An h-Adaptivity DG Method on Locally Curved Tetrahedral Mesh for Solving Compressible Flows

AN Wei[1,2,3], *HUANG Zenghui*[1], *LYU Hongqiang*[1*]

1. College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, P.R. China;
2. Key Laboratory of Aerodynamic Noise Control, China Aerodynamics Research and Development Center, Mianyang 621000, P.R. China;
3. State Key Laboratory of Aerodynamics, China Aerodynamics Research and Development Center, Mianyang 621000, P.R. China

(Received 3 June 2019; revised 11 November 2019; accepted 19 November 2019)

Abstract: For the numerical simulation of compressible flows, normally different mesh sizes are expected in different regions. For example, smaller mesh sizes are required to improve the local numerical resolution in the regions where the physical variables vary violently (for example, near the shock waves or in the boundary layers) and larger elements are expected for the regions where the solution is smooth. h-adaptive mesh has been widely used for complex flows. However, there are two difficulties when employing h-adaptivity for high-order discontinuous Galerkin (DG) methods. First, locally curved elements are required to precisely match the solid boundary, which significantly increases the difficulty to conduct the "refining" and "coarsening" operations since the curved information has to be maintained. Second, h-adaptivity could break the partition balancing, which would significantly affect the efficiency of parallel computing. In this paper, a robust and automatic h-adaptive method is developed for high-order DG methods on locally curved tetrahedral mesh, for which the curved geometries are maintained during the h-adaptivity. Furthermore, the reallocating and rebalancing of the computational loads on parallel clusters are conducted to maintain the parallel efficiency. Numerical results indicate that the introduced h-adaptive method is able to generate more reasonable mesh according to the structure of flow-fields.

Key words: h-adaptivity; discontinuous Galerkin (DG) method; curved mesh; tetrahedral mesh; compressible flows

CLC number: V211.3 **Document code:** A **Article ID:** 1005-1120(2020)05-0702-11

0 Introduction

Many flow fields have multiscale structures which are difficult to numerically capture since it is impossible to give a proper mesh before the computation. Adaptive meshing techniques are introduced to increase the solution accuracy, as well as to reduce the computational time and computer memory. There are three typical adaptive meshing techniques: r-adaptivity, h-adaptivity and p-adaptivity. The r-adaptivity makes the mesh distribution match the flow structures by changing the positions of the grid points[1-6]. The h-adaptivity refines the region including small structures by dividing the original elements into smaller elements, and coarsens the region where the solution is smooth enough by agglomerating smaller elements into larger elements[7-11]. The p-adaptiviy methods are only suited to high-order methods and they employ different orders in different regions instead of changing the mesh[12-17]. Besides, the three adaptive meshing techniques can be used simultaneously. For example, the h-adaptivity changes grids and the p-adaptivity uses different orders for different elements.

Due to the limit of the number of elements and

*Corresponding author, E-mail address: hongqiang.lu@nuaa.edu.cn.

How to cite this article: AN Wei, HUANG Zenghui, LYU Hongqiang. An h-adaptive DG method on locally curved tetrahedral mesh for solving compressible flows[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(5): 702-712.

http://dx.doi.org/10.16356/j.1005-1120.2020.05.005

the highest order, the r-adaptivity and the p-adaptivity methods may not be able to provide sufficiently accurate solutions for complex flows. Therefore, the h-adaptivity is adopted in this paper. Babuska introduced h-adaptivity to finite element method to improve the accuracy in 1975[7]. Lentini and Pereyra combined h-adaptivity and finite difference method in the same year[18]. Then, h-adaptivity is widely used for the finite volume method[19-21]. In recent years, some researchers have applied h-adaptivity and discontinuous Galerkin (DG) methods to solve 2D flows[22-29]. Wu and Yu[30] employed h-adaptivity and DG method to solve the steady-state 3D Euler equations.

Compared with the traditional finite volume method, there are extra difficulties to implement the h-adaptivity for high-order DG methods on tetrahedral mesh. The curved geometry has to be maintained during the refining and coarsening operations. Furthermore, re-balancing operation is necessary for ensuring the efficiency of the parallel computation. In this paper, a 3D h-adaptive high-order DG method with a re-balancing parallel strategy is introduced to solve compressible flow problems on locally curved tetrahedral mesh. The details of the computation scheme are given, which includes the numerical method, h-adaptivity indicator, 3D h-adaptivity and the parallel computing. Then, the numerical results are displayed.

1 Computation Scheme

1.1 Numerical method

1.1.1 Governing equations

The Navier-Stokes equations in conversation form are given by

$$\frac{\partial \boldsymbol{U}}{\partial t}+\nabla\cdot \boldsymbol{F}_c(\boldsymbol{U})-\nabla\cdot \boldsymbol{F}_v(\boldsymbol{U},\nabla \boldsymbol{U})=0 \tag{1}$$

where $\boldsymbol{U}$ are the conversation variables. $\boldsymbol{F}_c(\boldsymbol{U})$ and $\boldsymbol{F}_v(\boldsymbol{U},\nabla \boldsymbol{U})$ are the inviscid and viscous flux terms, respectively.

$$\boldsymbol{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{pmatrix}\quad \boldsymbol{F}_c^x=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ \rho uh\end{pmatrix}\quad \boldsymbol{F}_c^y=\begin{pmatrix}\rho v\\ \rho vu\\ \rho v^2+p\\ \rho vw\\ \rho vh\end{pmatrix}$$

$$\boldsymbol{F}_c^z=\begin{pmatrix}\rho w\\ \rho wu\\ \rho wv\\ \rho w^2+p\\ \rho wh\end{pmatrix}$$

$$\boldsymbol{F}_v^x=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ \tau_{xz}\\ u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x\end{pmatrix}$$

$$\boldsymbol{F}_v^y=\begin{pmatrix}0\\ \tau_{yx}\\ \tau_{yy}\\ \tau_{yz}\\ u\tau_{yx}+v\tau_{yy}+w\tau_{yz}-q_y\end{pmatrix}$$

$$\boldsymbol{F}_v^z=\begin{pmatrix}0\\ \tau_{zx}\\ \tau_{zy}\\ \tau_{zz}\\ u\tau_{zx}+v\tau_{zy}+w\tau_{zz}-q_z\end{pmatrix}$$

where ρ is the density, p the pressure, E the total internal energy per unit mass, h the total enthalpy per unit mass, and τ the stress tensor. u, v, and w are the velocity components.

