Journal of Computational Physics 229 (2010) 1681–1697

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A robust high-order residual distribution type scheme for steady Euler equations on unstructured grids

Guanghui Hu [a], Ruo Li [b], Tao Tang [a,*]

[a] Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
[b] CAPT, LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China

ARTICLE INFO

Article history:
Received 24 December 2008
Received in revised form 7 October 2009
Accepted 2 November 2009
Available online 10 November 2009

Keywords:
Steady Euler equations
High-order accuracy
Multigrid
Block LU-SGS
WENO
Hierarchical reconstruction

ABSTRACT

A robust high-order algorithm is proposed to solve steady Euler equations on unstructured grids. The main ingredients of the algorithm include a standard Newton method as the outer iterative scheme and a linear multigrid method as the inner iterative scheme with the block lower-upper symmetric Gauss–Seidel (LU-SGS) iteration as its smoother. The Jacobian matrix of the Newton-iteration is regularized by the local residual, instead of using the commonly adopted time-stepping relaxation technique based on the local CFL number. The local Jacobian matrix of the numerical fluxes are computed using the numerical differentiation, which can significantly simplify the implementations by comparing with the manually derived approximate derivatives. The approximate polynomial of solution on each cell is reconstructed by using the values on centroid of the cell, and limited by the WENO hierarchical limiting strategy proposed by Xu et al. [Z.L. Xu, Y.J. Liu, C.W. Shu, Hierarchical reconstruction for discontinuous galerkin methods on unstructured grids with a WENO-type linear reconstruction and partial neighboring cells, Journal of Computational Physics 228 (2009) 2194–2212]. It is found that the proposed algorithm is insensitive to the parameters used. More precisely, in our computations, only one set of the parameters (namely, the proportional constant α for the local residual, the relaxation parameter τ in the Newton-iteration, the weight μ in the WENO scheme and the number of smoothing steps in the multigrid solver) is employed for various geometrical configurations and free-stream configurations. The high-order and robustness of our algorithm are illustrated by considering two-dimensional airfoil problems with different geometrical configurations and different free-stream configurations.

1. Introduction

One of the major research areas of computational aerodynamics is the numerical simulation of the complex flow field with practical configurations. Many numerical methodologies have been developed for the simulations based on the finite difference method, the finite volume method and the finite element method. All these methods can be used to discretize the Euler or the Navier–Stokes equations, which may result in highly nonlinear algebraic systems. Developing robust and efficient solvers for the nonlinear systems is a main challenge in the numerical simulation. Many recent solvers have been developed to enhance the numerical efficiency, see, e.g. [6,30,41,22] and reference therein. These solvers have used some modern numerical techniques, including multigrid acceleration [2,7,8,18], post-residual smoothing [14,15,20] and enthalpy damping [17]. With these acceleration techniques, the nonlinear systems can be solved with several multigrid iterations (a

* Corresponding author. Tel.: +852 3411 5148; fax: +852 3411 5811.
E-mail addresses: ghhu@math.hkbu.edu.hk (G. Hu), rli@math.pku.edu.cn (R. Li), ttang@math.hkbu.edu.hk (T. Tang).

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.11.002

few seconds) to machine accuracy. An overview of these techniques can be found in [29]. However, most of known solvers have only second-order accuracy, while the experience on the structured grids suggests that higher-order accurate approximations can significantly improve the quality of numerical solutions [3]. Recently many efforts have been made for developing high-order methods on the unstructured grids.

On unstructured grids, the situation is more complex than that on the structured grids. In spite of this, many high-order methods have been presented on unstructured grids. Barth [3] developed the concept of *k-exact* reconstruction scheme for the median-dual finite volume scheme. Such technique was extended to the cell-centered finite volume scheme by Mitchell and Walters [32]. Based on the idea of selecting the locally smoothest stencil, ENO/WENO schemes [21,25,35] are proposed for the hyperbolic system of conservation laws. Recently, discontinuous Galerkin (DG) schemes [11] and spectral volume (SV) methods [36,40] have been developed; both employ the information inside the control volume. All these numerical schemes demonstrated excellent high-order behaviors.

In this paper, we present a robust high-order numerical algorithm for two-dimensional steady Euler equations on unstructured grids, based on the work done by Li et al. [24]. In [24], only linear reconstruction is used, so the numerical solution has just second-order accuracy (for smooth solutions). To obtain higher order numerical accuracy, we extend the reconstruction to the quadratic case, and use hierarchical reconstruction strategy of [27,26,42] to avoid the non-physical oscillations. The idea of hierarchical reconstruction is simple. After the initial reconstruction, the information on the centroid of each cell such as the function values, gradients and second-order derivatives is known. What the hierarchical reconstruction does is to recompute all the information level by level from the highest order terms to the lowest order terms, with certain non-oscillatory method. For the quadratic reconstruction which gives a quadratic polynomial in each cell, for example, coefficients of second-order terms are recomputed using constant term and gradients in different reconstruction patch, and several candidates for these coefficients are generated. Then the MUSCL scheme or ENO/WENO schemes are used to determine the limited coefficients for the second-order terms. For the linear terms, a similar way is used to generate limited coefficients with constant term and the updated second-order terms of the polynomial. In this paper, we adopt the WENO type hierarchical reconstruction to limit the approximate polynomial in each cell. The numerical results in the final section show that such limiting strategy works very well: the high-order numerical accuracy is kept in the smooth region, and at the mean time, the spurious oscillations nearby the shock region are removed or reduced significantly.

Different from the standard *k-exact* reconstruction which the approximate polynomial on each cell are generated based on cell averages, the approximate polynomial in this paper is given based on the functional value on the centroid of each cell. Then the polynomial is used directly to describe the steady-state of Euler equations in an integrated form. In fact, the scheme proposed in this paper fall into the category of the so-called *Residual Distribution* schemes, which have been extensively discussed in the literature [12,1,10]. Residual distribution schemes start with point values of the solution, then an interpolation to obtain a polynomial approximation of the solution, then an approximation of the residual resulting from an integrated version of the steady-state PDE (which, by the divergence theorem, involves only integrals of the flux along the cell boundaries), then a pseudo-time or another method to compute the resulting nonlinear system. The scheme in this paper differs from traditional residual distribution schemes only in the last step, namely the nonlinear system is solved directly by the Newton type method rather than through residual distribution and pseudo-time marching. We point out that our algorithm has almost the same efficiency as finite volume schemes on the unstructured mesh case, since for both schemes, a least square system (5×5 for our scheme and 6×6 for finite volume schemes) is solved in every cell in each Newton-iteration step.

The numerical algorithm to be presented in this paper is as follows. On unstructured grids, we first discretize the steady Euler equations to obtain the nonlinear algebraic system using residual distribution schemes, which is then linearized by using the standard Newton-iterations. In the steady case, this algebraic system is actually singular without the time evolution term and require to be regularized. In the standard methods, the system is regularized by adding a local artificial time relaxation. In our algorithm we use the term $\alpha\|RHS\|_1$ to regularize the system instead of using the artificial time relaxation, where *RHS* is the residual of the linearized system on each cell, and α is a positive constant.

Since the Newton-iteration is employed to linearize the system, we may need the differential forms for the flux to obtain the local Jacobian matrix. Note that the flux itself may be very complex, such as HLLC [37], CUPS [19], etc., so it is difficult to obtain the differential forms of these fluxes and to use them correctly in the implementations. In this paper, we use numerical differentiation rather than the exact formulas. From our numerical experience, the numerical differentiation can simplify the implementation significantly. We just use the exact forms of the flux and one simple C++ function to get the desired results. In the last section, a number of numerical experiments demonstrate the reliability of the numerical differentiation method.

The resulting regularized linear system for the Newton-iteration is solved by a linear multigrid method in which the block LU-SGS proposed by Wang [38] is utilized as the smoother. In implementing the multigrid methods, a sequence of coarse meshes is generated following the idea of [6]. The coarse meshes are generated to construct the projection operator from fine mesh to coarse mesh, applied on the sparse matrix and the right hand side of the regularized linear system. The standard *V*-cycle type multigrid iteration is used in our implementation.

The rest of this paper is organized as follows. In the next section, the numerical discretization for the Euler equations will be described. In Sections 3 and 4, the high-order reconstruction technique and the limiting strategy will be introduced. Then the Newton-iteration and the multigrid method for solving the corresponding nonlinear system will be discussed in Sections 5 and 6, respectively. Numerical experiments will be carried out in Section 7.

2. Numerical discretization for the Euler equations

The compressible inviscid Euler equations can be written as

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

where U is the vectors of the conservative variables and F is the inviscid flux. We consider the ideal flows with U and F defined as

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{bmatrix},\quad F(U)=\begin{bmatrix}\rho u & \rho v\\ \rho u^2+p & \rho uv\\ \rho uv & \rho v^2+p\\ u(E+p) & v(E+p)\end{bmatrix}, \tag{2}$$

where ρ is the density, $\mathbf{u}=(u,v)$ is the velocity, p is the pressure, and E is the total energy. Finally we close the system by using the equation of state

$$E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2),$$

where $\gamma=1.4$ is the ratio of specific heats for air.

For stationary solutions, we actually solve Eq. (1) without the time evolving term, namely

$$\nabla\cdot F(U)=0. \tag{3}$$

In this paper, we solve (3) in domain $\Omega:=\mathbb{R}^2-\Omega_c$, where Ω_c denotes the domain occupied by the airfoil, which is the body of the aircraft in the two-dimensional case. Since the problem domain is unbounded, the commonly used strategy is to solve the problem in the domain $\bar{\Omega}:=\Omega\bigcap\{|x|<R\}$ and adopt the far field vortex correction technique to remedy the error introduced by the abrupt domain truncation.

