

High-order compact gas-kinetic scheme in arbitrary Lagrangian-Eulerian formulation

Yue Zhang^a, Kun Xu^{a,b,c,*}

^a*Department of Mathematics, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

^b*Department of Mechanical and Aerospace Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong*

^c*Shenzhen Research Institute, Hong Kong University of Science and Technology, Shenzhen, China*

Abstract

This study proposes an extension of the high-order compact gas-kinetic scheme (CGKS) to compressible flow simulation in an arbitrary Lagrangian-Eulerian (ALE) formulation in unstructured mesh. The ALE method is achieved by subdividing arbitrary mesh into tetrahedrons and integrating flux function in a local coordinate system at the cell interface to ensure geometric conservation law. The scheme incorporates a compact reconstruction with third-order accuracy for updating both cell-averaged conservative flow variables and their gradients. HWENO-type nonlinear reconstruction and gradient compression factors are adopted to improve the accuracy and robustness of the scheme. A multi-stage multi-derivative (MSMD) time-stepping method is also implemented to achieve high-order time accuracy with fewer middle stages. The scheme is used to study problems involving moving boundaries. The numerical experiments demonstrate the effectiveness of the scheme in capturing the accurate solutions of both low-speed smooth flow and highly compressible ones with strong shock waves.

Keywords:

arbitrary Lagrangian-Eulerian (ALE), geometric conservation law (GCL), compact gas-kinetic scheme, multi-stage multi-derivative (MSMD) time discretization

1. Introduction

There are two common frameworks for describing fluid motion in computational fluid dynamics: the Eulerian and the Lagrangian systems. The Eulerian system describes flow on a fixed grid, while in the Lagrangian system, the grid moves with the fluid velocity. While the Eulerian system is more straightforward to implement, it has difficulties in handling moving boundaries and introducing dissipation when dealing with contact discontinuities. The Lagrangian system can resolve contact discontinuities well, such as material interfaces or free

*Corresponding author

Email addresses: yzhangn1@connect.ust.hk (Yue Zhang), makxu@ust.hk (Kun Xu)

surfaces, but may lead to computational breakdown due to grid distortion. Therefore, the development of arbitrary Lagrangian-Eulerian (ALE) methods [5] becomes attractive. There are two types of ALE methods: direct and indirect. Direct ALE methods involve moving both the object and the computational grid simultaneously. Indirect ALE methods separate the motion of the object and the computational grid into three stages: the Lagrangian stage, the rezoning stage [2], and the remapping stage [11]. During the Lagrangian stage, the flow field and grid are updated simultaneously. During the rezoning stage, a new computational grid is constructed to match the object's new position and shape, often involving techniques such as grid generation and deformation to ensure grid quality and accuracy. In the remapping stage, the solution is transferred into the rezoned mesh. The current research focuses on the direct ALE method with a fixed grid topology, which is designed to handle moving boundaries and shock waves. Therefore, the remapping step can be omitted.

The gas-kinetic scheme (GKS) is a kinetic theory-based numerical method to solve the Euler and Navier-Stokes equations [21]. Under the initial condition of a generalized Riemann problem, a time-evolving gas distribution function is constructed in GKS to calculate the numerical fluxes and evaluate the time-dependent flow variables at a cell interface. As a result, both cell-averaged flow variables and their gradients can be updated. Therefore, the HWENO-type method and the two-step multi-resolution WENO reconstruction can be developed in the scheme for the high-order spatial data reconstruction [27, 10]. At the same time, due to the time-accurate flux function, the multi-stage multi-derivative (MSMD) method can be used to update the solution with high-order temporary accuracy. Specifically, the two-stage fourth-order (S2O4) time-stepping method is used in the compact GKS (CGKS) [13, 12]. The CGKS has been constructed on both structured and unstructured meshes in 2D and 3D cases [6, 9, 23, 24, 26]. In order to further improve the robustness of the scheme in high-speed flow simulations, further improvements have been incorporated into the scheme. First, the evolution of possible discontinuous flow variables at different sides of a cell interface is constructed for updating reliable cell averaged gradient of flow variables [25]. Second, the nonlinear limiting process is implemented on the high-order time derivative of the flux function under the MSMD framework. Equipped with the above remedies, the CGKS on a 3D unstructured mesh is highly robust in hypersonic flow computation and a large time step, such as CFL number ≈ 0.8 , can be used in the fourth-order compact scheme. Alternatively, a gradient compression factor is designed to improve the robustness and efficiency of CGKS in case of low-quality mesh [7], and further improvements are studied in [22].

In recent years, high-order GKS has tended to incorporate ALE formulation. Ren [18] developed a one-stage DG-ALE gas-kinetic method for oscillating airfoil calculation. The variation of mesh velocity is considered in the time-dependent flux calculation along a cell interface, and the geometric conservation law is satisfied accurately. Pan developed a high-order ALE gas-kinetic method on structure mesh for two and three dimensions[17, 16], where a non-compact WENO scheme and the two-stage fourth-order discretization are used. A bilinear interpolation is used to parameterize both grid coordinates and grid moving velocity with the preservation of the geometric conservation law. However, this interpolation isn't a unified formulation for other types of mesh, such as triangular prism and pyramid. In

this paper, our target is to develop a compact gas-kinetic scheme on unstructured mesh in ALE formulation. Instead of bilinear interpolation, we use a subdivision of arbitrary meshes into tetrahedrons and a linear interpolation in triangular surfaces to preserve the geometry conservation law. A compact third-order WENO reconstruction with the requirement of Neumann neighboring cells only is adopted in spatial reconstruction. The radius basic function method [3] and Lagrangian nodal solver [14] are used to determine grid velocity.

The structure of the paper is the following. Section 2 introduces the finite volume method and GKS in ALE formulation. Section 3 discusses the grid velocity, the two-stage fourth-order time integrating method, and the geometric conservation law. Section 4 focuses on the compact reconstruction. In Section 5, several test cases are used to validate the proposed method. Finally, the conclusion is presented in the last section.

2. Gas-Kinetic Scheme in ALE Formulation

2.1. Governing equation on moving mesh

By using the finite volume method, the computational domain Ω is discretized into a series of non-overlap elements Ω_i ,

$$\Omega = \bigcup \Omega_i, \quad \Omega_i \cap \Omega_j = \emptyset (i \neq j).$$

The boundary of element Ω_i can be expressed as

$$\partial\Omega_i = \bigcup_{p=1}^{N_f} \Gamma_{ip},$$

where N_f is the number of element surface and Γ_{ip} is surface of the element. Reynolds transport theorem gives the conservation laws in an arbitrary Lagrangian-Eulerian (ALE) formulation on a moving cell

$$\frac{d}{dt} \int_{\Omega_i(t)} \mathbf{W} d\Omega + \int_{\partial\Omega_i(t)} (\mathbf{F}(\mathbf{W}) - \mathbf{W}\mathbf{U}) \cdot \mathbf{n} dS,$$

where \mathbf{U} is the mesh velocity, and $\mathbf{F}(\mathbf{W})$ is the flux pass though the element surfaces. For Navier-Stokes equation, $\mathbf{W} = [\rho, \rho V_1, \rho V_2, \rho V_3, \rho E]^T$ is the conservative values, and $\rho, \mathbf{V} = (V_1, V_2, V_3), E$ is the density, absolute velocity and energy of fluid.

- If $\mathbf{U} = \mathbf{V}$, the governing equation becomes Lagrangian form;
- If $\mathbf{U} = \mathbf{0}$, the governing equation becomes Eulerian form.

A unified partitioning approach is adopted to handle different types of grids and maintain consistency. Specifically, different types of grids (such as hexahedrons, prisms, and pyramids) are all partitioned into several tetrahedrons. The partitioning approach ensures computational consistency and facilitates the conversion between different types of grids.

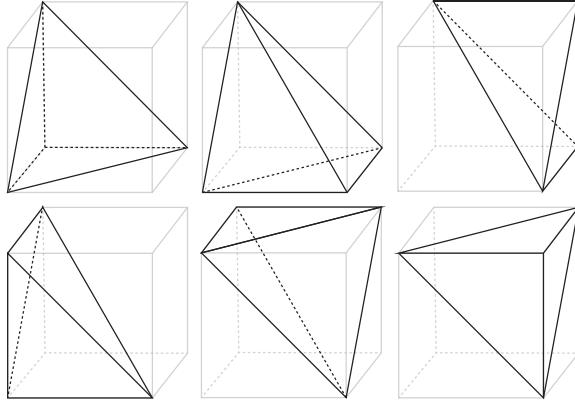


Figure 1: An example of mesh partition

An example of partitioning a hexahedron into six tetrahedrons is illustrated in Figure 1. To maintain consistency with the partitioning of elements, we divide the quadrilateral surface into two triangles. Under this condition, the conservation law in ALE formulation can be written as

$$\frac{d}{dt} \int_{\Omega(t)} \mathbf{W} d\Omega + \sum_{pt=1}^{N_t} \int_{\Sigma_{pt}} (\mathbf{F}(\mathbf{W}) - \mathbf{W}\mathbf{U}) \cdot \mathbf{n} dS = 0,$$

where N_t is the number of triangles after partitioning and Σ_{pt} is the individual triangle surfaces. The translation from Eulerian coordinates to local two-dimensional coordinates (ξ, η) as shown in Figure 2 is used to evaluate the surface integration. Thus, a linear

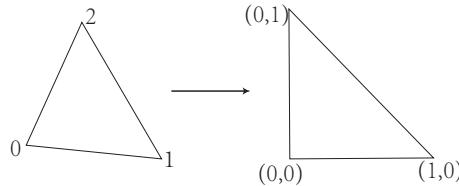


Figure 2: Translation a triangle surface to local coordinates

interpolation is introduced for each triangle

$$\mathbf{x}(\xi, \eta) = \mathbf{x}_0 + \xi(\mathbf{x}_1 - \mathbf{x}_0) + \eta(\mathbf{x}_2 - \mathbf{x}_0),$$

where $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2$ is the coordinates of the triangle vertices. With this transformation, the normal vector of triangle surfaces can be written as

$$\mathbf{n} = \mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta} = \frac{\partial \mathbf{x}}{\partial \xi} \times \frac{\partial \mathbf{x}}{\partial \eta}.$$

The surface integration can be written as

$$\int_{\Sigma_{pt}} (\mathbf{F}(\mathbf{W}) - \mathbf{WU}) \cdot \mathbf{n} \, dS = \int_0^1 \int_0^{1-\xi} (\mathbf{F}(\mathbf{W}) - \mathbf{WU}) \cdot (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) \, d\eta \, d\xi.$$

Assuming the grid velocity is time-independent, the coordinates of points on the surface can be written as $\mathbf{x} = \mathbf{x}^0 + \mathbf{U}t$, where \mathbf{x}^0 is the initial position of the points. Then, we have

$$\begin{aligned} \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{x}_{,\xi} \times \mathbf{x}_{,\eta}) \, d\eta \, d\xi &= \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) \, d\eta \, d\xi \\ &\quad + t \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{U}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) \, d\eta \, d\xi \\ &\quad + t \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{U}_{,\eta}) \, d\eta \, d\xi \\ &\quad + t^2 \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{U}_{,\xi} \times \mathbf{U}_{,\eta}) \, d\eta \, d\xi \end{aligned}.$$