1.1.2 DG discretization

Employing the well-known BR2 scheme[31], Eq.(1) can be discretized as

$$\frac{\partial}{\partial t}\int_{\Omega_e}\varphi_i \boldsymbol{U}_h \mathrm{d}\Omega_e+\int_{\partial\Omega_e}\varphi_i \boldsymbol{H}(\boldsymbol{U}_h^-,\nabla \boldsymbol{U}_h^-+\boldsymbol{r}_e^-,\boldsymbol{U}_h^+,\nabla \boldsymbol{U}_h^++\boldsymbol{r}_e^+,\boldsymbol{n})\mathrm{d}\sigma-\int_{\Omega_e}\nabla\varphi_i\cdot \boldsymbol{F}(\boldsymbol{U}_h,\nabla \boldsymbol{U}_h+\boldsymbol{R})\mathrm{d}\Omega_e=0$$
$$1\leqslant i\leqslant N(p) \tag{2}$$

where φ_i are the basis functions, $\int_{\Omega_e}\varphi_i \boldsymbol{r}_e \mathrm{d}\Omega_e=\int_e \varphi_i(\boldsymbol{U}_0-\boldsymbol{U})\boldsymbol{n}\mathrm{d}\sigma$ and $\boldsymbol{R}=\sum_{e\in\partial\Omega_e}\boldsymbol{r}_e$. $\boldsymbol{H}=\boldsymbol{H}_c+\boldsymbol{H}_v$, where

$$\boldsymbol{H}_c(\boldsymbol{U}_h^-,\boldsymbol{U}_h^+,\boldsymbol{n})=\frac{1}{2}[\boldsymbol{F}_c(\boldsymbol{U}_h^-)+\boldsymbol{F}_c(\boldsymbol{U}_h^+)]\cdot\boldsymbol{n}-\frac{1}{2}\alpha(\boldsymbol{U}_h^+-\boldsymbol{U}_h^-)$$

$$\boldsymbol{H}_v(\boldsymbol{U}_h^-,\nabla \boldsymbol{U}_h^-+\boldsymbol{r}_e^-,\boldsymbol{U}_h^+,\nabla \boldsymbol{U}_h^++\boldsymbol{r}_e^+,\boldsymbol{n})=\frac{1}{2}[\boldsymbol{F}_v(\boldsymbol{U}_h^-,\nabla \boldsymbol{U}_h^-+\boldsymbol{r}_e^-)+\boldsymbol{F}_v(\boldsymbol{U}_h^+,\nabla \boldsymbol{U}_h^++\boldsymbol{r}_e^+)]\cdot\boldsymbol{n}$$

The final discrete system can be written as

$$L(\boldsymbol{u})=\boldsymbol{M}\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t}+\boldsymbol{R}(\boldsymbol{u})=0$$

The above system can be solved using the following implicit relaxation method.

$$\boldsymbol{u}^{(0)}=\boldsymbol{u}^{(n)}$$

$$\begin{cases}\left[\dfrac{\boldsymbol{M}}{\Delta t}-\left(\dfrac{\partial \boldsymbol{R}}{\partial \boldsymbol{u}}\right)^{k}\right]\Delta\boldsymbol{u}=\boldsymbol{R}(\boldsymbol{u}^{k})-\dfrac{\boldsymbol{M}}{\Delta t}(\boldsymbol{u}^{k}-\boldsymbol{u}^{0}) & (3a)\\ \boldsymbol{u}^{k+1}=\boldsymbol{u}^{k}+c\Delta\boldsymbol{u} & (3b)\end{cases}$$

$$k=1,\cdots,k_{\max}$$

$$\boldsymbol{u}^{n+1}=\boldsymbol{u}^{(k_{\max})}$$

1.2 h-adaptivity indicator

In order to capture the shocks, an artificial viscosity approach[29] is employed for transonic and supersonic cases, for which an artificial viscosity term is introduced to the original governing equations

$$\frac{\partial \boldsymbol{U}}{\partial t}+\nabla\cdot\boldsymbol{F}_c(\boldsymbol{U})-\nabla\cdot\boldsymbol{F}_v(\boldsymbol{U},\nabla\boldsymbol{U})=\nabla\cdot(\varepsilon\nabla\boldsymbol{U}) \quad (4)$$

where ε is defined as the artificial viscosity coefficient in element e

$$\varepsilon_e=\begin{cases}0 & s_e<s_0-\kappa\\ \dfrac{\varepsilon_0}{2}\left(1+\sin\dfrac{\pi(s_e-s_0)}{2\kappa}\right) & s_0-\kappa\leqslant s_e\leqslant s_0+\kappa\\ \varepsilon_0 & s_e>s_0+\kappa\end{cases} \quad (5)$$

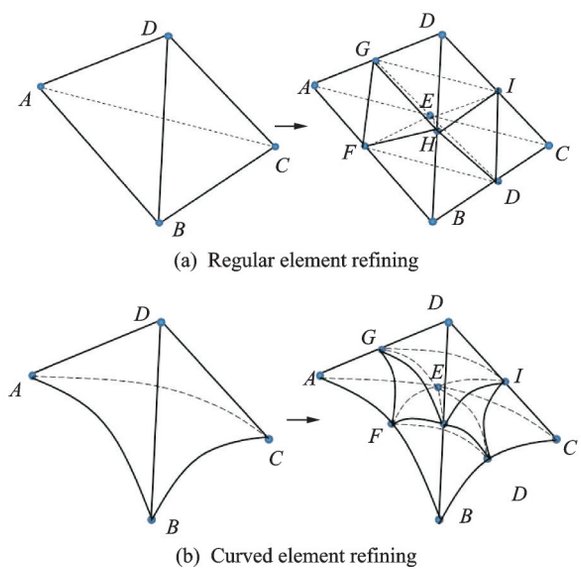
where $s_e=\log_{10}S_e$, $\varepsilon_0=D_e/p$, $s_0=\log_{10}(1/p^4)$, κ is an constant and D_e the size of element e. S_e is given as

$$S_e=\frac{(\boldsymbol{U}-\widetilde{\boldsymbol{U}},\boldsymbol{U}-\widetilde{\boldsymbol{U}})_e}{(\boldsymbol{U},\boldsymbol{U})_e} \quad (6)$$

Since ε tends to 0 in smooth region and becomes larger in the region with great gradient such as the region near shocks, it can be used as an h-adaptivity indicator. For the viscous flows without shocks, the magnitude of the vorticity can be employed as the h-adaptivity indicator.