We choose cell-centered scheme to discretize the continuous equation. Let $\mathcal{T}$ be a triangular partition of $\bar{\Omega}$, and $\mathcal{K}\in\mathcal{T}$ be one cell in the partition. We assume the intersection of two different cells can only be either an edge or a vertex. Let n_{ij} denote the outer unit normal on the edge e_{ij}, pointing from $\mathcal{K}_i$ to $\mathcal{K}_j$. Applying the Gauss's formula to (3) gives

$$\oint_{\partial\mathcal{K}_i}F(U)\cdot n dl=0, \tag{4}$$

where n is the unit out normal of $\partial\mathcal{K}_i$. By introducing the numerical flux, (4) can be approximated as

$$\oint_{\partial\mathcal{K}_i}F(U)\cdot n dl\approx\sum_{e_{ij}\in\partial\mathcal{K}_i}\int_{e_{ij}}\overline{F}(U_i,U_j)\cdot n_{ij}dl=0, \tag{5}$$

where $\overline{F}(U_i,U_j)$ denotes the numerical flux. In numerical simulations, HLLC flux [5] is used.

In order to achieve high-order accuracy, we need to reconstruct a high order polynomial on each cell to describe the variation of solution. There are already many papers about the high-order reconstruction, the classical one is the *k-exact* reconstruction given by Barth [3]. In next section, we use the similar least square method to reconstruct the polynomial, and limit it by using the WENO hierarchical reconstruction method.

3. High order reconstruction

In this paper, we use the quadratic reconstruction as an example to describe the procedure of reconstruction. Note that the reconstruction is performed on the primitive variables.

Expanding the solution $\widetilde{U}$ at the centroid of control volume $\mathcal{K}$ by using Taylor expansion, and truncating after the quadratic term, we get the following approximate polynomial solution:

$$h_q(x,y)=a_0+a_1(x-x_0)+a_2(y-y_0)+\frac{1}{2}a_3(x-x_0)^2+a_4(x-x_0)(y-y_0)+\frac{1}{2}a_5(y-y_0)^2. \tag{6}$$

Note that $a_0=\widetilde{U}(x_0,y_0)$ is the function value at the centroid of cell, our goal is to find out $a_1=\widetilde{U}_x(x_0,y_0)$, $a_2=\widetilde{U}_y(x_0,y_0)$, $a_3=\widetilde{U}_{xx}(x_0,y_0)$, $a_4=\widetilde{U}_{xy}(x_0,y_0)$ and $a_5=\widetilde{U}_{yy}(x_0,y_0)$.

The method we used for obtaining above unknowns is the least square method. The detail is as the follows. First, we choose the reconstruction patch for each cell. In this paper, we emphasize on the difference between linear reconstruction and the quadratic reconstruction, and follow [34] to give the linear and quadratic patch, respectively, with a modification to the quadratic case. For the linear case, we use the following criterion to choose the reconstruction patch: for certain cell $\mathcal{K}$, any cell which has one common edge with $\mathcal{K}$ is selected as one component of the patch $\mathcal{P}_l(\mathcal{K})$. Fig. 1 (left one) gives an illustration of $\mathcal{P}_l(\mathcal{K})$. For the quadratic case, we need more cells in the reconstruction patch, so the following criterion is used: any cell which has at least one common vertex with $\mathcal{K}$ is selected as one component of the patch $\mathcal{P}_q(\mathcal{K})$. Fig. 1 (right one) gives an

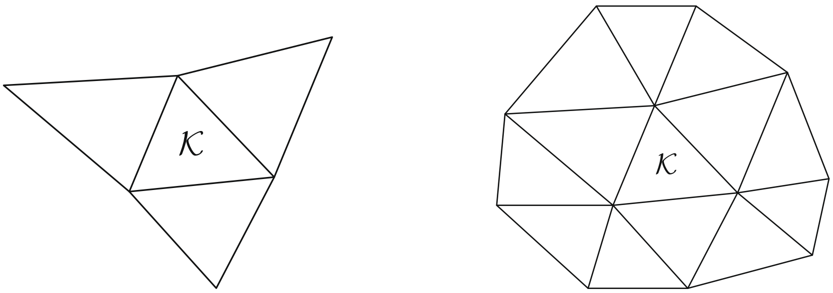

Fig. 1. The reconstruction patches $\mathcal{P}(\mathcal{K})$ used for the least square reconstruction. Left: linear case. Right: quadratic case.

illustration of $\mathcal{P}_q(\mathcal{K})$. In the numerical experiments, the criterion for quadratic case can always give enough cells (at least 6 cells for quadratic case) in the patch of each cell, even for cells nearby the boundary. However, the criterion for linear case works not well for the cell which is adjacent to the boundary, say the boundary cell just has two neighbors. To enlarge the reconstruction patch for boundary cells, the criterion for quadratic case is adopted for such cells.

We obtain the unknowns $a_1,\ a_2,\ a_3,\ a_4$ and a_5 by solving the following minimization problem:

$$\min_{a_1,a_2,a_3,a_4,a_5} \sum_{\forall \mathcal{K}_l \in \mathcal{P}_q(\mathcal{K})} \left\| a_{0,l} - a_0 - a_1 x_{0l} - a_2 y_{0l} - \frac{1}{2} a_3 x_{0l}^2 - a_4 x_{0l} y_{0l} - \frac{1}{2} a_5 y_{0l}^2 \right\|_2^2, \tag{7}$$

where $a_{0,l}$ means the solution value of at the centroid $(x_l, y_l) \in \mathcal{K}_l$, and $x_{0l} = x_l - x_0,\ y_{0l} = y_l - y_0$.

For the smooth solution, we can use the above approximate polynomial directly to calculate the numerical flux. But for the solution with discontinuity, if we do nothing to the polynomial, the numerical scheme will introduce non-physical oscillations nearby the discontinuity region. So we need to limit the polynomial in order to improve the behavior of numerical scheme. In the linear case, many good limiters have been proposed. But for higher order schemes, it seems not easy to give a limiting strategy which can cancel the spurious oscillations and also keep the high-order accuracy. In [13], two different limiter functions are designed for the linear and quadratic terms, respectively, and satisfactory numerical results are obtained. Recently, Liu et al. presented a series of papers [26,27,42] to describe a new strategy, called hierarchical reconstruction, to limit high-order polynomials. The hierarchical reconstruction limits each term of the polynomial in a unified way, while makes the implementation easier. In the next section, we will introduce a method to limit (6) following the idea of the hierarchical reconstruction.

Besides the high-order reconstruction, the proper approximation of the curved boundary is also a critical issue for keeping high-order numerical accuracy. For the linear reconstruction, the polygon is used to approximate the computational domain. For the quadratic reconstruction or higher order reconstruction, such crude approximation is not usable anymore. In [4], Bassi and Rebay used the standard second-order geometric transformation for the curved boundary element. Total six nodes (three vertexes and three midpoints of edges) are used to determine the quadratic transformation. Since there is just one curved edge in each cell adjacent to the boundary, Wang and Liu [39] gave a simpler approach. For example, only four nodes (three vertexes and the midpoint of the curved edge) are required to determine uniquely a quadratic transformation. Recently, Luo et al. [28] just adjusted the outer normal direction of quadrature point on the curved boundary to get the desired numerical accuracy.

It is noted that the method used in [28] is much simpler than others: no geometric transformation is needed, and just the outer normal directions of quadrature points on the boundary are adjusted. We adopt this strategy in the experiments. Note that the Gauss integral formula is not used for the line integral along the curved boundary; instead the Simpson formula is used with the quadrature points of the two vertices and the midpoint of the edge. With these three quadrature points, third-order accuracy can be kept, which is sufficient for the quadratic reconstruction. With the Simpson formula, only one point value (the midpoint of the edge) needs to be found out. With the Simpson formula, the stable convergence order is obtained in the numerical experiment.

For the edge not on the boundary, we use the two-point Gauss integral formula for both the linear and quadratic cases.

4. The WENO hierarchical limiting strategy

In the hierarchical limiting strategy, several candidates for certain term in the polynomial will be generated. In [26,27], the non-oscillatory MUSCL method or second-order ENO method is adopted to determine the last term from these candidates. Since the inherent non-differentiability of the ENO schemes, convergence of the solution to steady-state is impossible [31]. In this paper, we focus on the steady-state of the Euler equations, so we follow [42] where the WENO method is used in the hierarchical limiting strategy, to limit (6).

For each cell $\mathcal{K}$, the information on the centroid of the cell including the functional value (a_0 in (6)), gradients (a_1 and a_2 in (6)), second order derivatives ($a_3,\ a_4$ and a_5 in (6)) is known. The idea of hierarchical limiting method is to recompute these quantities level by level from the highest order to the lowest order. We recompute the second-order information first for each cell in the computational domain.

For each cell $\mathcal{K}_i \in \mathcal{P}_q$, the following minimization problem is solve

$$\min_{\tilde{a}_3,\tilde{a}_4,\tilde{a}_5} \sum_{\forall \mathcal{K}_l \in \mathcal{P}_q(\mathcal{K}), l \neq i} \left\| a_{0,l} - a_0 - a_1 x_{0l} - a_2 y_{0l} - \frac{1}{2}\tilde{a}_3 x_{0l}^2 - \tilde{a}_4 x_{0l} y_{0l} - \frac{1}{2}\tilde{a}_5 y_{0l}^2 \right\|_2^2, \tag{8}$$

where $\tilde{a}_3,\ \tilde{a}_4$ and $\tilde{a}_5$ are the candidates of coefficients of second-order terms on the cell $\mathcal{K}_i$.

