Journal of Computational Physics 227 (2008) 8875–8893

Contents lists available at ScienceDirect

Journal of Computational Physics

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

A discontinuous Galerkin method based on a Taylor basis for the compressible flows on arbitrary grids

Hong Luo [a,*], Joseph D. Baum [b], Rainald Löhner [c]

[a] Department of Mechanical and Aerospace Engineering, North Carolina State University, 2601 Stinson Drive – Campus Box 7910, Raleigh, NC 27695, USA
[b] Center for Applied Computational Sciences, Science Applications International Corporation, McLean, VA 22102, USA
[c] Department of Computational and Data Sciences, George Mason University, Fairfax, VA 22030, USA

ARTICLE INFO

Article history:
Received 14 October 2007
Received in revised form 17 June 2008
Accepted 27 June 2008
Available online 18 July 2008

Keywords:
Discontinuous Galerkin methods
Compressible flows
Arbitrary grids

ABSTRACT

A discontinuous Galerkin method based on a Taylor basis is presented for the solution of the compressible Euler equations on arbitrary grids. Unlike the traditional discontinuous Galerkin methods, where either standard Lagrange finite element or hierarchical node-based basis functions are used to represent numerical polynomial solutions in each element, this DG method represents the numerical polynomial solutions using a Taylor series expansion at the centroid of the cell. Consequently, this formulation is able to provide a unified framework, where both cell-centered and vertex-centered finite volume schemes can be viewed as special cases of this discontinuous Galerkin method by choosing reconstruction schemes to compute the derivatives, offer the insight why the DG methods are a better approach than the finite volume methods based on either TVD/MUSCL reconstruction or essentially non-oscillatory (ENO)/weighted essentially non-oscillatory (WENO) reconstruction, and has a number of distinct, desirable, and attractive features, which can be effectively used to address some of shortcomings of the DG methods. The developed method is used to compute a variety of both steady-state and time-accurate flow problems on arbitrary grids. The numerical results demonstrated the superior accuracy of this discontinuous Galerkin method in comparison with a second order finite volume method and a third-order WENO method, indicating its promise and potential to become not just a competitive but simply a superior approach than its finite volume and ENO/WENO counterparts for solving flow problems of scientific and industrial interest.

1. Introduction

The discontinuous Galerkin methods [4,6,7,10,11,17,18,21–24] (DGM) have recently become popular for the solution of systems of conservation laws to arbitrary order of accuracy. The discontinuous Galerkin methods combine two advantageous features commonly associated to finite element and finite volume methods. As in classical finite element methods, accuracy is obtained by means of high-order polynomial approximation within an element rather than by wide stencils as in the case of finite volume methods. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces. In this respect, the methods are therefore similar to finite volume methods. In fact, the basic cell-centered finite volume scheme exactly corresponds to the DG(0) method, i.e., to the discontinuous Galerkin method using piecewise constant polynomials. Consequently, the DG(p) method with $p > 0$ can be regarded as the natural extension of finite volume methods to higher order methods. The discontinuous

* Corresponding author.
E-mail address: hong_luo@ncsu.edu (H. Luo).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.06.035

Galerkin methods have many distinguished features: (1) The methods are well suited for complex geometries since they can be applied on unstructured grids. In addition, the methods can also handle non-conforming elements, where the grids are allowed to have hanging nodes; (2) The methods are compact, as each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal (elements only communicate with Von Neumann neighbors (adjacent elements with a common face) regardless of the order of accuracy of the scheme), they are highly parallelizable. The compactness also allows for structured and simplified coding for the methods; (3) they can easily handle adaptive strategies, since refining or coarsening a grid can be achieved without considering the continuity restriction commonly associated with the conforming elements. The methods allow easy implementation of *hp*-refinement, for example, the order of accuracy, or shape, can vary from element to element; (4) They have several useful mathematical properties with respect to conservation, stability, and convergence.

However, the discontinuous Galerkin methods have a number of their own weaknesses. In particular, how to effectively control spurious oscillations in the presence of strong discontinuities and a lack of efficient time integration scheme for both time-accurate, and steady-state solutions remain two of unresolved issues in the DGM. Like any higher-order schemes (>1), the discontinuous Galerkin methods will suffer from non-physical oscillations in the vicinity of discontinuities that exist in problems governed by hyperbolic conservation laws. Two common approaches to address this issue are a discontinuity capturing and an appropriate slope limiter. The former adds explicitly consistent artificial viscosity terms to the discontinuous Galerkin discretization. The main disadvantage of this approach is that it usually requires some user-defined parameters, which can be both mesh and problem dependent. Classical techniques of slope limiting are not directly applicable for high-order DGM because of the presence of volume terms in the formulation. Therefore, the slope limiter is not integrated in the computation of the residual, but effectively acts as a post-processing filter. Many slope limiters used in the finite volume method (FVM) can then be used or modified to meet the needs of the DGM. Unfortunately, the use of the limiter will reduce the order of accuracy to first order in the presence of discontinuities. Indeed, it is not an exaggeration to state that the design of efficient, effective, and robust limiters is one of the bottlenecks in the development of DGM for solving conservation laws. Most efforts in the development of the DGM have primarily been focused on the exploration of their advantages such as higher-order spatial discretizations, posteriori error estimations, adaptive algorithms, and parallelizations. The temporal discretization methods have lagged far behind. Usually, explicit temporal discretizations such as multi-stage TVD (total variation diminishing) Runge–Kutta schemes [6,7,11] are used to advance the solution in time. In general, explicit schemes and their boundary conditions are easy to implement, vectorize and parallelize, and require only limited memory storage. However, for large-scale simulations and especially for high-order solutions, the rate of convergence slows down dramatically, resulting in inefficient solution techniques to steady state solutions. To speed up convergence, a multigrid strategy or an implicit temporal discretization is required. In general, implicit methods require the solution of a linear system of equations arising from the linearization of a fully implicit scheme at each time step or iteration. Recently, efforts have been made to develop efficient implicit solution methods for DGM. Unfortunately, the drawback is that they require a considerable amount of memory to store the Jacobian matrix, which may be prohibitive for large-scale problems and high-order solutions. Even in the implementation of so-called matrix-free implicit methods [25], where only a block diagonal matrix is required to store, the memory requirements can still be extremely demanding. The block diagonal matrix requires a storage of ($n$eqns $\times$ $n$degr) $\times$ ($n$eqns $\times$ $n$degr) $\times$ $n$elem, where $n$eqns is the number of components in solution vector (4 for 2D, and 5 for 3D Euler equations), $n$degr is the degrees of freedom for the polynomial (3 for P_1, 6 for P_2, and 10 for P_3 for triangle element in 2D. Four for P_1, 10 for P_2, and 20 for P_3 for tetrahedral element in 3D), and nelem is the number of elements for the grid. For example, for a fourth-order (cubic polynomial finite element approximation P_3) DGM in 3D, the storage of this block diagonal matrix alone requires 10,000 words per element! Indeed, it is our belief that a lack of efficient solvers is one of reasons that the application of DG method for engineering-type problems does not exist.

In the traditional DG methods either standard Lagrange or hierarchical node-based finite element basis functions are used to represent numerical polynomial solutions in each element. As a result, the unknowns to be solved are the variables at the nodes and the polynomial solutions are dependent on the shape of elements. For example, for a linear polynomial approximation in 2D, a linear polynomial approximation is used for triangular elements and the unknowns to be solved are the variables at the three vertices and a bi-linear polynomial approximation is used for quadrilateral elements and the unknowns to be solved are the variables at the four vertices. In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the centroid of the cell, which can be further expressed as a combination of cell-averaged values and their derivatives at the centroid of the cell. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes. As a result, this formulation is able to provide a unified framework, where both cell-centered and vertex-centered finite volume schemes can be viewed as special cases of this discontinuous Galerkin method by choosing reconstruction schemes to compute the derivatives, offer the insight why the DG methods are a better approach than the finite volume methods based on either TVD/MUSCL reconstruction or ENO/WENO reconstruction, and possesses a number of distinct, desirable, and attractive features and advantages, which can be effectively used to address the shortcomings of the DG methods mentioned above. First, the same numerical polynomial solutions are used for any shapes of elements, which can be triangle, quadrilateral, and polygon in 2D, and tetrahedron, pyramid, prism, and hexahedron in 3D. Using this formulation, DG methods can be easily implemented on arbitrary meshes. The numerical method based on this formulation has the ability to compute 1D, 2D, and 3D problems using the very same code, which greatly alleviates the need and pain for code maintenance and upgrade. Secondly, cell-averaged variables and their derivatives are handily available in this formulation. This makes implementation of a WENO limiter straightforward

and efficient that is required to eliminate non-physical oscillations in the vicinity of discontinuities. Thirdly, the basis functions are hierarchic. This greatly facilitates implementation of p-multigrid methods and p-refinement. Last, cell-averaged variable equations are decoupled from their derivatives equations in this formulation. This makes development of fast, low-storage implicit methods possible.

The objective of the efforts presented in this paper is to fully explore and take advantage of this discontinuous Galerkin formulation in an attempt to develop an accurate and efficient numerical method for computing compressible flows on arbitrary grids. Numerical experiments for a wide range of flow conditions are conducted to demonstrate the accuracy and efficiency of the developed discontinuous Galerkin method for computing a variety of compressible flow problems for complex geometries. The numerical results obtained illustrate the superior accuracy of this discontinuous Galerkin method over a finite volume method and WENO method, demonstrating that the discontinuous Galerkin methods provide a viable, attractive, and competitive alternative to the more traditional, more established, and more elaborate finite volume, finite element, and finite-difference methods for computing compressible flows around complex geometries. The remainder of this paper is structured as follows: The governing equations are described in Section 2. The discontinuous Galerkin method based on a Taylor basis is presented in Section 3. Extensive numerical experiments are reported in Section 4. Concluding remarks are given in Section 5.

2. Governing equations

The Euler equations governing unsteady compressible inviscid flows can be expressed in conservative form as

$$\frac{\partial \mathbf{U}(\mathbf{x},t)}{\partial t}+\frac{\partial \mathbf{F}_j(\mathbf{U}(\mathbf{x},t))}{\partial x_j}=0, \quad \text{in } \Omega \tag{1}$$

where Ω is a bounded connected domain in $\mathbf{R}^d$, d is the number of spatial dimension, and conservative state vector $\mathbf{U}$ and inviscid flux vectors $\mathbf{F}$ are defined by

$$\mathbf{U}=\begin{pmatrix}\rho\\ \rho u_i\\ \rho e\end{pmatrix}, \quad \mathbf{F}_j=\begin{pmatrix}\rho u_j\\ \rho u_i u_j+p\delta_{ij}\\ u_j(\rho e+p)\end{pmatrix}, \tag{2}$$

where the summation convention has been used and ρ, p, and e denote the density, pressure, and specific total energy of the fluid, respectively, and u_i is the velocity of the flow in the coordinate direction x_i. This set of equations is completed by the addition of the equation of state

$$p=(\gamma-1)\rho\left(e-\frac{1}{2}u_j u_j\right), \tag{3}$$

which is valid for perfect gas, where γ is the ratio of the specific heats.

