Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 1567–1596
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A high order ENO conservative Lagrangian type scheme for the compressible Euler equations

Juan Cheng [a], Chi-Wang Shu [b,*]

[a] Institute of Applied Physics and Computational Mathematics, Beijing 100088, China
[b] Division of Applied Mathematics, Brown University, Providence, RI 02912, United States

Received 10 July 2007; received in revised form 21 September 2007; accepted 24 September 2007
Available online 1 October 2007

Abstract

We develop a class of Lagrangian type schemes for solving the Euler equations of compressible gas dynamics both in the Cartesian and in the cylindrical coordinates. The schemes are based on high order essentially non-oscillatory (ENO) reconstruction. They are conservative for the density, momentum and total energy, can maintain formal high order accuracy both in space and time and can achieve at least uniformly second-order accuracy with moving and distorted Lagrangian meshes, are essentially non-oscillatory, and have no parameters to be tuned for individual test cases. One and two-dimensional numerical examples in the Cartesian and cylindrical coordinates are presented to demonstrate the performance of the schemes in terms of accuracy, resolution for discontinuities, and non-oscillatory properties.

Keywords: Lagrangian type scheme; High order accuracy; Conservative scheme; ENO reconstruction; Compressible Euler equations; ALE method

1. Introduction

In numerical simulations of multi-dimensional fluid flow, there are two typical choices: a Lagrangian framework, in which the mesh moves with the local fluid velocity, and an Eulerian framework, in which the fluid flows through a grid fixed in space. More generally, the motion of the grid can also be chosen arbitrarily, this method is called the Arbitrary Lagrangian–Eulerian method (ALE; cf. [14,2,21,16,25]). Most ALE algorithms consist of three phases, a Lagrangian phase in which the solution and the grid are updated, a rezoning phase in which the nodes of the computational grid are moved to a more optimal position and a remapping phase in which the Lagrangian solution is transferred to the new grid.

In this paper, we focus on computational hydrodynamic methods for the Euler equations where the mesh moves with the flow velocity. Such methods, which we refer to as Lagrangian type methods, imply the use of

* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: cheng_juan@iapcm.ac.cn (J. Cheng), shu@dam.brown.edu (C.-W. Shu).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.09.017

distorted or non-uniform meshes. Particular examples include the Lagrangian methods, or the ALE methods which contain a Lagrangian phase.

Pure Lagrangian methods, and certain ALE methods which can capture contact discontinuities sharply (see e.g. [19]), are widely used in many fields for multi-material flow simulations such as astrophysics and computational fluid dynamics (CFD). We will only consider single material in this paper, however the pure Lagrangian method and the ALE method based on the HLLC flux have the potential to be applied to multi-material flows. Comparing with Eulerian methods, Lagrangian type methods avoid or can reduce a source of numerical error due to the advection terms in the conservation equations. For this reason, Lagrangian type methods are frequently preferred in one-dimensional computations where mesh distortion plays no role. Even though the Euler equations are much simpler in the Lagrangian framework as they do not contain the advection terms, in two or more space dimensions they are actually more difficult to solve since the mesh moves with the fluid and can easily lose its quality. In the past years, many efforts have been made to develop Lagrangian type methods. Some algorithms are developed from the non-conservative form of the Euler equations, for example, those discussed in [23,3–5,18,37]. The other class of Lagrangian type algorithms starts from the conservative form of the Euler equations which usually can guarantee exact conservation. See for example [2,8,9,7,17,22,34,20] etc.

Most existing Lagrangian type schemes for the Euler equations have first or at most second-order accuracy. Among them many Lagrangian schemes of non-conservative form are only first-order accurate, because of a first-order error due to the non-conservative formulation of the momentum equation. On the other hand, some of the conservative Lagrangian type schemes apply the linear interpolation strategy to achieve second-order accuracy, meanwhile they usually use a flux limiter to control spurious oscillations which leads to a possible loss of this second-order accuracy at some special points such as smooth extrema and sonic points.

Essentially non-oscillatory (ENO) schemes, first introduced by Harten and Osher [13] and Harten et al. [12], can achieve uniformly high order accuracy with sharp, essentially non-oscillatory shock transitions. In the subsequent years, ENO schemes in the Eulerian formulation have accomplished successful applications in many fields especially with problems containing both shocks and complicated smooth flow structures, see for example [29]. Eulerian ENO schemes on unstructured meshes are developed in [1]. However, the application of the ENO methodology in the Lagrangian formulation does not seem to have been extensively explored.

In this paper, we develop a class of Lagrangian type schemes for solving the Euler equations which are based on the high order ENO reconstruction both in the Cartesian and in the cylindrical coordinates. The schemes are conservative for the density, momentum and total energy, can maintain formal high order accuracy both in space and time and can achieve at least uniformly second-order accuracy on moving and distorted Lagrangian meshes, are essentially non-oscillatory, and have no parameters to be tuned for individual test cases. They should also be generalizable to higher than second-order accuracy by using curved meshes, but this generalization is not carried out in this paper. Several one and two-dimensional numerical examples in the Cartesian and cylindrical coordinates are presented which demonstrate the good performance of the schemes both in purely Lagrangian and in ALE calculations.

An outline of the rest of this paper is as follows. In Section 2, we describe the individual steps of the ENO Lagrangian type scheme in one space dimension. In Section 3, we present one-dimensional numerical results. In Section 4, we extend the scheme to two space dimensions both in the Cartesian and in the cylindrical coordinates, while in Section 5 two-dimensional numerical examples are given to verify the performance of the ENO Lagrangian type method. In Section 6 we give concluding remarks.

2. High order ENO conservative Lagrangian type scheme – one space dimension

2.1. The compressible Euler equations in Lagrangian formulation

The Euler equations for unsteady compressible flow in the reference frame of a moving control volume can be expressed in integral form in the Cartesian coordinates as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega(t)}\mathbf{U}\mathrm{d}\Omega+\int_{\Gamma(t)}\mathbf{F}\mathrm{d}\Gamma=0, \tag{2.1}$$

where $\Omega(t)$ is the moving control volume enclosed by its boundary $\Gamma(t)$. The vector of the conserved variables $\mathbf{U}$ and the flux vector $\mathbf{F}$ are given by

$$\mathbf{U}=\begin{pmatrix}\rho\\ \mathbf{M}\\ E\end{pmatrix},\quad \mathbf{F}=\begin{pmatrix}(\mathbf{u}-\dot{\mathbf{x}})\cdot\mathbf{n}\rho\\ (\mathbf{u}-\dot{\mathbf{x}})\cdot\mathbf{n}\mathbf{M}+p\cdot\mathbf{n}\\ (\mathbf{u}-\dot{\mathbf{x}})\cdot\mathbf{n}E+p\mathbf{u}\cdot\mathbf{n}\end{pmatrix}, \tag{2.2}$$

where ρ is the density, $\mathbf{u}$ is the velocity, $\mathbf{M}=\rho\mathbf{u}$ is the momentum, E is the total energy and p is the pressure, $\dot{\mathbf{x}}$ is the velocity of the control volume boundary $\Gamma(t)$, $\mathbf{n}$ denotes the unit outward normal to $\Gamma(t)$. The system (2.1) represents the conservation of mass, momentum and energy.

The set of equations is completed by the addition of an equation of state (EOS) with the following general form

$$p=p(\rho,e), \tag{2.3}$$

where $e=\frac{E}{\rho}-\frac{1}{2}|\mathbf{u}|^2$ is the specific internal energy. Especially, if we consider the ideal gas, then the equation of state has a simpler form,

$$p=(\gamma-1)\rho e,$$

where γ is a constant representing the ratio of specific heat capacities of the fluid.

This paper focuses on solving the governing Eqs. (2.1) and (2.2) in a Lagrangian framework, in which it is assumed that $\dot{\mathbf{x}}=\mathbf{u}$, and the vectors $\mathbf{U}$ and $\mathbf{F}$ then take the simpler form

$$\mathbf{U}=\begin{pmatrix}\rho\\ \mathbf{M}\\ E\end{pmatrix},\quad \mathbf{F}=\begin{pmatrix}0\\ p\cdot\mathbf{n}\\ p\mathbf{u}\cdot\mathbf{n}\end{pmatrix}. \tag{2.4}$$

2.2. The ENO conservative Lagrangian type scheme in one space dimension

Here we develop a conservative Lagrangian type finite volume scheme on a non-staggered mesh. We solve the conserved variables such as density, momentum and total energy directly. We remark that many Lagrangian type schemes are defined on staggered meshes, where the density and total energy (or internal energy) are cell-centered and the velocity (or momentum) is vertex centered. However, our computational experiments indicate that the non-staggered mesh performs better when the momentum rather than the velocity is solved directly for the momentum equation. In order to save space we will not give the description and results on staggered meshes.

The spatial domain Ω is discretized into N computational cells $I_{i+1/2}=[x_i,x_{i+1}]$ of sizes $\Delta x_{i+1/2}=x_{i+1}-x_i$ with $i=1,\ldots,N$. For a given cell $I_{i+1/2}$, the location of the cell center is denoted by $x_{i+1/2}$. The fluid velocity u_i is defined at the vertex of the mesh. All variables except the velocity are stored at the cell center $x_{i+1/2}$ in the form of cell averages and this cell is their common control volume. For example, the values of the cell averages for cell $I_{i+1/2}$, denoted by $\bar{\rho}_{i+1/2}$, $\overline{M}_{i+1/2}$ and $\overline{E}_{i+1/2}$, are defined as follows:

$$\bar{\rho}_{i+1/2}=\frac{1}{\Delta x_{i+1/2}}\int_{I_{i+1/2}}\rho\mathrm{d}x,\quad \overline{M}_{i+1/2}=\frac{1}{\Delta x_{i+1/2}}\int_{I_{i+1/2}}M\mathrm{d}x,\quad \overline{E}_{i+1/2}=\frac{1}{\Delta x_{i+1/2}}\int_{I_{i+1/2}}E\mathrm{d}x.$$

2.2.1. Spatial discretization

We first formulate the semi-discrete finite volume scheme of the governing (2.1) and (2.4) as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\bar{\rho}_{i+1/2}\Delta x_{i+1/2}\\ \overline{M}_{i+1/2}\Delta x_{i+1/2}\\ \overline{E}_{i+1/2}\Delta x_{i+1/2}\end{pmatrix}=-\begin{pmatrix}\hat{f}_D(\mathbf{U}^-_{i+1},\mathbf{U}^+_{i+1})-\hat{f}_D(\mathbf{U}^-_i,\mathbf{U}^+_i)\\ \hat{f}_M(\mathbf{U}^-_{i+1},\mathbf{U}^+_{i+1})-\hat{f}_M(\mathbf{U}^-_i,\mathbf{U}^+_i)\\ \hat{f}_E(\mathbf{U}^-_{i+1},\mathbf{U}^+_{i+1})-\hat{f}_E(\mathbf{U}^-_i,\mathbf{U}^+_i)\end{pmatrix} \tag{2.5}$$

where $\hat{f}_D$ is the numerical flux of mass across the boundary of its control volume $I_{i+1/2}(t)$, which is consistent with the physical flux (2.4) in the sense that $\hat{f}_D(\mathbf{U},\mathbf{U}) = 0$, $\hat{f}_M$ is the numerical flux of momentum with $\hat{f}_M(\mathbf{U},\mathbf{U}) = p$, and $\hat{f}_E$ is the numerical flux of total energy with $\hat{f}_E(\mathbf{U},\mathbf{U}) = pu$. $\mathbf{U}_i^{\pm}$ and $\mathbf{U}_{i+1}^{\pm}$ represent the left and right values of $\mathbf{U}$ at the cell's boundary x_i and x_{i+1}, respectively.

The first step for establishing the scheme is to determine the fluxes $(\hat{f}_D, \hat{f}_M, \hat{f}_E)$, and the first stage of this step is to identify the values of the primitive variables on each side of the boundary, that is $\mathbf{U}_i^{\pm}$, $i = 1,\ldots,N$. The information we have is the cell average values of the conserved variables $\overline{\mathbf{U}}_{i+1/2} = (\bar{\rho}_{i+1/2}, \overline{M}_{i+1/2}, \overline{E}_{i+1/2})$, thus we will have to reconstruct these variables. The simplest reconstruction is to assume that all the variables are piecewise constant, and equal to the given cell averages for each cell. Then we will have $\rho_i^- = \bar{\rho}_{i-1/2}$, $\rho_i^+ = \bar{\rho}_{i+1/2}$ etc., where ρ_i^- and ρ_i^+ are the values of density at the left side and the right side of the cell's boundary x_i. This reconstruction is very simple, but is only first-order accurate.

To obtain uniformly second or higher order accurate schemes, in this paper we will use the ENO idea [12] to reconstruct polynomial functions on each $I_{i+1/2}$ by using the information of the cell $I_{i+1/2}$ and its neighbors, such that they are second or higher order accurate approximations to the functions $\rho(x)$, $M(x)$ and $E(x)$ etc., on $I_{i+1/2}$. The procedure allows us to obtain arbitrary high order accurate approximation by a proper reconstruction. For simplicity, in this paper we will only discuss second- and third-order accurate schemes by performing the second and third-order accurate reconstructions. It is easy to extend the procedure to a higher order reconstruction.

The method of local characteristic decomposition is used in the procedure of the ENO reconstruction. We refer to [33] for the details of the Roe-type characteristic decomposition that we have used in this paper. As to the details of how to conservatively reconstruct a polynomial by the ENO idea in each cell, we refer to [12]. The approximate values of each conserved variable at both sides of the cell's boundary are obtained from its reconstructed polynomial. Finally, we obtain the values of the density ρ_i^-, ρ_i^+, the momentum M_i^-, M_i^+ and the total energy E_i^-, E_i^+ at the left side and the right side of the cell's boundary x_i, respectively.

Next, we will compute the fluxes given the primitive states at each side of a control volume's boundary. In this paper, we use the following four typical numerical fluxes:

1) The Godunov flux,
2) The Dukowicz flux,
3) The L–F (Lax–Friedrichs) flux,
4) The HLLC (Harten–Lax-van Leer contact wave) flux.

In the following, we will describe the implementation of the four fluxes in our Lagrangian type schemes. We have also tested a few other numerical fluxes, such as the Roe flux with an entropy fix. We will however not present these results to save space.

