Journal of Computational Physics 505 (2024) 112902

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Implicit high-order gas-kinetic schemes for compressible flows on three-dimensional unstructured meshes I: Steady flows

Yaqing Yang [a,b], Liang Pan [a,*], Kun Xu [b,c]

[a] Laboratory of Mathematics and Complex Systems, School of Mathematical Sciences, Beijing Normal University, Beijing, China
[b] Department of Mathematics, 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

ARTICLE INFO

Keywords:
High-order gas-kinetic scheme
Implicit method
Unstructured meshes
GPU accelerated computation

ABSTRACT

In the previous studies, the high-order gas-kinetic schemes (HGKS) have achieved successes for unsteady flows on three-dimensional unstructured meshes. In this paper, to accelerate the rate of convergence for steady flows, the implicit non-compact and compact HGKSs are developed. For non-compact scheme, the simple weighted essentially non-oscillatory (WENO) reconstruction is used to achieve the spatial accuracy, where the stencils for reconstruction contain two levels of neighboring cells. Incorporate with the nonlinear generalized minimal residual (GMRES) method, the implicit non-compact HGKS is developed. In order to improve the resolution and parallelism of non-compact HGKS, the implicit compact HGKS is developed with Hermite WENO (HWENO) reconstruction, in which the reconstruction stencils only contain one level of neighboring cells. The cell averaged conservative variable is also updated with GMRES method. Simultaneously, a simple strategy is used to update the cell averaged gradient by the time evolution of spatial-temporal coupled gas distribution function. To accelerate the computation, the implicit non-compact and compact HGKSs are implemented with the graphics processing unit (GPU) using compute unified device architecture (CUDA). A variety of numerical examples, from the subsonic to supersonic flows, are presented to validate the accuracy, robustness and efficiency of both inviscid and viscous flows.

1. Introduction

The simulation of compressible flows with complex geometry is an important issue for computational fluid dynamics, and the unstructured meshes are widely used due to the flexibility. For the spatial reconstruction, various high-order numerical methods on unstructured meshes have been developed in the past decades, such as essential non-oscillatory (ENO) [1] and weighted essential non-oscillatory (WENO) [14,46], Hermite WENO (HWENO) methods [12,26,48,27], discontinuous Galerkin (DG) [10,8], flux reconstruction (FR) [15] and correction procedure using reconstruction (CPR) [37], etc. For the temporal discretization, early efforts mainly focused on explicit schemes on unstructured meshes, and the most widely used method is the Runge-Kutta schemes [13]. However, for steady flows, the rate of convergence slows down dramatically. In order to speed up the convergence, the implicit temporal discretization is required. In general, the implicit method requires to solve a large system of equation, which arises from the

* Corresponding author.
E-mail addresses: yqyangbnu@163.com (Y. Yang), panliang@bnu.edu.cn (L. Pan), makxu@ust.hk (K. Xu).

https://doi.org/10.1016/j.jcp.2024.112902
Received 25 April 2023; Received in revised form 27 January 2024; Accepted 26 February 2024

Available online 1 March 2024

linearization of a fully implicit scheme at each time step. Several methods are used to solve the large sparse system on unstructured meshes, including approximate factorization methods and iterative solution methods. As an approximate factorization method, the lower-upper symmetric Gauss-Seidel (LUSGS) method on structured meshes was originally developed by Jameson and Yoon [43], and it has been successfully extended to unstructured meshes [31,7]. The most attractive feature of this method is that it does not require any extra memory compared with the explicit methods and is free from any matrix inversion. However, LUSGS method is not ideally effective because of slow convergence, and requires thousands of time steps to achieve the steady state. As an iterative method, the most successful and effective method is the Krylov subspace method, such as the nonlinear generalized minimal residual (GMRES) method [29,4]. The GMRES method always has a faster convergence speed, but the drawback is that they require a considerable amount of memory to store the Jacobian matrix, which may be prohibitive for large problems. To save the storage, the matrix-free GMRES method has been applied to unstructured meshes for steady and unsteady flows [21,22]. To improve the convergence, a wide variety of preconditioners [11,20,35,40,47] are also applied to the methods above.

In the past decades, the gas-kinetic scheme (GKS) based on the Bhatnagar-Gross-Krook (BGK) model [3,6] has been developed systematically for the computations from low speed flows to supersonic ones [39,38]. The gas-kinetic scheme presents a gas evolution process from the kinetic scale to hydrodynamic scale, and both inviscid and viscous fluxes can be calculated in one framework. With the two-stage temporal discretization, which was originally developed for the Lax-Wendroff type flow solvers [18], a reliable framework was provided to construct gas-kinetic scheme with fourth-order and even higher-order temporal accuracy [24,25]. With the simple WENO type reconstruction, the third-order gas-kinetic schemes on three-dimensional unstructured meshes [41,42] are developed, in which a simple strategy of selecting stencils for reconstruction is adopted and the topology independent linear weights are used. Based on the spatial and temporal coupled property of GKS solver and HWENO reconstruction, the explicit high-order compact gas-kinetic schemes are also developed [16,17,44,45]. In the compact scheme, the time-dependent gas distribution function at a cell interface is used to calculate the fluxes for the updating the cell-averaged flow variables, and to evaluate the cell-averaged gradients of flow variables. Numerical results demonstrate that the superior robustness in high speed flow computation and the favorable mesh adaptability for complex geometry of high-order compact schemes. For the gas-kinetic scheme, several implicit algorithms have also been developed to simulate from continuum and rarefied flows [51,41,42,33]. The implicit methods provide efficient techniques for speeding up the convergence of steady flows.

In this paper, the implicit non-compact and compact HGKSs are developed on the three-dimensional unstructured meshes. For non-compact GKS scheme, the third-order WENO reconstruction is used, where the stencils are selected from the neighboring cells and the neighboring cells of neighboring cells. Incorporate with the GMRES method, the implicit non-compact scheme is developed for steady problems. In order to balance the computational efficiency and memory storage, the GMRES method is based on numerical Jacobian matrix with Roe's approximation. To improve the resolution and parallelism, the implicit compact HGKS is also developed with HWENO reconstruction, where the stencils only contain one level of neighboring cells. Since the cell averaged conservative variables and gradients need to be updated simultaneously, the GMRES method is associated with a suitable update strategy at each time step. For steady problems, the update of cell averaged gradient can be driven by time evolution of gas distribution function directly. To further accelerate the computation, the Jacobi iteration is chosen as preconditioner for both non-compact and compact schemes. Various three-dimensional numerical experiments, from the subsonic to supersonic flows, are presented to validate the accuracy and robustness of current implicit scheme. To accelerate the computation, the current schemes are implemented to run on graphics processing unit (GPU) using compute unified device architecture (CUDA). The GPU code is implemented with single Nvidia Quadro RTX 8000 GPU, and the CPU code is run with Intel Xeon Gold 6230R CPU with 16 OpenMP threads. Compared with the CPU code, 8x speedup is achieved for GPU code. In the future, more challenging compressible flow problems will be investigated with multiple GPUs.

This paper is organized as follows. In Section 2, BGK equation and finite volume scheme will be introduced. The third-order non-compact and compact gas-kinetic scheme will be presented in Section 3. Section 4 includes the implicit method and its parallel computation. Numerical examples are included in Section 5. The last section is the conclusion.

2. BGK equation and finite volume scheme

The Boltzmann equation expresses the behavior of a many-particle kinetic system in terms of the evolution equation for a single particle gas distribution function. The BGK equation [3,6] is the simplification of Boltzmann equation, and the three-dimensional BGK equation can be written as

$$f_t + uf_x + vf_y + wf_z = \frac{g-f}{\tau}, \tag{1}$$

where $\boldsymbol{u} = (u, v, w)$ is the particle velocity, τ is the collision time, f is the gas distribution function. g is the equilibrium state given by Maxwellian distribution

$$g = \rho(\frac{\lambda}{\pi})^{(N+3)/2} e^{-\lambda[(u-U)^2+(v-V)^2+(w-W)^2+\xi^2]},$$

where ρ is the density, $\boldsymbol{U} = (U, V, W)$ is the macroscopic fluid velocity, and λ is the inverse of gas temperature, i.e., $\lambda = m/2kT$. In the BGK model, the collision operator involves a simple relaxation from f to the local equilibrium state g. The variable ξ accounts for the internal degree of freedom, $\xi^2 = \xi_1^2 + \cdots + \xi_N^2$, $N = (5-3\gamma)/(\gamma-1)$ is the internal degree of freedom, and γ is the specific heat ratio. The collision term satisfies the compatibility condition

$$\int \frac{g-f}{\tau}\psi \mathrm{d}\Xi = 0,$$

where $\psi = (1, u, v, w, \frac{1}{2}(u^2+v^2+w^2+\xi^2))^T$ and $\mathrm{d}\Xi = \mathrm{d}u\mathrm{d}v\mathrm{d}w\mathrm{d}\xi_1 \dots \mathrm{d}\xi_N$. According to the Chapman-Enskog expansion for BGK equation, the macroscopic governing equations can be derived. In the continuum region, the BGK equation can be rearranged and the gas distribution function can be expanded as

$$f = g - \tau D_{\boldsymbol{u}} g + \tau D_{\boldsymbol{u}}(\tau D_{\boldsymbol{u}})g - \tau D_{\boldsymbol{u}}[\tau D_{\boldsymbol{u}}(\tau D_{\boldsymbol{u}})g] + ...,$$

where $D_{\boldsymbol{u}} = \frac{\partial}{\partial t} + \boldsymbol{u}\cdot\nabla$. With the zeroth-order truncation $f = g$, the Euler equations can be obtained. For the first-order truncation

$$f = g - \tau(ug_x + vg_y + wg_z + g_t),$$

the Navier-Stokes equations can be obtained [39,38].

Taking moments of Eq. (1) and integrating with respect to space, the semi-discretized finite volume scheme can be expressed as

$$|\Omega_i|\frac{\mathrm{d}Q_i}{\mathrm{d}t} = \mathcal{L}(Q_i), \tag{2}$$

where $Q_i = (\rho, \rho U, \rho V, \rho W, \rho E)$ is the cell averaged conservative value of Ω_i, ρ is the density, U, V, W is the flow velocity, ρE is the total energy density and $|\Omega_i|$ is the volume of Ω_i. The operator $\mathcal{L}$ is defined as

$$\mathcal{L}(Q_i) = -\sum_{i_p \in N(i)} F_{i,i_p}(t) S_{i_p} = -\sum_{i_p \in N(i)} \iint_{\Sigma_{i_p}} \boldsymbol{F}(Q,t)\mathrm{d}\sigma, \tag{3}$$

where Σ_{i_p} is the common cell interface of Ω_i, S_{i_p} is the area of Σ_{i_p} and $N(i)$ is the set of index for neighboring cells of Ω_i. To achieve the expected order of accuracy, the Gaussian quadrature is used for the flux integration

$$\iint_{\Sigma_{i_p}} \boldsymbol{F}(Q,t)\mathrm{d}\sigma = \sum_G \omega_G F_G(t) S_{i_p},$$

where ω_G is the quadrature weights. The numerical flux $F_G(t)$ at Gaussian quadrature point $\boldsymbol{x}_G$ can be given by taking moments of gas distribution function

$$F_G(t) = \int \boldsymbol{\psi}\boldsymbol{u}\cdot\boldsymbol{n}_G f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi)\mathrm{d}\Xi,$$

where $F_G(t) = (F_G^{\rho}, F_G^{\rho U}, F_G^{\rho V}, F_G^{\rho W}, F_G^{\rho E})$ and $\boldsymbol{n}_G$ is the local normal direction of cell interface. With the coordinate transformation, the numerical flux in the global coordinate can be obtained. Based on the integral solution of BGK equation Eq. (1), the gas distribution function $f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi)$ in the local coordinate can be given by

$$f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi) = \frac{1}{\tau}\int_0^t g(\boldsymbol{x}', t', \boldsymbol{u}, \xi) e^{-(t-t')/\tau}\mathrm{d}t' + e^{-t/\tau} f_0(-\boldsymbol{u}t, \xi),$$

where $\boldsymbol{x}' = \boldsymbol{x}_G - \boldsymbol{u}(t-t')$ is the trajectory of particles, f_0 is the initial gas distribution function, and g is the corresponding equilibrium state. With the first order spatial derivatives, the second-order gas distribution function at cell interface can be expressed as

$$\begin{aligned} f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi) =& (1-e^{-t/\tau})g_0 + ((t+\tau)e^{-t/\tau} - \tau)(\overline{a}_1 u + \overline{a}_2 v + \overline{a}_3 w)g_0 \\ &+ (t - \tau + \tau e^{-t/\tau})\bar{A} g_0 \\ &+ e^{-t/\tau} g_r[1 - (\tau + t)(a_1^r u + a_2^r v + a_3^r w) - \tau A^r)](1 - H(u)) \\ &+ e^{-t/\tau} g_l[1 - (\tau + t)(a_1^l u + a_2^l v + a_3^l w) - \tau A^l)]H(u), \end{aligned} \tag{4}$$

where the equilibrium state g_0 and the corresponding conservative variables Q_0 can be determined by the compatibility condition

$$\int \psi g_0 \mathrm{d}\Xi = Q_0 = \int_{u>0} \psi g_l \mathrm{d}\Xi + \int_{u<0} \psi g_r \mathrm{d}\Xi.$$

With the reconstruction of macroscopic variables, the coefficients in Eq. (4) can be fully determined by the reconstructed derivatives and compatibility condition

$$\langle a_1^k \rangle = \frac{\partial Q_k}{\partial \boldsymbol{n_x}}, \langle a_2^k \rangle = \frac{\partial Q_k}{\partial \boldsymbol{n_y}}, \langle a_3^k \rangle = \frac{\partial Q_k}{\partial \boldsymbol{n_z}}, \langle a_1^k u + a_2^k v + a_3^k w + A^k \rangle = 0,$$

$$\langle \overline{a}_1 \rangle = \frac{\partial Q_0}{\partial \boldsymbol{n_x}}, \langle \overline{a}_2 \rangle = \frac{\partial Q_0}{\partial \boldsymbol{n_y}}, \langle \overline{a}_3 \rangle = \frac{\partial Q_0}{\partial \boldsymbol{n_z}}, \langle \overline{a}_1 u + \overline{a}_2 v + \overline{a}_3 w + \overline{A} \rangle = 0,$$

where $k = l$ and r, $\boldsymbol{n_x}$, $\boldsymbol{n_y}$, $\boldsymbol{n_z}$ are the unit directions of local coordinate at $\boldsymbol{x}_G$ and $\langle ... \rangle$ are the moments of the equilibrium g and defined by

$$\langle ... \rangle = \int g(...)\psi \mathrm{d}\Xi.$$

More details of the gas-kinetic scheme can be found in [39,38].

