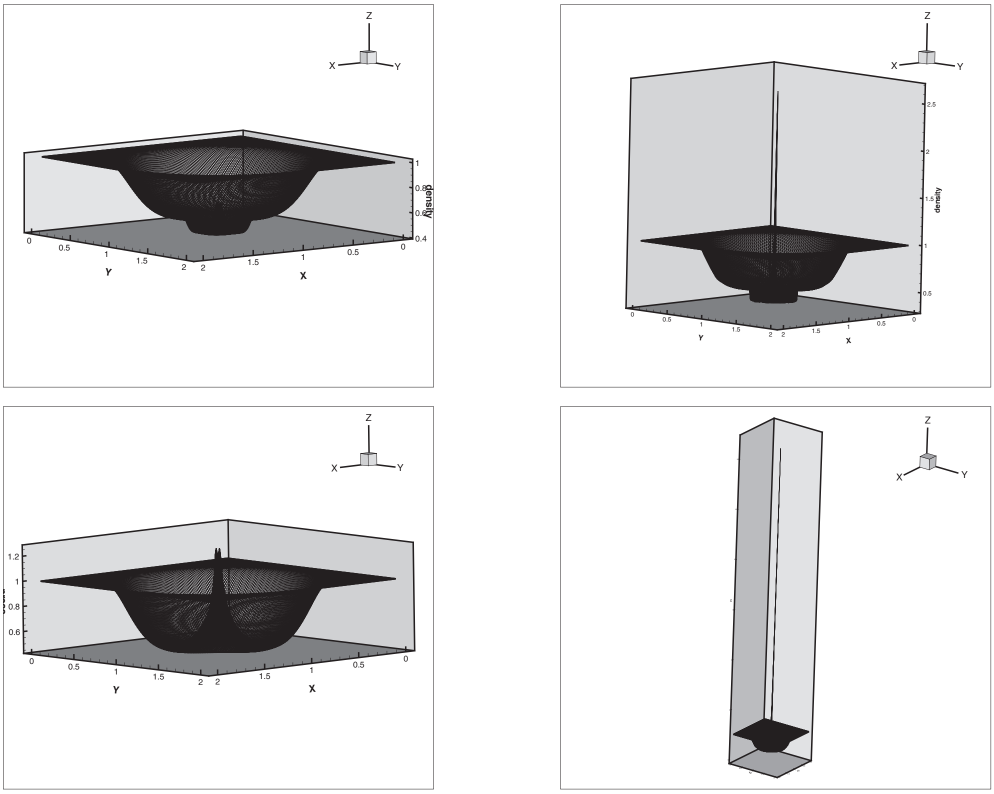

Figure 10: The results of our scheme for the implosion problem. Left: first-order; Right: second-order. Top: density; Bottom: press.

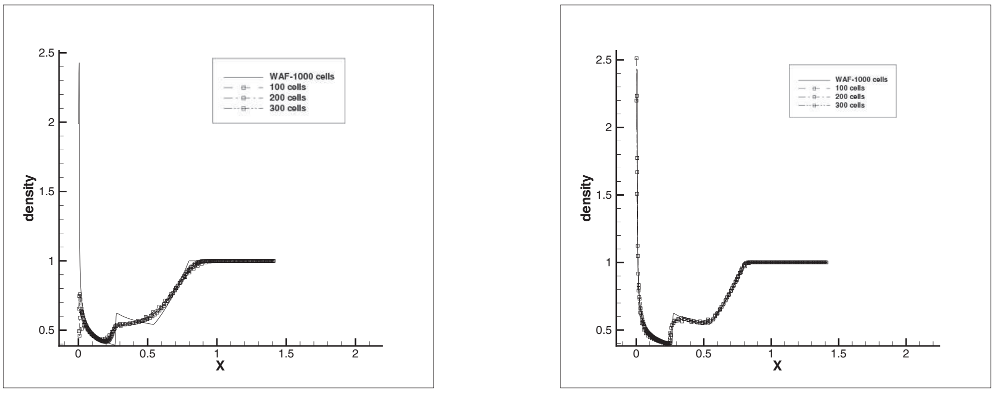

Figure 11: Comparison between the numerical solution and the reference "exact" solution of implosion problem at $t=0.25$. Left: first-order; Right: second-order.

uniform cells at $t=0.25$ as the reference "exact" solution. Fig. 11 shows a comparison between the reference "exact" solution and our solutions with the three grids 100×100, 200×200 and 300×300 along the radial line which towards the origin $(1,1)$. We can observe the convergence of the numerical solutions toward the reference “exact” solution, and the numerical result of the second-order scheme is obviously much better than that of the first-order scheme.