3. Discontinuous Galerkin method

3.1. Discontinuous Galerkin spatial discretization

To formulate the discontinuous Galerkin method, we first introduce the following weak formulation of (1), which is obtained by multiplying (1) by a test function $\mathbf{W}$, integrating over the domain Ω, and performing an integration by parts:

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\mathbf{W}\mathrm{d}\Omega+\int_\Gamma \mathbf{F}_j\mathbf{n}_j\mathbf{W}\mathrm{d}\Gamma-\int_\Omega \mathbf{F}_j\frac{\partial \mathbf{W}}{\partial x_j}\mathrm{d}\Omega=0, \tag{4}$$

where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω, and $\mathbf{n}_j$ the unit outward normal vector to the boundary.

Before discretizing (4), we introduce some notation. Assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e, which can be triangles, quadrilaterals, polygons, or their combinations in 2D and tetrahedral, prism, pyramid, and hexahedral or their combinations in 3D. We introduce the following broken Sobolev space V_h^p

$$V_h^p=\{v_h\in[L_2(\Omega)]^m : v_h|_{\Omega_e}\in[V_p^m]\forall\Omega_e\in\Omega\}, \tag{5}$$

which consists of discontinuous vector-valued polynomial functions of degree p, and where m is the dimension of conservative state vector and

$$V_p^m=\mathrm{span}\left\{\prod_{i=1}^{d}x_i^{\alpha_i} : 0\leqslant\alpha_i\leqslant p, 0\leqslant i\leqslant d\right\}, \tag{6}$$

where α denotes a multi-index. Then, we can obtain the following semi-discrete form by applying the weak formulation (4) on each element Ω_e

$$\begin{cases}\text{find} \quad \mathbf{U}_h\in V_h^p \quad \text{such as}\\ \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e}\mathbf{U}_h\mathbf{W}_h\mathrm{d}\Omega+\int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j\mathbf{W}_h\mathrm{d}\Gamma-\int_{\Omega_e}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial \mathbf{W}_h}{\partial x_j}\mathrm{d}\Omega=0 \quad \forall\mathbf{W}_h\in V_h^p,\end{cases} \tag{7}$$

where $\Gamma_e(=\partial\Omega_e)$ denotes the boundary of Ω_e, $\mathbf{U}_h$ and $\mathbf{W}_h$ represent the finite element approximations to the analytical solution $\mathbf{U}$ and the test function $\mathbf{W}$, respectively, and both belong to the finite element space V_h^p. Assume that B_i is the basis of polynomial function of degrees p, this is then equivalent to the following system of N equations,

$$\frac{d}{dt}\int_{\Omega_e}\mathbf{U}_h B_i d\Omega + \int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j B_i d\Gamma - \int_{\Omega_e}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial B_i}{\partial x_j}d\Omega = 0 \quad 1 \leqslant i \leqslant N, \tag{8}$$

where N is the dimension of the polynomial space. In the traditional DGM, numerical polynomial solutions $\mathbf{U}$ in each element are represented using either standard Lagrange finite element or hierarchical node-based basis as follows:

$$\mathbf{U}_h = \sum_{i=1}^{N}\mathbf{U}_i(t)B_i(\mathbf{x}). \tag{9}$$

As a result, the unknowns to be solved are the variables at the nodes $\mathbf{U}_i$, as illustrated in Fig. 1 for linear and quadratic polynomial approximations, where polynomial solutions are dependent on the shape of elements. For example, for a linear polynomial approximation in 2D as shown in Fig. 1, a linear polynomial is used for triangular elements and the unknowns to be solved are the variables at the three vertices and a bi-linear polynomial is used for quadrilateral elements and the unknowns to be solved are the variables at the four vertices. However, the numerical polynomial solutions $\mathbf{U}$ can be expressed in other forms as well. In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the centroid of the cell. For the sake of simplicity and easy presentation of the main ideas, let us consider P2 approximation in 2D, where numerical solutions on each cell are approximated using a quadratic polynomial. If we do a Taylor series expansion at the centroid of the cell, the quadratic polynomial solutions can be expressed as follows:

$$\mathbf{U}_h = \mathbf{U}_c + \left.\frac{\partial\mathbf{U}}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial\mathbf{U}}{\partial y}\right|_c (y-y_c) + \left.\frac{\partial^2\mathbf{U}}{\partial x^2}\right|_c \frac{(x-x_c)^2}{2} + \left.\frac{\partial^2\mathbf{U}}{\partial y^2}\right|_c \frac{(y-y_c)^2}{2} + \left.\frac{\partial^2\mathbf{U}}{\partial x\partial y}\right|_c (x-x_c)(y-y_c), \tag{10}$$

which can be further expressed as cell-averaged values and their derivatives at the centroid of the cell:

$$\begin{aligned}\mathbf{U}_h = {} & \widetilde{\mathbf{U}} + \left.\frac{\partial\mathbf{U}}{\partial x}\right|_c (x-x_c) + \left.\frac{\partial\mathbf{U}}{\partial y}\right|_c (y-y_c) + \left.\frac{\partial^2\mathbf{U}}{\partial x^2}\right|_c \left(\frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d\Omega\right) \\ & + \left.\frac{\partial^2\mathbf{U}}{\partial y^2}\right|_c \left(\frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d\Omega\right) + \left.\frac{\partial^2\mathbf{U}}{\partial x\partial y}\right|_c \left((x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d\Omega\right),\end{aligned} \tag{11}$$

where $\widetilde{\mathbf{U}}$ is the mean value of $\mathbf{U}$ in this cell. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes, as shown in Fig. 2. In this case, the dimension of the polynomial space is six and the six basis functions are

$$\begin{aligned} B_1 &= 1 \\ B_2 &= x - x_c \\ B_3 &= y - y_c \\ B_4 &= \frac{(x-x_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2}d\Omega \\ B_5 &= \frac{(y-y_c)^2}{2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2}d\Omega \\ B_6 &= (x-x_c)(y-y_c) - \frac{1}{\Omega_e}\int_{\Omega_e}(x-x_c)(y-y_c)d\Omega, \end{aligned} \tag{12}$$

and the discontinuous Galerkin formulation (8) then leads to the following six equations

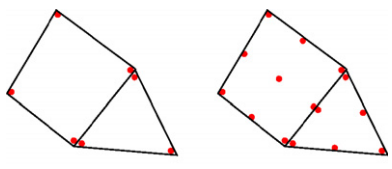

Fig. 1. Representation of polynomial solutions using finite element shape functions (a) Q1/P1 and (b) Q2/P2.

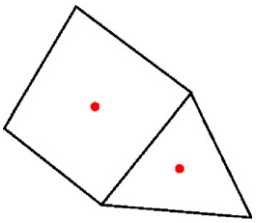

Fig. 2. Representation of polynomial solutions using a Taylor series expansion.

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_e}\widetilde{\mathbf{U}}\mathrm{d}\Omega+\int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j\mathrm{d}\Gamma=0\quad i=1, \tag{13}$$

$$\sum_{j=2}^{6}\int_{\Omega_e}B_iB_j\mathrm{d}\Omega\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix}\frac{\partial\mathbf{U}}{\partial x}|_c\\ \frac{\partial\mathbf{U}}{\partial y}|_c\\ \frac{\partial^2\mathbf{U}}{\partial x^2}|_c\\ \frac{\partial^2\mathbf{U}}{\partial y^2}|_c\\ \frac{\partial^2\mathbf{U}}{\partial x\partial y}|_c\end{pmatrix}+\int_{\Gamma_e}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_jB_i\mathrm{d}\Gamma-\int_{\Omega_e}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial B_i}{\partial x_j}\mathrm{d}\Omega=0\quad 2\leqslant i\leqslant 6. \tag{14}$$

Note that in this formulation, the equations for the cell-averaged variables are decoupled from the equations for their derivatives due to the judicious choice of the basis functions in our formulation and the fact

$$\int_{\Omega_e}B_1B_i\mathrm{d}\Omega=0,\quad 2\leqslant i\leqslant 6. \tag{15}$$

Using this formulation, the similarity and difference between DG and FV methods become clear, and the advantage of the discontinuous Galerkin methods is especially evident in comparison with the FV methods. In fact, the discretized governing equations for cell-averaged variables (13) and the assumption of a polynomial solution on each cell (10) are exactly the same for both methods. In other words, this DG method provides a unified formulation, where the existing finite volume methods can be recovered virtually. For example, the application of this DG method to the median dual control volume of a given mesh will lead to the classic vertex-centered finite volume scheme as shown in Fig. 3, while the application of this DG method to the cell itself of any given mesh will lead to the classic cell-centered finite volume scheme as shown in Fig. 2. The only difference between them is the way how to obtain the polynomial solutions, i.e., how to compute the derivatives of high-order polynomial solutions (>1). In the finite volume methods, the derivatives of the polynomial solutions of degree p are reconstructed using cell-averaged values of the flow variables in the neighboring cells, which can be obtained using either TVD/MUSCL [34,35] or ENO/WENO [1,13–15,20,31] reconstruction schemes. Unfortunately, the multi-dimensional TVD/MUSCL reconstruction schemes of arbitrary order based on the extension of one-dimensional MUSCL approach, which are praised to achieve high-order accuracy for multi-dimensional problems, suffer from two serious flaws in the context of unstructured grids: (1) uncertainty and arbitrariness in choosing the stencils and methods to compute the derivatives. This explains why a nominally second order finite volume scheme is hardly able to deliver a formal solution of second order accuracy in practice for unstructured grids. (2) Extended stencils required for the reconstruction of higher-order (>2nd) polynomial solutions. This is exactly the reason why the current finite volume methods using the TVD/MUSCL reconstruction are not practical at higher-order and have remained second order on unstructured grids. When the ENO/WENO reconstruction schemes are used for the construction of a polynomial of degree p on unstructured grids, the dimension of the polynomial space, $N=N(p,d)$ depends on the degree of the polynomials of the expansion p, and the number of spatial dimensions d. One must have three, six, and ten cells in 2D and four, ten, and twenty cells in 3D for the construction of a linear, quadratic, and cubic Lagrange polynomial, respectively. Undoubtedly, it is an overwhelmingly challenging, if not practically impossible, task

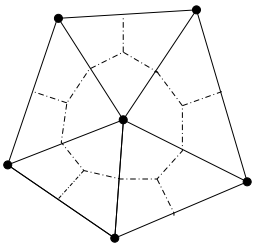

Fig. 3. Representation of polynomial solutions using a Taylor series expansion for the median dual control volume.

to judiciously choose a set of admissible and proper stencils that have such a large number of cells on unstructured grids especially for higher order polynomials and higher dimensions. This explains why the application of higher-order ENO/WENO methods hardly exist on unstructured grids, in spite of their tremendous success on structured grids and their superior performance over the MUSCL/TVD methods. Unlike the FV methods, where the derivatives are reconstructed using the mean values of the neighboring cells, the DG method solves the equations for the derivatives in a manner similar to the mean variables. This is natural, unique, compact, rigorous, and elegant mathematically in contrast with arbitrariness characterizing the reconstruction schemes used in the FV methods with respect how to compute the derivatives and how to choose the stencils. It is our belief that this is one of the main reasons why the second order DG methods are more accurate than the FV methods using either TVD/MUSCL or ENO/WENO reconstruction schemes, which will numerically be demonstrated in this paper. In addition, the perception that DG methods are more expensive than the FV methods in terms of both computing costs and storage requirements, is actually baseless, as the storage of derivatives is also required for the FV methods as well in the context of unstructured grids, and solving the discretized equations of the derivatives are relatively inexpensive due to the fact that the numerical Riemann fluxes, representing the most dominant CPU consuming operation, are already computed for the cell-averaged equations, and the fact that a quadrature-free formulation can be used to significantly reduce the number of flux evaluations, and thus the computational costs associated with numerical quadrature. Furthermore, the higher-order DG methods can be easily constructed by simply increasing the degree p of the polynomials locally, in contrast to the finite volume methods which use the extended stencils to achieve higher-order of accuracy. Many other methods such as ADER scheme by Titarev and Toro [12,32], compact finite differencing scheme by Lele, and Visbal and Gaitonde [19,36], and CE/SE scheme by Chang [9] also solve the governing equations for the derivatives instead of using the reconstruction schemes. Note that in the case of P1 approximation (piece-wise linear), this formulation results in the so-called moment approach to the approximation of the weak solution of the Euler equations, that was first introduced by van Leer [35], and was also used by Allmaras et al. [3], Agarwal et al. [2], and Huynh [16].