The total flux can be approximated as

$$\begin{aligned} \int_{\Sigma_{pt}} (\mathbf{F}(\mathbf{W}) - \mathbf{WU}) \cdot \mathbf{n} \, dS &\approx \int_0^1 \int_0^{1-\xi} (\mathbf{F}(\mathbf{W}) - \mathbf{WU}) \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) \, d\eta \, d\xi \\ &\quad + t \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{U}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) \, d\eta \, d\xi \\ &\quad + t \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{U}_{,\eta}) \, d\eta \, d\xi \\ &\quad + t^2 \int_0^1 \int_0^{1-\xi} \mathbf{WU} \cdot (\mathbf{U}_{,\xi} \times \mathbf{U}_{,\eta}) \, d\eta \, d\xi \\ &= \mathcal{L}_{pt}(\mathbf{W}, \Omega) + \mathbb{V}_{pt}(t) \end{aligned},$$

where $\mathcal{L}_{pt}(\mathbf{W}, \Omega)$ is the surface flux, and $\mathbb{V}_{pt}(t)$ represents the flux correction due to the motion of grid, except the translation. Thus, the semi-discretized equation can be written as

$$\frac{d}{dt} \int_{\Omega_i(t)} \mathbf{W} \, d\Omega + \mathcal{L}(\mathbf{W}, \Omega) + \mathbb{V}(t) = 0. \quad (1)$$

The surface integration can be calculated by Gaussian integration. Three Gaussian points with local coordinates $(\frac{1}{6}, \frac{1}{6}), (\frac{2}{3}, \frac{1}{6}), (\frac{1}{6}, \frac{2}{3})$ and weights $\omega_1 = \omega_2 = \omega_3 = \frac{1}{3}$ are used in each

triangle surface. In this condition, the surface flux and correction can be written as

$$\begin{aligned}\mathcal{L}(\mathbf{W}, \Omega) &= \sum_{p_t=1}^{Nt} \mathcal{L}_{p_t}(\mathbf{W}, \Omega) = \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k [\mathbf{F}(\mathbf{W}_{p_t,k}) - \mathbf{W}_{p_t,k} \mathbf{U}_{p_t,k}] \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0)_{p_t}, \\ \mathbb{V}(t) &= \sum_{p_t=1}^{Nt} \mathbb{V}_{p_t}(t) = \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k \mathbf{W}_{p_t,k} \mathbf{U}_{p_t,k} \cdot [t(\mathbf{U}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) + t(\mathbf{x}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0) + t^2(\mathbf{U}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0)]_{p_t}.\end{aligned}$$

2.2. BGK equation in ALE Formulation

The gas-kinetic BGK equation can be written as

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_x f = \frac{g - f}{\tau},$$

where $f = f(\mathbf{x}, t, \mathbf{v}, \xi)$ is the gas distribution function, g is the corresponding equilibrium state, and τ is the collision time. $\mathbf{v} = (v_1, v_2, v_3)$ is the particle velocity. The collision term in the above equation describes the evolution process from a non-equilibrium state to an equilibrium one with the satisfaction of compatibility condition

$$\int \frac{g - f}{\tau} \Psi \, d\mathbf{w} \, d\Xi = 0,$$

where $\Psi = (1, v_1, v_2, v_3, \frac{1}{2}(v_1^2 + v_2^2 + v_3^2 + \xi^2))^T$ and $d\Xi = d\xi_1 \cdots d\xi_K$ (K is the number of internal degree of freedom, i.e. $K = (5 - 3\gamma)/(\gamma - 1)$ for three-dimensional flow and γ is the specific heat ratio). According to the Chapman-Enskog Expansion for BGK equation, the macroscopic governing equations can be derived[20]. In the continuum region, the BGK equation can be rearranged, and the gas distribution function can be expanded as

$$f = g - \tau \mathcal{D}_u g + \tau \mathcal{D}_u(\tau \mathcal{D}_u) g - \tau \mathcal{D}_u[\tau \mathcal{D}_u(\tau \mathcal{D}_u)] g + \cdots,$$

where $\mathcal{D}_u = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$. With the zeroth-order truncation $f = g$, the Euler equations can be obtained. For the first-order truncation $f = g - \tau \mathcal{D}_u g$, the Navier-Stokes equations with the dynamic viscosity coefficient $\mu = \tau p$ (p is pressure) and Prandtl number $\text{Pr} = 1$ can be obtained.

In this paper, a high-order gas-kinetic scheme will be constructed in the arbitrary Lagrangian-Eulerian (ALE) framework for three-dimensional flow. The particle speed related to grid $\mathbf{w} = (w_1, w_2, w_3)$ is

$$\mathbf{w} = \mathbf{v} - \mathbf{U},$$

where $\mathbf{U} = (U_1, U_2, U_3)$ is the grid velocity given arbitrary. The BGK equation on moving reference can be written as

$$\frac{\partial f}{\partial t} + \mathbf{w} \cdot \nabla_x f = \frac{g - f}{\tau}. \quad (2)$$

The flux in the term \mathcal{L} of ALE-formulation finite volume method (1) can be evaluated

by

$$\mathbb{F}(\mathbf{W}) = (\mathbf{F}(\mathbf{W}) - \mathbf{W}\mathbf{U}) \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) = \int f(\mathbf{x}, t, \mathbf{w}, \xi) \mathbf{w} \cdot \mathbf{n} \Psi \, d\mathbf{v} \, d\xi. \quad (3)$$

In the actual computation, the numerical flux can be obtained by taking moments of gas distribution function with related velocity

$$\mathbb{F}(\mathbf{W})' = \int f(\mathbf{x}, t, \mathbf{w}, \xi) \mathbf{w} \cdot \mathbf{n} \Psi' \, d\mathbf{v} \, d\xi, \quad (4)$$

where $\Psi = (1, w_1, w_2, w_3, \frac{1}{2}(w_1^2 + w_2^2 + w_3^2 + \xi^2))^T$. The component of \mathbb{F} can be given by the combination of \mathbb{F}' as follows

$$\begin{cases} \mathbb{F}_\rho = \mathbb{F}'_\rho \\ \mathbb{F}_{\rho V_1} = \mathbb{F}'_{\rho W_1} + U_1 \mathbb{F}'_\rho \\ \mathbb{F}_{\rho V_2} = \mathbb{F}'_{\rho W_2} + U_2 \mathbb{F}'_\rho \\ \mathbb{F}_{\rho V_3} = \mathbb{F}'_{\rho W_3} + U_3 \mathbb{F}'_\rho \\ \mathbb{F}_{\rho E} = \mathbb{F}'_{\rho E} + U_1 \mathbb{F}'_{\rho W_1} + U_2 \mathbb{F}'_{\rho W_2} + U_3 \mathbb{F}'_{\rho W_3} + \frac{1}{2}(U_1^2 + U_2^2 + U_3^2) \mathbb{F}'_\rho \end{cases}.$$

By defining the rotating matrix $\mathbf{T} = \text{diag}(1, \mathbf{T}', 1)$ and

$$\mathbf{T}' = \begin{pmatrix} n_1 & n_2 & n_3 \\ -n_2 & n_1 + \frac{n_3^2}{1+n_1} & -\frac{n_2 n_3}{1+n_1} \\ -n_3 & -\frac{n_2 n_3}{1+n_1} & 1 - \frac{n_3^2}{1+n_1} \end{pmatrix}, n_1 \neq -1,$$

and when $n_1 = -1$, \mathbf{T}' becomes $\text{diag}(-1, -1, 1)$, the flux (Eq. 5) can be evaluated by

$$\mathbb{F}(\mathbf{W})' = \mathbf{T}^{-1} \int f(\tilde{\mathbf{x}}_{p_t, k}, t, \tilde{\mathbf{w}}, \xi) \tilde{w}_1 \tilde{\Psi}' \, d\tilde{\mathbf{w}} \, d\xi, \quad (5)$$

where the origin point of local coordinate is $\tilde{\mathbf{x}} = (0, 0, 0)$ with x -direction in \mathbf{n} and $\tilde{\Psi}' = (1, \tilde{w}_1, \tilde{w}_2, \tilde{w}_3, \frac{1}{2}(\tilde{w}_1^2 + \tilde{w}_2^2 + \tilde{w}_3^2 + \xi^2))^T$. The microscopic velocities in local coordinate are given by $\tilde{\mathbf{w}} = \mathbf{T}' \mathbf{w}$ and $\tilde{w}_1 = n_1 w_1 + n_2 w_2 + n_3 w_3$.

2.3. Gas evolution model

In order to construct the numerical fluxes at $\mathbf{x} = (0, 0, 0)^T$, the integral solution of the BGK equation Eq.(2) is

$$f(\mathbf{x}, t, \mathbf{w}, \xi) = \frac{1}{\tau} \int_0^t g(\mathbf{x}', t' \mathbf{w}, \xi) e^{-(t-t')/\tau} \, dt' + e^{t/\tau} f_0(\mathbf{x}_0, \mathbf{w}), \quad (6)$$

with the trajectory

$$\mathbf{x} = \mathbf{x}' + \mathbf{w}(t - t').$$

In Eq.(6), f_0 is the initial gas distribution function, and g is the corresponding equilibrium state. \mathbf{x}_0 are the initial position by tracing back particles \mathbf{x} at time t back to $t = 0$.

