Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2017-0189

Vol. **24**, No. 5, pp. 1300-1325
November 2018

A Direct ALE Multi-Moment Finite Volume Scheme for the Compressible Euler Equations

Peng Jin[1], Xi Deng[1,*] and Feng Xiao[1]

[1] *Department of Mechanical Engineering, Tokyo Institute of Technology, 4259 Nagatsuta Midori-ku, Yokohama, 226-8502, Japan.*

Received 5 September 2017; Accepted (in revised version) 5 December 2017

Abstract. A direct Arbitrary Lagrangian Eulerian (ALE) method based on multi-moment finite volume scheme is developed for the Euler equations of compressible gas in 1D and 2D space. Both the volume integrated average (VIA) and the point values (PV) at cell vertices, which are used for high-order reconstructions, are treated as the computational variables and updated simultaneously by numerical formulations in integral and differential forms respectively. The VIAs of the conservative variables are solved by a finite volume method in the integral form of the governing equations to ensure the numerical conservativeness; whereas, the governing equations of differential form are solved for the PVs of the primitive variables to avoid the additional source terms generated from moving mesh, which largely simplifies the solution procedure. Numerical tests in both 1D and 2D are presented to demonstrate the performance of the proposed ALE scheme. The present multi-moment finite volume formulation consistent with moving meshes provides a high-order and efficient ALE computational model for compressible flows.

AMS subject classifications: 76M12, 76N15, 35L55

Key words: Compressible Euler equations, multi-moment finite volume method, direct ALE, Roe Riemann solver, HLLC Riemann solver, shock waves.

1 Introduction

Moving boundary problems exist widely in the practical applications of computational fluid dynamics (CFD), such as airfoil oscillations, flapping wings or fluid-structure interactions. When the boundary of fluid domain moves or undergoes deformations, the Arbitrary Lagrangian Eulerian method (ALE) [1,2] becomes a desirable numerical methodology to solve fluid dynamics in a moving and deforming grid which can fit the moving boundary of computational domain and get more accurate numerical solutions. In the

*Corresponding author. *Email addresses:* deng.x.aa@m.titech.ac.jp (X. Deng), jin.p.aa@m.titech.ac.jp (P. Jin), xiao@es.titech.ac.jp (F. Xiao)

ALE formulations, the grids of the computational domain can move arbitrarily independent of the fluid motions. This flexibility of mesh motion makes it more robust compared to the purely Lagrangian framework which may encounter difficulties when the mesh cells tangle or experience other topological changes due to flow fields of large distortions.

Efforts have been made so far to develop practical ALE algorithms. The existing works may be divided into two classes, i.e. indirect method and direct method. The indirect method was firstly proposed by Hirt et al. [1], which consists of three phases: (1) a Lagrangian phase where the solution and the grid are updated; (2) a rezoning phase that regularizes the tangled or heavily distorted mesh cells; (3) a remapping phase in which the Lagrangian solution is transferred to the rezoned mesh adjusted in step (2). However, rezoning and remapping procedures can be computationally expensive especially for two and three dimensional realistic calculations. The direct method [3, 4] does not need the remapping as a separate step because the mesh velocity is already taken into account in the numerical formulations that are consistent with the governing equations in the moving mesh framework. Both types of methods are widely implemented in the simulations of fluid dynamics.

In ALE schemes, the mesh geometry changes in time, which requires the geometrical quantities, such as volumes, boundary surfaces and vertices of moving cells, to be updated at each time step. Some existing works started from purely Lagrangian framework for developing mesh moving problems. Munz et al. [5] devised a cell-centered Godunov-type scheme using Roe [6] and HLL [7] flux solvers for Lagrangian hydrodynamics equations. Maire et al. [8–10] built a relationship between nodal displacement and flux formulation and presented a robust cell-centered Lagrangian method on multi-dimensional meshes with first and second order accuracy. To achieve high order accuracy, Cheng et al. [11] developed a class of Lagrangian type schemes based on high order essentially non-oscillatory (ENO) [12] reconstruction and obtained third order accuracy with curved mesh in [13]. Dumbser et al. [14] proposed a one-dimensional high order Lagrangian ADER (arbitrary high-order accurate) finite volume method. Besides, the Discontinuous Galerkin (DG) method has also been used to solve the gas dynamics equations in the total Lagrangian formulation for high-order accuracy [15]. These purely Lagrangian methods move mesh cells with the flow velocity so that the advection terms are removed from the governing equations. Particular care must be paid when the mesh is distorted to an unacceptable extent. A remedy is to improve the mesh quality with subsequent rezoning and remapping steps, which results in the indirect ALE method.

These Lagrangian methods can also be extended to direct ALE methods. High order discontinuous Galerkin (DG) direct ALE method has been implemented in [3, 16, 17] for compressible Euler equations. In the conventional finite volume framework, the direct ALE scheme can be constructed upon cell-centered Lagrangian solver completed with an edge-based upwinded formulation of the numerical fluxes as presented in [18]. Moreover, Boscheri et al. developed high-order direct ALE ADER finite volume schemes in [4, 19] with WENO (weighted essentially non-oscillatory) reconstruction and MOOD (multi-dimensional optimal order detection) approach. In this paper, we focus on direct

ALE method where the hydrodynamics equations are solved in an arbitrarily moving coordinate frame.

The ALE schemes based on conventional finite volume method face the same difficulties as their Eulerian versions do when one wants to generate high order reconstruction polynomials on unstructured meshes. It is not a trivial task to choose a reliable cell stencil for high order reconstruction on either stationary or moving unstructured meshes. The multi-moment type methods get around the problems by adding the degrees of freedoms (DOFs) locally on each cell. Numerical schemes with compact stencil provide a flexible and efficient way for implementing high order reconstructions such as the discontinuous Galerkin (DG) method [43, 44] and the spectral finite volume (SV) method [45, 46], as well as the multi-moment finite volume method [20–22]. Different from other existing methods, the multi-moment finite volume method makes use of the point values (PVs) at the vertices of mesh cells as the computational variables which are updated with the cell average value simultaneously. As shown later, it provides a new efficient ALE framework for fluid-structure interaction (FSI) problems. With the values at the nodes on fluid-structure interface directly updated, the multi-moment finite volume ALE is able to implement the FSI computations in a spirit similar to the finite element methodology formulated in [26].

In the Eulerian framework, the multi-moment constraint finite volume method based on local reconstruction achieved good accuracy and robustness on unstructured grids [20–22]. Being an option of practical significance, the polynomial interpolation is built by the VPM (volume integrated average (VIA) and point value (PV) based multi-moment) reconstruction to realize third order spatial discretization. In the VPM formulation, both VIA and PVs at cell vertices are treated as the computational variables, which are solved respectively by the integral and differential forms of the governing equations. Successive works have been done for implementing the VPM formulation to compressible flows [23–25], where the multi-dimensional limiting process (MLP) scheme [27] is used to limit the VPM reconstruction for suppressing the numerical oscillations in presence of discontinuous solutions, like shock waves or other jumps in physical fields. Updating simultaneously the VIA and PVs at cell vertices as computational variables through different but consistent numerical formulations, the multi-moment finite volume method provides an efficient formulation to construct high order ALE computational model as well.

In this study, we construct a direct ALE scheme based on the multi-moment constraint finite volume method. Because the VIA and PVs at the moving cell vertices are always known as the computational variables, local high-order reconstructions consistent with the moving mesh can be built easily in a straightforward way similar to the Eulerian framework. In the present ALE multi-moment finite volume scheme, we update the PV moment in terms of the primitive variables instead of the conservative variables, so as to circumvent the additional source terms as addressed in [28].

We present a direct ALE multi-moment finite volume scheme for compressible Euler equations in one and two dimensions. Roe's Riemann solver [6] is implemented pointwisely to update the PV moment, while HLLC Riemann solver [29] is used to update the

VIA moment in 2D and the numerical flux can be more simply calculated from the cell boundary PVs in 1D.

The rest of this paper is organized as follows. Section 2 gives the integral form and differential form of governing equations. In Section 3, we present the numerical scheme and benchmark tests in one-dimensional space. In Section 4, we give the numerical scheme and some tests for verifications in two-dimensional space. At last, we end the paper with some conclusion remarks in Section 5.

2 Governing equations

The numerical method developed in this paper is to solve the unsteady Euler equations for the inviscid compressible fluid. In the present multi-moment finite volume method, the numerical formulations to compute two kinds of moments, i.e. VIA and PV, are derived respectively from the integral and differential forms of Euler equations.

2.1 Integral form

The VIA is updated by a finite volume formulation where we use the integral form of Euler equations. We cast Euler equations in a moving control volume frame in the Cartesian coordinate as [11]

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

where $\Omega(t)$ is the moving control volume enclosed by its boundary $\Gamma(t)$. **U** and **F(U)** are the vectors of conservative variables and fluxes given below

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

with ρ, **u**, **M**, p and E being the density, velocity, momentum, pressure and specific total energy of the fluid respectively. In addition, **n** is the unit outward normal vector to $\Gamma(t)$ whose moving velocity is $\mathbf{u}_g$. If $\mathbf{u}_g=0$, the system reduces to a Eulerian framework, and if $\mathbf{u}_g=\mathbf{u}$, it corresponds to the Lagrangian framework where the control volume moves with the local fluid velocity.

The system is completely closed by the adding the equation of state (EOS), which establishes the relationship among three thermodynamic variables, in the following general form

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

where $e=E-\frac{1}{2}|\mathbf{u}|^2$ is the specific internal energy. We consider the ideal gas in this research, then EOS has a simpler form

$$p=(\gamma-1)\rho e, \tag{2.4}$$

where γ is the ratio of specific heats.

Besides, the volume variation is given by following geometric conservation law [30]

$$\frac{dV}{dt}-\int_{\Gamma(t)}\mathbf{u}_g\cdot\mathbf{n}d\Gamma=0, \tag{2.5}$$

where V denotes the cell volume.