1.3 h-adaptivity

Tetrahedron elements are employed in the entire computational domain here, and the h-adaptivity operations consist of refining and coarsening. Fig.1(a) displays the refining procedure, where the original "father" element on the left is divided into eight "child" elements on the right. To ensure the accuracy of the solid boundaries, the elements near the solid boundaries are usually curved. Fig.1(b) demonstrates the way to divide a curved element into smaller ones, where "child" elements are also curved to accurately match the geometry of "father" element. The coarsening procedure is exactly the inverse operation of those shown in Fig.1. Note that the way to describe curved elements is not unique. For example, we can use both high-order polynomials and governing points located on the edges and faces of an element to define the element geometry. In this paper, high-order polynomials are used, from which all required curved information can be computed, thereby maintaining the curved geometries during the refining and coarsening operations.



(a) Regular element refining

(b) Curved element refining

Fig.1 Element refining

A generation label is created for each element over the computational domain. All of the original elements are labelled 0 and the "child" elements generated after refining n times will be labelled n. To avoid the significant difference of the neighboring elements in size, the difference in the generation label of the neighboring elements are limited to be 1. For example, once the element "e" in Fig.2 is refined, the element "f" will also be refined to keep the local mesh size relatively smooth.

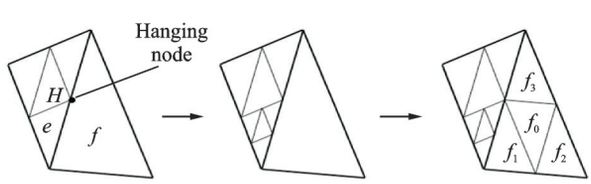


Fig.2 Refining the neighboring elements

The elements over the computational domain are numbered as shown in Fig.3. Once the element E in Fig.3(a) is refined into eight "child" elements, one inherits the number E and the others are numbered from $N+1$ to $N+7$. Fig.3(b) indicates the inverse operation, where the elements E and f_1—f_7 are agglomerated back to their "father" element E. This dynamic ordering rule maintains the chain structure of the original mesh, for which the computational module does not need to be modified during the h-adaptivity.

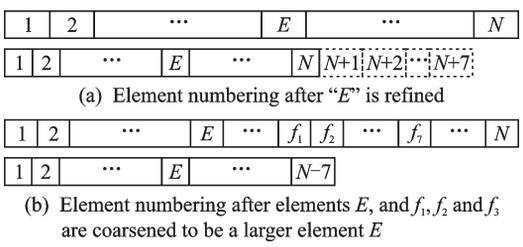


Fig.3 Data structure of elements during refining and coarsening

1.4 Parallel computing

It is well known that DG methods are well suited for parallel computing. The computation in this paper starts from partitioning the initial mesh into N partitions which have nearly equal number of elements with each other. However, after conducting h-adaptivity, the partition load balance would be broken, which means that the number of elements of each partition may be quite different. The unbalanced partition load would result in poor efficiency of the parallel computing.

This paper re-balances the mesh partitions after h-adaptivity. First, the elements to be refined or coarsened are marked. Second, all the processes send the marked h-adaptivity information to the first process. Third, the first process conducts the refining and coarsening over the entire domain. Finally, the first process redivides the entire computational domain into N partitions and distributes them to the corresponding processors. Then the computation continues on the new mesh.

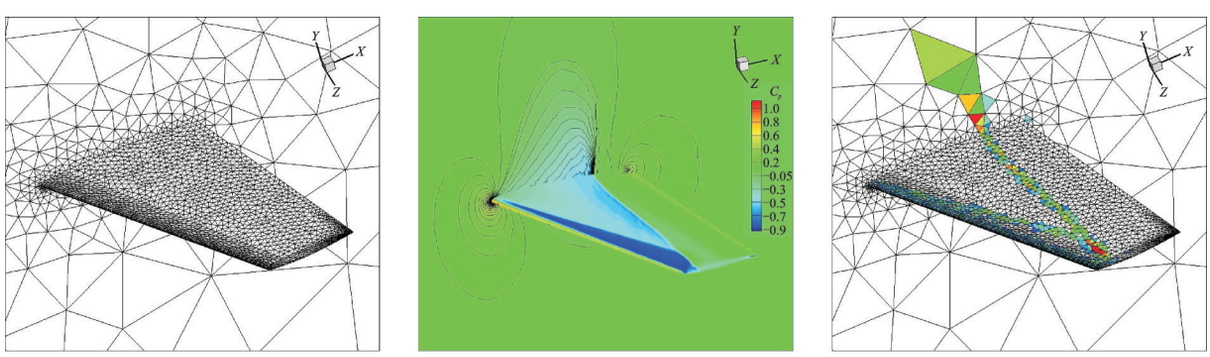
2 Numerical Results

2.1 Inviscid flow around ONERA M6 wing

To validate the h-adaptivity method for the shock capturing, the typical inviscid transonic flow around the ONERA M6 wing is numerically simulated, for which $Ma_\infty = 0.84$, $\alpha = 3.06$. The 3D Euler equations are solved using the DG method ($p=2$), where the artificial viscosity term is added into the Euler equations to capture the shocks, as shown in Eq.(3).

Fig.4 presents the results computed on the initial mesh. Fig.4(a) shows the mesh distribution and there are 31 044 tetrahedral elements in total over the entire domain. The contours of the pressure coefficient are displayed in Fig.4(b). The distribution of ε is given in Fig.4(c), which matches the shocks well. In the region where ε is relatively large, the elements are going to be refined to improve the resolution of the shocks.

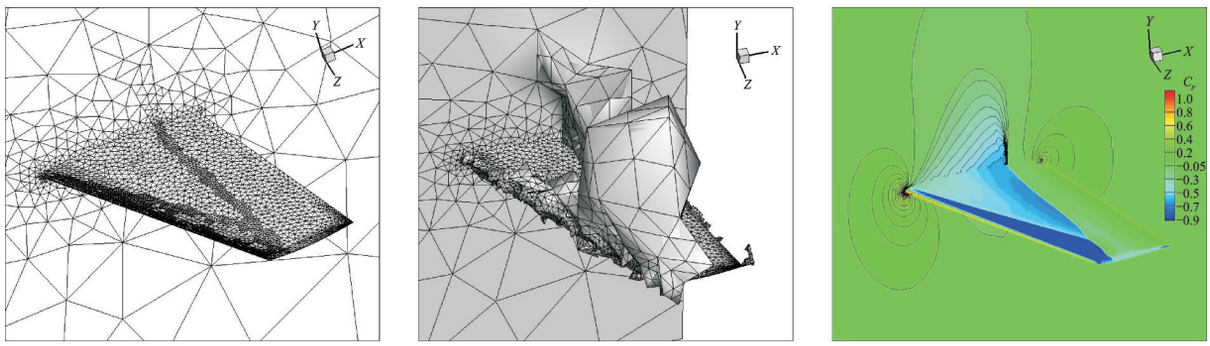
Using the h-adaptivity method developed previously, a more accurate numerical result is obtained after only one time h-adaptivity. Figs.5(a),(b) display the final mesh distribution on the surface of the wing and the refined elements in the space, respectively. Fig.5(c) presents the new contours of the pressure distribution, where the shocks are much