Before constructing the initial distribution function f_0 and equilibrium state g , we first denote

$$\mathbf{a} = (a_1, a_2, a_3) = \nabla_x g / g, A = g_t / g.$$

In the following derivation, quadratic terms of time will be ignored directly. With the consideration of possible discontinuity at an interface, the initial distribution is constructed as

$$f_0(\mathbf{x}_0, \mathbf{w}_0) = f_0^l(\mathbf{x}_0, \mathbf{w})(1 - \mathbb{H}(x_1)) + f_0^r(\mathbf{x}_0, \mathbf{w})\mathbb{H}(x_1), \quad (7)$$

where \mathbb{H} is the Heaviside function. f_0^l and f_0^r are the initial gas distribution functions on the left and right sides of the interface, which are determined by corresponding initial macroscopic variables and their spatial derivatives. With the second-order accuracy, $f_0^k(\mathbf{x}, \mathbf{w})$ is constructed by Taylor expansion around (\mathbf{x}, \mathbf{w})

$$f_0^k(\mathbf{x}_0, \mathbf{w}) = f_G^k(\mathbf{x}, \mathbf{w}) - \mathbf{w}t \cdot \nabla_x f_G^k \quad (8)$$

for $k = l, r$. Based on the Chapman-Enskog expansion, f_G^k is given by

$$f_G^k = g^k[1 - \tau(A^k + \mathbf{a}^k \cdot \mathbf{w})], \quad (9)$$

where g^k is the equilibrium distribution function defined by the macroscopic variables \mathbf{W}^k at both sides of a cell interface, \mathbf{a}^k are defined by the spatial derivatives of g^k

$$\begin{aligned} a_i^k &= (\frac{\partial g^k}{\partial \rho^k} \frac{\partial \rho^k}{\partial x_i} + \frac{\partial g^k}{\partial W_1^k} \frac{\partial W_1^k}{\partial x_i} + \frac{\partial g^k}{\partial W_2^k} \frac{\partial W_2^k}{\partial x_i} + \frac{\partial g^k}{\partial W_3^k} \frac{\partial W_3^k}{\partial x_i} + \frac{\partial g^k}{\partial \lambda^k} \frac{\partial \lambda^k}{\partial x_i}) / g^k \\ &= a_{i1}^k + a_{i2}^k w_1 + a_{i3}^k w_2 + a_{i4}^k w_3 + a_{i5}^k \frac{1}{2}(w_1^2 + w_2^2 + w_3^2 \xi^2), \end{aligned}$$

and

$$A^k = A_1^k + A_2^k w_1 + A_3^k w_2 + A_4^k w_3 + A_5^k \frac{1}{2}(w_1^2 + w_2^2 + w_3^2 \xi^2)$$

are determined by compatibility condition

$$\int (f_G^k - g^k) \Psi \, d\mathbf{w} \, d\Xi = 0. \quad (10)$$

Substituting Eq.(8) and (9) into (7), the initial gas distribution has following form

$$f_0 = \begin{cases} g^l [1 - (\mathbf{a}^l \cdot \mathbf{w}) t - \tau (A^l + \mathbf{a}^l \cdot \mathbf{w})], & x_1 < 0, \\ g^r [1 - (\mathbf{a}^r \cdot \mathbf{w}) t - \tau (A^r + \mathbf{a}^r \cdot \mathbf{w})], & x_1 \geq 0, \end{cases} \quad (11)$$

Then, the equilibrium distribution is defined by the Taylor expansion

$$\begin{aligned} g(\mathbf{x}', t', \mathbf{w}) &= \bar{g}(\mathbf{x}, 0, \mathbf{w}) + \nabla_x \bar{g} \cdot (\mathbf{x}' - \mathbf{x}) + \bar{g}_t t' \\ &= \bar{g}(\mathbf{x}, 0, \mathbf{w}) - \nabla_x \bar{g} \cdot \mathbf{w}(t - t') + \bar{g}_t t' \\ &= \bar{g}(\mathbf{x}, 0, \mathbf{w}) \{1 - \bar{\mathbf{a}} \cdot \mathbf{w}(t - t') + At'\}, \end{aligned} \quad (12)$$

where \bar{g} and $\bar{\mathbf{a}}$ are determined from the reconstruction of macroscopic flow variables presented in section 4.5, and A is obtained by compatibility condition (10).

In smooth flow region, the collision time for viscous flow is determined by $\tau = \mu/p$, where μ is the dynamic viscosity coefficient and p is the pressure at the cell interface, and for inviscid flow, the collision time should be set as zero. In order to properly capture the un-resolved shock structure, additional numerical dissipation is needed. The physical collision time τ in the exponential function part can be replaced by a numerical collision time τ_n . For inviscid flow, it is set as

$$\tau_n = C_1 \Delta t + C_2 \frac{|p_l - p_r|}{p_l + p_r} \Delta t,$$

and for viscous flow, it is

$$\tau_n = \tau + C_2 \frac{|p_l - p_r|}{p_l + p_r} \Delta t,$$

where p_l and p_r denote the pressure on the left and right sides of the cell interface. In this paper, we have $C_1 = 0.01$, $C_2 = 5.0$.

By substituting Eq. (11) and Eq. (12) into Eq. (6) with τ and τ_n , the second-order time-dependent gas distribution function at a cell interface becomes

$$\begin{aligned} f(\mathbf{x}, t, \mathbf{w}, \xi) &= (1 - e^{-t/\tau_n}) \bar{g} + e^{-t/\tau_n} [\mathbb{H}(w_1) g^l + (1 - \mathbb{H}(w_1)) g^r] + t \bar{A} \bar{g} \\ &\quad - \tau (1 - e^{-t/\tau_n}) \bar{g} (\bar{\mathbf{a}} \cdot \mathbf{w} + \bar{A}) \\ &\quad - \tau e^{-t/\tau_n} \mathbb{H}(w_1) g^l (\mathbf{a}^l \cdot \mathbf{w} + A^l) \\ &\quad - \tau e^{-t/\tau_n} (1 - \mathbb{H}(w_1)) g^r (\mathbf{a}^r \cdot \mathbf{w} + A^r) \\ &\quad + t e^{-t/\tau_n} [(\bar{\mathbf{a}} \cdot \mathbf{w}) \bar{g} - \mathbb{H}(w_1) (\mathbf{a}^l \cdot \mathbf{w}) g^l - (1 - \mathbb{H}(w_1)) (\mathbf{a}^r \cdot \mathbf{w}) g^r]. \end{aligned} \quad (13)$$

The fluxes in Eq.(5) can be obtained by taking the moments of the above distribution function. The calculation of moments can be found in [21].

In order to account for the Prandtl number not equal to 1, additional correction heat flux should be added in energy flux:

$$\left(\frac{1}{\text{Pr}} - 1 \right) q,$$

where $q = F_{v4} - W_1 F_{v1} - W_2 F_{v2} - W_3 F_{v3}$ can be obtained through the viscous flux $\mathbf{F}_{vis} =$

$[0, F_{v1}, F_{v2}, F_{v3}, F_{v4}]$. The viscous flux can be obtained by

$$\mathbf{F}_{vis} = -\tau \int w_1 \bar{g} [(\bar{\mathbf{a}} \cdot \mathbf{w}) + \bar{A}] \, d\mathbf{w} \, d\Xi.$$

2.4. Evolution of the cell-averaged spatial gradients

By taking moments of the above gas distribution function in Eq. (13), the time-accurate conservative flow variables in the moving frame at the Gaussian points can also be obtained

$$\mathbf{W}'_{pt,k}(t^{n+1}) = \mathbf{T}' \left(\int \tilde{\Psi} f(\tilde{\mathbf{x}}_{pt,k}, t^{n+1}, \tilde{\mathbf{w}}, \xi) \, d\mathbf{w} \, d\Xi \right), \quad (14)$$

by which the corresponding conservative flow variables in the absolute inertia frame $\mathbf{W}_{pt,k}(t^{n+1})$ can be obtained. According to the Divergence theorem, the cell averaged gradients over cell Ω_i at time t^{n+1} are

$$\overline{\nabla \mathbf{W}}_i^{n+1} = \frac{1}{|\Omega_i^{n+1}|} \sum_{pt=1}^{N_t} \int_{\Sigma_{pt}} \mathbf{W}^{n+1} \mathbf{n}_p dS, \quad (15)$$

where the surface integration can be calculated by Gaussian quadrature

$$\int_{\Sigma_{pt}} \mathbf{W}^{n+1} \mathbf{n}_{pt} dS \approx \sum_{k=1}^3 |S_{pt}| \omega_k \mathbf{W}_{pt,k}(t^{n+1}) \mathbf{n}_{pt}. \quad (16)$$

Besides evaluating the cell averaged gradients, the solution updates of the scheme are presented next.

3. Temporal discretization and Geometrical Conservation Law

3.1. Mesh Velocity

This article considers two types of mesh motion when the mesh velocity is not given directly. The first method is the radial basic function (RBF) method, which is used to address moving boundary problems. The second method is the Lagrangian velocity method, which is used to handle moving shock problems.

3.1.1. Radial Basic Function (RBF) Method

For moving boundary problems, we can obtain the boundary moving vector in the next step on boundary mesh, and the moving vectors of interior grids can be obtained by radial basic function interpolation [3]. We chose the radial basic function as

$$\psi(\xi) = (1 - \xi)^4(1 + 4\xi),$$

where $\xi = |\mathbf{x}|/R$ is the related distance, and R is the compactly supported radius. $|\mathbf{x}|$ is calculated by $\sqrt{x_1^2 + x_2^2 + x_3^2}$. Then, we can define an interpolation function

$$f(\mathbf{x}) = \sum_i \lambda_i \psi(|\mathbf{x} - \mathbf{x}_i|/R),$$

where the \mathbf{x}_i is a series of given points, and

$$f(\mathbf{x}_j) = \sum_i \lambda_i \psi(|\mathbf{x}_j - \mathbf{x}_i|/R) = g_j,$$

so we can solve λ_i in above equation. The whole process of RBF interpolation is shown as Algorithm 1.

Algorithm 1 RBF interpolation

Require: This algorithm requires a series of points \mathbf{x}_i and their corresponding data g_i , as well as the number of points N to be selected for interpolation, a compactly supported radius R , and a target interpolation point \mathbf{x}_0 .

Ensure: This algorithm ensures that an interpolated result g_0 is returned.

- 1: Select the closest N points to point \mathbf{x}_0 .
 - 2: Calculate an $N \times N$ matrix M_{ij} using the formula $M_{ij} = \psi(|\mathbf{x}_i - \mathbf{x}_j|/R)$.
 - 3: Solve the equation $M_{ij}\lambda_j = g_i$ by obtaining the values of λ_j .
 - 4: Initialize g_0 to 0.
 - 5: **for** i in range(N) **do**
 - 6: Compute $\xi_i = |\mathbf{x}_i - \mathbf{x}_0|/R$.
 - 7: **if** $\xi_i < 1.0$ **then**
 - 8: Add $\lambda_i \psi(\xi_i)$ to g_0 .
 - 9: **end if**
 - 10: **end for**
 - 11: **return** g_0
-

3.1.2. Lagrangian Velocity Method

The mesh velocity can be determined by the cell-centered Lagrangian nodal solver [14]. The basic idea is to solve two half one-dimensional Riemann problems at each cell interface by assuming that its velocity equals the point velocity. We denote $C(p)$ as the set of cells c that share the common vertex p and $F_p(c)$ as the set of faces of cell c that share the common vertex p . The grid speed \mathbf{U}_p can be given by

$$\mathbf{U}_p = \mathbb{M}_p^{-1} \sum_{c \in C(p)} \sum_{f \in F_p(c)} [S_f p_c \mathbf{N}_f^c + \mathbb{M}_{pcf} \mathbf{V}_c],$$

where p_c and \mathbf{V}_c are the pressure and velocity of cell c and S_f, \mathbf{N}_f is the surface area and normal direction. The matrix \mathbb{M}_{pcf} is calculated by

$$\mathbb{M}_{pcf} = S_f \rho_c a_c (\mathbf{N}_f^c \otimes \mathbf{N}_f^c),$$

where a_c and ρ_c are the sound speed and density of cell c and \otimes is the operator of tensor production, and matrix \mathbb{M}_p is the sum of \mathbb{M}_{pcf} for faces which is a symmetric positive definite matrix. In this case, the coordinate of points in the next time step can be calculated by

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{U}_p \Delta t.$$

With the Lagrangian velocity, the meshes become distorted, and computation may break down. Therefore, the meshes need to be smoothed by

$$\tilde{\mathbf{x}}_p = (1 - \omega) \mathbf{x}_p + \omega \frac{1}{N_p} \sum_{ip \in N(p)} \mathbf{x}_{ip},$$

where $N(p)$ is the set of vertexes jointing with vertex p , N_p is the size of $N(p)$, and ω is relaxation coefficient.