2.2 Differential form

In the present work, differential form of compressible Euler equations is used to predict the PV moment, which is written in terms of primitive variables as follows

$$\left.\frac{\partial\mathbf{W}}{\partial t}\right|_{\chi}+A(\mathbf{W})\frac{\partial\mathbf{W}}{\partial x}+B(\mathbf{W})\frac{\partial\mathbf{W}}{\partial y}=0, \tag{2.6}$$

where $\mathbf{W}$ is the vector of primitive variables, $A(\mathbf{W})$ and $B(\mathbf{W})$ are the coefficient matrices for spatial derivatives of $\mathbf{W}$ with respect to x and y coordinates respectively. They take the following form

$$\begin{aligned}
&\mathbf{W}=\begin{bmatrix}\rho\\ u\\ v\\ p\end{bmatrix},\quad A(\mathbf{W})=\begin{bmatrix}u-u_g & \rho & 0 & 0\\ 0 & u-u_g & 0 & \frac{1}{\rho}\\ 0 & 0 & u-u_g & 0\\ 0 & \gamma p & 0 & u-u_g\end{bmatrix},\\
&B(\mathbf{W})=\begin{bmatrix}v-v_g & 0 & \rho & 0\\ 0 & v-v_g & 0 & 0\\ 0 & 0 & v-v_g & \frac{1}{\rho}\\ 0 & 0 & \gamma p & v-v_g\end{bmatrix},
\end{aligned} \tag{2.7}$$

where ρ is the density, p the pressure, u and v are the x and y components of fluid velocity $\mathbf{u}=(u,v)$, u_g and v_g the x and y components of the grid velocity $\mathbf{u}_g=(u_g,v_g)$ respectively. Note that in (2.6), $(\partial\mathbf{W}/\partial t)|_{\chi}$ denotes the time derivative with respect to the grid moving at speed $\mathbf{u}_g$. The grid velocity $\mathbf{u}_g$ is defined by

$$\mathbf{u}_g=\frac{d\mathbf{x}_g}{dt}, \tag{2.8}$$

where $\mathbf{x}_g$ is the coordinate of the grid. This is the so-called local kinematic equation, which corresponds to the geometric conservation law in the integral form. If $\mathbf{u}_g=0$, the system reduces to the differential form of Euler equations in the Eulerian framework and if $\mathbf{u}_g=\mathbf{u}$, the system becomes the Lagrangian framework.

3 Multi-moment finite volume ALE scheme in one dimension

The computational domain is divided into I non-overlapping cells $\Omega_i=[x_{i-\frac{1}{2}},x_{i+\frac{1}{2}}]$ of size $\Delta x_i=x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}$ with $i=1,2,\cdots,I$. For a given cell Ω_i, two kinds of moments are defined for field variable $\phi(x,t)$. They are the volume integrated average (VIA) over each cell

$$\overline{\phi}_i=\frac{1}{\Delta x_i}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\phi(x,t)dx, \tag{3.1}$$

and the point value (PV) at cell boundary

$$\phi_{i+\frac{1}{2}}=\phi(x_{i+\frac{1}{2}},t), \tag{3.2}$$

where ϕ stands for all state variables, including conservative and primitive variables.

3.1 The CIP-CSL3 reconstruction [31, 32]

Given one VIA $\overline{\phi}_i$ and two PVs $\phi_{i\pm\frac{1}{2}}$ over Ω_i, as well as the first order cell center derivative d_i approximated from the VIA and PV moments from Ω_i and its neighboring cells, a piecewise cubic interpolation function for conservative variables $\phi(x,t)$ can be constructed, following the CIP-CSL3 (constrained interpolation profile-conservative semi-Lagrangian scheme with third-order polynomial function) scheme, for each element as

$$\Phi_i(x)=a_3(x-x_{i-\frac{1}{2}})^3+a_2(x-x_{i-\frac{1}{2}})^2+a_1(x-x_{i-\frac{1}{2}})+a_0, \tag{3.3}$$

and the unknown coefficients are computed by imposing the following constraint conditions:

$$\begin{cases}\Phi_i(x_{i-\frac{1}{2}})=\phi_{i-\frac{1}{2}},\\ \Phi_i(x_{i+\frac{1}{2}})=\phi_{i+\frac{1}{2}},\\ \dfrac{1}{\Delta x_i}\displaystyle\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\Phi_i(x)dx=\overline{\phi}_i,\\ \dfrac{d\Phi_i(x)}{dx}\Big|_{x_i}=d_i.\end{cases} \tag{3.4}$$

Thus, we have coefficients of (3.3) as

$$\begin{cases}a_0=\phi_{i-\frac{1}{2}},\\ a_1=\dfrac{2(3\overline{\phi}_i-3\phi_{i-\frac{1}{2}}-\Delta x_i d_i)}{\Delta x_i},\\ a_2=\dfrac{3(-2\overline{\phi}_i-\phi_{i+\frac{1}{2}}+3\phi_{i-\frac{1}{2}}+2\Delta x_i d_i)}{\Delta x_i^2},\\ a_3=\dfrac{4(\phi_{i+\frac{1}{2}}-\phi_{i-\frac{1}{2}}-\Delta x_i d_i)}{\Delta x_i^3}.\end{cases} \tag{3.5}$$

When the cell center derivative d_i is simply computed by differencing the PVs at the two boundaries of cell Ω_i as

$$d_i=\frac{\phi_{i+\frac{1}{2}}-\phi_{i-\frac{1}{2}}}{\Delta x_i}, \tag{3.6}$$

the coefficient of the third-order term a_3 vanishes in (3.5), and the interpolation function then degrades to a piecewise quadratic interpolation function for cell Ω_i and the CIP-CSL2 (constrained interpolation profile-conservative semi-Lagrangian scheme with second-order polynomial function) scheme [33] is retrieved. As shown in [31,32], the cell center derivative can be approximated by conventional slope limiter schemes to suppress the numerical oscillations. We use the superbee limiter [34] as follows

$$d_i=\text{maxmod}\left(\text{minmod}\left(d_{i-\frac{1}{2}},2d_{i+\frac{1}{2}}\right),\text{minmod}\left(2d_{i-\frac{1}{2}},d_{i+\frac{1}{2}}\right)\right), \tag{3.7}$$

where

$$d_{i-\frac{1}{2}}=\frac{2(\overline{\phi}_i-\overline{\phi}_{i-1})}{\Delta x_{i-1}+\Delta x_i},\quad d_{i+\frac{1}{2}}=\frac{2(\overline{\phi}_{i+1}-\overline{\phi}_i)}{\Delta x_i+\Delta x_{i+1}}.$$

The minmod and maxmod functions are given respectively as

$$\text{minmod}(a_1,a_2)=\begin{cases} s\cdot\min(|a_1|,|a_2|), & \text{if } s=\text{sign}(a_1)=\text{sign}(a_2),\\ 0, & \text{otherwise,}\end{cases} \tag{3.8}$$

and

$$\text{maxmod}(a_1,a_2)=\begin{cases} s\cdot\max(|a_1|,|a_2|), & \text{if } s=\text{sign}(a_1)=\text{sign}(a_2),\\ 0, & \text{otherwise.}\end{cases} \tag{3.9}$$

As long as the CIP-CSL3 interpolation is constructed over Ω_i, we can obtain the spatial derivatives of the physical variables at two boundary points $x_{i\pm\frac{1}{2}}$ which are then used to update the PVs at cell boundaries.

3.2 Solution of Euler equation for VIA moment in one dimension

We first formulate the semi-discrete finite volume scheme of (2.1) for updating VIA moments as

$$\frac{d}{dt}\begin{pmatrix}\overline{\rho}_i\Delta x_i\\ \overline{M}_i\Delta x_i\\ \overline{(\rho E)}_i\Delta x_i\end{pmatrix}=-\begin{pmatrix}\hat{f}^D_{i+\frac{1}{2}}-\hat{f}^D_{i-\frac{1}{2}}\\ \hat{f}^M_{i+\frac{1}{2}}-\hat{f}^M_{i-\frac{1}{2}}\\ \hat{f}^E_{i+\frac{1}{2}}-\hat{f}^E_{i-\frac{1}{2}}\end{pmatrix}, \tag{3.10}$$

with the fluxes approximated from PVs as

$$\begin{cases}\hat{f}^D_{i+\frac{1}{2}}=(u_{i+\frac{1}{2}}-u_{g,i+\frac{1}{2}})\rho_{i+\frac{1}{2}},\\ \hat{f}^M_{i+\frac{1}{2}}=(u_{i+\frac{1}{2}}-u_{g,i+\frac{1}{2}})M_{i+\frac{1}{2}}+p_{i+\frac{1}{2}},\\ \hat{f}^E_{i+\frac{1}{2}}=(u_{i+\frac{1}{2}}-u_{g,i+\frac{1}{2}})(\rho E)_{i+\frac{1}{2}}+p_{i+\frac{1}{2}}u_{i+\frac{1}{2}}.\end{cases} \tag{3.11}$$

It is noted that by solving the physical variables as computational variables at cell boundaries we don't need to solve any Riemann problem to find the numerical fluxes at cell boundaries which are required in the finite volume formulations for the VIAs of the conservative variables. Thus, the present method is computationally efficient.

3.3 Solution of Euler equation for PV moment in one dimension

The governing equations (2.6) of PVs in one-dimensional case are formulated by

$$\frac{\partial \mathbf{W}}{\partial t}+A(\mathbf{W})\frac{\partial \mathbf{W}}{\partial x}=0, \tag{3.12}$$

with $\mathbf{W}=[\rho,u,p]$ being the primitive variables. The Jacobian matrix can be factorized into $A=R_A\Lambda_A L_A$, where Λ_A is the diagonal matrix of eigenvalues, and L_A and R_A are the corresponding left and right eigen matrices respectively.

