INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2008; **73**:597–623
Published online 8 May 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2081

On the computation of steady-state compressible flows using a discontinuous Galerkin method

Hong Luo[1,*,†], Joseph D. Baum[2] and Rainald Löhner[3]

[1] *Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, NC 27695, U.S.A.*
[2] *Center for Applied Computational Sciences, Science Applications International Corporation, McLean, VA 22102, U.S.A.*
[3] *School of Computational Sciences, George Mason University, Fairfax, VA 22030, U.S.A.*

SUMMARY

Computation of compressible steady-state flows using a high-order discontinuous Galerkin finite element method is presented in this paper. An accurate representation of the boundary normals based on the definition of the geometries is used for imposing solid wall boundary conditions for curved geometries. Particular attention is given to the impact and importance of slope limiters on the solution accuracy for flows with strong discontinuities. A physics-based shock detector is introduced to effectively make a distinction between a smooth extremum and a shock wave. A recently developed, fast, low-storage p-multigrid method is used for solving the governing compressible Euler equations to obtain steady-state solutions. The method is applied to compute a variety of compressible flow problems on unstructured grids. Numerical experiments for a wide range of flow conditions in both 2D and 3D configurations are presented to demonstrate the accuracy of the developed discontinuous Galerkin method for computing compressible steady-state flows. Copyright © 2007 John Wiley & Sons, Ltd.

Received 15 December 2006; Revised 21 March 2007; Accepted 23 March 2007

KEY WORDS: discontinuous Galerkin methods; shock detectors; p-multigrid; compressible flows

1. INTRODUCTION

The discontinuous Galerkin methods [1–13] (DGMs) have recently become popular for the solution of systems of conservation laws to arbitrary order of accuracy. The DGMs combine two advantageous features commonly associated with finite element and finite volume methods (FVMs). 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 FVMs. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from

[*] Correspondence to: Hong Luo, Department of Mechanical and Aerospace Engineering, North Carolina State University, 2601 Stinson Drive – Campus Box 7910, Raleigh, NC 27695, U.S.A.

[†] E-mail: hong_luo@ncsu.edu

the discontinuous representation of the solution at element interfaces. In this respect, the methods are therefore similar to FVMs. In fact, the first-order cell-centered finite volume scheme exactly corresponds to the DG(P0) method, i.e. to the DGM using piece-wise constant polynomials. Consequently, the DG(P_k) method with $k>0$ can be regarded as the natural extension of FVMs to higher-order methods. The DGMs 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 highly parallelizable, as they are compact and each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal, domain decomposition can be efficiently employed. 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; and (4) they have several useful mathematical properties with respect to conservation, stability, and convergence.

However, the DGMs have a number of their own weaknesses. In particular, how to efficiently handle curved geometries and how to effectively control spurious oscillations in the presence of strong discontinuities remain two of the unresolved issues in the DG methods. It is widely recognized and accepted that DGM solutions are more sensitive to the error arising at curved boundaries than those obtained with FVM of the same order of accuracy [8]. The common remedy to this problem is to use higher-order geometrical approximation [8]. Unfortunately, curved element meshes are associated with extra computational expenses. Curved elements need to be mapped onto the computational straight-sided element by a non-linear mapping. To account for the non-constant Jacobian and the variation of the metric, a higher number of quadrature points are required to compute volume and boundary integrals. It is well known that the non-physical oscillations exist in the vicinity of discontinuities. A discontinuity capturing and an appropriate slope limiter are two common strategies to cure this problem. 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 flux limiting are not directly applicable to high-order DGMs 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 FVM can then be used or modified to meet the needs of the DGMs. Such a filter is easily integrated in an explicit method, but not into an implicit method. Unfortunately, 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 and a degeneration of the solution. In fact, it is neither necessary nor desirable to apply slope limiters everywhere from a consideration of both computational efficiency and numerical accuracy. To address this concern, the limiters are applied only where they are really needed. This is accomplished using the so-called discontinuity detectors [14, 15], which are designed to distinguish regions where solutions are smooth and discontinuous. Then, the limiting is only used near discontinuities and high-order accuracy can be preserved in smooth regions. However, almost all discontinuity detectors, will mistakenly identify local high gradient smooth extrema such as stagnation regions as shock regions due to the high comparable gradient there. For aerodynamic applications, the active limiters at the leading edge of an airfoil will produce spurious entropy, thus pollute the solution in the flow field, and ultimately lead to the loss of the higher-order accuracy.

DOI: 10.1002/nme

The objective of the effort discussed in this paper is to develop a high-order DGM for computing steady-state solutions to the compressible Euler equations on unstructured grids. Both the areas required in the practical application of DG methods, implementation of boundary conditions for curved geometries and limiting for discontinuities are discussed. The solid wall boundary conditions in curved geometries are imposed using a novel approach [16], 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. Particular attention is given to the impact and importance of slope limiters on the solution accuracy. It will be demonstrated that DGMs are very sensitive to the treatment and implementation of the slope limiters, and the use of limiter everywhere will completely destroy the formal high-order accuracy of the DG methods and the resulting DG solutions are no more accurate than the FV solutions. To address this concern, a physics-based shock detector is developed and found to be very helpful and effective in making a distinction between a smooth extremum and a shock wave. The limiter is then only applied in these regions identified by this shock detector in order to reduce the computational cost and maintain the high-order accuracy of the DG methods. The developed DG method is applied to compute a variety of compressible flow problems for a wide range of flow conditions in both 2D and 3D configurations. The numerical results obtained illustrate the superior accuracy of this DGM over a vertex-based FVM, demonstrating that the DGMs provide a viable, attractive, and competitive alternative to the traditional finite-volume, finite-element, and finite-difference methods for computing compressible flows on unstructured grids. The remainder of this paper is structured as follows. The governing equations are listed in Section 2. The underlying DGM is described in detail in Section 2.1. Extensive numerical experiments are reported in Section 3. Concluding remarks are given in Section 4.

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 the conservative state vector $\mathbf{U}$ and the inviscid flux vectors $\mathbf{F}$ are defined by

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

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

$$p=(\gamma-1)\rho(e-\tfrac{1}{2}u_j u_j) \tag{3}$$

which is valid for perfect gas, where γ is the ratio of the specific heats, together with the initial and boundary conditions

$$\mathbf{U}(\mathbf{x},0)=\mathbf{U}_0(\mathbf{x}) \tag{4}$$

$$\mathbf{U}|_\Gamma=\mathbf{U}_\Gamma \tag{5}$$

where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω.

2.1. Discontinuous Galerkin spatial discretization

To formulate the DGM, 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:

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\mathbf{W}\,\mathrm{d}\Omega+\int_\Gamma \mathbf{F}_j\mathbf{n}_j\mathbf{W}\,\mathrm{d}\Gamma-\int_\Omega \mathbf{F}_j\frac{\partial \mathbf{W}}{\partial x_j}\,\mathrm{d}\Omega=0\quad\forall\mathbf{W} \tag{6}$$

where $\mathbf{n}_j$ denotes the unit outward normal vector to the boundary.

Assuming that Ω_h is a classical triangulation of Ω where the domain Ω is subdivided into a collection of non-overlapping elements Ω_e, triangles in 2D and tetrahedra in 3D, the following semi-discrete form of (6) is obtained by applying (6) on each element Ω_e

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_\mathrm{e}}\mathbf{U}_h\mathbf{W}_h\,\mathrm{d}\Omega+\int_{\Gamma_\mathrm{e}}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j\mathbf{W}_h\,\mathrm{d}\Gamma-\int_{\Omega_\mathrm{e}}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial \mathbf{W}_h}{\partial x_j}\,\mathrm{d}\Omega=0\quad\forall\mathbf{W}_h \tag{7}$$

where $\Gamma_\mathrm{e}(=\partial\Omega_\mathrm{e})$ denotes the boundary of Ω_e, and $\mathbf{U}_h$ and $\mathbf{W}_h$ represent the finite element approximations to the analytical solution $\mathbf{U}$ and the test function $\mathbf{W}$, respectively. Assume the approximate solution and test function to be piece-wise polynomials in each element, then $\mathbf{U}_h$ and $\mathbf{W}_h$ can be expressed as

$$\mathbf{U}_h(\mathbf{x},t)=\sum_{m=1}^{N}\mathbf{U}_m(t)B_m^p(\mathbf{x}),\quad \mathbf{W}_h(\mathbf{x})=\sum_{m=1}^{N}\mathbf{W}_m B_m^p(\mathbf{x}) \tag{8}$$

where $B_m^p(\mathbf{x})$, $1\leqslant m\leqslant N$ is the basis function of the polynomials 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, as

$$N=\frac{(p+1)(p+2)\ldots(p+d)}{d!}\quad\text{for } d=1,2,3 \tag{9}$$

(7) must be satisfied for any test function $\mathbf{W}_h$. Since B_n^p is the basis for $\mathbf{W}_h$, (7) is, therefore, equivalent to the following system of N equations:

$$\frac{\mathrm{d}\mathbf{U}_m}{\mathrm{d}t}\int_{\Omega_\mathrm{e}}B_m^pB_n^p\,\mathrm{d}\Omega+\int_{\Gamma_\mathrm{e}}\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_jB_n^p\,\mathrm{d}\Gamma-\int_{\Omega_\mathrm{e}}\mathbf{F}_j(\mathbf{U}_h)\frac{\partial B_n^p}{\partial x_j}\,\mathrm{d}\Omega=0,\quad 1\leqslant n\leqslant N \tag{10}$$

where $\mathbf{U}_h$ is replaced with (8). Since the numerical solution $\mathbf{U}_h$ is discontinuous between element interfaces, the interface fluxes are not uniquely defined. The flux function $\mathbf{F}_j(\mathbf{U}_h)\mathbf{n}_j$ appearing in the second term of (10) is replaced by a numerical Riemann flux function $\mathbf{H}(\mathbf{U}_h^\mathrm{L},\mathbf{U}_h^\mathrm{R},\mathbf{n})$, where

$\mathbf{U}_h^{\mathrm{L}}$ and $\mathbf{U}_h^{\mathrm{R}}$ are the conservative state vector at the left and right side of the element boundary. In order to guarantee consistency and conservation, $\mathbf{H}(\mathbf{U}^{\mathrm{L}}, \mathbf{U}^{\mathrm{R}}, \mathbf{n})$ is required to satisfy

$$\mathbf{H}(\mathbf{U}, \mathbf{U}, \mathbf{n}) = \mathbf{F}_j(\mathbf{U})\mathbf{n}_j, \quad \mathbf{H}(\mathbf{U}, \mathbf{V}, \mathbf{n}) = -\mathbf{H}(\mathbf{V}, \mathbf{U}, \mathbf{n}) \tag{11}$$

This scheme is called DGM of degree p, or in short notation 'DG(p) method.' Note that discontinuous Galerkin formulations are very similar to finite volume schemes, especially in their use of numerical fluxes. Indeed, the classical first-order cell-centered finite volume scheme exactly corresponds to the DG(0) method, i.e. to the DGM using piece-wise constant polynomials. Consequently, the DG(p) methods with $p>0$ can be regarded as a 'natural' generalization of FVMs to higher-order methods. By simply increasing the degree p of the polynomials, DG methods of corresponding higher orders are obtained.

In the present work, the Riemann flux function is approximated using the HLLC approximate Riemann solver [17], which has been successfully used to compute compressible viscous and turbulent flows on both structured grids [18] and unstructured grids [19]. 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, and (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.