This formulation has a number of distinct, desirable, and attractive features and advantages in the context of DG methods. First, the same numerical polynomial solutions are used for any shapes of elements, which can be triangle, quadrilateral, and polygon in 2D, and tetrahedron, pyramid, prism, and hexahedron in 3D. Using this formulation, DG method can be easily implemented on arbitrary meshes. The numerical method based on this formulation has the ability to compute 1D, 2D, and 3D problems using the very same code, which greatly alleviates the need and pain for code maintenance and upgrade. Secondly, cell-averaged variables and their derivatives are handily available in this formulation. This makes implementation of WENO limiter straightforward and efficient [20,22,27–29], that is required to eliminate non-physical oscillations in the vicinity of discontinuities. Thirdly, the basis functions are hierarchic. This greatly facilitates implementation of p-multigrid methods [21,24] and p-refinement. Last, cell-averaged variable equations are decoupled from their derivatives equations in this formulation, which makes development of fast, low-storage implicit methods possible.

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix (14) as follows:

$$
\begin{aligned}
\widetilde{B}_1 &= 1 \\
\widetilde{B}_2 &= \frac{x-x_c}{\Delta x} \\
\widetilde{B}_3 &= \frac{y-y_c}{\Delta y} \\
\widetilde{B}_4 &= \frac{(x-x_c)^2}{2\Delta x^2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)^2}{2\Delta x^2}\,d\Omega \\
\widetilde{B}_5 &= \frac{(y-y_c)^2}{2\Delta y^2} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(y-y_c)^2}{2\Delta y^2}\,d\Omega \\
\widetilde{B}_6 &= \frac{(x-x_c)(y-y_c)}{\Delta x\Delta y} - \frac{1}{\Omega_e}\int_{\Omega_e}\frac{(x-x_c)(y-y_c)}{\Delta x\Delta y}\,d\Omega,
\end{aligned}
\tag{16}
$$

where $\Delta x = 0.5(x_{max}-x_{min})$, and $\Delta y = 0.5(y_{max}-y_{min})$, and x_{max}, y_{max}, x_{min}, and y_{min} are the maximum and minimum coordinates in the cell Ω_e in x-, and y-direction, respectively. The quadratic polynomial solutions can be rewritten as

$$
\mathbf{U}_h = \widetilde{\mathbf{U}} + \left.\frac{\partial \mathbf{U}}{\partial x}\right|_c \Delta x\widetilde{B}_2 + \left.\frac{\partial \mathbf{U}}{\partial y}\right|_c \Delta y\widetilde{B}_3 + \left.\frac{\partial^2 \mathbf{U}}{\partial x^2}\right|_c \Delta x^2\widetilde{B}_4 + \left.\frac{\partial^2 \mathbf{U}}{\partial y^2}\right|_c \Delta y^2\widetilde{B}_5 + \left.\frac{\partial^2 \mathbf{U}}{\partial x\partial y}\right|_c \Delta x\Delta y\widetilde{B}_6. \tag{17}
$$

This is especially helpful and important to remove the stiffness of the system matrix for higher-order DG approximations.

In the present work, the Riemann flux function is approximated using the HLLC approximate Riemann solver [33], which has been successfully used to compute compressible viscous and turbulent flows on both structured grids [8] and unstructured grids [26]. This HLLC scheme is found to have the following properties: (1) exact preservation of isolated contact and shear waves, (2) positivity-preserving of scalar quantity, (3) enforcement of entropy condition. In addition, the implementation of HLLC Riemann solver is easier and the computational cost is lower compared with other available Riemann solvers.

Although the domain and boundary integrals in Eqs. (13) and (14) can be approximated using the quadrature-free DG formulation [4], they are evaluated using Gauss quadrature formulas in the current implementation. The number of quad-

rature points used is chosen to integrate exactly polynomials of order of $2p$ on the reference element. In the case of linear, quadratic, and cubic shape function, the domain integrals are evaluated using three, six, and twelve points for triangles and using four, nine, and sixteen points for quadrilaterals, respectively, and the boundary integrals are evaluated using two, three, and four points, respectively for 2D. In 3D, integration over the elements for P1 and P2 approximation is performed using four and eleven quadrature points, respectively, and integration over the element boundaries for P0, P1, and P2 is performed using one, four, and seven quadrature points, respectively.

The solid wall boundary conditions in curved geometries are imposed using a novel approach [23], where the curved elements are not required. Instead an accurate representation of the boundary normals is used in the quadrature points for imposing solid wall boundary conditions for curved geometries. In our implementation, the normals in the quadrature points are computed using the local true surface normal based on the analytically defined boundary geometries.

By assembling together all the elemental contributions, a system of ordinary differential equations governing the evolution in time of the discrete solution can be written as

$$M\frac{d\mathbf{U}}{dt} = \mathbf{R}(\mathbf{U}), \tag{18}$$

where M denotes the mass matrix, $\mathbf{U}$ is the global vector of the degrees of freedom, and $\mathbf{R}(\mathbf{U})$ is the residual vector. Since the shape functions $B^p|_{\Omega_e}$ are nonzero within element Ω_e only, the mass matrix M has a block diagonal structure that couples the N degrees of freedom of each component of the unknown vector only within Ω_e. As a result, the inverse of the mass matrix M can be easily computed by hand considering one element at a time in advance.

3.2. Time integration

The semi-discrete system can be integrated in time using explicit methods. For example, the following explicit three-stage third-order TVD Runge–Kutta scheme [10,11]

$$\mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^n), \tag{19}$$

$$\mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^n + \frac{1}{4}[\mathbf{U}^{(1)} + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^{(1)})], \tag{20}$$

$$\mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^n + \frac{2}{3}[\mathbf{U}^{(2)} + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^{(2)})], \tag{21}$$

is widely used to advance the solution in time. This method is linearly stable for a Courant number less than or equal to $1/(2p+1)$. The inefficiency of the explicit method due to this rather restrictive CFL condition motivates us to develop the p-multigrid method [21,24] to accelerate the convergence of the Euler equations to a steady-state solution. Unlike the traditional p-multigrid methods where the same time integration scheme is used on all approximation levels, this p-multigrid method uses the above multi-stage Runge–Kutta scheme as the iterative smoother on the higher level approximations ($p > 0$), and a matrix-free implicit SGS method as the iterative smoother on the lowest level approximation ($p = 0$). As a result, this p-multigrid method has two remarkable features: (1) Low memory requirements. The implicit smoothing is only used on the lowest level P_0, where the storage requirement is not as demanding as on the higher level; (2) Natural extension to flows with discontinuities such as shock waves and contact discontinuities. A monotonic limiting procedure required to eliminate spurious oscillations of high-order approximations in the vicinity of discontinuities can be easily implemented as a post-processing filter (smoothing) in an explicit method, but not in an implicit method. This p-multigrid is found to be orders of magnitude faster than its explicit counterpart without significant increase in memory.

3.3. Hermite WENO reconstruction

The DG method described above will produce non-physical oscillations and even nonlinear instability for flows with strong discontinuities. A common solution to this problem is to use a slope limiter as in the finite volume methods. Unfortunately, DGM are very sensitive to the treatment and implementation of the slope limiters [23]. Slope limiters frequently identify regions near smooth extrema as requiring limiting, and this typically results in a reduction of the optimal high-order convergence rate. Alternatively, the ENO/WENO methodology can be used as a limiter for the discontinuous Galerkin methods, as it is more robust than the slope limiter methodology, and can achieve both uniform high order accuracy and a sharp, ENO shock transitions. This is accomplished by replacing the solution polynomials with reconstructed polynomials, which maintain the original cell averages of flow variables (full conservation of mass, momentum, and total energy), have the same high-order of accuracy as before in the regions where the solution is smooth, but oscillation-free behavior in the vicinity of discontinuities.

For the construction of a polynomial of degree p, the dimension of the polynomial space, $N = N(p,d)$ depends on the degree of the polynomials of the expansion p, and the number of spatial dimensions d. One must have three, six, and ten cells in 2D and four, ten, and twenty cells in 3D for the construction of a linear, quadratic, and cubic Lagrange polynomial, respectively. Undoubtedly, it is an overwhelmingly challenging, if not practically impossible, task to judiciously choose a set of admissible and proper stencils that have such a large number of cells on unstructured grids especially for higher order poly-

nomials and higher dimensions. This explains why the application of higher-order ENO/WENO methods hardly exist on unstructured grids, in spite of their tremendous success on structured grids and their superior performance over the MUSCL/TVD methods.

The inapplicability and impracticability of the WENO limiters on unstructured grids have motivated us to develop a Hermite polynomial WENO limiter on unstructured grids [22]. The idea comes from the observation that the number of cells needed for a polynomial reconstruction can be significantly reduced, if a Hermite polynomial is used instead of a Lagrange one. This is only possible, if the derivatives of the function to be reconstructed are known on the cells. Fortunately, this is