3. Spatial reconstruction

To deal with the complex geometry, the three-dimensional unstructured meshes are considered, and the tetrahedral and hexahedral meshes are used in this paper for simplicity. In the previous studies, the high-order gas-kinetic schemes have be developed with the third-order non-compact WENO reconstruction [41,42] and compact HWENO reconstruction [17,44]. Successes have been achieved for the unsteady flows from the subsonic to supersonic flow problems. For the unstructured meshes, the classical WENO reconstruction, in which the high-order accuracy is achieved by non-linear combination of lower order polynomials, becomes extremely complicated for three-dimensional problems [14,46]. In this paper, the idea of simple WENO reconstruction is adopted [49,50]. The high-order accuracy is achieved by non-linear combination of high-order and lower-order polynomials, and the constant linear weights are adopted.

3.1. Selection of stencils

For the cell Ω_i, the faces are labeled as F_p, where $p = 1, \dots, 4$ for tetrahedral cell, and $p = 1, \dots, 6$ for hexahedral cell. The neighboring cell of Ω_i, which shares the face F_p, is denoted as Ω_{i_p}. Meanwhile, the neighboring cells of Ω_{i_p} are denoted as $\Omega_{i_{pm}}$. To achieve the third-order accuracy, a big stencil for non-compact reconstruction for cell Ω_i is selected as follows

$$S_i^{WENO} = \{\Omega_i, \Omega_{i_p}, \Omega_{i_{pm}}\},$$

which is consisted of the neighboring cells and neighboring cells of neighboring cells of Ω_i. Meanwhile, a big stencil for compact reconstruction for cell Ω_i is selected as follows

$$S_i^{HWENO} = \{\Omega_i, \Omega_{i_p}\},$$

which is only consist of the neighboring cells of Ω_i.

To deal with the discontinuity, the sub-stencils $S_{i_m}^{WENO}$ in non-compact WENO reconstruction for cell Ω_i are selected, where $m = 1, \dots, M$ and M is the number of sub-stencils. For the hexahedral cell, $M = 8$ and the sub-candidate stencils are selected as

$$\begin{aligned}
&S_{i_1}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_2}, \Omega_{i_3}\}, \ S_{i_5}^{WENO} = \{\Omega_i, \Omega_{i_6}, \Omega_{i_2}, \Omega_{i_3}\},\\
&S_{i_2}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_3}, \Omega_{i_4}\}, \ S_{i_6}^{WENO} = \{\Omega_i, \Omega_{i_6}, \Omega_{i_3}, \Omega_{i_4}\},\\
&S_{i_3}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_4}, \Omega_{i_5}\}, \ S_{i_7}^{WENO} = \{\Omega_i, \Omega_{i_6}, \Omega_{i_4}, \Omega_{i_5}\},\\
&S_{i_4}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_5}, \Omega_{i_2}\}, \ S_{i_8}^{WENO} = \{\Omega_i, \Omega_{i_6}, \Omega_{i_5}, \Omega_{i_2}\}.
\end{aligned}$$

The linear polynomials can be determined based on above stencils, which contain the target cell Ω_i and three neighboring cells. For the tetrahedral cells, in order to avoid the centroids of Ω_i and three of neighboring cells becoming coplanar, additional cells are needed for the sub-candidate stencils. For the tetrahedral cell, four sub-candidate stencils are selected as

$$\begin{aligned}
&S_{i_1}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_2}, \Omega_{i_3}, \Omega_{i_{11}}, \Omega_{i_{12}}, \Omega_{i_{13}}\},\\
&S_{i_2}^{WENO} = \{\Omega_i, \Omega_{i_1}, \Omega_{i_2}, \Omega_{i_4}, \Omega_{i_{21}}, \Omega_{i_{22}}, \Omega_{i_{23}}\},\\
&S_{i_3}^{WENO} = \{\Omega_i, \Omega_{i_2}, \Omega_{i_3}, \Omega_{i_4}, \Omega_{i_{31}}, \Omega_{i_{32}}, \Omega_{i_{33}}\},\\
&S_{i_4}^{WENO} = \{\Omega_i, \Omega_{i_3}, \Omega_{i_1}, \Omega_{i_4}, \Omega_{i_{41}}, \Omega_{i_{42}}, \Omega_{i_{43}}\},
\end{aligned}$$

where $\Omega_{i_{pn}} \neq \Omega_i, p = 1, \dots, 4$ and $n = 1, 2, 3$. The cells of sub-candidate stencils are consist of the three neighboring cells and three neighboring cells of one neighboring cell. With such an enlarged sub-stencils, the linear polynomials can be determined.

Meanwhile, the sub-stencils $S_{i_m}^{HWENO}$ in compact HWENO reconstruction for cell Ω_i can be selected more simply. The sub-candidate stencils are selected as

$$S_{i_m}^{HWENO} = \{\Omega_i, \Omega_{i_f}\},$$

where Ω_{i_f} is one of the neighboring cells of the target cell Ω_i, $m = 1, \dots, M$ and M equal to the number of the faces of cell Ω_i. The linear polynomials can be determined on such small stencils with additional degree of freedoms. Noticed that the sub-stencils $S_{i_m}^{WENO}$ for hexahedral cells also belong to compact sub-candidate stencils.

3.2. Non-compact WENO reconstruction

For the target cell Ω_i, the big stencil S_i^{WENO} is rearranged as $\{\Omega_0, \Omega_1, ..., \Omega_K\}$ and the sub-stencil $S_{i_m}^{WENO}$ is rearranged as $\{\Omega_0, \Omega_{m_1}, \ldots, \Omega_{m_K}\}$, where Ω_0 is the target cell. A quadratic polynomial and several linear polynomials can be constructed based on the big stencil S_i^{WENO} and the sub-stencils $S_{i_m}^{WENO}$ as follows

$$\begin{aligned} P_0(\boldsymbol{x}) &= Q_0 + \sum_{|\boldsymbol{d}|=1}^{2} a_{\boldsymbol{d}} p_{\boldsymbol{d}}(\boldsymbol{x}), \\ P_m(\boldsymbol{x}) &= Q_0 + \sum_{|\boldsymbol{d}|=1} b_{\boldsymbol{d}}^m p_{\boldsymbol{d}}(\boldsymbol{x}), \end{aligned} \tag{5}$$

where $m = 1, \ldots, M$, Q_0 is the cell averaged variables over Ω_0 with newly rearranged index, the multi-index $\boldsymbol{d} = (d_1, d_2, d_3)$ and $|\boldsymbol{d}| = d_1 + d_2 + d_3$. The base function $p_{\boldsymbol{d}}(\boldsymbol{x})$ is defined as

$$p_{\boldsymbol{d}}(\boldsymbol{x}) = x^{d_1} y^{d_2} z^{d_3} - \frac{1}{|\Omega_0|} \iiint_{\Omega_0} x^{d_1} y^{d_2} z^{d_3} \mathrm{d}V.$$

To determine these polynomials, the following constrains need to be satisfied

$$\begin{aligned} &\frac{1}{|\Omega_k|} \iiint_{\Omega_k} P_0(\boldsymbol{x}) \mathrm{d}V = Q_k, \ \Omega_k \in S_i^{WENO}, \\ &\frac{1}{|\Omega_{m_k}|} \iiint_{\Omega_{m_k}} P_m(\boldsymbol{x}) \mathrm{d}V = Q_{m_k}, \ \Omega_{m_k} \in S_{i_m}^{WENO}, \end{aligned}$$

where $k = 0, \ldots, K$, $m_k = 0, \ldots, m_K$, Q_k and Q_{m_k} are the conservative variable with newly rearranged index. The over-determined linear systems can be generated and the least square method is used to obtain the coefficients $a_{\boldsymbol{d}}$ and $b_{\boldsymbol{d}}^m$.

3.3. Compact HWENO reconstruction

For the target cell Ω_i, the big stencil S_i^{HWENO} is rearranged as $\{\Omega_0, \Omega_1, ..., \Omega_K\}$ and the sub-stencil $S_{i_m}^{HWENO}$ is rearranged as $\{\Omega_0, \Omega_{m_1}\}$, where Ω_0 is the target cell. The quadratic polynomial and linear polynomials in Eq. (5) can be also reconstructed based on the compact big stencil S_i^{HWENO} and sub-stencils $S_{i_m}^{HWENO}$ respectively. To determine these polynomials with a smaller stencil, the additional constrains need to be added for all cells as follows

$$\begin{aligned} &\frac{1}{|\Omega_k|} \iiint_{\Omega_k} P_0(\boldsymbol{x}) \mathrm{d}V = Q_k, \ \Omega_k \in S_i^{HWENO}, \\ &\frac{1}{|\Omega_k|} \iiint_{\Omega_k} \frac{\partial}{\partial \boldsymbol{\tau}} P_0(\boldsymbol{x}) \mathrm{d}V = (Q_{\boldsymbol{\tau}})_k, \ \Omega_k \in S_i^{HWENO}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} &\frac{1}{|\Omega_{m_k}|} \iiint_{\Omega_{m_k}} P_m(\boldsymbol{x}) \mathrm{d}V = Q_{m_k}, \ \Omega_{m_k} \in S_{i_m}^{HWENO}, \\ &\frac{1}{|\Omega_{m_k}|} \iiint_{\Omega_{m_k}} \frac{\partial}{\partial \boldsymbol{\tau}} P_m(z) \mathrm{d}V = (Q_{\boldsymbol{\tau}})_{m_k}, \ \Omega_{m_k} \in S_{i_m}^{HWENO}, \end{aligned} \tag{7}$$

where $\boldsymbol{\tau}$ is the unit directions of local coordinate, i.e., $\boldsymbol{n}_x, \boldsymbol{n}_y, \boldsymbol{n}_z$, $k = 0, \ldots, K$, $m_k = 0, m_1$, Q_k, Q_k and Q_{m_k} are the cell averaged conservative variables and $(Q_{\boldsymbol{n}_x})_k$, $(Q_{\boldsymbol{n}_y})_k$, $(Q_{\boldsymbol{n}_z})_k$ and $(Q_{\boldsymbol{n}_x})_{m_k}$, $(Q_{\boldsymbol{n}_y})_{m_k}$, $(Q_{\boldsymbol{n}_z})_{m_k}$ are the cell averaged directional derivatives over $\Omega_k \in S_i^{HWENO}$ and $\Omega_{m_k} \in S_{i_m}^{HWENO}$ respectively with newly rearranged index.

In order to solve the systems in Eq. (6) and Eq. (7), the cell averaged directional derivatives on the right-hand side need to be calculated. Different from the traditional Riemann solvers, the gas-kinetic scheme provides a time-dependent gas distribution function by Eq. (4). Meanwhile, the macroscopic conservative variables can be obtained by taking moments of the distribution function as well

$$Q(\boldsymbol{x}_G, t) = \int \boldsymbol{\psi} f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi) \mathrm{d}\Xi. \tag{8}$$

According to the Gauss's theorem, the cell averaged gradient of the flow variable Q can be calculated as follows

$$
\begin{aligned}
|\Omega_k|(\nabla Q)_k(t) &= \iiint_{\Omega_k} \nabla Q(t)\mathrm{d}V = \iint_{\partial\Omega_k} Q(t)\boldsymbol{\tau}\mathrm{d}S \\
&= \sum_{i_p\in N(k)} \Big(\sum_G \omega_G \Big(\int \boldsymbol{\psi} f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi)\mathrm{d}\Xi \Big)\boldsymbol{\tau} S_{i_p} \Big),
\end{aligned} \tag{9}
$$

where Ω_k is arbitrary cell in computational mesh, $(\nabla Q)_k$ is the cell averaged gradient of the flow variable Q_k over cell Ω_k, ∇Q is the distribution of flow gradient, $\boldsymbol{\tau} = (\boldsymbol{n_x}, \boldsymbol{n_y}, \boldsymbol{n_z})$ are the unit directions of local coordinate on the cell interface. With Gaussian quadrature rule, the cell-averaged gradient $(\nabla Q)_k$ at $t = t^n$ can be obtained. It means that the computation of cell-averaged gradient $(\nabla Q)_k$ can be turned to calculate the point value of Q of cell interface in a local orthogonal coordinate. In practice, the derivative parts in Eq. (6) and Eq. (7) are scaled by $h = |\Omega_k|^{1/3}$ and $|\Omega_{m_k}|^{1/3}$ for a smaller condition number of the matrix in the linear system of $a_{\boldsymbol{d}}$ and $b_{\boldsymbol{d}}$. If the coefficients $a_{\boldsymbol{d}}$ and $b_{\boldsymbol{d}}$ are solved by the least square method, the linear instability will be introduced for the compact reconstruction. In order to overcome this drawback, the constrained least-square method is used for solving the above linear systems, where the conservative variable equations are set as strictly satisfied and others are satisfied in the sense of least square [19].