As introduced in [24], (3.12) is solved by Roe's Riemann solver point-wisely as

$$\frac{\partial \mathbf{W}_{i+\frac{1}{2}}}{\partial t}=-\frac{1}{2}\left(\tilde{A}\left(\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x}+\frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}\right)+\tilde{R}_A|\tilde{\Lambda}_A|\tilde{L}_A\left(\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x}-\frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}\right)\right), \tag{3.13}$$

where $\frac{\partial \mathbf{W}^-_{i+\frac{1}{2}}}{\partial x}$ and $\frac{\partial \mathbf{W}^+_{i+\frac{1}{2}}}{\partial x}$ denote the derivatives of variables $\mathbf{W}_{i+\frac{1}{2}}$ on the left and right sides of point $x_{i+\frac{1}{2}}$ and the Jacobian matrix is calculated from the Roe-averaging. Each Roe-averaging value is computed from the VIA values in the surrounding cells by

$$\tilde{\phi}_{i+\frac{1}{2}}=\frac{\sqrt{\overline{\rho}_i}\cdot\overline{\phi}_i+\sqrt{\overline{\rho}_{i+1}}\cdot\overline{\phi}_{i+1}}{\sqrt{\rho_i}+\sqrt{\rho_{i+1}}}, \tag{3.14}$$

where $\tilde{\phi}_{i+\frac{1}{2}}$ stands for $\tilde{u}_{i+\frac{1}{2}}$ and $\tilde{H}_{i+\frac{1}{2}}$ and $\overline{H}_i=\left(\overline{(\rho E)}_i+\overline{p}_i\right)/\overline{\rho}_i$ is the total enthalpy. We compute the sound speed by

$$\tilde{a}_{i+\frac{1}{2}}=\left((\gamma-1)(\tilde{H}_{i+\frac{1}{2}}-\frac{1}{2}\tilde{u}^2_{i+\frac{1}{2}})\right)^{\frac{1}{2}}. \tag{3.15}$$

As shown in Subsection 3.1, the interpolation function is reconstructed for the conservative variables ρ, M and ρE, thus the spatial derivatives of conservative variables must be transferred to the spatial derivatives of primitive variables $\mathbf{W}=[\rho,u,p]$. We use the following formulations for this purpose in this work

$$\frac{\partial u^{\pm}_{i+\frac{1}{2}}}{\partial x}=\frac{1}{\tilde{\rho}_{i+\frac{1}{2}}}\frac{\partial M^{\pm}_{i+\frac{1}{2}}}{\partial x}-\frac{\tilde{M}_{i+\frac{1}{2}}}{\tilde{\rho}^2_{i+\frac{1}{2}}}\frac{\partial \rho^{\pm}_{i+\frac{1}{2}}}{\partial x}, \tag{3.16}$$

$$\frac{\partial p^{\pm}_{i+\frac{1}{2}}}{\partial x}=(\gamma-1)\left(\frac{\partial (\rho E)^{\pm}_{i+\frac{1}{2}}}{\partial x}-\tilde{u}_{i+\frac{1}{2}}\frac{\partial M^{\pm}_{i+\frac{1}{2}}}{\partial x}+\frac{1}{2}\tilde{u}^2_{i+\frac{1}{2}}\frac{\partial \rho^{\pm}_{i+\frac{1}{2}}}{\partial x}\right). \tag{3.17}$$

The semi-discrete time evolution equations (3.13) are then solved to update the PVs at cell boundaries.

3.4 Time discretization

In order to achieve third order accuracy in time, we implement the third order Runge-Kutta scheme [35] to update the semi-discrete equations (3.10) and (3.13) in time. Since the mesh changes with the time in ALE scheme, not only VIAs and PVs but also the position of each vertex and the size of each cell should be updated in each Runge-Kutta step.

We summarize the solution procedure using the third order Runge-Kutta scheme to update the numerical solutions of Euler equations from time level n ($t=t^n$) to $n+1$ ($t=t^n+\Delta t$) as follows,

Step 1:

$$x_{i+\frac{1}{2}}^{(1)}=x_{i+\frac{1}{2}}^{n}+u_{i+\frac{1}{2}}^{n}\Delta t,\quad \Delta x_i^{(1)}=x_{i+\frac{1}{2}}^{(1)}-x_{i-\frac{1}{2}}^{(1)},$$
$$\overline{\mathbf{U}}_i^{(1)}\Delta x_i^{(1)}=\overline{\mathbf{U}}_i^{n}\Delta x_i^{n}+\Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{n},\mathbf{W}_{i+\frac{1}{2}}^{n}),$$
$$\mathbf{W}_{i+\frac{1}{2}}^{(1)}=\mathbf{W}_{i+\frac{1}{2}}^{n}+\Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{n},\mathbf{W}_{i+\frac{1}{2}}^{n});$$

Step 2:

$$x_{i+\frac{1}{2}}^{(2)}=\frac{3}{4}x_{i+\frac{1}{2}}^{n}+\frac{1}{4}\left[x_{i+\frac{1}{2}}^{(1)}+u_{i+\frac{1}{2}}^{(1)}\Delta t\right],\quad \Delta x_i^{(2)}=x_{i+\frac{1}{2}}^{(2)}-x_{i-\frac{1}{2}}^{(2)},$$
$$\overline{\mathbf{U}}_i^{(2)}\Delta x_i^{(2)}=\frac{3}{4}\overline{\mathbf{U}}_i^{n}\Delta x_i^{n}+\frac{1}{4}\left[\overline{\mathbf{U}}_i^{(1)}\Delta x_i^{1}+\Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{(1)},\mathbf{W}_{i+\frac{1}{2}}^{(1)})\right],$$
$$\mathbf{W}_{i+\frac{1}{2}}^{(2)}=\frac{3}{4}\mathbf{W}_{i+\frac{1}{2}}^{n}+\frac{1}{4}\left[\mathbf{W}_{i+\frac{1}{2}}^{(1)}+\Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{(1)},\mathbf{W}_{i+\frac{1}{2}}^{(1)})\right];$$

Step 3:

$$x_{i+\frac{1}{2}}^{n+1}=\frac{1}{3}x_{i+\frac{1}{2}}^{n}+\frac{2}{3}\left[x_{i+\frac{1}{2}}^{(2)}+u_{i+\frac{1}{2}}^{(2)}\Delta t\right],\quad \Delta x_i^{n+1}=x_{i+\frac{1}{2}}^{n+1}-x_{i-\frac{1}{2}}^{n+1},$$
$$\overline{\mathbf{U}}_i^{n+1}\Delta x_i^{n+1}=\frac{1}{3}\overline{\mathbf{U}}_i^{n}\Delta x_i^{n}+\frac{2}{3}\left[\overline{\mathbf{U}}_i^{(2)}\Delta x_i^{(2)}+\Delta t\mathbf{R}_1(\overline{\mathbf{U}}_i^{(2)},\mathbf{W}_{i+\frac{1}{2}}^{(2)})\right],$$
$$\mathbf{W}_{i+\frac{1}{2}}^{n+1}=\frac{1}{3}\mathbf{W}_{i+\frac{1}{2}}^{n}+\frac{2}{3}\left[\mathbf{W}_{i+\frac{1}{2}}^{(2)}+\Delta t\mathbf{R}_2(\overline{\mathbf{U}}_i^{(2)},\mathbf{W}_{i+\frac{1}{2}}^{(2)})\right],$$

where $\mathbf{R}_1$ and $\mathbf{R}_2$ stand for the numerical operators of the right sides of (3.10) and (3.13) respectively. The time step is simply determined by

$$\Delta t=C_E\min_{1\leqslant i\leqslant I}\left(\frac{\Delta x_i^n}{|u_i^n|+c_i^n}\right),\tag{3.18}$$

where we set $C_E=0.4$ in our computations except for the blast wave problem.

3.5 Numerical results in one dimension

In this section, one-dimensional ALE multi-moment finite volume scheme for Euler equations is tested by some benchmark tests. To verify that the mesh can move arbitrarily for the proposed ALE scheme, we set the grid velocity as a fraction of the local fluid velocity in this paper. With $u_g=0$ for Eulerian framework and $u_g=u$ for Lagrangian framework, we include the case of $u_g=0.5u$ to demonstrate the present ALE framework with a specified speed to move the mesh. It is noted that numerical formulation of the present ALE works well for any specified mesh speed. The ideal gas with $\gamma=1.4$ is used unless otherwise stated and also for later 2D tests.

3.5.1 Accuracy test

The accuracy of the proposed scheme is evaluated by a benchmark test with smooth solutions [11]. The initial conditions are

$$\rho(x,0)=2+\sin(2\pi x),\quad u(x,0)=1+0.1\sin(2\pi x),\quad p(x,0)=1,\quad x\in[0,1], \tag{3.19}$$

and the periodic boundary condition is used. We tested the present ALE scheme in the Eulerian and Lagrangian frameworks as two special cases and also another moving-mesh case with the grid velocity specified as half of the fluid velocity ($u_g=0.5u$). We summarize in Table 1 the errors and numerical convergence rates for the density field at $t=1$ with the reference solution calculated from the third order multi-moment constraint finite volume method (MCV) [20] in the Eulerian framework with 8000 grids. To show the accuracy improvements of the present multi-moment ALE scheme in comparison with the conventional finite volume ALE method, we also calculated the same benchmark test by a conventional third-order finite volume ALE scheme using the quadratic polynomials constructed over a three-cell stencil. Numerical results of third-order finite volume ALE scheme are shown in Table 2 for comparison, and we compare the computation time of both schemes in Table 3.

