Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 1620–1642
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Efficient quadrature-free high-order spectral volume method on unstructured grids: Theory and 2D implementation

Rob Harris [a,*], Z.J. Wang [a,1], Yen Liu [b,2]

[a] *Department of Aerospace Engineering, Iowa State University, 2271 Howe Hall, Ames, IA 50011, United States*
[b] *NASA Ames Research Center, Mail Stop T27B-1, Moffett Field, CA 94035, United States*

Received 9 March 2007; received in revised form 18 August 2007; accepted 12 September 2007
Available online 19 November 2007

Abstract

An efficient implementation of the high-order spectral volume (SV) method is presented for multi-dimensional conservation laws on unstructured grids. In the SV method, each simplex cell is called a spectral volume (SV), and the SV is further subdivided into polygonal (2D), or polyhedral (3D) control volumes (CVs) to support high-order data reconstructions. In the traditional implementation, Gauss quadrature formulas are used to approximate the flux integrals on all faces. In the new approach, a nodal set is selected and used to reconstruct a high-order polynomial approximation for the flux vector, and then the flux integrals on the internal faces are computed analytically, without the need for Gauss quadrature formulas. This gives a significant advantage over the traditional SV method in efficiency and ease of implementation. For SV interfaces, a quadrature-free approach is compared with the Gauss quadrature approach to further evaluate the accuracy and efficiency. A simplified treatment of curved boundaries is also presented that avoids the need to store a separate reconstruction for each boundary cell. Fundamental properties of the new SV implementation are studied and high-order accuracy is demonstrated for linear and non-linear advection equations, and the Euler equations. Several well known inviscid flow test cases are utilized to show the effectiveness of the simplified curved boundary representation.

MSC: 65M60; 65M70; 35L65

Keywords: Spectral volume method; Euler equations; Unstructured grid; Hyperbolic conservation laws

1. Introduction

The spectral volume (SV) method is a recently developed finite volume method for hyperbolic conservation laws on unstructured grids [30–34,20,25]. The SV method belongs to a general class of Godunov-

* Corresponding author. Tel.: +1 515 294 0083.
E-mail addresses: rharris@iastate.edu (R. Harris), zjw@iastate.edu (Z.J. Wang), liu@nas.nasa.gov (Y. Liu).
[1] Tel.: +1 515 294 1614.
[2] Tel.: +1 650 604 6667.

0021-9991/$ - see front matter
doi:10.1016/j.jcp.2007.09.012

type finite volume method [12,28], which has been under development for several decades, and is considered to be the current state-of-the-art for the numerical solution of hyperbolic conservation laws. For a more detailed review of the literature on the Godunov-type method, refer to [30,29], and the references therein. Many of the most popular numerical methods, such as the k-exact finite volume [3,11], the essentially non-oscillatory (ENO) [14,1], and weighted ENO [18,37] methods are also Godunov-type methods. The SV method is also closely related to the discontinuous Galerkin (DG) [7–10,2,4,37] method, a popular finite-element method for conservation laws, and the spectral difference [24] method. Both the SV and DG methods employ multiple degrees of freedom within a single element. Each simplex in the SV method utilizes a "structured" set of sub-cells, thus resulting in more cell faces where Riemann problems must be solved. The number of interior faces in 3D may be high, making the 3D SV method expensive. However, this inherent property of sub-cell resolution may give the SV method the ability to capture discontinuities with higher resolution than the DG method. For more comparisons of the SV and DG methods, refer to [30,39].

The aforementioned additional cell faces present in the SV method number in the dozens per SV for 2D and in the hundreds per SV for 3D. In the traditional implementation of the SV method, all face integrals are computed by means of Gauss quadrature formulas, which are appropriate to the shape and dimension of the face. For example, a pentagonal face existing in the partition of a tetrahedral SV (3D) is split into 3 triangles. To carry out the integration, a Gauss quadrature formula of appropriate precision is then employed for each triangle. This procedure can be efficient in 2D, where the required Gauss quadrature points number in the dozens per SV. However in 3D, the partition of a tetrahedron can be so complicated that hundreds or thousands of Gauss quadrature points per SV may be necessary to compute the face integrals to the desired precision.

In this paper, we present two different approaches to handle the face integrals more efficiently. In each approach, a nodal set is defined within each SV to support a polynomial reconstruction for the flux vector. The shape functions defined by this nodal set are then integrated analytically over a standard element, resulting in an analytical representation for the flux integral on all faces. This reconstruction is universal for all SVs if a nodal set is distributed in a geometrically similar manner for all SVs. The first approach employs the above methodology for the internal faces only, while using the Gauss quadrature formulas for faces bounding an SV. This approach will be referred to as the partial quadrature (PQ) approach. The second approach employs the above methodology for all faces, and is thus referred to as the quadrature-free (QF) approach. Both approaches are shown to be stable and convergent.

It is well known that high-order methods are very sensitive to the treatment of curved wall boundaries. It has been shown recently [34] that in the high-order SV method, unless care is taken to represent curved boundaries with high-fidelity, the solution accuracy may be reduced to low-order. The customary approach is to introduce higher order elements for cells on curved wall boundaries. This approach has been shown to produce desired results [34], but this necessitates storage of a separate reconstruction for each cell bounding a curved wall. For large problems, and especially in 3D, the memory requirement for this approach may be large. Recently, Krivodonova and Berger [19] and Luo et al. [21] have presented a simplified treatment for curved boundaries which utilizes a high-order approximation of the normal vector to the physical rather than computational geometry. This eliminates the need to store a separate reconstruction for wall boundary cells. We implement a similar technique in this paper, for both the PQ and QF framework.

The paper is organized as follows. In Section 2, we review the basic formulation of the SV method. After that, the PQ and QF approaches are described in detail in Section 3. In Section 4, multi-dimensional total variation diminishing (TVD) limiters are discussed in detail. Section 5 outlines the methodology for a simplified curved boundary representation. Numerical results including accuracy studies for linear and non-linear advection, as well as the Euler equations are presented in Section 6. In addition, computations of inviscid flows over a circular cylinder, and a NACA 0012 airfoil are carried out to demonstrate the effectiveness of the curved boundary representation. Finally, conclusions and some possibilities for future work are summarized in Section 6.1.

2. Review of the basic spectral volume method

Consider the multi-dimensional conservation law

$$\frac{\partial Q}{\partial t}+\frac{\partial f(Q)}{\partial x}+\frac{\partial g(Q)}{\partial y}+\frac{\partial h(Q)}{\partial z}=0, \tag{2.1a}$$

on domain $\Omega\times[0,T]$ and $\Omega\subset R^3$ with the initial condition

$$Q(x,y,z,0)=Q_0(x,y,z), \tag{2.1b}$$

and appropriate boundary conditions on $\partial\Omega$. In (2.1), x, y, and z are the Cartesian coordinates and $(x,y,z)\in\Omega$, $t\in[0,T]$ denotes time, Q is the vector of conserved variables, and f, g and h are the fluxes in the x, y and z directions, respectively. Domain Ω is discretized into I non-overlapping triangular (2D), or tetrahedral (3D) cells. In the SV method, the simplex grid cells are called SVs, denoted S_i, which are further partitioned into CVs, denoted $C_{i,j}$, which depend on the degree of the polynomial reconstruction. Examples of partitions supporting linear, quadratic and cubic reconstructions are shown in Fig. 1 (2D), and quadratic and cubic reconstructions are shown in Fig. 2 (3D). The partitions shown in Fig. 2 were taken from Chen [6]. Volume-averaged conserved variables on the CVs are then used to reconstruct a high-order polynomial inside the SV. To represent the solution as a polynomial of degree m, we need N pieces of independent information, or degrees of freedom (DOFs). Where N is calculated as follows:

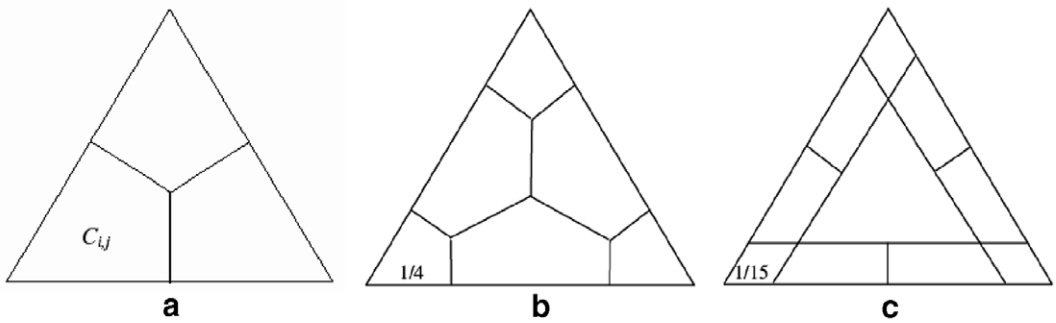

Fig. 1. Partitions of a triangular SV supporting linear, quadratic and cubic data reconstructions, shown in (a), (b) and (c), respectively.