3.4. Non-linear combination

With the reconstructed polynomial $P_m(\boldsymbol{x}), m = 0, ..., M$, the point-value $Q(\boldsymbol{x}_G)$ and the spatial derivatives $\partial_{x,y,z} Q(\boldsymbol{x}_G)$ for reconstructed variables at Gaussian quadrature point can be given by the non-linear combination

$$
\begin{aligned}
Q(\boldsymbol{x}_G) &= \overline{\omega}_0 \Big(\frac{1}{\gamma_0} P_0(\boldsymbol{x}_G) - \sum_{m=1}^{M} \frac{\gamma_m}{\gamma_0} P_m(\boldsymbol{x}_G) \Big) + \sum_{m=1}^{M} \overline{\omega}_m P_m(\boldsymbol{x}_G), \\
\partial_{x,y,z} Q(\boldsymbol{x}_G) &= \overline{\omega}_0 \Big(\frac{1}{\gamma_0} \partial_{x,y,z} P_0(\boldsymbol{x}_G) - \sum_{m=1}^{M} \frac{\gamma_m}{\gamma_0} \partial_{x,y,z} P_m(\boldsymbol{x}_G) \Big) + \sum_{m=1}^{M} \overline{\omega}_m \partial_{x,y,z} P_m(\boldsymbol{x}_G),
\end{aligned} \tag{10}
$$

where $\gamma_0, \gamma_1, \ldots, \gamma_M$ are the linear weights. The non-linear weights ω_m and normalized non-linear weights $\overline{\omega}_m$ are defined as

$$
\overline{\omega}_m = \frac{\omega_m}{\sum_{m=0}^{M} \omega_m}, \ \omega_m = \gamma_m \Big[1 + \Big(\frac{\tau_Z}{\beta_m + \epsilon} \Big) \Big], \ \tau_Z = \sum_{m=1}^{M} \Big(\frac{|\beta_0 - \beta_m|}{M} \Big),
$$

where ϵ is a small positive number. The smooth indicator β_m is defined as

$$
\beta_m = \sum_{|l|=1}^{r_m} |\Omega_i|^{\frac{2|l|}{3}-1} \int_{\Omega_i} \Big(\frac{\partial^l P_m}{\partial_x^{l_1} \partial_y^{l_2} \partial_z^{l_3}}(x, y, z) \Big)^2 \mathrm{d}V,
$$

where $r_0 = 2$ and $r_m = 1$ for $m = 1, \ldots, M$. It can be proved that Eq. (10) ensures third-order accuracy and more details can be found in [41,17]. In the computation, the linear weights are set as $\gamma_i = 0.025$, $\gamma_0 = 1 - \gamma_i M$ for both non-compact and compact scheme without special statement.

4. Implicit method

4.1. Implicit method for non-compact scheme

The backward Euler method for the semi discretized scheme Eq. (2) at t^{n+1} is given by

$$
\frac{|\Omega_i|}{\Delta t} \Delta Q_i^n = \mathcal{L}(Q_i^{n+1}), \tag{11}
$$

where $\Delta Q_i^n = Q_i^{n+1} - Q_i^n$, Δt is the time step and $\mathcal{L}(Q_i^{n+1})$ can be linearized as

$$
\mathcal{L}(Q_i^{n+1}) \approx \mathcal{L}(Q_i^n) + \Big(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}Q} \Big) \Delta Q_i^n,
$$

where $\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}Q}$ is the Jacobian matrix. Eq. (11) can be rewritten as

$$
\Big(\frac{|\Omega_i|}{\Delta t} I - \Big(\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}Q} \Big) \Big) \Delta Q_i^n = \mathcal{L}(Q_i^n), \tag{12}
$$

where I is an identity matrix. In order to preserve the temporal evolution advantage brought by the GKS solver, the residual term at time step $t = t^n$ for both non-compact and compact schemes is given by taking integration of Eq. (3) over the time interval

$$
\mathcal{L}(Q_i^n) = -\sum_{i_p\in N(i)} \Big(\sum_G \omega_G \Big(\frac{1}{\Delta t} \int_0^{\Delta t} \int \boldsymbol{\psi} \boldsymbol{u} f(\boldsymbol{x}_G, t, \boldsymbol{u}, \xi) \boldsymbol{n}_G \mathrm{d}\Xi \mathrm{d}t \Big) S_{i_p} \Big).
$$

In the previous study, LUSGS method was implemented for the high-order GKS on the unstructured meshes [41,42]. For the steady state problem, LUSGS method converges more efficiently than the explicit methods. However, it still requires a great number of time steps to achieve a satisfactory final residual. In order to speed up the convergence, the Newton-GMRES method [29,4] will be implemented for the high-order GKS on the three-dimensional unstructured meshes. The linearized equation Eq. (12) can be simply written into the following form

$$\boldsymbol{A}\Delta Q^n = \boldsymbol{R}^n.$$

The purpose of the GMRES method is finding an approximate solution over a finite dimensional orthogonal space

$$\Delta Q^n \approx \Delta Q^{(m)} = \Delta Q^{(0)} + \boldsymbol{u}_m,$$

such that

$$\|\boldsymbol{r}^{(m)}\|_2 = \min_{\boldsymbol{u}_m \in K_m} \|\boldsymbol{R}^n - \boldsymbol{A}(\Delta Q^{(0)} + \boldsymbol{u}_m)\|_2 = \min_{\boldsymbol{u}_m \in K_m} \|\boldsymbol{r}^{(0)} - \boldsymbol{A}\boldsymbol{u}_m\|_2.$$

With the initial condition Q^n and arbitrary initial solution $\Delta Q^{(0)}$, which is chosen as $\Delta Q^{(0)} = 0$, the initial residual is given by

$$\boldsymbol{r}^{(0)} = \boldsymbol{R}^n - \boldsymbol{A}\Delta Q^{(0)}.$$

The Krylov subspace of $\boldsymbol{r}^{(0)}$, which is used for searching the approximation solution, can be generated by

$$K_m \equiv \text{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_m\} = \text{span}\{\boldsymbol{r}^{(0)}, \boldsymbol{A}\boldsymbol{r}^{(0)}, \dots, \boldsymbol{A}^{m-1}\boldsymbol{r}^{(0)}\}.$$

To save the memory storage, the matrix-free GMRES method is widely employed [21,22]. In order to avoid calculating the Jabobian matrix in $\boldsymbol{A}$ directly, the F-derivative is used as follows

$$(\frac{\partial \mathcal{L}}{\partial Q})\Delta Q^n \approx \frac{\mathcal{L}(Q^n + \sigma\Delta Q^n) - \mathcal{L}(Q^n)}{\sigma}, \tag{13}$$

where the subscript is omitted and σ is a small scalar to be chosen carefully. Although the above matrix-free approximation reduces the memory storage requirement, the reconstruction processes combined with GKS solver involved in Eq. (13) will cost too much time, which will reduce the computational efficiency greatly. Taking the balance of computational efficiency and memory storage into account, the GMRES method with an approximation matrix is employed in this paper. The system Eq. (11) can be written into the flux form

$$\frac{|\Omega_i|}{\Delta t}\Delta Q_i^n + \sum_{i_p \in N(i)} \left(F_{i,i_p}^{n+1} - F_{i,i_p}^n\right) S_{i_p} = -\sum_{i_p \in N(i)} F_{i,i_p}^n S_{i_p} = \mathcal{L}(Q_i^n). \tag{14}$$

According to the total differential formulation,

$$\begin{aligned} F_{i,i_p}^{n+1} - F_{i,i_p}^n &= F(Q_i^n + \Delta Q_i^n, Q_{i_p}^n + \Delta Q_{i_p}^n) - F(Q_i^n, Q_{i_p}^n) \\ &= (\frac{\partial F}{\partial Q_i})^n \Delta Q_i^n + (\frac{\partial F}{\partial Q_{i_p}})^n \Delta Q_{i_p}^n. \end{aligned}$$

Substituting the term above into Eq. (14), the backward Euler method can be rewritten as

$$\left(\frac{|\Omega_i|}{\Delta t} I + \sum_{i_p \in N(i)} (\frac{\partial F}{\partial Q_i})^n S_{i_p}\right)\Delta Q_i^n + \sum_{i_p \in N(i)} \left((\frac{\partial F}{\partial Q_{i_p}})^n S_{i_p}\right)\Delta Q_{i_p}^n = \mathcal{L}(Q_i^n). \tag{15}$$

According to the following approximation

$$\begin{cases} \dfrac{\partial F}{\partial Q_i} = \dfrac{1}{2}(J(Q_i) + |\lambda_{i,i_p}|I), \\ \dfrac{\partial F}{\partial Q_{i_p}} = \dfrac{1}{2}(J(Q_{i_p}) - |\lambda_{i,i_p}|I), \end{cases} \tag{16}$$

where

$$|\lambda_{i,i_p}| \geq |\boldsymbol{u}_{i,i_p} \cdot \boldsymbol{n}_{i,i_p}| + a_{i,i_p},$$

$\boldsymbol{u}_{i,i_p}$ and a_{i,i_p} are the velocity and the speed of sound as at the interface S_{i_p}, $\boldsymbol{n}_{i,i_p}$ is the unit normal direction on the cell interface S_{i_p}, and $J(Q)$ represents the Jacobian of the inviscid flux. Substituting Eq. (16) into Eq. (15), the coefficient matrix $\boldsymbol{A}$ of Eq. (15) can be determined. In order to avoid storing the whole matrix $\boldsymbol{A}$ directly and be able to calculate the matrix multiplication in parallel, a suitable strategy for the storage of matrix $\boldsymbol{A}$ and its parallel computation is designed, and the details will be shown in following section.

4.2. Implicit method for compact scheme

In this section, the GMRES method is combined with third-order compact HWENO reconstruction for solving the steady state problems. In practice, the coefficient matrix $\boldsymbol{A}$ of Eq. (15) can be fixed with the known cell averaged conservative variable Q_i^n at each time step, which has nothing to do with the cell averaged derivatives. Meanwhile, the right side $\mathcal{L}(Q_i^n)$ is calculated by the compact HWENO reconstruction with the known cell averaged conservative variable Q_i^n and cell averaged directional derivatives $(Q_{\boldsymbol{n}_x})_i^n$, $(Q_{\boldsymbol{n}_y})_i^n$, $(Q_{\boldsymbol{n}_z})_i^n$. After solving the system Eq. (15) at each time step, the cell averaged conservative variable Q_i^{n+1} and cell averaged derivatives $(Q_{\boldsymbol{n}_x})_i^{n+1}$, $(Q_{\boldsymbol{n}_y})_i^{n+1}$, $(Q_{\boldsymbol{n}_z})_i^{n+1}$ should be updated simultaneously. So that $\mathcal{L}(Q_i^{n+1})$ can be obtained as the right side of Eq. (15) in the next time step.

For the implicit compact scheme, the GMRES method need to associate with a suitable strategy for updating cell averaged directional derivatives at each time step. Due to the small change of flow variables in the steady state problems, the update of cell averaged directional derivatives can be driven by time evolution directly. Similar technique has been used in other equations [28]. According to Eq. (8) and Eq. (9), the cell averaged directional derivatives at $t = t^{n+1}$ can be updated directly by

$$(Q_{\boldsymbol{\tau}})_i^{n+1} = (Q_{\boldsymbol{\tau}})_i^n = \frac{1}{|\Omega_i|} \sum_{i_p \in N(i)} \Big(\sum_G \omega_G \Big(\int \psi f(\boldsymbol{x}_G, t^n, \boldsymbol{u}, \xi) \boldsymbol{\tau} \mathrm{d}\Xi \Big) S_{i_p} \Big),$$

where $\boldsymbol{\tau} = \boldsymbol{n}_x, \boldsymbol{n}_y, \boldsymbol{n}_z$.

The numerical tests show that the above update strategy is suitable for steady flows, and the high-order spatial accuracy can be maintained in the computation. In the future, the optimization of above implicit update strategy will be studied for unsteady flows, for example, designing nonlinear weights in time direction for preserving temporal and spatial accuracy.