3.2. Solution Update and Geometrical Conservation Law

According to the semi-discretization equation Eq. (1), the two-stage fourth-order (S2O4) temporal discretization is adopted here for the solution updates [13],

$$\begin{aligned} (\Omega_i \mathbf{W}_i)^* &= (\Omega_i \mathbf{W}_i)^n + \frac{1}{2} \Delta t \mathcal{L}(\mathbf{W}_i^n, \Omega^n) + \frac{1}{8} \Delta t^2 \frac{\partial}{\partial t} \mathcal{L}(\mathbf{W}_i^n, \Omega^n) + \int_0^{1/2\Delta t} \mathbb{V}(t) dt, \\ (\Omega_i \mathbf{W}_i)^{(n+1)} &= (\Omega_i \mathbf{W}_i)^n + \Delta t \mathcal{L}(\mathbf{W}_i^n, \Omega^n) + \frac{1}{6} \Delta t^2 \left(\frac{\partial}{\partial t} \mathcal{L}(\mathbf{W}_i^n, \Omega^n) + 2 \frac{\partial}{\partial t} \mathcal{L}(\mathbf{W}_i^*, \Omega^*) \right) + \int_0^{\Delta t} \mathbb{V}(t) dt, \end{aligned}$$

which is a two-step time integration method. And constant grid velocities are considered during the time-integrating process. For simplicity, the conservative value at Gaussian points appears in correction term $\mathbb{V}(t)$ set as the moments of equilibrium state \bar{g} .

With a uniform flow field, the time evolution of the cell moving volume can be calculated

as

$$\begin{aligned}
(\Omega_i)^* &= (\Omega_i)^n + \frac{1}{2} \Delta t \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k \mathbf{U}_{p_t,k} \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0)_{p_t} \\
&\quad + \frac{1}{4} \Delta t^2 \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k \mathbf{U}_{p_t,k} \cdot \left[\frac{1}{2} (\mathbf{U}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) + \frac{1}{2} (\mathbf{x}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0) + \frac{1}{6} \Delta t (\mathbf{U}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0) \right]_{p_t}, \\
(\Omega_i)^{n+1} &= (\Omega_i)^n + \Delta t \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k \mathbf{U}_{p_t,k} \cdot (\mathbf{x}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0)_{p_t} \\
&\quad + \Delta t^2 \sum_{p_t=1}^{Nt} \sum_{k=1}^3 \omega_k \mathbf{U}_{p_t,k} \cdot \left[\frac{1}{2} (\mathbf{U}_{,\xi}^0 \times \mathbf{x}_{,\eta}^0) + \frac{1}{2} (\mathbf{x}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0) + \frac{1}{3} \Delta t (\mathbf{U}_{,\xi}^0 \times \mathbf{U}_{,\eta}^0) \right]_{p_t},
\end{aligned}$$

which automatically satisfies the Geometrical Conservation Law (GCL) when the volume of the element is taken as the sum of the tetrahedral volumes.

The gas distribution function at Gaussian points Eq. (13) can be written as a form

$$f = f + t f_t, \quad (17)$$

which denotes the second-order time accuracy of the gas distribution function. A two-step evolution of the gas distribution function can be constructed, which is updated by

$$\begin{aligned}
f^* &= f^n + \frac{1}{2} \Delta t f_t^n, \\
f^{n+1} &= f^n + \Delta t f_t^*.
\end{aligned}$$

Then, the time-dependent conservative values at each Gauss point can be obtained by Eq. (14). Then, by Eq. (15) and Eq. (16), the cell-averaged slopes can be updated.

3.3. The Algorithm of ALE Method

Algorithm 2 presents the complete computational algorithm, where the bold text indicates the special treatments compared to the Eulerian formulation. The mesh velocity is determined once per time step, but the geometry requires the update in every substep of the two-stage fourth-order temporal discretization. This necessitates the updating of the volume and reconstruction matrix of cells, as well as the normal vector, area, and Gaussian points coordinates of interfaces in every substep.

4. HWENO Reconstruction

The 3rd-order compact reconstruction [8] is adopted here with cell-averaged values and cell-averaged first-order spatial derivative. In order to capture shock, WENO weights [28] and gradient compression factor (CF) [7] are used. The reconstruction was further improved [22], where only one large stencil and one sub-stencil are involved in the new WENO procedure.

Algorithm 2 CGKS in ALE formulation

```

1: while the computation uncompleted do
2:   calculate the time step  $\Delta t$  according to CFL number
3:   determine mesh velocity
4:   for i=1,2(for S2O4) do
5:     define boundary condition for ghost cell
6:     reconstruct the cell distribution
7:     evolution for interfaces
8:     update the geometry
9:     update cell average conservative values and the first-order spatial derivatives of
       conservative values
10:    end for
11: end while

```

4.1. Third-order compact reconstruction for large stencil

Firstly, a linear reconstruction is presented. A quadratic polynomial p^2 with a third-order accuracy in space is constructed as follows

$$\begin{aligned}
p^2 = & a_0 + \frac{1}{h}[a_1(x - x_0) + a_2(y - y_0) + a_3(z - z_0)] \\
& + \frac{1}{2h^2}[a_4(x - x_0)^2 + a_5(y - y_0)^2 + a_6(z - z_0)^2] \\
& + \frac{1}{h^2}[a_7(x - x_0)(y - y_0) + a_8(y - y_0)(z - z_0) + a_9(x - x_0)(z - z_0)],
\end{aligned}$$

where $h = \frac{|\Omega_0|}{\max_j S_j}$ ($|\Omega_0|$ is the volume of cell and S_j is the area of cell's surface) is the cell size, and (x_0, y_0, z_0) is the coordinate of cell center.

The p^2 on Ω_0 is constructed on the compact stencil S including Ω_0 and its all von Neumann neighbors Ω_m ($m = 1, \dots, N_f$, where $N_f = 6$ for hexahedron cell or $N_f = 5$ for triangular prism). The cell averages \bar{Q} over Ω_0 and Ω_m and cell averages of space partial derivatives \bar{Q}_x, \bar{Q}_y and \bar{Q}_z over Ω_m are used to obtain p^2 .

The polynomial p^2 is required to exactly satisfy cell averages over both Ω_0 and Ω_m ($m = 1, \dots, N_f$)

$$\iiint_{\Omega_0} p^2 dV = \bar{Q}_0 |\Omega_0|, \iiint_{\Omega_m} p^2 dV = \bar{Q}_m |\Omega_m|,$$

with the following condition satisfied in a least-square sense

$$\begin{aligned}\iiint_{\Omega_m} \frac{\partial}{\partial x} p^2 dV &= (\bar{Q}_x)_m |\Omega_m| \\ \iiint_{\Omega_m} \frac{\partial}{\partial y} p^2 dV &= (\bar{Q}_y)_m |\Omega_m| \\ \iiint_{\Omega_m} \frac{\partial}{\partial z} p^2 dV &= (\bar{Q}_z)_m |\Omega_m|.\end{aligned}$$

The constrained least-square method is used to solve the above system.

4.2. Green-Gauss reconstruction for the sub stencil

The classical Green-Gauss reconstruction with only cell-averaged values is adopted to provide the linear polynomial p^1 for the sub-stencil.

$$p^1 = \bar{Q} + \frac{1}{|\Omega_0|} \mathbf{x} \cdot \sum_{m=1}^{N_f} \frac{\bar{Q}_m + \bar{Q}_0}{2} S_m \mathbf{n}_m,$$

where S_m is the area of the cell's surface and \mathbf{n}_m is the surface's normal vector.

4.3. Gradient compression Factor

The CF was first proposed in [7], and several improvements have been made in [22]. Denote $\alpha_i \in [0, 1]$ as gradient compression factor at targeted cell Ω_i

$$\alpha_i = \prod_{p_t=1}^{n_t} \prod_{k=0}^{M_p} \alpha_{p_t,k},$$

where $\alpha_{p_t,k}$ is the CF obtained at the k th Gaussian point of the interface p_t around cell Ω_i , which can be calculated by

$$\begin{aligned}\alpha_{p_t,k} &= \frac{1}{1 + A^2}, \\ A &= \frac{|p^l - p^r|}{p^l} + \frac{|p^l - p^r|}{p^r} + (\text{Ma}_n^l - \text{Ma}_n^r)^2 + (\text{Ma}_t^l - \text{Ma}_t^r)^2,\end{aligned}$$

where p is pressure, Ma_n and Ma_t are the Mach numbers defined by normal and tangential velocity, and superscript l, r denote the left and right values of the Gaussian points.

Then, the updated slope is modified by

$$\widetilde{\nabla \mathbf{W}}_i^{n+1} = \alpha_i \overline{\nabla \mathbf{W}}_i^{n+1},$$

and the Green-Gauss reconstruction is modified as

$$p^1 = \bar{Q} + \alpha \frac{1}{|\Omega_0|} \mathbf{x} \cdot \sum_{m=1}^{N_f} \frac{\bar{Q}_m + \bar{Q}_0}{2} S_m \mathbf{n}_m.$$

4.4. Non-linear WENO weights

In order to deal with discontinuity, the idea of multi-resolution WENO reconstruction is adopted [7, 28]. Here only two polynomials are chosen

$$P_2 = \frac{1}{\gamma_2} p^2 - \frac{\gamma_1}{\gamma_2} p^1, P_1 = p^1.$$

Here, we choose $\gamma_1 = \gamma_2 = 0.5$. So, the quadratic polynomial p^2 can be written as

$$p^2 = \gamma_1 P_1 + \gamma_2 P_2. \quad (18)$$

Then, we can define the smoothness indicators

$$\beta_j = \sum_{|\alpha|=1}^{r_j} \Omega^{\frac{2}{3}|\alpha|-1} \iiint_{\Omega} (D^\alpha p^j(\mathbf{x}))^2 dV,$$

where α is a multi-index and D is the derivative operator, and $r_1 = 1$ and $r_2 = 2$. Special care is given for β_1 for better robustness

$$\beta_1 = \min(\beta_{1,\text{Green-Gauss}}, \beta_{1,\text{least-square}}),$$

where $\beta_{1,\text{Green-Gauss}}$ is the smoothness indicator defined by Green-Gauss reconstruction, and $\beta_{1,\text{least-square}}$ is the smoothness indicator defined by second-order least-square reconstruction. Then, the smoothness indicators β_i are non-dimensionalized by

$$\tilde{\beta}_i = \frac{\beta_i}{Q_0^2 + \beta_1 + 10^{-40}}.$$

The nondimensionalized global smoothness indicator $\tilde{\sigma}$ can be defined as

$$\tilde{\sigma} = \left| \tilde{\beta}_1 - \tilde{\beta}_0 \right|.$$

Therefore, the corresponding nonlinear weights are given by

$$\tilde{\omega}_m = \gamma_m \left(1 + \left(\frac{\tilde{\sigma}}{\epsilon + \tilde{\beta}_m} \right)^2 \right), \epsilon = 10^{-5},$$

$$\bar{\omega}_m = \frac{\tilde{\omega}_m}{\sum \tilde{\omega}_m}, m = 1, 2.$$

Replacing γ_m in equation (18) by $\bar{\omega}_m$, the final non-linear reconstruction can be obtained

$$R(\mathbf{x}) = \bar{\omega}_2 P_2 + \bar{\omega}_1 P_1.$$

The desired non-equilibrium states at Gaussian points become

$$Q_{p,k}^{l,r} = R^{l,r}(\mathbf{x}_{p,k}), (Q_{x_i}^{l,r})_{p,k} = \frac{\partial R^{l,r}}{\partial x_i}(\mathbf{x}_{p,k}).$$