(a) Mesh distribution (b) Contours of pressure coefficient (c) Distribution of ε

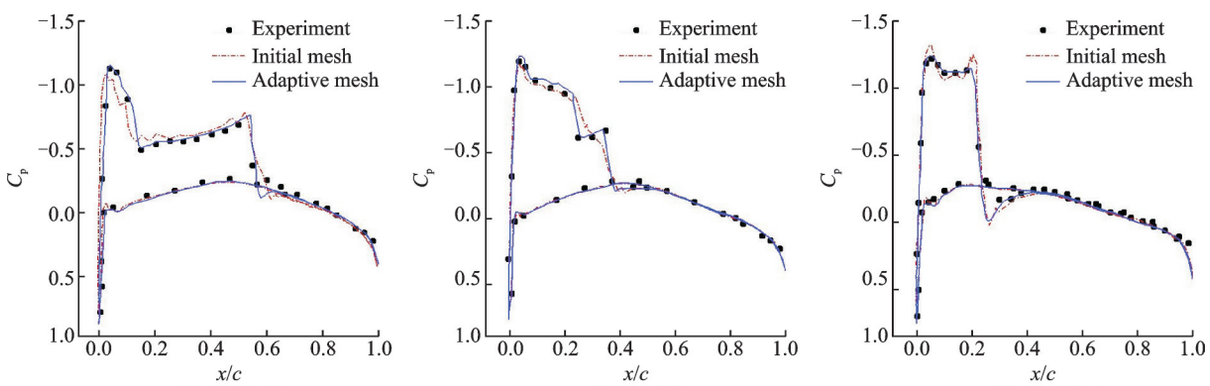
Fig.4 Numerical results obtained on the initial mesh of ONERA M6 wing

more accurately captured than that in Fig.4(b). The distributions of the pressure coefficients on different spanwise profiles are given in Fig.6, where the shocks obtained after the h-adaptivity are sharper than those captured on the initial mesh and match better with the experimental results.



(a) Final mesh distribution on solid surface (b) Refined elements (c) Distribution of pressure coefficients

Fig.5 Numerical results obtained on the final mesh of ONERA M6 wing



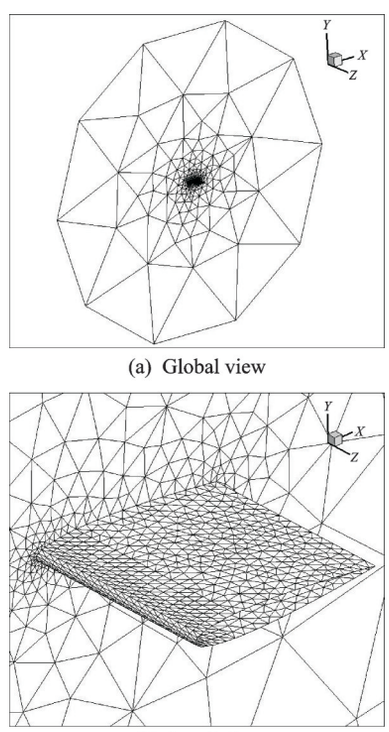
(a) $Z=0.44$ (b) $Z=0.8$ (c) $Z=0.95$

Fig.6 Distributions of pressure coefficients at different spanwise profiles of ONERA M6 wing

2.2 Inviscid flow around pitching NACA0012 wing

To verify the applicability of the introduced h-adaptivity method to unsteady flows, the inviscid flow around a periodically pitching NACA0012 wing is numerically simulated, for which the infinite Mach number is 0.755. The wing pitches around $Z=0.25$ periodically and the attack angle varies as $\alpha(\tau)=\alpha_0+\alpha_m\sin(2K\tau)$, where $\alpha_0=0.016°$, $\alpha_m=2.51°$, $K=0.081\,4$. The DG method of 3rd-order accuracy ($p=2$) is employed and the explicit Runge-Kutta time-stepping is used.

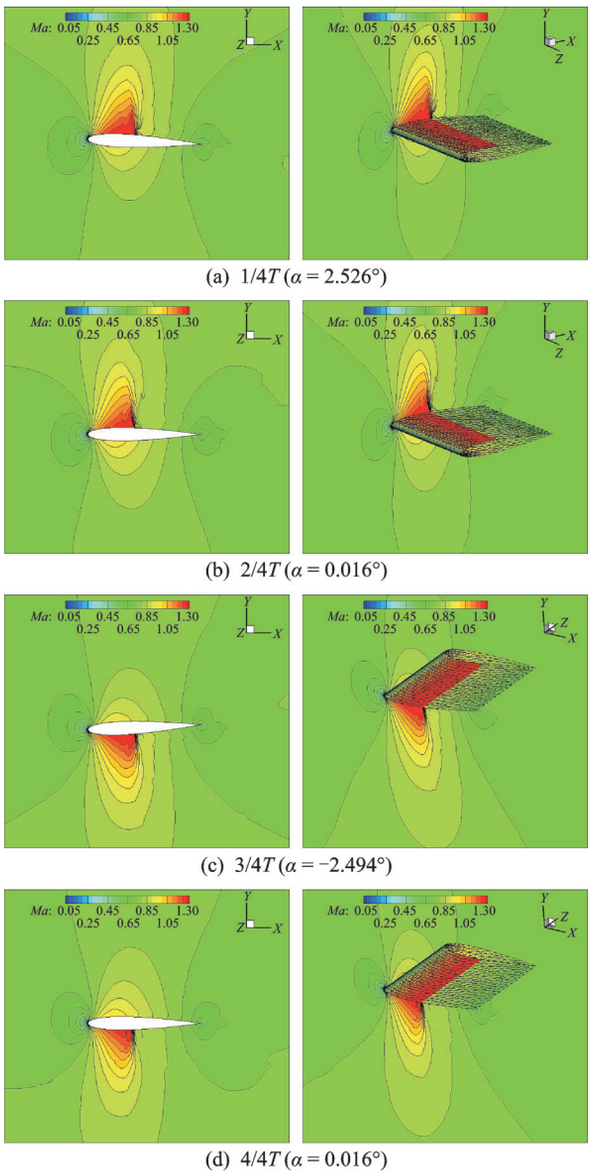
Fig.7 displays the global view and the local view of the initial mesh, for which there are 20 312 elements in total. Fig.8 gives the contours of Mach number obtained on the initial mesh at different time steps during one periodic cycle. Note that significant numerical oscillation can be observed near the