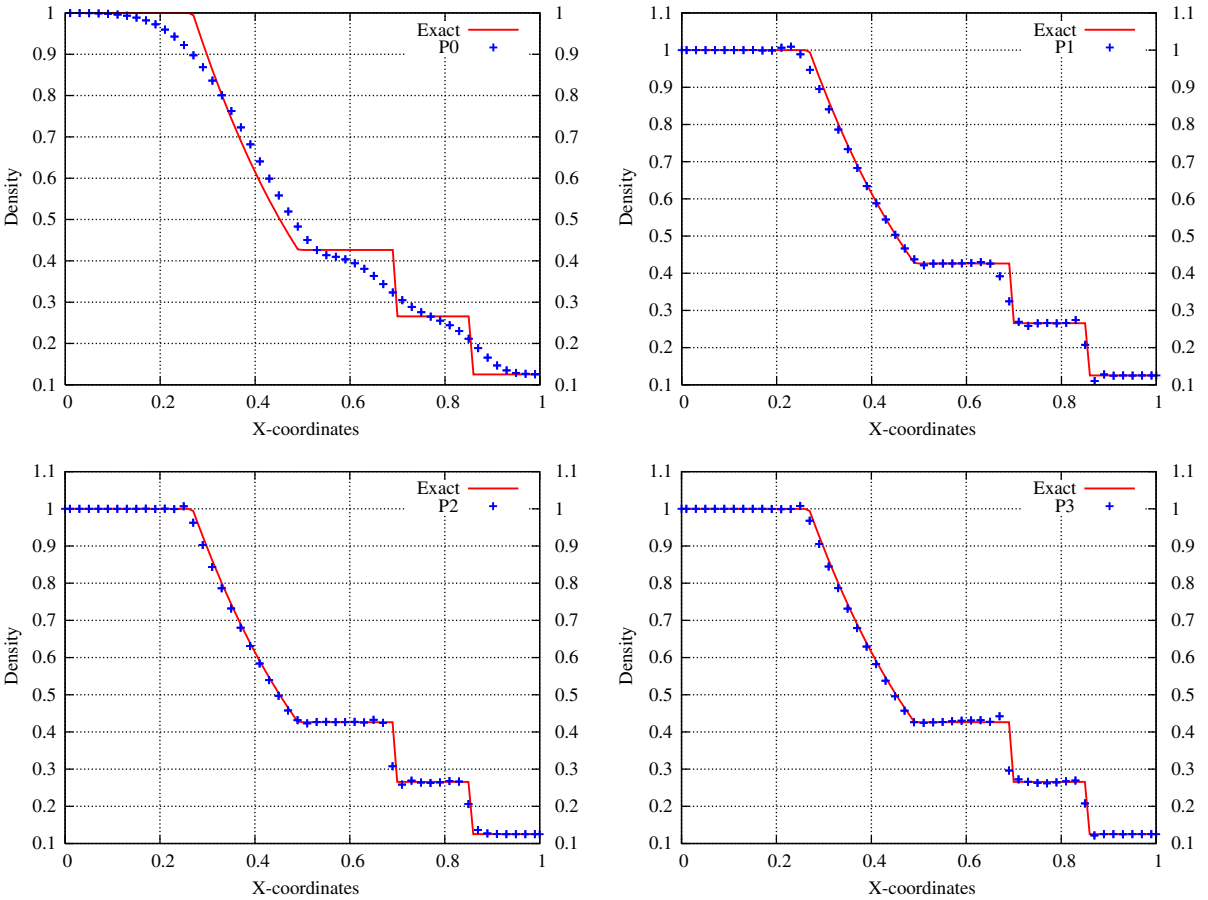

Fig. 4. Comparison of computed density profile for Sod shock tube problem obtained by unlimited DG(P0), DG(P1), DG(P2), and DG(P3) solutions with the analytical solution.

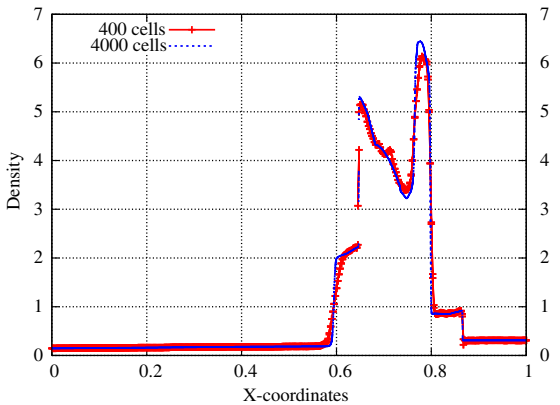

Fig. 5. Comparison of the computed DG(P1) solution between coarse and fine meshes for Woodward–Colella blast wave problem.

exactly the case for the discontinuous Galerkin methods where the derivatives are handily available on each cell. This HWENO limiter [22] has been used to compute a variety of compressible flow problems for a wide range of flow conditions in both 2D and 3D configurations. The superior robustness of the HWENO limiter for the DG methods has been demonstrated in terms of both solution accuracy and convergence performance in comparison with TVD/MUSCL type limiters. The numerical experiments have indicated that using this HWENO limiter, the accuracy of the second order DG solutions is comparable to, if not better than, that of the third-order WENO solutions using the same mesh resolution. The extension of this HWENO limiter to arbitrary grids is easy and straightforward.

4. Numerical examples

All computations are performed on a Dell XPS M1210 laptop computer with 1 GBytes memory running the Suse 10.2 Linux operating system. The explicit three-stage third-order TVD Runge–Kutta scheme is used for unsteady flow computations and the *p*-multigrid for steady-state flow problems. 1D, 2D, and 3D examples are presented to demonstrate the versatility of the DG method. An elaborate and well-tested finite volume code [25,26] is used as a reference to quantitatively compare the accuracy of the DG method, for steady-state solutions, although it is not our objective to compare the performance of FV and DG methods in terms of computational efficiency and numerical accuracy. To plot a flow variable on the surface of the solid body in 2D, its values at two end points of a face on the solid body are drawn using a line. This is the most accurate way to represent P1 solution for profile plotting, as the solution is linear on each face and multiple values exist for a vertex due to the discontinuous representation of DG solution. For unsteady flow problems, the WENO unstructured grid solutions [15] are used as a reference to qualitatively compare the accuracy of the present DG method.

4.1. Sod shock tube problem

The shock tube problem constitutes a particularly interesting and difficult test case, since it presents an exact solution to the full system of one-dimensional Euler equations containing simultaneously a shock wave, a contact discontinuity, and an

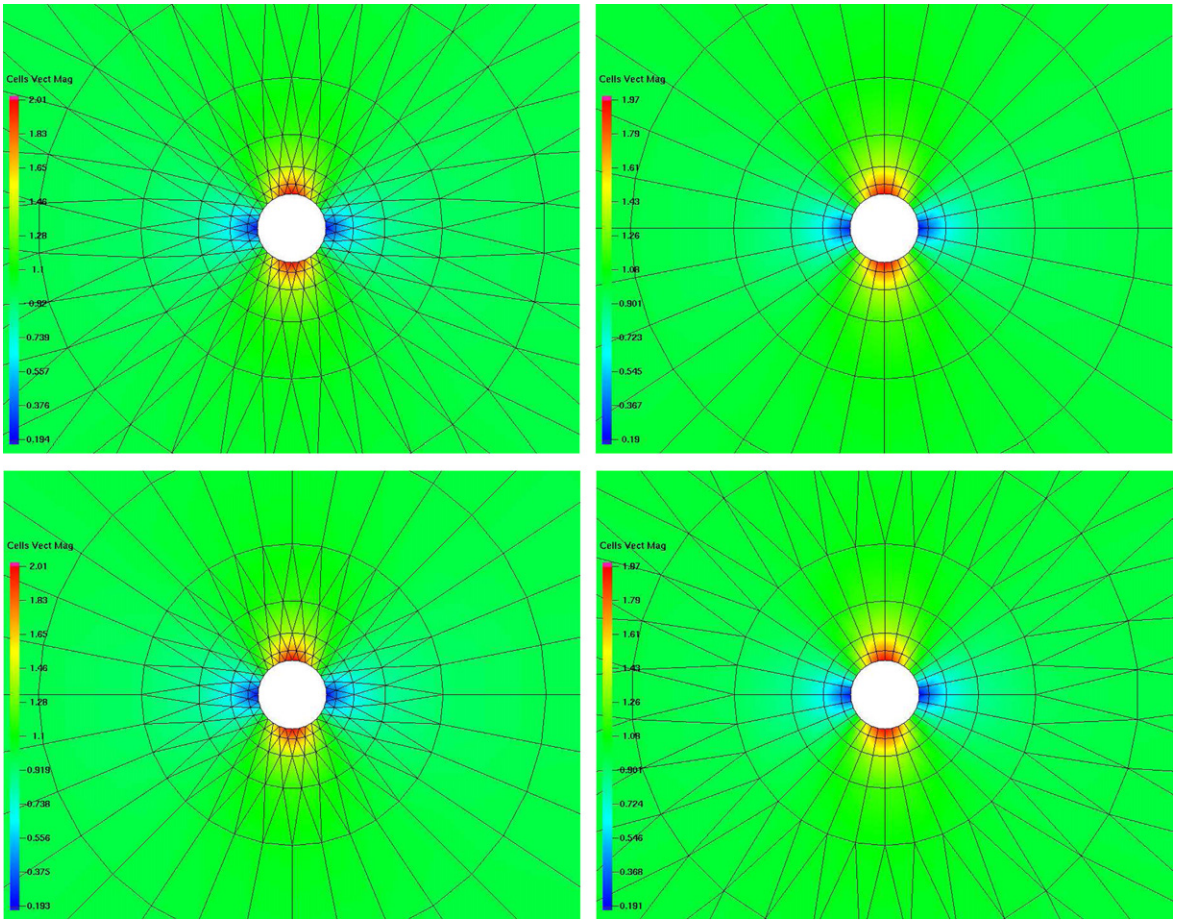

Fig. 6. The computed velocity contours in the flow field using DG(P2) solution on different types of grids for subsonic flows past a circular cylinder.

expansion fan. This test case is chosen to demonstrate the robustness of the DG methods in comparison with the finite volume methods. The initial conditions in the present computation are the following: $\rho = 1.000,\ u_L = 0,\ p = 1.0$ for $x \in [0.0, 0.5],\ \rho = 0.125,\ u_R = 0,\ p = 0.1$ for $x \in [0.5, 1.0]$. Fig. 4 shows a comparison of the density profiles obtained by the unlimited DG(P0), DG(P1), DG(P2), and DG(P3) and the exact solutions, respectively. This is a 2D simulation of a 1D problem. The mesh consists of 50 cells in the *X*-direction and 1 cell in the *Y*-direction. As expected, unlimited higher-order DG(P1), DG(P2), and DG(P3) solutions exhibit small oscillations in the vicinity of discontinuities. A higher order DG solution yields a sharper resolution for both contact discontinuity and shock wave than a lower order DG solution. Note that the reconstruction scheme used in the finite volume methods will unavoidably generate negative values for density and energy in this case, leading to a breakdown of the finite volume solution process. This example demonstrates clearly the superior robustness of the discontinuous Galerkin methods in comparison with the second order finite volume methods.

4.2. Woodward–Colella blast wave problem

In this example, two interacting blast waves given by Woodward and Collella [37] are computed using DG(P1) method, where HWENO limiter is used to eliminate spurious oscillations in the vicinity of discontinuities. The initial conditions for the Woodward–Collela blast wave are the following: $\rho = 1.000,\ u = 0.0,\ p = 1000.0$ for $x \in [0.0, 0.1],\ \rho = 1.00,\ u = 0,\ p = 0.01$ for $x \in [0.1, 0.9]$, and $\rho = 1.00,\ u = 0,\ p = 100.$ for $x \in [0.0, 1.0]$. Four hundred cells are used in this case, and the computed results are shown in Fig. 5 for the density distribution at $t = 0.038$, where the dashed lines are obtained from the same calculation with 4000 cells.