The domain and boundary integrals in (10) are calculated using $2p$ and $2p+1$ order accurate Gauss quadrature formulas, respectively. The number of quadrature points necessary for a given order depends on the quadrature rule used. In the case of linear, quadratic, and cubic shape function, the domain integrals are evaluated using three, six, and 12 points, 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 11 quadrature points, respectively, and integration over the element boundaries for P0, P1, and P2 is performed using one, four, and seven quadrature points, respectively.

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

$$M\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = \mathbf{R}(\mathbf{U}) \tag{12}$$

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 $B^p|_{\Omega_\mathrm{e}}$ are non-zero 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.

2.2. *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 [1, 7]:

$$\mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^n) \tag{13}$$

$$\mathbf{U}^{(2)} = \tfrac{3}{4}\mathbf{U}^n + \tfrac{1}{4}[\mathbf{U}^{(1)} + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^{(1)})] \tag{14}$$

$$\mathbf{U}^{n+1} = \tfrac{1}{3}\mathbf{U}^n + \tfrac{2}{3}[\mathbf{U}^{(2)} + \Delta t M^{-1}\mathbf{R}(\mathbf{U}^{(2)})] \tag{15}$$

DOI: 10.1002/nme

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 Courant–Friedrichs–Lewy condition motivates us to develop a p-multigrid method [20, 21] 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 P0, 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.

2.3. *Curved wall boundary conditions*

Bassi and Rebay [8] have shown that DG methods are far more sensitive to errors arising at curved boundaries than those obtained with FVM of the same order of accuracy. Use of straight-sided linear elements in the high-order DG methods will result in spurious production of entropy on the boundary, and lead to numerical instabilities and loss of the higher-order accuracy. A common solution to this problem is to use boundary-fitted higher-order curved elements. Unfortunately, curved element meshes are associated with extra computational expenses. First, curved elements need to be mapped onto the computational straight-sided element by a non-linear mapping. Second, to account for the non-constant Jacobian and the variation of the metric, a higher number of quadrature points are required to compute volume and boundary integrals. In a novel approach, suggested by Krivodonova and Berger [16], the elements adjacent to the solid wall boundaries remain straight-sided elements. However, an accurate representation of the boundary normals is used to define a ghost state at quadrature points. Let the interior density, pressure, and velocity vector be ρ, p, u_i. Then, the flow variables at the ghost state g are computed with:

$$\rho^g = \rho \tag{16}$$

$$p^g = p \tag{17}$$

$$u_i^g = u_i - 2(u_i n_i) n_i \tag{18}$$

where n_i is the unit normal to the physical boundary, not the straight-sided element face unit normal. Once the ghost state values are determined at integration points, numerical fluxes in (7) are computed as usual straight-sided elements. It has been numerically shown that this approach does maintain the formal order of the DG method [16]. In this work, this strategy is adapted and generalized to both 2D and 3D configurations. In our implementation, the normals at the quadrature points are obtained using the local true surface normals based on the analytically defined boundary geometries. This is different from Krivodonova's approach, where the normals at the quadrature points are reconstructed from the straight-sided elements and are only approximation of the true normals of curved geometries. In addition, the accurate reconstruction of normals at the quadrature

DOI: 10.1002/nme

points from a linear element representation of boundaries can be difficult, if not impossible, for complex 3D configurations of scientific and industrial interests. However, our approach does require the geometric boundary information, which, fortunately, is handily available in our flow code, as our geometry definition file is incorporated into our flow code for the purpose of h-refinement and remeshing. Using only straight-sided elements instead of boundary-fitted elements represents a huge simplification of the code implementation and tremendous saving in both storage requirements and computing costs. As a result of this approach, higher-order body-fitted curved elements are not necessarily required in order to maintain higher-order accuracy of the DG methods.

2.4. Monotonicity limiter

It is well known that high-order numerical schemes produce spurious oscillations in the vicinity of discontinuities, which can lead to numerical instabilities and unbounded solutions. Even though the fluxes at the inter-element boundaries are computed using an appropriate Riemann flux function, spurious oscillations may still be generated by the discontinuous Galerkin finite element methods near strong discontinuities, when polynomials of higher degree are used ($p>0$). First-order schemes DG(0) are the only approaches that are able to maintain a monotonic solution at discontinuities. Unfortunately, numerical solutions obtained by these schemes exhibit too much dissipation. Generally speaking, there are two strategies of curing for this problem: a discontinuity capturing method and an appropriate slope limiting method. The former adds explicitly some form of non-linear dissipation terms to the discontinuous Galerkin discretization. The main disadvantage of this approach is that it requires some user-defined parameters, which can be both mesh and problem dependent. The latter is designed to restrict or suppress oscillations near discontinuities through a non-linear procedure based on comparing elemental solution features, such as slopes or curvatures, with those of neighboring elements. Classical techniques of flux limiting are not directly applicable for high-order DG methods because of the presence of volume terms in the formulation. Therefore, a slope limiter is not integrated in the computation of the residual, but effectively acts as a post-processing filter. Such a limiting procedure can be easily integrated in an explicit method, but not into an implicit method. Many slope limiters used in the FVM can then be applied or modified to meet the needs of the DGMs. Note that it is not an exaggeration to state that most research efforts for DG methods are focused on developing and designing appropriate stabilization methods, that can produce a solution with neither excessive diffusion nor spurious oscillations and does not adversely affect the formal order of accuracy of the DG methods, although construction of an accurate, efficient, and robust limiter remains one of the issues and challenges for the FVMs as well [22].

Following Barth and Jespersen [23], slopes are limited so that the solution at the quadrature points $\mathbf{x}_j$, $j=1,2,\ldots,K_{\Gamma_i}$ in an element $\mathbf{\Omega}_i$, $\mathbf{U}_i(\mathbf{x}_j)$, is in the range spanned by the neighboring solution averages

$$\overline{\mathbf{U}}_i^{\min}\leqslant\mathbf{U}_i(\mathbf{x}_j)\leqslant\overline{\mathbf{U}}_i^{\max} \tag{19}$$

where $\overline{\mathbf{U}}_i^{\min}$ and $\overline{\mathbf{U}}_i^{\max}$ are the minimum and maximum element averaged solution on the elements sharing faces with Ω_i. If (19) is violated for any quadrature points, then it is assumed that the element is close to a discontinuity, and the solution at this element $\mathbf{\Omega}_i$ is locally modified (limited) as

$$\mathbf{U}_i(\mathbf{x})=\overline{\mathbf{U}}_i+\alpha\nabla\mathbf{U}_i\cdot(\mathbf{x}-\mathbf{x}_i)\quad\forall\mathbf{x}\in\mathbf{\Omega}_i \tag{20}$$

DOI: 10.1002/nme

where $\overline{\mathbf{U}}_i$ is the cell-averaged solution at the element Ω_i, $\mathbf{x}_i$ is the position vector of the centroid of Ω_i, and where

$$\alpha = \min_{1\leqslant j\leqslant K_{\Gamma_i}} \alpha_j \tag{21}$$

$$\alpha_j = \begin{cases} \min\left(1, \dfrac{\overline{\mathbf{U}}_i^{\max} - \overline{\mathbf{U}}_i}{\mathbf{U}_i(\mathbf{x}_j) - \overline{\mathbf{U}}_i}\right) & \text{if } \mathbf{U}_i(\mathbf{x}_j) - \overline{\mathbf{U}}_i > 0 \\[2ex] \min\left(1, \dfrac{\overline{\mathbf{U}}_i^{\min} - \overline{\mathbf{U}}_i}{\mathbf{U}_i(\mathbf{x}_j) - \overline{\mathbf{U}}_i}\right) & \text{if } \mathbf{U}_i(\mathbf{x}_j) - \overline{\mathbf{U}}_i < 0 \\[2ex] 1 & \text{otherwise} \end{cases} \tag{22}$$

As one will see in the next section, just like DGMs are more sensitive to the treatment and implementation of slip boundary conditions at curved boundaries than those obtained with FVM of the same order of accuracy, DGMs are also much more sensitive to the treatment and implementation of the slope limiters than their FV counterparts. 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 and a degeneration of the solution. In fact, it is neither necessary nor desirable to apply slope limiters everywhere from the perspective of both computational efficiency and numerical accuracy. The limiters should only be used where they are really needed, i.e. in the vicinity of strong discontinuities. This can be accomplished using so-called discontinuity detectors [14, 15], which are designed to distinguish regions where solutions are smooth and discontinuous. Then, the limiting is only applied near discontinuities and high-order accuracy can be preserved in smooth regions. A number of discontinuity detectors [15] have been implemented and experimented during the course of this work. Unfortunately, almost all of them will mistakenly identify local high-gradient smooth extrema such as stagnation regions as shock regions due to the high comparable gradient there. For aerodynamic applications, the active limiters at the leading edge of an airfoil will produce spurious entropy, thus contaminate the solution in the flow field and ultimately lead to the loss of the higher-order accuracy. To address this concern, a physics-based shock detector is introduced here, which is found to be quite helpful and effective in making a distinction between a stagnation point and a shock wave. This physics-based shock detector is based on the two physics facts about a shock wave, namely (1) the normal Mach number is greater than 1 before a shock and less than 1 after a shock, and (2) a shock wave can only be a compressive wave. This can be easily done using the following three steps:

1. Compute shock wave direction $\mathbf{n}$ on each element

$$\mathbf{n} = \frac{\nabla q}{|\nabla q|}$$

   where q is the velocity vector magnitude.
2. Compute maximum and minimum normal Mach number on each element by taking flow variables at the adjacent faces into consideration.

DOI: 10.1002/nme

3. Compute the following wave indicator:

$$k=\frac{\partial(\mathbf{v}\cdot\mathbf{n})}{\partial n}$$

where $\mathbf{v}$ is the velocity vector. For compression wave $k<0$, and for expansion wave $k>0$.

Roughly speaking, the first test will identify the sonic regions, and the second one will eliminate the expansion waves. Our experience indicates that this shock detector can effectively eliminate high-gradient smooth regions, and performs better than alternative indicators for the steady-state flow problems considered here.

3. NUMERICAL EXAMPLES

All computations are performed on a Dell Precision M70 laptop computer with 2 GBytes memory running the Suse 10.0 Linux operating system. All computations are started with uniform flow. An elaborate and well-tested vertex-centered finite volume code [19, 24] is used as a reference to quantitatively compare the accuracy of the DG method, although it is not our objective to compare the performance of FV and DG methods in terms of computational efficiency and numerical accuracy. For most of the test cases, the following entropy production ε defined as

$$\varepsilon=\frac{S-S_\infty}{S_\infty}=\frac{p}{p_\infty}\left(\frac{\rho_\infty}{\rho}\right)^\gamma-1$$

is served as a criterion to measure the accuracy and quality of the numerical solutions, where S is the entropy. Note that the drag coefficient is very sensitive to spurious entropy generated by the numerical methods, entropy production should be zero (isentropic) for subsonic flows under consideration, and can only be increased across shock waves for transonic or supersonic flows.