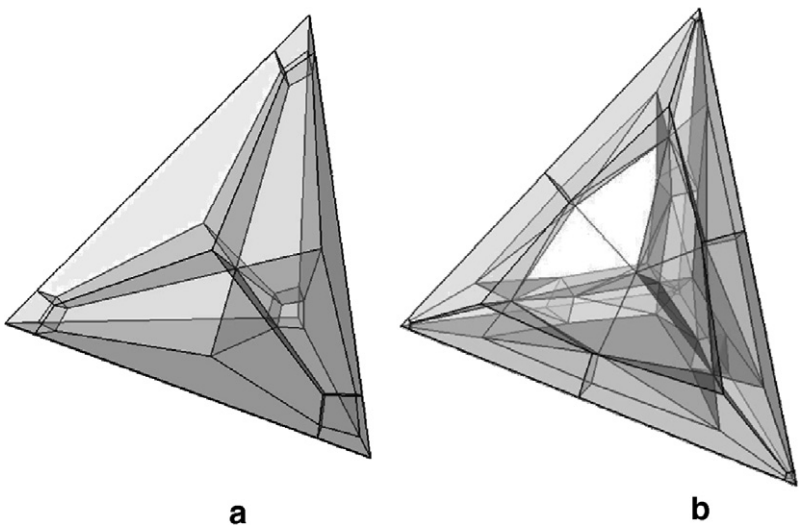

Fig. 2. Partitions of a tetrahedral SV supporting quadratic and cubic data reconstructions, shown in (a) and (b), respectively.

$$N = \frac{(m+1)(m+2)\cdots(m+d)}{d!}, \tag{2.2}$$

where d is the spatial dimension of the problem. The DOFs in the SV method are the volume-averaged conserved variables at the N CVs. Define the CV-averaged conserved variable for $C_{i,j}$ as

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q \, \mathrm{d}V, \quad j = 1, \ldots, N, \;\; i = 1, \ldots, I, \tag{2.3}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. Given the CV-averaged conserved variables for all CVs in S_i, a polynomial $p_i(x,y,z) \in P^m$ (the space of polynomials of at most degree m) can be reconstructed such that it is a $(m+1)$th order-accurate approximation to $Q(x,y,z)$ inside S_i.

$$p_i(x,y,z) = Q(x,y,z) + O(h^{m+1}), \quad (x,y,z) \in S_i, \quad i = 1, \ldots, I, \tag{2.4}$$

where h is the maximum edge length of all the CVs. This reconstruction can be solved analytically by satisfying the following conditions:

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} p_i(x,y,z) \mathrm{d}V = \overline{Q}_{i,j}, \quad j = 1, \ldots, N. \tag{2.5}$$

This polynomial $p_i(x,y,z)$ is the $(m+1)$th order approximation we are looking for as long as the solution $Q(x,y,z)$ is smooth in the region covered by S_i. The reconstruction can be expressed more conveniently as

$$p_i(x,y,z) = \sum_{j=1}^{N} L_j(x,y,z) \overline{Q}_{i,j}, \tag{2.6}$$

where $L_j(x,y,z) \in P^m$ are the shape functions which satisfy

$$\frac{1}{V_{i,j}} \int_{C_{i,j}} L_n(x,y,z) \mathrm{d}V = \delta_{jn}. \tag{2.7}$$

Integrating (2.1a) in $C_{i,j}$, we obtain

$$\frac{\mathrm{d}\overline{Q}_{i,j}}{\mathrm{d}t} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} (\vec{F} \cdot \vec{n}) \mathrm{d}A = 0, \quad j = 1, \ldots, N, \;\; i = 1, \ldots, I, \tag{2.8}$$

where $\vec{F} = (f, g, h)$, A_r represents the rth face of $C_{i,j}$, $\vec{n}$ is the outward unit normal vector of A_r, and K is the number of faces in $C_{i,j}$. More details of this, including representative plots of the shape functions can be found in [31]. If $\vec{F}$ is a non-linear function of the conserved variables, then the surface integration on each face is performed with a $(m+1)$th order-accurate Gauss quadrature formula; i.e.,

$$\int_{A_r} (\vec{F} \cdot \vec{n}) \mathrm{d}A \cong \sum_{q=1}^{J} w_{rq} \vec{F}(Q(x_{rq}, y_{rq}, z_{rq})) \cdot \vec{n}_r A_r, \tag{2.9}$$

where $J = \text{integer}[(m+2)/2]$ is the number of quadrature points on the rth face in 2D, w_{rq} are the Gauss quadrature weights, and (x_{rq}, y_{rq}, z_{rq}) are the Gauss quadrature points. Since the reconstructed solution is discontinuous across SV interfaces, the interface flux is not uniquely defined. Therefore the normal flux in (2.9) is replaced by a "numerical" Riemann flux, such as the Rusanov [23], Roe [22] or HLLC [15,26,27,5] flux. For example, the Rusanov flux can be written as

$$\widehat{F}(Q_{\mathrm{L}}, Q_{\mathrm{R}}, \vec{n}) \equiv \frac{1}{2}[\vec{F}(Q_{\mathrm{L}}) \cdot \vec{n} + \vec{F}(Q_{\mathrm{R}}) \cdot \vec{n} - \alpha(Q_{\mathrm{R}} - Q_{\mathrm{L}})] \cong \vec{F}(Q) \cdot \vec{n} \equiv F_n, \tag{2.10}$$

where Q_{L} and Q_{R} are the reconstructed solution at the left (inside $C_{i,j}$) and right side (outside $C_{i,j}$) of A_r, α is the maximum absolute eigenvalue of the flux Jacobian in $\vec{n}$ direction, evaluated at an average state of Q_{L} and Q_{R}. In [30], it was shown that the above SV scheme is $(m+1)$th order-accurate.

For time integration, we use either the 2nd or 3rd order Strong Stability-Preserving (SSP) Runge–Kutta scheme [13].

3. Formulation of the quadrature-free approach

In the new approach, a nodal set, such as those shown in Fig. 3, is selected from Hesthaven [16,17]. This nodal set is then used to reconstruct a high-order polynomial approximation for the flux vector, and then the flux integrals are computed analytically, without the need for Gauss quadrature formulas. If we assume Q, f, g and h to be all polynomials in (2.1a), obviously f, g and h should be one degree higher than Q. Therefore, a polynomial of degree $m+1$ is employed to reconstruct the flux vector.

The flux vector $\vec{F}$ can be computed at any point (x,y,z) by the following

$$\vec{F}(x,y,z) = \sum_{i=1}^{N_S} M_i(x,y,z)\vec{F}_i, \tag{3.1}$$

where N_S is the number of nodes in the nodal set, calculated from (2.2), and $\vec{F}_i$ is the flux vector at node i computed using the conserved variables at node i given by (2.6). Again, $Q(x,y,z)$ and $\vec{F}(x,y,z)$ are taken to be continuous polynomials within each SV, thus (2.6) and (3.1) are valid within a given SV and not across SV boundaries, so interpolations are only carried out locally within a given SV.

$M_i(x,y,z)$ are the shape functions defined by the nodal set which satisfy

$$M_n(x_j,y_j,z_j) = \delta_{jn}. \tag{3.2}$$

Some representative 2D examples of the shape functions are shown in Figs. 4 and 5.

In the PQ approach, (3.1) is integrated over each internal face, resulting in an analytical representation for the flux integral on all internal faces. This is done using a program capable of symbolic calculus, such as Mathematica [35]. This allows for the flux integral on each internal face to be computed as a weighted average of the flux evaluated at the nodal set, i.e.;

$$\int_{A_r} \vec{F}\cdot\vec{n}\,\mathrm{d}S = A_r \sum_{i=1}^{N_S} \overline{M}_i\vec{F}_i\cdot\vec{n}_r, \tag{3.3}$$

where $\overline{M}_i$ are the face-averaged shape functions for face r. In practice, face-averaged shape functions are computed during preprocessing for a standard element and then multiplied by the physical face area for computation. This eliminates the need to store all of the Jacobians. The face integrals on SV-bounding faces are then computed from (2.9), via the traditional approach.