It is observed that the present ALE scheme can achieve third order accuracy, and the numerical errors are clearly smaller than the third order conventional finite volume method. Though two kinds of moment are used, the computational cost does not increase

Table 1: Errors and convergence rates of density with initially uniform cells by using the present ALE scheme.

cells	Eulerian ($u_g=0$)				ALE ($u_g=0.5u$)				Lagrangian ($u_g=u$)			
	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate
50	1.96e-4	-	7.70e-4	-	1.06e-4	-	5.25e-4	-	6.88e-5	-	3.12e-4	-
100	2.52e-5	2.96	1.04e-4	2.89	1.39e-5	2.93	7.00e-5	2.91	8.88e-6	2.95	4.17e-5	2.90
200	3.16e-6	3.00	1.32e-5	2.98	1.74e-6	3.00	8.97e-6	2.96	1.11e-6	3.00	5.36e-6	2.96
400	3.96e-7	3.00	1.66e-6	2.99	2.19e-7	2.99	1.13e-6	2.99	1.40e-7	2.99	6.76e-7	2.99
800	4.94e-8	3.00	2.08e-7	3.00	2.73e-8	3.00	1.41e-7	3.00	1.74e-8	3.01	8.46e-8	3.00

Table 2: Errors and convergence rates of density with initially uniform cells by using 3rd-order finite volume ALE scheme.

cells	Eulerian ($u_g=0$)				ALE ($u_g=0.5u$)				Lagrangian ($u_g=u$)			
	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate
50	1.12e-3	-	4.34e-3	-	6.33e-4	-	3.11e-3	-	3.82e-4	-	1.68e-3	-
100	1.46e-4	2.93	5.95e-4	2.87	8.34e-5	2.93	4.17e-4	2.90	4.93e-5	2.95	2.59e-4	2.70
200	1.84e-5	2.99	7.54e-5	2.98	1.05e-5	2.99	5.27e-5	2.98	6.23e-6	2.98	3.36e-5	2.95
400	2.28e-6	3.01	9.41e-6	3.00	1.32e-6	3.00	6.59e-6	3.00	7.82e-7	2.99	4.25e-6	2.98
800	2.54e-7	3.17	1.18e-6	3.00	1.67e-7	2.98	8.56e-7	2.95	1.02e-7	2.94	5.63e-7	2.91

Table 3: Computational time comparison of the present ALE scheme and 3rd-order finite volume ALE scheme. The elapse time of computations was measured on a PC with an Intel Core i7-4790 CPU @ 3.60GHz. 500,1000,2000,4000 and 8000 time-steps were used respectively for the cases of 50,100,200,400 and 800 cells.

cells	The present scheme			3rd-order FVM		
	Eulerian ($u_g=0$)	ALE ($u_g=0.5u$)	Lagrangian ($u_g=u$)	Eulerian ($u_g=0$)	ALE ($u_g=0.5u$)	Lagrangian ($u_g=u$)
50	0.05s	0.04s	0.04s	0.04s	0.04s	0.03s
100	0.15s	0.15s	0.14s	0.11s	0.12s	0.12s
200	0.55s	0.57s	0.55s	0.45s	0.46s	0.45s
400	2.15s	2.19s	2.17s	1.79s	1.78s	1.77s
800	8.52s	8.53s	8.57s	6.90s	7.01s	7.08s

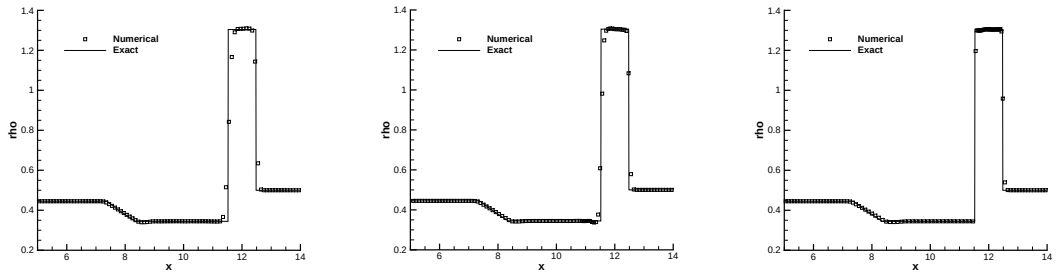
substantially. The Lagrangian framework of both methods shows the smallest numerical errors compared to the other two frameworks.

3.5.2 Lax problem

Known as Lax's Riemann problem for Euler equations, this benchmark test starts from the following initial conditions,

$$(\rho_0,u_0,p_0)=\begin{cases}(0.445,0.698,3.528), & \text{for } 4\leqslant x\leqslant 10,\\ (0.5,0,0.571), & \text{for } 10<x\leqslant 14.\end{cases} \tag{3.20}$$

Fig. 1 shows the results of the Eulerian framework (a), the ALE framework (b) and the Lagrangian framework (c) with 100 initially uniform cells at time $t=1$. The present scheme can resolve the discontinuous solution well with the superbee limiter. We can see from Fig. 1 that the Lagrangian framework can capture sharper contact discontinuity than the Eulerian framework. The ALE scheme with $u_g=0.5u$ gives numerical results lying between Eulerian and Lagrangian frameworks.



(a) (b) (c)

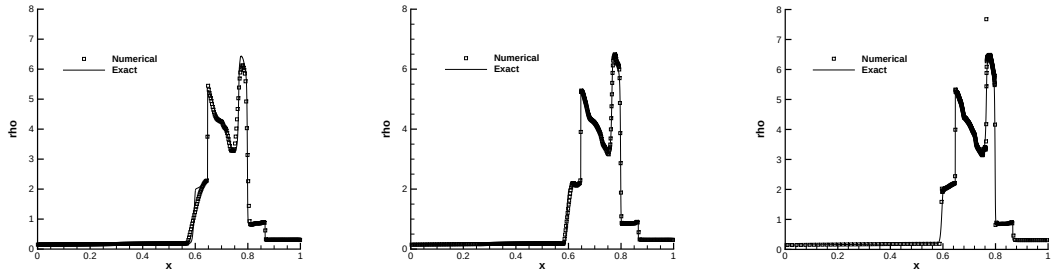
Figure 1: The density results for Lax problem with the present ALE scheme: (a) Eulerian ($u_g=0$); (b) ALE ($u_g=0.5u$); (c) Lagrangian ($u_g=u$).

3.5.3 Two interacting blast waves

The benchmark test of two interacting blast waves [36] was computed. The initial conditions were specified as

$$\rho_0=1,\ u_0=1,\ p_0=\begin{cases}1000, & \text{for } 0\leqslant x\leqslant 0.1,\\ 0.01, & \text{for } 0.1<x\leqslant 0.9,\\ 100, & \text{for } 0.9<x\leqslant 1.\end{cases} \tag{3.21}$$

The reflective boundary conditions were applied at the two ends of computation domain, x=0 and x=1. We solved this problem with 400 initially uniform cells to time t=0.38, and the CFL number is set as 0.1. The density results of the Eulerian, ALE and Lagrangian framework are plotted in Fig. 2. The left-most contact discontinuity in the Eulerian framework is diffused as observed in the numerical solutions of other high-resolution schemes, while the numerical solutions of the Lagrangian framework demonstrate much supe-



(a) (b) (c)

Figure 2: The density results for blast waves with the present ALE scheme: (a) Eulerian ($u_g=0$); (b) ALE ($u_g=0.5u$); (c) Lagrangian ($u_g=u$).

rior capability in resolving both discontinuities and smooth solutions. The Lagrangian framework can resolve the contact discontinuity within two cells, which is very challenging to Eulerian sort approach even with high-order reconstruction. Meanwhile, we also observe some overshoots in the Lagrangian framework quite similar to that reported in [11]. Again, the ALE result with $u_g=0.5u$ shows a result lying between Eulerian and Lagrangian frameworks.

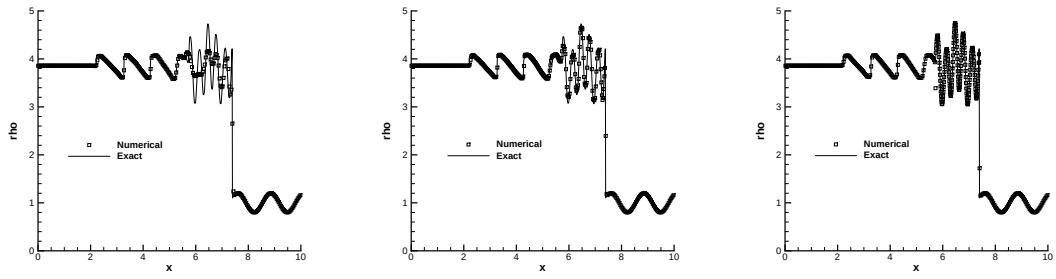
3.5.4 Shock entropy wave interaction

This problem was proposed in [37]. The initial conditions are given as

$$(\rho_0,u_0,p_0)=\begin{cases}(3.857148,2.629369,10.333333), & \text{for } -5\leqslant x\leqslant 1,\\ (1+0.2\sin(5x-5),0,1), & \text{for } 1<x\leqslant 10.\end{cases} \tag{3.22}$$

The numerical results of density at time $t=1.8$ with 400 initially uniform mesh are displayed in Fig. 3. The Lagrangian framework resolves finer structure in the numerical solutions which are much superior to the Eulerian framework. The ALE result with $u_g=0.5u$ shows good results for both shocks and density perturbations, and performs as a balanced solver between Eulerian and Lagrangian frameworks as expected.

In this section, a direct ALE multi-moment finite volume scheme is developed for one-dimensional Euler equations. It can achieve third order accuracy for smooth solutions and can successfully solve the 1D compressible Euler equations include discontinuous solutions on moving meshes. The Lagrangian framework can capture the discontinuities more sharply than the Eulerian framework.



(a) (b) (c)

Figure 3: The density results for shock-turbulence interaction test with the present ALE scheme: (a) Eulerian ($u_g=0$); (b) ALE ($u_g=0.5u$); (c) Lagrangian ($u_g=u$).

4 Multi-moment finite volume ALE scheme in two dimension

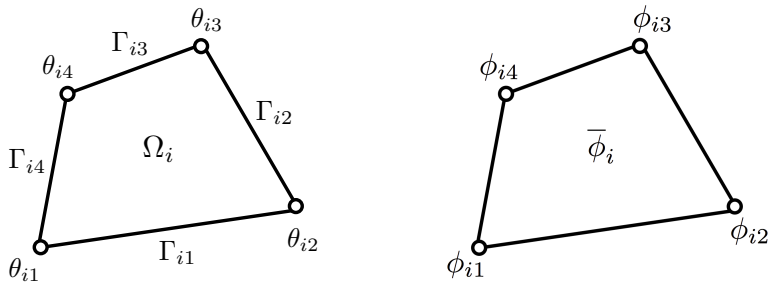
The 1D ALE formulation presented in the previous section can be extended straightforwardly to 2D case. In the present 2D multi-moment finite volume scheme, we solve the

PV at each cell vertex as the computational variable as well in addition to the VIA moment. Since the vertex is moved at a specified velocity, the basic idea developed in the previous section can be applied to update the two kinds of moments in 2D case.