3.1. Subsonic flows past a circular cylinder

The first example is a well-known test case: subsonic flow past a circular cylinder at a Mach number of $M_\infty=0.38$. This test case is chosen to verify the implementation of boundary conditions for curved geometries for DG methods and numerically compare accuracy between DG and FV methods. Figure 1(a) shows four successively refined o-type grids having 16×5, 32×9, 64×17, and 128×33 points, respectively. 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. The radius of the cylinder is $r_1=0.5$, the domain is bounded by $r_{33}=20$, and the radii of concentric circles for 128×33 mesh are set up as

$$r_i=r_1\left(1+\frac{2\pi}{128}\sum_{j=0}^{i-1}\alpha^j\right),\quad i=2,\ldots,33$$

where $\alpha=1.1580372$. The coarser grids are generated by successively un-refining the finest mesh.

DOI: 10.1002/nme

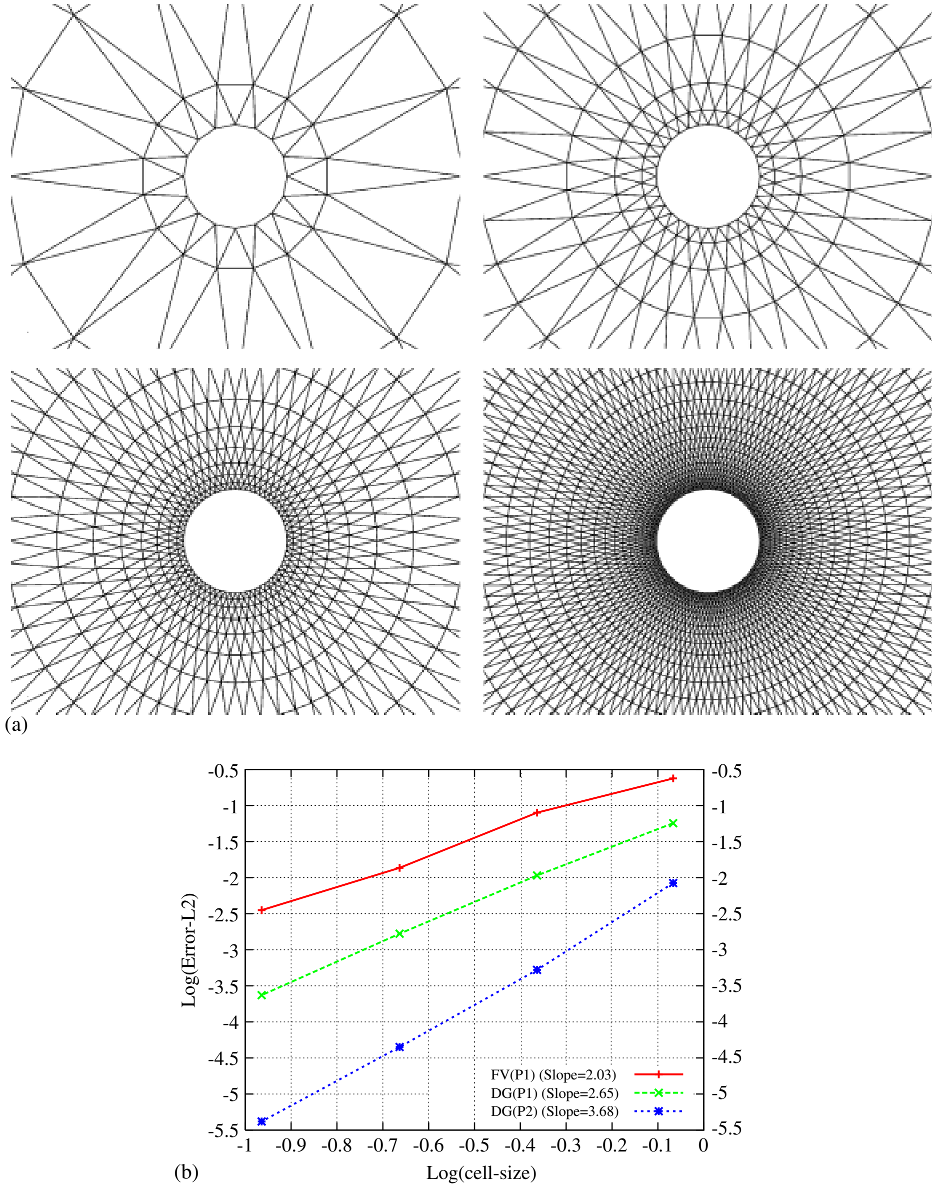


Figure 1. (a) Sequences of four successively globally refined meshes 16 × 5, 32 × 9, 64 × 17, 128 × 33 for computing subsonic flow past a circular cylinder; (b) results of grid-refinement study for flow past a cylinder obtained by the second-order FV method, second-order DG, and third-order DG methods; (c) computed Mach number contours obtained by the finite volume method (left) and DG(P1) method (right) on 64 × 17 mesh for subsonic flow past a circular cylinder at $M_\infty = 0.38$; (d) computed Mach number contours obtained by DG(P2) method (left) on 32 × 9 mesh and DG(P2) method (right) on 64 × 17 mesh for subsonic flow past a circular cylinder at $M_\infty = 0.38$; and (e) comparison of computed pressure coefficient (left) and entropy production (right) on the lower surface obtained by the FV(P1), DG(P1) on 64 × 17 mesh and DG(P2) on 32 × 9 for subsonic flow past a circular cylinder at $M_\infty = 0.38$.

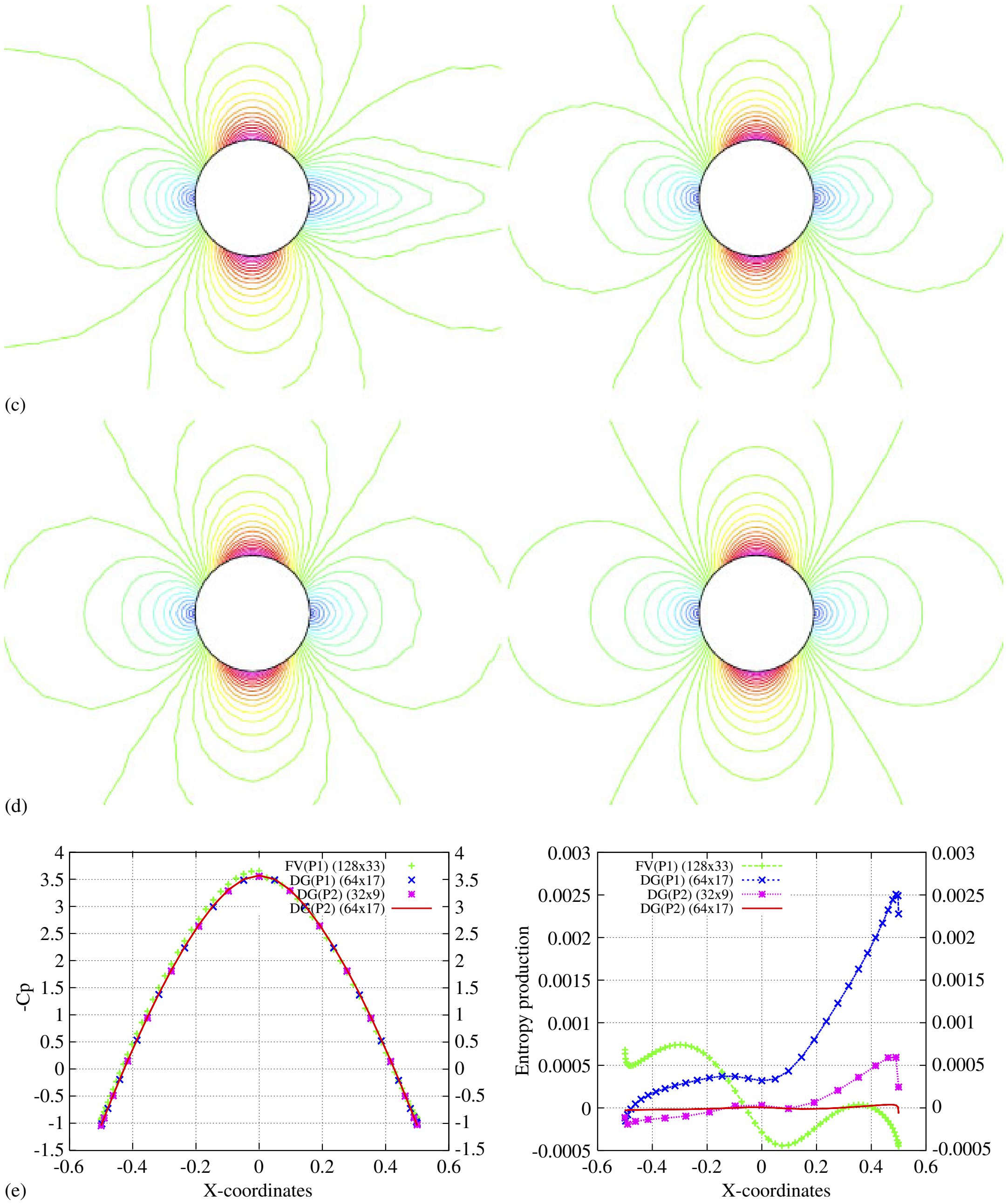


Figure 1. *Continued.*

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. The detailed results of this test case are presented in Table I(a–c). They show the mesh size,

DOI: 10.1002/nme

Table I. Subsonic circular cylinder test case: (a) FV(P1) is in the order of $O(h^2)$, (b) DG(P1) is in the order of $O(h^2)$ and (c) DG(P2) is in the order of $O(h^3)$.

	Mesh	No. DOFs	L^2-error	Order
(a)	16×5	80	2.37148E−01	—
	32×9	288	7.76551E−02	1.595
	64×17	1088	1.36962E−02	2.551
	128×33	4224	3.54568E−03	1.951
(b)	16×5	360	5.68722E−02	—
	32×9	1536	1.07103E−02	2.443
	64×17	6144	1.67302E−03	2.688
	128×33	24 576	2.34369E−04	2.838
(c)	16×5	768	8.40814E−03	—
	32×9	3072	5.26017E−04	4.055
	64×17	12 288	4.48952E−05	3.563
	128×33	49 152	4.16294E−06	3.434

the number of degrees of freedom, the L^2-error of the solutions, and the order of convergence. Figure 1(b) provides the details of the spatial accuracy of each method for this numerical experiment. The results obtained by DG methods, very similar to those found in the literature [8, 16], indicate that the DGM applied to the steady compressible Euler equations exhibits a full $O(h^{p+1})$ order of convergence on smooth solutions, and the novel approach for curved geometries instead of using body-fitted curved elements does maintain the formal order of the DG method.

It can be clearly seen that a higher-order DG method requires much lesser number of degrees of freedom than a lower-order DG method to achieve the same accuracy. In fact, DG(P2) solution on a given coarse mesh is actually better than DG(P1) solution on a doubled size fine mesh. Amazingly, the results obtained by DG(P1) are much better than those obtained by its finite volume counterpart FV(P1), which can clearly be seen on the computed Mach number contours in the flow field shown in Figures 1(c) and (d). In fact, DG(P1) solution on a given mesh is more accurate than FV(P1) solution on a doubled size fine mesh. These observations become especially apparent in Figure 1(e), where one compares the pressure coefficient and entropy production on the surface of cylinder obtained by DG(P1), DG(P2), and FV(P1) on the 64×17 mesh and DG(P2) on the 32×9 mesh. The advantage of the DG method over the FV method is self-evident here.

3.2. Subsonic flows past a sphere

As in the 2D case, a subsonic flow past a sphere at a Mach number of $M_\infty = 0.5$ is considered in this test case to validate the implementation of boundary conditions for curved geometries, and to numerically verify that the formal order of the DG method is not compromised using this approach instead of the curved boundary elements for 3D configuration.