4.5. Reconstruction of equilibrium states

After reconstructing the non-equilibrium state, a kinetic weighted average method can be used to get equilibrium states and tangential derivatives [8],

$$\begin{aligned} \int \bar{g}\Psi \, d\mathbf{v} \, d\Xi &= \mathbf{W}_0 = \int_{\tilde{w}_1 > 0} g^l \Psi \, d\mathbf{v} \, d\Xi + \int_{\tilde{w}_1 < 0} g^r \Psi \, d\mathbf{v} \, d\Xi, \\ \int \bar{a}_i \bar{g} \Psi \, d\mathbf{v} \, d\Xi &= \frac{\partial \mathbf{W}_0}{\partial \tilde{x}_i} = \int_{\tilde{w}_1 > 0} a_i^l g^l \Psi \, d\mathbf{v} \, d\Xi + \int_{\tilde{w}_1 < 0} a_i^r g^r \Psi \, d\mathbf{v} \, d\Xi, i = 2, 3. \end{aligned}$$

For the normal derivatives, the above solution is further modified according to the idea in linear diffusive generalized Riemann problem (dGRP) [4]

$$\int \bar{a}_1 \bar{g} \Psi \, d\mathbf{v} \, d\Xi = \frac{\partial \mathbf{W}_0}{\partial \tilde{x}_1} = \int_{\tilde{w}_1 > 0} a_1^l g^l \Psi \, d\mathbf{v} \, d\Xi + \int_{\tilde{w}_1 < 0} a_1^r g^r \Psi \, d\mathbf{v} \, d\Xi + \frac{\mathbf{W}^r - \mathbf{W}^l}{(\mathbf{x}_{rc} - \mathbf{x}_{lc}) \cdot \mathbf{n}}, \quad (19)$$

where \mathbf{x}_{rc} and \mathbf{x}_{lc} are the coordinates of left and right cell centroid, \mathbf{W}^l and \mathbf{W}^r are the reconstructed left and right conservative value at Gaussian points and \mathbf{n} is the normal vector of interface. By adding a penalty term in Eq. (19), the whole scheme is essentially free from the odd-even decoupling phenomenon [1].

5. Numerical Experiments

Both two-dimensional and three-dimensional cases are tested in this section. For two-dimensional problems, the three-dimensional solver is used with one layer and mirror boundary condition in the z direction. The time step is given by $\Delta t = \min \Delta t_i$, where Δt_i is defined in each cell

$$\Delta t_i = C_{CFL} \min\left(\frac{h}{|V|_i + c_i}, \frac{h^2}{3\nu_i}\right),$$

where C_{CFL} is the CFL number, $|V|_i$, c_i and $\nu_i = (\mu/\rho)_i$ is the magnitude of related velocities, sound speed and kinematic viscosity coefficient of cell i . Here, we set the CFL number as 0.5 if not specified.

5.1. Accuracy Test

The advection of density perturbation for three-dimensional flow is presented to test the order of accuracy. The computation domain is $[0, 1] \times [0, 1] \times [0, 1]$. Three uniform meshes

with $10^3, 20^3, 40^3$ tetrahedral cells are used in this case. The flow at time t is

$$\begin{aligned}\rho(x, y, z) &= 1 + 0.2 \sin(2\pi x_r) \sin(2\pi y_r) \sin(2\pi z_r), \\ p(x, y, z) &= 1, V_1 = V_2 = V_3 = 1,\end{aligned}$$

where the (x_r, y_r, z_r) depends on time

$$(x_r, y_r, z_r) = (x, y, z) - (1, 1, 1)t.$$

In order to validate the order of accuracy with moving meshes, the time-dependent moving meshes are considered in this form

$$\begin{cases} x &= x_0 + 0.05 \sin(\pi t) \sin(2\pi x_0) \sin(2\pi y_0) \sin(2\pi z_0) \\ y &= y_0 + 0.05 \sin(\pi t) \sin(2\pi x_0) \sin(2\pi y_0) \sin(2\pi z_0) \\ z &= z_0 + 0.05 \sin(\pi t) \sin(2\pi x_0) \sin(2\pi y_0) \sin(2\pi z_0). \end{cases}$$

The errors and numerical orders are shown in Table 1, which shows that third-order accuracy is achieved under moving mesh. As a reference, the accuracy test result on stationary mesh

Table 1: The errors and accuracy in moving mesh

mesh	$Error_{L^1}$	O_{L^1}	$Error_{L^2}$	O_{L^2}	$Error_{L^\infty}$	O_{L^∞}
10^3	1.40E-2		1.83E-02		5.17E-02	
20^3	2.25E-3	2.64	2.96E-03	2.63	8.02E-03	2.69
40^3	3.04E-4	2.88	3.94E-04	2.91	1.08E-03	2.89

is shown in Table 2, which indicates that the errors of moving and fixed mesh are compatible with each other.

Table 2: The errors and accuracy in stationary mesh

mesh	$Error_{L^1}$	O_{L^1}	$Error_{L^2}$	O_{L^2}	$Error_{L^\infty}$	O_{L^∞}
10^3	1.38E-02		1.81E-02		5.16E-02	
20^3	2.07E-03	2.74	2.79E-03	2.70	7.92E-03	2.70
40^3	2.68E-04	2.95	3.61E-04	2.95	1.05E-03	2.92

Next, geometric conservation law, which is about the maintenance of uniform flow passing through the moving mesh, is tested. The same mesh and same mesh moving method are set in the test. The initial condition is set as

$$\rho(x, y, z) = 1, p(x, y, z) = 1, V_1 = V_2 = V_3 = 1.$$

The density errors at $t = 1$ are shown in Table 3. The results indicate that the errors are reduced to machine zero, implying the satisfaction of the geometric conservation law.

Table 3: The errors of geometric conservation law in moving mesh

mesh	$Error_{L^1}$	$Error_{L^2}$	$Error_{L^\infty}$
10^3	1.20148e-15	1.49810e-15	6.21725e-15
20^3	1.92325e-15	2.49943e-15	1.43219e-14
40^3	4.19724e-15	6.13169e-15	9.50351e-14

5.2. Flow inside a cylinder

This case is provided by High-Fidelity CFD Verification Workshop [19]. It is about the flow inside a cylinder's interior volume. Five meshes (with mesh number $4 \times 4 \times 5, 8 \times 8 \times 5, 16 \times 16 \times 5, 32 \times 32 \times 5, 64 \times 64 \times 5$) are tested in this case. The center of the cylinder is initially located on the original point, and the initial radius of the cylinder is $r_0 = 0.5$. The motion of meshes is defined by

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & \alpha(t) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(A_\theta\alpha(t)) & -\sin(A_\theta\alpha(t)) & 0 \\ \sin(A_\theta\alpha(t)) & \cos(A_\theta\alpha(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \psi(t) & 0 & 0 \\ 0 & \frac{1}{\psi(t)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_0 \cos(\theta_g(t)) \\ r_0 \sin(\theta_g(t)) \\ 1 \end{bmatrix},$$

as the composite of four motions (translation, rotation, const-volume deformation, and non-unit geometry mapping Jacobian). The translation motion is a motion in y direction as $\alpha(t)$. And the rotation motion is about the center of the cylinder with a rotation amplitude $A_\theta = \pi$. The transformation of the cylinder deforming into an ellipse such that the interior area remains constant during deformation is facilitated by the function

$$\psi(t) = 1 + (A_a - 1)\alpha(t),$$

which varies from 1 to $A_a = 1.5$ over $t = [0, 2]$. Next, a parameterized function η is defined as

$$\eta(\lambda, \omega, \epsilon) = \sin(\omega\lambda + \epsilon(1 - \cos(\omega\lambda))),$$

which is designed to break spatial and temporal symmetries due to the fact that the integral of η over a period ω does not equal 0 for appropriate values of ϵ . The parameter λ represents the independent variable (e.g. t or θ), whereas ω and ϵ represent the function frequency and shape characteristics. A subsequent function f_g is defined here to prescribe a deformation that ensures non-unit geometry mapping Jacobians ($g \neq 1$) while utilizing the function η to break spatial and temporal symmetries. This is given by

$$f_g(t, r_0, \theta_0) = \left(16r_0^4 + \frac{t^6}{t^6 + 0.01} \eta(t, 10, 0.7) (\cos(32\pi r_0^4) - 1) \right) \eta(\theta_0, 1, 0.7)$$

The prescribed deformation takes the form of a perturbation in θ as

$$\theta_g(t, r_0, \theta_0) = \theta_0 + A_g f_g(t, r_0, \theta_0),$$

which the volume deformation amplitude A_g is set as 0.15. The origin mesh, deforming mesh when $\alpha = 0$ and $\alpha = 1$ are shown in Figure 3, which indicates the large deforming and non-symmetry during the simulation.

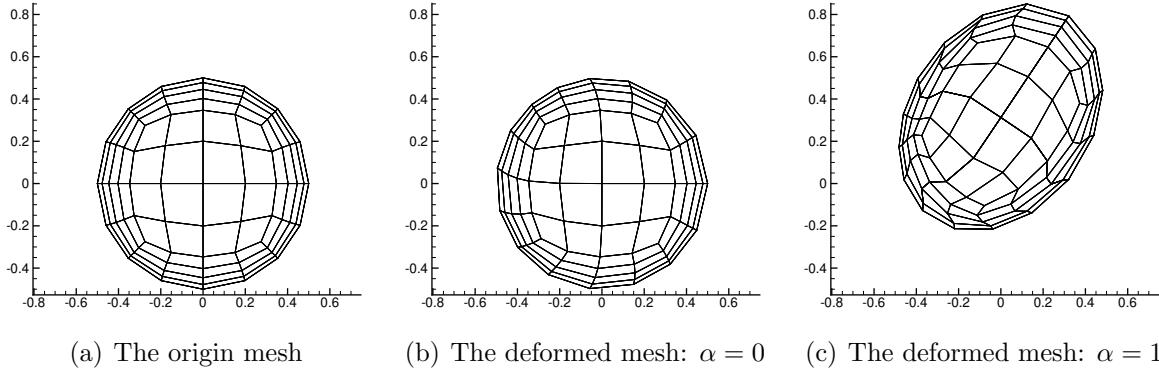


Figure 3: An example mesh with cell number $8 \times 8 \times 5$

The cylinder interior is prescribed with a no-slip, adiabatic wall boundary condition. The Reynolds number defined by cylinder diameter $d = 1$ and reference velocity $V = 1.0$ is $Re = 1000$, a Prandtl number of 0.72, and constant viscosity is used. The initial condition is given by a constant zero velocity field, which $\rho = 1$ and $\rho E = 50$.