4.1 The grid configuration and notations

In this study, 2D computational domain is divided into I non-overlapping straight-line quadrilateral elements, denoted by Ω_i, $i=1,2,\cdots,I$. The nodes (vertices of element cells) are denoted by θ_k, $k=1,2,\cdots,K$.

For convenience of local reconstruction, we further denote the four vertices of cell Ω_i by θ_{im} $(m=1,2,3,4)$ with the coordinate (x_{im},y_{im}), and its four boundary line segments by $\Gamma_{i1}=\overline{\theta_{i1}\theta_{i2}}$, $\Gamma_{i2}=\overline{\theta_{i2}\theta_{i3}}$, $\Gamma_{i3}=\overline{\theta_{i3}\theta_{i4}}$ and $\Gamma_{i4}=\overline{\theta_{i4}\theta_{i1}}$ as shown in Fig. 4 (a). The outward unit normal vector and length magnitude of segment Γ_{ij}, $j=1,2,3,4$, are $\mathbf{n}_{ij}=(n_{xij},n_{yij})$ and $|\Gamma_{ij}|$ respectively. We use the symbol Ω_{ij} to denote the surrounding cells adjacent to the target cell Ω_i with the common boundary segment Γ_{ij}. The four cells that share node θ_k are denoted by Ω_{km}, $m=1,2,3,4$ respectively. We also denote the mass center and element area of Ω_i by θ_{ic} and V_i respectively.



(a) (b)

Figure 4: The quadrilateral mesh element (a) and the definition of discrete moments (b).

Shown in Fig. 4 (b), the computational variables, i.e. the volume integrated averages (VIA) over Ω_i and the point values (PV) at the vertices of cell Ω_i are defined as

$$\overline{\phi}_i=\frac{1}{V_i}\int_{\Omega_i}\phi(x,y,t)d\Omega, \quad i=1,2,\cdots,I, \tag{4.1}$$

$$\phi_{im}=\phi(x_{im},y_{im},t), \qquad m=1,2,3,4, \tag{4.2}$$

where ϕ stands for all state variables same as that in 1D.

4.2 Multi-moment reconstruction function

With VIA and PV moments in each cell, high order polynomials are reconstructed for conservative variables $\mathbf{U}$. For algorithmic simplicity, following [21,22,38], we transform

the quadrilateral cell Ω_i on physical coordinates $\boldsymbol{x}=(x,y)$ to a standard square element $[-1\leqslant\xi\leqslant 1,-1\leqslant\eta\leqslant 1]$ on local coordinate $\boldsymbol{\xi}=(\xi,\eta)$ with the transformation

$$\boldsymbol{x}=\sum_{m=1}^{4}\boldsymbol{x}_{im}\mathcal{N}_{im}, \tag{4.3}$$

where

$$\mathcal{N}_{i1}=\frac{1}{4}(1-\xi-\eta+\xi\eta),\quad \mathcal{N}_{i2}=\frac{1}{4}(1+\xi-\eta-\xi\eta),$$
$$\mathcal{N}_{i3}=\frac{1}{4}(1+\xi+\eta+\xi\eta),\quad \mathcal{N}_{i4}=\frac{1}{4}(1-\xi+\eta-\xi\eta).$$

As detailed in [21, 22, 38], the first-order and second-order derivatives of the physical field are firstly computed in the physical coordinates and then transformed to the local coordinates. With the supplementary of the first and second order derivatives at cell center $(\phi_{\xi ic},\phi_{\eta ic})$ and $(\phi_{\xi^2 ic},\phi_{\eta^2 ic})$, the reconstruction polynomial Φ for conservative variable $\phi(\xi,\eta)$ can be formulated on the local coordinates as

$$\Phi_i(\phi,\boldsymbol{\xi})=\sum_{m=1}^{4}\psi_m\phi_{im}+\overline{\psi}\,\overline{\phi}_i+\psi_\xi\phi_{\xi ic}+\psi_\eta\phi_{\eta ic}+\psi_{\xi^2}\phi_{\xi^2 ic}+\psi_{\eta^2}\phi_{\eta^2 ic}, \tag{4.4}$$

where $\psi_m,\overline{\psi},\psi_\xi,\psi_\eta,\psi_{\xi^2},\psi_{\eta^2}$ denote the basis functions of the corresponding DOFs. Following the limiting procedure in [23, 24], the high order reconstruction is degraded to a linear polynomial by MLP [27] scheme to eliminate spurious oscillations. The first-order derivatives at vertices are computed from the reconstruction functions in local coordinates and then transformed back to the physical coordinates.

Given the spatial derivatives of physical field $\mathbf{U}$ at point k, $(\frac{\partial \mathbf{U}_{km}}{\partial x},\frac{\partial \mathbf{U}_{km}}{\partial y})$, which are calculated from the surrounding cells Ω_{km} $(m=1,2,3,4)$, we approximate the upwind-biased derivatives with respect to x and y at vertex point k by

$$\frac{\partial \mathbf{U}_k^{\pm}}{\partial x}=\sum_{m=1}^{4}\omega_{km}^{x\pm}\frac{\partial \mathbf{U}_{km}}{\partial x}\quad\text{and}\quad\frac{\partial \mathbf{U}_k^{\pm}}{\partial y}=\sum_{m=1}^{4}\omega_{km}^{y\pm}\frac{\partial \mathbf{U}_{km}}{\partial y}, \tag{4.5}$$

respectively. The weights for derivatives on left and right sides of point k in x and y directions are calculated by

$$\omega_{km}^{\tau\pm}=\frac{\max(0,\mathbf{n}^{\tau\pm}\cdot\overrightarrow{\theta_{kmc}\theta_k})}{\sum_{m=1}^{4}\max(0,\mathbf{n}^{\tau\pm}\cdot\overrightarrow{\theta_{kmc}\theta_k})}, \tag{4.6}$$

where $\overrightarrow{\theta_{kmc}\theta_k}$ denotes the vector from the mass center θ_{kmc} to vertex θ_k, and the unit normal vector $\mathbf{n}^{\tau\pm}$ represents $\mathbf{n}^{x\pm}(\pm1,0)$ and $\mathbf{n}^{y\pm}(0,\pm1)$ respectively.

4.3 Solutions of Euler equation for VIA moment in two dimension

We formulate the semi-discrete finite volume scheme (2.1) in 2D with a Q-point Gaussian quadrature for numerical fluxes as

$$\frac{d}{dt}\left(\overline{\mathbf{U}}_i V_i\right) = -\sum_{j=1}^{4}\sum_{q=1}^{Q}\omega_q \tilde{\mathbf{F}}_{ij}(G_q,t)|\Gamma_{ij}|, \tag{4.7}$$

where G_q is the Gaussian quadrature point on cell edge Γ_{ij} which separates cells Ω_i and Ω_{ij}. The four-point Gauss-Lobatto integral formula is used in this work, which reads

$$\begin{aligned}
&G_1 = P_1, \quad G_2 = \frac{1}{2}(P_1+P_2) - \frac{\sqrt{5}}{10}(P_2-P_1),\\
&G_3 = \frac{1}{2}(P_1+P_2) + \frac{\sqrt{5}}{10}(P_2-P_1), \quad G_4 = P_2,\\
&\omega_1 = \omega_4 = \frac{1}{12}, \quad \omega_2 = \omega_3 = \frac{5}{12},
\end{aligned}$$

for the edge with endpoints P_1 and P_2.

The HLLC scheme for moving grids [39] is implemented in this study to calculate flux $\tilde{\mathbf{F}}_{ij}$ on edge Γ_{ij}, which is denoted by $\mathbf{F}_{ij}^{HLLC}$ and formulated as

$$\mathbf{F}_{ij}^{HLLC} = \begin{cases} \mathbf{F}(\mathbf{U}_i), & S_i > 0,\\ \mathbf{F}(\mathbf{U}_i^\star), & S_i \leqslant 0 < S^\star,\\ \mathbf{F}(\mathbf{U}_{ij}^\star), & S^\star \leqslant 0 \leqslant S_{ij},\\ \mathbf{F}(\mathbf{U}_{ij}), & S_{ij} < 0, \end{cases} \tag{4.8}$$

where

$$\mathbf{U}_i^\star = \begin{pmatrix} \rho_i^\star \\ \mathbf{M}_i^\star \\ (\rho E)_i^\star \end{pmatrix} = \frac{1}{S_i - S^\star}\begin{pmatrix} (S_i - u_{n_i})\rho_i \\ (S_i - u_{n_i})\mathbf{M}_i + (p^\star - p_i)\mathbf{n} \\ (S_i - u_{n_i})(\rho E)_i - p_i u_{n_i} + p^\star S^\star \end{pmatrix}, \tag{4.9}$$

$$\mathbf{U}_{ij}^\star = \begin{pmatrix} \rho_{ij}^\star \\ \mathbf{M}_{ij}^\star \\ (\rho E)_{ij}^\star \end{pmatrix} = \frac{1}{S_{ij} - S^\star}\begin{pmatrix} (S_{ij} - u_{n_{ij}})\rho_{ij} \\ (S_{ij} - u_{n_{ij}})\mathbf{M}_{ij} + (p^\star - p_{ij})\mathbf{n} \\ (S_{ij} - u_{n_{ij}})(\rho E)_{ij} - p_{ij} u_{n_{ij}} + p^\star S^\star \end{pmatrix}, \tag{4.10}$$

$$\mathbf{F}(\mathbf{U}_i^\star) = \begin{pmatrix} S^\star \rho_i^\star \\ S^\star \mathbf{M}_i^\star + p^\star \mathbf{n} \\ S^\star (\rho E)_i^\star + (S^\star + \mathbf{u}_{gij}\cdot\mathbf{n})p^\star \end{pmatrix}, \tag{4.11}$$

$$\mathbf{F}(\mathbf{U}_{ij}^{\star})=\begin{pmatrix} S^{\star}\rho_{ij}^{\star} \\ S^{\star}\mathbf{M}_{ij}^{\star}+p^{\star}\mathbf{n} \\ S^{\star}(\rho E)_{ij}^{\star}+(S^{\star}+\mathbf{u}_{gij}\cdot\mathbf{n})p^{\star} \end{pmatrix}, \tag{4.12}$$

$$p^{\star}=\rho_i(u_{n_i}-S_i)(u_{n_i}-S^{\star})+p_i=\rho_{ij}(u_{n_{ij}}-S_{ij})(u_{n_{ij}}-S^{\star})+p_{ij}, \tag{4.13}$$

$$u_{n_i}=(\mathbf{u}_i-\mathbf{u}_{gij})\cdot\mathbf{n}, \quad u_{n_{ij}}=(\mathbf{u}_{ij}-\mathbf{u}_{gij})\cdot\mathbf{n}, \tag{4.14}$$

$$S^{\star}=\frac{\rho_{ij}u_{n_{ij}}(S_{ij}-u_{n_{ij}})-\rho_i u_{n_i}(S_i-u_{n_i})+p_i-p_{ij}}{\rho_{ij}(S_{ij}-u_{n_{ij}})-\rho_i(S_i-u_{n_i})}. \tag{4.15}$$