(a) Global view

(b) Local view

Fig.7 Initial mesh of oscillating NACA0012 wing



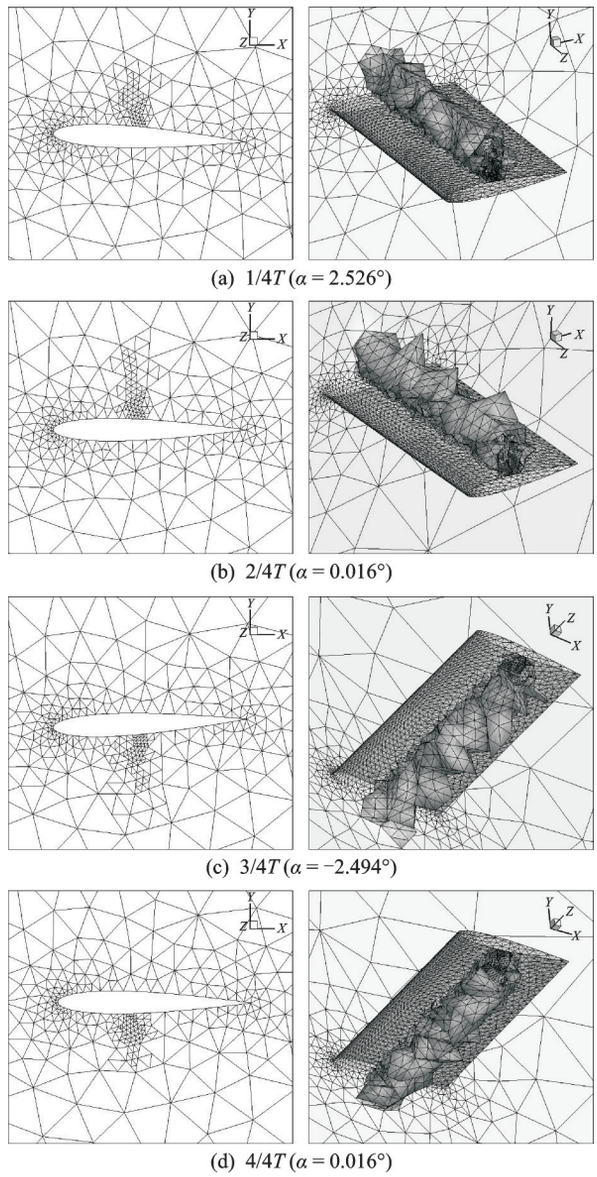
(a) $1/4T$ ($\alpha = 2.526°$)

(b) $2/4T$ ($\alpha = 0.016°$)

(c) $3/4T$ ($\alpha = -2.494°$)

(d) $4/4T$ ($\alpha = 0.016°$)

Fig.8 Mach contours computed on the initial mesh of oscillating NACA0012 wing



(a) $1/4T$ ($\alpha = 2.526°$)

(b) $2/4T$ ($\alpha = 0.016°$)

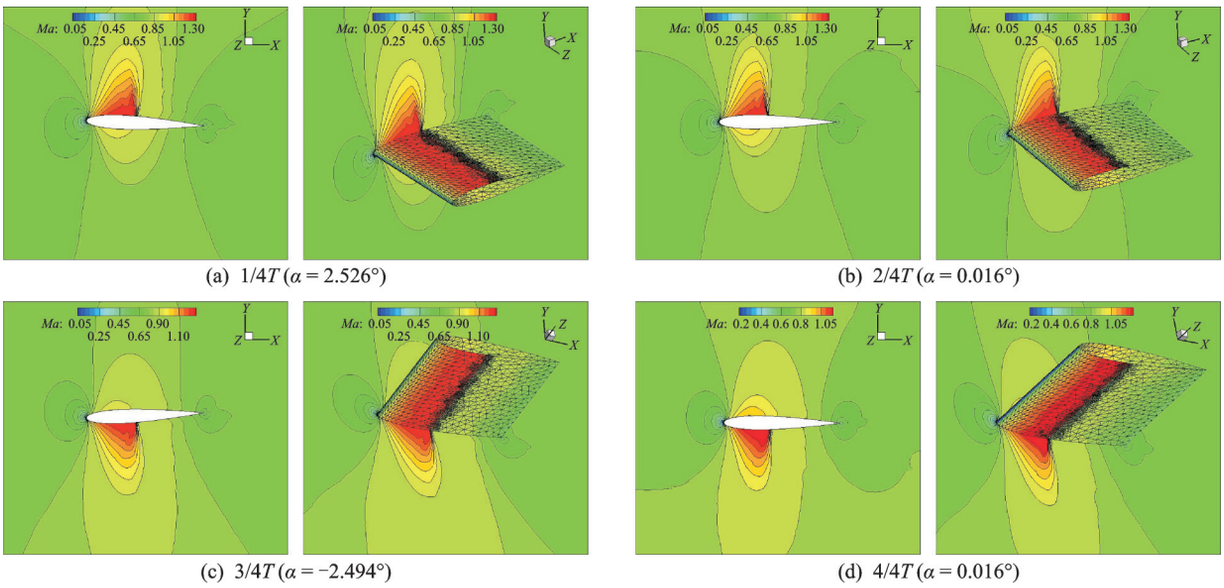
(c) $3/4T$ ($\alpha = -2.494°$)

(d) $4/4T$ ($\alpha = 0.016°$)

Fig.9 Refined mesh of oscillating NACA0012 wing

shocks, which means the numerical solution is not accurate enough. The mesh distributions with h-adaptivity during the pitching in one period cycle are presented in Fig.9, where the maximum time of refinement is limited as two. In Figs.9(a), (b), the elements on the upper surface of the wing near the shocks are refined. Figs.9(c), (d) display the mesh distributions when the shocks appear on the lower surface, for which the elements near the shocks are refined and the previously refined regions in Figs.9(a), (b) are coarsened. Fig.10 displays the Mach contours at different time steps. During the h-adaptivity, the total number of the elements over the entire computational domain remains less than 42 000 since the refining and the coarsening operations are conducted simultaneously.