Figure 2(a) shows three successively refined unstructured grids having 2174, 17 140, and 137 028, elements, respectively, and the computed Mach number contours in the flow field obtained by DG(P2) on the coarse mesh, DG(P1) on the medium mesh, and DG(P0) on the fine mesh. Note that only a quarter of configuration is modeled due to the symmetry of the problem, and that the number of elements on a successively refined mesh is not exactly 8 times the coarse mesh's elements, due to a smoothing procedure after dividing a tetrahedron into eight smaller tetrahedra.

DOI: 10.1002/nme

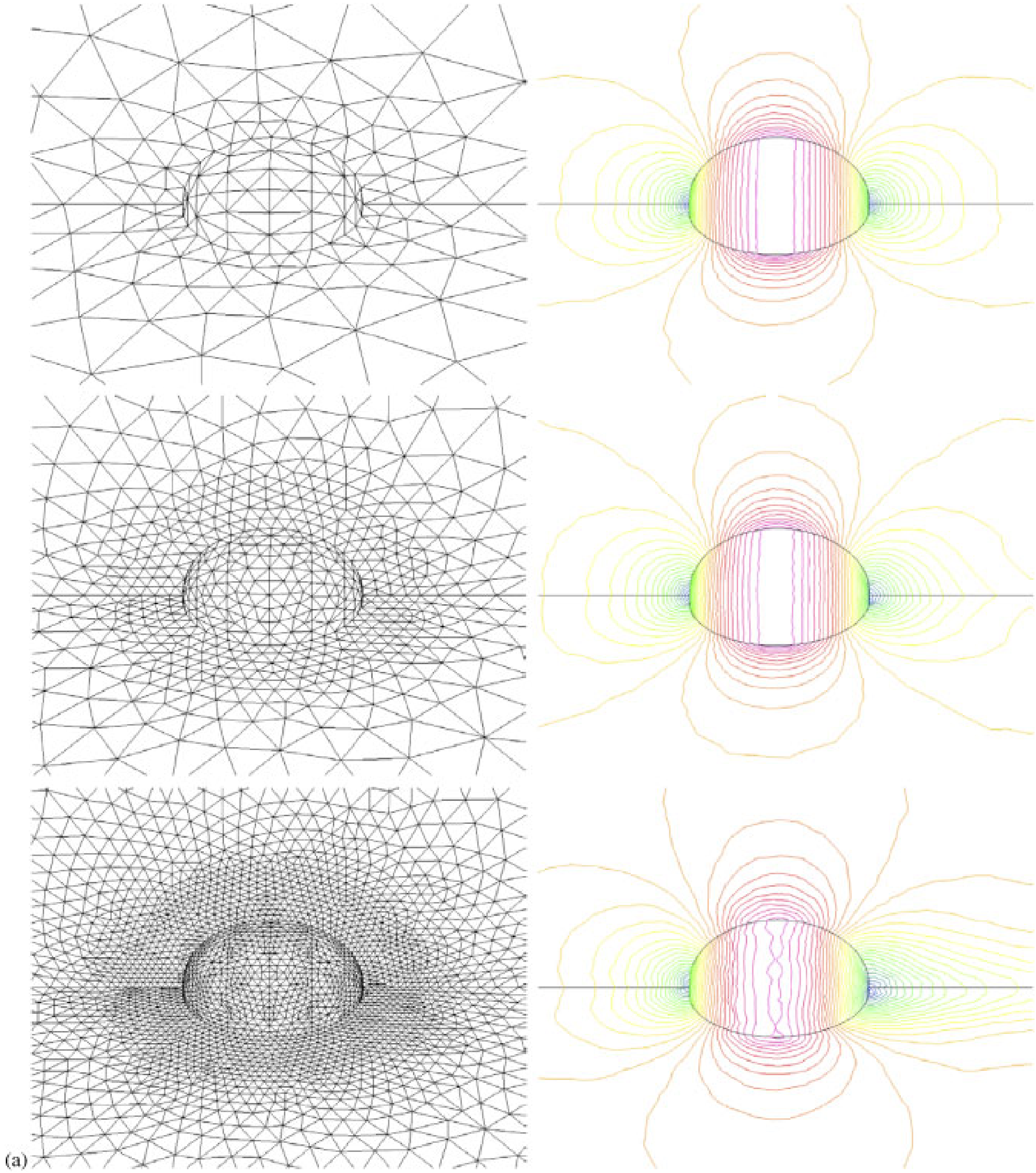


Figure 2. (a) Sequences of three globally refined unstructured surface meshes used for computing subsonic flow past a sphere. Right: computed Mach number contours obtained by DG(2) on the coarse mesh (top), by DG(1) on the medium mesh (middle), and DG(0) on the fine mesh (bottom) for computing subsonic flow past a sphere $M_\infty = 0.5$; (b) accuracy summary for subsonic flow past a sphere for DG(0), DG(1), and DG(2) computations; and (c) L^2-error of numerical solutions against the number of degrees of freedom for subsonic flow past a sphere by DG(0), DG(1), and DG(2) methods.

DOI: 10.1002/nme

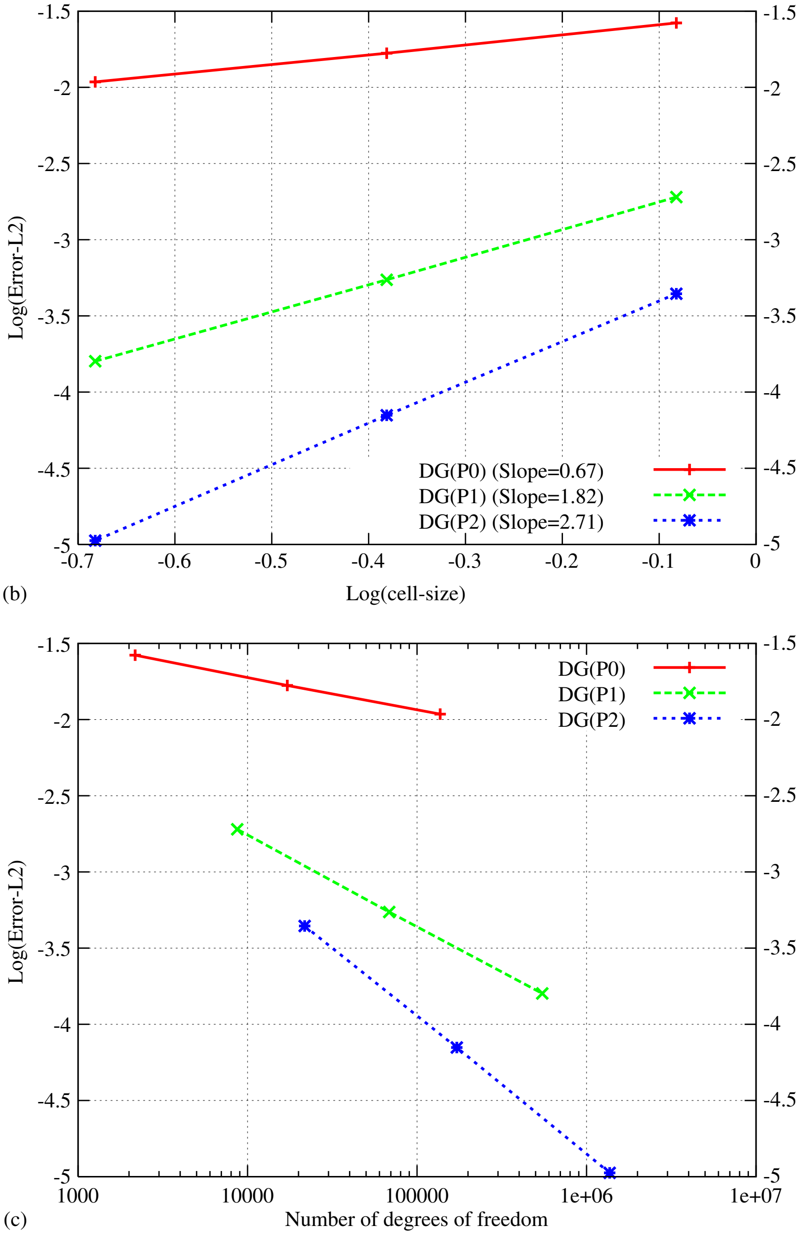


Figure 2. *Continued.*

Numerical solutions to this problem are computed using DG(0), DG(1), and DG(2) methods on these three grids to obtain quantitative measurement of the order of accuracy and discretization errors. As in the 2D case, the entropy production serves as the error measurement. Figure 2(b) provides spatial accuracy details of each method for this numerical experiment. Figure 2(c) shows the L^2-error of the DG(0), DG(1), and DG(2) methods plotted against the number of degrees of freedom. The results obtained by the DG method, perhaps not as impressive as those shown in the 2D study likely due the 3D grid quality, do indicate that the DGM exhibits a $O(h^{p+1})$ order of convergence on smooth solutions. Results also show that our simplified implementation of boundary conditions for curved geometries well conserves the formal order of the DG method. In addition, the higher-order DG method requires a significantly reduced number of degrees of freedom than the lower-order DG method to achieve the same accuracy. In fact, DG(2) solution on a given coarse mesh is actually better than DG(1) solution on a double refined mesh. The advantage of the higher-order method is again demonstrated for 3D configurations.

3.3. Transonic flow past a NACA0012 airfoil

The third example is the transonic flow past a NACA0012 airfoil. This test case is chosen primarily to demonstrate the superior accuracy of second-order DG method over the second-order FVM and the impact and importance of slope limiters on the accuracy of DG solutions for flows with shock waves. Two grids of similar quality are generated as shown in Figure 3(a). The coarse one has 1999 elements, 1048 grid points, and 97 boundary points, and the fine one consists of 8006 elements, 4102 grid points, and 198 boundary points. The computation is performed at a Mach number of 0.8, and an angle of attack 1.25°, characterized by the existence of a strong shock on the upper surface and a weak shock on the lower surface. Figure 3(b) shows the computed Mach number contours in the flow field obtained using DG(P1), DG(P2) without a limiter on the coarse mesh and FV(P1) methods without a limiter, on the coarse and fine mesh, respectively. The pressure coefficient and entropy production distributions on the surface of the airfoil are compared for these four solutions in Figure 3(c), respectively. As expected, the over- and under-shoots in the vicinity of shocks do appear in all four solutions, which are especially apparent for the two DG solutions due to a lack of numerical dissipation. Clearly, the DG solutions are much more accurate than their finite volume counterparts. In fact, the FV solution on the coarse mesh is so dissipative that it is unable to resolve the weak shock on the lower surface of the airfoil. Judging the accuracy of these two numerical methods based on the spurious entropy production at the leading edge of the airfoil, one can easily come to the conclusion that the DG(P1) solution on the coarse mesh is even more accurate (less dissipative) than the FV(P1) solution on the fine mesh, demonstrating the superior accuracy of the DG method over the FV method. However, when one compares the results, in Figures 3(d) and (e), obtained by the DG(P1) and FV(P1) solutions using Barth–Jespersen limiter everywhere, the DG(P1) solution is similar to, if not worse than, the FV(P1) solution, clearly demonstrating the sensitivity of DG methods to the limiters, and the importance of the limiters on the DG solutions. Figures 3(f) and (g) show the computed solutions obtained using the DG(P1) method with Barth–Jespersen limiter only active in the discontinuities identified by Krivodonova's discontinuity detector and the physics-based shock detector, respectively. Clearly, Krivodonova's discontinuity detector [14] mistakenly identifies the stagnation area as shock regions due to the high comparable gradient there, leading to the production of spurious entropy, and pollute the solution in the flow field. Note that Krivodonova's discontinuity detector is actually found to be one of the best shock detectors tested in Reference [15]. The physics-based shock detector is very effective in making a distinction between a smooth extremum (stagnation point) and a shock wave. As a result, the DG(P1) method is able to capture very sharply both strong and weak shock waves with one point with negligible amount of entropy production in the leading edge of the airfoil.