4.3. Subsonic flows past a circular cylinder

This is a well-known test case: subsonic flow past a circular cylinder at a Mach number of $M_\infty = 0.38$. Numerical solutions to this problem are computed using DG(P2) method on four types of grids: a triangular element mesh, a quad-

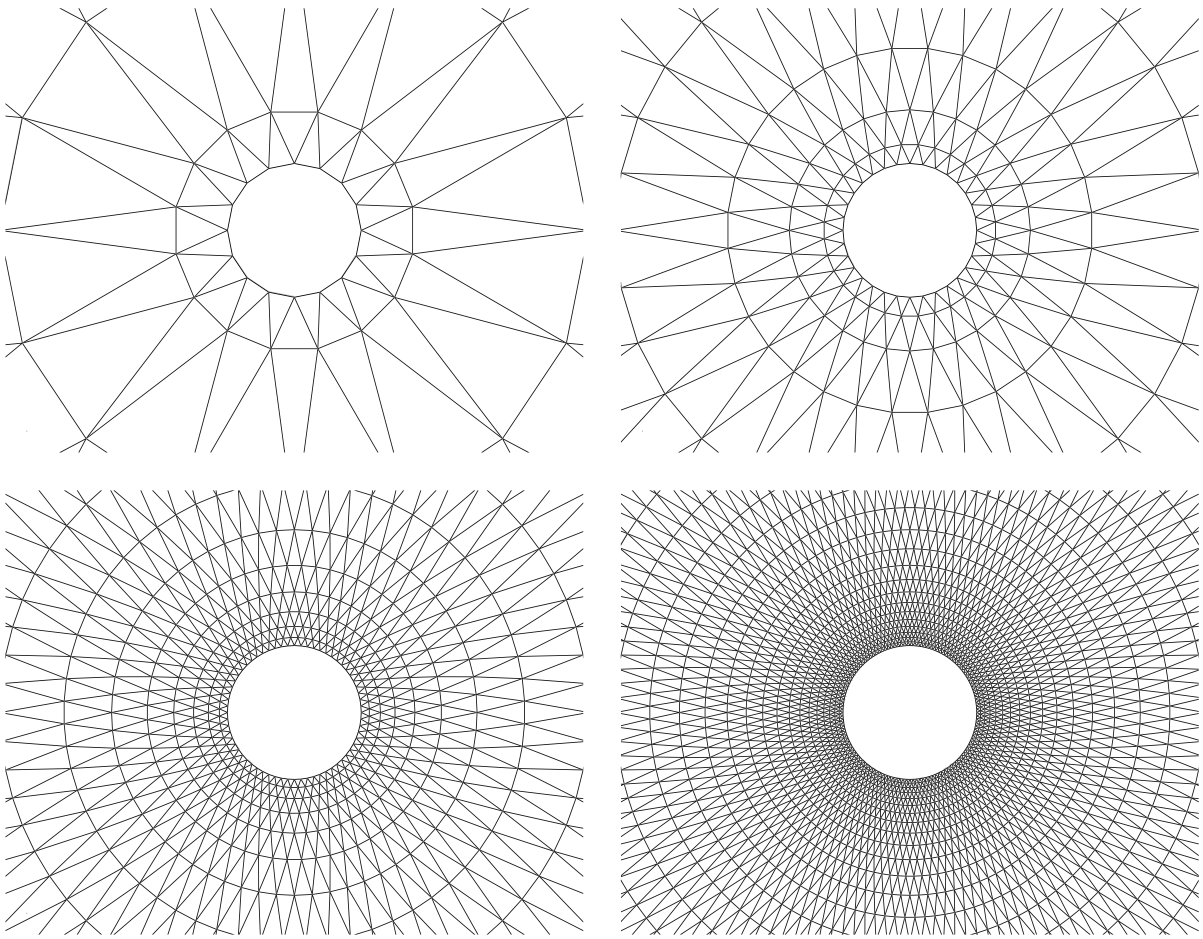

Fig. 7. Sequences of four successively globally refined meshes 16×5, 32×9, 64×17, 128×33 for computing subsonic flow past a circular cylinder.

rilateral element mesh, two hybrid triangular/quadrilateral element meshes to demonstrate the applicability of the DG method for any type of grids. All four meshes consist of 32 points in the circular direction and 9 points in the radial

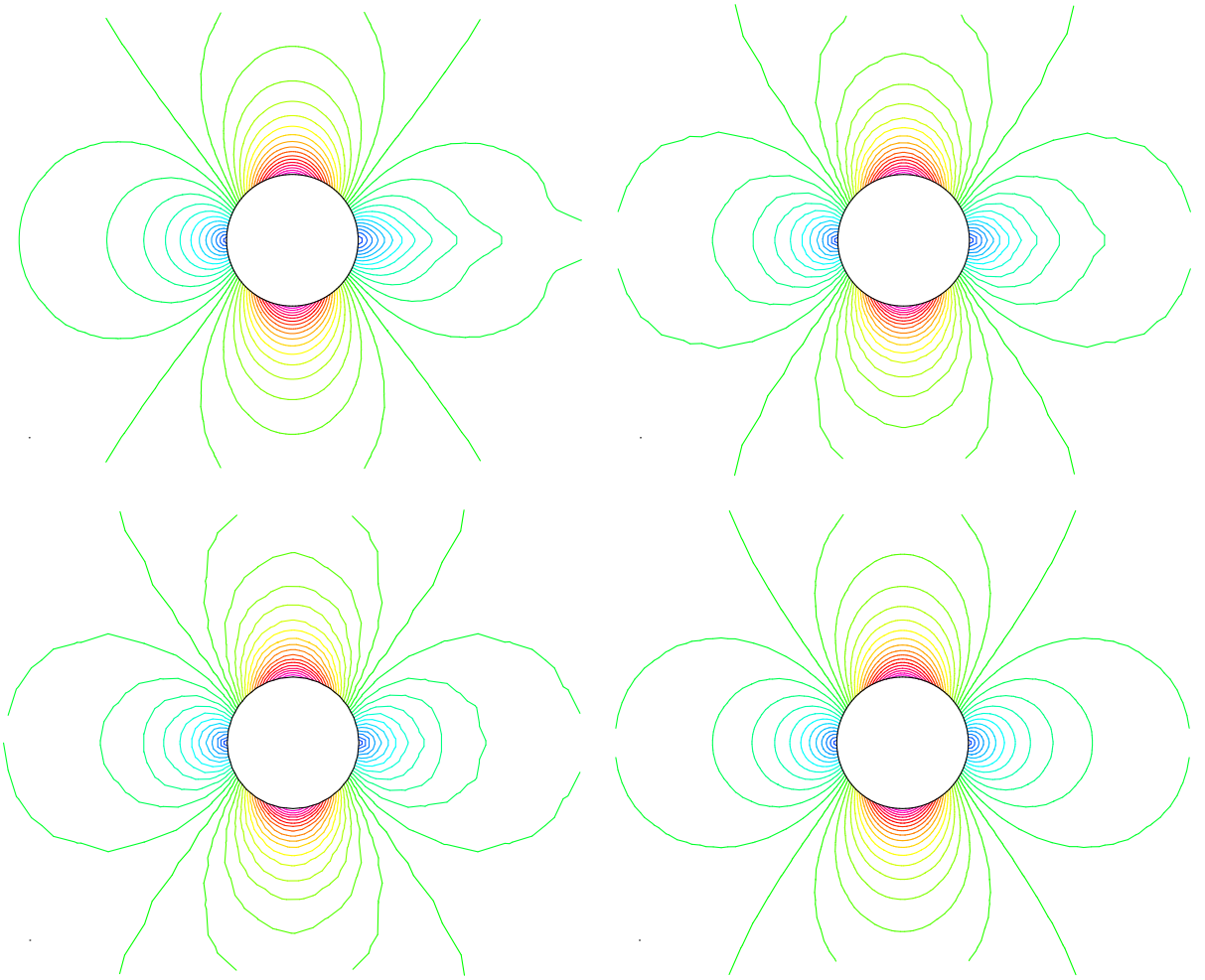

Fig. 8. Computed Mach number contours obtained by FV(P1) solution on 128 × 33 mesh (top left), DG(P1) solution on 64 × 17 mesh (top right), DG(P2) solution on 32 × 9 mesh (bottom left), and DG(P2) solution on 64 × 17 mesh (bottom right) for subsonic flow past a circular cylinder at $M_\infty = 0.38$.

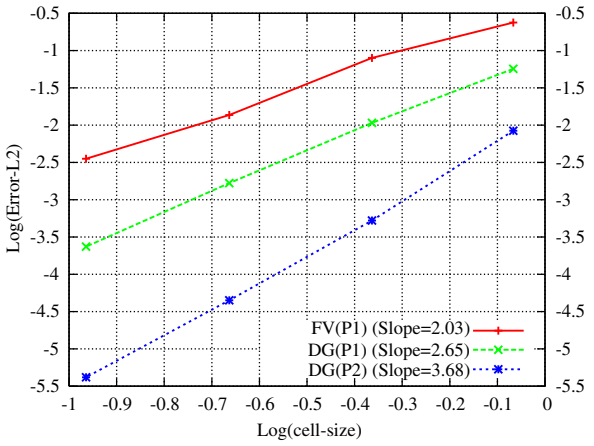

Fig. 9. Results of grid-refinement study for flow past a cylinder obtained by the FV(P1), DG(P1), and DG(P2) methods.

direction. The computed velocity contours in the flow field and the grids used in the computation are shown in Fig. 6. For this test case, a grid convergence study is also conducted to numerically compare accuracy between DG and FV methods. Fig. 7 shows four successively refined o-type grids having 16×5, 32×9, 64×17, and 128×33 points respectively, used in this study. The first number refers to the number of points in the circular direction, and the second designates the number of concentric circles in the mesh. Numerical solutions to this problem are computed using FV(P1), DG(P1), and DG(P2) methods on these four grids to obtain quantitative measurement of the order of accuracy and discretization errors. Judging the quality of the numerical solutions by the symmetry of the computed Mach number contours in the flow field shown in Fig. 8, one can see that DG(P2) solution on 32×8 mesh is actually more accurate than DG(P1) solution on 64×17 mesh, which in turn is more accurate than the FV(P1) solution on 128×33 mesh. In fact, the second order DG solutions on any given mesh are consistently more accurate than the second order finite volume solutions on a mesh twice as fine, as witnessed by Fig. 9, where spatial accuracy details of each method for this numerical experiment is provided. Note that the entropy production is served as a criterion to measure the accuracy and quality of the numerical solutions.