Signal velocities S_i and S_{ij} are defined as

$$S_i=\min\left[u_{n_i}-c_i,(\hat{\mathbf{u}}-\mathbf{u}_{gij})\cdot\mathbf{n}-\hat{c}\right], \quad S_{ij}=\max\left[u_{n_{ij}}+c_{ij},(\hat{\mathbf{u}}-\mathbf{u}_{gij})\cdot\mathbf{n}+\hat{c}\right], \tag{4.16}$$

with $\hat{\mathbf{u}}$ and $\hat{c}$ being the values computed by Roe-averaging for the velocity and the sound speed. $\mathbf{U}_i$ and $\mathbf{U}_{ij}$ represent the approximate values at Gaussian quadrature points on edge Γ_{ij} which are computed from the reconstruction functions over Ω_i and Ω_{ij} respectively, and $\mathbf{u}_{gij}$ represents the approximate grid velocity at Gaussian points. It is noted that the grid velocity varies linearly along edge Γ_{ij}.

4.4 Solutions of Euler equation for PV moment in two dimension

We solve (2.6) point-wisely by Roe's Riemann solver as

$$\begin{aligned}\frac{\partial\mathbf{W}_k}{\partial t}=&-\frac{1}{2}\left(\tilde{A}\left(\frac{\partial\mathbf{W}_k^-}{\partial x}+\frac{\partial\mathbf{W}_k^+}{\partial x}\right)+\tilde{R}_A|\tilde{\Lambda}_A|\tilde{L}_A\left(\frac{\partial\mathbf{W}_k^-}{\partial x}-\frac{\partial\mathbf{W}_k^+}{\partial x}\right)\right)\\&-\frac{1}{2}\left(\tilde{B}\left(\frac{\partial\mathbf{W}_k^-}{\partial y}+\frac{\partial\mathbf{W}_k^+}{\partial y}\right)+\tilde{R}_B|\tilde{\Lambda}_B|\tilde{L}_B\left(\frac{\partial\mathbf{W}_k^-}{\partial y}-\frac{\partial\mathbf{W}_k^+}{\partial y}\right)\right),\end{aligned} \tag{4.17}$$

where the matrices are approximated from the Roe-averaging values of the surrounding VIAs by

$$\tilde{\phi}_k=\frac{\sum_{m=0}^{4}\sqrt{\overline{\rho}_{km}}\cdot\overline{\phi}_{km}}{\sum_{m=0}^{4}\sqrt{\overline{\rho}_{km}}}, \tag{4.18}$$

and the sound speed is calculated as

$$\tilde{a}_k=\left((\gamma-1)(\tilde{H}_k-\frac{1}{2}(\tilde{u}_k^2+\tilde{v}_k^2))\right)^{\frac{1}{2}}. \tag{4.19}$$

The spatial derivatives of primitive variables $(\frac{\partial\mathbf{W}_k^{\pm}}{\partial x},\frac{\partial\mathbf{W}_k^{\pm}}{\partial y})$ can be obtained from the spatial derivatives of the conservative variables $(\frac{\partial\mathbf{U}_k^{\pm}}{\partial x},\frac{\partial\mathbf{U}_k^{\pm}}{\partial y})$ in analogy to (3.16) and (3.17).

The overshoots/undershoots of PV can be simply removed by

$$\phi_k = \min\left(\phi_k, \max(\overline{\phi}_{km})\right), \quad \phi_k = \max\left(\phi_k, \min(\overline{\phi}_{km})\right), \tag{4.20}$$

where $\max(\overline{\phi}_{km})$ and $\min(\overline{\phi}_{km})$ are the maximum and minimum VIAs on the surrounding cells sharing node k.

4.5 Volume discretization of the geometrical conservation laws

We only consider straight-line element in this research, and as introduced before, we assume that the grid velocity has a linear distribution over edges. Given the grid velocity at two ends of boundary segment Γ_{ij} denoted as $\mathbf{u}_{gij0}$ and $\mathbf{u}_{gij1}$, the face velocity is calculated as

$$\overline{\mathbf{u}}_{gij} = \frac{1}{2}(\mathbf{u}_{gij0} + \mathbf{u}_{gij1}). \tag{4.21}$$

Thus the semi-discretization of volume variation (2.5) can be written as

$$\frac{dV_i}{dt} = \sum_{m=1}^{4} \overline{\mathbf{u}}_{gij} \cdot \mathbf{n} |\Gamma_{im}|. \tag{4.22}$$

Maire [9] proved that this discretization satisfies the Geometrical conservation law. However, we should point out that the straight-line approximation to cell edges might not be adequate for high-order ALE formulation as will be observed later in the numerical tests. As shown in [13], curvilinear meshes are preferred for high-order accuracy.

4.6 Time step limitation

Time marching for all the semi-discrete time evolution equations (4.7), (4.17) and (4.22) in this section is implemented by third order Runge-Kutta scheme analogously to Subsection 3.4. Following [8, 40], a classical CFL stability condition and a geometrical time step limitation are both considered for a suitable time step.

At time t^n, the classical CFL time step is given as

$$\Delta t_E = C_E \min_{1 \leqslant i \leqslant I} \frac{\lambda_i}{\sqrt{u_i^2 + v_i^2} + a_i}, \tag{4.23}$$

where λ_i is the in-circle diameter of cell i and a_i is the sound speed in the cell. We set the CFL number $C_E = 0.3$ for later simulation.

The second criterion limits the variation rate of the element volume in one time step, which is given as

$$\Delta t_V = C_V \min_{1 \leqslant i \leqslant I} \frac{|S_i^n|}{|S_i'|}, \tag{4.24}$$

where

$$S_i^{'}=\sum_{j=1}^{4}\overline{\mathbf{u}}_{gij}\cdot\mathbf{n}|\Gamma_{ij}|, \tag{4.25}$$

and C_V is set as 0.1.

Finally, the interval for time stepping is

$$\Delta t=\min(\Delta t_E,\Delta t_V,C_M\Delta t^n), \tag{4.26}$$

where C_M is set as 1.1 in this paper.

4.7 Numerical results in two dimension

In this section, the multi-moment ALE method is verified with some 2D benchmark tests. For the Lagrangian framework, the grid velocity is simply calculated from the Roe average of the surrounding VIAs as shown in (4.18) with ϕ representing the fluid velocity $\mathbf{u}$. As shown in the 1D cases, the Lagrangian and Eulerian frameworks have been evaluated as two special cases of the general ALE formulations. We also assessed the performance of the ALE scheme on moving meshes with specified nodal velocities.

4.7.1 2D isentropic vortex

Advection of an isentropic vortex [41] is computed to examine the convergence rate of the present ALE scheme. we computed this problem on a computational domain of $[0,10]\times[0,10]$ with the initial conditions specified as

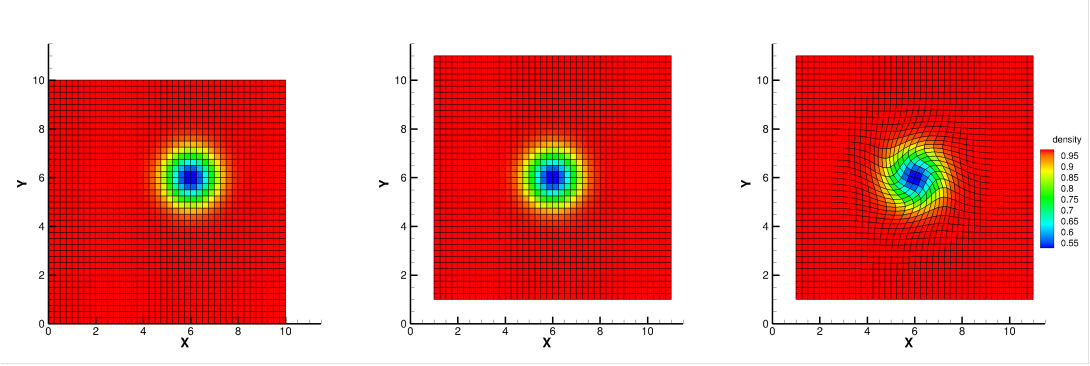
$$(\rho_0,u_0,v_0,p_0)=(1+\delta\rho,1+\delta u,1+\delta v,1+\delta p), \tag{4.27}$$

where $\delta\rho,\delta u,\delta v,\delta p$ are the initial perturbations to the primitive variables, which are centered in the computational domain and given by

$$\begin{aligned}
\delta\rho&=(1+\delta T)^{\frac{1}{\gamma-1}}-1,\\
\delta u&=-\frac{\epsilon}{2\pi}e^{\frac{1-r^2}{2}}(y-5),\\
\delta v&=\frac{\epsilon}{2\pi}e^{\frac{1-r^2}{2}}(x-5),\\
\delta p&=(1+\delta T)^{\frac{\gamma}{\gamma-1}}-1,
\end{aligned} \tag{4.28}$$

where the vortex strength $\epsilon=5$, the temperature perturbation $\delta T=-\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2}e^{1-r^2}$ and $r^2=(x-5)^2+(y-5)^2$. The periodic boundary conditions were applied to the boundaries of the computational domain for Eulerian and Lagrangian framework and zero gradient boundary condition is set for the following specific ALE framework.