In the QF approach, the face integrals on internal faces are evaluated as in the PQ approach, but some care is required for the faces defining the SV. Here, (2.10) is integrated over each SV-bounding CV face, and the resulting face integral can be expressed as the integral of a Riemann flux as follows:

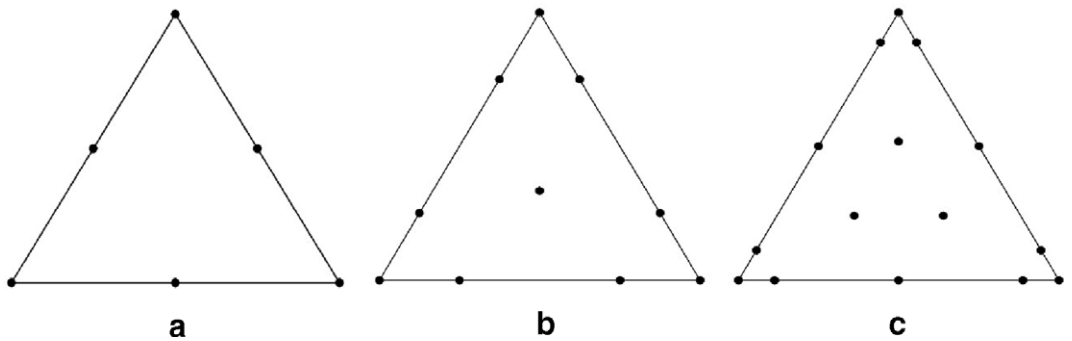

Fig. 3. Nodal sets in a triangular SV supporting quadratic, cubic and quartic data reconstructions for the flux vector, shown in (a), (b) and (c), respectively.

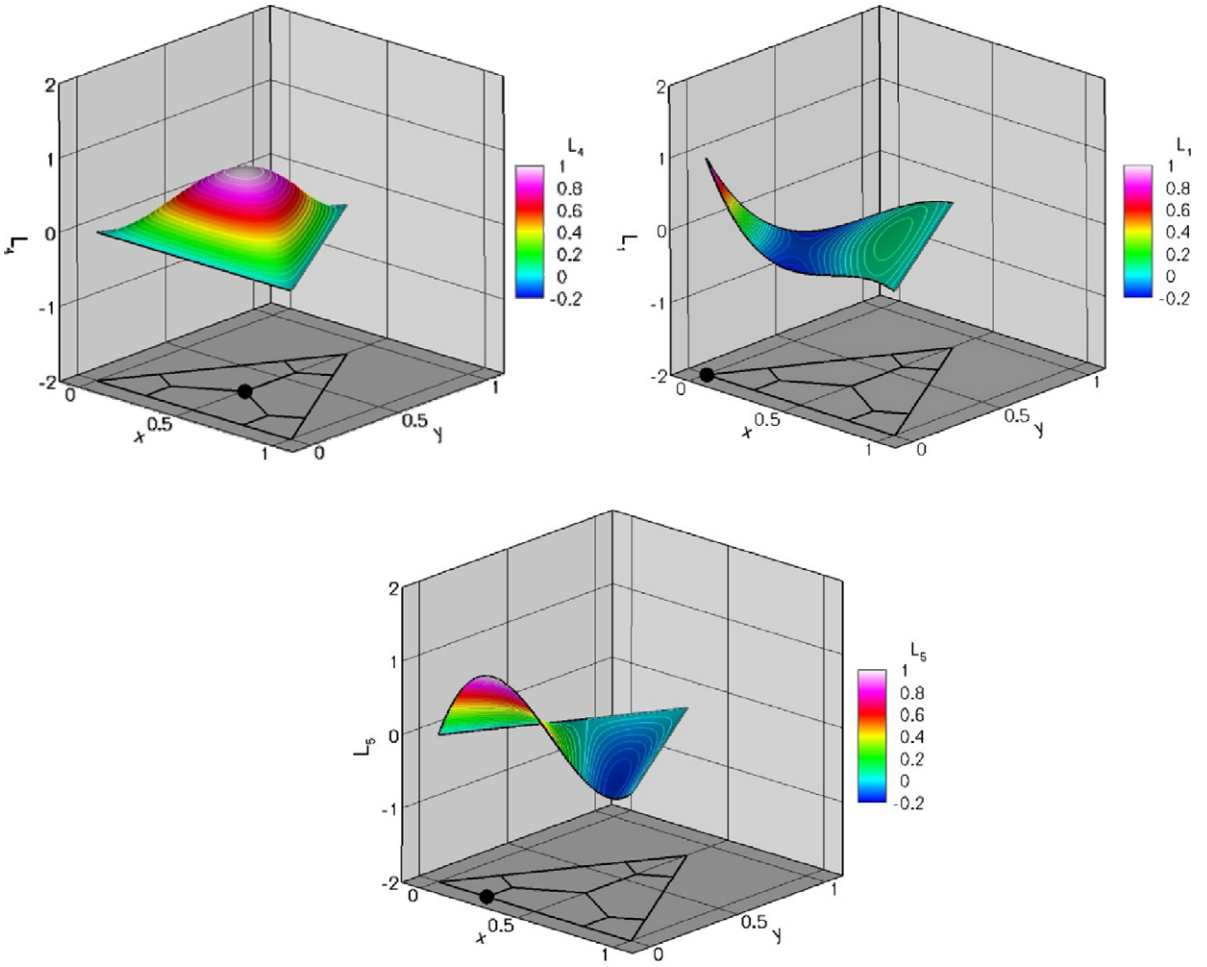

Fig. 4. Shape functions in a triangular SV supporting a cubic data reconstruction for the flux vector. The node which each function corresponds to is shown in bold.

$$\int_{A_r} \widehat{F}\,\mathrm{d}A = \frac{A_r}{2}[\overline{F}_{n,\mathrm{L}} + \overline{F}_{n,\mathrm{R}} - \alpha_c(\overline{Q}_{\mathrm{R}} - \overline{Q}_{\mathrm{L}})], \tag{3.4}$$

where $\overline{F}_{n,\mathrm{L}}$ and $\overline{F}_{n,\mathrm{R}}$ denote the face-averaged normal component of the flux vector due to the SV to the left and right of the interface, respectively, and α_c is taken as either the maximum absolute eigenvalue as in the Rusanov flux [23], or the dissipation matrix as in the Roe flux [22], which are evaluated at the face center. $\overline{Q}_{\mathrm{R}}$ and $\overline{Q}_{\mathrm{L}}$ are the face-averaged conserved variables due to the SV to the right and left of the interface, respectively. Eq. (3.4) can be deduced from the following analysis, which assumes a Rusanov flux. Since $p_i(x,y,z) = Q(x,y,z) + O(h^{m+1})$, the following is also true,

$$\begin{aligned} Q_{\mathrm{R}} - Q &= O(h^{m+1}),\\ Q_{\mathrm{L}} - Q &= O(h^{m+1}),\\ Q_{\mathrm{R}} - Q_{\mathrm{L}} &= O(h^{m+1}), \end{aligned} \tag{3.5a}$$

and the maximum absolute eigenvalue can be expanded according to its value at the face center, i.e., $\alpha = \alpha_c + O(h)$. So we can say that

$$\alpha(Q_{\mathrm{R}} - Q_{\mathrm{L}}) = \alpha_c(Q_{\mathrm{R}} - Q_{\mathrm{L}}) + O(h^{m+2}). \tag{3.5b}$$