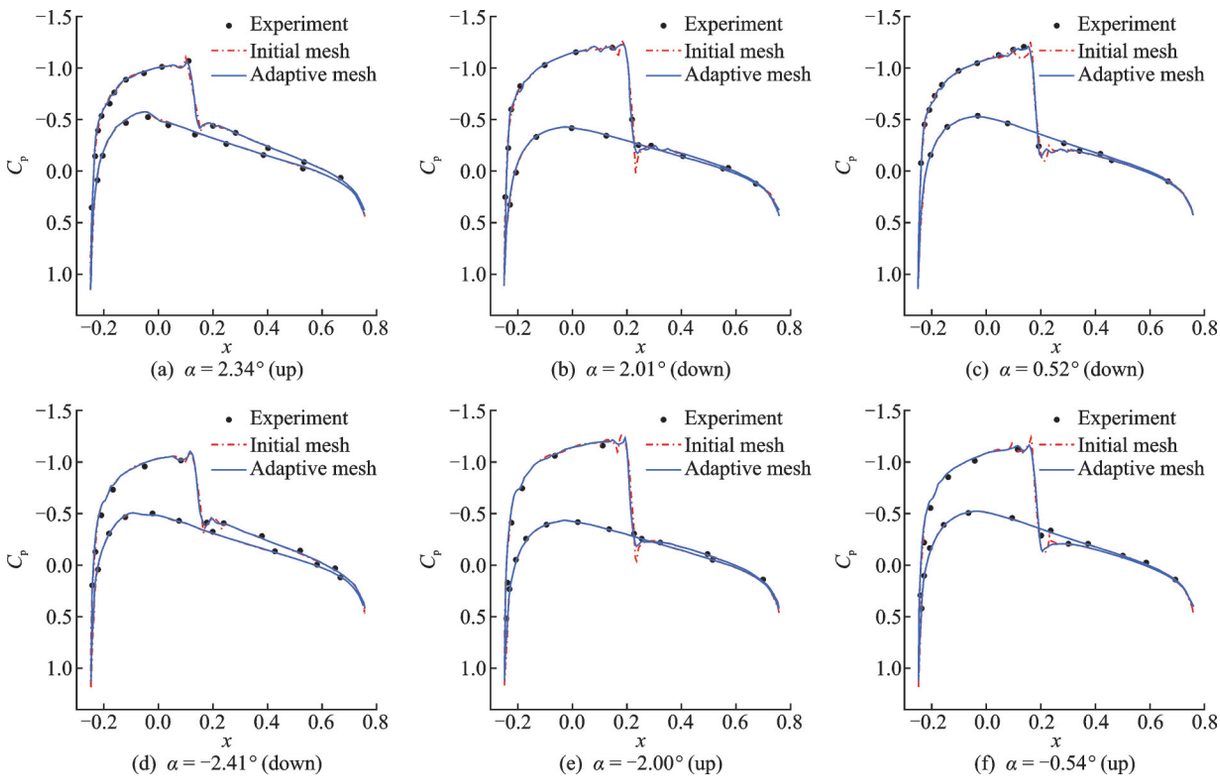
Fig.11 displays the distributions of the pressure coefficient on the $Z=0$ profile at different attack angles, where the results computed on the adaptive mesh match better with the experiments than those obtained on the initial mesh, and the numerical oscillations near the shocks are significantly reduced by the h-adaptivity. The comparison of the hysteresis loops is given in Fig.12, where the one computed on adaptive mesh is closer to the experimental one than the result obtained on the initial mesh.



(a) $1/4T$ ($\alpha = 2.526°$) (b) $2/4T$ ($\alpha = 0.016°$)

(c) $3/4T$ ($\alpha = -2.494°$) (d) $4/4T$ ($\alpha = 0.016°$)

Fig.10 Mach contours computed on the final mesh of oscillating NACA0012 wing



(a) $\alpha = 2.34°$ (up) (b) $\alpha = 2.01°$ (down) (c) $\alpha = 0.52°$ (down)

(d) $\alpha = -2.41°$ (down) (e) $\alpha = -2.00°$ (up) (f) $\alpha = -0.54°$ (up)

Fig.11 Comparisons of pressure coefficient profiles at different time steps

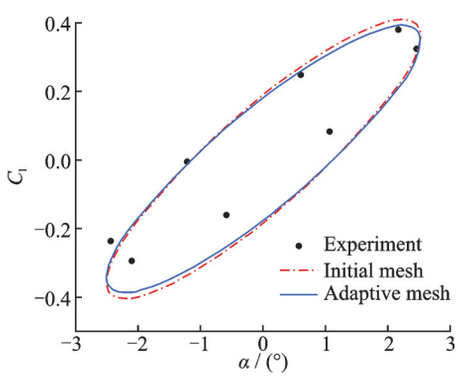


Fig.12 Comparison of hysteresis loops

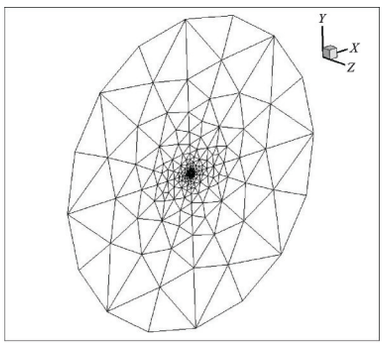
2.3 Viscous flow around sphere

For the viscous flows without shocks, we use the magnitude of vorticity as an h-adaptivity indicator. Here we use the 3rd-order DG ($p=2$) to solve the viscous flow around sphere, for which $Ma_\infty = 0.5$ and $Re = 118$. The governing equations are the N-S equations without the artificial viscosity term in Eq.(3).

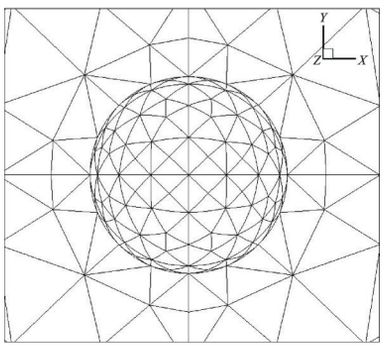
The initial mesh is given in Fig.13 which contains 2 412 elements. The steady-state flow field is

solved using the widely used Newton method. The pressure contours and the streamlines obtained on the initial mesh are displayed in Fig. 14(a), where the pressure distribution is not smooth enough. The distribution of the magnitude of the vorticity is given in Fig. 14(b), where the magnitude of the vorticity in the region near the solid surface and part of the wake zone is relatively large.

The initial mesh is refined one time according to the distribution of the vorticity. Figs. 15(a), (b) display the final mesh in the symmetric profile and in the spaces, respectively. The Mach contours in Fig. 16(a) are smooth and the accuracy of the pressure distribution is improved compared to that in Fig. 14(a).