1. The Godunov flux

To solve the exact Riemann problem at the cell's boundary x_i, we need to know the left and right states at the boundary. We would like to note that in a Lagrangian scheme the velocity at each side of x_i used here should be the relative fluid velocity, that is, $u_i^- - \dot{x}_i$ and $u_i^+ - \dot{x}_i$, where $u_i^- = M_i^-/\rho_i^-$, $u_i^+ = M_i^+/\rho_i^+$ and $\dot{x}_i$ is the cell boundary's reference moving velocity which can be numerically determined as the Roe average $\frac{\sqrt{\rho_i^-}u_i^- + \sqrt{\rho_i^+}u_i^+}{\sqrt{\rho_i^-}+\sqrt{\rho_i^+}}$. The pressure values at the two sides of the vertex x_i are of the form $p_i^- = (\gamma-1)[E_i^- - \frac{1}{2}(M_i^-)^2/\rho_i^-]$, $p_i^+ = (\gamma-1)[E_i^+ - \frac{1}{2}(M_i^+)^2/\rho_i^+]$ if the ideal gas is considered. With the left state $\{\rho_i^-, u_i^- - \dot{x}_i, p_i^-\}$ and the right state $\{\rho_i^+, u_i^+ - \dot{x}_i, p_i^+\}$ at x_i, we can now obtain the pressure p_i^* and the relative velocity u_i' at x_i by the Riemann solver. The absolute velocity u_i^* at x_i should be $u_i' + \dot{x}_i$. Thus the fluxes $\hat{f}_D$, $\hat{f}_M$ and $\hat{f}_E$ in Scheme (2.5) have the following form

$$\begin{cases} \hat{f}_D(\mathbf{U}_i^-, \mathbf{U}_i^+) = 0 \\ \hat{f}_M(\mathbf{U}_i^-, \mathbf{U}_i^+) = p_i^* \\ \hat{f}_E(\mathbf{U}_i^-, \mathbf{U}_i^+) = p_i^* u_i^*. \end{cases} \tag{2.6}$$

2. The Dukowicz flux

The exact Godunov flux needs an iterative procedure to compute, hence the computational complexity and cost are very high, especially for materials with complex equations of state. Dukowicz [8] developed a simplified and non-iterative approximate Riemann solver to overcome this difficulty.

In this two-shock approximation, the velocity u_i^* at the cell boundary x_i is obtained by solving the following semi-quadratic equation,

$$\rho_i^+ A_i^+ |u_i^* - u_{\min}^*|(u_i^* - u_{\min}^*) + \rho_i^- A_i^- |u_i^* - u_{\max}^*|(u_i^* - u_{\max}^*) + p^{+,*} - p^{-,*} = 0 \tag{2.7}$$

where

$$\begin{aligned} &u_{\min}^* = u_i^+ - c_i^+/2A_i^+, \quad u_{\max}^* = u_i^- + c_i^-/2A_i^-; \\ &p^{+,*} = p_i^+ - \frac{1}{4}\rho_i^+ (c_i^+)^2/A_i^+, \quad p^{-,*} = p_i^- - \frac{1}{4}\rho_i^- (c_i^-)^2/A_i^-. \end{aligned} \tag{2.8}$$

Here $c_i^\pm$ are the left and right values of the sound speed at x_i and $A^\pm$ are parameters directly related to the shock density ratio in the limit of strong shocks. We refer to [8] for the details of the definition of $A^\pm$. In particular, for an ideal gas, $A_i^\pm = (\gamma+1)/2$.

After we have calculated the velocity u_i^*, the pressure p_i^* at x_i is easily obtained by the following equation,

$$p_i^* = \frac{1}{2}(p^{-,*} + p^{+,*}) + \frac{1}{2}\rho_i^+ A_i^+ |u_i^* - u_{\min}^*|(u_i^* - u_{\min}^*) - \frac{1}{2}\rho_i^- A_i^- |u_i^* - u_{\max}^*|(u_i^* - u_{\max}^*) \tag{2.9}$$

If p_i^* is found to be negative which predicts cavitation, then we set $p_i^* = 0$.

Substituting u_i^* and p_i^* obtained from 2.7, 2.8 and 2.9 into (2.6), we then obtain the fluxes $\hat{f}_D$, $\hat{f}_M$ and $\hat{f}_E$.

3. The L–F (Lax–Friedrichs) flux

The Lax–Friedrichs flux is given by

$$\begin{cases} \hat{f}_D(\mathbf{U}_i^-, \mathbf{U}_i^+) &= \frac{1}{2}[0 - \alpha_i(\rho_i^+ - \rho_i^-)] \\ \hat{f}_M(\mathbf{U}_i^-, \mathbf{U}_i^+) &= \frac{1}{2}[(p_i^- + p_i^+) - \alpha_i(M_i^+ - M_i^-)] \\ \hat{f}_E(\mathbf{U}_i^-, \mathbf{U}_i^+) &= \frac{1}{2}[(p_i^- u_i^- + p_i^+ u_i^+) - \alpha_i(E_i^+ - E_i^-)] \end{cases} \tag{2.10}$$

where α_i is taken as an upper bound for the eigenvalues of the Jacobian, here in the Lagrangian formulation, $\alpha_i = \max(c_i^-, c_i^+)$. In order to reduce the numerical viscosity, we perform the local characteristic decomposition and then choose a different α_i for each characteristic field based on the size of the corresponding eigenvalue, rather than using the same α_i for all components as in (2.10).

4. The HLLC flux

We use the version of the HLLC flux described in [19] (see also [36]) for the ALE formulation which is defined by

$$\widehat{\mathbf{F}}_i^{\mathrm{HLLC}} = \begin{cases} \widehat{\mathbf{F}}_i^-, & \text{if } S_i^- > 0, \\ \widehat{\mathbf{F}}(\mathbf{U}_i^*), & \text{if } S_i^- \leqslant 0 < S_M, \\ \widehat{\mathbf{F}}(\mathbf{U}_i^{**}), & \text{if } S_M \leqslant 0 < S_i^+, \\ \widehat{\mathbf{F}}_i^+, & \text{if } S_i^+ < 0, \end{cases} \tag{2.11}$$

where

$$\widehat{\mathbf{F}}_i^- = \begin{pmatrix} 0 \\ p_i^- \\ p_i^- u_i^- \end{pmatrix}, \quad \widehat{\mathbf{F}}_i^+ = \begin{pmatrix} 0 \\ p_i^+ \\ p_i^+ u_i^+ \end{pmatrix},$$

and

$$\mathbf{U}_i^* = \begin{pmatrix} \rho_i^* \\ M_i^* \\ E_i^* \end{pmatrix} = \frac{1}{S_i^- - S_M} \begin{pmatrix} (S_i^- - v_i^-)\rho_i^- \\ (S_i^- - v_i^-)M_i^- + (p^* - p_i^-) \\ (S_i^- - v_i^-)E_i^- - p_i^- v_i^- + p^* S_M \end{pmatrix}, \tag{2.12}$$

$$\mathbf{U}_i^{**} = \begin{pmatrix} \rho_i^{**} \\ M_i^{**} \\ E_i^{**} \end{pmatrix} = \frac{1}{S_i^+ - S_M} \begin{pmatrix} (S_i^+ - v_i^+)\rho_i^+ \\ (S_i^+ - v_i^+)M_i^+ + (p^* - p_i^+) \\ (S_i^+ - v_i^+)E_i^+ - p_i^+ v_i^+ + p^* S_M \end{pmatrix}, \tag{2.13}$$

$$\widehat{\mathbf{F}}(\mathbf{U}_i^*) = \begin{pmatrix} S_M \rho_i^* \\ S_M M_i^* + p^* \\ S_M E_i^* + (S_M + \dot{x}_i)p^* \end{pmatrix}, \quad \hat{\mathbf{F}}(\mathbf{U}_i^{**}) = \begin{pmatrix} S_M \rho_i^{**} \\ S_M M_i^{**} + p^* \\ S_M E_i^{**} + (S_M + \dot{x}_i)p^* \end{pmatrix}, \tag{2.14}$$

$$p^* = \rho_i^-(v_i^- - S_i^-)(v_i^- - S_M) + p_i^-, \tag{2.15}$$

and $v_i^- = u_i^- - \dot{x}_i$, $v_i^+ = u_i^+ - \dot{x}_i$. S_M is defined as

$$S_M = \frac{\rho_i^+ v_i^+ (S_i^+ - v_i^+) - \rho_i^- v_i^- (S_i^- - v_i^-) + p_i^- - p_i^+}{\rho_i^+ (S_i^+ - v_i^+) - \rho_i^- (S_i^- - v_i^-)}. \tag{2.16}$$

The signal velocities S_i^- and S_i^+ are defined as

$$S_i^- = \min[v_i^- - c_i^-, (u_i - \dot{x}_i) - c_i], \quad S_i^+ = \max[v_i^+ + c_i^+, (u_i - \dot{x}_i) + c_i], \tag{2.17}$$

where u_i and c_i are the Roe's average variables for the velocity and the speed of sound. Since we are considering the Euler equations in the Lagrangian formulation, here we have $\dot{x}_i = u_i$.

Each of the above approximate fluxes has its own special features. The Godunov flux solves exactly the Riemann problem at the cell boundary and thus it has the least numerical viscosity among all the first-order upwind fluxes. In particular, it has no numerical viscosity for the first equation hence it can maintain the mass of each cell unchanged during the time marching. But it also has the disadvantage of high computational cost. The Dukowicz flux shares the advantage of the Godunov flux in its zero viscosity for the first equation and small numerical viscosity for the other equations, but it has a much smaller computational cost. This will be the flux of our choice for our test cases. The L–F flux has relatively more numerical viscosity, but it has a very simple form and is more robust in applications. The HLLC flux's viscosity and cost are between the Godunov flux and the L–F flux. Since its viscosity vanishes at the Lagrangian contact where $p^- = p^+$, $v^- = v^+ = 0$, it can resolve the contact discontinuities sharply within the Lagrangian method. In fact, it has good performance in many applications. We remark that both the L–F flux and the HLLC flux usually apply the numerical viscosity in all the equations including the mass equation, causing a possible change in the cell mass during the time evolution. We do, however, find in the numerical tests that the L–F flux and HLLC flux perform well on capturing the contact discontinuities in our Lagrangian type schemes. For some of the numerical examples, results with more than one numerical fluxes will be shown and compared. In the actual simulation, especially in the ALE calculation, we can choose the most suitable flux depending on the requirement of the individual problem.

2.2.2. The determination of the vertex velocity

In the Lagrangian formulation, the grid moves with the fluid velocity which is defined at the vertex, thus we would need to know the velocity at the vertex to move the grid. Since the velocity is a derived quantity, we would need to obtain it from the conserved variables. In the following we describe how to determine the vertex's velocity in our schemes.

For the Godunov flux and the Dukowicz flux, since we solve (exactly or approximately) the Riemann problem at each vertex as a cell's boundary, we naturally obtain the velocity by the Riemann solver there.

For the L–F flux and the HLLC flux, the velocity at the cell's vertex is defined as the Roe's average of velocities from both sides,

$$u_i = \frac{\sqrt{\rho_i^-}u_i^- + \sqrt{\rho_i^+}u_i^+}{\sqrt{\rho_i^-} + \sqrt{\rho_i^+}}. \tag{2.18}$$

2.2.3. Time discretization

The time marching for the semi-discrete scheme (2.5) is implemented by a class of TVD Runge–Kutta type methods [31]. Since the mesh changes with the time advancing in the Lagrangian simulation, the velocity, the

position of each vertex and the size of each cell need to be updated at each Runge–Kutta stage. Thus the form of the Runge–Kutta method (we take the third-order case here as an example) in our Lagrangian type schemes is as follows.

Step 1,

$$x_i^{(1)} = x_i^n + u_i^n \Delta t^n, \quad \Delta x_{i+1/2}^{(1)} = x_{i+1}^{(1)} - x_i^{(1)},$$

$$\overline{\mathbf{U}}_{i+1/2}^{(1)} \Delta x_{i+1/2}^{(1)} = \overline{\mathbf{U}}_{i+1/2}^n \Delta x_{i+1/2}^n + \Delta t^n \mathbf{L}(\overline{\mathbf{U}}_{i+1/2}^n);$$

Step 2,

$$x_i^{(2)} = \frac{3}{4} x_i^n + \frac{1}{4}[x_i^{(1)} + u_i^{(1)} \Delta t^n], \quad \Delta x_{i+1/2}^{(2)} = x_{i+1}^{(2)} - x_i^{(2)},$$

$$\overline{\mathbf{U}}_{i+1/2}^{(2)} \Delta x_{i+1/2}^{(2)} = \frac{3}{4} \overline{\mathbf{U}}_{i+1/2}^n \Delta x_{i+1/2}^n + \frac{1}{4}[\overline{\mathbf{U}}_{i+1/2}^{(1)} \Delta x_{i+1/2}^{(1)} + \Delta t^n \mathbf{L}(\overline{\mathbf{U}}_{i+1/2}^{(1)})];$$

Step 3,

$$x_i^{n+1} = \frac{1}{3} x_i^n + \frac{2}{3}[x_i^{(2)} + u_i^{(2)} \Delta t^n], \quad \Delta x_{i+1/2}^{n+1} = x_{i+1}^{n+1} - x_i^{n+1},$$

$$\overline{\mathbf{U}}_{i+1/2}^{n+1} \Delta x_{i+1/2}^{n+1} = \frac{1}{3} \overline{\mathbf{U}}_{i+1/2}^n \Delta x_{i+1/2}^n + \frac{2}{3}[\overline{\mathbf{U}}_{i+1/2}^{(2)} \Delta x_{i+1/2}^{(2)} + \Delta t^n \mathbf{L}(\overline{\mathbf{U}}_{i+1/2}^{(2)})],$$

where $\mathbf{L}$ is the numerical spatial operator representing the right hand of the scheme (2.5). Here 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.

Notice that such Runge–Kutta schemes are simple convex combinations of Euler forward time stepping, and are hence conservative and stable whenever the Euler forward time stepping is conservative and stable.

Consistently with the order of the spatial discretization in the scheme (2.5), we apply the Runge–Kutta method of the same order for its time marching.