Algorithm 1 shows the whole process of preconditioned GMRES method combined with WENO and HWENO reconstruction, where the red lines are special steps of the non-compact scheme, the blue lines are special steps of the compact scheme and the black lines are common steps for both two schemes.

To ensure the convergence rate of the GMRES method, the condition number of the corresponding matrix $\boldsymbol{A}$ has to be as small as possible and the initial solution of flow evolution is preferably within the convergence domain of Newton iteration. In the computation, the GMRES method combined with preconditioning technique is considered for non-compact and compact schemes. It means that instead of solving the system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b},$$

an equivalent preconditioned linear system [21] is considered

$$\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{P}^{-1}\boldsymbol{b}, \tag{17}$$

where $\boldsymbol{P}$ is the preconditioning matrix which is an approximation of matrix $\boldsymbol{A}$. Usually, the LUSGS method is adopted as preconditioner of GMRES method, but the serial sweep part in LUSGS method is not easy to be implemented for parallel computation. In this paper, Jacobi iteration is adopted as a preconditioner, which is easy to be implemented in parallel. According to Eq. (14), the matrix $\boldsymbol{A}$ of Eq. (14) can be written as

$$\boldsymbol{A} = \boldsymbol{D} + \boldsymbol{L} + \boldsymbol{U},$$

where $\boldsymbol{D}$ represents the diagonal area, $\boldsymbol{L}$ represents the lower triangular area and $\boldsymbol{U}$ represents the upper triangular area of matrix $\boldsymbol{A}$. The Jacobi iteration is used as preconditioner to provide $\boldsymbol{P}^{-1}\boldsymbol{b}$ as follows

$$\boldsymbol{b}^0 = \boldsymbol{D}^{-1}\boldsymbol{b}^{\mathrm{ini}},$$
$$\boldsymbol{b}^k = \boldsymbol{D}^{-1}(\boldsymbol{b}^{\mathrm{ini}} - (\boldsymbol{L} + \boldsymbol{U})\boldsymbol{b}^{k-1}),$$
$$\boldsymbol{P}^{-1}\boldsymbol{b} = \boldsymbol{b}^{k_{\max}},$$

where $1 \le k \le k_{\max}$ and $\boldsymbol{b}^{\mathrm{ini}}$ represents $\boldsymbol{r}^0$ and $\boldsymbol{A}\boldsymbol{v}_j$ in Line 10 and Line 15 of Algorithm 1. With the process above, $\boldsymbol{P}^{-1}\boldsymbol{r}^0$ and $\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{v}_j$ can be fully given.

4.3. Parallel computation for implicit method

In the previous work, the computation of HGKS is mainly based on the central processing unit (CPU) code. To improve the efficiency, the OpenMP directives and message passing interface (MPI) are used for parallel computation [5]. However, the CPU computation is usually limited in the number of threads which are handled in parallel. Graphics processing unit (GPU) is a form of hardware acceleration, which is originally developed for graphics manipulation and execute highly-parallel computing tasks. Recently, GPU has applied to HGKS scheme for large-scale scientific computation [36]. In this work, the implicit HGKS scheme with WENO and HWENO reconstruction on unstructured meshes is implemented with both CPU using OpenMP and GPU using compute unified device architecture (CUDA).

The matrix $\boldsymbol{A}$ in Eq. (15) is a large asymmetric sparse block matrix of $(5N, 5N)$ dimension, where N is the total number of computational cells. The amount of memory storage would be unbearable if the whole matrix were to be stored. In this section, a

Algorithm 1: Program for preconditioned GMRES method.

Initial condition Q^n for non-compact scheme;
Initial condition Q^n, $(Q_{n_x})^n$, $(Q_{n_y})^n$, $(Q_{n_z})^n$ for compact scheme;
while *residual* $\leq$ *tolerance* **do**
  Calculation of time step;
  WENO reconstruction to calculate $\mathcal{L}(Q^n)$;
  HWENO reconstruction to calculate $\mathcal{L}(Q^n)$ and f^n;
  Calculation of $\boldsymbol{A}$;
  for $i = 1, restart\ times$ **do**
    Calculation of $\boldsymbol{r}^0$ and residual;
    Jacobi preconditioning: $\hat{\boldsymbol{r}}^0 = \boldsymbol{P}^{-1}\boldsymbol{r}^0$;
    Arnoldi process:
    $\boldsymbol{v}_1 = \hat{\boldsymbol{r}}^0 / ||\hat{\boldsymbol{r}}^0||_2$;
    for $j = 1, dimK_m$ **do**
      $\boldsymbol{y}_j = \boldsymbol{A}\boldsymbol{v}_j$;
      Jacobi preconditioning: $\boldsymbol{w}_j = \boldsymbol{P}^{-1}\boldsymbol{y}_j$;
      for $i = 1, j$ **do**
        $h_{ij} = (\boldsymbol{w}_j, \boldsymbol{v}_i)$;
        $\boldsymbol{w}_j = \boldsymbol{w}_j - h_{ij}\boldsymbol{v}_i$;
      end
      $h_{j+1,j} = ||\boldsymbol{w}_j||_2$;
      $\boldsymbol{v}_{j+1} = \boldsymbol{w}_j / h_{j+1,j}$;
    end
    Minimization process:
    QR decomposition and solve an upper triangular matrix;
    Get the solution vector y_m;
    $\Delta Q^n = \Delta Q^n + K_m y_m$;
  end
  Update:
  $Q^{n+1} = Q^n + \Delta Q^n$;
  Calculate $(Q_{n_x})^{n+1}$, $(Q_{n_y})^{n+1}$, $(Q_{n_z})^{n+1}$ by f^n for compact scheme;
end
Output of flow field and residual convergence history.

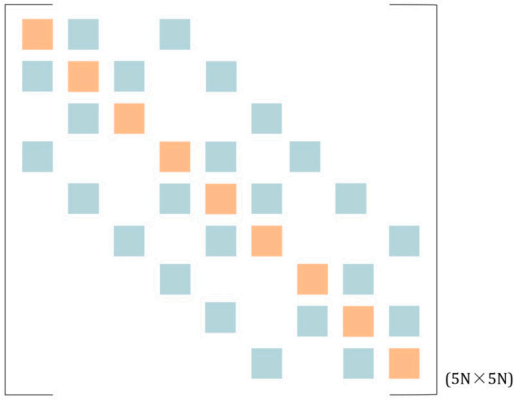

Fig. 1. The non-zero block distribution of $\boldsymbol{A}$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

simple strategy of storing the matrix $\boldsymbol{A}$ in Eq. (15) is introduced to balance the parallel efficiency and memory storage. It is noticed that the position of all non-zero blocks in each row (by block) in $\boldsymbol{A}$ only corresponds to its own cell and all neighboring cells. The non-zero block distribution of $\boldsymbol{A}$ is shown in Fig. 1, where the orange and blue blocks at the i-th row represent the non-zero blocks corresponding to cell Ω_i and its neighboring cells respectively. These blocks are stored in the structure array for the i-th cell. With the storage of these non-zero blocks, the computation is easy to be implemented in parallel with single GPU. For example, in the computation of $\boldsymbol{r} = \boldsymbol{R} - \boldsymbol{A}\Delta\boldsymbol{Q}$, the residual term $\boldsymbol{R}$, i.e., $\mathcal{L}(Q_i^n)$, can be calculated in parallel naturally, because WENO-type reconstruction and GKS flux solver have high parallelism. Meanwhile, denoting the i-th row of the matrix $\boldsymbol{A}$ as $\boldsymbol{A}_i = (\boldsymbol{a}_{ij})$, where $a_{ij} \neq 0$ with $j \in \{i\} \cup \{N(i)\}$. The vector multiplication of $\boldsymbol{A}_i$ and $\Delta\boldsymbol{Q}$ is equal to $\sum_{j\in\{i\}\cup\{N(i)\}} \boldsymbol{a}_{ij}\Delta\boldsymbol{Q}_j$. Such multiplication can be completed through cell structure array, and each row of the matrix term $\boldsymbol{A}\Delta\boldsymbol{Q}$ can be also calculated in parallel. In the future, the above strategy is easy to develop to multiple-GPU parallel because matrix partitioning is exactly the same as cell partitioning.

For the unstructured meshes, the data for cells, interfaces and nodes can be stored one-dimensionally. As shown in Fig. 2, the cells can be divided into D blocks, where N is the total number of cells. Meanwhile, the CUDA threads are organized into thread blocks, and thread blocks constitute a thread grid. The one-to-one correspondence can be set for computational cell and thread ID in parallel

Fig. 2. Connection between threads and unstructured cells.

computation. In practice, the whole Algorithm 1 can be implemented in parallel based on single GPU directly with the above storage strategy, except Line 23, 24 and 25, i.e., the minimization process in GMRES method. But the matrix multiplications involved in this minimization process part are all under 20 dimensions. It does not affect the computational efficiency in the large-scale computation even if these multiplications are completed serially on the host CPU.

5. Numerical tests

In this section, the numerical tests for both inviscid and viscous flows will be presented to validate the current scheme. For the inviscid flows, the collision time τ takes

$$\tau = \epsilon\Delta t + C|\frac{p_l - p_r}{p_l + p_r}|\Delta t,$$

where $\epsilon = 0.1$ and $C = 1$. For the viscous flows, we have

$$\tau = \frac{\mu}{p} + C|\frac{p_l - p_r}{p_l + p_r}|\Delta t,$$

where p_l and p_r denote the pressure on the left and right sides of the cell interface, μ is the dynamic viscous coefficient and p is the pressure at the cell interface at cell interface determined by Q_0. In smooth flow regions, it will reduce to

$$\tau = \frac{\mu}{p}$$

In the computation, the poly gas is used and the specific heat ratio takes $\gamma = 1.4$.

In order to eliminate the spurious oscillation and improve the stability, the non-compact WENO and compact HWENO reconstructions are performed for the characteristic variables. For a specific cell interface, denote Q^* and $Q^*_{\boldsymbol{\tau}}$ are the averaged values from both sides of cell interface in the local coordinate. R_* is the right eigenmatrix of Jacobian matrix $\left(\partial\mathbf{F}/\partial Q\right)$ at $Q = Q^*$, where $\mathbf{F}$ is the Euler flux. The cell averaged conservative variable and cell averaged directional derivatives can be projected into the characteristic field by $q^* = R_*^{-1}Q^*$ and $q^*_{\boldsymbol{\tau}} = R_*^{-1}Q^*_{\boldsymbol{\tau}}$. With the reconstructed values, the conservative variable and directional derivatives can be obtained by the inverse projection. In the following cases, the GMRES method with non-compact WENO reconstruction and the GMRES method with compact HWENO reconstruction are tested. As comparison, the numerical results of LUSGS method with non-compact WENO reconstruction are also provided. For simplicity, the LUSGS, GMRES method with non-compact WENO reconstruction and the GMRES method with compact HWENO reconstruction are denoted as LUSGS-WENO, GMRES-WENO and GMRES-HWENO respectively. In the computation, the dimension of Krylov subspace is 10, the restart times of GMRES method is 3 and the inner iteration times of Jacobi preconditioner is 2. The L_1 norm of the density residual is taken as test residual. The CFL number is set as 10 if there is no special statement.

5.1. Lid-driven cavity flow

The lid-driven cavity problem is one of the most important benchmarks for numerical Navier-Stokes solvers. The fluid is bounded by a unit cubic $[0,1]\times[0,1]\times[0,1]$ and driven by a uniform translation of the top boundary with $y = 1$. In this case, the flow is simulated with Mach number $Ma = 0.15$ and all the boundaries are isothermal and nonslip. Numerical simulations are conducted with the Reynolds numbers of $Re = 1000$ and 100. This case is performed by the tetrahedral mesh which contains 6×20^3 cells, where the mesh near the wall is refined and the size of the first layer cells is $h_{min} = 2.5\times10^{-2}$. The mesh distribution is shown in Fig. 3. The stencils $S_{i_m}^{HWENO}$ are selected as tetrahedral compact sub-stencils, which are dominated by the gradient informations.

The steady state U-velocity profiles along the vertical centerline and V-velocity profiles along the horizontal centerline and the benchmark data for $Re = 1000$ [2] and 100 [32] are shown in Fig. 4. The numerical results agree well with the benchmark data, especially the compact scheme. The histories of residual convergence with different implicit methods are shown in Fig. 5 for $Re = 1000$ and 100. It shows that the residual of GMRES method converges faster than the LUSGS method. Meanwhile, the residual of compact GMRES method is comparable with that of non-compact GMRES method.

5.2. Flows passing through a sphere

This case is used to test the capability in resolving from the low-speed to supersonic flows, and the initial condition is given as a free stream condition

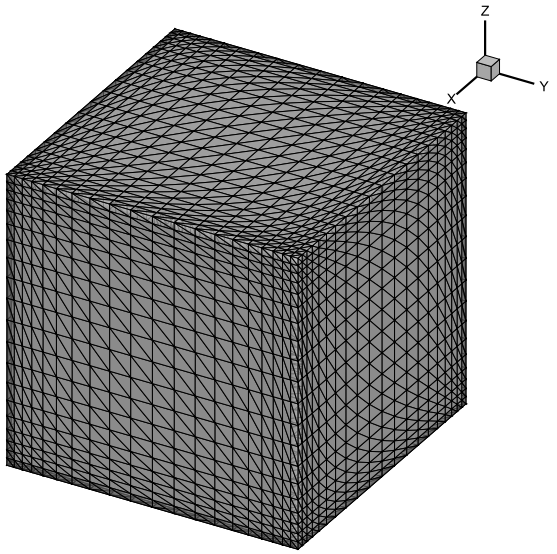

Fig. 3. Lid-driven cavity flow: the local computational mesh distribution with tetrahedral meshes.