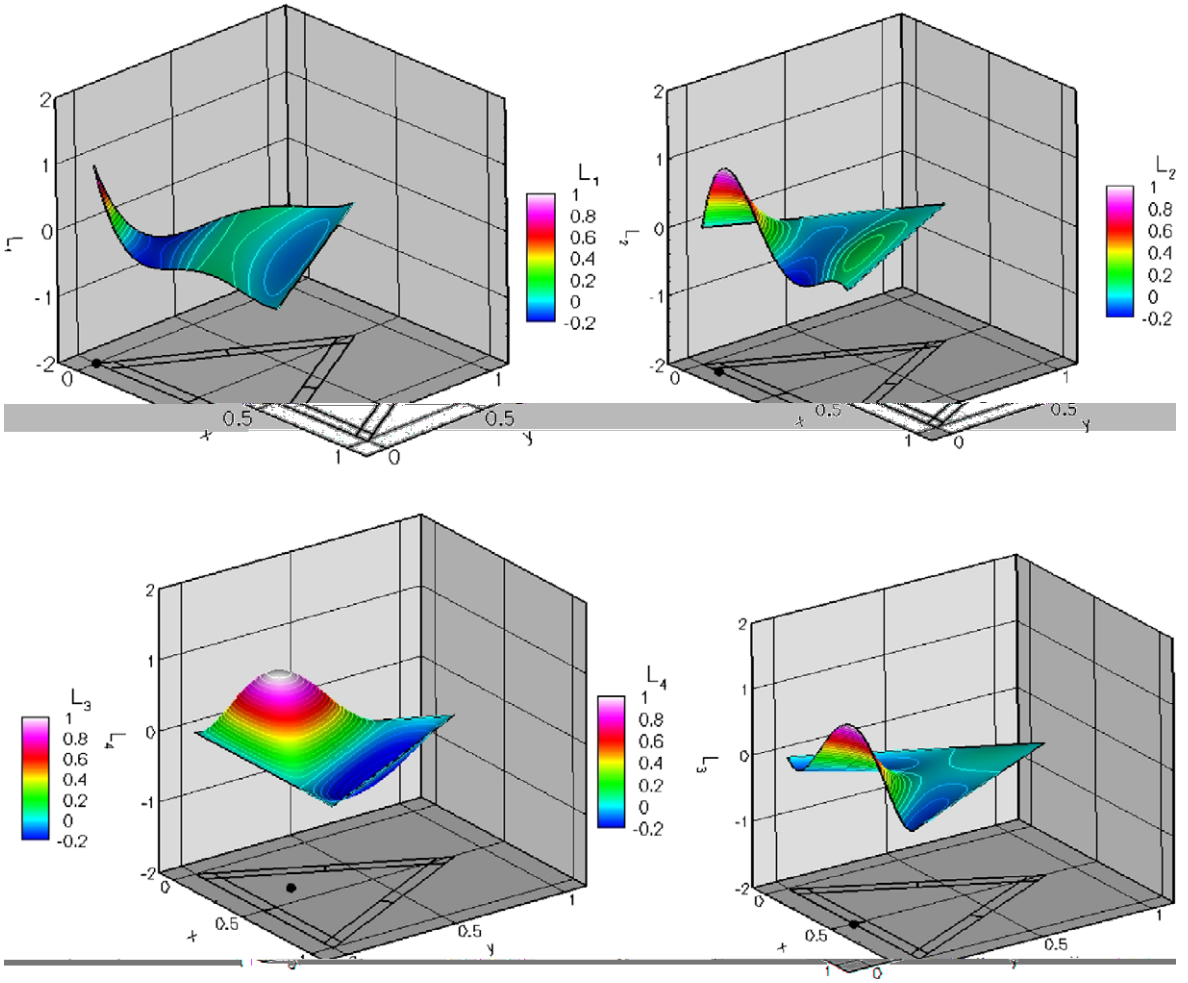

Fig. 5. Shape functions in a triangular SV supporting a quartic data reconstruction for the flux vector. The node which each function corresponds to is shown in bold.

Integrating (3.5b) over a face, we obtain

$$\int_{A_r} \alpha(Q_{\mathrm{R}} - Q_{\mathrm{L}})\mathrm{d}A = A_r \alpha_c(\overline{Q}_{\mathrm{R}} - \overline{Q}_{\mathrm{L}}) + O(A_r h^{m+2}). \tag{3.6}$$

At each node shown in Fig. 3, the flux is computed based on the reconstructed solution polynomial. Therefore, the flux error at each node of the nodal set is of order $O(h^{m+1})$, i.e.,

$$\begin{aligned} F_{n,\mathrm{L}} &= F_n + O(h^{m+1}), \\ F_{n,\mathrm{R}} &= F_n + O(h^{m+1}), \\ (F_{n,\mathrm{L}} + F_{n,\mathrm{R}})/2 &= F_n + O(h^{m+1}), \end{aligned} \tag{3.7}$$

where F_n is the (unknown) "exact" normal flux at the face. Integrating the 3rd equation in (3.7) over a face, we obtain

$$\int_{A_r} \frac{1}{2}(F_{n,\mathrm{L}} + F_{n,\mathrm{R}})\mathrm{d}A = \frac{1}{2}A_r(\overline{F}_{n,\mathrm{L}} + \overline{F}_{n,\mathrm{R}}) = \overline{F}_n A_r + O(A_r h^{m+1}). \tag{3.8}$$

So summarizing (3.6) and (3.8), we obtain

$$\int_{A_r} \vec{F}\cdot\vec{n}\,\mathrm{d}A \equiv \overline{F}_n A_r = \frac{A_r}{2}\left[\overline{F}_{n,\mathrm{L}} + \overline{F}_{n,\mathrm{R}} - \alpha_c(\overline{Q}_{\mathrm{R}} - \overline{Q}_{\mathrm{L}})\right]_r + O(A_r h^{m+1}). \tag{3.9}$$

If $\vec{F}$ = constant, the following identity exists:

$$\sum_{r=1}^{K}\int_{A_r} \vec{F}\cdot\vec{n}\,\mathrm{d}A = 0. \tag{3.10}$$

Therefore, we will gain an extra order of accuracy if we sum up the surface integrals for the faces of $C_{i,j}$; i.e.,

$$\sum_{r=1}^{K}\int_{A_r} \vec{F}\cdot\vec{n}\,\mathrm{d}A = \sum_{r=1}^{K}\frac{A_r}{2}\left[\overline{F}_{n,\mathrm{L}} + \overline{F}_{n,\mathrm{R}} - \alpha_c(\overline{Q}_{\mathrm{R}} - \overline{Q}_{\mathrm{L}})\right]_r + O(A_r h^{m+2}). \tag{3.11}$$

Since $O(V_{i,j}) = O(A_r h)$, we have

$$\frac{\mathrm{d}\overline{Q}_{i,j}}{\mathrm{d}t} + \frac{1}{V_{i,j}}\sum_{r=1}^{K}\frac{A_r}{2}\left[\overline{F}_{n,\mathrm{L}} + \overline{F}_{n,\mathrm{R}} - \alpha_c(\overline{Q}_{\mathrm{R}} - \overline{Q}_{\mathrm{L}})\right]_r = O(h^{m+1}). \tag{3.12}$$

Thus, if the surface integral in (2.8) is evaluated using (3.4), spatial accuracy of order $m+1$ is assured.

To summarize, the first and second terms in (3.4) are computed as a weighted average of the flux evaluated at the nodal set of the left, and right SV, respectively using (3.3), and the third and fourth terms are computed as a weighted average of the cell-averaged conserved variables of the right and left SV, respectively, i.e.

$$\int_{A_r} Q\,\mathrm{d}S = A_r\sum_{i=1}^{N}\overline{L}_i\overline{Q}_i, \tag{3.13}$$

where $\overline{L}_i$ are the face-averaged CV-based shape functions for face r. Again, these are computed during preprocessing for a standard element and then multiplied by the physical face area for computation.

In a quick comparison of the approaches, a 2D 3rd order SV method using the traditional "full quadrature" approach with 2 quadrature points per face necessitates 36 flux calculations per SV, the PQ approach requires 28 flux calculations per SV, and the QF approach requires only 10 flux calculations per SV. Similarly, for a 4th order SV method, the number of flux calculations per SV is 54, 39, and 15 for the "full quadrature", PQ, and QF approaches, respectively. So, the PQ and QF approaches are slightly more efficient than the "full quadrature" approach in 2D, but are much less costly to evaluate in 3D where the faces number in the hundreds. Table 1 summarizes the costs for the PQ and QF approaches in 2D and in 3D. It is evident that the QF approach is slightly less expensive than the PQ approach in 2D in terms of CPU time, as expected. Although the real savings presents itself in 3D, which can be seen from the required flux calculations presented in Table 1.