We computed this problem to time $t=1$ using the Eulerian ($u_g=0,v_g=0$), Lagrangian ($u_g=u,v_g=v$) and ALE ($u_g=1-\pi\tilde{y}/2,v_g=1+\pi\tilde{x}/2$) frameworks respectively. In the



(a) (b) (c)

Figure 5: The density results for 2D isentropic vortex solved by the present ALE scheme. (a) Eulerian ($u_g=0, v_g=0$); (b) ALE ($u_g=1-\pi\tilde{y}/2, v_g=1+\pi\tilde{x}/2$); (c) Lagrangian ($u_g=u, v_g=v$). It is noted that the coordinate axes are fixed, while the meshes in (b) and (c) move.

ALE framework, where $\tilde{x}=(x-(5+t))$ and $\tilde{y}=(y-(5+t))$, the mesh moves at a velocity field that combines a translational motion $(1,1)$ and a rigid-rotational motion around the vortex center with an angular speed of $\pi/2$. Up to $t=1$, the mesh is moved over a distance of 1 in both x and y directions, and rotated by $90^o(\pi/2)$ as shown in Fig. 5(b).

The density fields of numerical solutions using different mesh-moving strategies are shown in Fig. 5. The exact solution of the Euler equations from the particular initial conditions is the pure translation of the vortex in the diagonal direction of the computational domain. Our numerical experiments started with Cartesian grids of gradually refined resolutions. The Eulerian framework remains the initial Cartesian grids fixed, while the grids nodes were moved at specified velocities in ALE and Lagrangian cases. We measured the L_1 and L_∞ errors of the numerical experiments, which are shown with the convergence rates in Table 4.

Table 4: Errors and convergence rates of density for 2D isentropic vortex.

cells	Eulerian				ALE				Lagrangian			
	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate	L_1	rate	L_∞	rate
20×20	1.15e-3	-	1.94e-2	-	1.02e-3	-	9.38e-3	-	1.25e-3	-	1.14e-2	-
40×40	2.00e-4	2.52	3.60e-3	2.43	1.54e-4	2.69	2.09e-3	2.17	2.17e-4	2.53	2.87e-3	1.98
80×80	2.66e-5	2.91	5.12e-4	2.81	2.06e-5	2.90	3.24e-4	2.69	3.84e-5	2.50	6.77e-4	2.08
160×160	3.36e-6	2.96	6.69e-5	2.94	2.73e-6	2.92	4.39e-5	2.88	7.72e-6	2.31	1.49e-4	2.18

It is observed that both the Eulerian and ALE frameworks can achieve third order accuracy, while the Lagrangian framework gets second order accuracy. This difference is due to the fact that in Eulerian and ALE frameworks the edges of mesh elements always remain to be straight lines in this test, so the formulation for moving mesh in the present method does not loss numerical accuracy. In the Lagrangian case, however, the flow velocity may deform the cell edges into curved lines, and the straight-line formulation is

not adequate to maintain the high-order accuracy. Higher-order geometrical approximations are needed when the specified mesh velocity might bend cell edges as addressed in [13].

4.7.2 2D Sod problem

We solved the well-known Sod shock tube problem on 2D meshes within the computational domain $[0,1]\times[0,0.1]$. The initial conditions are

$$(\rho_0,u_0,v_0,p_0)=\begin{cases}(1,0,0,1), & \text{for } 0\leqslant x\leqslant 0.5,\\ (0.125,0,0,0.1), & \text{for } 0.5<x\leqslant 1,\end{cases} \tag{4.29}$$

which will generate rarefaction wave, contact discontinuity and shock wave.

The computational domain is partitioned into 100×10 initially uniform quadrilateral elements. We computed the problem to time $t=0.25$ for the Eulerian, Lagrangian and ALE framework. The grid for Lagrangian frame at time $t=0.25$ is shown in Fig. 6. Fig. 7 shows the density results computed by the present scheme on different frameworks. We can find that the Lagrangian framework gets more accurate results around contact discontinuity than the Eulerian framework.

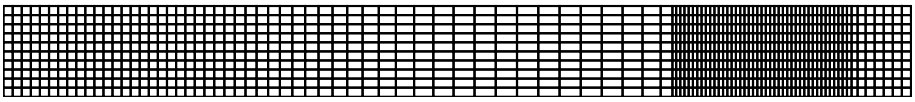
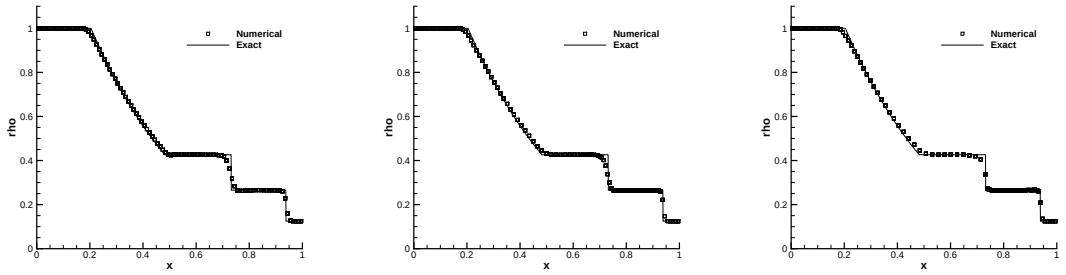


Figure 6: 2D Sod problem grid at time $t=0.25$ for Lagrangian framework.



(a) (b) (c)

Figure 7: The density profiles of the results for Sod problem with different mesh configurations: (a) Eulerian ($u_g=0$); (b) ALE ($u_g=0.5u$); (c) Lagrangian ($u_g=u$).

4.7.3 The Sedov problem

We tested the presented scheme with the Sedov problem, which is a circular blast wave generated from a point source at the origin of the Cartesian grid. The exact solution of this problem is available based on the self-similarity arguments [42]. We computed this problem on a quarter of the physical domain, $[0,1.2]\times[0,1.2]$, with 30×30 initially uniform quadrilateral elements. The initial conditions are given by $(\rho_0,u_0,v_0,p_0)=(1,0,0,10^{-6})$ except in the first zone, where an initial delta-function energy source was set in terms of pressure by

$$p_{\text{or}}=(\gamma-1)\rho_0\frac{\mathcal{E}_0}{V_{\text{or}}}, \tag{4.30}$$

where V_{or} denotes the volume of the cell containing the origin and $\mathcal{E}_0$ is the total amount of released energy which was set as $\mathcal{E}_0=0.244816$ in our numerical tests. Reflective boundary conditions were imposed on the left and lower boundaries of the computational domain. Numerical tests were carried out up to time $t=1$. As discussed in [42], the solution consists of an axis-symmetrically diverging shock whose front is located at radius $R=1$ at time $t=1$ with the peak density reaching the value of 6.

We computed this test case in Lagrangian, Eulerian and ALE frameworks respectively, with the grid velocity being 0.7 of the fluid velocity for ALE framework. Fig. 8 presents the density results for different frameworks, where panels (a), (c) and (e) show the density distribution over the computational domain as well as the computational meshes, and panels (b), (d) and (f) compare the profiles of density along the radial directions with the exact solution. The Lagrangian solution is the most accurate among the three frameworks, which well captures the discontinuity and the peak value.

5 Conclusion remarks

In this paper, we present a direct ALE multi-moment finite volume scheme for compressible Euler equations in one and two dimensions. In the multi-moment finite volume method, both the volume integrated averages (VIA) and the point values (PV) at the vertices of mesh cells, which are treated as prognostic variables and updated simultaneously, are used for high-order reconstructions. The governing equations are written with respect to the conservative variables in the integral form to update the VIA, while non-conservative equations of the primitive variables in the differential form are used to update the PV moment. The PVs are computed point-wisely with Roe Riemann solver using the spatial derivatives of primitive variables, while the VIAs are updated by the finite volume formulation of flux form where the numerical fluxes are calculated directly from the cell boundary PVs in 1D case and from the high-order multi-moment reconstructions in 2D case. Third order Runge-Kutta scheme is implemented for time marching. The mesh is moved by giving specified velocities to the mesh vertices, which results in a simple and efficient straight-line formulation for ALE computations.