DOI: 10.1002/nme

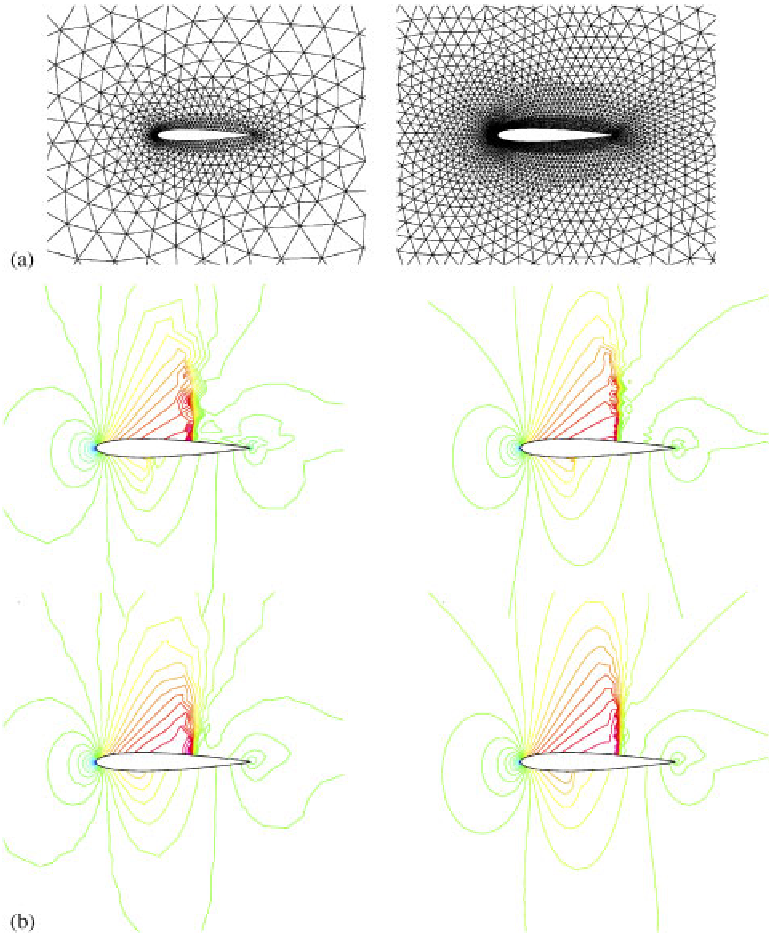


Figure 3. (a) Coarse (left, $n$elem $= 1999$, $n$poin $= 1048$, $n$boun $= 97$) and fine (right, $n$elem $= 8006$, $n$poin $= 4102$, $n$boun $= 198$) grids for a NACA0012 airfoil used in the computation; (b) computed Mach number contours obtained by the unlimited DG(P1) solution (top left), DG(P2) solution (top right), FV(P1) solution on the coarse mesh (bottom left), and FV(P1) solution on the fine mesh (bottom right) for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$; (c) comparison of computed pressure coefficient (top) and entropy production (bottom) distributions on the surface of airfoil obtained by the unlimited DG(P1), DG(P2), FV(P1) solutions on the coarse mesh, and FV(P1) solution on the fine mesh for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$; (d) computed Mach number contours obtained by the limited DG(P1) solution (left), and FV(P1) solution (right) on the coarse mesh for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$; (e) comparison of computed pressure coefficient (left) and entropy production (right) distributions on the surface of airfoil obtained by the limited DG(P1) and FV(P1) solutions on the coarse mesh for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$; (f) computed Mach number contours obtained by the limited DG(P1) solutions with Krivodonova's discontinuity detector (left), and the physics-based shock detector (right) on the coarse mesh for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$; and (g) comparison of computed pressure coefficient (left) and entropy production (right) distributions on the surface of airfoil obtained by the limited DG(P1) solutions with Krivodonova's discontinuity detector, and the physics-based shock detector on the coarse mesh for transonic flow past a NACA0012 airfoil at $M_\infty = 0.8$, $\alpha = 1.25°$.

DOI: 10.1002/nme

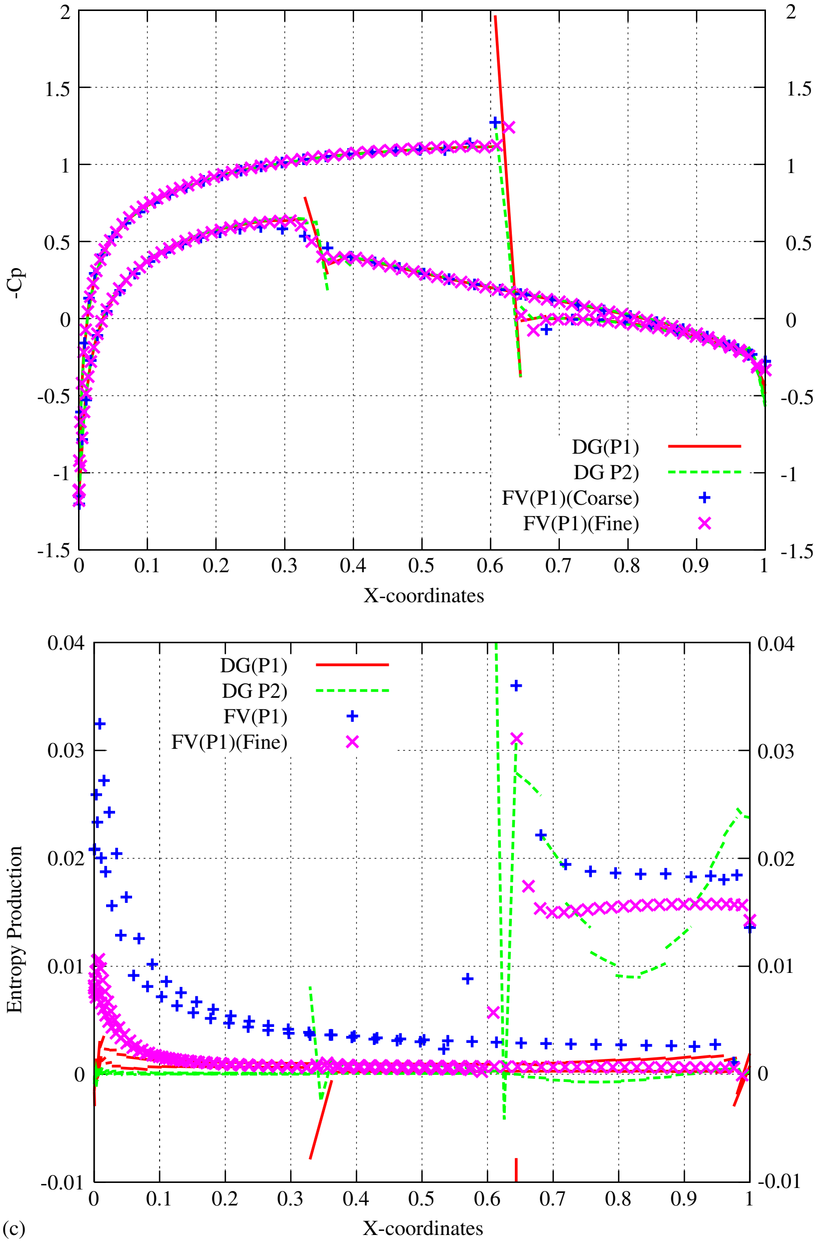


Figure 3. *Continued.*

3.4. *Transonic flow past a RAE 2822 airfoil*

The fourth test case is the transonic flow past a RAE2822 airfoil at a Mach number of 0.73, and an angle of attack 2.8°. Two grids of similar quality are generated as shown in Figure 4(a). The

DOI: 10.1002/nme

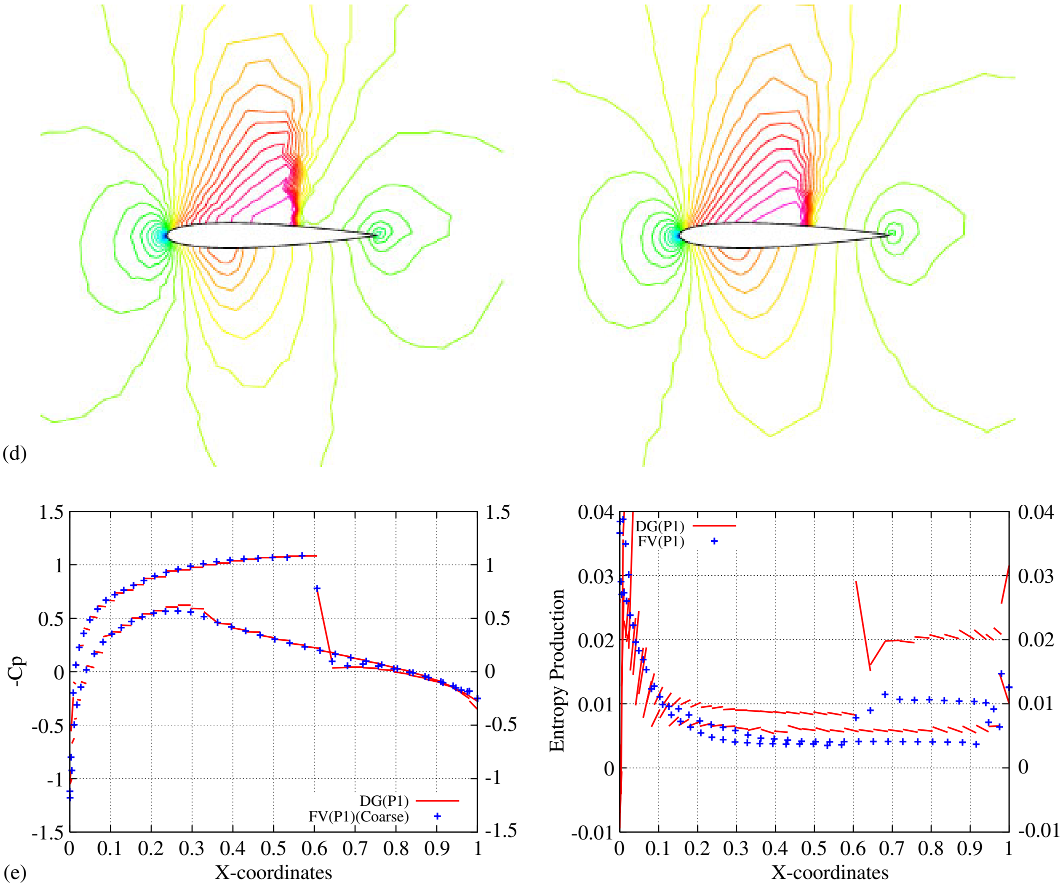


Figure 3. *Continued.*

coarse one has 2582 elements, 1360 grid points, and 138 boundary points, and the fine one consists of 10 408 elements, 5342 grid points, and 276 boundary points. The numerical experiments are performed using FV methods on both grids, and DG(P1) and DG(P2) methods on the coarse grid, respectively. Figure 4(b) shows the computed Mach number contours in the flow field obtained by FV solution on the coarse and fine grids, and DG(P1) and DG(P2) solutions on the coarse mesh using Barth–Jespersen limiter with the physics-based limiter, respectively. The computed pressure coefficient and entropy production distributions on the airfoil obtained by these four solutions are compared in Figure 4(c), where the experimental measurement is also given as a reference. Due to a lack of mesh resolution, FV solution on the coarse mesh is unable to capture the suction peak at the leading edge. 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. 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. a convergent solution is reached. Note that the shock wave is captured very well with only one point by all four solutions,