4. Data limiting

For the non-linear Euler equations, it is necessary to perform data limiting to maintain stability if the solution contains discontinuities. There are many different ways of applying limiters in the system setting. In this

Table 1
Comparison of costs for PQ and QF approaches for vortex propagation problem on $80\times80\times2$ regular grid with single-stage time integration and Rusanov flux

Order	Approach	CPU time (μs)/SV/iteration	Number of flux calculations/SV/iteration	
		2D	2D	3D
3	PQ	31.63	28	200
	QF	25.42	10	20
4	PQ	51.28	39	371
	QF	46.92	15	35

paper, we choose the primitive variable approach and enforce monotonicity using a TVD limiter. To this end, we first establish the following numerical monotonicity criterion for each SV

$$\bar{\psi}_i^{\min} \leqslant \psi_i(x_{rn}, y_{yn}, z_{rn}) \leqslant \bar{\psi}_i^{\max}, \tag{4.1}$$

where $\bar{\psi}_i^{\min}$and $\bar{\psi}_i^{\max}$ are the minimum and maximum SV-averaged primitive variables $\{p, \rho, u, v, w\}$ among all the neighboring SVs sharing a face (called face neighbors), or sharing a node (called node-neighbors), with S_i, including the SV under consideration, and $\psi_i(x_{rn}, y_{rn}, z_{rn})$ are the reconstructed primitive variables at the nodal set of the SV under consideration. The SV-averaged primitive variables are computed by

$$\bar{\psi}_{i,j} = \sum_{k=1}^{N_S} \overline{M}_{i,j,k}\psi_k \tag{4.2}$$

and

$$\bar{\psi}_i = \frac{1}{V_i}\sum_{j=1}^{N} \bar{\psi}_{i,j}V_{i,j}, \tag{4.3}$$

where $\bar{\psi}_{i,j}$ are the CV-averaged primitive variables, $\overline{M}_{i,j,k}$ are the CV-averaged node-based shape functions, ψ_k are the primitive variables evaluated at node k, and V_i is the volume of SV i. The CV-averaged node-based shape functions, $\overline{M}_{i,j,k}$, are computed during preprocessing for a standard element.

In the cases considered here, only p and ρ are used in determining whether a given SV is to be limited to guarantee the positivity of p and ρ. Additionally, $\bar{\psi}_i^{\min}$ and $\bar{\psi}_i^{\max}$ are scaled by $(1-\epsilon)$ and $(1+\epsilon)$, respectively, where $\epsilon \ll 1$. This prevents unnecessary limiting based purely on numerical round-off. Here we use $10^{-5} < \epsilon < 10^{-3}$. Obviously if ϵ is large, the limiter is no longer strictly TVD.

If (4.1) is violated for any node in the nodal set of the SV under consideration, then it is assumed that the SV is close to a discontinuity, and all primitive variables in the SV are linear, i.e.,

$$\nabla\psi_i = \text{Const}, \quad \forall r \in S_i. \tag{4.4}$$

The magnitude of this gradient is maximized subject to the monotonicity condition given in (4.1). The initial guess of the gradient $\nabla\psi_i$ for each SV is computed at the SV center using

$$\nabla\psi_i(x,y,z) = \sum_{j=1}^{N} \nabla L_{i,j}(x,y,z)\bar{\psi}_{i,j}, \tag{4.5}$$

where $\nabla L_{i,j}$ represent the gradients of the CV-based shape functions. The gradient $\nabla\psi_i$ may not satisfy (4.1). Therefore it is limited by multiplying by a scalar $0 \leqslant \varphi \leqslant 1$. The scalar is computed from

$$\varphi = \begin{cases} \min\left(1, \frac{\bar{\psi}_i^{\max} - \bar{\psi}_i}{\psi_i(x_{rn}, y_{rn}, z_{rn}) - \bar{\psi}_i}\right) & \text{if } \psi_i(x_{rn}, y_{rn}, z_{rn}) > \bar{\psi}_i, \\ \min\left(1, \frac{\bar{\psi}_i^{\min} - \bar{\psi}_i}{\psi_i(x_{rn}, y_{rn}, z_{rn}) - \bar{\psi}_i}\right) & \text{if } \psi_i(x_{rn}, y_{rn}, z_{rn}) < \bar{\psi}_i, \\ 1 & \text{otherwise.} \end{cases} \tag{4.6}$$

5. Formulation for curved boundary representation

One of the biggest advantages of high-order methods over low-order methods is their ability to achieve better solution accuracy using far fewer degrees-of-freedom when the solution is smooth. However, for problems involving curved wall boundaries which are represented by line-segments or planar-facets, an inordinate amount of elements may be necessary just to preserve the geometry with adequate precision. A much more desirable approach is to represent curved boundaries with high-order polynomials that are compatible with the order of the data reconstruction. Favorable results have been obtained [34] when applying this technique to the SV method. One drawback of this approach is that it necessitates computation and storage of a separate reconstruction for each and every wall-bounding cell. For large problems, and especially in 3D, the memory requirement can become large. Here we follow an approach by Krivodonova and Berger [19] originally devel-

oped for the DG method, in which the geometry is represented with high-fidelity, but a separate reconstruction for every wall-bounding cell is avoided.The basic idea employed here is that straight-sided SVs, rather than curved SVs, are employed at wall boundaries, but a no-flow boundary condition is enforced for the physical boundary as opposed to the computational (polygonal or polyhedral) boundary. This allows us to utilize the same reconstruction for the SVs on curved boundaries that is already in use for all other SVs.

This procedure involves a number of steps. First, a set of points is defined for each SV face to construct a polynomial representation of that face. Here we employ a cubic polynomial for this case. Thus, 4 points are needed to define a cubic polynomial for each SV face in 2D. Two of the points are taken as the SV face endpoints, while the 3rd and 4th points are computed from a cubic polynomial fit between the SV face endpoints and the opposite endpoints of both neighboring SV faces. If the SV face in question is at the end of the curved wall, or at a sharp trailing edge, then a 2-point extrapolation is used to replace the missing opposite endpoint of the adjacent SV face with a fictitious point. The 3rd and 4th points are then computed as in the above. Now that we have a cubic polynomial defined for each SV face on a curved boundary, the normal can be computed at any point on the curved face. To determine the points at which the normal vectors must be computed, points in the standard SV can be mapped into a simplified cubic SV (SCSV), as shown in Fig. 6, via the following

$$r = \sum_{j=1}^{5} C_j(\xi,\eta) r_j, \tag{5.1}$$

where

$$\begin{aligned} C_1(\xi,\eta) &= 1 - 5.5\xi - \eta + 9\xi(\xi+\eta) - 4.5\xi(\xi+\eta)^2, \\ C_2(\xi,\eta) &= \xi[1 - 4.5(\xi+\eta) + 4.5(\xi+\eta)^2], \\ C_3(\xi,\eta) &= \eta, \\ C_4(\xi,\eta) &= 9\xi[1 - 2.5(\xi+\eta) + 1.5(\xi+\eta)^2], \\ C_5(\xi,\eta) &= \xi[-4.5 + 18(\xi+\eta) - 13.5(\xi+\eta)^2], \end{aligned} \tag{5.2}$$

and obviously for points on the curved wall, $\eta = 0$. The unit-normal vector at any point on the curved face is then computed by

$$\vec{n} = \frac{\frac{dy}{d\xi}, -\frac{dx}{d\xi}}{\sqrt{\left(\frac{dy}{d\xi}\right)^2 + \left(\frac{dx}{d\xi}\right)^2}}. \tag{5.3}$$

It is important to note that the vector $\vec{n}$ is used only to define a ghost state at quadrature points, or nodes, located on curved-wall boundaries. The numerical flux is still computed through the straight-sided boundary. Thus, for the PQ approach, $\vec{n}$ must be computed and stored for each quadrature point on every curved-wall bounding SV. Whereas, in the QF approach, $\vec{n}$ must be computed and stored for each node in the nodal set of wall-bounding SVs that lies on the wall-bounding face. Additionally, $\vec{n}$ must be computed and stored at the

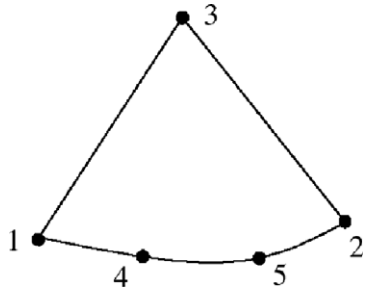

Fig. 6. Simplified cubic SV (SCSV) with one curved boundary.

CV face centers for curved-wall bounding SVs, to be used for computing the eigenvalue or dissipation matrix. Note that for simple geometries such as a cylinder, the above process is unnecessary as the normals can be computed directly.