5.2.1. Short Time Study

The time-activation function $\alpha(t)$ for this case is defined by

$$\alpha(t) = t^3(8 - 3t)/16,$$

which varies from 0 to 1 on the interval $t = [0, 2]$. This case runs from $t = 0$ until $t = 1$. The time history of force in x and y direction on the surface of the cylinder and power defined by

$$P = \int_{\text{surface}} \mathbf{U} \cdot \mathbf{f}_{\text{surf}} \, ds,$$

where \mathbf{f}_{surf} is the surface stress vector, are shown in Figure 4. The results indicate that the solutions obtained from Mesh 3, Mesh 4, and Mesh 5 are in good agreement with each other. This suggests that we have achieved mesh-independent solutions.

5.2.2. Long-time Study of Conservation

The motion of the long-time study has the same form as that of the composite motion for the short-time study but with a different $\alpha(t)$ function. The $\alpha(t)$ is selected as

$$\alpha(t) = 0.3 \sin(6t) \frac{t^6}{t^6 + 1}.$$

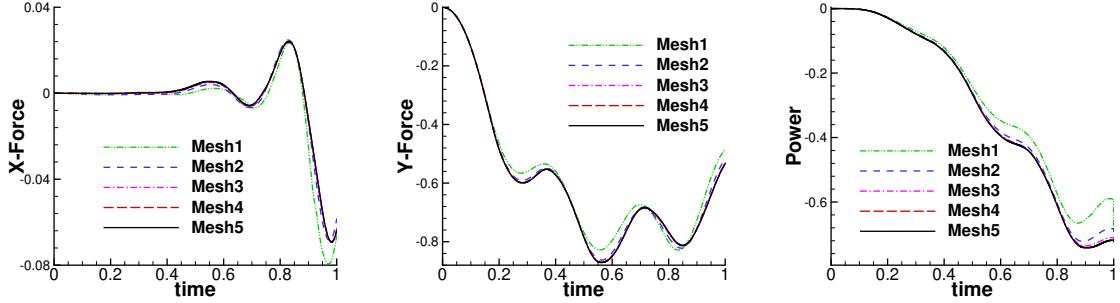


Figure 4: The time history force and power of short-time motion

The purpose of the second term in the equation is to start the sinusoidal motion slowly at $t = 0$. We simulate the process from $t = 0$ to $t = 40$. The mesh with $32 \times 32 \times 5$ cells is used in this study. The time history of mass, force in y direction on the surface of the cylinder, and power are shown in Figure 5. The total mass of the entire domain remains constant at

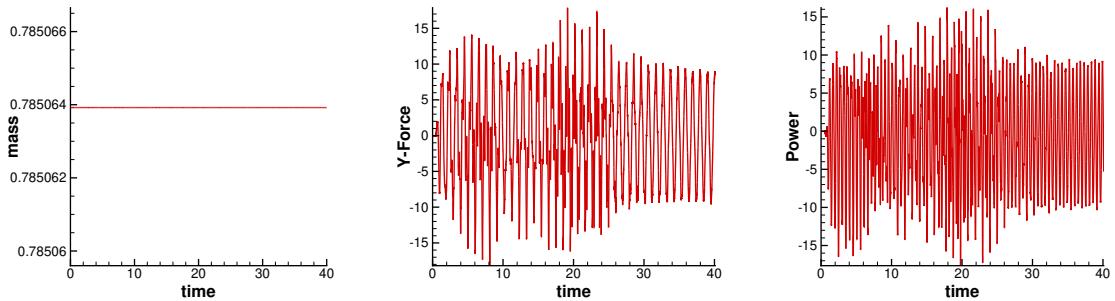


Figure 5: The time history force and power of long-time motion

0.785065 throughout the process, which indicates the mass conservation properties of our method. Additionally, the power and Y-Force exhibit periodic behavior after $t = 30$.

5.3. Heaving-pitching airfoil

This case considers a NACA0012 airfoil undergoing a smooth flapping-type motion. The airfoil moves smoothly upward for a distance of one chord length over a duration of $T = 2$ time units by heaving and pitching about a point located at the airfoil's $1/3$ chord location. The functions governing the motion are provided below:

$$\begin{aligned}\Delta h(t) &= t^3(8 - 3t)/16, \\ \Delta\theta(t) &= \frac{80\pi}{180}(-t^6 + 6t^5 - 12t^4 + 8t^3).\end{aligned}$$

Two types of motion are considered in this case. The first type of motion is defined solely by the heaving function $\Delta h(t)$, while the second type of motion is defined by both the heaving function $\Delta h(t)$ and the pitching function $\Delta\theta(t)$.

The incoming flow is set as $\rho_\infty = 1.0$, $T_\infty = 25$, $U_\infty = 1$, which has a Mach number of 0.2. The Reynolds number based on the chord of the airfoil and incoming velocity is $Re = 1000$, a Prandtl number of 0.72, and a constant viscosity is used. As shown in figure 6, three meshes are used in this case. The adiabatic non-slip wall is set on the airfoil surface, and the far-field

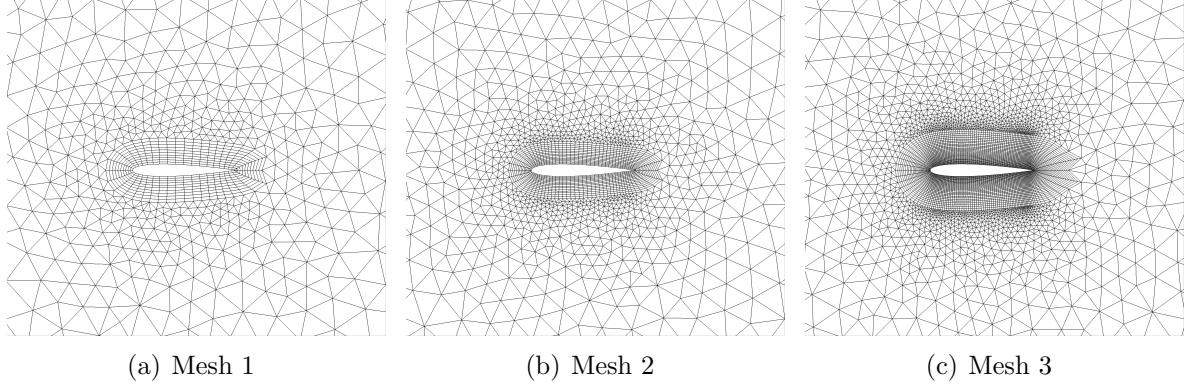


Figure 6: Mesh of NACA 0012

boundary condition is set as the outer boundary. The initial condition is the steady-state solution for the initial position $\Delta h = \Delta\theta = 0$. The radius basic function method is used for mesh motion.

Figure 7 displays the time history of the force in the x and y directions, as well as the power. The results indicate that the three meshes produce nearly identical solutions for the force in the y direction and power. Although the solutions for the force in the x direction appear to differ slightly among the meshes, the difference is negligible since the force in this direction is very small. Figure 8 illustrates the vorticity contour and mesh distribution at

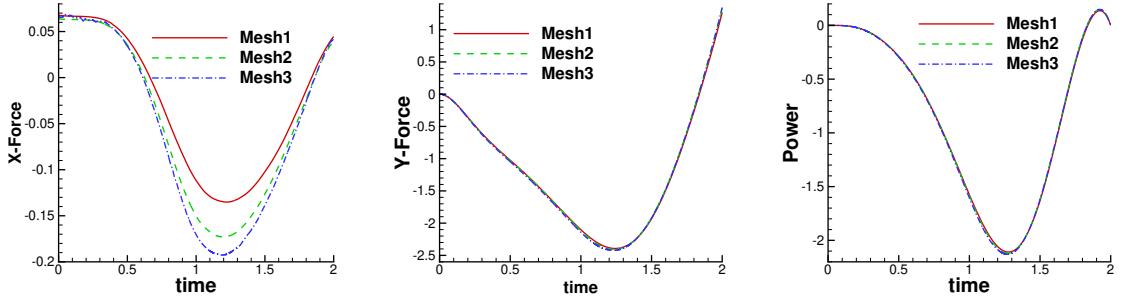


Figure 7: NACA0012: force and power history of Motion 1 (translation)

$t = 1.5$. At this time, the airfoil has moved approximately 0.738 times its chord length in the y direction, while the mesh surrounding the airfoil remains of good quality. The vorticity contour indicates the generation of a vortex at the front edge of the airfoil.

Figure 9 displays the force and power history of Motion 2. The results indicate that the solutions obtained from Mesh 2 and Mesh 3 are in good agreement, suggesting that the

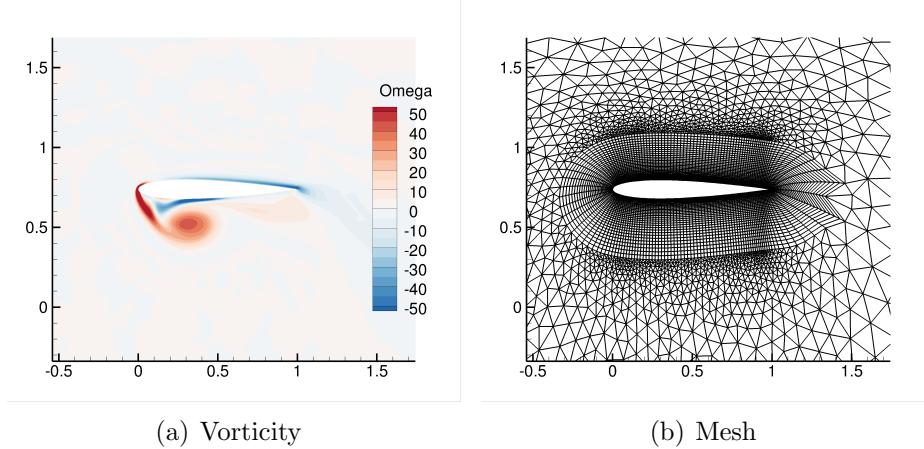


Figure 8: NACA0012: the solution of Motion 1 (translation) at time $t = 1.5$

solutions are mesh-independent. Figure 10 depicts the solution at time $t = 0.5$ when the

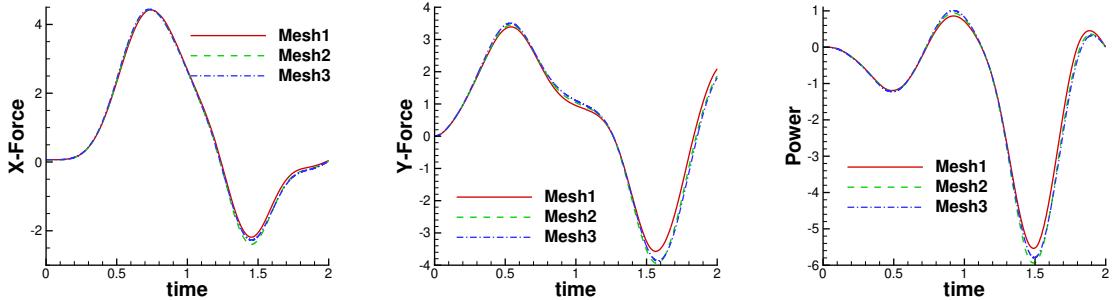


Figure 9: NACA0012: force and power history of Motion 2 (translation + rotation)

airfoil has rotated 33.75 degrees. At this time, the mesh remains of high quality with only slight distortion at the trailing edge. The vorticity contour shows the shedding of a vortex at the trailing edge.