DOI: 10.1002/nme

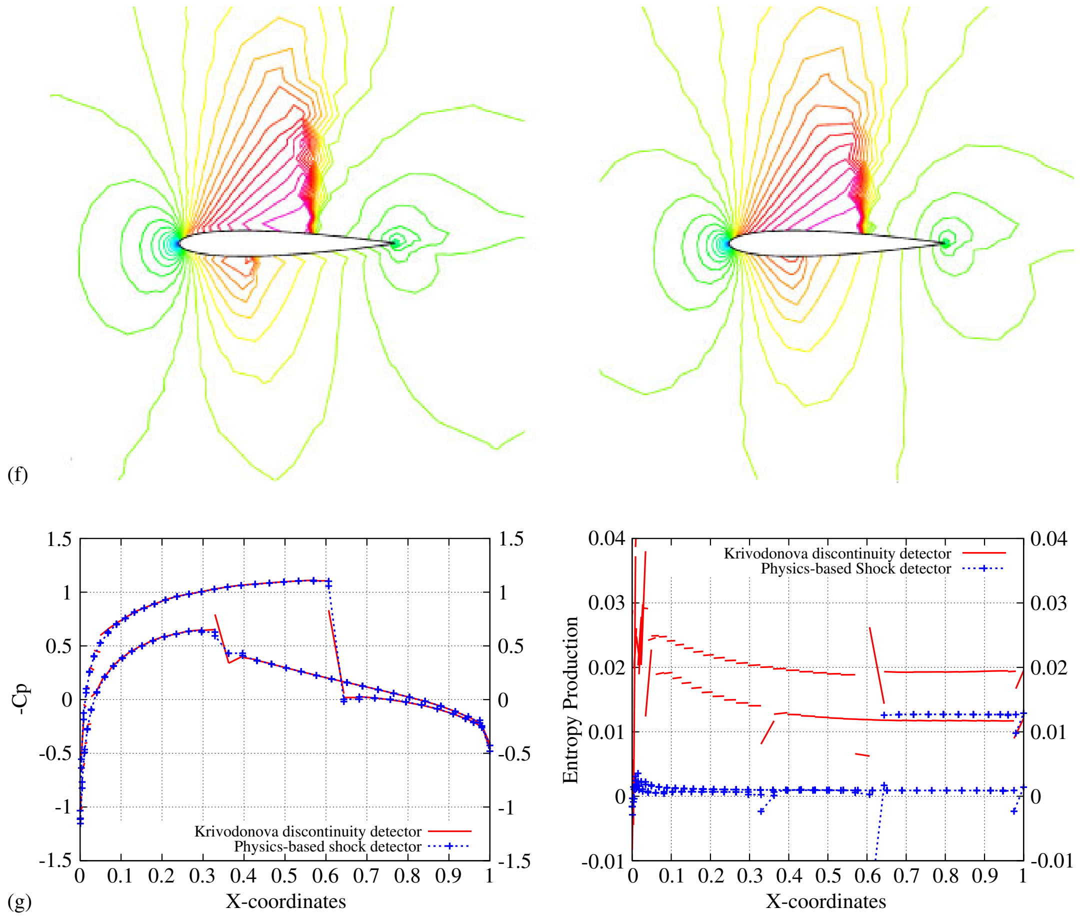


Figure 3. *Continued.*

and the results obtained by the two DG solutions and the FV solution on the fine mesh compare closely with experimental data, except the location of shock wave, due to a lack of viscous effects. The quality of DG(P2) solution can only be appreciated by looking at the almost zero entropy production before the shock wave, illustrating the high accuracy of the third-order DG solution.

3.5. Supersonic flow inlet flow

This example considers the supersonic flow entering a generic inlet configuration that is typical of scramjet engines. This test case is chosen to test the ability of higher-order DG method for accurately computing the supersonic flows. The configuration is taken from Reference [11], and the prescribed Mach number at the inlet is 3. The mesh used in the computation, which contains 7993 elements, 4276 points, and 559 boundary points, is depicted in Figure 5(a). Due to the special configuration inside the inlet, very complex flow features will appear. The computed Mach Number contours in the flow field obtained using FV(P1), DG(P1), and DG(P2) computations are shown

DOI: 10.1002/nme

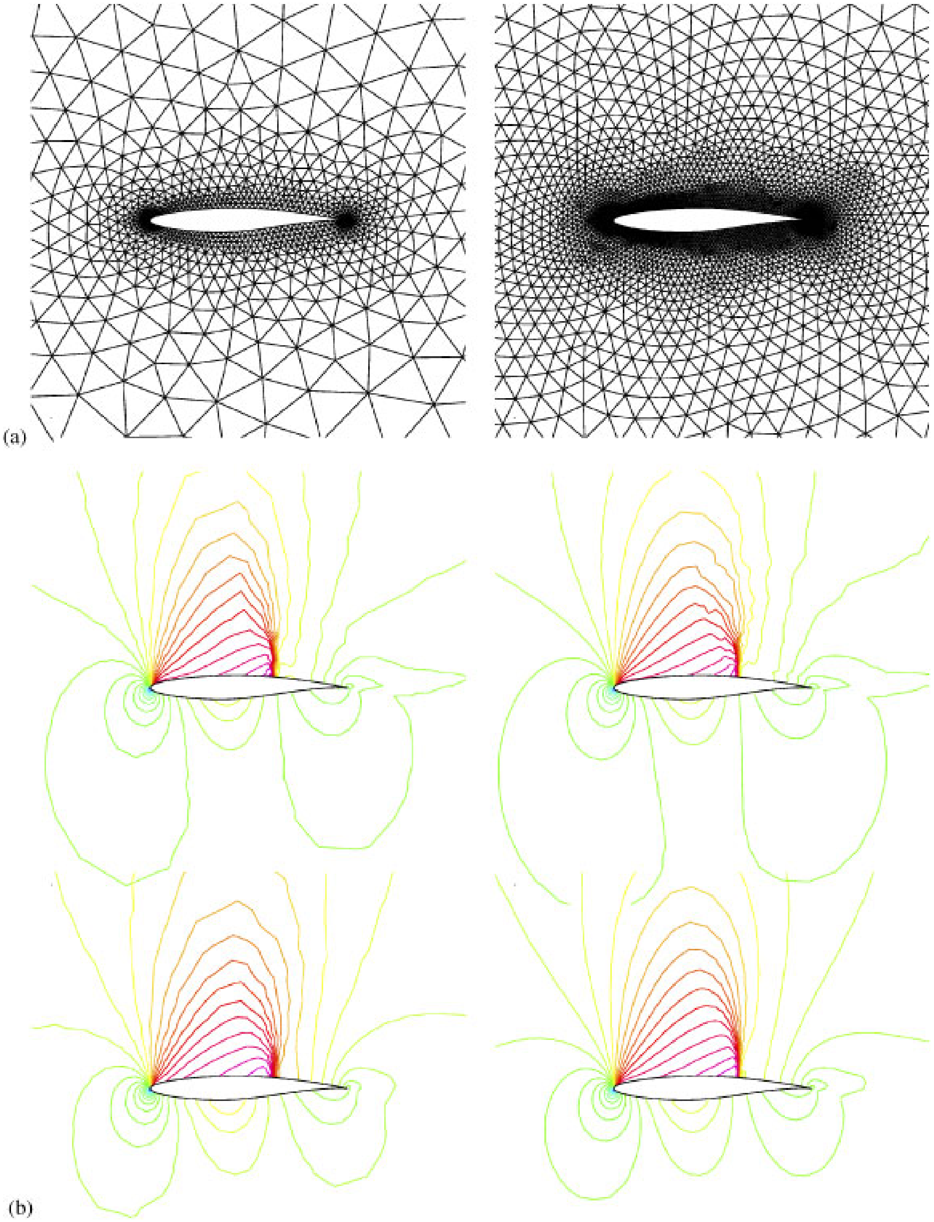


Figure 4. (a) Coarse (left, $n$elem = 2582, $n$poin = 1360, $n$boun = 138) and fine (right, $n$elem = 10 408, $n$poin = 5342, $n$boun = 276) grids for a RAE2822 airfoil used in the computation; (b) computed Mach number contours obtained by the DG(P1) solution (top left), DG(P2) solution (top right), FV(P1) solution on the coarse mesh (bottom left), and FV(P1) solution on the fine mesh (bottom right) for transonic flow past a RAE2822 airfoil at $M_\infty = 0.73$, $\alpha = 2.8°$; and (c) comparison of computed pressure coefficient (top) and entropy production (bottom) distributions on the surface of airfoil obtained by the DG(P1), DG(P2), FV(P1) solutions on the coarse mesh, and FV(P1) solution on the fine mesh for transonic flow past a RAE2822 airfoil at $M_\infty = 0.73$, $\alpha = 2.8°$.