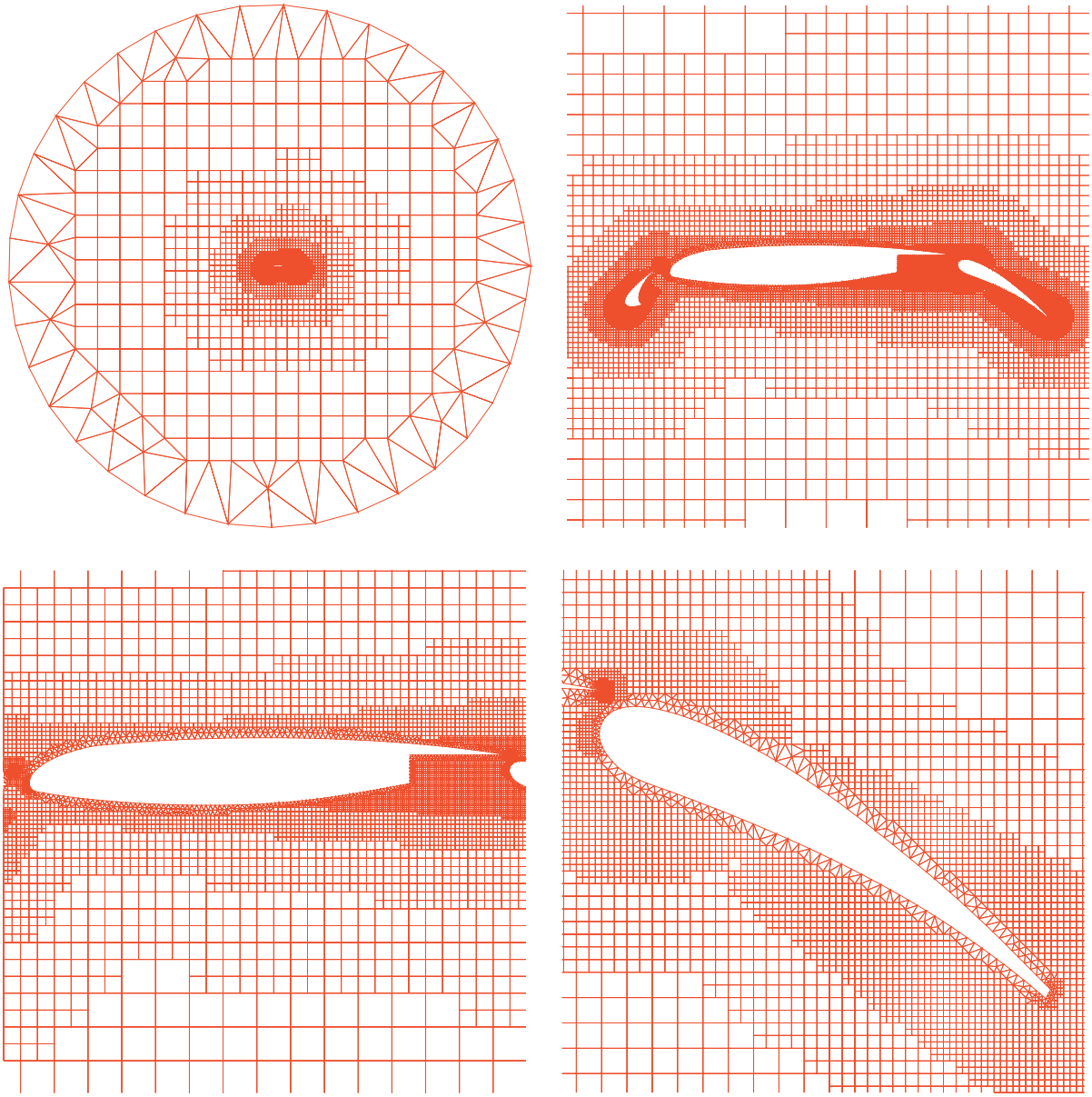

Fig. 10. The hybrid unstructured Cartesian and triangular grid for computing subsonic flows past a 3-element airfoil.

4.4. Subsonic flow past a three-element airfoil

A subsonic flow past 3-element airfoil is presented in this test case. A hybrid unstructured Cartesian and triangular grid [38] is used in this computation in order to demonstrate the applicability of the DG method for non-conforming grids. The mesh, shown in Fig. 10 consists of 2681 triangles and 13,892 Cartesian cells. The computation is performed using DG(P1)

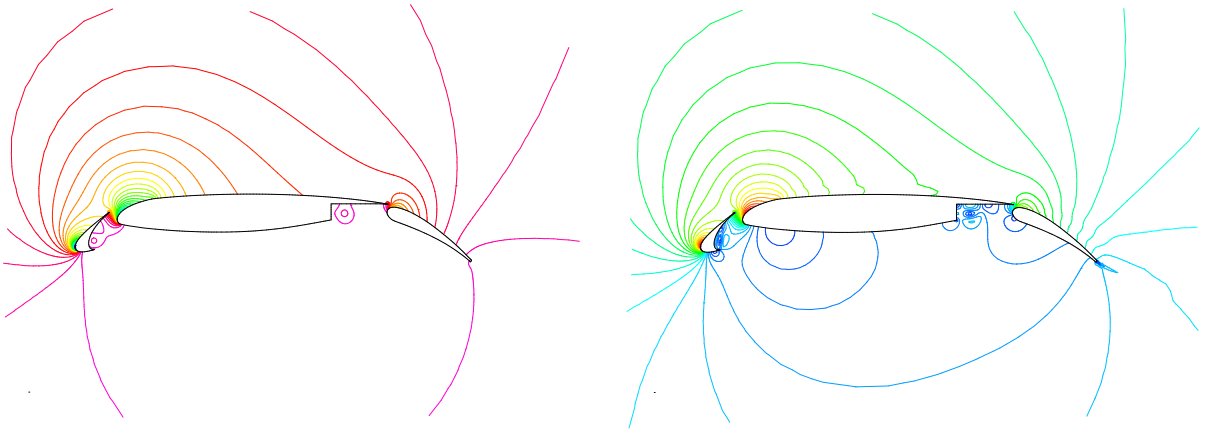

Fig. 11. The computed pressure (left) and Mach number (right) contours in the flow field for subsonic flow past a 3-element airfoil using DG(P1) solution.

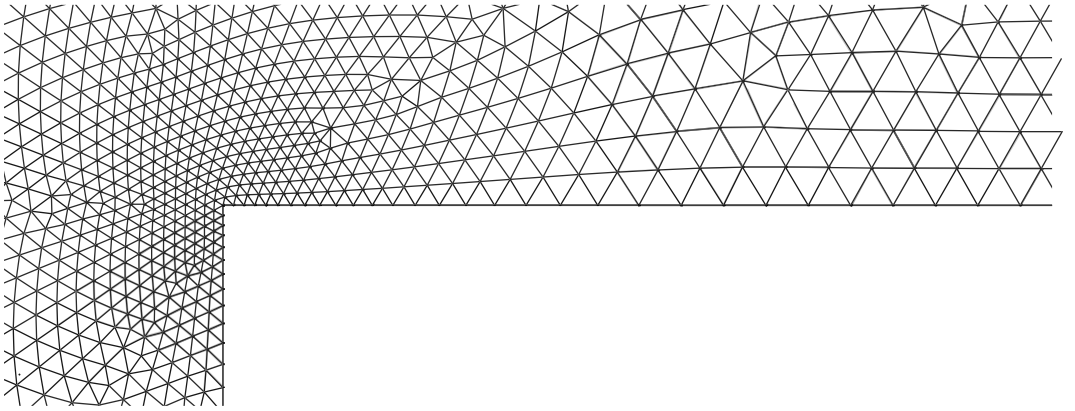

Fig. 12. Unstructured mesh (*n*elem = 10,245, *n*poin = 5294, *n*boun = 341) used for computing supersonic flow in a wind tunnel with a step at $M_\infty = 3$.

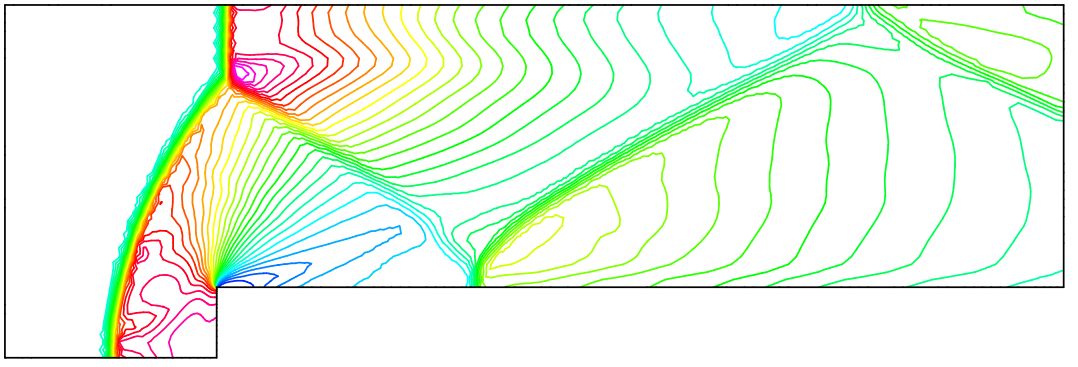

Fig. 13. Computed density contours by the present second order DG method with the HWENO limiter for supersonic flow in a wind tunnel with a step at $M_\infty = 3$.

method at a Mach number of 0.2, and an angle of attack 16°. Fig. 11 shows the computed pressure and Mach number contours in the flow field, respectively.

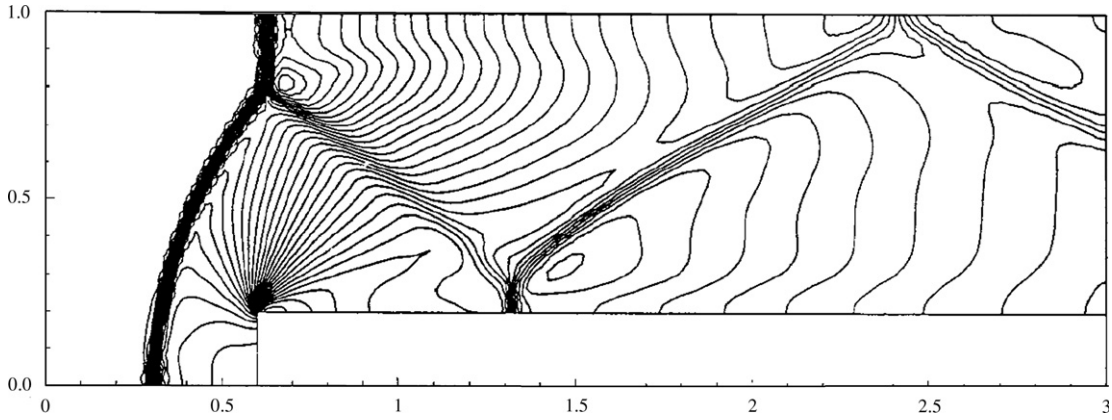

Fig. 14. Computed density contours by a third-order WENO method for supersonic flow in a wind tunnel with a step at $M_\infty = 3$.