After looping all cells in the computational domain, each cell $\mathcal{K}$ has several candidates for second-order information. All these candidates are combined to give the final second-order information in $\mathcal{K}$ with the WENO procedure. For example, assume that there are candidates $\tilde{a}_{3,j},\ \tilde{a}_{4,j}$ and $\tilde{a}_{5,j},\ j = 1, \ldots, m$, in the cell $\mathcal{K}$. First, the smoothness indicator is introduced [16] for each candidate

$$S_{q,j} = \int_{\mathcal{K}} |\mathcal{K}| (\tilde{a}_{3,j}^2 + \tilde{a}_{4,j}^2 + \tilde{a}_{5,j}^2) d\Omega, \tag{9}$$

where $|\mathcal{K}|$ is the area of the cell $\mathcal{K}$. Then the weight of each candidate is given as

$$\omega_j = \frac{\tilde{\omega}_j}{\sum_i \tilde{\omega}_i}, \quad \tilde{\omega}_i = \frac{1}{(\epsilon + S_{q,i})^{\mu}}, \tag{10}$$

where $\epsilon = 10^{-4}$ and $\mu = 1$. Finally, the second-order information in the cell $\mathcal{K}$ is presented as

$$\tilde{\tilde{a}}_i = \sum_{j=1}^{m} \omega_j \tilde{a}_{i,j}, \quad i = 3, 4, 5. \tag{11}$$

The linear information in each cell $\mathcal{K}_i$ is also reconstructed in a similar way. For obtaining candidates of coefficients of linear term $\tilde{a}_1$ and $\tilde{a}_2$, the following minimization problem is solved

$$\min_{\tilde{a}_1,\tilde{a}_2} \sum_{\forall \mathcal{K}_l \in \mathcal{P}_q(\mathcal{K}), i \neq l} \left\| a_{0,l} - a_0 - \tilde{a}_1 x_{0l} - \tilde{a}_2 y_{0l} - \frac{1}{2}\tilde{\tilde{a}}_3 x_{0l}^2 - \tilde{\tilde{a}}_4 x_{0l} y_{0l} - \frac{1}{2}\tilde{\tilde{a}}_5 y_{0l}^2 \right\|_2^2. \tag{12}$$

In the WENO procedure, the following smoothness indicator is used for the linear part

$$S_{l,j} = \int_{\mathcal{K}} (\tilde{a}_{1,j}^2 + \tilde{a}_{2,j}^2) d\Omega. \tag{13}$$

The coefficients of linear term in each cell is determined with the same way to the quadratic case

$$\tilde{\tilde{a}}_i = \sum_{j=1}^{m} \omega_j \tilde{a}_{i,j}, \quad i = 1, 2, \tag{14}$$

where ω_j depends on $S_{l,j}$.

Finally, the approximate polynomial in the cell $\mathcal{K}$ after the limiting strategy is

$$h_q(x,y) = a_0 + \tilde{\tilde{a}}_1(x - x_0) + \tilde{\tilde{a}}_2(y - y_0) + \frac{1}{2}\tilde{\tilde{a}}_3(x - x_0)^2 + \tilde{\tilde{a}}_4(x - x_0)(y - y_0) + \frac{1}{2}\tilde{\tilde{a}}_5(y - y_0)^2. \tag{15}$$

The convergence of the WENO hierarchical limiting strategy is studied by using the smooth function $f = \sin(\pi x)\cos(2\pi y)$ on the domain $[0,1] \times [0,1]$. The mesh used is generated by EasyMesh [33]. Table 1 gives L_2 errors on six successively refined meshes, and the expected convergence order is observed.

Table 1
L_2 error and convergence order of quadratic reconstruction with hierarchical limiting strategy.

No. of cells	L_2 error	Order
64	4.99e−02	
256	9.36e−03	2.4
1024	1.34e−03	2.8
4096	1.69e−04	3.0
16,384	2.11e−05	3.0
65,536	2.65e−06	3.0

5. Newton-iteration

We use Newton-iteration to linearize the nonlinear system (5):

$$\sum_{e_{ij}\in\partial\mathcal{K}_i}\int_{e_{ij}}\overline{F}^{(n)}\cdot n_{ij}dl+\sum_{e_{ij}\in\partial\mathcal{K}_i}\int_{e_{ij}}\left(\frac{\partial\overline{F}^{(n)}}{\partial U_i}\delta U_i^{(n)}\right)\cdot n_{ij}dl+\sum_{e_{ij}\in\partial\mathcal{K}_i}\int_{e_{ij}}\left(\frac{\partial\overline{F}^{(n)}}{\partial U_j}\delta U_j^{(n)}\right)\cdot n_{ij}dl=0, \tag{16}$$

where $\overline{F}^{(n)}=\overline{F}(U_i^{(n)},U_j^{(n)})$. The U_i is updated by

$$U_i^{(n+1)}=U_i^{(n)}+\tau_i\delta U_i^{(n)}, \tag{17}$$

where τ_i is a relaxation parameter on $\mathcal{K}_i$.

The calculation of the derivatives of the numerical flux to the conservative variables $\partial\overline{F}/\partial U_j$ has been discussed extensively in the references. Often the derivatives are obtained by using chain rule, which may lead to long and tedious formulas. It is in general difficult to get the exact expression of these derivatives, especially when the boundary condition is evolved. Even these approximated derivatives are often very complex, which gives rise to serious difficulties in the implementation. As an alternative, we approximate the derivatives using numerical differentiation, which makes the algorithm converge successfully. The Jacobians are based on the data at the Gauss points on the boundary of each cell. First, the piecewise polynomial is reconstructed on each cell, then we can obtain the left and right states of conserved variables on each edge, and so the numerical flux of this edge. Finally, the numerical differentiation $\partial\overline{F}/\partial U$ can be obtained by perturbing the left or right states of conserved variables a little bit. We use the following forms to approximate $\partial\overline{F}/\partial U_i$ and $\partial\overline{F}/\partial U_j$, say, the element in the Jacobian matrix is approximated as:

$$\frac{\partial\overline{F}_l}{\partial U_{i,k}}\approx\frac{\overline{F}_l(U_{i,k}^{(n)}+\varepsilon\delta U_{i,k}^{(n)},U_{j,k}^{(n)})-\overline{F}_l(U_{i,k}^{(n)},U_{j,k}^{(n)})}{|\varepsilon\delta U_{i,k}^{(n)}|},$$
$$\frac{\partial\overline{F}_l}{\partial U_{j,k}}\approx\frac{\overline{F}_l(U_{i,k}^{(n)},U_{j,k}^{(n)}+\varepsilon\delta U_{j,k}^{(n)})-\overline{F}_l(U_{i,k}^{(n)},U_{j,k}^{(n)})}{|\varepsilon\delta U_{j,k}^{(n)}|},$$

where $\frac{\partial\overline{F}_l}{\partial U_{i,k}}$ means the derivative of the lth element of vector $\overline{F}$ respect to the kth element of vector U_i, and ε is chosen as 10^{-6} in our implementation which is about half-word length of the machine epsilon.

Without the time evolving term, (16) is in fact a singular system. We propose to regularize it using the local residual [24]:

$$\alpha\|R_i^{(n)}\|_{l^1}\delta U_i^{(n)}+\sum_{e_{ij}\in\mathcal{K}_i}\int_{e_{ij}}\left(\frac{\partial\overline{F}^{(n)}}{\partial U_i}\delta U_i^{(n)}\right)\cdot n_{ij}dl+\sum_{e_{ij}\in\mathcal{K}_i}\int_{e_{ij}}\left(\frac{\partial\overline{F}^{(n)}}{\partial U_j}\delta U_j^{(n)}\right)\cdot n_{ij}dl=-R_i^{(n)}, \tag{18}$$

where α is a positive constant and

$$R_i^{(n)}:=\sum_{e_{ij}\in\mathcal{K}_i}\int_{e_{ij}}\overline{F}^{(n)}\cdot n_{ij}dl. \tag{19}$$

The flow chart of the Newton-iteration algorithm is given below.

```
Algorithm 1. Newton-iteration

1. Let $U^{(0)}$ be initial value, and let $n = 0$;
2. Solve linear system (18) to get $\delta U^{(n)}$;
3. Update $U_i^{(n+1)}$ by using $U_i^{(n+1)} = U_i^{(n)} + \tau_i\delta U_i^{(n)}$ in every cell;
4. Reconstruct $U^{(n+1)}$ on each cell using the method in Sections 3 and 4;
5. Check if the residual $R^{(n+1)}$ is small enough: if yes, stop; otherwise, set $n = n + 1$ and go to step 2.
```

6. The geometric multigrid solver

In step 2 of Algorithm 1, we use the geometric multigrid method to solve the linear system. There are two main components of the multigrid method: projection operator and smoother. For constructing project operator, we need to build the so-called element patches. First, the seed cells are chosen by using the following principles: any two seed cells have no common vertexes or sides; and according to [6], it is advised that each element patch contains at least four (for 2D) cells. In the implementation, we sort the cells around the seed cell according to the numbers of adjoining sides in descending order, and choose

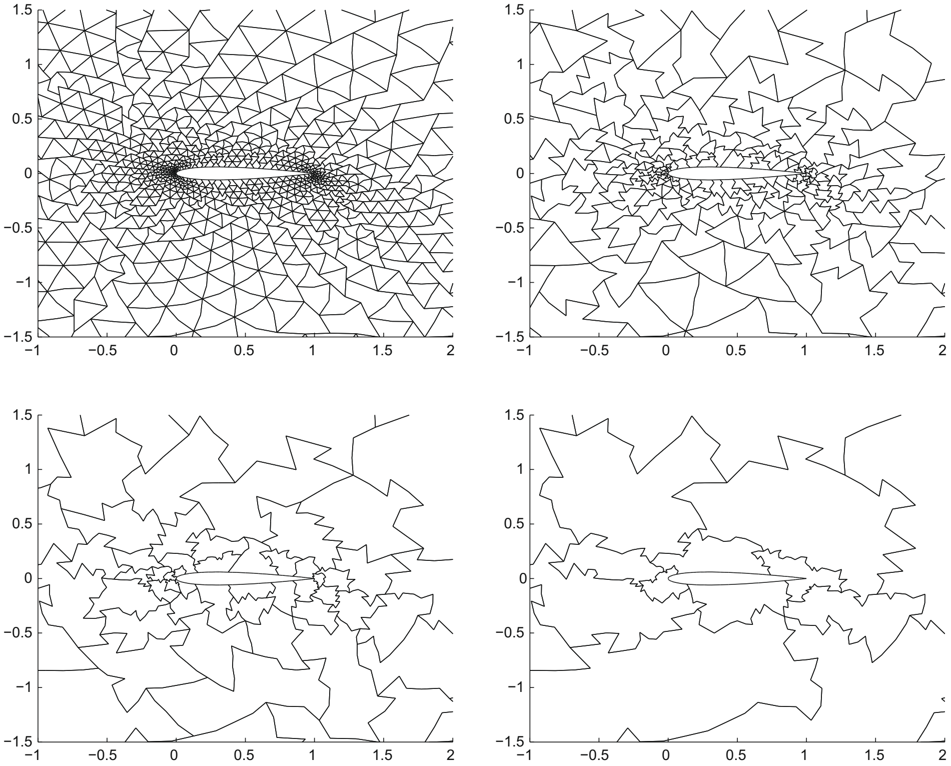

Fig. 2. Element patches near the airfoil body of NACA 0012 generated by the aggregation from a quasi-uniform mesh. It shows the element patches on four continuous levels.

the first three cells as the elements of the patch. Note that if we implement strictly with the principles above, there may be some single cells left. We eliminate such cells by merging it to the 'nearest' patch (i.e., the patch which has the most common sides with the cell). Fig. 2 shows the element patches generated on four continuous levels.