5.4. 2D Noh Problem

This case is a typical test case for verifying ALE code. A two-dimensional domain with dimensions $[0, 1] \times [0, 1]$ is considered. An unstructured mesh with mesh size $h = 1/40$ is used for the computation, as shown in Figure 11(a). The initial conditions are set to uniform density $\rho = 1$ and internal energy $e = \frac{p}{\gamma-1} = 1 \times 10^{-4}$, with $\gamma = 5/3$. The initial velocity is given by $\mathbf{V} = (-x/r, -y/r)$, where r is the distance from the origin. An axially symmetric shock is generated at the origin and propagates at a constant speed. At time $t = 0.6$, the shock is located at $r = 0.2$. An exact solution of density [15] is

$$\rho = \begin{cases} 16, & r < 0.2 \\ 1 + t/r, & r > 0.2 \end{cases}.$$

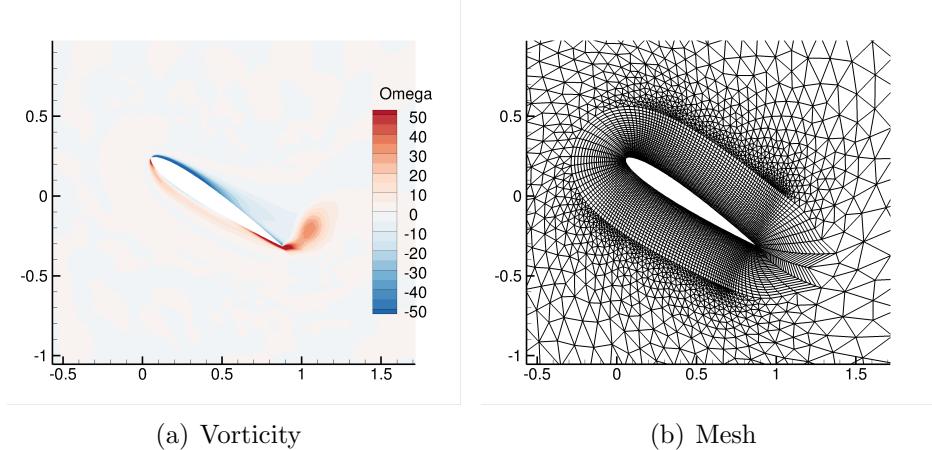


Figure 10: NACA0012: the solution of Motion 2 (translation + rotation) at time $t = 0.5$

An asymmetric boundary condition is enforced at the lines $x = 0$ and $y = 0$, while a non-reflective boundary condition is used at all other boundaries. Mesh movement is achieved using Lagrangian velocity, and a smoothing process is applied every 20 time-steps with a relaxation coefficient $\omega = 0.4$. The CFL number is set to 0.3 for this case. The final mesh is shown in Figure 11(b), and it is observed that the mesh is concentrated in the post-shock region. The density contour at time $t = 0.6$ is shown in Figure 12(a), and the distribution

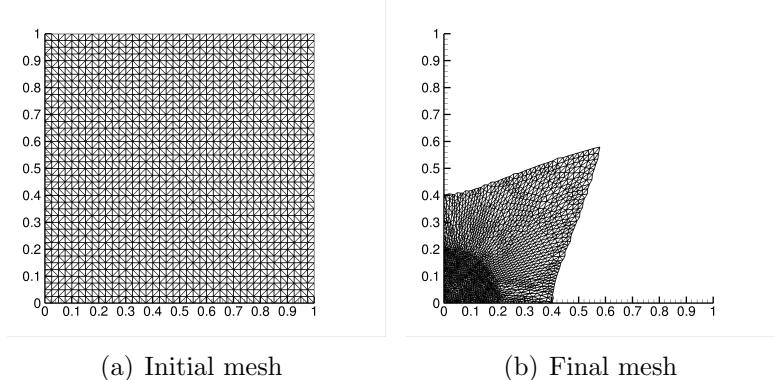


Figure 11: The mesh used in Noh Problem

with respect to r of the entire domain is presented in Figure 12(b), which agrees well with the exact solution.

5.5. Sedov Problem

The Sedov blast wave is a standard hydrodynamics test problem that models a blast wave emanating from a singular energy point. It is a benchmark problem for both the Lagrangian velocity method and the ALE method. The computational domain is $[0, 1.2]^3$, and two uniform grids with 20^3 and 40^3 cells are used in this test case. The fluid is modeled

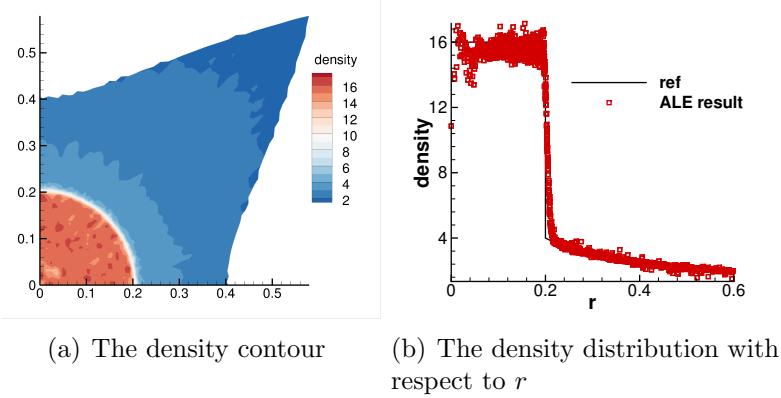


Figure 12: The result of Noh problem at $t = 0.6$

by the ideal gas equation of state with $\gamma = 1.4$. The initial condition is a uniform static field with density $\rho = 1$ and pressure $p = 1e - 6$, except for the cell containing the origin. In this cell, the initial pressure is defined by $p = (\gamma - 1)\epsilon_0/V$, where $\epsilon_0 = 0.106384$ is the total amount of released energy and V is the cell volume. Asymmetric boundary condition is enforced at the plane $x = 0$, $y = 0$ and $z = 0$, while non-reflecting boundary conditions are used at the plane $x = 1.2$, $y = 1.2$ and $z = 1.2$. Mesh movement is accomplished using Lagrangian velocity, and a smoothing process is applied every ten time-steps with a relaxation coefficient of $\omega = 0.8$. A small CFL number of 0.01 is used to avoid the instability caused by the initial singularity at the origin. After ten steps, the simulation transitions to a normal CFL number of 0.3. Figure 13 displays the mesh distribution and density contour, which reveal that the mesh is refined adaptively around the shock. Figure 14 displays the

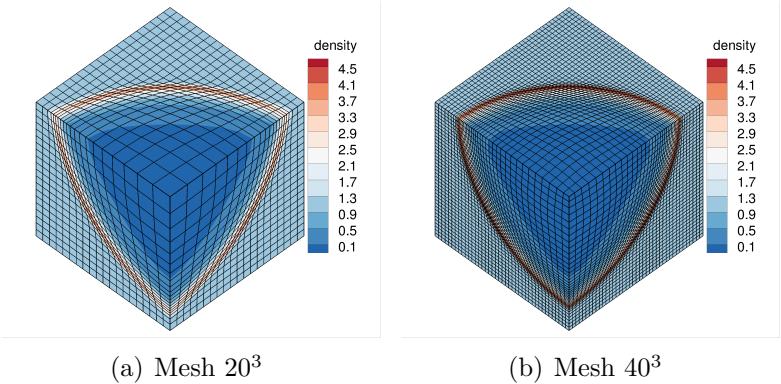


Figure 13: Sedov problem: the mesh distribution and density contour at $t = 1$

density and pressure distribution along the diagonal from $(0, 0, 0)$ to $(1.2, 1.2, 1.2)$, as well as the solutions of a uniform stationary mesh. The results show that the solution of the ALE method agrees much more closely with the exact solution than the static method. The solutions at the extreme points of the ALE method are closer to the exact solution

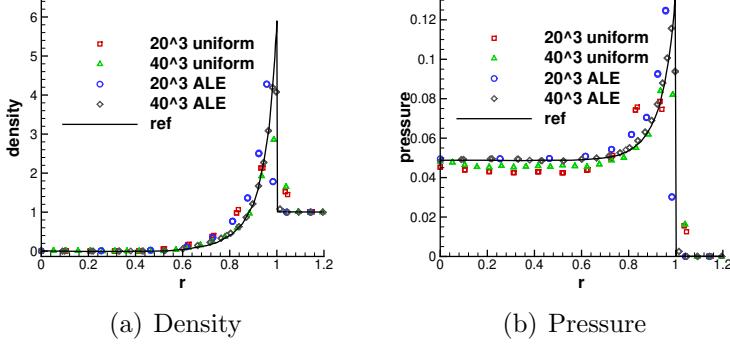


Figure 14: Sedov problem: the density and pressure distribution along diagonal line

than those of the uniform mesh. Additionally, the fine mesh ALE solution provides a more accurate solution for the shock position than the coarse mesh one.