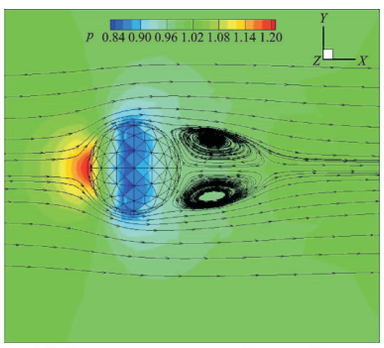


(a) Global view

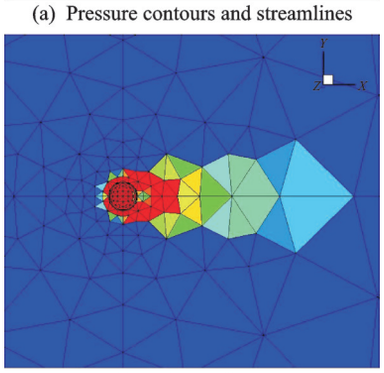


(b) Local view

Fig.13 Initial mesh for sphere

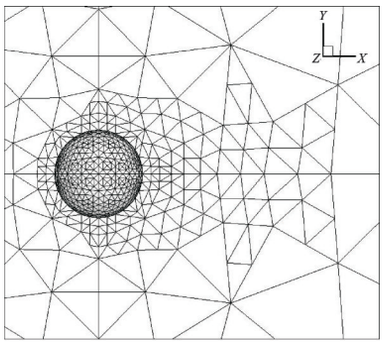


(a) Pressure contours and streamlines

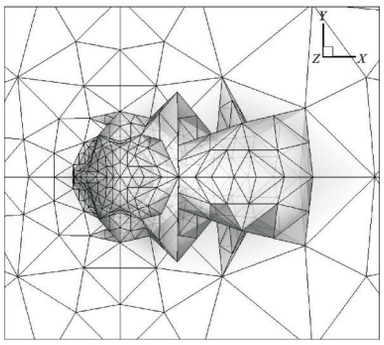


(b) Magnitude of vorticity

Fig.14 Numerical results on the initial mesh

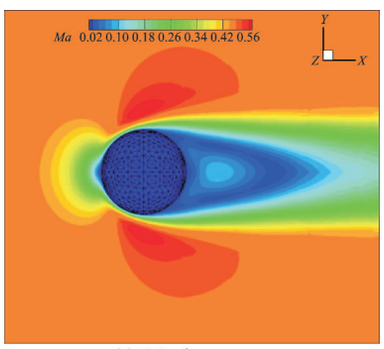


(a) Final mesh on symmetric profile

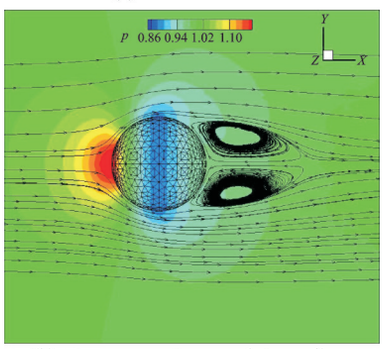


(b) Refined mesh in space

Fig.15 The final mesh



(a) Mach contours



(b) Pressure contours and streamlines

Fig.16 Numerical results on the final mesh

3 Conclusions

(1) An h-adaptive DG method on locally curved tetrahedral mesh is developed, which contains both refining and coarsening operations. The curved geometry is well maintained during both refining and coarsening, which ensures the applicability of the obtained mesh to high-order DG methods.

(2) To ensure the parallel efficiency, a re-balance strategy for parallel computing is conducted after the h-adaptivity, which can provide a nearly equivalent computational load on different processors. However, one possible disadvantage of the re-balance method is that it may be time-consuming when the number of elements is large since only one processor is used to conduct the h-adaptivity and the re-balancing.

(3) Both steady-state and unsteady-state cases are numerically tested. Numerical results indicate that the accuracy can be significantly improved with the introduced h-adaptive method at a relatively low expense since only local regions are refined according to the given h-adaptivity indicators. Furthermore, the choice of the h-adaptivity indicators is flexible.

Typical test cases are numerically computed to validate the introduced h-adaptive DG method. In the future, we should consider further improving the efficiency of the h-adaptivity and the re-balancing operation for more complex computational fluid dynamics (CFD) problems, such as the large eddy simulation (LES) for complex geometries.

References

[1] DWYER H A, KEE R J, SANDERS B R. Adaptive grid method for problems in fluid mechanics and heat transfer[J]. AIAA Journal, 2012, 1(10): 1205-1212.

[2] SALTZMAN J, BRACKBILL J. Applications and generalizations of variational methods for generating adaptive meshes[J]. Applied Mathematics and Computation, 1982, 10: 865-884.

[3] BERGER M J, OLIGER J. Adaptive mesh refinement for hyperbolic partial differential equations[J]. Journal of Computational Physics, 2009, 53(3): 484-512.

[4] LÖHNER R, MORGAN K, ZIENKIEWICZ O C. Adaptive grid refinement for the compressible Euler equations [C]//Proceedings of Accuracy Estimates and Adaptive Refinements in Finite Element Computations. Chichester, UK: Wiley-Intersci Publ, 1986: 281-297.

[5] CARPENTER J G, MCRAE V D. Adaption of unstructured meshes using node movement[C]//International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields. Starkville, USA: Mississippi State Univ Press, 1996.

[6] MA Z, CHEN H, ZHOU C. A study of point moving adaptivity in gridless method[J]. Computer Methods in Applied Mechanics and Engineering, 2008, 197(21/22/23/24): 1926-1937.

[7] BABUSKA I, RHEINBOLDT W, MESZTENYI C. Self-adaptive refinements in the finite element method: ORO-3443-53[R]. College Park: Maryland University, 1975.

[8] LENTINI M, PEREYRA V. An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers[J]. SIAM Journal on Numerical Analysis, 1977, 14(1): 91-111.

[9] PERAIRE J, PEIRO J, MORGAN K. Adaptive remeshing for three-dimensional compressible flow computations[J]. Journal of Computational Physics, 1992, 103(2): 269-285.

[10] DOMPIERRE J, VALLET M G, FORTIN M, et al. Anisotropic mesh adaptation—Towards a solver and user independent CFD[C]//Proceedings of the 35th Aerospace Sciences Meeting and Exhibit. Reston, USA: AIAA, 1997: 861.

[11] LEPAGE C, SUERICH-GULICK F, HABASHI W. Anisotropic 3-D mesh adaptation on unstructured hybrid meshes[C]//Proceedings of the 40th AIAA Aerospace Sciences Meeting & Exhibit. Reston, USA: AIAA, 2002: 859.

[12] SZABO B A. Estimation and control of error based on p-convergence[M]//Accuracy Estimates and Adaptive Refinements in Finite Element Computations. New York: Wiley, 1986: 61-78.