At the end of this section, we list the explicit form of our first-order Lagrangian type scheme as an example,

$$\begin{pmatrix} \bar{\rho}_{i+1/2}^{n+1} \Delta x_{i+1/2}^{n+1} - \bar{\rho}_{i+1/2}^n \Delta x_{i+1/2}^n \\ \overline{M}_{i+1/2}^{n+1} \Delta x_{i+1/2}^{n+1} - \overline{M}_{i+1/2}^n \Delta x_{i+1/2}^n \\ \overline{E}_{i+1/2}^{n+1} \Delta x_{i+1/2}^{n+1} - \overline{E}_{i+1/2}^n \Delta x_{i+1/2}^n \end{pmatrix} = -\Delta t^n \begin{pmatrix} \hat{f}_D(\mathbf{U}_{i+1}^{n-}, \mathbf{U}_{i+1}^{n+}) - \hat{f}_D(\mathbf{U}_i^{n-}, \mathbf{U}_i^{n+}) \\ \hat{f}_M(\mathbf{U}_{i+1}^{n-}, \mathbf{U}_{i+1}^{n+}) - \hat{f}_M(\mathbf{U}_i^{n-}, \mathbf{U}_i^{n+}) \\ \hat{f}_E(\mathbf{U}_{i+1}^{n-}, \mathbf{U}_{i+1}^{n+}) - \hat{f}_E(\mathbf{U}_i^{n-}, \mathbf{U}_i^{n+}) \end{pmatrix}, \tag{2.19}$$

where the time step Δt^n is determined as

$$\Delta t^n = \lambda \min_{i=1,\ldots,N}(\Delta x_i^n / c_i^n),$$

with the Courant number λ chosen as $\lambda = 0.6$ in our computation.

3. Numerical results in one space dimension

In this section, we perform some numerical experiments in one space dimension. Purely Lagrangian computation and the ideal gas with $\gamma = 1.4$ are used to do the following tests unless otherwise stated. We mainly show the results obtained with the Dukowicz flux but we also show the results with the other fluxes (the Godunov flux, the L–F flux and the HLLC flux) for some test cases for comparison.

3.1. Accuracy test

We first test the accuracy of our schemes on a problem with smooth solutions. The initial condition we choose for the accuracy test is

$$\rho(x,0) = 2 + \sin(2\pi x), \quad u(x,0) = 1 + 0.1\sin(2\pi x), \quad p(x,0) = 1, \quad x \in [0,1],$$

with a periodic boundary condition. Since we do not know the exact solution, we take the numerical results by using the fifth-order Eulerian WENO scheme [15] with 8000 grids as the reference "exact" solution. In Tables 1 and 2, we summarize the errors and numerical rate of convergence of our first and second-order Lagrangian type schemes with the Dukowicz flux at $t = 1$. The results for the other three fluxes are similar. We can clearly see from Tables 1 and 2 that the first and second-order schemes achieve the designed order of accuracy, at least in the L_1 norm. However, we also see only second-order accuracy in the results of our third-order scheme. This is related to an accuracy degeneracy phenomenon of ENO schemes, originally discussed in [26] for Eulerian formulated schemes. To solve this problem, we use the modified third-order ENO scheme in one-dimension via the introduction of a biasing factor introduced in [28]. The effect of using this factor in the stencil determination procedure is to bias towards a linearly stable stencil in smooth regions. Table 3 shows the error results of the modified ENO scheme with the Dukowicz flux by using a factor of 2, according to the suggestion in [28]. From this table, we can see the modified ENO scheme recovers the correct third-order accuracy. The following third-order non-oscillatory tests are all performed by the modified third-order ENO scheme, verifying its essentially non-oscillatory property for problems with discontinuities.

Table 1
Errors of the first-order scheme on 1D meshes using N_x initially uniform cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
100	L_1	0.11E−1	–	0.15E−1	–	0.29E−1	–
	L_∞	0.34E−1	–	0.46E−1	–	0.78E−1	–
200	L_1	0.55E−2	0.93	0.77E−2	0.94	0.15E−1	0.93
	L_∞	0.19E−1	0.89	0.25E−1	0.86	0.42E−1	0.88
400	L_1	0.28E−2	0.97	0.39E−2	0.97	0.77E−2	0.96
	L_∞	0.97E−2	0.93	0.13E−1	0.92	0.22E−1	0.93
800	L_1	0.14E−2	0.98	0.20E−2	0.98	0.39E−2	0.98
	L_∞	0.50E−2	0.96	0.69E−2	0.95	0.11E−1	0.96

Table 2
Errors of the second-order ENO scheme on 1D meshes using N_x initially uniform cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
100	L_1	0.16E−2	–	0.22E−2	–	0.43E−2	–
	L_∞	0.52E−2	–	0.10E−1	–	0.17E−1	–
200	L_1	0.48E−3	1.76	0.61E−3	1.86	0.12E−2	1.80
	L_∞	0.25E−2	1.09	0.42E−2	1.25	0.72E−2	1.21
400	L_1	0.13E−3	1.93	0.16E−3	1.94	0.32E−3	1.95
	L_∞	0.93E−3	1.39	0.17E−2	1.32	0.28E−2	1.35
800	L_1	0.34E−4	1.90	0.42E−4	1.91	0.84E−4	1.91
	L_∞	0.35E−3	1.40	0.65E−3	1.38	0.11E−2	1.38

Table 3
Errors of the modified third-order ENO scheme on 1D meshes using N_x initially uniform cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
100	L_1	0.52E−4	–	0.69E−4	–	0.13E−3	–
	L_∞	0.27E−3	–	0.47E−3	–	0.69E−3	–
200	L_1	0.66E−5	2.99	0.88E−5	2.98	0.17E−4	2.99
	L_∞	0.34E−4	2.96	0.60E−4	2.96	0.88E−4	2.96
400	L_1	0.82E−6	3.00	0.11E−5	3.00	0.21E−5	3.00
	L_∞	0.43E−5	2.99	0.75E−5	2.99	0.11E−4	2.99
800	L_1	0.10E−6	3.00	0.14E−6	3.00	0.26E−6	3.00
	L_∞	0.54E−6	3.00	0.94E−6	3.00	0.14E−5	3.00

3.2. Non-oscillatory tests

Example 3.1. (Lax problem). The first non-oscillatory test is the Riemann problem proposed by Lax. Its initial condition is as follows

$$(\rho_L, u_L, p_L) = (0.445, 0.698, 3.528), \quad (\rho_R, u_R, p_R) = (0.5, 0, 0.571).$$

Fig. 1 shows the results performed by the four fluxes introduced in the previous section with 100 initially uniform cells at $t = 1$, respectively.

Comparing with the exact solution, we observe satisfactory non-oscillatory results in the pictures of Fig. 1 with the high resolution for the high order schemes for all these fluxes. If we observe closely, we can find that the results with the Godunov flux and the Dukowicz flux have slightly better resolution than that with the HLLC flux, which in turn has better resolution than that with the L–F flux. This is consistent with our discussion in the previous section on the numerical viscosity of these fluxes.

Since most of the results of the four fluxes in the following tests are similar, we will only show the results of the Dukowicz flux to save space unless their performance is distinctly different.

Example 3.2. (The Noh problem [24]). The computational domain is $[0, 1]$. The initial state of the fluid is uniform with $(\rho, u, e) = (1, -1, 0)$. The shock is generated in a perfect ideal gas ($\gamma = 5/3$) by bringing the cold gas to rest at a rigid wall ($x = 0$). The analytic post shock density is 4 and the shock speed is 1/3. The left pictures in Fig. 2 show the computational densities with 200 initially uniform cells against the exact density at $t = 0.6$. We observe the typical errors near the left boundary for all orders of accuracy. The shock resolution is better for the higher order schemes.

Example 3.3. (Two interacting blast waves). The initial data 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 condition is applied at both $x = 0$ and $x = 1$. In the right pictures of Fig. 2, the computed densities with 400 initially uniform cells at the final time $t = 0.038$ are plotted against the reference "exact" solution, which is computed using a fifth-order Eulerian WENO scheme [15] with 16000 grid points. We can see the very satisfactory resolution in the results of high order scheme with relatively coarse grids which demonstrates the advantage of the Lagrangian scheme. Meanwhile, we observe some overshoots in these figures and in some examples later. Apparently such overshoots are caused by the Lagrangian framework rather than by the high order ENO reconstruction, since they are already present for the first-order scheme which does not involve any ENO reconstruction.

Example 3.4. (Leblanc shock tube problem). In this extreme shock tube problem, the computational domain is [0,9] filled with an ideal perfect gas with $\gamma = 5/3$. The initial condition consists of large ratio jumps for the energy and density and is given by the following data

$$(\rho, u, e) = (1, 0, 0.1), \quad 0 \leqslant x < 3$$

$$(\rho, u, e) = (0.001, 0, 10^{-7}), \quad 3 < x \leqslant 9.$$

It is very difficult for a scheme to obtain accurate positions of the contact and shock discontinuities in such a severe test case [35]. The internal energy results of our schemes with the Dukowicz flux are shown in Fig. 3 (the left pictures) with 1000 initially uniform cells at $t = 6$. By comparing with the exact solution, we can see that the shape and the position of the contact discontinuity and the shock can be maintained better when the high order ENO schemes are used. In this test, the results performed by the different fluxes are visually different, so we also show the results with the HLLC flux as an example in the right pictures of Fig. 3. From the figures, we can observe that the overall resolution with the HLLC flux is better than that with the Dukowicz flux, although the latter does resolve the contact more sharply than the former.

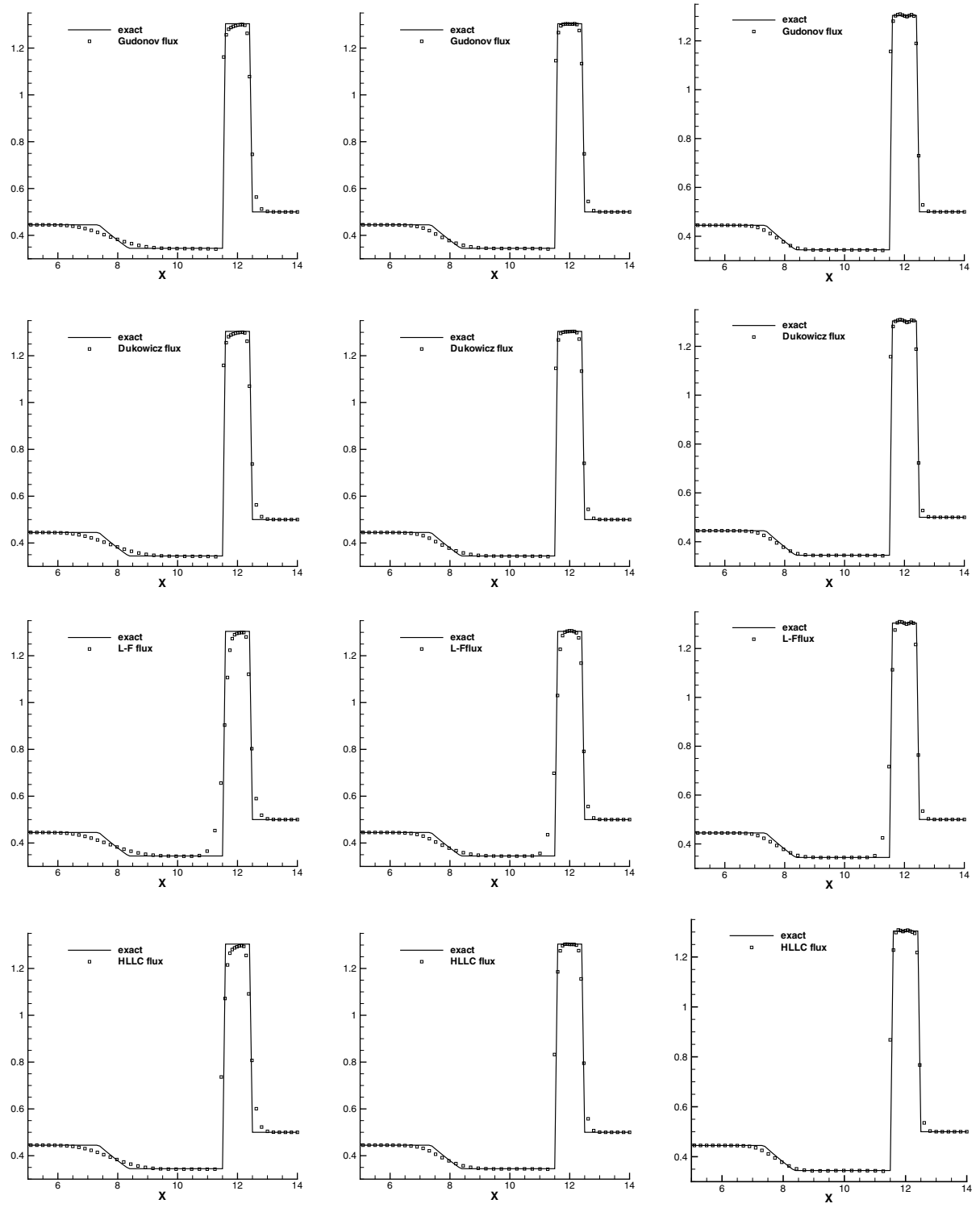


Fig. 1. The density results of the Lax problem on a 100 initially uniform cells. Left: first-order; Middle: second-order; Right: third-order. From Top to bottom: Godunov flux, Dukowicz flux, L–F flux, HLLC flux.