5.6. Saltzmann Problem

The Saltzmann problem is a test case that simulates fluid in a cylinder with a piston. It examines the motion of a planar shock on a skewed Cartesian grid. The computational domain is $[0, 1] \times [0, 0.1] \times [0, 0.1]$, with uniform $h = 0.01$ in three directions. A mesh distortion of x -coordinate is defined by

$$\Delta x = (0.5y + z - 15yz) \sin(\pi x).$$

The initial mesh is shown in Figure 15. Initially, an ideal monatomic gas with $\rho = 1, e = 10^{-4}, \gamma = 5/3$ is filled in the box. The wall on the left-hand side moves at a constant speed of 1 from left to right, acting as a piston. The other boundaries act as inviscid reflecting walls. As the gas is compressed, a shock is generated and propagates to the right faster than the piston. Mesh movement is accomplished using Lagrangian velocity, and a smoothing process is applied every 20 time steps with a relaxation coefficient of $\omega = 0.8$. Figure 16 displays the entire solution of the domain at different times, indicating that the solution tends towards a one-dimensional solution. The drop in density may be due to the entropy increase at the wall, and further high-order boundary conditions will be considered. At $t = 0.6$, the

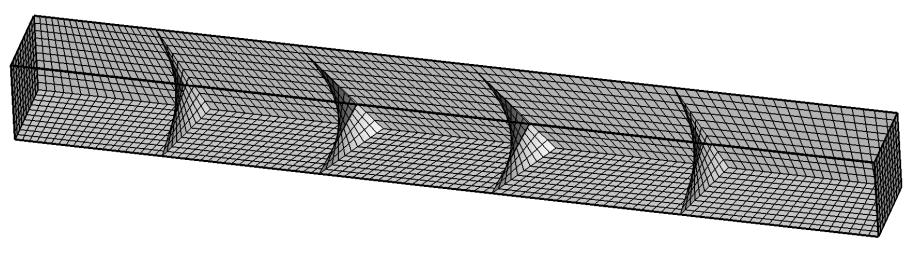


Figure 15: The initial mesh of Saltzmann problem

$10^{-4}, \gamma = 5/3$ is filled in the box. The wall on the left-hand side moves at a constant speed of 1 from left to right, acting as a piston. The other boundaries act as inviscid reflecting walls. As the gas is compressed, a shock is generated and propagates to the right faster than the piston. Mesh movement is accomplished using Lagrangian velocity, and a smoothing process is applied every 20 time steps with a relaxation coefficient of $\omega = 0.8$. Figure 16 displays the entire solution of the domain at different times, indicating that the solution tends towards a one-dimensional solution. The drop in density may be due to the entropy increase at the wall, and further high-order boundary conditions will be considered. At $t = 0.6$, the

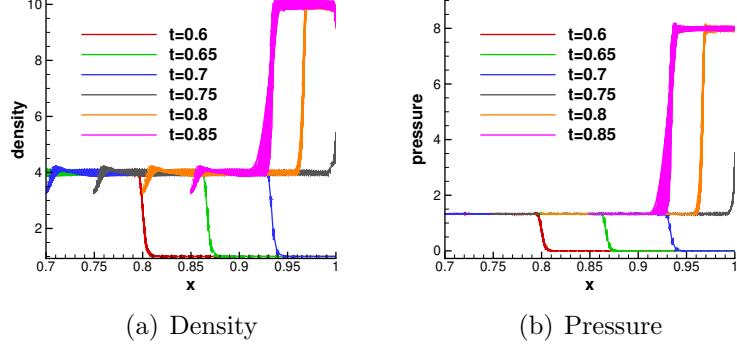


Figure 16: Saltzmann problem: the density and pressure distribution at different time

shock is located at $x = 0.8$, and the post-shock solutions are $\rho = 4$ and $p = 1.333$. At time $t = 0.75$, the shock will reach the right boundary and bounce back due to the wall. Figure 17 demonstrates that the Lagrangian velocity method is capable of handling shock motion issues, as it shows the mesh distribution at different time intervals.

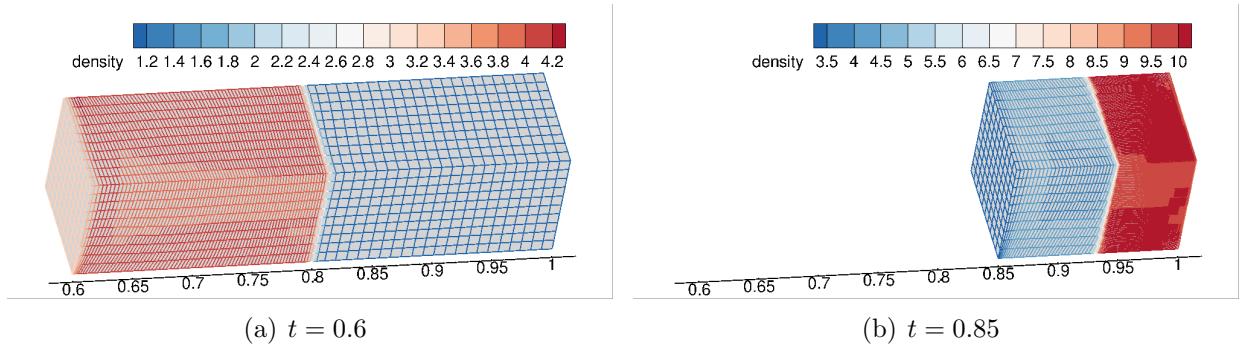


Figure 17: Saltzmann problem: mesh and density distribution at different time

6. Conclusion

In this paper, a third-order compact gas-kinetic scheme is developed in an arbitrary Lagrangian-Eulerian (ALE) formulation for three-dimensional unstructured mesh. In order to achieve geometric consistency, the arbitrary mesh is subdivided into tetrahedrons, and the quadrilateral surface is split into two triangles. By accounting for the variation of grid velocity within the interface surface and dealing with it in a local framework, the geometric conservation law is preserved. The kinetic equation at the moving frame with grid velocity is used to construct a time-accurate gas distribution function at the Gaussian point of the cell interface, allowing for updating both cell-averaged flow variables and their gradients for the use in the high-order compact initial reconstruction. The scheme is validated using several test cases. For close boundary problems, mass conservation is well-preserved. With

the help of radial basis function interpolation, the method can handle moving boundary problems and obtain high-fidelity solutions. For blast problems with strong discontinuities, the scheme retains robustness, and the Lagrangian nodal solver is adapted to track material interfaces.

Acknowledgements

The current research is supported by the National Key R&D Program of China 2022YFA1004500, National Natural Science Foundation of China (Grant No. 12172316), and Hong Kong research grant council (16208021, 16301222).

References

- [1] Blazek, J., 2015. Chapter 4 - structured finite-volume schemes, in: Blazek, J. (Ed.), Computational Fluid Dynamics: Principles and Applications (Third Edition). third edition ed.. Butterworth-Heinemann, Oxford, pp. 73–120.
- [2] Brackbill, J., Saltzman, J., 1982. Adaptive zoning for singular problems in two dimensions. *Journal of Computational Physics* 46, 342–368. doi:10.1016/0021-9991(82)90020-1.
- [3] De Boer, A., Van der Schoot, M.S., Bijl, H., 2007. Mesh deformation based on radial basis function interpolation. *Computers & structures* 85, 784–795.
- [4] Gassner, G., Lörcher, F., Munz, C.D., 2007. A contribution to the construction of diffusion fluxes for finite volume and discontinuous Galerkin schemes. *Journal of Computational Physics* 224, 1049–1063.
- [5] Hirt, C., Amsden, A., Cook, J., 1974. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *Journal of Computational Physics* 14, 227–253. doi:10.1016/0021-9991(74)90051-5.
- [6] Ji, X., Pan, L., Shyy, W., Xu, K., 2018. A compact fourth-order gas-kinetic scheme for the Euler and Navier–Stokes equations. *Journal of Computational Physics* 372, 446–472.
- [7] Ji, X., Shyy, W., Xu, K., 2021a. A gradient compression-based compact high-order gas-kinetic scheme on 3D hybrid unstructured meshes. *International Journal of Computational Fluid Dynamics* 35, 485–509.
- [8] Ji, X., Zhao, F., Shyy, W., Xu, K., 2020. A HWENO reconstruction based high-order compact gas-kinetic scheme on unstructured mesh. *Journal of Computational Physics* 410, 109367.
- [9] Ji, X., Zhao, F., Shyy, W., Xu, K., 2021b. Compact high-order gas-kinetic scheme for three-dimensional flow simulations. *AIAA Journal* 59, 2979–2996.
- [10] Ji, X., Zhao, F., Shyy, W., Xu, K., 2021c. Two-step multi-resolution reconstruction-based compact gas-kinetic scheme on tetrahedral mesh. arXiv:2102.01366 [physics] ArXiv: 2102.01366.
- [11] Kucharik, M., Shashkov, M., 2014. Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics* 258, 268–304. doi:10.1016/j.jcp.2013.10.050.
- [12] Li, J., 2019. Two-stage fourth order: temporal-spatial coupling in computational fluid dynamics (cfd). *Advances in Aerodynamics* 1, 1–36. doi:10.1186/s42774-019-0004-9.
- [13] Li, J., Du, Z., 2016. A two-stage fourth order time-accurate discretization for Lax–Wendroff type flow solvers I. hyperbolic conservation laws. *SIAM Journal on Scientific Computing* 38, A3046–A3069.
- [14] Maire, P.H., Nkonga, B., 2013. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics* , 799–821doi:10.1016/j.jcp.2008.10.012.
- [15] Noh, W., 1987. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics* 72, 78–120. doi:10.1016/0021-9991(87)90074-X.
- [16] Pan, L., Xu, K., 2021. An Arbitrary-Lagrangian-Eulerian high-order gas-kinetic scheme for three-dimensional computations. *Journal of Scientific Computing* 88, 1–26. doi:10.1007/s10915-021-01525-9.

- [17] Pan, L., Zhao, F., Xu, K., 2020. High-order ALE gas-kinetic scheme with WENO reconstruction. *Journal of Computational Physics* 417, 109558. doi:10.1016/j.jcp.2020.109558.
- [18] Ren, X., Xu, K., Shyy, W., 2016. A multi-dimensional high-order DG-ALE method based on gas-kinetic theory with application to oscillating bodies. *Journal of Computational Physics* 316, 700–720. doi:10.1016/j.jcp.2016.04.028.
- [19] Wukie, N.A., Fidkowski, K., Persson, P.O., Wang, Z.J., 2023. High-fidelity CFD verification workshop 2024: Mesh motion, in: AIAA SCITECH 2023 Forum, p. 1243.
- [20] Xu, K., 1998. Gas-kinetic schemes for unsteady compressible flow simulations. *Computational Fluid Dynamics, Annual Lecture Series*, 29 th, Rhode-Saint-Genese, Belgium .
- [21] Xu, K., 2001. A gas-kinetic BGK scheme for the Navier–Stokes equations and its connection with artificial dissipation and Godunov method. *Journal of Computational Physics* 171, 289–335.
- [22] Zhang, Y., Ji, X., Xu, K., 2023. A high-order compact gas-kinetic scheme in a rotating coordinate frame and on sliding mesh. *International Journal of Computational Fluid Dynamics* , 1–20.
- [23] Zhao, F., Ji, X., Shyy, W., Xu, K., 2019. Compact higher-order gas-kinetic schemes with spectral-like resolution for compressible flow simulations. *Advances in Aerodynamics* 1, 13.
- [24] Zhao, F., Ji, X., Shyy, W., Xu, K., 2020. An acoustic and shock wave capturing compact high-order gas-kinetic scheme with spectral-like resolution. *International Journal of Computational Fluid Dynamics* 34, 731–756.
- [25] Zhao, F., Ji, X., Shyy, W., Xu, K., 2021. Direct modeling for computational fluid dynamics and the construction of high-order compact scheme for compressible flow simulations. arXiv:2107.06555 [physics] ArXiv: 2107.06555.
- [26] Zhao, F., Ji, X., Shyy, W., Xu, K., 2022. A compact high-order gas-kinetic scheme on unstructured mesh for acoustic and shock wave computations. *Journal of Computational Physics* 449, 110812.
- [27] Zhu, J., Qiu, J., 2018. New finite volume weighted essentially nonoscillatory schemes on triangular meshes. *SIAM Journal on Scientific Computing* 40, A903–A928.
- [28] Zhu, J., Shu, C.W., 2020. A new type of third-order finite volume multi-resolution WENO schemes on tetrahedral meshes. *Journal of Computational Physics* 406, 109212.