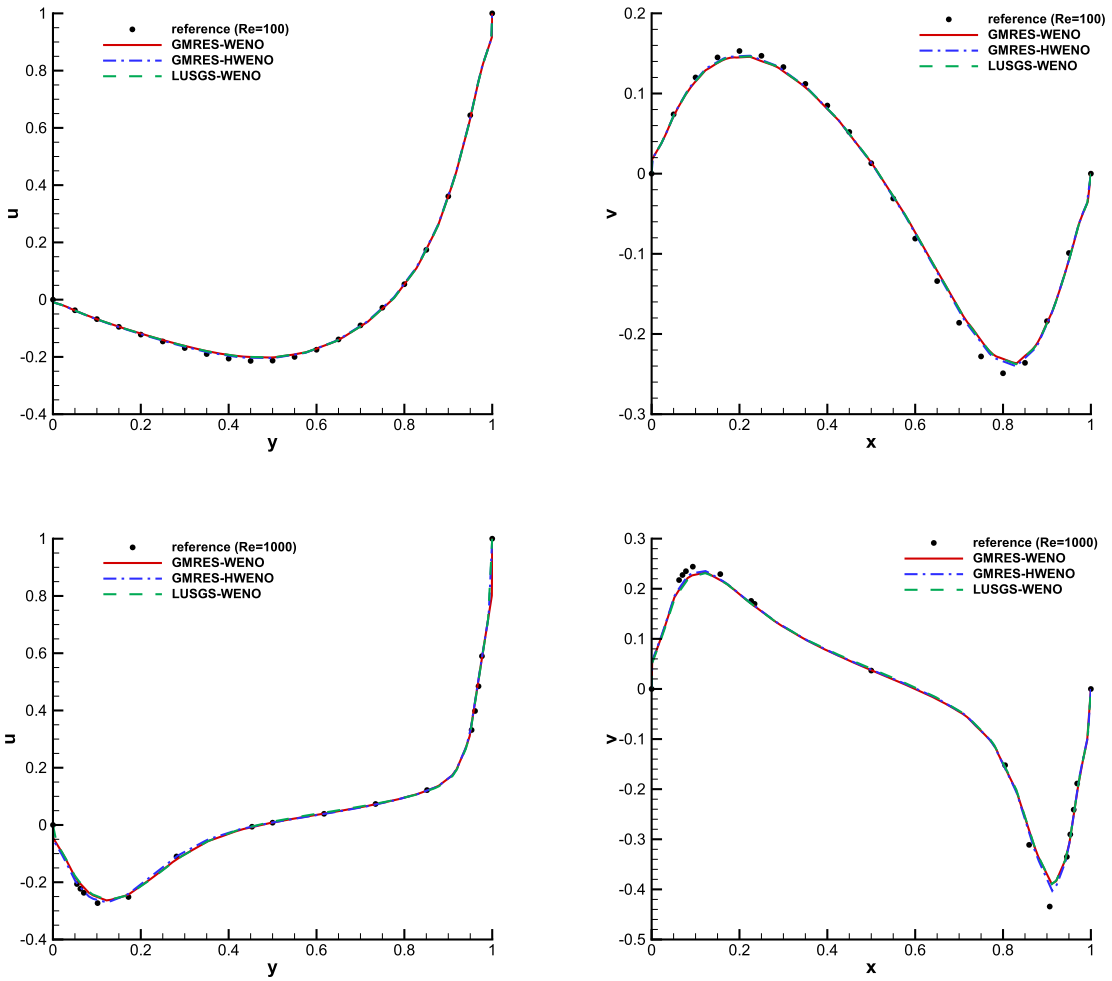

Fig. 4. Lid-driven cavity flow: the steady state U-velocity profiles along the vertical centerline (left), V-velocity profiles along the horizontal centerline (right) for $Re = 1000$ and 100.

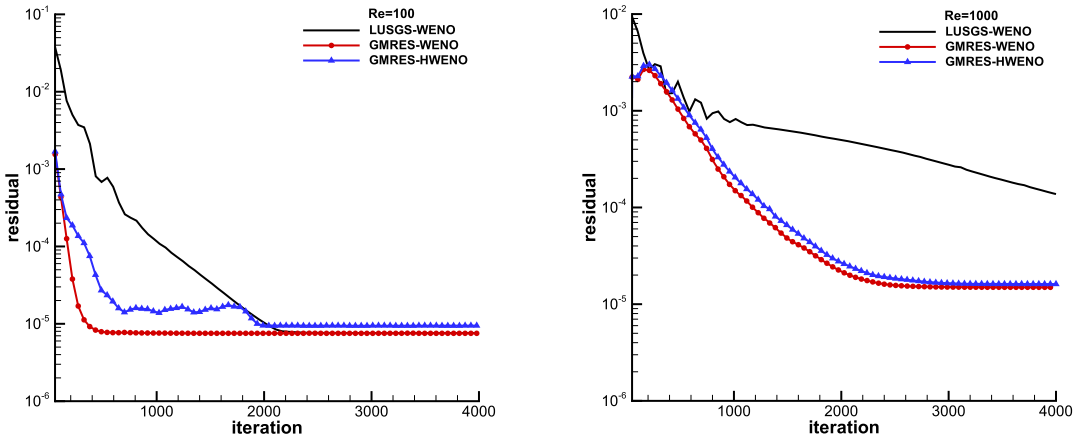

Fig. 5. Lid-driven cavity flow: the residual comparison with the LUSGS method, the non-compact GMRES method and compact GMRES method for $Re = 100$ and 1000.

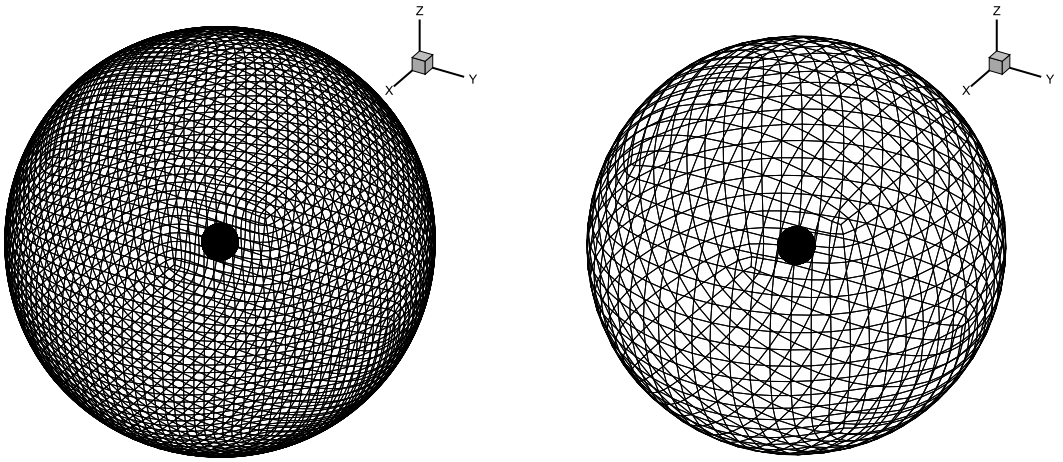

Fig. 6. Flows passing through a sphere: the local computational mesh distribution with hexahedral meshes for viscous flow (left) and inviscid flow (right).

Table 1
Subsonic inviscid flow passing through a sphere: quantitative comparisons of closed wake length L and separation angle θ for $Re = 118$ and $Ma_\infty = 0.2535$.

Scheme	Computational Mesh	L	θ
LUSGS GKS	190464 cells, Hex	0.91	124.5
noncompact GMRES GKS	190464 cells, Hex	0.91	124.5
compact GMRES GKS	190464 cells, Hex	0.95	128.1
4th-order DDG GMRES [9]	1608680 cells, Hybrid	0.96	123.7
Experiment [34]	-	1.07	151

$$(\rho, U, V, W, p)_\infty = (1, Ma_\infty, 0, 0, 1/\gamma),$$

where Ma_∞ is the Mach number of the free stream. As shown in Fig. 6, this case is performed by two unstructured hexahedral meshes. The viscous flows are tested on the hexahedral mesh which contains 190464 cells, where the size of the first layer cells on the surface of sphere is $h_{min} = 3 \times 10^{-2}$. The inviscid flows are tested on the hexahedral mesh which contains 50688 cells and $h_{min} = 2 \times 10^{-2}$. The supersonic or subsonic inlet and outlet boundary conditions are given according to far field normal velocity, the slip adiabatic boundary condition is used for inviscid flows and the non-slip adiabatic boundary condition is imposed for viscous flows on the surface of sphere. The stencils $S_{i_m}^{WENO}$ are selected as hexahedral compact sub-stencils, which are dominated by the conservative variables informations.

To validate the linear stability of the implicit methods, the subsonic case with $Re = 118$ and $Ma_\infty = 0.2535$ is provided, and the linear weights are used in both WENO and HWENO reconstruction. In this case, the CFL number is taken as 20. The density, velocity and streamline distributions at vertical centerline planes are shown in Fig. 7. The quantitative results of separation angle θ and closed wake length L are given in Table 1. The compact GMRES method with linear weights gives the better values of L and θ,

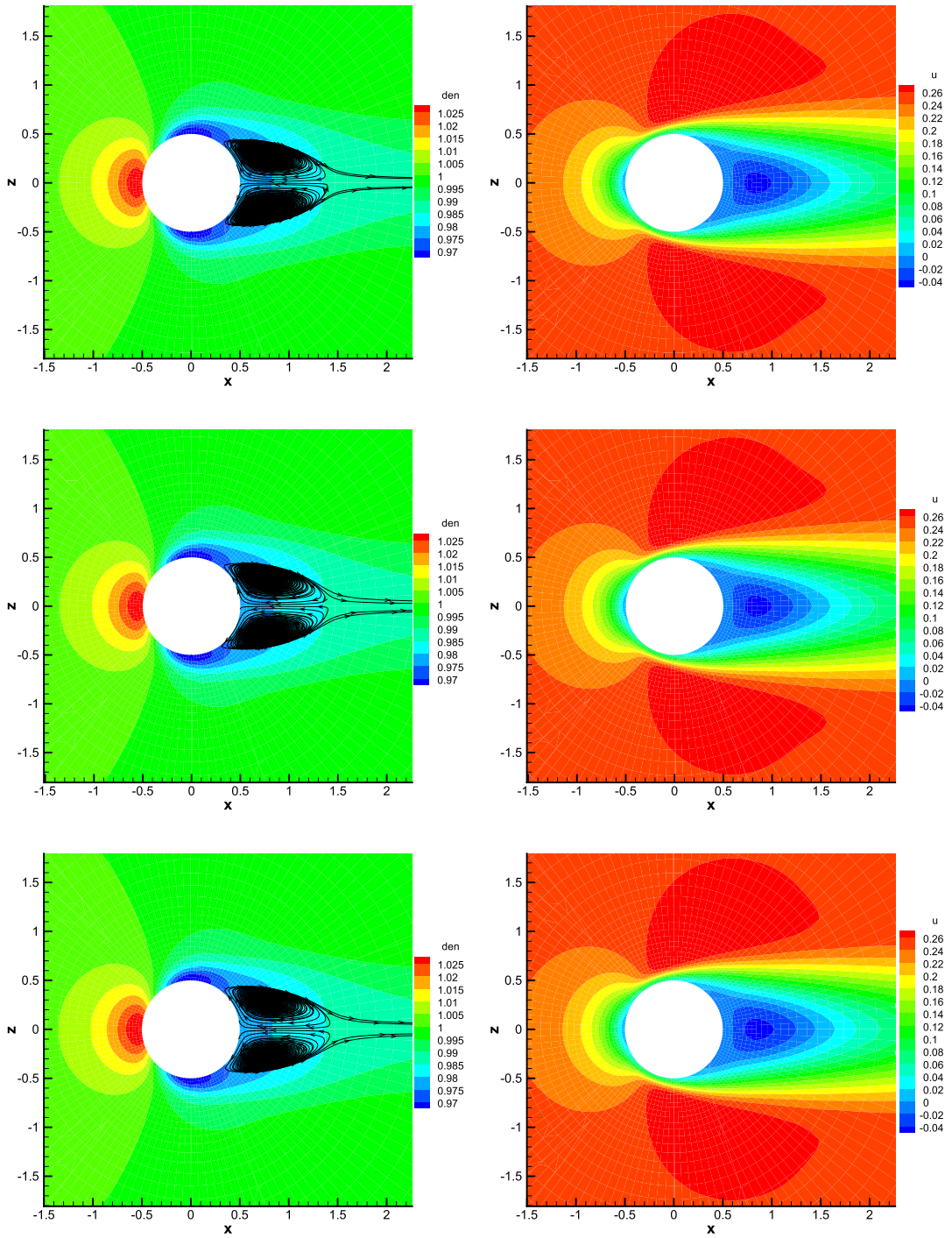

Fig. 7. Subsonic viscous flow passing through a sphere: the density and streamline distributions (left) and velocity distribution (right) at vertical centerline planes with the LUSGS method (top), the non-compact GMRES method (middle) and the compact GMRES method for $Re = 118$ and $Ma_\infty = 0.2535$.