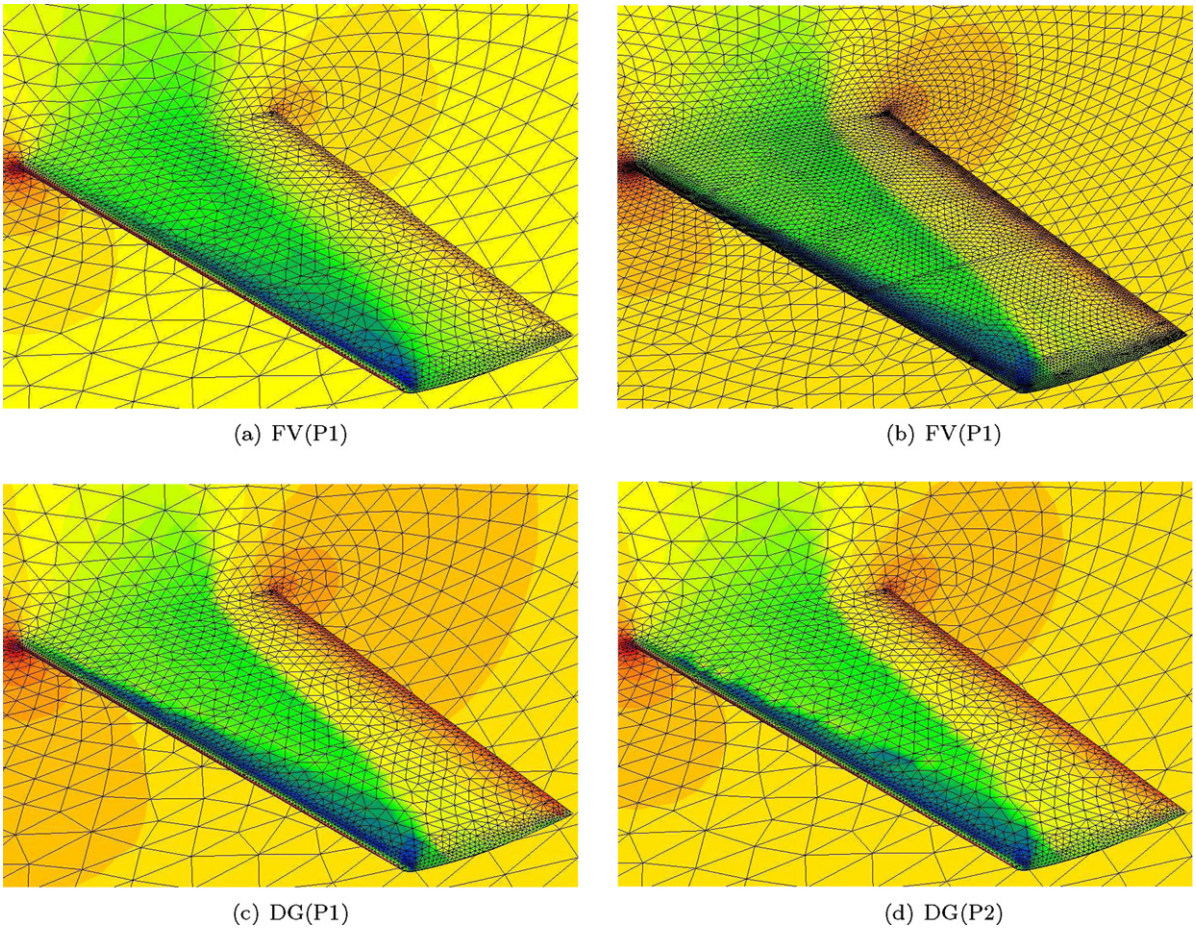

Fig. 15. Computed pressure contours on the unstructured surface mesh obtained by the FV(P1) solution on the coarse mesh (top left, *n*elem = 136,705, *n*poin = 25,616, *n*boun = 5017), the FV(P1) solution on the fine mesh (top right, *n*elem = 710,971, *n*poin = 131,068, *n*boun = 20,659), the DG(P1) solution on the coarse mesh, and the DG(P2) solution on the coarse mesh for transonic flow past an ONERA M6 wing at $M_\infty = 0.84, \alpha = 3.06^\circ$.

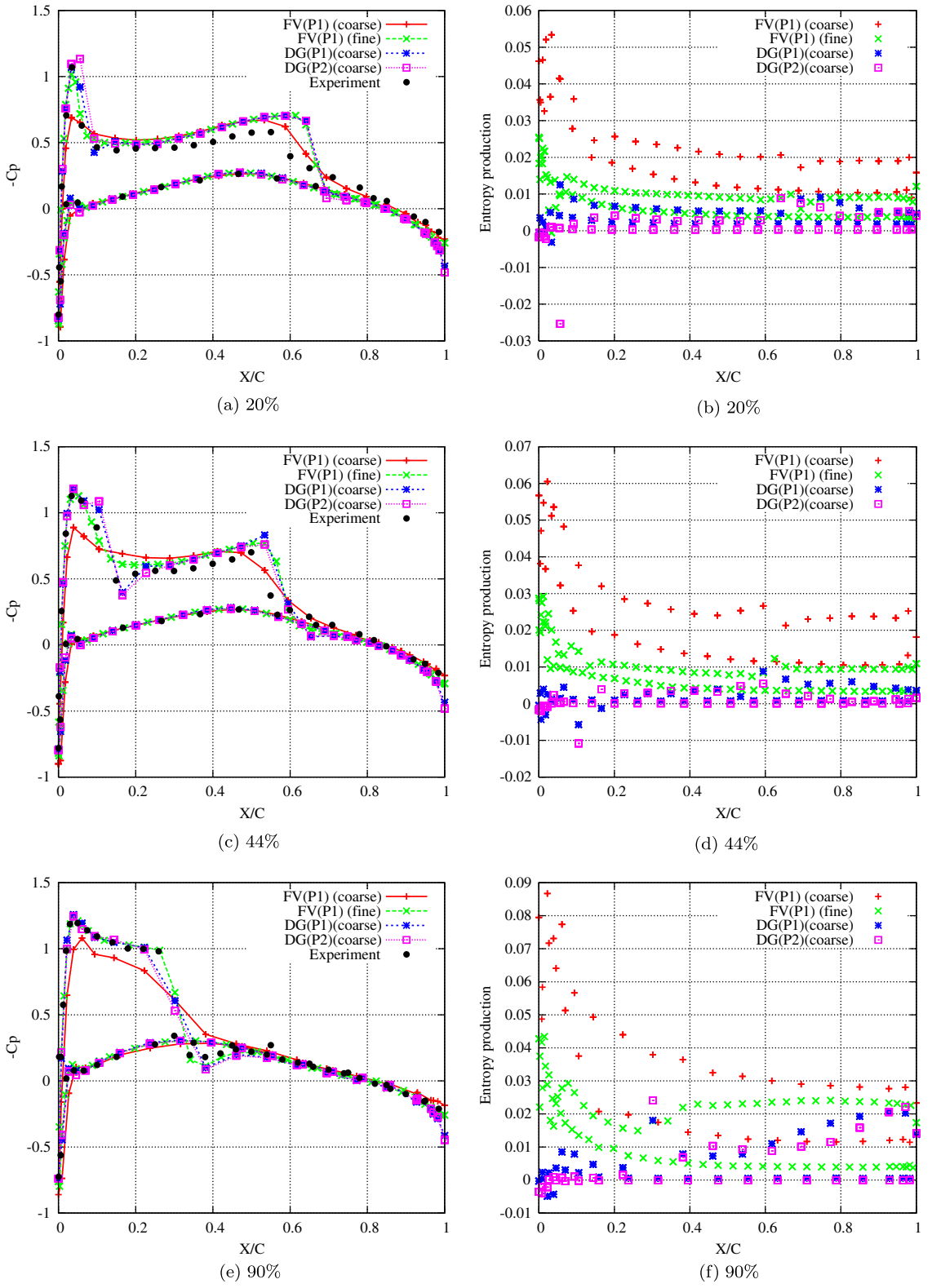

Fig. 16. Comparison of computed pressure coefficient (left) and entropy production (right) distributions for wing section at different semi-span locations obtained by the FV solutions on the coarse mesh and fine mesh and DG(P1) and DG(P2) solutions on the coarse mesh with experimental data for transonic flow past an ONERA wing at $M_\infty = 0.84, \alpha = 3.06^\circ$.

4.5. A Mach 3 wind tunnel with a Step

The test case is a classical example for testing the accuracy of numerical schemes for computing unsteady shock waves. The problem under consideration is a Mach 3 flow in a wind tunnel with a step. The tunnel is 1 length unit high and 3 length units long. The step is 0.2 length units high and is located at 0.6 length units from the left-hand end of the tunnel. The boundary conditions are that of a reflecting surface along the walls of the tunnel, and characteristic boundary conditions are used at the inlet and exit. The initial condition is the uniform flow, where $(\rho, u, v, p) = (1.4, 3, 0, 1)$. The numerical experiment is performed on a coarse grid, which has about the same mesh size as that used on the reference 13, where an element size of 0.025 is used everywhere else while an element size of one-quarter of that, i.e., 0.00625 is used in the corner. The resulting unstructured triangular mesh has 10,245 elements, 5294 grid points, and 341 boundary points. Figs. 12–14 show the mesh used in the computation, the computed density number contours obtained by the DG method and a third-order WENO method [15], respectively. Note that 30 lines are plotted from 0.32 to 6.15 for both density contours. One can see that the shock resolution of the 3rd order WENO scheme is slightly more diffusive than the present second DG scheme, and the slip line coming from the lambda shock is also more visible in the 2nd DG solution than 3rd order WENO solution, qualitatively demonstrating that the present second order DG solution is as accurate as, if not more accurate than, the third-order WENO solution.

4.6. Transonic flows past a ONERA M6 wing

A transonic flow over the ONERA M6 wing geometry is considered in this test case. The M6 wing has a leading edge sweep angle of 30°, an aspect of 3.8, and a taper ratio of 0.562. The airfoil section of the wing is the ONERA "D" airfoil, which is a 10% maximum thickness-to-chord ratio conventional section. The flow solutions are presented at a Mach number of 0.84 and an angle of attack of 3.06° using the FV method on a coarse mesh and a fine mesh and DG(P1) and DG(P2) methods on the coarse mesh, respectively. The coarse mesh contains 136,705 elements, 25,616 points, and 5017 boundary points, and the fine one 1,092,270 elements, 192,727 points, and 20,066 boundary points. Fig. 15 shows the computed the pressure contours on the upper wing surface obtained by these four solutions, respectively. The computed pressure coefficient and entropy production distributions obtained by these four solutions are compared at three span-wise stations in Fig. 16, where experimental data for the pressure coefficients is also given as a reference. The FV(P1) solution on the coarse mesh is so dissipative that it has the difficulty to capture the suction peak at the leading edge due to a lack of mesh resolution. The FV(P1) solution is significantly improved by doubling the mesh size, even though the solution is still not as good as the one obtained by DG(P1) solution on the coarse mesh by judging the entropy production on the surface of the wing. Note that the entropy production corresponds directly to the error of the numerical methods, as it should be zero everywhere with exception of shock waves where it should increase. The DG(P2) solution provides a further improvement over the DG(P1) solution, although the difference is relatively small, indicating that the obtained solution is order-independent, i.e., the solution is convergent. Note that the results obtained by the two DG solutions and the FV solution on the fine mesh compare closely with experimental data, except at the root stations, due to a lack of viscous effects. Convergence histories versus time steps and CPU time for these solutions are shown in Figs. 17 and 18, respectively, to demonstrate an indicative performance of the DG method.

4.7. Transonic flows past a Boeing 747 Aircraft

Finally, an illustrative example is presented to demonstrate that the developed method can be applied to problems of scientific and industrial interests. The computation is performed on a complete Boeing 747 aircraft using the DG(P1) method

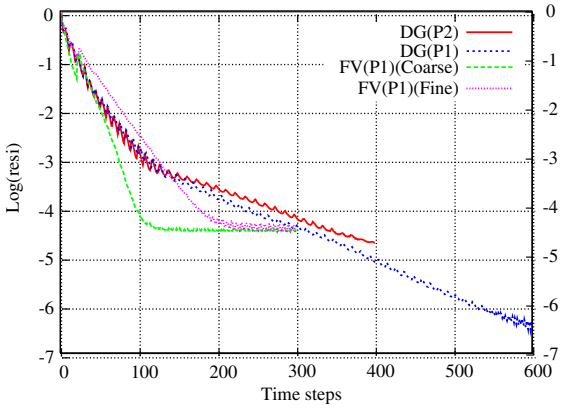

Fig. 17. Convergence history versus time steps for the DG(P1), DG(P2), FV(P1) solutions on the coarse grid, and FV(P1) solution on the fine grid for transonic flow past an ONERA wing at $M_\infty = 0.84, \alpha = 3.06°$.