We denote the original partition on $\bar{\Omega}$ as $\mathcal{T}_0 = \mathcal{T}$. The aggregated multigrid method will give a sequence of meshes denoted as $\mathcal{T}_m,\ m = 1, 2, \ldots$ For every cell $\mathcal{K}_{i,m+1} \in \mathcal{T}_{m+1}$, it is a union of some cells in mesh $\mathcal{T}_m$:

$$\mathcal{K}_{i,m+1} = \cap_{j\in\mathcal{I}(i,m+1)}\mathcal{T}_{j,m}, \tag{20}$$

where $\mathcal{I}(i, m+1)$ is the set of indices for cells in $\mathcal{T}_m$ as a part of cell $\mathcal{K}_{i,m+1}$.

The projected linear system on mesh level $m+1$ from level m is given as

$$\sum_j A_{ij,m+1}\delta U_{i,m+1} = -R_{i,m+1}, \tag{21}$$

where

$$A_{ij,m+1} = \sum_{\xi\in\mathcal{I}(i,m+1)}\ \sum_{\eta\in\mathcal{I}(j,m+1)} A_{\xi\eta,m}, \tag{22}$$

$$R_{i,m+1} = \sum_{j\in\mathcal{I}(i,m+1)} (R_{j,m} + \sum_{\xi} A_{j\xi,m}\delta U_{\xi,m}). \tag{23}$$

As the corrections obtained from the mesh level $m+1,\ \delta U_{i,m+1}$ is added back to the solution on the mesh level m as

$$\delta U_{j,m} \to \delta U_{j,m} + \delta U_{i,m+1}, \quad \text{if } j \in \mathcal{I}(i, m+1). \tag{24}$$

We choose block LU-SGS method proposed by Chen and Wang [9] as our smoother, which can be formulated as two symmetric loops:

1. For $\mathcal{K}_{i,m} \in \mathcal{T}_m$, loop for i increasingly

$$\delta U_{i,m} := A_{ii,m}^{-1}\left(R_{i,m} - \sum_{i\neq j} A_{ij,m}\delta U_{j,m}\right). \tag{25}$$

2. For $\mathcal{K}_{i,m} \in \mathcal{T}_m$, loop for i decreasingly

$$\delta U_{i,m} := A_{ii,m}^{-1}\left(R_{i,m} - \sum_{i \neq j} A_{ij,m}\delta U_{j,m}\right). \tag{26}$$

The *V*-cycle type iteration is adopted in the implementation of our linear multigrid solver.

The multigrid algorithm, as the inner iterations of the complete Newton-iteration in Algorithm 1, is used for only a few steps in each Newton-iteration. According to the numerical experiences, too many iterations in the multigrid method will make the solver inefficient, especially when the Mach number is bigger than 0.9. For the low Mach number case, increasing number of iterations appropriately in the multigrid method can decrease the number of iterations of the outer Newton method. In the numerical simulations, we just set the number of iterations in the multigrid to be 2, and it works very well for all cases in the next section.

7. Numerical results

In this section, we mainly show two aspects of our algorithm. One is that with the WENO hierarchical limiting strategy; the high-order numerical accuracy is obtained in the smooth region, while the numerical oscillations nearby the shock are removed or reduced significantly. The other one is about robustness; our algorithm can work well for several test examples with one fixed set of parameters.

The two aspects above are demonstrated in the following subsections. In the simulations, we always use $\alpha = 2$ (see (18)), $\tau_i = \tau = 0.2$ (see (17)), $\mu = 1.0$ (see (10)), and the smoothing steps in the multigrid solver is 2.

All the computations are carried out on IBM ThinkPad T60 type Notebook with core speed to be 1.83 GHz. The code is based on the Adaptive Finite Element Package (AFEPack) developed by Li and Liu [23].

In all numerical experiments in this section, the following quantities are used as the initial guess of the Newton-iteration: the density $\rho = 1.0$, the velocity $V = (u, v) = (\cos\theta, \sin\theta)$, where θ is the attack angle. The other quantities such as pressure p and energy e are determined by ρ, V together with the Mach number.

7.1. Numerical convergence

Example 7.1. The test problem is the two-dimensional steady-state, subsonic flow around a disk at Mach number $M_\infty = 0.38$ [4].

The computations have been performed on four successively refined grids, i.e. 16 × 12, 32 × 24, 64 × 48, 128 × 96 points. Fig. 3 shows the initial mesh of the whole domain and the detail mesh around the inner circle. The grids extend about 20 diameters away from the circle.

From the quality of numerical solution by the symmetry of Mach number contours around the circle in the flow field shown in Figs. 4 and 5, we can see that numerical results obtained with quadratic reconstruction are much more accurate than those with the piecewise linear reconstruction. By using the entropy production as a criterion to measure the accuracy

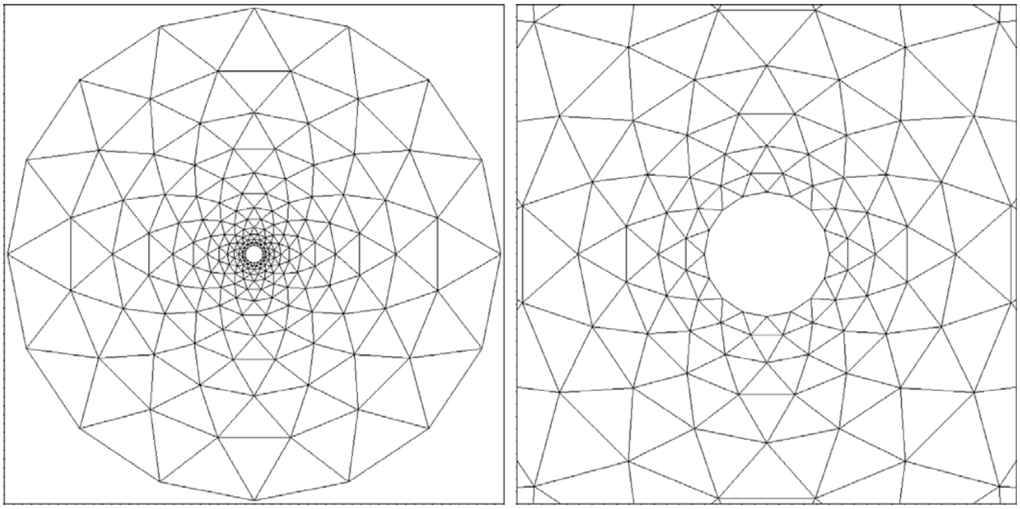

Fig. 3. Mesh used in the simulation of subsonic flow around a disk. The left one is the whole mesh, the right one is the detail of mesh around the inner disk.

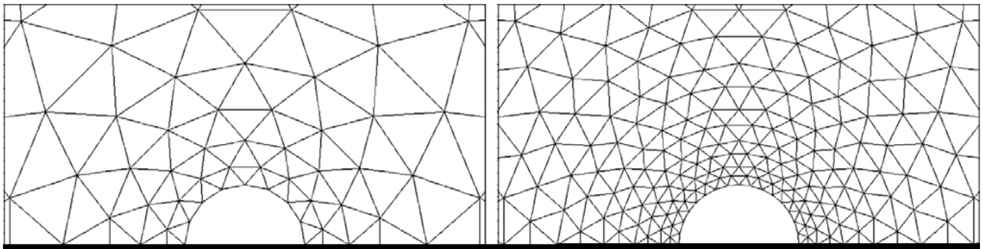

Fig. 4. The mesh details around the circle (top), Mach isolines with linear reconstruction (middle) and with quadratic reconstruction (bottom). Left column is with 16 × 12 grid and right is with 32 × 24.

and quality of the numerical solution, the convergence of numerical methods is also studied, and the results are shown in Table 2. As expected, the quality of the numerical solutions is improved when meshes are refined, and the order of convergence for the quadratic case is consistent with the theoretical one. Besides the results obtained with the WENO hierarchical limiting strategy, the results without using limiting procedure are also listed in Table 2. It can be observed that with the limiting procedure, both the numerical accuracy and convergence order are affected. However, with the mesh refinement, the