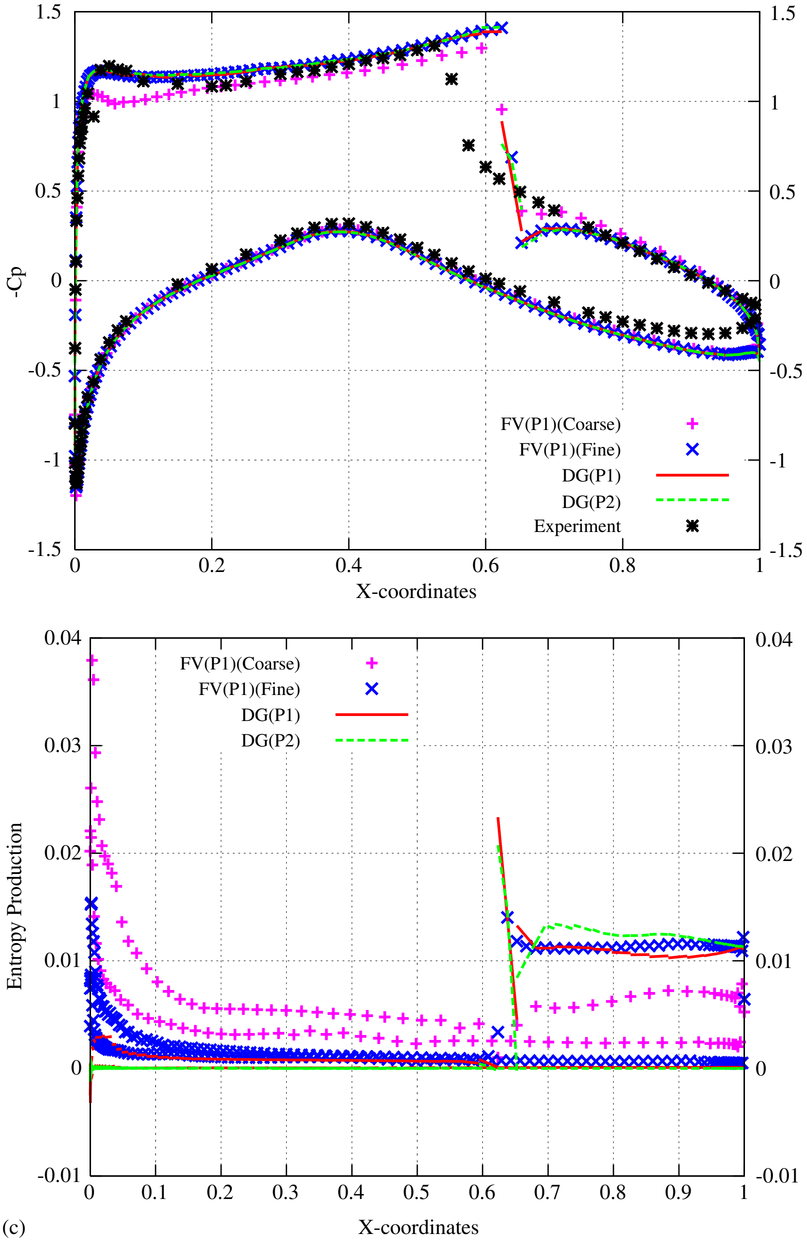


Figure 4. *Continued.*

in Figures 5(b)–(d), respectively. Both DG computations use the Barth–Jespersen limiter with the physics-based shock detector. Although both FV(P1) and DG(P1) methods are able to produce the similar flow features, DG(P1) produces better resolution than its second-order counterpart,

DOI: 10.1002/nme

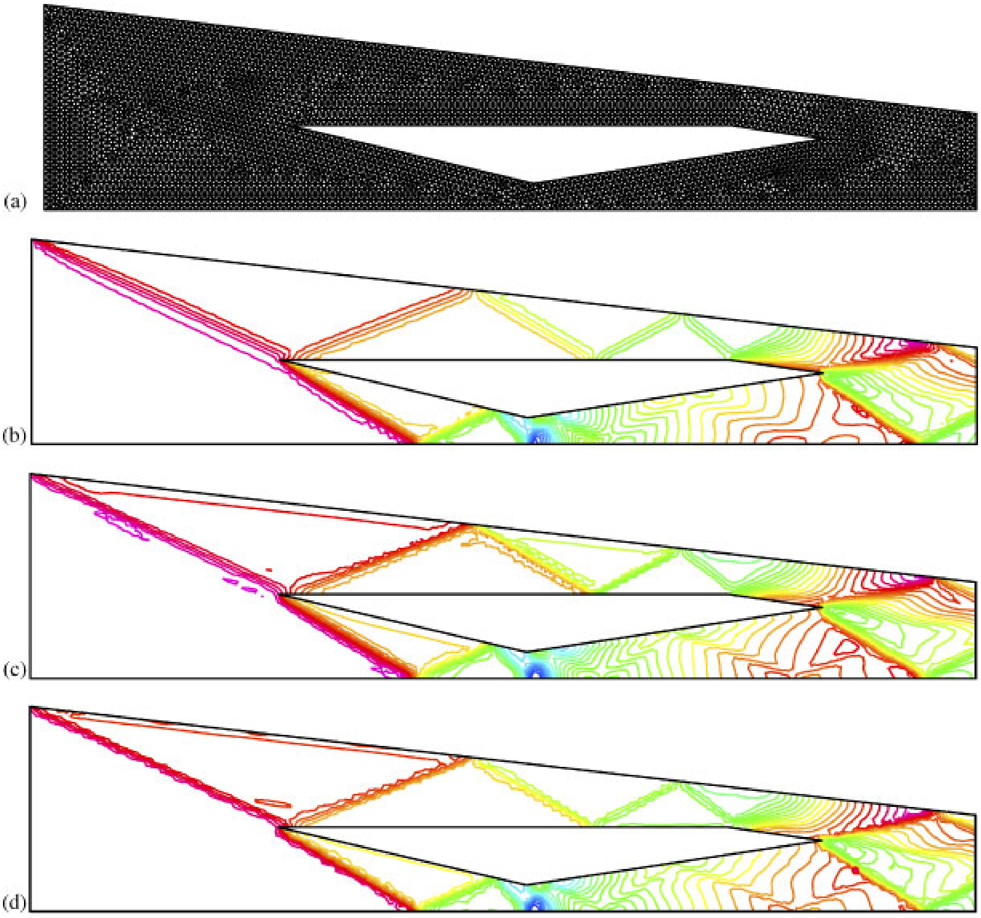


Figure 5. (a) Unstructured mesh used for computing supersonic flow inside an inlet ($n$elem$=7993$, $n$poin$=4276$, $n$boun$=559$); (b) computed Mach number contours by FV(P1) method with BJ limiter for supersonic flow inside an inlet at $M_\infty=3$; (c) computed Mach number contours by DG(P1) method with BJ limiter and physics-based shock detector for supersonic flow inside an inlet at $M_\infty=3$; and (d) computed Mach number contours by DG(P2) method with BJ limiter and physics-based shock detector for supersonic flow inside an inlet at $M_\infty=3$.

FV(P1). As expected, the DG(P2) delivers the best solution witnessed by the sharp resolution of shock waves and slip line.

3.6. Transonic flows past an ONERA M6 wing

A transonic flow over an 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)

DOI: 10.1002/nme

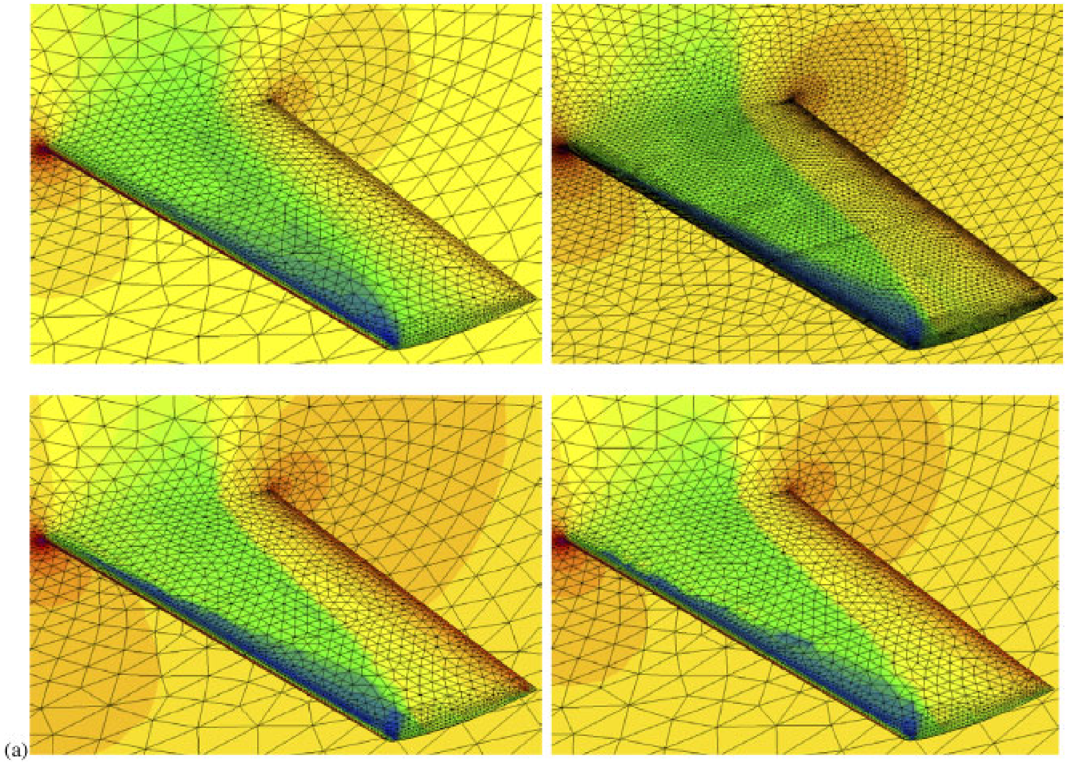
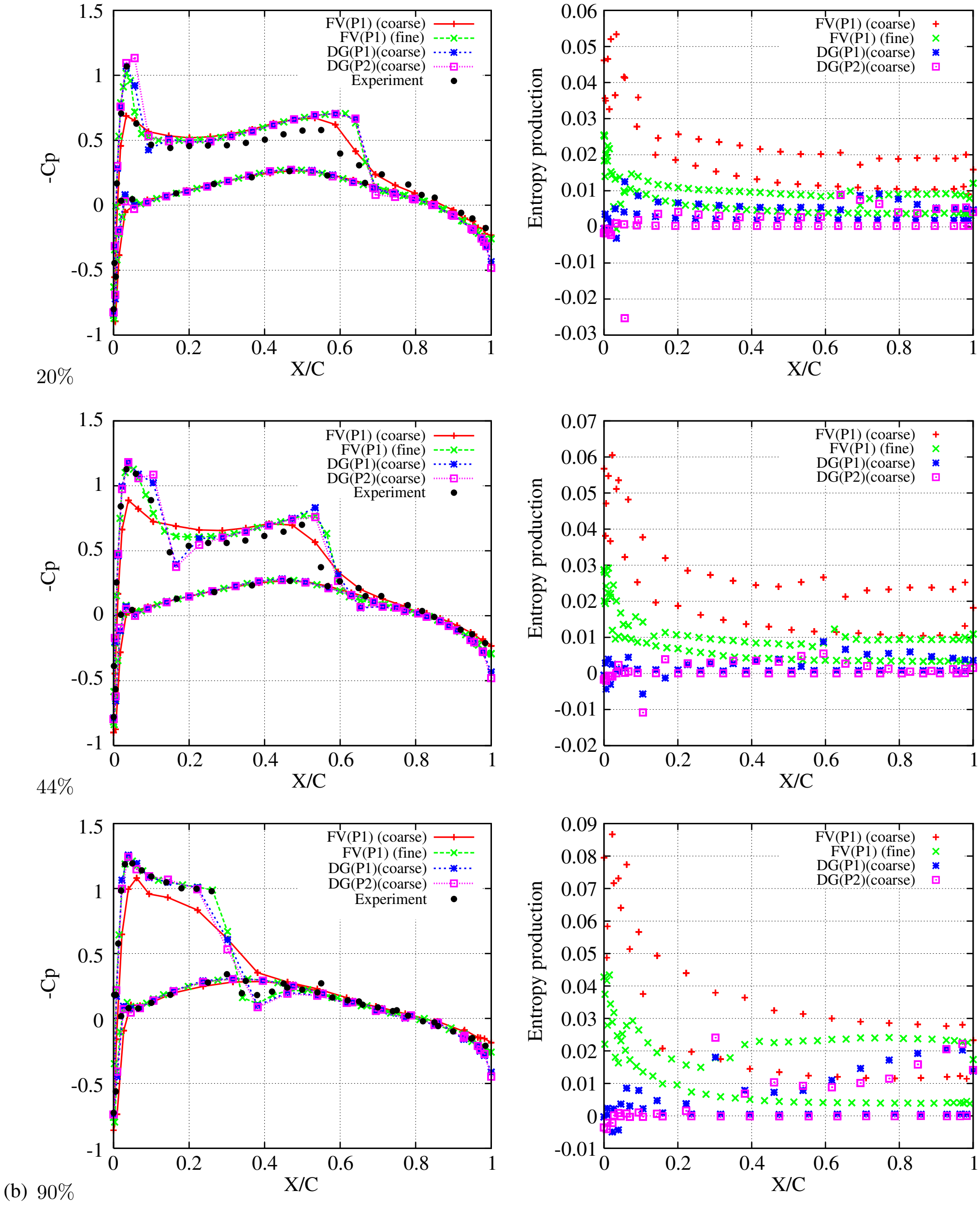