[13] BABUŠKA L, SURI M. The optimal convergence rate of the p-version of the finite element method[J]. SIAM Journal on Numerical Analysis, 1987, 24(4): 750-776.

[14] TAMMA K K, SAW K C. Hierarchical p-version finite elements and adaptive a posteriori computational

formulations for two-dimensional thermal analysis[J]. Computers & Structures, 1989, 32(5): 1183-1194.

[15] SCHWAB C. P-and hp-finite element methods: Theory and applications in solid and fluid mechanics[M]. Oxford: Oxford University Press, 1998.

[16] RAVICHANDRAN R, SALAGAME R. Error estimation and adaptivity in three-dimensional p-version finite element analysis[C]//Proceedings of the 41st Structures, Structural Dynamics, and Materials Conference and Exhibit. Reston, USA: AIAA, 2000: 1353.

[17] PROMWUNGKWA A. Data structure and error estimation for an adaptive p-version finite element method in 2-D and 3-D solids[D]. Blacksburg, Virginia: Virginia Polytecnic Institute and Sate University, 1998.

[18] LENTINI M, PEREYRA V. An adaptive finite difference solver for nonlinear two-point boundary problems with mild boundary layers[J]. SIAM Journal on Numerical Analysis, 1977, 14(1): 91-111.

[19] JONES W, NIELSEN E, PARK M. Validation of 3D adjoint based error estimation and mesh adaptation for sonic boom prediction[C]//Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit. Reston, USA: AIAA, 2006: 1150.

[20] BALASUBRAMANIAN R, NEWMAN III J C. Comparison of adjoint-based and feature-based grid adaptation for functional outputs[J]. International Journal for Numerical Methods in Fluids, 2007, 53(10): 1541-1569.

[21] DWIGHT R P. Goal-oriented mesh adaptation for finite volume methods using a dissipation-based error indicator[J]. International Journal for Numerical Methods in Fluids, 2008, 56(8): 1193-1200.

[22] FIDKOWSKI K J. A Simplex cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations[D]. Massachusetts: Massachusetts Institute of Technology, 2007.

[23] LANDMANN B, KESSLER M, WAGNER S, et al. A parallel, high-order discontinuous Galerkin code for laminar and turbulent flows[J]. Computers & Fluids. 2008, 37(4): 427-438.

[24] LANDMANN B. A parallel discontinuous Galerkin code for the Navier-Stokes and Reynolds-averaged Navier-Stokes equations[D]. Stuttgart: University of Stuttgart, 2008.

[25] OLIVER T A. A high-order, adaptive, discontinuous Galerkin finite element method for the Reynolds-averaged Navier-Stokes equations[D]. Massachusetts: Massachusetts Institute of Technology, 2008.

[26] Burge S S N K. An adaptive discontinuous Galerkin solver for aerodynamic flows[D]. Wyoming: University of Wyoming, 2011.

[27] BURGESS N, MAVRIPLIS D. Robust computation of turbulent flows using a discontinuous Galerkin method[C]//Proceedings of the 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Nashville, Reston, USA: AIAA, 2012: 457.

[28] BURGESS N, NASTASE C, MAVRIPLIS D. Efficient solution techniques for discontinuous Galerkin discretizations of the Navier-Stokes equations on hybrid anisotropic meshes[C]//Proceedings of the 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition. Reston, USA: AIAA, 2010: 1448.

[29] LU H Q, SUN Q. A straightforward hp-adaptivity strategy for shock-capturing with high-order discontinuous Galerkin methods[J]. Adv Appl Math Mech, 2014, 6(1): 135-144.

[30] WU D, YU X J. Adaptive discontinuous Galerkin method for Euler equations[J]. Chinese Journal of Computational Physics, 2010, 27(4): 492-500.

[31] BASSI F, CRIVELLINI A S, et al. Discontinuous Galerkin solution of the Reynolds-averaged Navier-Stokes and k-ω turbulence model equations[J]. Computers & Fluids, 2005, 34(4/5): 507-540.

Acknowledgements This work was supported by the funding of the Key Laboratory of Aerodynamic Noise Control (No.ANCL20190103), the State Key Laboratory of Aerodynamics (No.SKLA20180102), the Aeronautical Science Foundation of China (Nos.2018ZA52002, 2019ZA052011), the National Natural Science Foundation of China (Nos. 61672281, 61732006).

Authors Dr. **AN Wei** received the B.S. degree in aircraft design and engineering from Nanjing University of Aeronautics and Astronautics (NUAA) in 2017. He is currently an doctoral candidate majoring in fluid mechanics of NUAA. His research domain is the discontinuous Galekin method.

Prof. **LYU Hongqiang** is a Ph.D. tutor in College of Aerospace Engineering of NUAA. His research interests are computational fluid dynamics, artificial intelligence application, computational electromagnetism, and aerodynamic noise.

He has published more than 30 academic papers on well-known domestic and foreign journals.

Author contributions Prof. **LYU Hongqiang** designed the whole study. Dr. **AN Wei** performed the simulation, and wrote the manuscript. Dr. **AN Wei** and Mr. **HUANG Zenghui** performed the result investigation and figure generation.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Tong)

基于局部弯曲四面体网格的可压流自适应高阶间断伽辽金方法

安 慰[1,2,3], 黄增辉[1], 吕宏强[1]

(1.南京航空航天大学航空学院,南京 210016,中国;
2.中国空气动力研究与发展中心,气动噪声控制重点实验室,绵阳 621000,中国;
3.中国空气动力研究与发展中心,空气动力学国家重点实验室,绵阳 621000,中国)

摘要:对于可压缩流的数值模拟,通常需要在不同的计算区域采用不同尺度的网格单元。在诸如激波、附面层等物理量变化剧烈的区域通常需要小尺度的网格单元来提高数值模拟的精度,在物理量相对光滑的区域采用较大尺度的网格单元即可。虽然对于复杂流动自适应网格已被广泛应用,但对于高阶间断伽辽金方法仍然存在两个困难。首先,高阶间断伽辽金方法需要的局部弯曲网格显著提升了网格单元细分和粗化的难度,因为弯曲的信息需要始终被保持。其次,对于并行计算,网格自适应会破坏分区负载平衡,从而导致并行效率显著下降。本文基于局部弯曲的四面体网格针对高阶间断伽辽金方法发展了一种鲁棒、自动的网格自适应方法,在自适应的过程中物面的弯曲信息可以被很好地保持。另外在每次网格自适应后,采用了重新分配分区负载的策略保证并行效率。数值结果表明,所提出的网格自适应方法能够根据流场结构生成更合理的网格。

关键词:网格自适应;间断伽辽金方法;弯曲网格;四面体网格;可压缩流