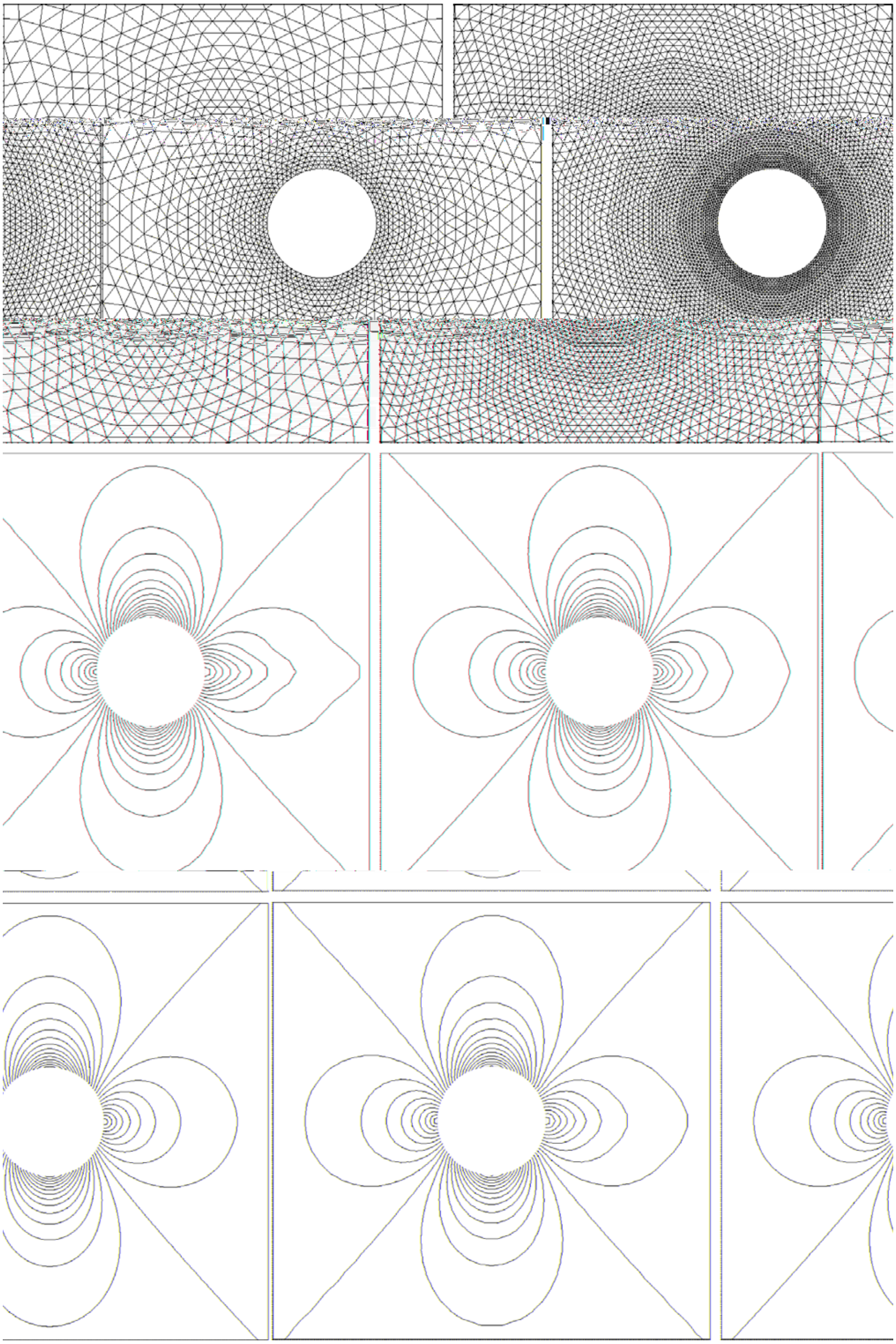

Fig. 5. Same as Fig. 4, except with 64 × 48 grid (left) and 128 × 96 grid (right).

difference of the convergence orders becomes smaller. Regardless of using limiting procedure or not, the numerical accuracy given by the quadratic reconstruction is better than that obtained with the linear reconstruction.

It is pointed out that the region used to measure the errors in Table 2 is the cirque with radius from 0.5 to 2.0, which contains the inner boundary. As we discussed in Section 3 for the quadratic reconstruction, if the polygon is used to approximate the curved boundary of computational domain, then the second-order error will be introduced from the curved

Table 2
The L_2 entropy errors and orders of convergence for the linear reconstruction and quadratic reconstruction computations for the flow around a circle on four successive grids; errors are measured in regions $0.5 \leqslant r \leqslant 2.0$. "L-lim" and "Q-lim" stand for linear and quadratic reconstruction with limiting procedure, respectively. "Q." stands for quadratic reconstruction without limiting procedure.

Mesh	Err. (L-lim)	Order	Err. (Q-lim)	Order	Err. (Q.)	Order
16 × 12	2.50e−01		2.30e−01		1.61e−01	
32 × 24	6.22e−02	2.0	5.05e−02	2.2	2.26e−02	2.8
64 × 48	1.03e−02	2.6	5.99e−03	3.1	2.52e−03	3.2
128 × 96	1.74e−03	2.6	6.33e−04	3.2	2.56e−04	3.3

boundary, which will pollute the numerical accuracy in the whole computational domain. From Table 2, we can see that the method used to take care of the curved boundary in this paper works very well.

Example 7.2. The test problem is a NACA 0012 airfoil in the flow field with the free-stream condition, Mach number 0.8 and the attack angle 1.25°.

The computations have been performed on three successively refined grids (2662 cells, 10,648 cells and 42,592 cells, respectively). Fig. 6 shows a coarse mesh with 2662 elements. The numerical results in Fig. 7 show the Mach contour lines around the whole airfoil and nearby the shock region with three different meshes. The results are obtained by using the WENO hierarchical limiting procedure. It is observed from Fig. 7 that the Mach contour lines become smoother and the shock profiles become sharper when the mesh is refined.

With the help of the WENO hierarchical limiting strategy, the numerical oscillations around the shock profiles are reduced effectively. This can be observed from the comparison between the results shown in Figs. 7 (middle) and 8 (where the limiting strategy is not used) that the Mach isolines obtained with the limiting procedure are much more ordered. However, it should be pointed out that the numerical oscillations are not removed thoroughly, as the slight overshoot/undershoot phenomenon are still observed in Fig. 7. This seems not avoidable with the present hierarchical reconstruction strategy; see Liu and co-workers [27,26,42] where overshoot/undershoot are also observed. More efforts have to be made to improve the limiting strategy.

7.2. Robustness

We applied our algorithm to problems with different free-stream configurations and different geometric configurations.

7.2.1. On different free-stream configurations

Three free-stream configurations are tested for the airfoil NACA 0012 or RAE 2822:

Example 7.3. Low free-stream Mach number of 0.3 and a big attack angle of 3°, the airfoil is NACA 0012. Results are shown in Fig. 9.

Example 7.4. Moderate free-stream Mach number of 0.73 and attack angle of 1°, the airfoil is RAE 2822. Results are shown in Fig. 10.

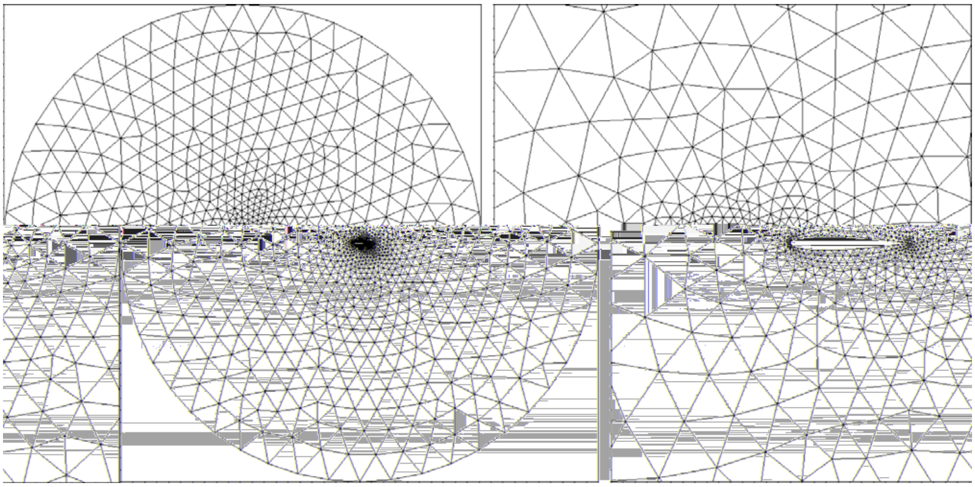

Fig. 6. The mesh grids generated using EasyMesh around NACA 0012 airfoil, containing 2662 elements. The left one is the whole mesh, and the right one shows the mesh near the body of the airfoil.