5 Concluding remarks

In this paper we have proposed a new Lagrangian type scheme to solve the compressible Euler equations. In this new scheme the system of equations is discretized by the RKDG method, and the mesh moves with the fluid flow. The Lax-Friedrichs numerical flux and the HLLC numerical flux are used in our scheme. A slope limiter is used to control spurious numerical oscillations near discontinuity. This new scheme is conservative for the mass, momentum and total energy. It can maintain second-order accuracy, avoids solving the geometrical part, and has free parameter for individual test cases. Several two-dimensional numerical examples have been presented to demonstrate the good performance of the scheme.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11171038), the Science Foundation of China Academy of Engineering Physics, China (Grant No. 2013A0202011) and the National Council for Scientific and Technological Development of Brazil (CNPq).

References

[1] E. J. Caramana, D. E. Burton and M. J. Shashkov et al., The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys., 146 (1998), 227-262.

[2] J. Von Neumann and R. D. Richtmyer, A method for the numerical calculations of hydrodynamical shocks, J. Appl. Phys., 21(1950), 232-238.

[3] J. C. Campbell and M. J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys., 172 (2001), 739-765.

[4] J. Cheng and C. W. Shu, A high-order ENO conservative Lagrangian type scheme for the compressible Euler equations, J. Comput. Phys., 227 (2007), 1567-1596.

[5] J. Cheng and C. W. Shu, A third order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations, Commun. Comput. Phys., 4 (2008), 1008-1024.

[6] P. H. Maire, R. Abgrall and J. Breil et al., A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM J. Sci. Comput., 29 (2007), 1781-1824.

[7] P. H. Maire and J. Breil, A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems, Int. J. Numer. Meth. Fluids, 56 (2007), 1417-1423.

[8] P. H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J.Comput. Phys., 228 (2009), 2391-2425.

[9] W. H. Hui, P. Y. Li and Z. W. Li, A unified coordinate system for solving the two-dimensional Euler equations, J. Comput. Phys., 153 (1999), 596-637.

[10] Z. P. Jia and S. D. Zhang, A new high-order discontinuous Galerkin spectral finite element method for Lagrangian gas dynamics in two dimensions, J. Comput. Phys., 230 (2011), 2496-2522.

[11] R. Loubère, J. Ovadia and R. Abgrall, A Lagrangian discontinuous Galerkin-type method on unstructed meshes to solve hydrodynamics problems, Int. J. Numer. Meth. Fluids, 44 (2004), 645-663.

[12] G. Z. Zhao, Variational iteration method and RKDG finite element method used in Lagrangian coordinate, Ph.D. Thesis, China Academy of Engineering Physics, 2011.

[13] H. Luo, J. D. Baum and R. Löhner, A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids, J. Comput. Phys., 227 (2008), 8875-8893.

[14] H. Luo, J. D. Baum and R. Löhner, On the computation of multi-material flows using ALE formulation, J. Comput. Phys., 194 (2004), 304-328.

[15] B. Cockburn and C. W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, Math. Comp., 52 (1989), 411-435.

[16] 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.

[17] E. F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Berlin, Springer-Verlag, 1999.

[18] J. K. Dukowicz, B. J. A. Meltz, Vorticity errors in multi-dimensional Lagrangian codes, J. Comput. Phys., 99 (1992), 115-134.

[19] B. Després and C. Mazeran, Lagrangian gas dynamics in two-dimensions and lagrangian systems, Arch. Ration. Mech. Anal., 178 (2005), 327-372.

[20] L. Landau and E. Lifchitz, M'ecanique des fluides, Mir, Moscow, 1989.

[21] 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.

[22] Z. P. Jia and X. J. Yu, A finite volume ALE method based on approximate Riemann solution, Chinese J. Comput. Phys., 24 (2007), 543-549.

[23] M. L. Wilkins, Calculation of elastic-plastic flow, in Methods in Computational Physics, Vol. 3, Academic Press, New York, 1964, pp. 211-263.

[24] F. Vilar, Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics, Comput. Fluids, 64 (2012), 64-73.

[25] P. H. Maire, A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids, Int. J. Numer. Meth. Fluids, 65 (2011), 1281-1294.

[26] P. H. Maire, A high-order one-step sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids, Comput. Fluids, 46 (2011), 341-347.