6. Numerical tests

In this section, the efficient SV implementation is evaluated for 2D scalar advection equations, and for the 2D Euler equations. Both 3rd order and 4th order partitions are evaluated. The 3rd order partition, and the corresponding nodal set to support a cubic reconstruction for the flux vector are shown in Fig. 1b, and Fig. 3a, respectively. The 4th order partition, and the corresponding nodal set to support a quartic reconstruction for the flux vector are shown in Fig. 1c, and Fig. 3b, respectively. The numerical error is assessed using both regular and irregular grids, as shown in Fig. 7. The finer grids are generated recursively by cutting each courser grid cell into four finer grid cells. The 3rd order SSP Runge–Kutta scheme is used for time integration in all cases. Also, all errors presented are time-step independent because the time step Δt was made small enough so that the errors are dominated by the spatial discretization. The initial CV-averaged solutions were computed using the CV-averaged node-based shape functions. In this way, the solution is initialized at the nodal set, and the CV-averaged solutions are computed as a weighted average of these nodal values. This avoids the need for a high-order Gauss quadrature initialization. The solution is taken as converged when the L_2 norm of the residual is reduced by eight orders of magnitude in all cases. The L_2 norm of the density residual is used for the Euler equation simulations.

6.1. Accuracy study with linear advection equation

In this case, we test the accuracy of the SV method on the 2D linear advection equation:

$$\begin{aligned} &\frac{\partial Q}{\partial t}+\frac{\partial Q}{\partial x}+\frac{\partial Q}{\partial y}=0, \quad -1\leqslant x\leqslant 1,\ -1\leqslant y\leqslant 1,\\ &Q(x,y,0)=\sin\pi(x+y), \quad \text{periodic boundary condition.} \end{aligned} \tag{6.1}$$

The numerical solutions were carried out until $t = 1$ s for this time-accurate problem. In Tables 2 and 3, we present the L_1 and L_∞ errors and orders for regular and irregular grids, respectively. Grid refinement is carried out until both the L_1 and L_∞ orders are sufficiently demonstrated. For this case, the PQ results are identical to the QF results, as expected for a linear equation, so just the QF results are presented. The formal order of accuracy is observed, using both regular and irregular grids.

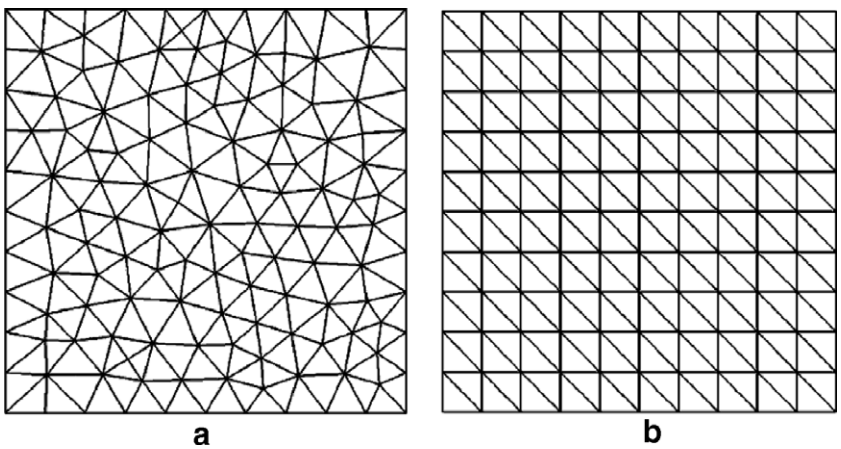

Fig. 7. Regular and irregular grids for validation cases: (a) regular ($10\times10\times2$) and (b) irregular ($10\times10\times2$).

Table 2
Accuracy of 2D linear advection equation at $t = 1$ s on regular grids

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	1.33e−3	–	3.29e−3	–
	$20 \times 20 \times 2$	1.73e−4	2.94	4.47e−4	2.88
	$40 \times 40 \times 2$	2.19e−5	2.98	5.77e−5	2.95
	$80 \times 80 \times 2$	2.76e−6	2.99	7.29e−6	2.98
4	$10 \times 10 \times 2$	7.10e−5	–	2.44e−4	–
	$20 \times 20 \times 2$	4.38e−6	4.02	1.56e−5	3.97
	$40 \times 40 \times 2$	2.73e−7	4.00	9.78e−7	3.99
	$80 \times 80 \times 2$	1.71e−8	4.00	6.12e−8	4.00

Table 3
Accuracy of 2D linear advection equation at $t = 1$ s, on irregular grids

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	3.12e−3	–	1.57e−2	–
	$20 \times 20 \times 2$	4.09e−4	2.93	1.93e−3	3.02
	$40 \times 40 \times 2$	5.36e−5	2.93	3.37e−4	2.52
	$80 \times 80 \times 2$	6.99e−6	2.94	6.18e−5	2.45
4	$10 \times 10 \times 2$	2.62e−4	–	1.50e−3	–
	$20 \times 20 \times 2$	1.56e−5	4.07	1.13e−4	3.73
	$40 \times 40 \times 2$	9.67e−7	4.02	8.23e−6	3.77
	$80 \times 80 \times 2$	6.03e−8	4.00	5.38e−7	3.94

6.2. Accuracy study with Burgers equation

In this case, we test the accuracy of the SV method on the 2D Burgers equation:

$$\begin{aligned} &\frac{\partial Q}{\partial t} + \frac{\partial Q^2/2}{\partial x} + \frac{\partial Q^2/2}{\partial y} = 0, \quad -1 \leqslant x \leqslant 1, \ -1 \leqslant y \leqslant 1, \\ &Q(x, y, 0) = \frac{1}{4} + \frac{1}{2}\sin \pi(x + y), \quad \text{periodic boundary condition.} \end{aligned} \tag{6.2}$$

The numerical solutions were carried out until $t = 0.1$ s for this time-accurate problem. At this time the solution is still smooth, so data limiting is not necessary. In Tables 4 and 5, we present the L_1 and L_∞ errors and orders for regular and irregular grids, respectively. Grid refinement is carried out until both the L_1 and L_∞ orders are sufficiently demonstrated. Results from the PQ and QF approaches are not identical for this case, but they are very similar, so again only the QF results are shown. The formal order of accuracy is again observed, using both regular and irregular grids.

Table 4
Accuracy of 2D Burgers equation at $t = 0.1$ s on regular grids

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	3.96e−4	–	1.63e−3	–
	$20 \times 20 \times 2$	6.49e−5	2.61	3.97e−4	2.03
	$40 \times 40 \times 2$	1.03e−5	2.66	7.01e−5	2.50
	$80 \times 80 \times 2$	1.55e−6	2.73	1.12e−5	2.65
4	$10 \times 10 \times 2$	3.28e−5	–	3.77e−4	–
	$20 \times 20 \times 2$	2.39e−6	3.78	3.26e−5	3.53
	$40 \times 40 \times 2$	1.74e−7	3.77	2.89e−6	3.49
	$80 \times 80 \times 2$	1.23e−8	3.83	2.08e−7	3.80

Table 5
Accuracy of 2D Burgers equation at $t = 0.1$ s on irregular grids

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	6.42e−4	–	3.35e−3	–
	$20 \times 20 \times 2$	1.17e−4	2.45	9.96e−4	1.75
	$40 \times 40 \times 2$	1.88e−5	2.65	2.66e−4	1.90
	$80 \times 80 \times 2$	2.91e−6	2.69	6.12e−5	2.12
	$160 \times 160 \times 2$	4.41e−7	2.72	1.14e−5	2.43
	$320 \times 320 \times 2$	6.51e−8	2.76	1.88e−6	2.59
4	$10 \times 10 \times 2$	8.35e−5	–	1.16e−3	–
	$20 \times 20 \times 2$	6.20e−6	3.75	1.43e−4	3.02
	$40 \times 40 \times 2$	4.57e−7	3.76	1.26e−5	3.51
	$80 \times 80 \times 2$	3.28e−8	3.80	1.06e−6	3.56

6.3. Accuracy study with vortex propagation problem

The unsteady 2D Euler equations in conservative form can be written as

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{6.3}$$

where Q is the vector of conserved variables, F and G are the inviscid flux vectors given below:

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix}, \quad F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{Bmatrix}, \quad G = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{Bmatrix}. \tag{6.4}$$

Here ρ is the density, u and v are the velocity components in x and y directions, p is the pressure, and E is the total energy. The pressure is related to the total energy by

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2), \tag{6.5}$$

with ratio of specific heats γ. In all of the simulations in this paper, γ is taken to be 1.4. This is an idealized problem for the Euler equations in 2D. the mean flow is $\{\rho, u, v, p\} = \{1, 1, 1, 1\}$. An isotropic vortex is then added to the mean flow, i.e. with perturbations in u, v, and temperature $T = p/\rho$, and no perturbation in entropy $S = p/\rho^\gamma$:

$$(\delta u, \delta v) = \frac{\varepsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}),$$
$$\delta T = -\frac{(\gamma - 1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2},$$
$$\delta S = 0,$$

where $r^2 = \bar{x}^2 + \bar{y}^2$, $\bar{x} = x - 5$, $\bar{y} = y - 5$, and the vortex strength $\varepsilon = 5$. If the computational domain extends to infinity, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with mean velocity $(1, 1)$. In the following accuracy study, the computational domain is taken to be $[0, 10] \times [0, 10]$, with characteristic inflow and outflow boundary conditions imposed on the boundaries. The numerical solution is carried out until $t = 2$ on both regular and irregular grids. No limiters are employed in this study since the problem is smooth, and the Rusanov flux is used in all simulations. In Tables 6 and 7, we present the L_1 and L_∞ errors and orders in the CV-averaged density for the regular grids using the PQ and QF approaches, respectively. Tables 8 and 9 show the L_1 and L_∞ errors and orders for the irregular grids using the PQ and QF approaches, respectively. Comparison of the PQ and QF approaches for this case show very similar behavior. The formal order of accuracy is demonstrated for all cases.