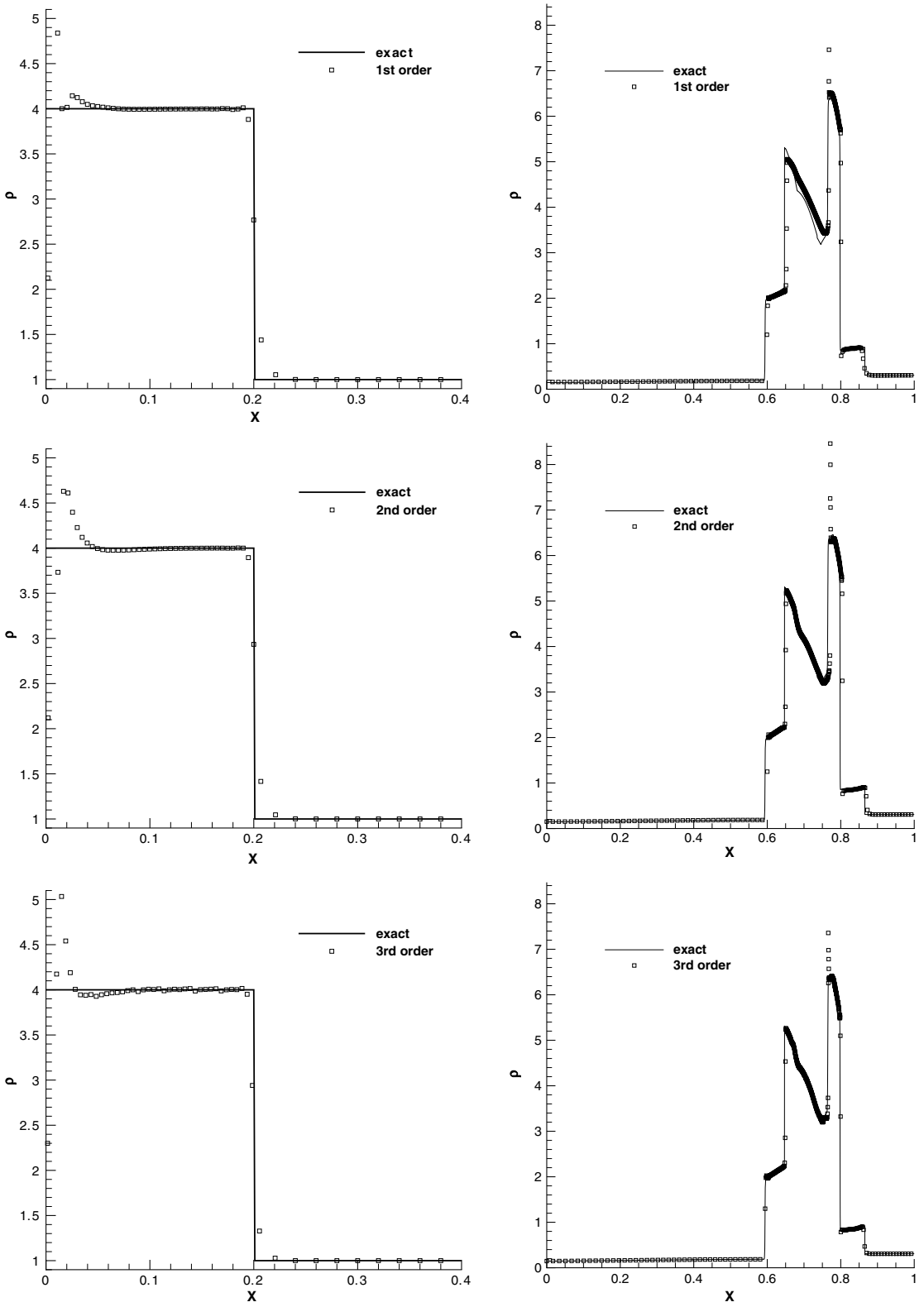


Fig. 2. The density results. Left: the Noh problem at $t = 0.6$; Right: the blast wave problem at $t = 0.038$. Top: first-order; Middle: second-order; Bottom: third-order.

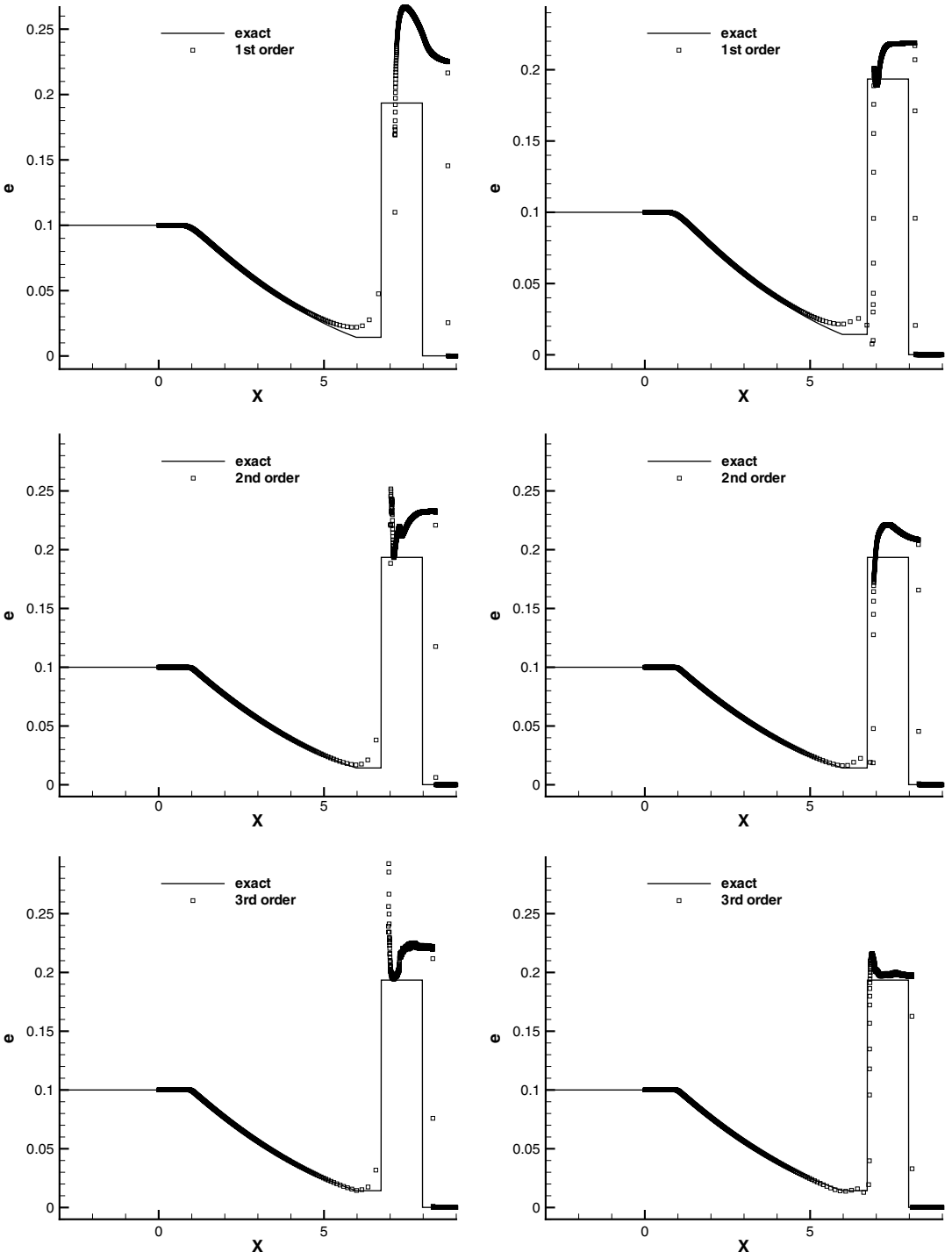


Fig. 3. The internal energy of the Leblanc problem. Left: with the Dukowicz flux; Right: with the HLLC flux. Top: first-order; Middle: second-order; Bottom: third-order.

Example 3.5. (Shock entropy wave interactions [32]). On a computational domain $[-10,5]$, the initial condition is

$$(\rho, u, e) = (3.85714, 2.629369, 10.33333), \quad x < -4$$

$$(\rho, u, e) = (1 + \epsilon \sin(kx), 0, 1), \quad x \geqslant -4$$

where ϵ and k are the amplitude and wave number of the entropy wave. In our test, we take $\epsilon = 0.2$ 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 complex smooth region structures.

In Fig. 4 (the left pictures), the computed density by the Dukowicz flux with 400 cells is plotted against the reference "exact" solution, which is obtained using the fifth-order Eulerian WENO scheme [15] with 2000 grid points. We observe that the fine structure in the density profile makes the higher order schemes perform much better than the lower order methods. For this example, we also show the density results with the HLLC flux in right pictures in Fig. 4 for a comparison, which have much smaller spurious overshoots and undershoots than the results with the Dukowicz flux.

4. High order ENO conservative Lagrangian type scheme-two space dimensions

4.1. The scheme in the Cartesian coordinates

The 2D spatial domain Ω is discretized into $M \times N$ computational cells. $I_{i+1/2,j+1/2}$ is a quadrilateral cell constructed by the four vertices $\{(x_{i,j}, y_{i,j}), (x_{i+1,j}, y_{i+1,j}), (x_{i+1,j+1}, y_{i+1,j+1}), (x_{i,j+1}, y_{i,j+1})\}$. $S_{i+1/2,j+1/2}$ is denoted to be the area of the cell $I_{i+1/2,j+1/2}$ with $i = 1, \ldots, M$, $j = 1, \ldots, N$. For a given cell $I_{i+1/2,j+1/2}$, the location of the cell center is denoted by $(x_{i+1/2,j+1/2}, y_{i+1/2,j+1/2})$. The fluid velocity $(u_{i,j}, v_{i,j})$ is defined at the vertex of the mesh. On the non-staggered mesh, all the variables except velocity are stored at the cell center of $I_{i+1/2,j+1/2}$ in the form of cell averages. For example, the values of the cell averages for the cell $I_{i+1/2,j+1/2}$ denoted by $\bar{\rho}_{i+1/2,j+1/2}$, $\overline{M}^x_{i+1/2,j+1/2}$, $\overline{M}^y_{i+1/2,j+1/2}$ and $\overline{E}_{i+1/2,j+1/2}$ are defined as follows:

$$\bar{\rho}_{i+1/2,j+1/2} = \frac{1}{S_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} \rho \mathrm{d}x\mathrm{d}y, \quad \overline{M}^x_{i+1/2,j+1/2} = \frac{1}{S_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} M_x \mathrm{d}x\mathrm{d}y,$$

$$\overline{M}^y_{i+1/2,j+1/2} = \frac{1}{S_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} M_y \mathrm{d}x\mathrm{d}y, \quad \overline{E}_{i+1/2,j+1/2} = \frac{1}{S_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} E \mathrm{d}x\mathrm{d}y$$

where ρ, M_x, M_y and E are the density, x-momentum, y-momentum and total energy, respectively.

4.1.1. Spatial discretization

The conservative semi-discrete scheme for the Eqs. (2.1) and (2.4) has the following form on the 2D non-staggered mesh

$$\frac{\mathrm{d}}{\mathrm{d}t} \begin{pmatrix} \bar{\rho}_{i+1/2,j+1/2} S_{i+1/2,j+1/2} \\ \overline{M}^x_{i+1/2,j+1/2} S_{i+1/2,j+1/2} \\ \overline{M}^y_{i+1/2,j+1/2} S_{i+1/2,j+1/2} \\ \overline{E}_{i+1/2,j+1/2} S_{i+1/2,j+1/2} \end{pmatrix} = -\int_{\partial I_{i+1/2,j+1/2}} \hat{\mathbf{F}} \mathrm{d}l = -\int_{\partial I_{i+1/2,j+1/2}} \begin{pmatrix} \hat{f}_D(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_{M_x}(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_{M_y}(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_E(\mathbf{U}_n^-, \mathbf{U}_n^+) \end{pmatrix} \mathrm{d}l. \tag{4.1}$$

Here $\mathbf{U}^{\pm} = (\rho^{\pm}, M_x^{\pm}, M_y^{\pm}, E^{\pm})$ are the values of mass, x-momentum, y-momentum and total energy at two sides of the boundary. $\mathbf{U}_n^{\pm} = (\rho^{\pm}, M_n^{\pm}, E^{\pm})$, where $M_n^{\pm}$ are the left and right component values of the momentum which is normal to the cell boundary, i.e. $M_n^{\pm} = (M_x^{\pm}, M_y^{\pm}) \cdot \mathbf{n}$, where $\mathbf{n} = (n_x, n_y)$ in (2.4) is the outward unit normal of the quadrilateral boundary $\partial I_{i+1/2,j+1/2}$. $\hat{f}_D$, $\hat{f}_{M_x}$, $\hat{f}_{M_y}$ and $\hat{f}_E$ are the numerical fluxes of mass, x-momentum, y-momentum and total energy across the cell boundary, respectively. Here in the Lagrangian formulation, we have

$$\begin{cases} \hat{f}_D(\mathbf{U}_n, \mathbf{U}_n) = 0 \\ \hat{f}_{M_x}(\mathbf{U}_n, \mathbf{U}_n) = p n_x \\ \hat{f}_{M_y}(\mathbf{U}_n, \mathbf{U}_n) = p n_y \\ \hat{f}_E(\mathbf{U}_n, \mathbf{U}_n) = p u_n, \end{cases} \tag{4.2}$$

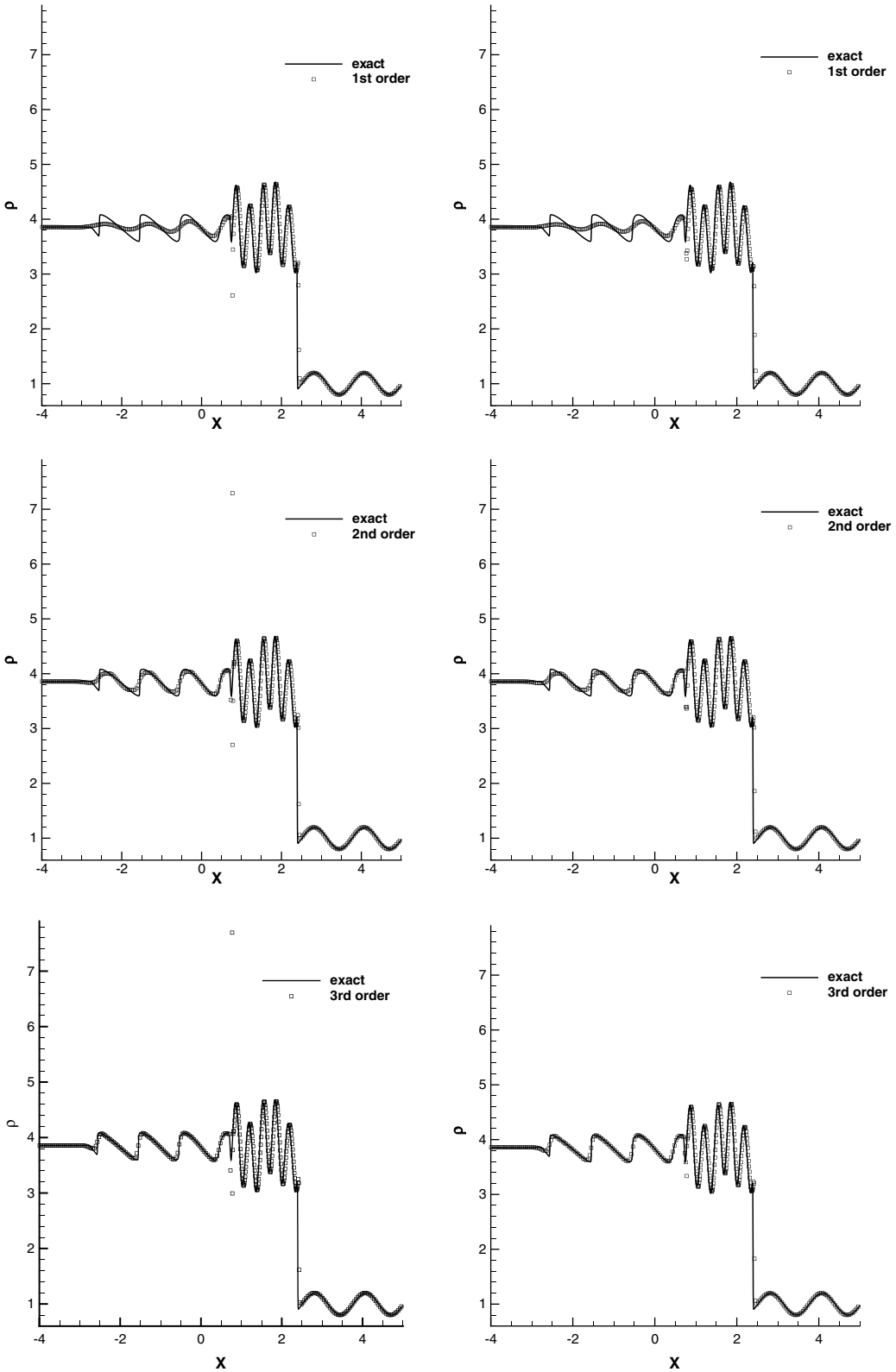


Fig. 4. The density of the shock entropy wave interactions problem. Left: with the Dukowicz flux; Right: with the HLLC flux. Top: first-order; Middle: second-order; Bottom: third- order.

where $u_n = \mathbf{u}\cdot\mathbf{n}$ is the normal velocity at the cell boundary.