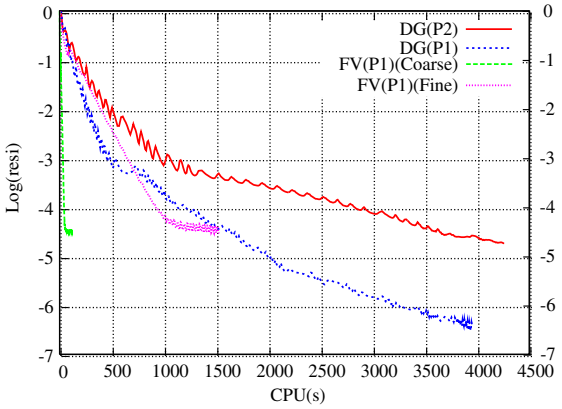

Fig. 18. Convergence history versus cpu time for the DG(P1), DG(P2), FV(P1) solutions on the coarse grid, and FV(P1) solution on the fine grid for transonic flow past an ONERA wing at $M_\infty = 0.84, \alpha = 3.06^\circ$.

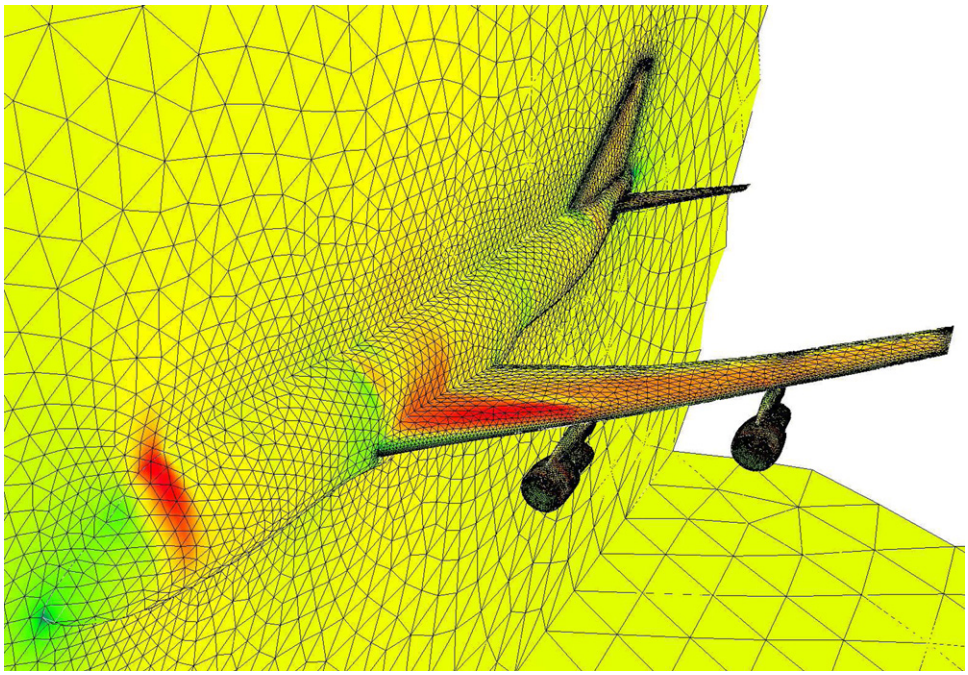

Fig. 19. Computed Mach number contours and unstructured surface mesh for transonic flow past a complete B747 aircraft(*n*elem = 489,376, *n*poin = 91,911, *n*boun = 18,261).

with HWENO limiter. The 747 configuration includes the fuselage, the wing, horizontal and vertical tails, underwing pylons, and flow-through engine nacelle. The mesh, used in the computation, contains 91,911 grid points, 489,376 elements and 18,261 boundary points for the half-span airplane. A solution is computed for the aircraft at a free stream of Mach number of 0.84 and an angle of attack of 2.73°. The CPU time required for this computation is 2516 s after a decrease of a three order-of-magnitude in the L_2 norm of the density residual. The computed Mach number contours on the surface of the airplane, along with the surface mesh, are shown in Fig. 19. For such a level of grid resolution, the shock waves are captured well, thus confirming the accuracy and robustness of the HWENO limiter for computing complicated flows of practical importance.

5. Concluding remarks

A discontinuous Galerkin formulation based on a Taylor basis has been presented for solving the compressible Euler equations. This formulation is able to provide a unified framework, where the finite volume schemes can be recovered as

special cases of the discontinuous Galerkin method by choosing reconstruction schemes to compute the derivatives, offer the insight why the DG methods are a better approach than the finite volume methods based on either TVD/MUSCL reconstruction or essentially non-oscillatory (ENO)/weighted essentially non-oscillatory (WENO) reconstruction, and has a number of distinct, desirable, and attractive features, which can be effectively used to address some of shortcomings of the DG methods. The developed method is used to compute a variety of both steady-state and time-accurate flow problems on arbitrary grids. The numerical results demonstrated the superior accuracy of this discontinuous Galerkin method in comparison with a second order finite volume method and a third-order WENO method, indicating its promise and potential to become not just a competitive but simply a superior approach than its finite volume and ENO/WENO counterparts for solving flow problems of scientific and industrial interest. The versatility of this DG method is also demonstrated in its ability to compute 1D, 2D, and 3D problems using the very same code, greatly alleviating the need and pain for code maintenance and upgrade.

References

[1] R. Abgrall, On essential non-oscillatory schemes on unstructured meshes, Journal of Computational Physics 114 (1994) 45–58.
[2] R.K. Agarwal, D.W. Halt, A compact high-order unstructured grids method for the solution of Euler equations, International Journal of Numerical Methods in Fluids 31 (1999) 121–147.
[3] S.R. Allmaras, M.B. Giles, A second-order flux split scheme for the unsteady 2D Euler equations on arbitrary meshes, AIAA Paper (1987) 87–1119.
[4] H.L. Atkins, C.W. Shu, Quadrature free implementation of the discontinuous Galerkin method for hyperbolic equations, AIAA Journal 36 (5) (1998) 775–782.
[6] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, Journal of Computational Physics 138 (1997) 251–285.
[7] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, Journal of Computational Physics 131 (1997) 267–279.
[8] P. Batten, M.A. Leschziner, U.C. Goldberg, Average-state Jacobians and implicit methods for compressible viscous and turbulent flows, Journal of Computational Physics 137 (1997) 38–78.
[9] S.C. Chang, The method of space–time conservation element and solution element – a new approach for solving the Navier–Stokes and Euler equations, Journal of Computational Physics 119 (2) (1995) 295–324.
[10] B. Cockburn, G. Karniadakis, C.W. Shu, The development of discontinuous Galerkin method, in: B. Cockburn, G.E. Karniadakis, C.W. Shu (Eds.), Discontinuous Galerkin Methods, Theory, Computation, and Applications, Lecture Notes in Computational Science and Engineering, vol. 11, Springer-Verlag, New York, 2000, pp. 5–50.
[11] B. Cockburn, C.W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional system, Journal of Computational Physics 141 (1998) 199–224.
[12] M. Dumbser, M. Kaser, V.A. Titarev, E.F. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, Journal of Computational Physics 226 (1) (2007) 204–243.
[13] O. Friedrich, Weighted essential non-oscillatory schemes for the interpolation of mean values on unstructured grids, Journal of Computational Physics 144 (1998) 194–212.
[14] A. Harden, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high-order accurate essential non-oscillatory schemes III, Journal of Computational Physics 71 (1987) 231–303.
[15] C. Hu, C.W. Shu, Weighted essential non-oscillatory schemes on unstructured triangular meshes, Journal of Computational Physics 150 (1999) 97–127.
[16] H.T. Huynh, An upwind moment scheme for conservation laws, in: Proceedings of the Third International Conference on Computational Fluid Dynamics, Toronto, Canada, 12–16, July 2004.
[17] D. Lacasse, A. Garon, D. Pelletier, Development of an adaptive discontinuous Galerkin finite element method for solving advection–reaction equations, Computer Methods in Applied Mechanics and Engineering 196 (2007) 2071–2083.
[18] D. Lacasse, A. Garon, D. Pelletier, Mechanical hemolysis in blood flow: user-independent predictions with the solution of a partial differential equation, Computer Methods in Biomechanics and Biomedical Engineering 10 (1) (2007) 1–12.
[19] S.K. Lele, Compact finite difference schemes with spectral-like resolution, Journal of Computational Physics 103 (1) (1992) 16–42.
[20] X. Liu, S. Osher, T.F. Chen, Weighted essential non-oscillatory schemes, Journal of Computational Physics 115 (1994) 200–212.
[21] H. Luo, J.D. Baum, R. Löhner, A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, Journal of Computational Physics 211 (2) (2006) 767–783.
[22] H. Luo, J.D. Baum, R. Löhner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, Journal of Computational Physics 225 (1) (2007) 686–713.
[23] H. Luo, J.D. Baum, R. Löhner, On the computation of steady-state compressible flows using a discontinuous Galerkin method, International Journal for Numerical Methods in Engineering 73 (5) (2008) 597–623.
[24] H. Luo, J.D. Baum, R. Löhner, Fast, *p*-multigrid discontinuous Galerkin method for compressible flows at all speeds, AIAA Journal 46 (3) (2008) 635–652.
[25] H. Luo, J.D. Baum, R. Löhner, A fast, matrix-free implicit method for compressible flows on unstructured grids, Journal of Computational Physics 146 (2) (1998) 664–690.
[26] H. Luo, J.D. Baum, R. Löhner, High-Reynolds number viscous flow computations using an unstructured-grid method, Journal of Aircraft 42 (2) (2005) 483–492.
[27] J. Qiu, C.W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, SIAM Journal of Scientific Computing 26 (2005) 907–929.
[28] J. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one dimensional case, Journal of Computational Physics 193 (1) (2004) 115–135.
[29] J. Qiu, C.W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two dimensional case, Computers and Fluids 34 (2005) 642–663.
[31] T. Sonar, On the construction of essential non-oscillatory finite volume approximation to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy, and stencil selection, Computer Methods in Applied Mechanics and Engineering 140 (1997) 157–182.
[32] V.A. Titarev, E.F. Toro, ADER schemes for three-dimensional nonlinear hyperbolic systems, Journal of Computational Physics 204 (2) (2005) 715–736.
[33] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL-Riemann solver, Shock Waves 4 (1994) 25–34.
[34] B. van Leer, Towards the ultimate conservative difference scheme, II. Monotonicity and conservation combined in a second order scheme, Journal of Computational Physics 14 (1974) 361–370.
[35] B. van Leer, Towards the ultimate conservative difference scheme IV. A new approach to numerical convection, Journal of Computational Physics 23 (1977) 276–298.
[36] M.R. Visbal, D.V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, Journal of Computational Physics 181 (1) (2002) 155–185.

[37] P.R. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, Journal of Computational Physics 54 (1984) 115–173.
[38] H. Luo, G. Chen, R. Löhner, A hybrid unstructured cartesian and triangular/tetrahedral grid generation method for complex geometries, AIAA-2008-0531 (2008).