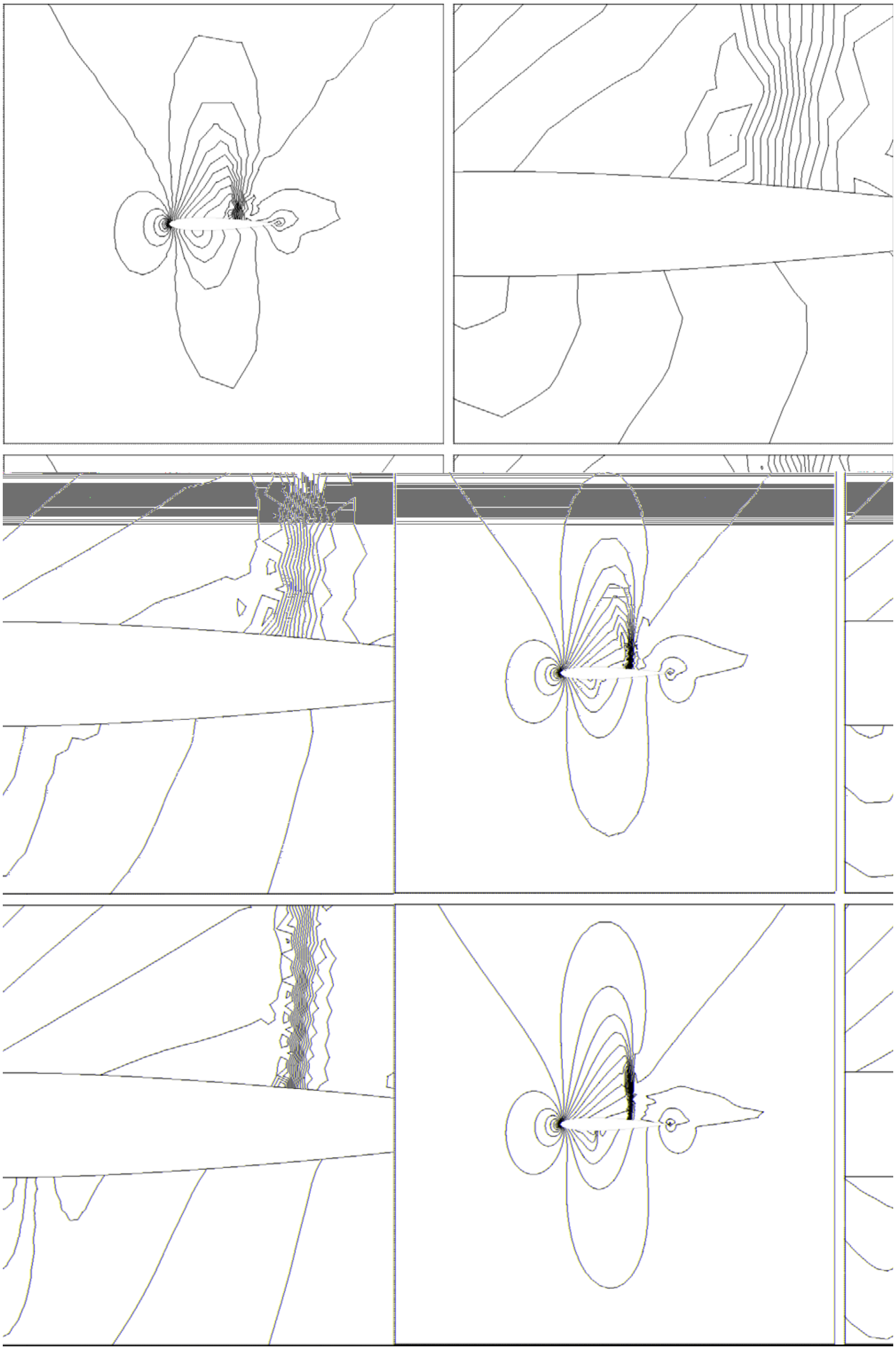

Fig. 7. Mach isolines around the whole airfoil (left) and around the shock profiles (right). The free-stream condition is Mach number 0.8 and attack angle 1.25°. Top: 2662 cells, middle: 10,648 cells and bottom: 42,582 cells. The results are given with the quadratic reconstruction, and the WENO hierarchical limiting procedure is used.

Example 7.5. High free-stream Mach number of 0.99 and attack angle of 0.0°, the airfoil is NACA0012. Results are shown in Fig. 11.

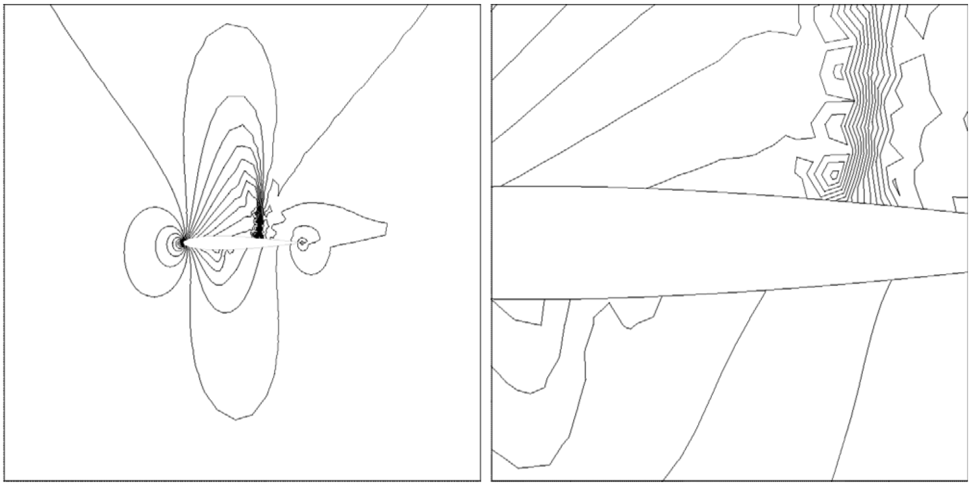

Fig. 8. Same as Fig. 7 (middle row), expect that the limiting strategy is not used.

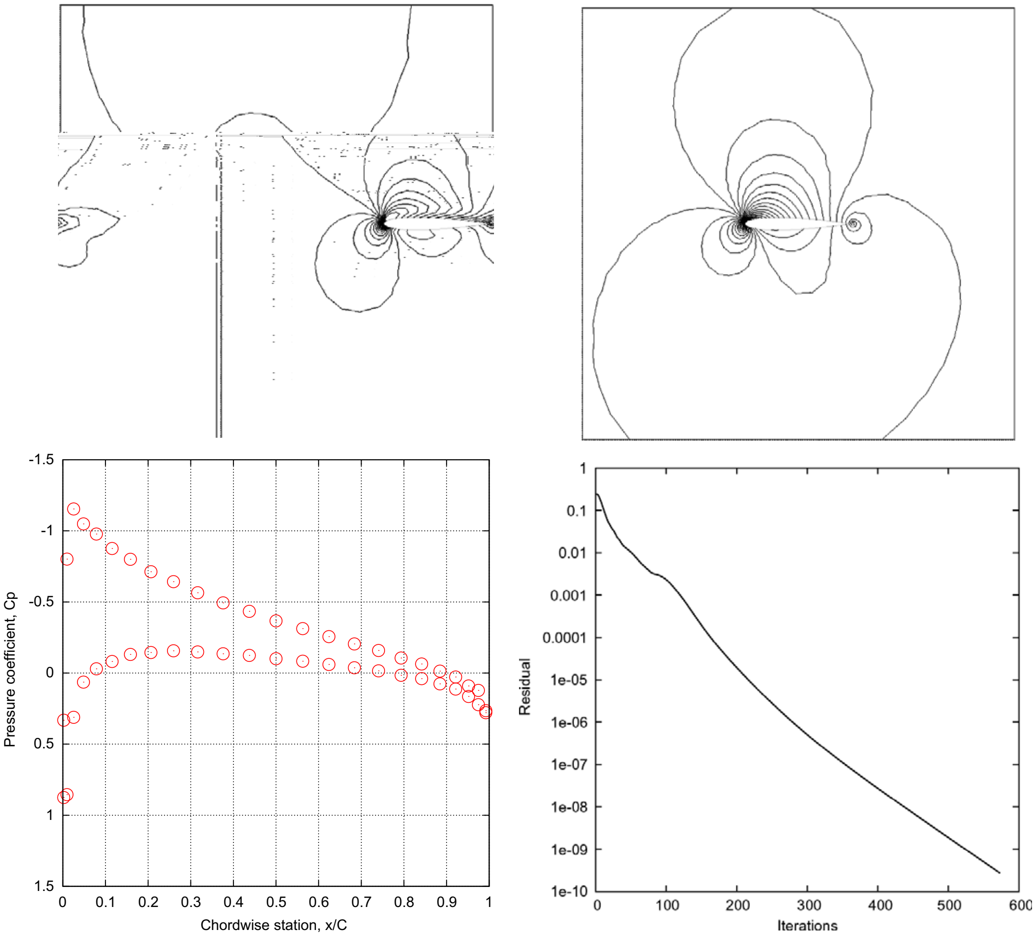

Fig. 9. The contours of Mach number (top left), pressure (top right), surface pressure profiles of the airfoil (bottom left), and convergence history (bottom right) for NACA 0012 airfoil with free-stream configuration: the Mach number is 0.3 and the attack angle is 3°.

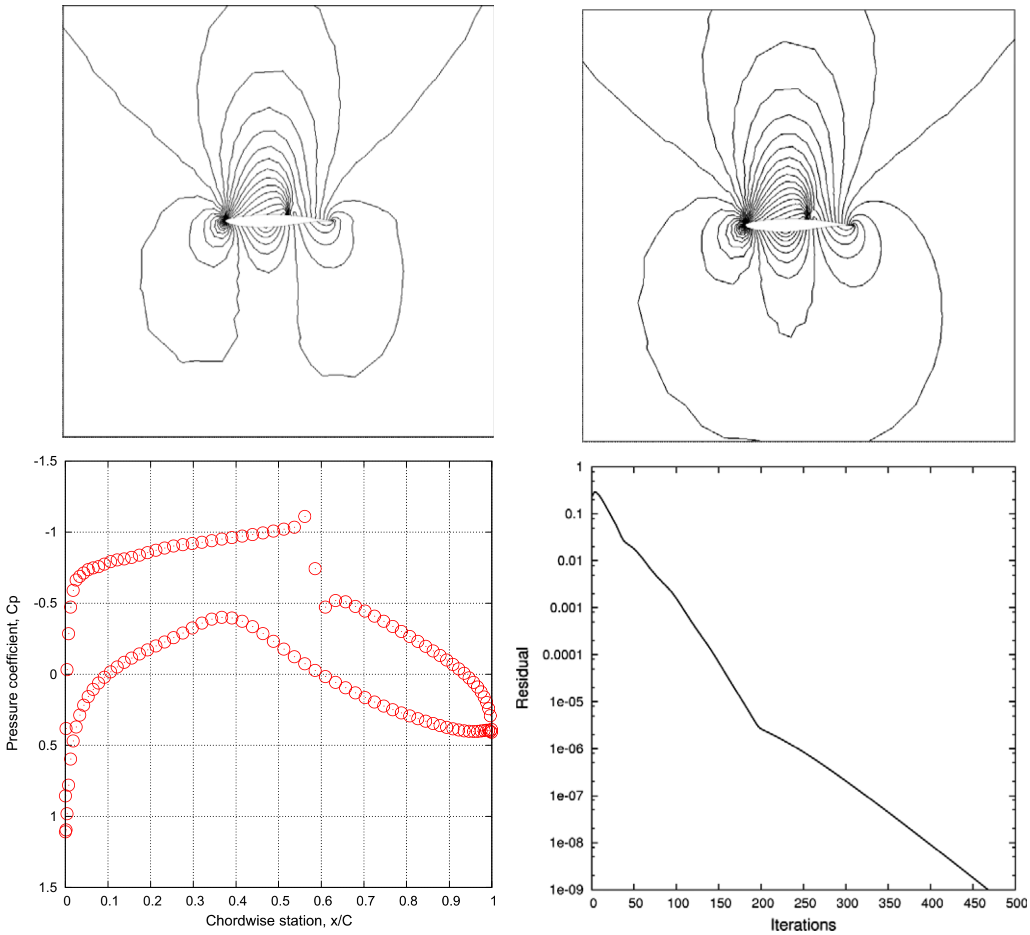

Fig. 10. Same as Fig. 9 except with Mach number 0.73 and attack angle 1.0°, and the airfoil is RAE 2822.