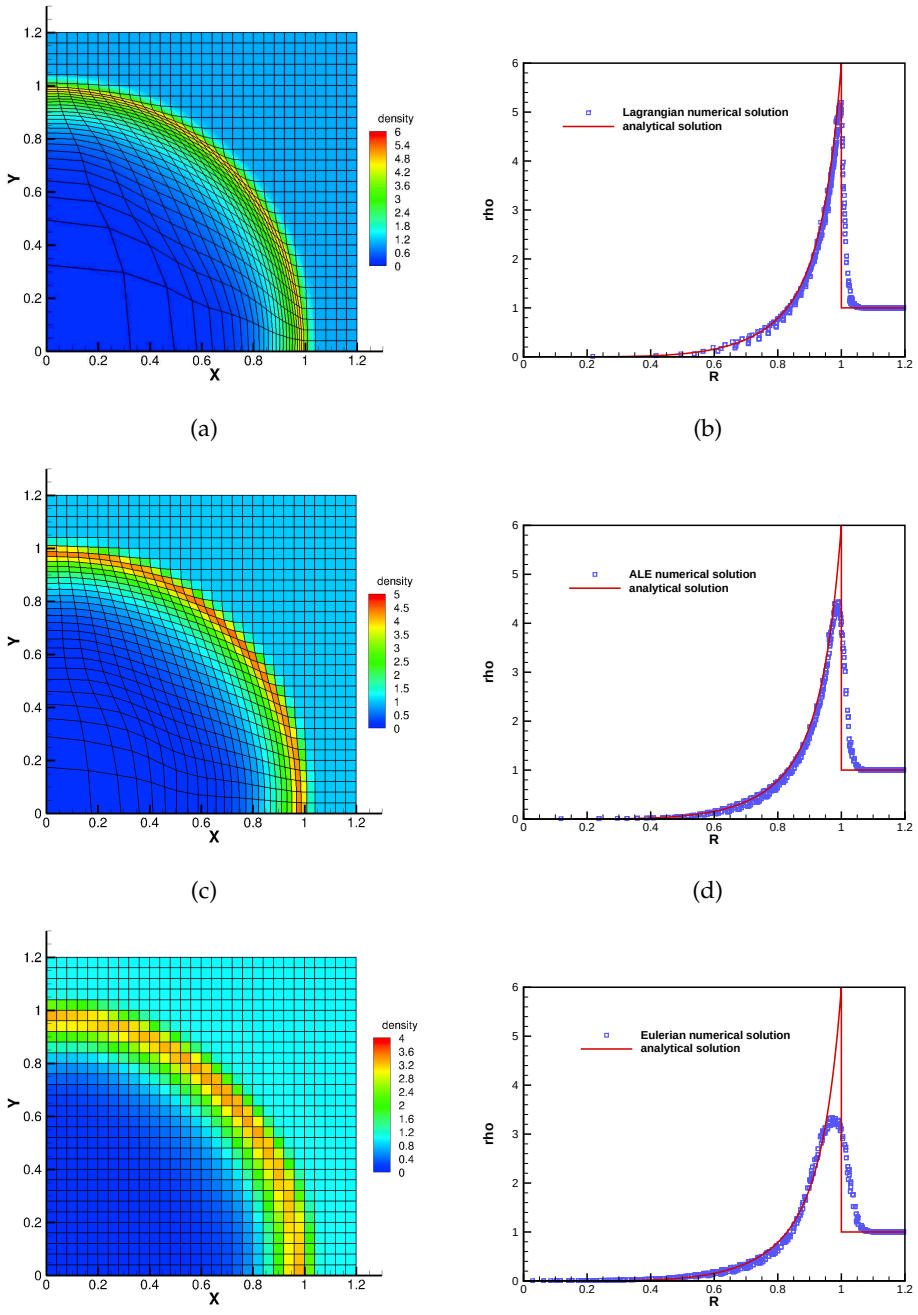


Figure 8: Mesh and density results of Sedove problem with present scheme: (a)(b) Lagrangian ($\mathbf{u}_g=\mathbf{u}$); (c)(d) ALE ($\mathbf{u}_g=0.7\mathbf{u}$); (e)(f) Eulerian ($\mathbf{u}_g=0$).

The proposed direct ALE multi-moment finite volume formulation provides a high-order and efficient ALE computational model for compressible flows. The numerical verifications show that the present model can achieve third-order convergence rate in 1D and 2D cases where the cell edges of moving mesh remain as straight lines. It can produce more accurate results than conventional finite volume ALE scheme with some increase in memory storage. The Lagrangian framework, as a particular case of the ALE model, has superior capability to capture moving discontinuities. It also reveals that the curvilinear moving mesh should be further explored when the cell nodal velocity might lead to bent cell edges. It is noted that though not included in the present paper the more profound impact of the present work lies in the fluid-structure interface applications. As aforementioned, given the nodal values on the structure solved as computational variables at every time step, boundary conditions on fluid-structure interface can be formulated in a more accurate and straightforward way in comparison to the conventional finite volume method where only the cell average values are defined and updated.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Numbers 15H03916 and 17K18838.

References

[1] C. Hirt, A. A. Amsden and J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys., 14.3 (1974), 227-253.
[2] J. Donea, A. Huerta, J.-P. Ponthot and A. Rodriguez-Ferran, Encyclopedia of Computational Mechanics Vol. 1, Fundamentals. Chapter 14: Arbitrary Lagrangian-Eulerian Methods, (2004).
[3] P.-O. Persson, J. Bonet and J. Peraire, Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains, Comput. Methods. Appl. Mech. Eng., 198.17 (2009), 1585-1595.
[4] W. Boscheri, R. Loube're and M. Dumbser, Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimentional hyperbolic conservation laws, J. Comput. Phys., 292 (2005), 56-87.
[5] C. Munz, On Godunov-type schemes for Lagrangian gas dynamics, SIAM J. Numer. Anal., 31.1 (1994), 17-42.
[6] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys., 43.2 (1981), 357-372.
[7] A. Harten, P.D. Lax and B.V. Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev., 25.1 (1983), 35-61.
[8] P.-H. Maire, R. Abgrall, J. Breil and J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM J. Sci. Comput., 29.4 (2007), 1781-1824.
[9] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys., 228.7 (2009), 2391-2425.

[10] P.-H. Maire and B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, J. Comput. Phys., 228.3 (2009), 799-821.
[11] J. Cheng and C.-W. Shu, A high order ENO conservative Lagrangian type scheme for the compressible Euler equations, J. Comput. Phys., 227.2 (2007), 1567-1596.
[12] A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes III, J. Comput. Phys., 71.2 (1987), 231-303.
[13] 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.
[14] M. Dumbser, A. Uuriintsetseg and O. Zanotti, On arbitrary-Lagrangian-Eulerian one-step WENO schemes for stiff hyperbolic balance laws, Commun. Comput. Phys., 14.2 (2013), 301-327.
[15] F. Vilar, P.-H. Maire and R. Abgrall, A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids, J. Comput. Phys., 276 (2014), 188-234.
[16] D. J. Mavriplis and C. R. Nastase, On the geometric conservation law for high-order discontinuous Galerkin discretizations on dynamically deforming meshes, J. Comput. Phys., 230.11 (2011), 4285-4300.
[17] X. Ren, K. Xu and W. Shyy, A multi-dimensional high-order DG-ALE method based on gas-kinetic theory with application to oscillating bodies, J. Comput. Phys., 316 (2016), 700-720.
[18] G. Clair, J.-M. Ghidaglia and J.-P. Perlat, A multi-dimensional finite volume cell-centered direct ALE solver for hydrodynamics, J. Comput. Phys., 326 (2016), 312-333.
[19] W. Boscheri and M. Dumbser, A direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3D, J. Comput. Phys., 275 (2014), 484-523.
[20] S. Ii and F. Xiao, High order multi-moment constrained finite volume method, Part I, Basic formulation, J. Comput. Phys., 228.10 (2009), 3669-3707.
[21] B. Xie, S. Ii, A. Ikebata and F. Xiao, A multi-moment finite volume method for incompressible Navier-Stokes equations on unstructured grids: volume-average/point-value formulation, J. Comput. Phys., 277 (2014), 138-162.
[22] B. Xie and F. Xiao, Two and three dimensional multi-moment finite volume solver for incompressible Navier-Stokes equations on unstructured grids with arbitrary quadrilateral and hexahedral elements, Comput. Fluids, 104 (2014), 40-54.
[23] B. Xie, X. Deng, Z. Sun and F. Xiao, A hybrid pressure-density-based Mach uniform algorithm for 2D Euler equations on unstructured grids by using multi-moment finite volume method, J. Comput. Phys., 335 (2017), 637-663.
[24] X. Deng, B. Xie and F. Xiao, A finite volume multi-moment method with boundary variation diminishing principle for Euler equation on three-dimensional hybrid unstructured grids, Comput. Fluids, 153 (2017), 85-101.
[25] X. Deng, B. Xie and F. Xiao, Multimoment Finite Volume Solver for Euler Equations on Unstructured Grids, AIAA J., (2017).
[26] Y.Bazilevs, K.Takizawa and T.Tezduyar, Computational fluid-structure interaction-methods and applications, 2013, A John Wiley & Sons. Ltd. Publication.
[27] J. S. Park, S.-H. Yoon and C. Kim, Multi-dimensional limiting process for hyperbolic conservation laws on unstructured grids, J. Comput. Phys., 229.3 (2010), 788-812.
[28] K. Liu, Y. Lu and C. You, High-order ALE method for the Navier-Stokes equations on a moving hybrid unstructured mesh using flux reconstraction method, Int. J. Comput. Fluid Dyn., 27.6-7 (2013), 251-267.

[29] E.F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.
[30] P. Thomas and C. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J., 17.10 (1979), 1030-1037.
[31] F. Xiao and T. Yabe, Completely conservative and oscillationless semi-Lagrangian schemes for advection transportation, J. Comput. Phys., 170.2 (2001), 498-522.
[32] S. Ii and F. Xiao, CIP/multi-moment finite volume method for Euler equations: a semi-Lagrangian characteristic formulation, J. Comput. Phys., 222.2 (2007), 849-871.
[33] T. Yabe, F. Xiao and T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys., 169.2 (2001), 556-593.
[34] R.J. LeVeque, Finite volume methods for hyperbolic problems, Vol. 31. Cambridge university press, 2002.
[35] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys., 77.2 (1988), 439-471.
[36] P. Woodward and P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys., 54.1 (1984), 115-173.
[37] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys., 83.1 (1989), 32-78.
[38] B. Xie and F. Xiao, A multi-moment constrained finite volume method on arbitrary unstructured grids for incompressible flows, J. Comput. Phys., 327 (2016), 747-778.
[39] H. Luo, J.D. Baum and R.Löhner, On the computation of multi-material flows using ALE formulation, J. Comput. Phys., 194.1 (2004), 304-328.
[40] Z. Shen, W. Yan and G. Yuan, A robust and contact resolving Riemann solver on unstructured mesh, Part II, ALE method, J. Comput. Phys., 268 (2014), 456-484.
[41] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, Advanced numerical approximation of nonlinear hyperbolic equations, Springer Berlin Heidelberg, (1998), 325-432.
[42] J.R. Kamm and F. Timmes, On efficient generation of numerically robust Sedov solutions, Technical Report LA-UR-07-2849, Los Alamos National Laboratory, (2007).
[43] B. Cockburn and C.W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput., 52 (1989), 411-435.
[44] 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 system, J. Comput. Phys., 84 (1989), 90-113.
[45] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys., 178 (2002), 210–251.
[46] Z.J. Wang and Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys., 179 (2002), 665-697.