Table 6
Accuracy of 2D vortex propagation problem at $t = 2$ s on regular grids using PQ approach

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	1.41e−03	–	4.09e−02	–
	$20 \times 20 \times 2$	2.51e−04	2.49	4.88e−03	3.06
	$40 \times 40 \times 2$	4.30e−05	2.54	1.05e−03	2.22
	$80 \times 80 \times 2$	6.88e−06	2.65	1.95e−04	2.42
	$160 \times 160 \times 2$	1.05e−06	2.71	3.16e−05	2.63
4	$10 \times 10 \times 2$	6.08e−04	–	9.98e−03	–
	$20 \times 20 \times 2$	4.95e−05	3.62	1.12e−03	3.16
	$40 \times 40 \times 2$	3.03e−06	4.03	1.14e−04	3.30
	$80 \times 80 \times 2$	1.92e−07	3.98	7.28e−06	3.96
	$160 \times 160 \times 2$	1.36e−08	3.82	6.22e−07	3.55

Table 7
Accuracy of 2D vortex propagation problem at $t = 2$ s on regular grids using QF approach

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	1.44e−03	–	5.00e−02	–
	$20 \times 20 \times 2$	2.53e−04	2.51	5.05e−03	3.31
	$40 \times 40 \times 2$	4.32e−05	2.55	1.07e−03	2.24
	$80 \times 80 \times 2$	6.89e−06	2.65	1.98e−04	2.44
	$160 \times 160 \times 2$	1.05e−06	2.71	3.17e−05	2.64
4	$10 \times 10 \times 2$	5.49e−04	–	9.16e−03	–
	$20 \times 20 \times 2$	4.82e−05	3.51	8.90e−04	3.36
	$40 \times 40 \times 2$	3.04e−06	3.99	1.07e−04	3.06
	$80 \times 80 \times 2$	1.92e−07	3.99	6.97e−06	3.94
	$160 \times 160 \times 2$	1.36e−08	3.82	6.12e−07	3.51

Table 8
Accuracy of 2D vortex propagation problem at $t = 2$ s on irregular grids using PQ approach

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	1.11e−03	–	2.67e−02	–
	$20 \times 20 \times 2$	1.85e−04	2.58	3.23e−03	3.04
	$40 \times 40 \times 2$	3.00e−05	2.63	6.91e−04	2.23
	$80 \times 80 \times 2$	4.76e−06	2.66	1.32e−04	2.39
	$160 \times 160 \times 2$	7.36e−07	2.69	1.95e−05	2.75
4	$10 \times 10 \times 2$	3.46e−04	–	8.38e−03	–
	$20 \times 20 \times 2$	3.07e−05	3.49	4.52e−04	4.21
	$40 \times 40 \times 2$	2.07e−06	3.89	4.87e−05	3.21
	$80 \times 80 \times 2$	1.42e−07	3.87	3.89e−06	3.65
	$160 \times 160 \times 2$	9.49e−09	3.90	3.66e−07	3.41

6.4. Zalesak's disk rotation problem

The rigid body rotation of Zalesak's disk (notched circle in 2D) under a constant vorticity velocity field [38] is employed to demonstrate the ability of the limiter to preserve discontinuities. The computational domain is taken to be $[0,1] \times [0,1]$, and the notched circle centers at $(0.5, 0.75)$ with radius 0.15 and the slot has a width of 0.05 and length of 0.2479. The velocity is given by

$$\begin{aligned} u &= 0.5 - y, \\ v &= x - 0.5. \end{aligned} \tag{6.6}$$

The conserved variable is initialized to $+1$ inside the notched circle, and -1 everywhere else. The numerical solution is carried out until $t = 2\pi$, which constitutes one full revolution of the disk. Fig. 8 shows results

Table 9
Accuracy of 2D vortex propagation problem at $t = 2$ s on irregular grids using QF approach

Order of accuracy	Grid	L_1 error	L_1 order	L_∞ error	L_∞ order
3	$10 \times 10 \times 2$	1.11e−03	–	3.21e−02	–
	$20 \times 20 \times 2$	1.87e−04	2.57	3.36e−03	3.26
	$40 \times 40 \times 2$	3.01e−05	2.63	7.15e−04	2.23
	$80 \times 80 \times 2$	4.76e−06	2.66	1.33e−04	2.43
	$160 \times 160 \times 2$	7.36e−07	2.69	1.96e−05	2.76
4	$10 \times 10 \times 2$	3.44e−04	–	6.75e−03	–
	$20 \times 20 \times 2$	2.96e−05	3.54	4.42e−04	3.93
	$40 \times 40 \times 2$	2.05e−06	3.85	4.60e−05	3.26
	$80 \times 80 \times 2$	1.41e−07	3.86	3.75e−06	3.61
	$160 \times 160 \times 2$	9.49e−09	3.90	3.58e−07	3.39

for 2nd and 3rd order QF schemes on regular $80 \times 80 \times 2$ and $160 \times 160 \times 2$ grids. It is evident that both the 2nd and 3rd order schemes capture the sharp features of the disk reasonably well. The 3rd order SV scheme obviously captures the discontinuity with higher resolution than the 2nd order one, as expected.

6.5. Mach 3 wind tunnel with a step

This problem was studied extensively by Woodward and Colella [36], and has been widely used to assess the performance of shock-capturing methods. The 2D wind tunnel is 3 units long and 1 unit wide, with a step of 0.2 units high located at 0.6 units from the tunnel inlet. The initial condition is a Mach 3 right-going uniform flow. Inviscid wall boundary conditions (reflective) are used for tunnel wall boundaries, while inflow and outflow boundary conditions are used at the inlet and exit of the wind tunnel. It is well known that the corner of the step is a singularity, and often leads to a spurious Mach stem at the downstream bottom wall, and an erroneous entropy layer at the bottom wall. In [36], various numerical treatments were used to remedy these artifacts. In the present study, no special treatments were used for the singularity to see how the singularity affects the numerical solutions.

It was shown in [33] that the 4th order partition in Fig. 1c produces large errors for this case, due likely to the non-uniform sub-cell mesh, so this study will focus on the 2nd and 3rd order partitions shown in Fig. 1a and b, respectively. Two different grids are considered here, consisting of 8746 and 37,146 triangles, the coarsest of which is shown in Fig. 9. Thus, the number of degrees-of-freedom for the 2nd order case is 26,238 and 111,438, respectively, and the number of degrees-of-freedom for the 3rd order case is 52,476 and 222,876, respectively. All simulations are carried out until $t = 4$. The results obtained using the QF approach with Rusanov flux are nearly identical to those obtained using the PQ approach, so only the QF results with Rusanov flux are shown here. The node-neighbor stencil is used for all cases here, as the face-neighbor stencil caused too many cells to be limited, essentially reducing the schemes to low order.

The computed density contours for the 2nd and 3rd order simulations on all grids are shown in Figs. 10 and 11, respectively. Obviously, the resolution of the 3rd order results is better than that of the 2nd order results while the 3rd order results are more oscillatory than the 2nd order results. This behavior is consistent with the results found in [33]. For example, the spurious Mach stems in the 3rd order results are considerably weaker than those in the 2nd order results. Also, the instability generated by the slip line begins to become visible in the 3rd order results on the fine mesh. It should be noted that the resolution of the results in [33] is slightly better than those presented here due to the presence of sub-cell resolution in the limiter. Nevertheless, the results presented here show similar trends for the density contours, and the strength of the spurious Mach stem is dramatically reduced under *h*-refinement.