Suppose the cell boundary $\partial I_{i+1/2,j+1/2}$ consists of M edges. The line integral in Eq. (4.1) is discretized by a q-point Gaussian integration formula,

$$\int_{\partial I_{i+1/2,j+1/2}} \hat{\mathbf{F}}\mathrm{d}l \approx \sum_{m=1}^{M}\sum_{k=1}^{q} \omega_k \hat{\mathbf{F}}(\mathbf{U}_n(G_k,t))\Delta l^m, \tag{4.3}$$

where Δl^m is the length of the boundary edge m and G_k are the Gaussian quadrature points at the edge. Here $\hat{\mathbf{F}}(\mathbf{U}_n(G_k,t))$ is a numerical flux. For example the L–F flux is given by

$$\widehat{\mathbf{F}}(\mathbf{U}_n(G_k,t)) = \frac{1}{2}[(\widehat{\mathbf{F}}(\mathbf{U}_n^-(G_k,t)) + \widehat{\mathbf{F}}(\mathbf{U}_n^+(G_k,t))) - \alpha(\mathbf{U}^+(G_k,t) - \mathbf{U}^-(G_k,t))], \tag{4.4}$$

where α has the same meaning as that in the one-dimensional case.

We use the high order ENO reconstruction with Roe-type characteristic decomposition [33] to obtain $\mathbf{U}^{\pm}$ and $\mathbf{U}_n^{\pm}$ at the boundary and also use sufficiently high order quadrature to construct schemes up to the expected high order spatial accuracy, for example the four-point Gauss–Lobatto integral formula is used, which has $G_1 = P_1$, $G_2 = \frac{1}{2}(P_1+P_2) - \frac{\sqrt{5}}{10}(P_2-P_1)$, $G_3 = \frac{1}{2}(P_1+P_2) + \frac{\sqrt{5}}{10}(P_2-P_1)$, $G_4 = P_2$ and $\omega_1 = \omega_4 = \frac{1}{12}$, $\omega_2 = \omega_3 = \frac{5}{12}$ for the line with endpoints P_1 and P_2. We have discussed in detail the high order ENO reconstruction needed in our framework in [6], in the context of remapping. Therefore we do not repeat the details here and refer the readers to [6]. We do mention here, however, that we have found in numerical tests that the following WENO procedure is more robust than the ENO procedure for the third-order case, hence this WENO procedure is used in the third-order numerical tests. In this procedure, the coefficients of the reconstruction polynomial are chosen as the weighted averages of those determined by the final three possible stencils introduced in [6]. To be more specific, we use density as an example. To determine the coefficients $\{a_{mn}, m+n \leqslant 2\}$ of the quadratic polynomial reconstruction function inside the cell $I_{i+1/2,j+1/2}$,

$$\rho_{i+1/2,j+1/2}(x,y) = \sum_{m+n\leqslant 2} a_{mn}(x - x_{i+1/2,j+1/2})^m (y - y_{i+1/2,j+1/2})^n,$$

suppose the coefficients of the reconstruction polynomials of the three candidate stencils are $a^i_{mn}, i = 1,2,3$, then we choose $a_{mn} = \sum_{i=1}^{3} w^i a^i_{mn}$, where w^i is the weight chosen as $w^i = (1/\sum_{m+n=2}|a^i_{mn}|^2)/c$ with $c = \sum_{i=1}^{3}(1/\sum_{m+n=2}|a^i_{mn}|^2)$. This crude WENO reconstruction, which does not increase the accuracy of each candidate stencil but is very easy to compute, performs quite nicely in our numerical experiments.

The four numerical fluxes introduced in the one-dimensional case are also applied here. The form of these fluxes in two-dimensions is similar to that in one-dimension except that the left and right values at the cell's boundary are chosen as $\mathbf{U}_n^{\pm}$ in two-dimensions rather than $\mathbf{U}^{\pm}$ in one-dimension.

4.1.2. The determination of the vertex velocity

Considering a vertex (i,j) shared by four edges which are given a serial number $k = 1,2,3,4$, we 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). Along each edge k we can obtain the left value of velocity $(u^{k-}, v^{k-}) = (M_x^{k-}/\rho^{k-}, M_y^{k-}/\rho^{k-})$ and the right velocity $(u^{k+}, v^{k+}) = (M_x^{k+}/\rho^{k+}, M_y^{k+}/\rho^{k+})$ at this vertex in the procedure of the flux computation since the vertex (i,j) is one of the Gaussian quadrature points for our choice. We then split the left and right velocities into normal and tangential components along the edge k. Denote (n_x^k, n_y^k) to be the clockwise unit normal of the edge k and denote w_t^{k-} and w_t^{k+} to be their tangential components and w_n^{k-} and w_n^{k+} to be their normal components. Then the tangential velocity of the vertex (i,j) along the edge k is defined as

$$w_t^k = \frac{1}{2}(w_t^{k-} + w_t^{k+}), \quad k = 1,2,3,4. \tag{4.5}$$

As to the normal velocity, for the Godunov and the Dukowicz fluxes, we obtain it by the Riemann solver here and for the L–F flux and the HLLC flux, we get it by the Roe average of the normal velocities from its two sides as in the one-dimensional case, that is

$$w_n^k = \frac{\sqrt{\rho^-}w_n^{k-} + \sqrt{\rho^+}w_n^{k+}}{\sqrt{\rho^-} + \sqrt{\rho^+}}, \quad k = 1,2,3,4, \tag{4.6}$$

where $\rho^{\pm}$ are the densities from the left and right cells of the edge k, respectively.

Thus by the formulas (4.5) and (4.6), we can get four x-velocities and y-velocities at the vertex (i,j) which have the following form,

$$w_x^k = w_n^k n_x^k - w_t^k n_y^k, \quad w_y^k = w_n^k n_y^k + w_t^k n_x^k, \quad k = 1,2,3,4. \tag{4.7}$$

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

$$u_{i,j} = \frac{1}{4}(w_x^1 + w_x^2 + w_x^3 + w_x^4), \quad v_{i,j} = \frac{1}{4}(w_y^1 + w_y^2 + w_y^3 + w_y^4). \tag{4.8}$$

4.1.3. Time discretization

The time discretization is also similar to that in one-dimension. We only list the first-order Lagrangian type scheme as a representative here to save space

$$\begin{pmatrix} \bar{\rho}^{n+1}_{i+1/2,j+1/2}S^{n+1}_{i+1/2,j+1/2} - \bar{\rho}^{n}_{i+1/2,j+1/2}S^{n}_{i+1/2,j+1/2} \\ \overline{M}^{x,n+1}_{i+1/2,j+1/2}S^{n+1}_{i+1/2,j+1/2} - \overline{M}^{x,n}_{i+1/2,j+1/2}S^{n}_{i+1/2,j+1/2} \\ \overline{M}^{y,n+1}_{i+1/2,j+1/2}S^{n+1}_{i+1/2,j+1/2} - \overline{M}^{y,n}_{i+1/2,j+1/2}S^{n}_{i+1/2,j+1/2} \\ \overline{E}^{n+1}_{i+1/2,j+1/2}S^{n+1}_{i+1/2,j+1/2} - \overline{E}^{n}_{i+1/2,j+1/2}S^{n}_{i+1/2,j+1/2} \end{pmatrix} = -\Delta t^n \sum_{m=1}^{M}\sum_{k=1}^{q} \omega_k \hat{\mathbf{F}}(\mathbf{U}_n(G_k,t))\Delta l^m, \tag{4.9}$$

where $S^n_{i+1/2,j+1/2}$ and $S^{n+1}_{i+1/2,j+1/2}$ are the areas of Cell $I_{i+1/2,j+1/2}$ at the n-th and $(n+1)$th time steps, respectively. $S^{n+1}_{i+1/2,j+1/2}$ is determined by the following simple formulas,

$$\begin{aligned} &x^{n+1}_{i,j} = u^n_{i,j}\Delta t^n + x^n_{i,j}, \quad y^{n+1}_{i,j} = v^n_{i,j}\Delta t^n + y^n_{i,j}, \\ &S^{n+1}_{i,j} = \frac{1}{2}[(x^{n+1}_{i+1,j+1} - x^{n+1}_{i,j})(y^{n+1}_{i,j+1} - y^{n+1}_{i+1,j}) + (x^{n+1}_{i,j+1} - x^{n+1}_{i+1,j})(y^{n+1}_{i+1,j+1} - y^{n+1}_{i,j})], \\ &\quad i = 1,\ldots,M, \quad j = 1,\ldots,N. \end{aligned} \tag{4.10}$$

The time step Δt^n is chosen as follows

$$\Delta t^n = \lambda \min_{i=1,\ldots,M,j=1,\ldots,N}(\Delta l^n_{i+1/2,j+1/2}/c^n_{i+1/2,j+1/2}), \tag{4.11}$$

where $\Delta l^n_{i+1/2,j+1/2}$ is the shortest edge length of the cell $I_{i+1/2,j+1/2}$, and $c^n_{i+1/2,j+1/2}$ is the sound speed within this cell. The Courant number λ in the following tests is set to be 0.5 unless otherwise stated.

4.2. The scheme in the cylindrical coordinates

We seek to study the flow governed by the axisymmetric compressible Euler equations which have the following integral form in the Lagrangian formulation,

$$\begin{cases} \frac{\mathrm{d}}{\mathrm{d}t}\int\int_{\Omega(t)} \rho r \mathrm{d}x\mathrm{d}r = 0 \\ \frac{\mathrm{d}}{\mathrm{d}t}\int\int_{\Omega(t)} M_x r \mathrm{d}x\mathrm{d}r = -\int_{\Gamma(t)} p n_x r \mathrm{d}l \\ \frac{\mathrm{d}}{\mathrm{d}t}\int\int_{\Omega(t)} M_r r \mathrm{d}x\mathrm{d}r = -\int_{\Gamma(t)} p n_r r \mathrm{d}l + \int\int_{\Omega(t)} (p + \rho u_\theta^2)\mathrm{d}x\mathrm{d}r \\ \frac{\mathrm{d}}{\mathrm{d}t}\int\int_{\Omega(t)} M_\theta r \mathrm{d}x\mathrm{d}r = -\int\int_{\Omega(t)} \rho u_\theta u_r \mathrm{d}x\mathrm{d}r \\ \frac{\mathrm{d}}{\mathrm{d}t}\int\int_{\Omega(t)} E r \mathrm{d}x\mathrm{d}r = -\int_{\Gamma(t)} p u_n r \mathrm{d}l, \end{cases} \tag{4.12}$$

where ρ is the density, p is the pressure, M_x, M_r, M_θ are the momentum components in the axial, radial and azimuthal directions, and u_x, u_r, u_θ are the velocity components in the above mentioned directions. $\mathbf{n} = (n_x, n_r)$ is the unit outward normal to the boundary $\Gamma(t)$ in the $x - r$ coordinates. $u_n = (u_x, u_r)\cdot\mathbf{n}$ is the normal velocity. E is the total energy.

The values of the cell averages for the cell $I_{i+1/2,j+1/2}$, denoted by $\bar{\rho}_{i+1/2,j+1/2}$, $\overline{M}^x_{i+1/2,j+1/2}$, $\overline{M}^r_{i+1/2,j+1/2}$, $\overline{M}^\theta_{i+1/2,j+1/2}$ and $\overline{E}_{i+1/2,j+1/2}$, are defined as follows

$$\bar{\rho}_{i+1/2,j+1/2} = \frac{1}{V_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} \rho r \mathrm{d}x\mathrm{d}r, \quad \overline{M}^x_{i+1/2,j+1/2} = \frac{1}{V_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} M_x r \mathrm{d}x\mathrm{d}r,$$

$$\overline{M}^r_{i+1/2,j+1/2} = \frac{1}{V_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} M_r r \mathrm{d}x\mathrm{d}r, \quad \overline{M}^\theta_{i+1/2,j+1/2} = \frac{1}{V_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} M_\theta r \mathrm{d}x\mathrm{d}r,$$

$$\overline{E}_{i+1/2,j+1/2} = \frac{1}{V_{i+1/2,j+1/2}} \int\int_{I_{i+1/2,j+1/2}} E r \mathrm{d}x\mathrm{d}r.$$

where $V_{i+1/2,j+1/2} = \int\int_{I_{i+1/2,j+1/2}} r\mathrm{d}x\mathrm{d}r$ is the volume of the cell $I_{i+1/2,j+1/2}$.