which are closer to the experiment data. It means that the capability of implicit compact scheme to capture the structure of viscous flow is more accurate. To verify the robustness of the implicit methods, the supersonic viscous flow with $Re = 300$ and $Ma_\infty = 1.5$ is tested as well. The non-linear weights are used in both WENO and HWENO reconstruction. In this case, the CFL number is taken as 5. The density, velocity and streamline distributions at vertical centerline planes are shown in Fig. 8. The quantitative results of closed wake length L and separation angle θ are given in Table 2. The current implicit compact scheme gives a bigger value of θ. The performance of closed wake length L would be better with a refined mesh. The residual convergence history with different implicit

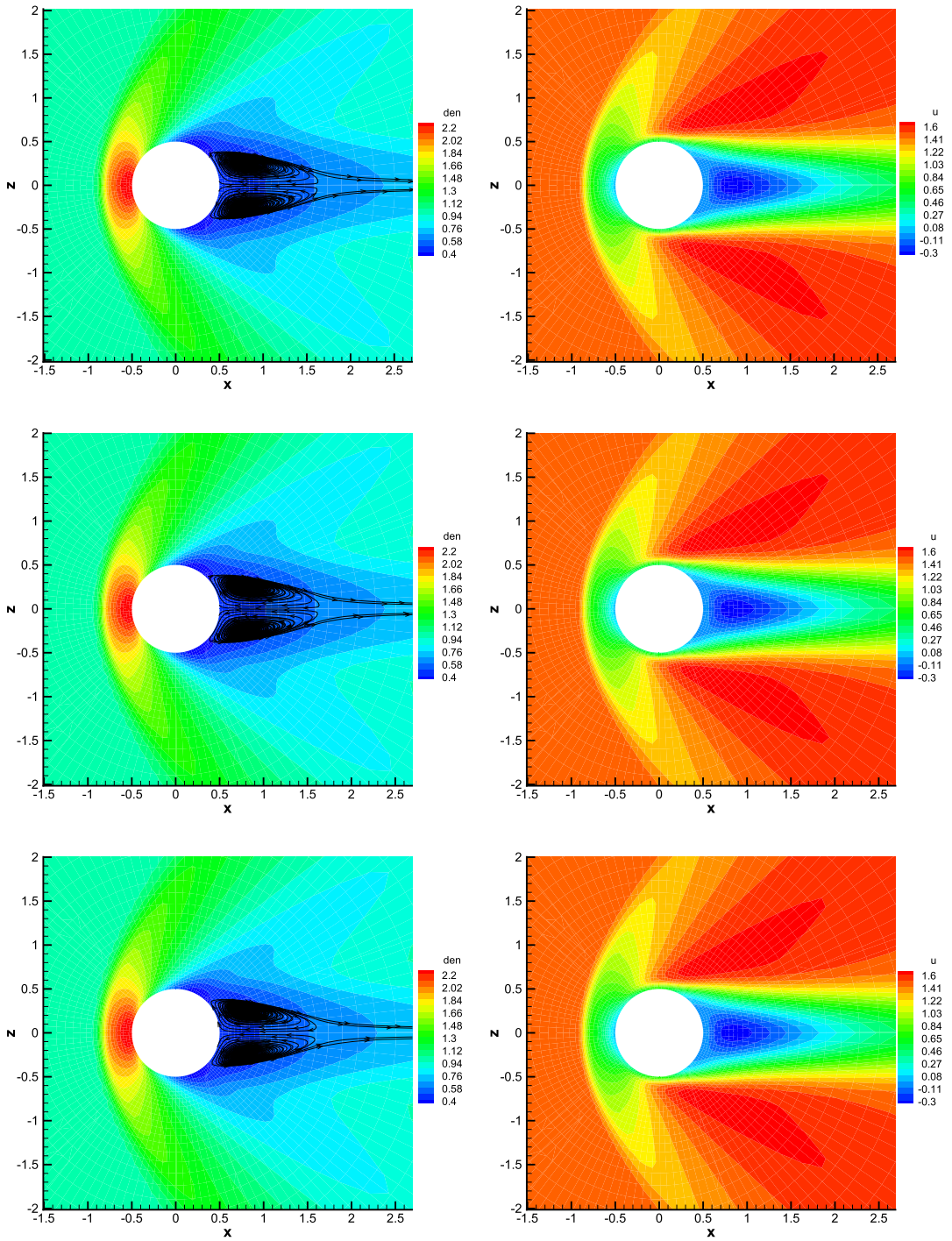

Fig. 8. Supersonic viscous flow passing through a sphere: the density and streamline distributions (left) and velocity distribution (right) at vertical centerline planes with the LUSGS method (top), the non-compact GMRES method (middle) and the compact GMRES method (bottom) for $Re = 300$ and $Ma_\infty = 1.5$.

methods is given in Fig. 9. It also shows that the GMERS method converges much faster than the LUSGS method, and the steady state residual of GMRES method is at least five orders of magnitude smaller than that of LUSGS method.

For the inviscid flows, the case with $Ma_\infty = 3.0$ is performed to test the robustness of implicit scheme. In this case, the CFL number is taken as 3. The density, velocity and streamline distributions at vertical centerline planes are shown in Fig. 10. The shock structure at the leeward side of the sphere is well captured with current implicit schemes. The residual convergence histories with different implicit methods are also shown in Fig. 11. The GMRES method also shows advantage on the convergence rate and the

Table 2
Supersonic viscous flow passing through a sphere: quantitative comparisons of closed wake length L and separation angle θ for $Re = 300$ and $Ma_\infty = 1.5$.

Scheme	Computational Mesh	L	θ
LUSGS GKS	190464 cells, Hex	1.17	135.1
non-compact GMRES GKS	190464 cells, Hex	1.17	135.4
compact GMRES GKS	190464 cells, Hex	1.16	135.9
WENO-CU6-FP RK3 [23]	909072 cells, Hex	0.96	137.2

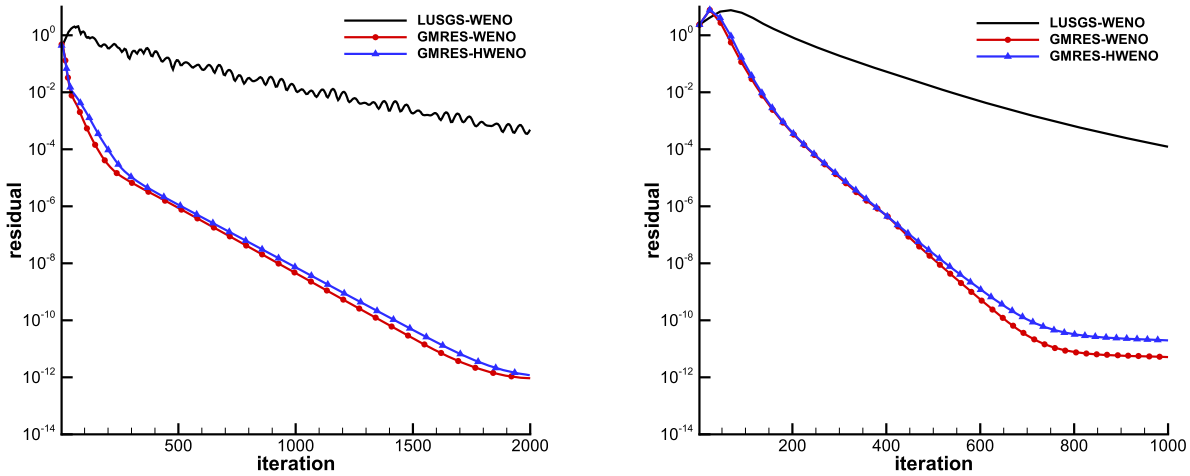

Fig. 9. Subsonic viscous flow passing through a sphere: the residual convergence comparison with the LUSGS method, the non-compact GMRES method and compact GMRES method for $Re = 118$ and $Ma_\infty = 0.2535$ (left) and $Re = 300$ and $Ma_\infty = 1.5$ (right).

order of magnitude of residual. In this case, the minimum number of CFL number is not sufficiently large. In order to explore the relationship between the magnification of CFL number and the characteristic of high-order scheme, the above supersonic inviscid case with $Ma_\infty = 3.0$ is retested by the implicit GMRES method with first order KFVS flux solver. The velocity distribution at vertical centerline planes and residual convergence histories with different CFL numbers are shown in Fig. 12. It shows that the magnification of the CFL number for lower-order schemes on good quality meshes will be more ideal. In the follow-up optimization work, a more reasonable combination between the implicit method and high-order scheme will be investigated.

5.3. Transonic flow around ONERA M6 wing

The transonic flow around the ONERA M6 wing is a standard benchmark for engineering simulations. Besides the three-dimensional geometry, the flow structures are complex including the interaction of shock and turbulent boundary. Thus, it is a good candidate to test the performance of the extended BGK model and implicit high-order gas-kinetic scheme. The inviscid flow around the wing is tested, which corresponds to a rough prediction of the flow field under a very high Reynolds number. The incoming Mach number and angle of attack are given by

$$Ma_\infty = 0.8395,\ AoA = 3.06^\circ.$$

This case is performed by the tetrahedral meshes, which includes 294216 cells. The mesh distribution is shown in Fig. 13. The stencils $S_{i_m}^{HWENO}$ are selected as tetrahedral compact sub-stencils, which are dominated by the gradient informations. The subsonic inflow and outflow boundaries are all set according to the local Riemann invariants, and the adiabatic and slip wall condition is imposed on the solid wall. The local pressure distributions with the LUSGS method, the non-compact GMRES method and compact GMRES method are shown in Fig. 14, and the λ shock is well resolved by all the current implicit schemes. The comparisons on the pressure distributions at the semi-span locations $Y/B = 0.20$, 0.44, 0.65, 0.80, 0.90 and 0.95 of the wing are given in Fig. 15. The numerical results quantitatively agree well with the experimental data [30]. The histories of residual convergence with different implicit methods are given in Fig. 16. It shows that the non-compact GMERS method converges faster than the LUSGS method. Although the residual convergence is affected by the dominant gradient information in tetrahedral compact sub-stencils, the compact GMRES still shows the advantage over the LUSGS method. In the future, the selection of stencils and the evolution of the cell averaged gradients in time will be further investigated.

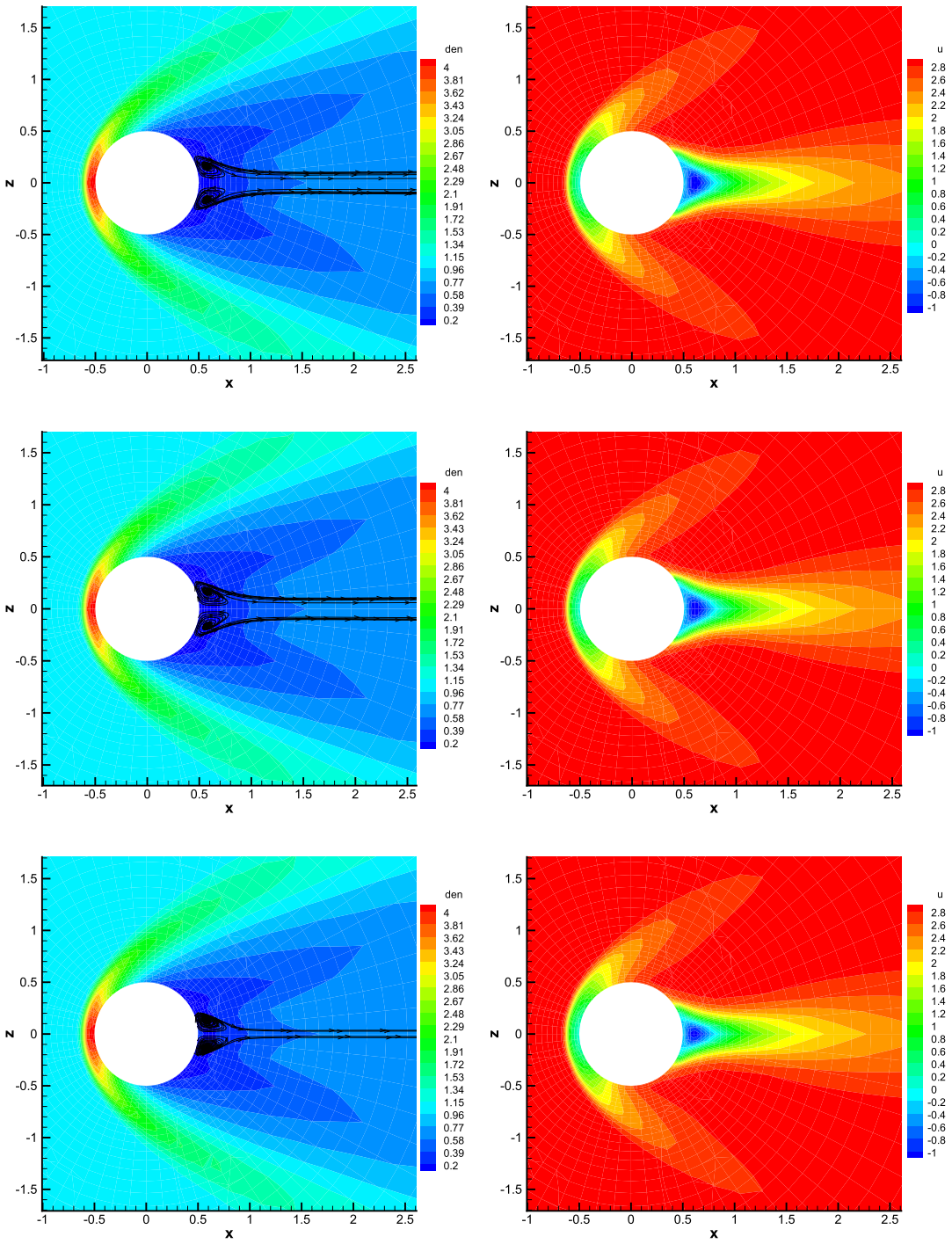

Fig. 10. Supersonic inviscid flow passing through a sphere: the density and streamline distributions (left) and velocity distribution (right) at vertical centerline planes with the LUSGS method (top), the non-compact GMRES method (middle) and the compact GMRES method (bottom) for $Ma_\infty = 3.0$.