Figure 6. (a) Computed pressure contours on the unstructured surface mesh obtained by the FV(P1) solution on the coarse mesh (top left, $n$elem = 136 705, $n$poin = 25 616, $n$boun = 5017), the FV(P1) solution on the fine mesh (top right, $n$elem = 710 971, $n$poin = 131 068, $n$boun = 20 659), the DG(P1) solution on the coarse mesh, and the DG(P2) solution on the coarse mesh for transonic flow past an ONERA M6 wing at $M_\infty = 0.84$, $\alpha = 3.06°$ and (b) comparison of computed pressure coefficient (left) and entropy production (right) distributions for wing section at different semispan 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 $M_\infty = 0.84$, $\alpha = 3.06°$.

and DG(P2) methods with Barth–Jespersen limiter and physics-based shock detector on the coarse mesh, respectively. The coarse mesh contains 136 705 elements, 25 616 points, and 5017 boundary points, and the fine one 710 971 elements, 131 068 points, and 20 659 boundary points. Figure 6(a) shows the computed 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 spanwise stations in Figure 6(b), where experimental data for the pressure coefficients are 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. 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.



20%

44%

(b) 90%

Figure 6. *Continued.*

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.

DOI: 10.1002/nme

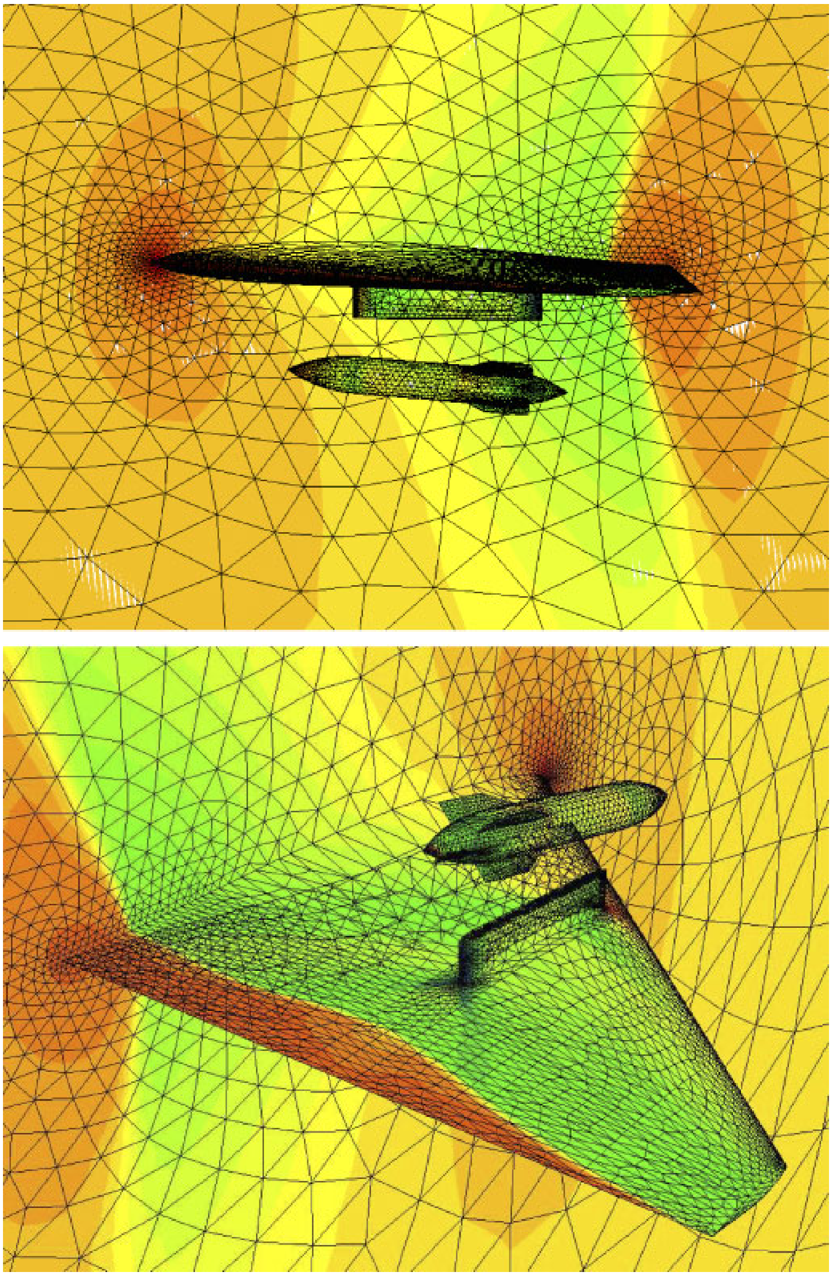


Figure 7. Computed pressure contours and unstructured surface mesh ($n$elem = 399 211, $n$poin = 72 455, $n$boun = 9242) for transonic flow past a wing/pylon/nacelle configuration at $M_\infty = 0.95$, $\alpha = 0°$.

DOI: 10.1002/nme

3.7. Transonic flows past a wing/pylon/nacelle configuration

Finally, as an illustrative example, a computation is performed on a wing/pylon/nacelle configuration to demonstrate that the developed DG method can be applied to problems of scientific and industrial interests. The mesh, used in the computation, contains 399 211 elements, 72 455 grid points, and 9242 boundary points for the half-span configuration. The computed solution is presented at a free stream of Mach number of 0.95 and an angle of attack of 0°. The computed pressure contours on the surface of the configuration, along with the surface mesh, are shown in Figure 7, where shock waves are captured well, confirming the accuracy and robustness of the developed DG method for computing complicated flows of practical importance.

4. CONCLUDING REMARKS

Computation of compressible steady-state flows using a high-order discontinuous Galerkin finite element method is discussed in this paper. An accurate representation of the boundary normals based on the definition of the geometries is used for imposing solid wall boundary conditions for curved geometries instead of using boundary-fitted elements, resulting in a huge simplification of the code implementation and tremendous saving in both storage requirements and computing costs. It is numerically demonstrated that limiters have a big impact on the accuracy of DG solutions, and application of limiters everywhere adversely affects the formal order of accuracy of the DG methods. In order to reduce the computational cost and maintain the high-order accuracy of the DG methods, it is necessary to use a slope limiter only in the vicinity of shock waves. This can be accomplished using a physics-based shock detector which is found to be very helpful and effective in making a distinction between a stagnation point and a shock wave. The developed DG method 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 numerical results obtained illustrate the superior accuracy of the developed DGM over a vertex-based finite volume method, demonstrating that the DGMs provide a viable, attractive, and competitive alternative to the traditional finite-volume, finite element, and finite-difference methods for computing compressible flows on unstructured grids. Future work will explore application of this method for the solution of the Navier–Stokes equations, as well as development of a reliable, and accurate shock detector for unsteady flows and more accurate, and robust limiters based on the strategies of the ENO/WENO schemes to effectively eliminate the spurious oscillations in the vicinity of strong discontinuities, thus alleviating the burden of a shock detector to precisely locate shock waves.

REFERENCES

1. Cockburn B, Hou S, Shu CW. TVD Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Mathematics of Computation* 1990; **55**:545–581.
2. Lin SY, Chin YS. Discontinuous Galerkin finite element method for Euler and Navier–Stokes equations. *AIAA Journal* 1993; **31**:2016–2023.
3. Biswas R, Devine KD, Flaherty J. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics* 1994; **14**:255–284.
4. Bey KS, Oden JT, Patra A. A parallel *hp*-adaptive discontinuous Galerkin method for hyperbolic conservation laws. *Applied Numerical Mathematics* 1996; **20**:321–336.
5. Atkins HL, Shu CW. Quadrature free implementation of discontinuous Galerkin method for hyperbolic equations. *AIAA Journal* 1998; **36**(5):775–782.

DOI: 10.1002/nme

6. Lockard DP, Atkins HL. Efficient implementations of the quadrature free discontinuous Galerkin method. *AIAA Paper*, *AIAA-99-3309*, 1999.
7. Cockburn B, Shu CW. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional system. *Journal of Computational Physics* 1998; **141**:199–224.
8. Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics* 1997; **138**:251–285.
9. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics* 1997; **131**:267–279.
10. van der Vegt JJW, van der Ven H. Discontinuous Galerkin finite element method with anisotropic local grid refinement for inviscid compressible flows. *Journal of Computational Physics* 1998; **141**:46–77.
11. Tu S, Aliabadi S. A slope limiting procedure in discontinuous Galerkin finite element method for gasdynamics applications. *International Journal of Numerical Analysis and Modeling* 2005; **2**(2):163–178.
12. Bassi F, Rebay S. GMRES discontinuous Galerkin solution of the compressible Navier–Stokes equations. In *Discontinuous Galerkin Methods*, *Theory*, *Computation*, *and Applications*, Cockburn B, Karniadakis GE, Shu CW (eds). Lecture Notes in Computational Science and Engineering, vol. 11. Springer: New York, 2000; 197–208.
13. Rasetarinera P, Hussaini MY. An efficient implicit discontinuous spectral Galerkin method. *Journal of Computational Physics* 2001; **172**:718–738.
14. Krivodonova L *et al*. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, geometries. *Applied Numerical Mathematics* 2004; **48**:323–338.
15. Qiu J, Shu CW. A comparison of trouble cell indicators for Runge–Kutta discontinuous Galerkin methods using WENO limiters. *SIAM Journal on Scientific Computing* 2005; **27**(3):995–1013.
16. Krivodonova L, Berger M. High-order implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics* 2006; **211**(2):492–512.
17. Toro EF, Spruce M, Speares W. Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves* 1994; **4**:25–34.
18. Batten P, Leschziner MA, Goldberg UC. Average-state Jacobians and implicit methods for compressible viscous and turbulent flows. *Journal of Computational Physics* 1997; **137**:38–78.
19. Luo H, Baum JD, Löhner R. High-Reynolds number viscous flow computations using an unstructured-grid method. *Journal of Aircraft* 2005; **42**(2):483–492.
20. Luo H, Baum JD, Löhner R. A *p*-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids. *Journal of Computational Physics* 2006; **211**(2):767–783.
21. Luo H, Baum JD, Löhner R. A fast, *p*-multigrid discontinuous Galerkin method for compressible flows on unstructured grids. *AIAA Paper 2006-0110*, 2006.
22. Berger M, Aftosmis M. Analysis of slope limiters on irregular grids. *AIAA Paper 2005-0490*, 2005.
23. Barth TJ, Jespersen DC. The design of application of upwind schemes on unstructured grids. *AIAA Paper 1989-0366*, 1989.
24. Luo H, Baum JD, Löhner R. A fast, matrix-free implicit method for compressible flows on unstructured grids. *Journal of Computational Physics* 1998; **146**(2):664–690.

DOI: 10.1002/nme