The conservative semi-discrete scheme for the Eq. (4.12) has the following form

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} \bar{\rho}_{i+1/2,j+1/2}V_{i+1/2,j+1/2} \\ \bar{M}^x{}_{i+1/2,j+1/2}V_{i+1/2,j+1/2} \\ \bar{M}^r{}_{i+1/2,j+1/2}V_{i+1/2,j+1/2} \\ \bar{M}^\theta{}_{i+1/2,j+1/2}V_{i+1/2,j+1/2} \\ \bar{E}_{i+1/2,j+1/2}V_{i+1/2,j+1/2} \end{pmatrix} = -\int_{\partial I_{i+1/2,j+1/2}} \widehat{\mathbf{F}}\mathrm{d}l + \begin{pmatrix} 0 \\ 0 \\ \int\int_{I_{i+1/2,j+1/2}} (p+\rho u_\theta^2)\mathrm{d}x\mathrm{d}r \\ -\int\int_{I_{i+1/2,j+1/2}} \rho u_\theta u_r \mathrm{d}x\mathrm{d}r \\ 0, \end{pmatrix} \tag{4.13}$$

where

$$\int_{\partial I_{i+1/2,j+1/2}} \widehat{\mathbf{F}}\mathrm{d}l = \int_{\partial I_{i+1/2,j+1/2}} \begin{pmatrix} \hat{f}_D(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_{M^x}(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_{M^r}(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_{M^\theta}(\mathbf{U}_n^-, \mathbf{U}_n^+) \\ \hat{f}_E(\mathbf{U}_n^-, \mathbf{U}_n^+) \end{pmatrix} \mathrm{d}l, \tag{4.14}$$

and

$$\begin{cases} \hat{f}_D(\mathbf{U}_n, \mathbf{U}_n) &= 0, \\ \hat{f}_{M^x}(\mathbf{U}_n, \mathbf{U}_n) &= p n_x r \\ \hat{f}_{M^r}(\mathbf{U}_n, \mathbf{U}_n) &= p n_r r \\ \hat{f}_{M^\theta}(\mathbf{U}_n, \mathbf{U}_n) &= 0 \\ \hat{f}_E(\mathbf{U}_n, \mathbf{U}_n) &= p u_n r. \end{cases} \tag{4.15}$$

The calculation of the first term on the right-hand side of Eq. (4.13) is similar to that in the Cartesian coordinates introduced in the above subsection. The calculation of the second term is performed by a suitable Gaussian integral in the corresponding cell to guarantee its high order accurate approximation.

We use the same method as that used in the Cartesian coordinates to decide the velocity components (u_x, u_r) at the vertex in the x and r directions (since the grid moves just in the $x - r$ coordinates, we only need to know u_x, u_r).

We also use the Runge–Kutta method to discretize the time derivatives in (4.13). The method to calculate the time step is the same as that in the Cartesian coordinates.

5. Numerical results in two space dimensions

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, although we have run most examples using the first, second and third-order schemes with the Godunov flux, the Dukowicz flux, the HLLC flux and the L–F flux, respectively, we will only show the results performed by the Dukowicz flux as representatives unless the results of the different fluxes are obviously different.

5.1. Numerical results in the Cartesian coordinates

5.1.1. Accuracy test

In the Cartesian coordinates, we choose the two-dimensional vortex evolution problem [30] as our accuracy test function. The 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 perturbations in (u, v) and the temperature $T = p/\rho$, no perturbation in the entropy $S = p/\rho^{\gamma}$.

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

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

The computational domain is taken as $[0, 10] \times [0, 10]$, and periodic boundary conditions are used.

The convergence results for the first, second and third-order ENO Lagrangian type schemes at $t = 1$ are listed in Tables 4–6, respectively. In Tables 4 and 5, we can see the desired first and second-order accuracy. However in Table 6 we cannot observe the expected third-order accuracy. Further exploration indicates that this accuracy degeneracy cannot be cured by the modified ENO scheme via the introduction of a biasing factor

Table 4
Errors of the first-order scheme on 2D Cartesian coordinates for the vortex problem using $N_x \times N_y$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.83E−2	–	0.23E−1	–	0.41E−1	–
	L_∞	0.73E−1	–	0.19E+0	–	0.35E+0	–
40	L_1	0.52E−2	0.69	0.13E−1	0.84	0.24E−1	0.76
	L_∞	0.50E−1	0.56	0.10E+0	0.89	0.20E+0	0.84
80	L_1	0.29E−2	0.85	0.69E−2	0.91	0.13E−1	0.90
	L_∞	0.26E−1	0.91	0.53E−1	0.96	0.10E+0	0.93
160	L_1	0.15E−2	0.93	0.36E−2	0.95	0.66E−2	0.95
	L_∞	0.14E−1	0.97	0.27E−1	0.98	0.53E−1	0.95

Table 5
Errors of the second-order ENO scheme on 2D Cartesian coordinates for the vortex problem using $N_x \times N_y$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.58E−2	–	0.13E−1	–	0.23E−1	–
	L_∞	0.91E−1	–	0.15E+0	–	0.35E+0	–
40	L_1	0.16E−2	1.85	0.40E−2	1.74	0.72E−2	1.71
	L_∞	0.28E−1	1.72	0.48E−1	1.66	0.12E+0	1.52
80	L_1	0.51E−3	1.65	0.12E−2	1.77	0.21E−2	1.79
	L_∞	0.85E−2	1.71	0.18E−1	1.44	0.39E−1	1.63
160	L_1	0.16E−3	1.70	0.33E−3	1.81	0.59E−3	1.82
	L_∞	0.36E−2	1.26	0.67E−2	1.41	0.13E−1	1.57

Table 6
Errors of the third-order ENO scheme on 2D Cartesian coordinates for the vortex problem using $N_x \times N_y$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.31E−2	–	0.57E−2	–	0.11E−1	–
	L_∞	0.30E−1	–	0.58E−1	–	0.18E+0	–
40	L_1	0.77E−3	2.01	0.14E−2	2.03	0.28E−2	2.04
	L_∞	0.97E−2	1.64	0.20E−1	1.51	0.60E−1	1.58
80	L_1	0.19E−3	2.04	0.33E−3	2.08	0.66E−3	2.06
	L_∞	0.25E−2	1.98	0.49E−2	2.07	0.13E−1	2.24
160	L_1	0.47E−4	1.99	0.81E−4	2.01	0.16E−3	2.02
	L_∞	0.81E−3	1.59	0.13E−2	1.89	0.34E−2	1.88

in the stencil-choosing process. It represents a fundamental problem in our way of formulating the Lagrangian schemes. In a Lagrangian simulation, each cell represents a material particle, thus its shape may change with the movement of fluid, that means the cell with a quadrilateral shape initially may not keep its shape as a quadrilateral at a later time. It usually becomes a curved quadrilateral, while during our Lagrangian simulation the mesh is always supposed to be quadrilateral which is determined by the movement of its four vertices. This approximation of the mesh will bring second-order error into the scheme. Thus for a Lagrangian scheme in multi-dimensions, it can be at most second-order accurate if curved meshes are not used. We will not explore curved meshes in this paper, as it would require a new reconstruction procedure based on such cells with curved boundaries. Our "third-order" scheme is therefore only second-order accurate. However, in the following examples, we indeed often find better resolution in the fluid field by the third-order scheme compared with that obtained by the lower order schemes, despite its formal second-order accuracy. We also observe third-order accuracy in some test problems, e.g. Table 9 in Section 5.2.1, when the mesh does not lose its quadrilateral shape.

5.1.2. Non-oscillatory tests

Example 5.1. (The Saltzman problem [10]). This is a well known difficult test case to validate the robustness of a Lagrangian scheme when the mesh is not aligned with the fluid flow. 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 initial mesh is 100 cells in the x-direction and 10 cells in the y-direction which is defined by

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

where $\Delta x = \Delta y = 0.01$. The initial mesh is displayed in Fig. 5. Notice 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 0.5 after $t = 0.01$. The analytic post shock density is 4.0 and the shock speed is 1.333. The purely Lagrangian numerical results by using the Dukowicz flux are shown in Fig. 6 for the time $t = 0.6$. At this time, the shock is expected to be located at $x = 0.8$. We can observe that our high order schemes preserve one-dimensional solution well except for the region near the up and bottom wall boundaries where the results are affected by the boundary conditions. Also the higher order schemes give better shock resolution for this example. For comparison, Fig. 7 shows the results of our scheme with the L–F flux which demonstrate that the solution of the L–F flux is more dissipative but less oscillatory, in comparison with the solution of the Dukowicz flux for this problem.

Example 5.2. (The Sedov blast wave problem in a Cartesian coordinate system [27]). The Sedov blast wave problem models the expanding wave by an intense explosion in a perfect gas. The simulation is first performed on a Cartesian grid whose initial uniform grid consists of 30×30 rectangular cells with a total edge length of 1.1 in both directions. The initial density is unity and the initial velocity is zero. The specific internal energy is zero except in the first zone where it has a value of 182.09. The analytical solution gives a shock at radius unity at time unity with a peak density of 6. Fig. 8 shows the results by the purely Lagrangian calculations at the time $t = 1$. We can clearly see that the high order ENO scheme obtains more precise solution than the lower order one.

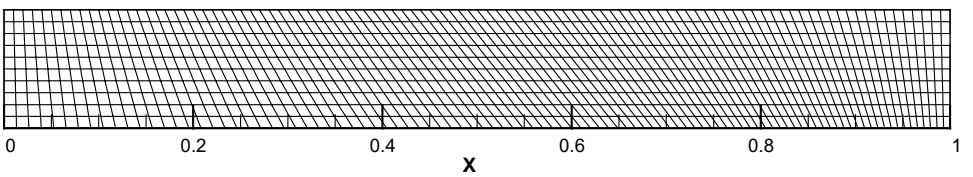
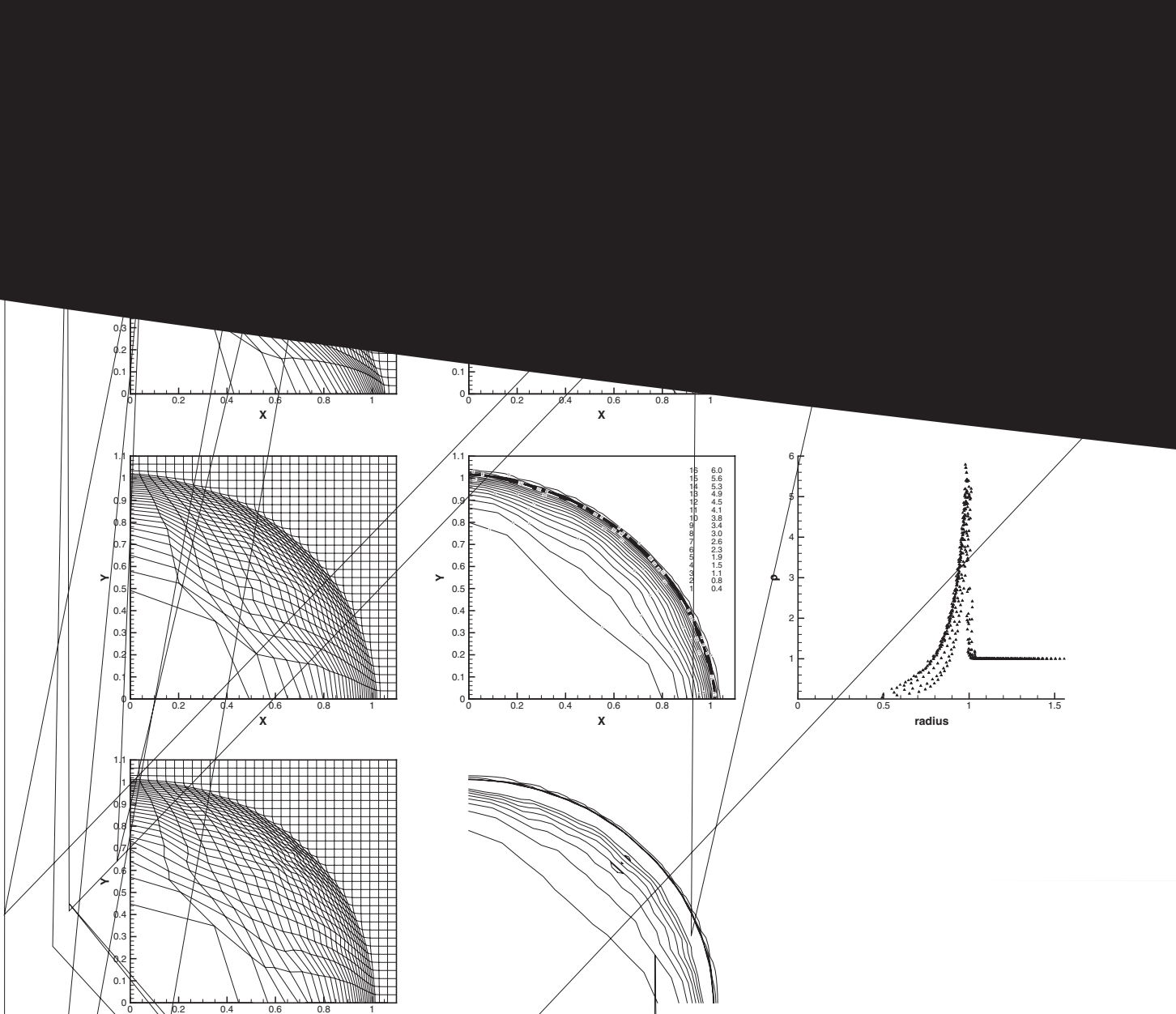


Fig. 5. The initial mesh of the Saltzman problem.



direction. The right region is a 40×30 mesh uniformly slanted at 30° to the horizontal direction. See Fig. 9. The initial conditions of the two regions are $\rho_L = 1,\ u_L = 0, p_L = 1$ and $\rho_R = 1.5, u_R = 0, p_R = 1$, respectively. The upper and lower boundaries are reflective and the left boundary is a piston, which moves from the left with velocity 1.48. The problem is run to a time of 1.3, just before the shock would leave the right region. The exact solution to the problem is shown in Fig. 10 which is only valid away from the boundary as it is obtained under the assumption of an infinite medium. Fig. 11 shows the pure Lagrangian results of our first and second schemes. We have been unable to obtain stable results to the final time for our third-order scheme in the pure Lagrangian simulations for this problem, as the mesh gets tangled near the bottom boundary due to the numerical boundary condition for the ghost cells. In the ALE calculations, in order to make the results of our schemes of different order comparable, we have attempted to keep the number of rezoning and remapping steps for these schemes to be the same, which is set as 16 in this test. For example, for the second-order scheme, when the area of the minimum cell is less than 2×10^{-4} or the time step is less than 2×10^{-3}, we rezone the meshes by keeping the vertices at the left and right boundaries unchanged and redistributing the points in

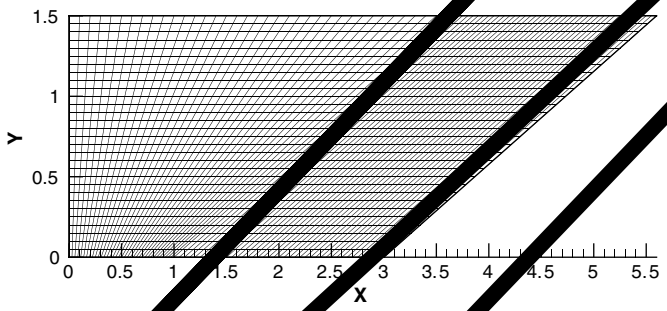
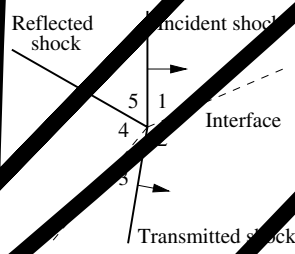
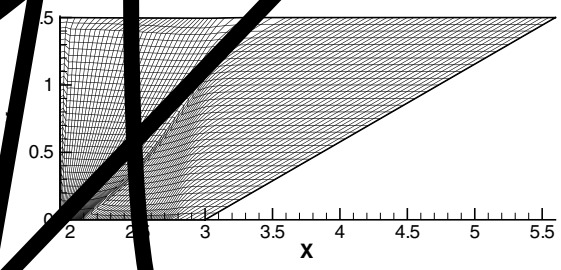


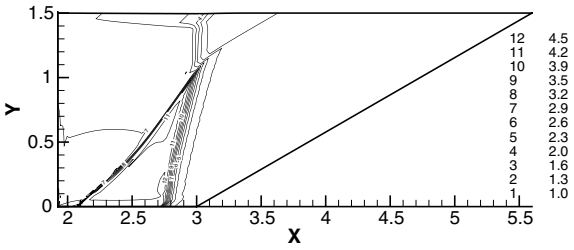
Fig. 9. The initial computational grid.