5.4. Efficiency comparison of CPU and GPU

The efficiency comparison of CPU and GPU codes is provided for both implicit non-compact and compact schemes. The CPU code is run with Intel Xeon Gold 6230R CPU using Intel Fortran compiler with 16 OpenMP threads, while Nvidia Quadro RTX 8000 is used for GPU code with Nvidia CUDA and NVFORTRAN compiler. The clock rates of GPU and CPU are 1.77 GHz and 2.10 GHz respectively, and the double precision is used in computation. The lid-driven cavity flow with $Re = 1000$ on three dimensional unstructured tetrahedral mesh are used to test the efficiency. The HGKS with GMRES method combined by WENO and HWENO

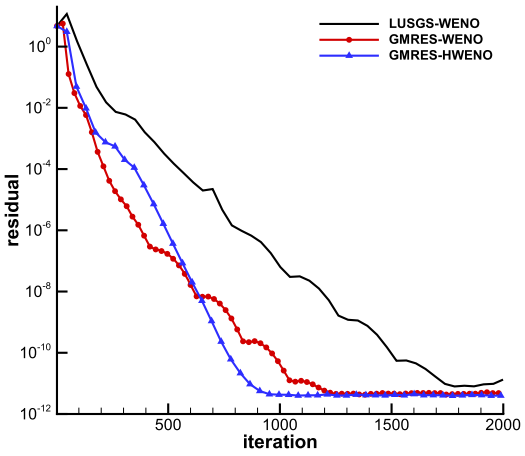

Fig. 11. Supersonic inviscid flow passing through a sphere: the residual convergence comparison with the LUSGS method, the non-compact GMRES method and compact GMRES method for $Ma_\infty = 3.0$.

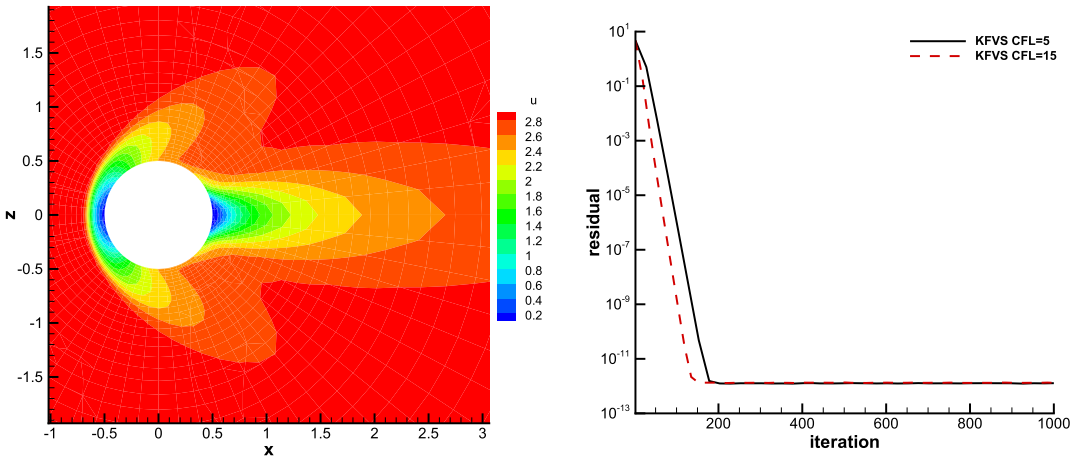

Fig. 12. Supersonic inviscid flow passing through a sphere: the velocity distribution at vertical centerline planes with $CFL = 5$ (left) and the residual convergence comparison (right) tested by the GMRES method combined with first order KFVS reconstruction for $Ma_\infty = 3.0$.

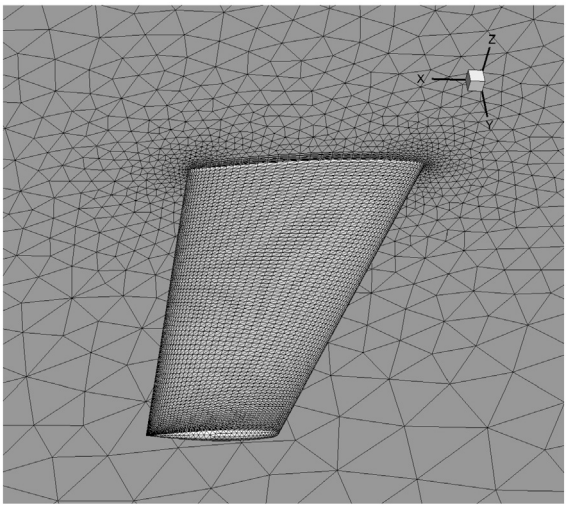

Fig. 13. Transonic flow around ONERA M6 wing: the local computational mesh distribution with tetrahedral meshes.

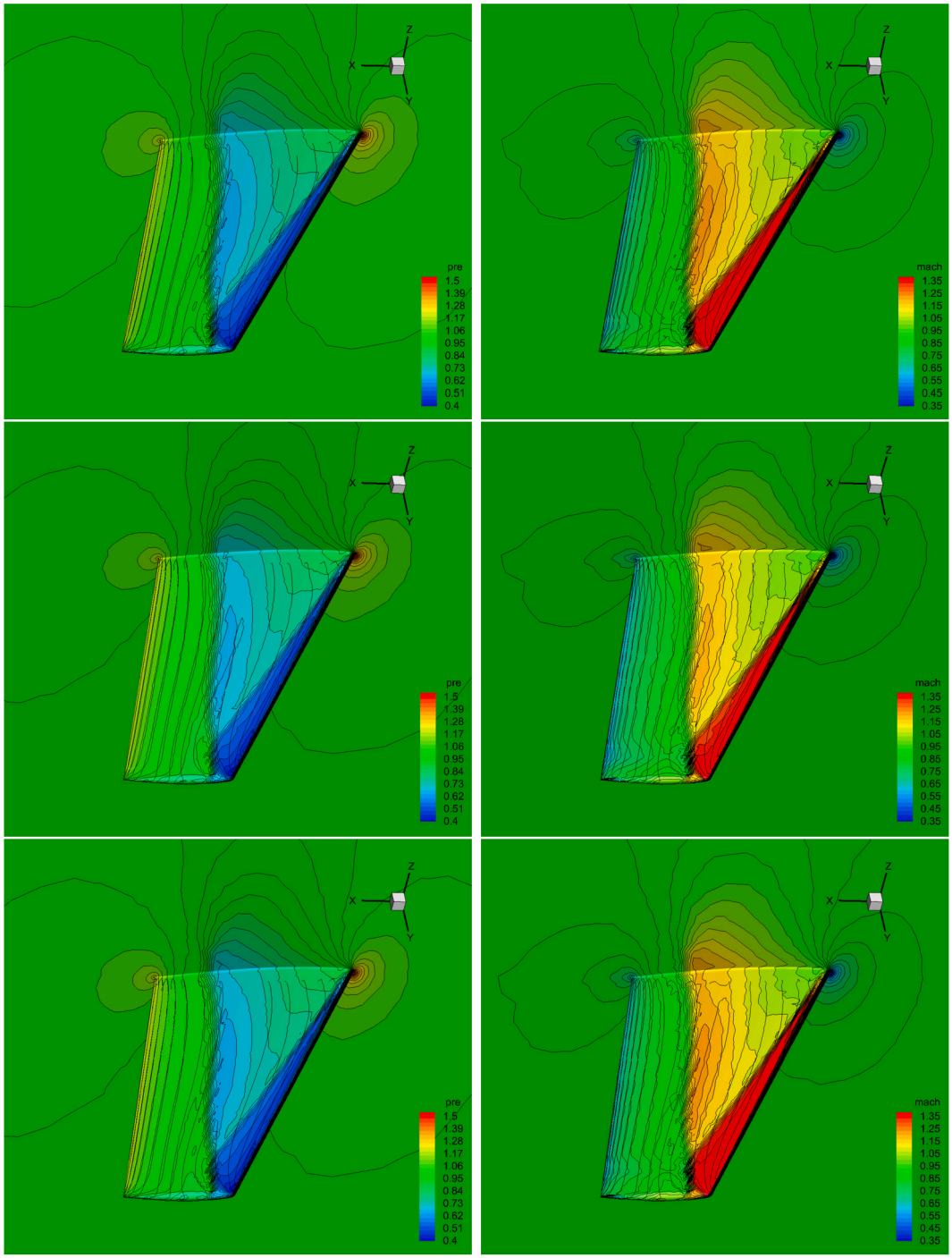

Fig. 14. Transonic flow around ONERA M6 wing: the local pressure distribution with the LUSGS method (top), the non-compact GMRES method (middle) and the compact GMRES method (bottom).

reconstruction are implemented with both GPU and CPU. As shown in Table 3, 8x speedup is achieved for non-compact and compact GMRES GPU code compared with non-compact and compact GMRES CPU code respectively. Taking the number of thread of CPU into account, the speedup of GPU code approximately equals to 130. However, the compact scheme needs to store more local information than the non-compact scheme, the computational scale is constrained by the limited available memory of single GPU. In the future, the implicit compact HGKS on unstructured meshes will be further upgraded with multiple GPUs using MPI and CUDA, and more challenging problems for compressible flows will be investigated.

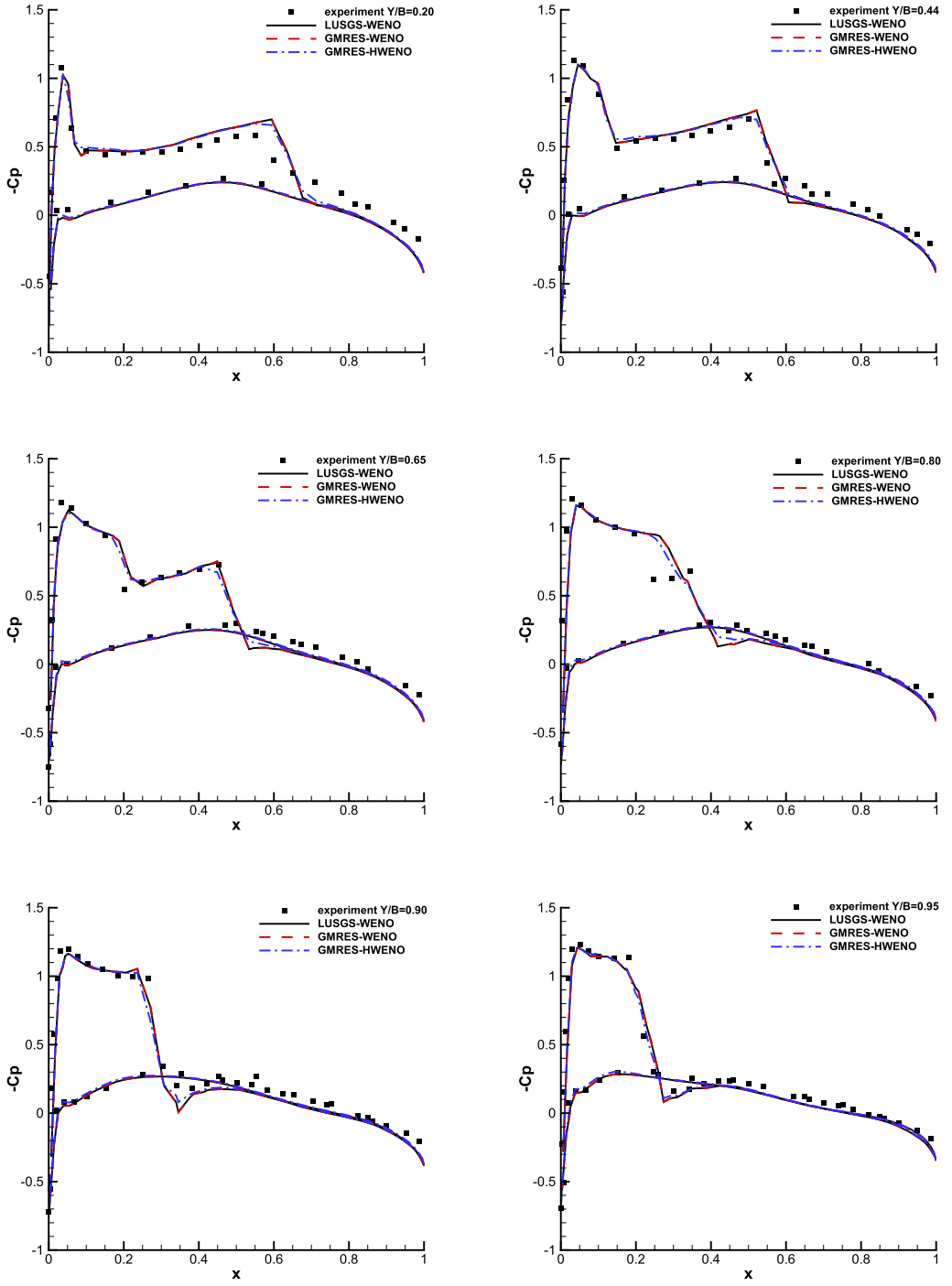

Fig. 15. ONERA M6 wing: the pressure coefficient distributions at $Y/B = 0.20$, 0.44, 0.65, 0.80, 0.90 and 0.95 for the inviscid flow with tetrahedral and hybrid meshes.