The mesh used in all the numerical tests are generated by EasyMesh. The mesh for NACA 0012 airfoil contains 2662 cells, and the mesh for RAE 2822 airfoil contains 3444 cells. From results shown in Figs. 9–11, we can see that our algorithm works very well with the *same set* of parameters indicated at the beginning of this section. For all free-stream configurations, the residual of the system reduced to the 10^{-10} with around 500 Newton-iteration steps. Though there are still slight numerical oscillations which can be observed from the plot of surface pressure coefficients, the WENO hierarchical limiting strategy improved the quality of numerical solutions nearby the shock region significantly.

7.2.2. On multi-airfoil in the flow fields

To test the robustness of the algorithm, we also present the following example which has two airfoils in the flow field.

Example 7.6. One NACA 0012 airfoil and one RAE 2822 airfoil in the flow field with Mach number 0.73 and the attack angle 1.0°.

Fig. 12 presents the numerical solutions of Example 7.6. It can be seen from the figure that our algorithm works successfully. It is also observed from the convergence history that the residual for this case can achieve the machine accuracy with almost the same number of iterations as that of the single airfoil case (see Fig. 10).

We emphasize again that one of the advantages of our proposed scheme is insensitive to the parameters used in the computations: $\alpha = 2$ (see (18)), $\tau_i = \tau = 0.2$ (see (17)), $\mu = 0.2$ in the WENO procedure, and the smoothing steps in the multigrid solver is 2. This is in contrast with many existing algorithms where quite a number of parameters have to be fixed case by case.

Of course, by adjusting the parameters used, the efficiency can be further improved. For example, for simulations shown in Figs. 9 and 11, if the value $\tau = \tau_i = 0.6$ is used, the iteration steps can reduced significantly compared with that using $\tau = \tau_i = 0.2$, see Fig. 13.

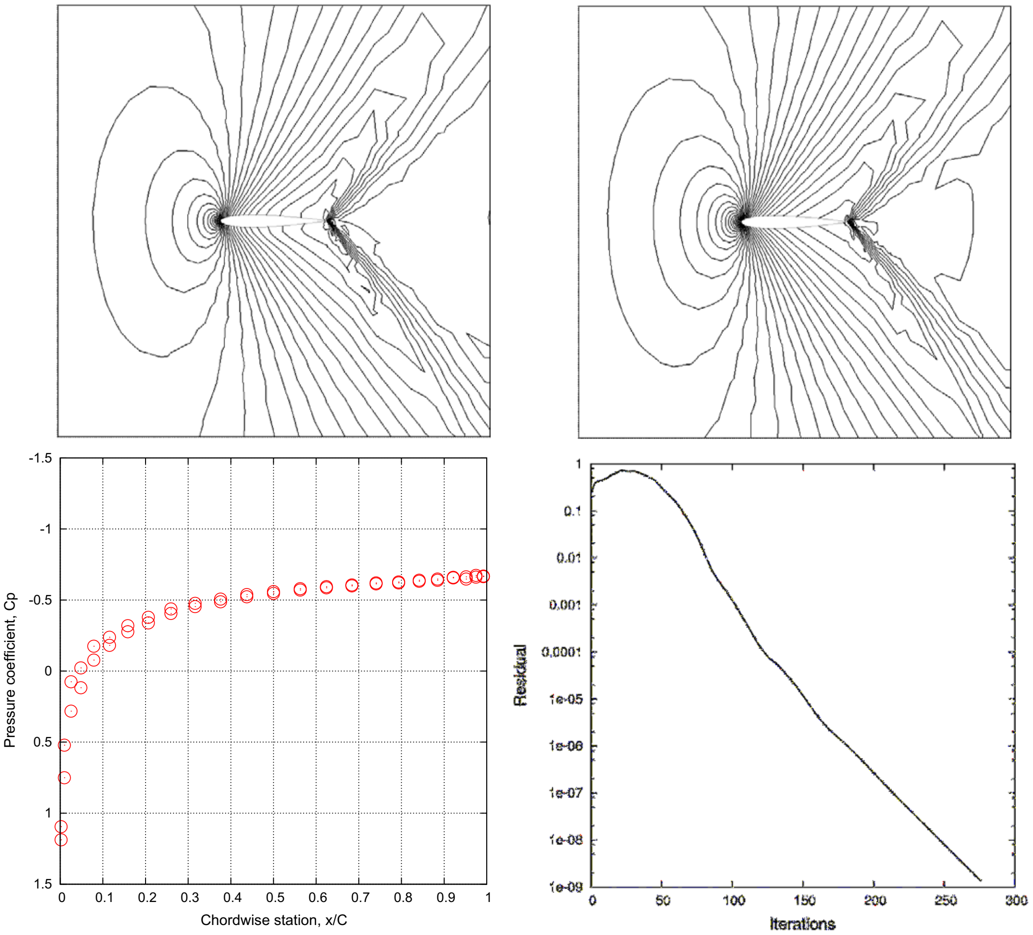

Fig. 11. Same as Fig. 9 except with Mach number 0.99 and attack angle 0.0°.

8. Conclusion remarks

In this work, we developed a high-order and robust algorithm for steady two-dimensional Euler equations on unstructured grids, which uses the Newton-iteration and the multigrid method to solve the linearized Jacobian matrix. The block LU-SGS iteration was adopted as the smoother of the multigrid algorithm. The WENO hierarchical limiting strategy was presented, which can reduce the numerical oscillations significantly and can also keep the high numerical accuracy generated by the quadratic reconstruction. The local Jacobian matrix of the numerical fluxes were computed using the numerical differentiation, which can simplify the implementation significantly. Apart from the high-order accuracy, the numerical examples also demonstrated the robustness of our algorithm: one set of the four parameters (i.e., the proportional constant α for the local residual, the relaxation parameter τ in the Newton-iteration, the weight μ in the WENO scheme, and the number of smoothing steps in the multigrid solver) is sufficient for various geometrical configurations and free-stream configurations. This is regarded as a strong indicator of robustness. Of course, the efficiency can be optimized by adjusting these parameters case by case.

The WENO hierarchical limiting strategy used in this paper worked smoothly for all numerical experiments. However, the slight undershoot/overshoot phenomenon can still be observed near the shocks. This will be further studied in future by improving the limiting strategy.

It is important to point out that the scheme in this paper is not cheaper than traditional finite volume schemes, since the least square interpolation used in this paper is not less expensive than traditional least square reconstruction from cell averages in a finite volume setting. This is very different from the residual distribution scheme in Chou and Shu [10] in which a non-smooth tensor product mesh is assumed and then the interpolation is avoided. A numerical integration for the flux along the cell boundary is directly computed based on the point values of the solution along the line of the cell boundary, thus resulting in significant cost saving over finite volume schemes.

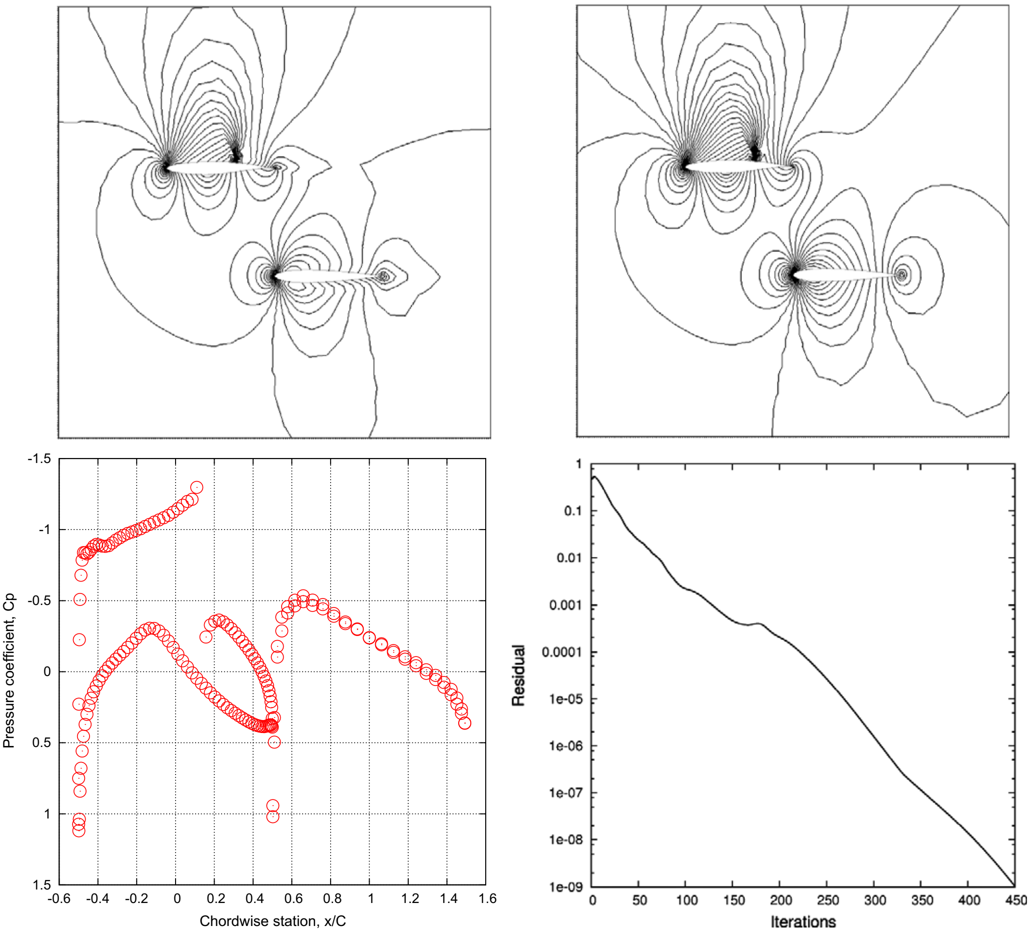

Fig. 12. Same as Fig. 10 except with two airfoils in the flow filed: NACA 0012 and RAE 2822.