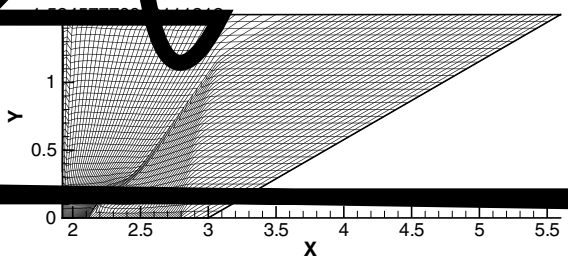


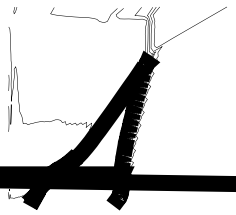
Region	1	2	3	4	5
Angle	60	131.02	21.90	90.96	56.12
Density	1	1.5	4.29	2.93	2.67
Pressure	1	1	5.15	5.15	4.50

Fig. 10. The exact solution for the Dukowicz problem.









the x-direction evenly. The numerical results using the ALE calculation are shown at time $t = 1.3$ in Fig. 12. The density contour plots in Fig. 12 give consistent results with the exact solution. The results of our schemes show the interface along with the incident and the transmitted shocks clearly. The reflective shock does not show up clearly due to the small difference in density across it. Also the higher order schemes produce results with better resolution for this example.

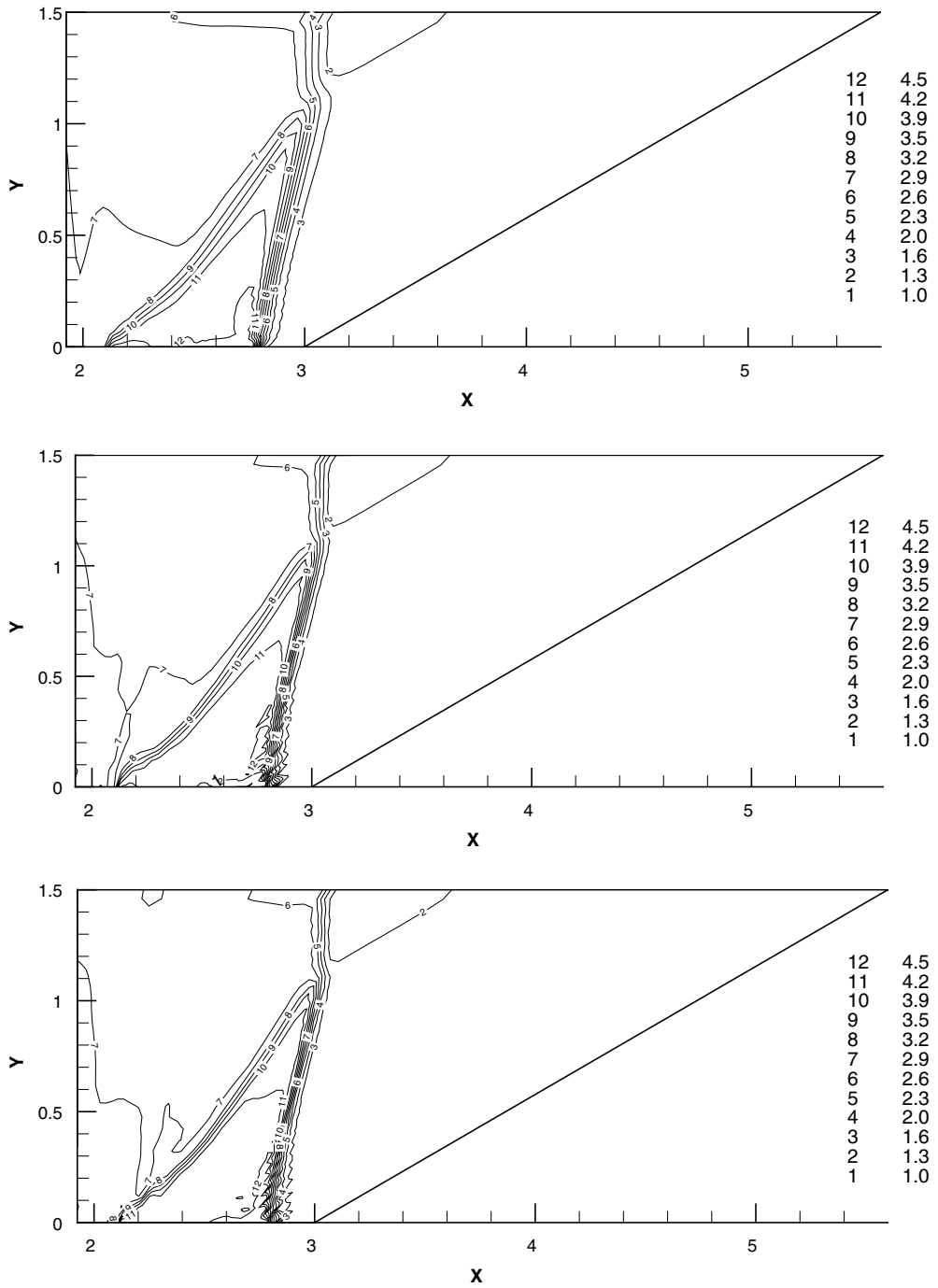


Fig. 12. The density contour plot of the Dukowicz problem by using the ALE method. Top: first-order; Middle: second-order; Bottom: third-order.

Example 5.4. (Double Mach reflection). The computational domain for this problem is chosen to be $[0,4]\times[0,1]$. The reflecting wall lies at the bottom of the computational domain starting $x = 1/6$. Initially a right-moving Mach 10 shock is positioned at $x = 1/6$, $y = 0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = 1/6$ and a reflec-

tive boundary condition is used for the rest. At the top boundary of our computational domain, the flow values are set to describe the exact motion of the Mach 10 shock. In this example, we let the mesh return to its original state after each time step. We should note that initially the Courant number for the third-order scheme takes a smaller value at 0.45 and after time = 0.01 it returns to the usual value 0.5. The density results at the time $t = 0.2$ on a 240×60 uniform grid are shown in Fig. 13, which demonstrate that our schemes also perform well using the Eulerian mesh.

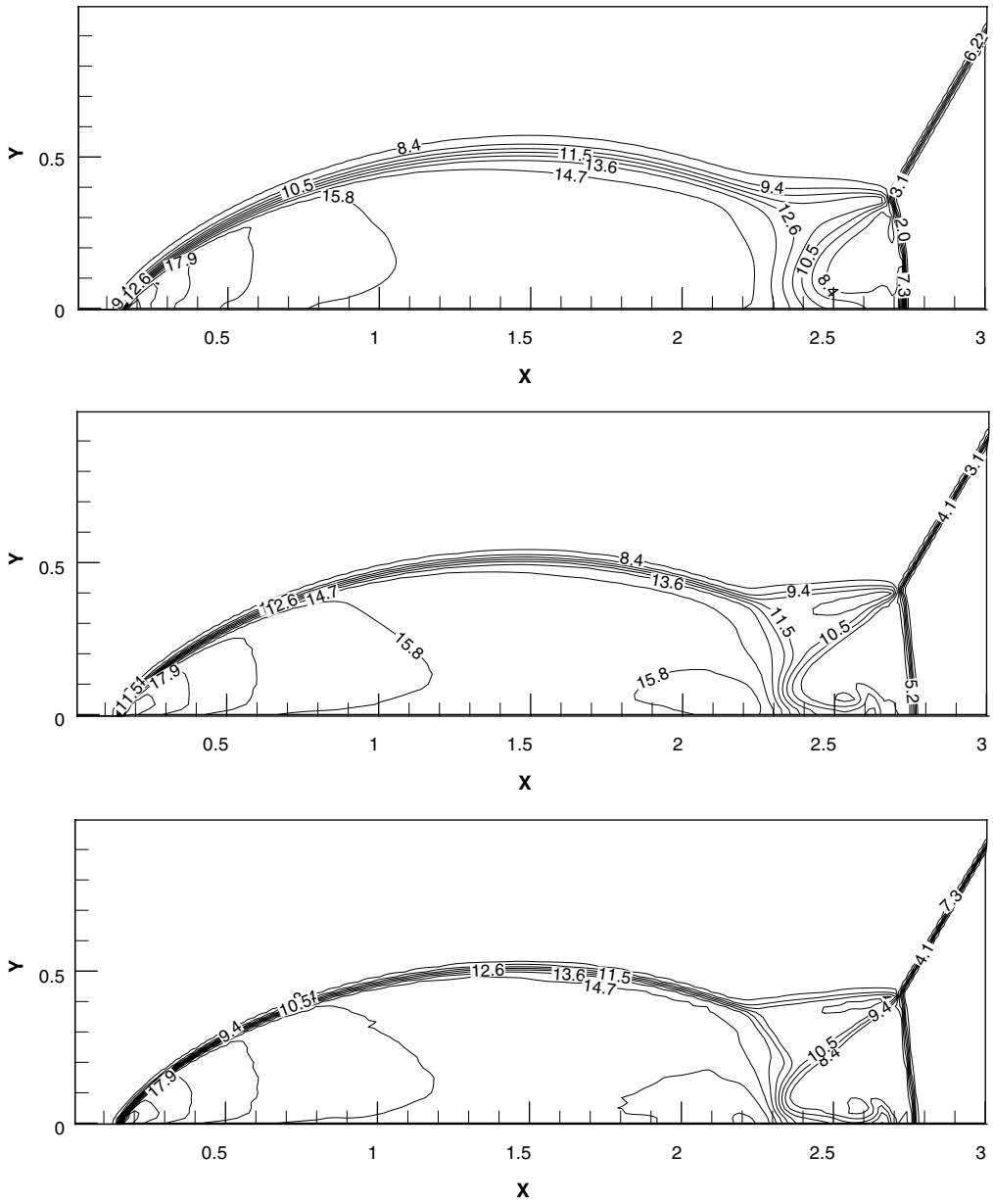


Fig. 13. The results of the double Mach reflection problem with Eulerian meshes. Top: first-order; Middle: second-order; Bottom: third-order. Density contour plots from 2 to 20 with 19 equally spaced contours.

5.2. Numerical results in the cylindrical coordinates

5.2.1. Accuracy test

In the cylindrical coordinates, we test a problem with non-trivial velocities in the x and θ directions. The longitudinal vortex located in the $r-\theta$ coordinates moves along the symmetric x-coordinate with a constant velocity. Here we let $u_x = 2, u_r = 0$. u_θ is the azimuthal velocity of the vortex which has the same definition as that introduced in the previous subsection. To avoid the influence of the coordinate singularity at $r = 0$, we perform our accuracy tests on the computational domain $[0, 10] \times [1, 11]$. The results are presented in Tables 7–9 which show a satisfactory convergence performance in the cylindrical coordinates. In particular, we happily observe the desired third-order accuracy in the results of our third-order schemes, which is due to the mesh being able to keep its quadrilateral shape as the time marches in this test. The result also reinforces our previous claim about the reason of the accuracy degeneracy phenomenon due to the appearance of curved quadrilaterals, which is approximated by quadrilaterals in our numerical procedure, in the 2D Cartesian accuracy test.

Table 7
Errors of the first-order scheme on 2D cylindrical coordinates for the vortex problem using $N_x \times N_r$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.74E−3	–	0.23E−2	–	0.49E−2	–
	L_∞	0.16E−1	–	0.11E+0	–	0.10E+0	–
40	L_1	0.38E−3	0.97	0.11E−2	1.07	0.25E−2	0.97
	L_∞	0.14E−1	0.14	0.74E−1	0.60	0.73E−1	0.50
80	L_1	0.19E−3	1.02	0.51E−3	1.07	0.12E−2	1.02
	L_∞	0.11E−1	0.41	0.47E−1	0.67	0.47E−1	0.61
160	L_1	0.91E−4	1.02	0.25E−3	1.04	0.61E−3	1.02
	L_∞	0.74E−2	0.55	0.29E−1	0.69	0.29E−1	0.71

Table 8
Errors of the second-order ENO scheme on 2D cylindrical coordinates for the vortex problem using $N_x \times N_r$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.30E−3	–	0.95E−3	–	0.18E−2	–
	L_∞	0.91E−2	–	0.59E−1	–	0.53E−1	–
40	L_1	0.60E−4	2.31	0.21E−3	2.19	0.35E−3	2.38
	L_∞	0.41E−2	1.15	0.19E−1	1.65	0.21E−1	1.34
80	L_1	0.17E−4	1.85	0.55E−4	1.93	0.93E−4	1.91
	L_∞	0.11E−2	1.84	0.48E−2	1.97	0.53E−2	1.97
160	L_1	0.42E−5	1.99	0.14E−4	1.96	0.23E−4	1.99
	L_∞	0.34E−3	1.77	0.13E−2	1.88	0.13E−2	1.97

Table 9
Errors of the third-order ENO scheme on 2D cylindrical coordinates for the vortex problem using $N_x \times N_r$ initially uniform mesh cells