Table 3
Efficiency comparison: the computational time per 100 steps and speedup for GPU and CPU code. Taking the number of thread of CPU into account, the speedup of GPU code approximately equals to 130.

Scheme	Computational Mesh	CPU+OpenMP	GPU+CUDA	Speedup
noncompact GMRES	48000 Tet	259 s	30 s	8.63
compact GMRES	48000 Tet	267 s	32 s	8.34

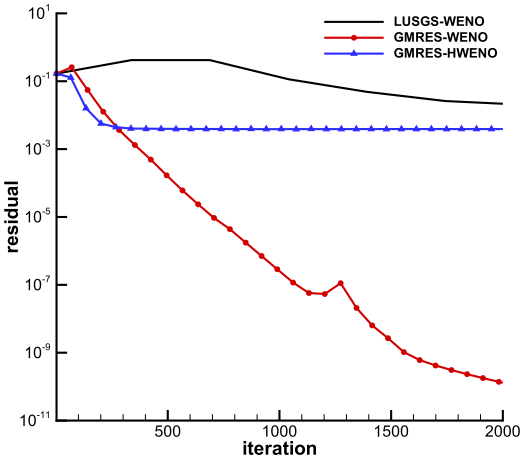

Fig. 16. Transonic flow around ONERA M6 wing: the residual convergence comparison with the LUSGS method, the noncompact GMRES method and compact GMRES method.

6. Conclusion

In this paper, the implicit non-compact and compact HGKSs are developed on the three-dimensional unstructured meshes. For non-compact GKS scheme, the third-order WENO reconstruction is used, where the stencils are selected from the neighboring cells and the neighboring cells of neighboring cells. Incorporate with the GMRES method based on numerical Jacobian matrix, the implicit non-compact HGKS is developed for steady problems. To improve the resolution and parallelism, the implicit compact HGKS is also developed with HWENO reconstruction, where the stencils only contain one level of neighboring cells. The cell averaged conservative variable is updated with GMRES method. Simultaneously, a simple strategy is used to update the cell averaged gradient with the spatial-temporal coupled gas-kinetic flow solver. To further accelerate the computation, the Jacobi iteration is chosen as preconditioner for both non-compact and compact schemes. Various three-dimensional numerical experiments, from the subsonic to supersonic flows, are presented to validate the accuracy and robustness of current implicit scheme. To accelerate the computation, the current schemes are implemented to run on graphics processing unit (GPU) using compute unified device architecture (CUDA). In the future, the implicit HGKS on arbitrary unstructured meshes with multiple GPUs will be developed for the engineering turbulent flows with high-Reynolds numbers.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank Mr. Yue Zhang for helpful discussion. The current research of L. Pan is supported by Beijing Natural Science Foundation (1232012), National Natural Science Foundation of China (11701038) and the Fundamental Research Funds for the Central Universities, China. The work of K. Xu is supported by National Key R&D Program of China (2022YFA1004500), National Natural Science Foundation of China (12172316), and Hong Kong research grant council (16208021,16301222).

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[2] S. Albensoeder, H.C. Kuhlmann, Accurate three-dimensional lid-driven cavity flow, J. Comput. Phys. 206 (2005) 536–558.
[3] P.L. Bhatnagar, E.P. Gross, M. Krook, A model for collision processes in gases I: small amplitude processes in charged and neutral one-component systems, Phys. Rev. 94 (1954) 511–525.
[4] P.N. Brown, Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Comput. 11 (1990) 450–481.
[5] G.Y. Cao, L. Pan, K. Xu, High-order gas-kinetic scheme with parallel computation for direct numerical simulation of turbulent flows, J. Comput. Phys. 448 (2022) 110739.
[6] S. Chapman, T.G. Cowling, The Mathematical Theory of Non-Uniform Gases, third edition, Cambridge University Press, 1990.
[7] R.F. Chen, Z.J. Wang, Fast, block lower-upper symmetric Gauss-Seidel scheme for arbitrary grids, AIAA J. 38 (2000) 2238–2245.

[8] J. Cheng, X.Q. Yang, X.D. Liu, T.D. Liu, H. Luo, A direct discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids, J. Comput. Phys. 327 (2016) 484–502.
[9] J. Cheng, X. Liu, T.G. Liu, H. Luo, A. Parallel, High-order direct discontinuous Galerkin method for the Navier-Stokes equations on 3D hybrid grids, Commun. Comput. Phys. 21 (2017) 1231–1257.
[10] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[11] A. Crivellini, F. Bassi, An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations, Comput. Fluids 50 (2011) 81–93.
[12] Z.F. Du, J.Q. Li, A Hermite WENO reconstruction for fourth order temporal accurate schemes based on the GRP solver for hyperbolic conservation laws, J. Comput. Phys. 355 (2018) 385–396.
[13] S. Gottlieb, C.W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.
[14] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[15] H.T. Huynh, A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods, AIAA Paper 2007-4079.
[16] X. Ji, L. Pan, W. Shyy, K. Xu, A compact fourth-order gas-kinetic scheme for the Euler and Navier-Stokes equations, J. Comput. Phys. 372 (2018) 446–472.
[17] X. Ji, F. Zhao, W. Shyy, K. Xu, A HWENO reconstruction based high-order compact gas-kinetic scheme on unstructured mesh, J. Comput. Phys. 410 (2020) 109367.
[18] J.Q. Li, Z.F. Du, A two-stage fourth order time-accurate discretization for Lax-Wendroff type flow solvers I. Hyperbolic conservation laws, SIAM J. Sci. Comput. 38 (2016) 3046–3069.
[19] W.A. Li, Y.X. Ren, The multi-dimensional limiters for solving hyperbolic conservation laws on unstructured grids II: extension to high order finite volume schemes, J. Comput. Phys. 231 (2012) 4053–4077.
[20] H. Luo, J.D. Baum, R. Lohner, J. Cabello, Implicit Schemes and Boundary Conditions for Compressible Flows on Unstructured Meshes, AIAA Paper 94-0816, 1994.
[21] H. Luo, J.D. Baum, R. Löhner, A. Fast, Matrix-free implicit method for compressible flows on unstructured grids, J. Comput. Phys. 146 (1998) 664–690.
[22] H. Luo, J.D. Baum, R. Löhner, An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids, Comput. Fluids 30 (2001) 137–159.
[23] T. Nagata, T. Nonomura, S. Takahashi, Y. Mizuno, K. Fukuda, Investigation on subsonic to supersonic flow around a sphere at low Reynolds number of between 50 and 300 by direct numerical simulation, Phys. Fluids 28 (2016) 056101.
[24] L. Pan, K. Xu, Q.B. Li, J.Q. Li, An efficient and accurate two-stage fourth-order gas-kinetic scheme for the Navier-Stokes equations, J. Comput. Phys. 326 (2016) 197–221.
[25] L. Pan, J. Li, K. Xu, A few benchmark test cases for higher-order Euler solvers, Numer. Math., Theory Methods Appl. 10 (4) (2017) 711–736.
[26] J.X. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case, J. Comput. Phys. 193 (2004) 115–135.
[27] J.X. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method, III: unstructured meshes, J. Sci. Comput. 39 (2009) 293–321.
[28] Y.P. Reng, Y.L. Xing, J.X. Qiu, High order finite difference Hermite WENO fast sweeping methods for static Hamilton-Jacobi equations, https://doi.org/10.48550/arXiv.2009.03494, 2020.
[29] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Comput. 7 (1986) 856–869.
[30] V. Schmitt, F. Charpin, Pressure distributions on the ONERA-M6-wing at transonic Mach numbers, Experimental Data Base for Computer Program Assessment, Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, 1979.
[31] D. Sharov, K. Nakahashi, Reordering of 3D hybrid unstructured grids for vectorized LU-SGS Navier-Stokes computations, AIAA Pap. 97 (1997) 2102–2117.
[32] C. Shu, L. Wang, Y.T. Chew, Numerical computation of three-dimensional incompressible Navier-Stokes equations in primitive variable form by DQ method, Int. J. Numer. Methods Fluids 43 (2003) 345–368.
[33] S. Tan, Q.B. Li, Time-implicit gas-kinetic scheme, Comput. Fluids 144 (2017) 44–59.
[34] S. Taneda, Studies on wake vertices, experimental investigation of the wake behind a sphere at low Reynolds numbers, J. Phys. Soc. Jpn. 11 (1956) 1104–1108.
[35] V. Venkatakrishnan, D.J. Mavriplis, Implicit solvers for unstructured meshes, J. Comput. Phys. 105 (1993) 83–91.
[36] Y.H. Wang, G.Y. Cao, L. Pan, Multiple-GPU accelerated high-order gas-kinetic scheme for direct numerical simulation of compressible turbulence, J. Comput. Phys. 448 (476) (2023) 111899.
[37] Z.J. Wang, H. Gao, A unifying lifting collocation penalty formulation including the discon- tinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids, J. Comput. Phys. 228 (2009) 8161–8186.
[38] K. Xu, A gas-kinetic BGK scheme for the Navier-Stokes equations and its connection with artificial dissipation and Godunov method, J. Comput. Phys. 171 (2001) 289–335.
[39] K. Xu, Direct Modeling for Computational Fluid Dynamics: Construction and Application of Unfied Gas Kinetic Schemes, World Scientific, 2015.
[40] X.Q. Yang, J. Cheng, H. Luo, Q.J. Zhao, Robust implicit direct discontinuous Galerkin method for simulating the compressible turbulent flows, AIAA Paper 2016-1326.
[41] Y.Q. Yang, L. Pan, K. Xu, High-order gas-kinetic scheme on three-dimensional unstructured meshes for compressible flows, Phys. Fluids 33 (2021) 096102.
[42] Y.Q. Yang, L. Pan, K. Xu, Three-dimensional third-order gas-kinetic scheme on hybrid unstructured meshes for Euler and Navier-Stokes equations, Comput. Fluids 255 (2023) 105834.
[43] S. Yoon, A. Jameson, Lower-upper symmetric-Gauss-Seidel method for the Euler and Navier-Stokes equations, AIAA J. 26 (1988) 1025–1026.
[44] F.X. Zhao, X. Ji, W. Shyy, K. Xu, Compact higher-order gas-kinetic schemes with spectral-like resolution for compressible flow simulations, Adv. Aerodyn. 1 (2019) 13.
[45] F.X. Zhao, X. Ji, W. Shyy, K. Xu, A compact high-order gas-kinetic scheme on unstructured mesh for acoustic and shock wave computations, J. Comput. Phys. 449 (2022) 110812.
[46] F.X. Zhao, L. Pan, S.H. Wang, Weighted essentially non-oscillatory scheme on unstructured quadrilateral and triangular meshes for hyperbolic conservation laws, J. Comput. Phys. 374 (2018) 605–624.
[47] L.P. Zhang, W. Liu, M. Li, H.X. Zhang, A class of hybrid DG/FV methods for conservation laws IV: 2D viscous flows and implicit algorithm for steady cases, Comput. Fluids 97 (2014) 110–125.
[48] M. Zhang, Z. Zhao, A fifth-order finite difference HWENO scheme combined with limiter for hyperbolic conservation laws, J. Comput. Phys. 472 (2023) 111676.
[49] J. Zhu, J.X. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, J. Comput. Phys. 318 (2016) 110–121.
[50] J. Zhu, J.X. Qiu, New finite volume weighted essentially non-oscillatory scheme on triangular meshes, SIAM J. Sci. Comput. 40 (2018) 903–928.
[51] Y.J. Zhu, C.W. Zhong, K. Xu, Implicit unified gas-kinetic scheme for steady state solutions in all flow regimes, J. Comput. Phys. 315 (2016) 16–38.