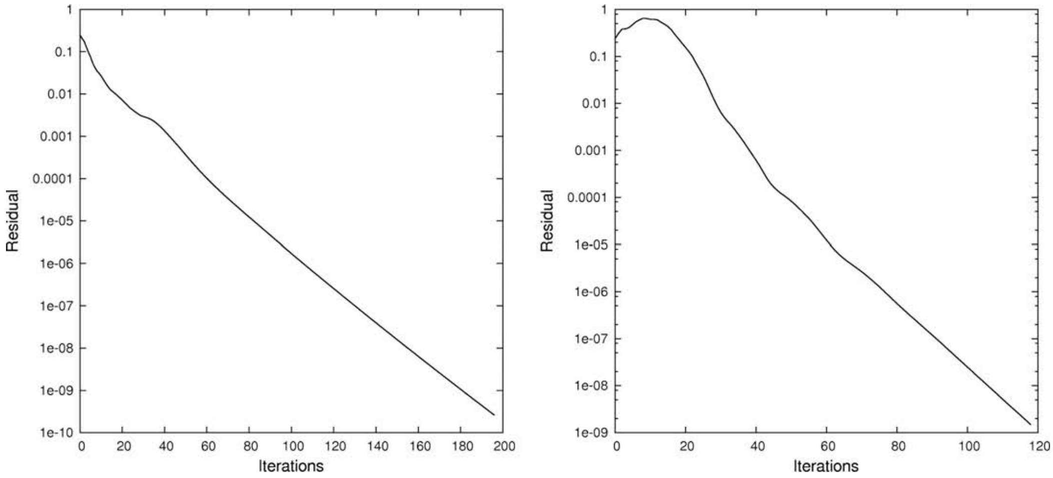

Fig. 13. Convergence history for NACA 0012 airfoil in the flow filed with free-stream configuration: Mach number 0.3, attack angle 3° (left), and Mach number 0.99, attack angle 0° (right). The parameter $\tau = \tau_i = 0.6$ is used.

Acknowledgments

The research of Hu is supported by a studentship from Hong Kong Baptist University. The research of Li was supported in part by the National Basic Research Program of China under the Grant 2005CB321701 and the National Science Foundation of China under the Grant 10731060. The research of Tang was supported in part by Hong Kong Research Grants Council, the FRG grants of Hong Kong Baptist University and an Cheung Kong Chair Professorship of the Chinese Ministry of Education through Beijing University of Aeronautics and Astronautics.

References

[1] R. Abgrall, C.W. Shu, Development of residual distribution schemes for discontinuous Galerkin method: the scalar case with linear elements, Commun. Comput. Phys. 5 (2009) 376–390.

[2] M.F. Adams, Y. Nishimura, Parallel algebraic multigrid methods in gyrokinetic turbulence simulations, Commun. Comput. Phys. 2 (2007) 881–899.

[3] T.J. Barth, Recent developments in high order k-exact reconstruction on unstructured meshes, AIAA Paper, 93:0668, 1993.

[4] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2d Euler equations, J. Comput. Phys. 138 (1997) 251–285.

[5] P. Batten, M.A. Leschziner, U.C. Goldberg, Average-state Jacobians and implicit methods for compressible viscous and turbulent flows, J. Comput. Phys. 137 (1997) 38–78.

[6] J. Blazek, Computational Fluid Dynamics: Principles and Applications, Elsevier Science Publication, 2005.

[7] A. Brandt, Guide to Multigrid Development, Multigrid Methods I, vol. 960, Springer Verlag, New York, 1981.

[8] Q. Chang, Z. Huang, Efficient algebraic multigrid algorithms and their convergence, SIAM J. Sci. Comput. 24 (2002) 597–618.

[9] R.F. Chen, Z.J. Wang, Fast block lower–upper symmetric Gauss–Seidel scheme for arbitrary grid, AIAA J. 38 (12) (2000) 2238–2245.

[10] C.S. Chou, C.W. Shu, High order residual distribution conservative finite difference WENO schemes for steady state problems on non-smooth meshes, J. Comput. Phys. 214 (2006) 698–724.

[11] B. Cockburn, C.W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws v: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.

[12] A. Csik, M. Ricchiuto, H. Deconinck, A conservative formulation of the multidimensional upwind residual distribution schemes for general nonlinear conservation laws, J. Comput. Phys. 179 (2002) 286–312.

[13] M. Delanaye, J.A. Essers, Finite volume scheme with quadratic reconstruction on unstructured adaptive meshes applied to turbomachinery flows, in: ASME IGTI Gas Turebine Conference, Houstion, USA, 1995.

[14] R. Enander, A.R. Karlsson, Implicit explicit residual smoothing in multigrid cycle, AIAA Paper, 95-0204, 1995.

[15] R. Enander, B. Sjogreen, Implicit explicit residual smoothing for upwind schemes, Report No. 179, Uppsala University, 1996.

[16] C.Q. Hu, C.W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.

[17] A. Jameson, Iterative solution of transonic over airfoil and wings including at Mach 1, Commun. Pure Appl. Math. 27 (1974) 283–309.

[18] A. Jameson, Solution of the Euler equations for two dimensional transonic flow by a multigrid method, Appl. Math. Comput. 13 (1983) 327–355.

[19] A. Jameson, Analysis and design of numerical schemes for gas dynamics 1: artificial diffusion, upwind biasing, limiters and their effect on accuracy and multigrid convergence, Int. J. Comput. Fluid Dyn. (1994).

[20] A. Jameson, T.J. Baker, Solution of the Euler equations for complex configurations, AIAA Paper, 83-1929, 1983.

[21] G. Jiang, C.W. Shu, Efficient implementation of weighted ENO, J. Comput. Phys. 126 (1996) 202.

[22] B.V. Leer, Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes, Commun. Comput. Phys. 1 (2006) 192–206.

[23] R. Li, W.B. Liu, <http://circus.math.pku.edu.cn/AFEPack>.

[24] R. Li, X. Wang, W.B. Zhao, A multigrid block lower-upper symmetric Gauss–Seidel algorithm for steady Euler equation on unstructured grids, Numer. Math. Theor., Meth. Appl. 1 (2008) 92–112.

[25] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[26] Y.J. Liu, C.W. Shu, E. Tadmor, M. Zhang, Central discontinuous galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction, SIAM J. Numer. Anal. 45 (2007) 2442–2467.

[27] Y.J. Liu, C.W. Shu, E. Tadmor, M. Zhang, Non-oscillatory hierarchical reconstruction for central and finite volume schemes, Commun. Comput. Phys. 2 (5) (2007) 933–963.

[28] H. Luo, J.D. Baum, R. Lohner, A discontinuous galerkin method based on a taylor basis for the compressible flows on arbitrary grids, J. Comput. Phys. 227 (2008) 8875–8893.

[29] D.J. Mavriplis, On convergence acceleration techniques for unstructured meshes, AIAA Paper 98-2966, 1998.

[30] D.J. Mavriplis, A. Jameson, L. Martinelli. Multigrid solution of the Navier–Stokes equations on triangular meshes, ICASE Report 89-11, February 1989.

[31] K. Michalak, C. Ollivier-Gooch, Limiters for unstructured higher-order accurate solutions of the Euler equations, in: AIAA 46th Aerospace Sciences Meeting, 2008.

[32] C.R. Mitchell, R.W. Walters, K-exact reconstruction for the Navier–Stokes equation on arbitrary grids , AIAA Paper, 93:0536, 1993.

[33] B. Niceno, <http://www-dinma.univ.trieste.it/nirftc/research/easymesh/>.

[34] C. Ollivier-Gooch, M.V. Altena, A high-order-accurate unstructured mesh finite-volume scheme for the advection–diffusion equation, J. Comput. Phys. 181 (2002) 729–752.

[35] C.W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, Springer, 1998, pp. 325–432.

[36] Y. Sun, Z.J. Wang, Y. Liu, High-order multidomain spectral difference method for the Navier–Stokes equations on unstructured hexahedral grids, Commun. Comput. Phys. 2 (2007) 310–333.

[37] R.F. Tomaro, W.Z. Strang, L.N. Sankar, An implicit algorithm for solving time dependent flows on unstructured grids, AIAA Paper, 97-0333, 1997.

[38] Z.J. Wang, A fast nested multi-grid viscous flow solver for adaptive Cartesian/Quad grids, J. Numer. Mech. Fluids 33 (2000) 657–680.

[39] Z.J. Wang, Y. Liu, Extension of the spectral volume method to high-order boundary representation, J. Comput. Phys. 211 (2006) 154–178.

[40] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids iv: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2004) 716–741.

[41] D.L. Whitaker, D.C. Slack, R.W. Walters, Solution algorithms for the two-dimensional Euler equations on unstructured meshes, in: AIAA, Aerospace Sciences Meeting, 28th, Reno, NV, January 1990.

[42] Z.L. Xu, Y.J. Liu, C.W. Shu, Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO-type linear reconstruction and partial neighboring cells, J. Comput. Phys. 228 (2009) 2194–2212.