N_x	Norm	Density	Order	Momentum	Order	Energy	Order
20	L_1	0.22E−4	–	0.96E−4	–	0.81E−4	–
	L_∞	0.63E−3	–	0.31E−2	–	0.24E−2	–
40	L_1	0.24E−5	3.18	0.98E−5	3.30	0.89E−5	3.18
	L_∞	0.84E−4	2.90	0.27E−3	3.49	0.46E−3	2.39
80	L_1	0.26E−6	3.20	0.11E−5	3.13	0.10E−5	3.09
	L_∞	0.19E−4	2.11	0.55E−4	2.31	0.12E−3	1.97
160	L_1	0.30E-7	3.15	0.13E−6	3.13	0.14E−6	2.89
	L_∞	0.32E−5	2.60	0.10E−4	2.39	0.18E−4	2.69

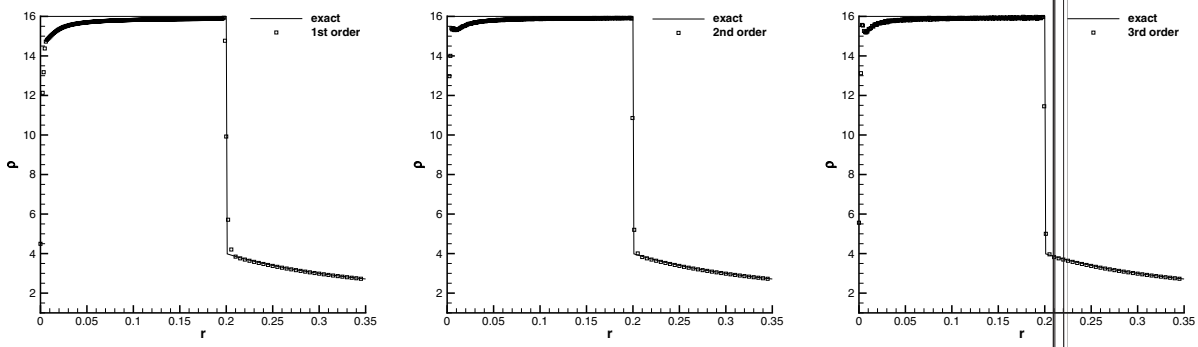
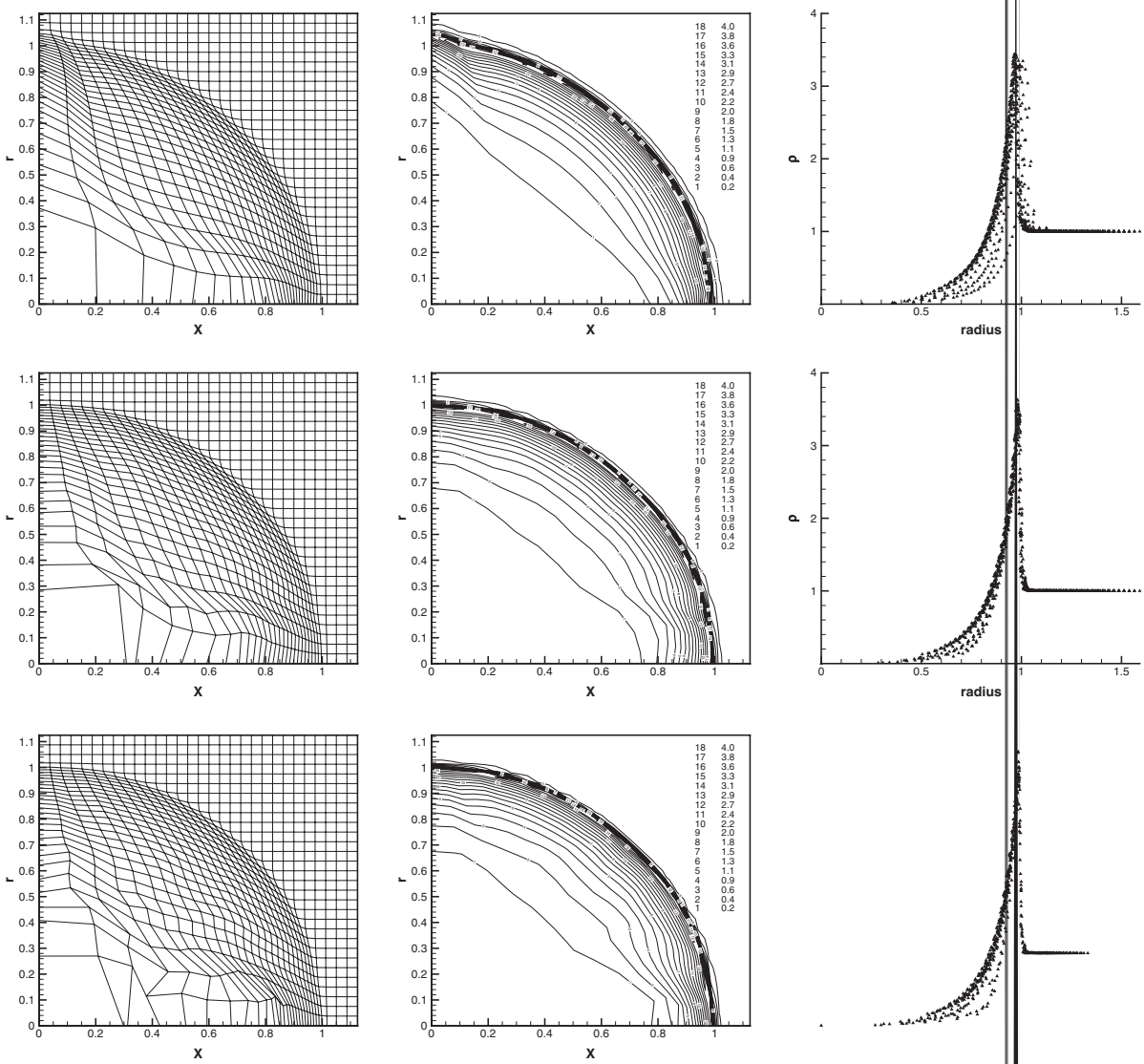


Fig. 14. The density results of the Noh problem. Left: first-order; Middle: second-order; Right: third-order.



Example 5.7. (Interaction of a shock with longitudinal vortex [11]). The computational domain is $[-8,4]\times[0,5]$. At $t=0$, there is a mean flow with a stationary shock at $x=0$, that is

$$(\rho,p,u_x,u_r,u_\theta)=\begin{cases}(\rho_1,p_1,u_{x,1},u_{r,1},u_{\theta,1})=(1,1,\gamma^{1/2}M_1,0,0), & x<0\\ (\rho_2,p_2,u_{x,2},u_{r,2},u_{\theta,2})=\left[\frac{(\gamma+1)M_1^2}{2+M_1^2(\gamma-1)},\frac{2\gamma M_1^2-(\gamma-1)}{(\gamma+1)},M_2\sqrt{\gamma\frac{p_2}{\rho_2}},0,0\right], & x>0,\end{cases} \tag{5.1}$$

where M_1 is the Mach number at the upstream of the shock ($x<0$) and $M_2=\frac{2+M_1^2(\gamma-1)}{2\gamma M_1^2-(\gamma-1)}$ is the Mach number at the downstream of the shock ($x>0$).

Next, we superimpose an isentropic vortex with its axis along $r=0$ on the upstream of the mean flow. The perturbation of azimuthal velocity u'_θ and temperature T' associated with the vortex are given by

$$u'_\theta=\frac{\epsilon r}{2\pi}e^{0.5(1-r^2)},\quad T'=-\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2r_0^2}e^{1-r^2}, \tag{5.2}$$

where r_0 is the vortex core radius and ϵ is a non-dimensional circulation at $r=1$. The axial and radial velocities u'_x,u'_r are zero. With no perturbation of the entropy $S=\log(p/\rho^\gamma)$ of the original mean flow, the final perturbed flow at $x<0$ is as follows

$$\rho=\left(\frac{T_1+T'}{p_1/\rho_1^\gamma}\right)^{1/(\gamma-1)},\quad p=(T_1+T')\rho,\quad u_x=u_{x,1},\quad u_r=0,\quad u_\theta=u'_\theta,$$

where $T_1=p_1/\rho_1$. In this test, we set $M_1=2,\epsilon=7$ and $r_0=1$. The supersonic inflow, characteristic, symmetry and Neumann conditions are used at the left, right, bottom and upper boundaries, respectively. The initial grid is uniform. After every three Lagrangian time steps, we take the rezoning and remapping steps to return the Lagrangian grid to the initial grid. The density results at two typical times of our schemes are given in Fig. 16. From these figures we observe that the resolution of high order schemes on coarser meshes are comparable to that of low order schemes on finer meshes, and more details of the fluid are captured by using higher order schemes. These results are also consistent with the results shown in [11].

6. Concluding remarks

In this paper we have described a class of Lagrangian type schemes for solving Euler equations which are based on high order essentially non-oscillatory (ENO) reconstruction both in the Cartesian coordinates and in the cylindrical coordinates. The schemes are conservative for density, momentum and total energy, maintain formal high order accuracy both in space and time and can achieve at least uniformly second-order accuracy with moving and distorted Lagrangian meshes, are essentially non-oscillatory, and have no parameters to be tuned for individual test cases. It is possible to extend the method to higher order accuracy by using curved meshes, but we have not studied this generalization in this paper. Comparing with many current Lagrangian type schemes, our ENO schemes overcome some of their disadvantages such as non-conservativity of momentum and total energy, low accuracy, and the existence of parameters which must be adjusted for individual test cases. One-dimensional and two-dimensional examples in the Cartesian as well as cylindrical coordinates have been presented which demonstrate the good performance of the schemes both in purely Lagrangian and in ALE calculations. Although we have only performed tests on quadrilateral grids, the strategy can be used on any polygon grid such as triangles. The investigation and improvement of these high order schemes in multi-dimensions in terms of accuracy, resolution, and desirable properties such as symmetry preservation, constitute future work.

Acknowledgements

The research of the first author is supported in part by NSFC grant 10572028, with additional support provided by the National Basic Research Program of China under grant 2005CB321702, by the Foundation of National Key Laboratory of Computational Physics under grant 9140C6902010603 and by the National Hi-Tech Inertial Confinement Fusion Committee of China. The research of the second author is supported

in part by NSFC grant 10671190 and by the Chinese Academy of Sciences during his visit to the University of Science and Technology of China and the Institute of Computational Mathematics and Scientific/Engineering Computing, with additional support provided by ARO grant W911NF-04-1-0291 and NSF grant DMS-0510345.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, Journal of Computational Physics 114 (1994) 45–58.
[2] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235–394.
[3] D.J. Benson, Momentum advection on a staggered mesh, Journal of Computational Physics 100 (1992) 143–162.
[4] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, Journal of Computational Physics 172 (2001) 739–765.
[5] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, Journal of Computational Physics 146 (1998) 227–262.
[6] J. Cheng and C.-W. Shu, A high order accurate conservative remapping method on staggered meshes, Applied Numerical Mathematics, in press, doi:10.1016/j.apnum.2007.04.015.
[7] B. Després, C. Mazeran, Lagrangian gas dynamics in two-dimensions and lagrangian systems, Archive for Rational Mechanics and Analysis 178 (2005) 327–372.
[8] J.K. Dukowicz, A general non-iterative Riemann solver for Godunov method, Journal of Computational Physics 61 (1985) 119–137.
[9] J.K. Dukowicz, M.C. Cline, F.L. Addessio, A General topology Godunov method, Journal of Computational Physics 82 (1989) 29–63.
[10] J.K. Dukowicz, B.J.A. Meltz, Vorticity errors in multi-dimensional Lagrangian codes, Journal of Computational Physics 99 (1992) 115–134.
[11] G. Erlebacher, M.Y. Hussaini, C.-W. Shu, Interaction of a shock with a longitudinal vortex, Journal of Fluid Mechanics 337 (1997) 129–153.
[12] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, Journal of Computational Physics 71 (1987) 231–303.
[13] A. Harten, S. Osher, Uniformly high-order accurate non-oscillatory schemes I, SIAM Journal on Numerical Analysis 24 (1987) 279–309.
[14] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, Journal of Computational Physics 14 (1974) 227–253.
[15] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.
[16] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Computer Methods in Applied Mechanics and Engineering 158 (1998) 81–116.
[17] B. Koobus, C. Farhat, Second-order time-accurate and geometrically conservative implicit schemes for flow computations on unstructured dynamic meshes, Computer Methods in Applied Mechanics and Engineering 170 (1999) 103–129.
[18] R. Loubère, M.J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, Journal of Computational Physics 209 (2005) 105–138.
[19] H. Luo, J.D. Baum, R. Löhner, On the computation of multi-material flows using ALE formulation, Journal of Computational Physics 194 (2004) 304–328.
[20] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM Journal on Scientific Computing 29 (2007) 1781–1824.
[21] L.G. Margolin, Introduction to An Arbitrary Lagrangian–Eulerian computing method for all flow speeds, Journal of Computational Physics 135 (1997) 198–202.
[22] C.D. Munz, On Godunov-type schemes for Lagrangian gas dynamics, SIAM Journal on Numerical Analysis 31 (1994) 17–42.
[23] J. von Neumann, R.D. Richtmyer, A method for the calculation of hydrodynamics shocks, Journal of Applied Physics 21 (1950) 232–237.
[24] W.F. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, Journal of Computational Physics 72 (1987) 78–120.
[25] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Computer Methods in Applied Mechanics and Engineering 187 (2000) 591–619.
[26] A. Rogerson, E. Meiberg, A numerical study of the convergence properties of ENO schemes, Journal of Scientific Computing 5 (1990) 151–167.
[27] L.I. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.
[28] C.-W. Shu, Numerical experiments on the accuracy of ENO and modified ENO schemes, Journal of Scientific Computing 5 (1990) 127–149.
[29] C.-W. Shu, Preface to the republication of "uniformly high order essentially non-oscillatory schemes, III", by Harten, Engquist, Osher, and Chakravarthy, Journal of Computational Physics 131 (1997) 1–2.
[30] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor (Eds.), Advanced Numerical Approximation of Non-linear Hyperbolic Equations, in: A. Quarteroni (Ed.), Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, 1998, pp. 325-432.

[31] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics 77 (1988) 439–471.
[32] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes II, Journal of Computational Physics 83 (1989) 32–78.
[33] C.-W. Shu, T.A. Zang, G. Erlebacher, D. Whitaker, S. Osher, High-order ENO schemes applied to two- and three-dimensional compressible flow, Applied Numerical Mathematics 9 (1992) 45–71.
[34] R.W. Smith, AUSM(ALE): a geometrically conservative arbitrary Lagrangian–Eulerian flux splitting scheme, Journal of Computational Physics 150 (1999) 268–286.
[35] H.Z. Tang, T.T. Liu, A note on the conservative schemes for the Euler equations, Journal of Computational Physics 218 (2006) 451–459.
[36] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin, 1999.
[37] P.R. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.