6.6. Subsonic flow over a circular cylinder

A subsonic inviscid flow around a circular cylinder at Mach = 0.3 is selected to test the curved boundary representation. Due to flow symmetry, only the top half of the physical domain is selected as the computa-

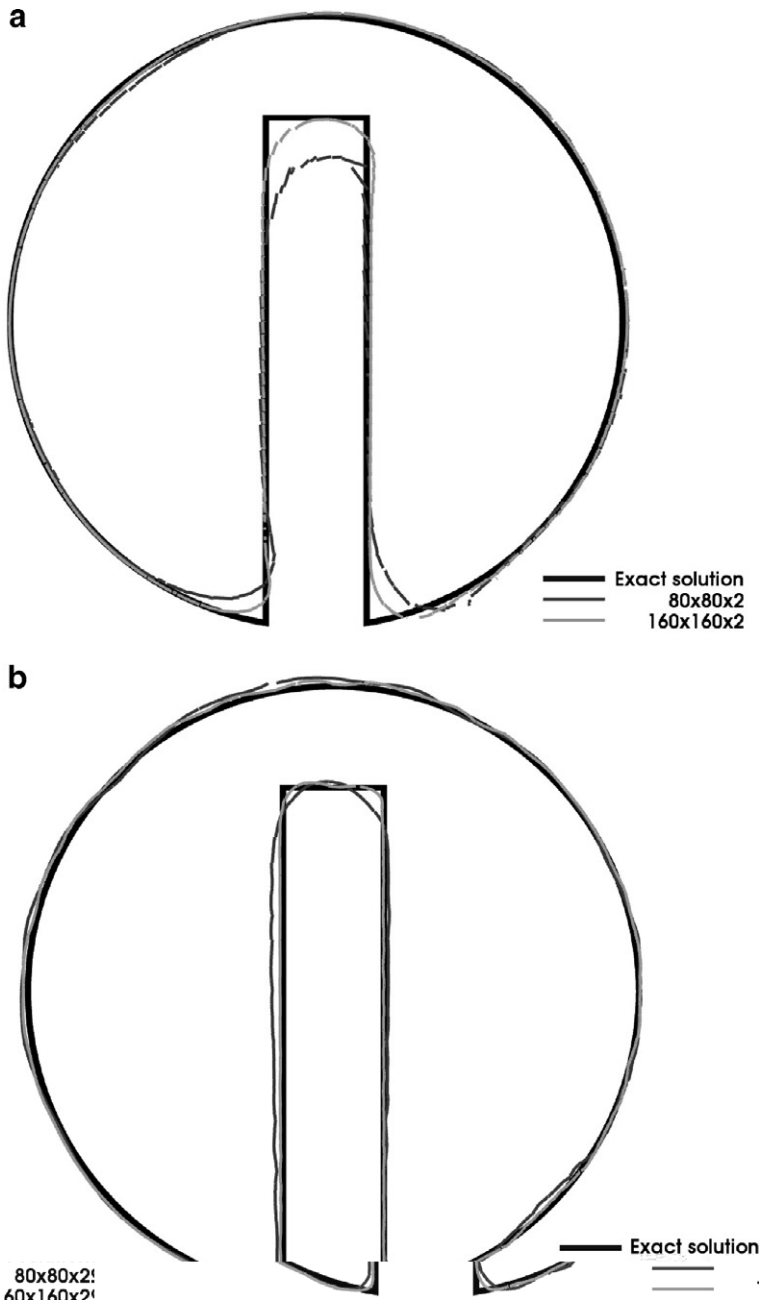

Fig. 8. Zero contour for Zalesak's disk rotation problem. (a) 2nd order and (b) 3rd order.

tional domain although the complete physical domain is displayed in all of the figures. The far field boundary is located 20 diameters away from the cylinder center and the characteristic inflow/outflow boundary conditions are used at the far field boundary. In order to perform a grid refinement accuracy study, three different triangular grids were generated from three structured grids with 32×8, 64×16 and 128×32 cells, which are shown in Fig. 12. The solution is initialized to a uniform free stream. All simulations employ the simplified curved boundary treatment given in Section 6. The simulations were also run using a piece-wise linear boundary representation, but the solution quickly diverged in all cases. Also, even with the curved

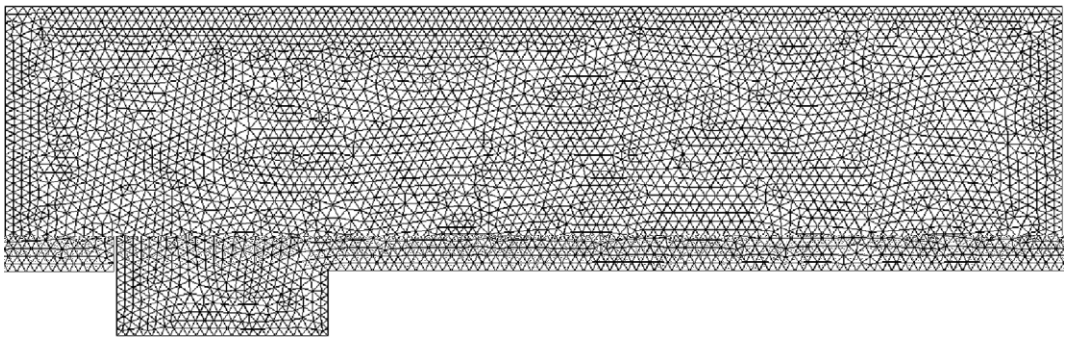

Fig. 9. A coarse mesh for the supersonic wind tunnel problem with 8746 triangles.

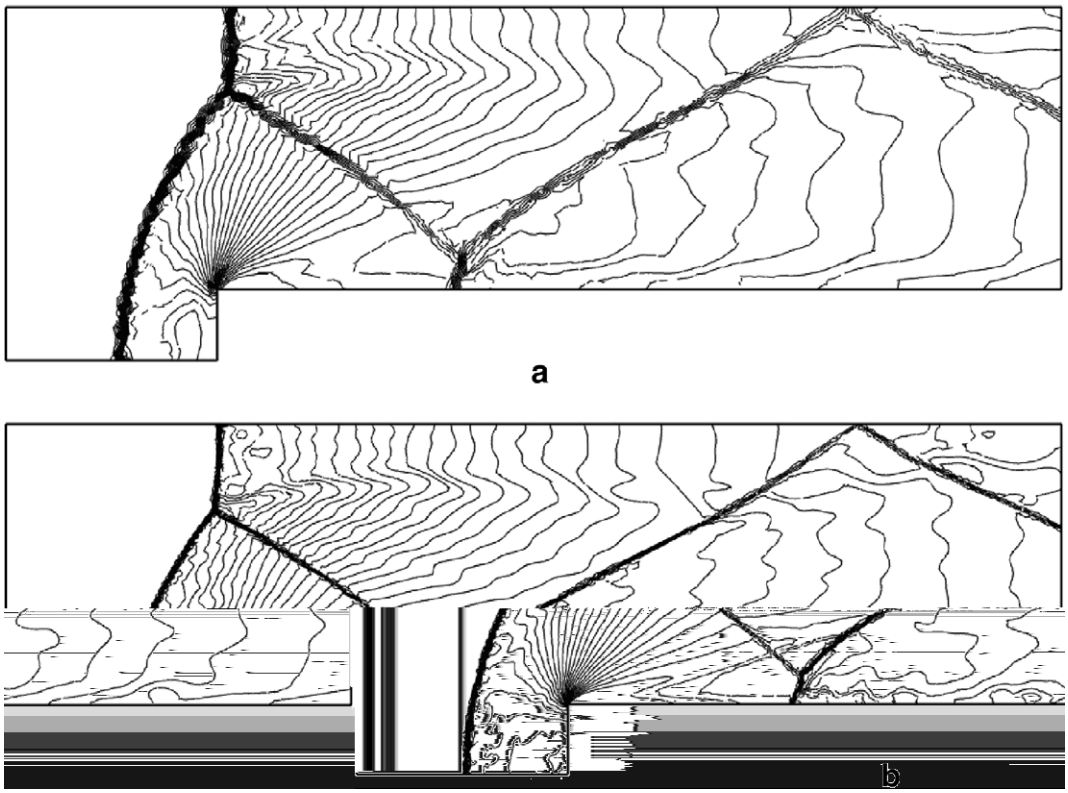

Fig. 10. Density contours for the 2nd order scheme. Thirty even contour lines between 0.09 and 4.53; (a) 26,238 DOFs and (b) 111,438 DOFs.

boundary treatment, the use of the Rusanov flux in the QF approach can lead to instabilities due to spurious entropy production at the trailing edge. This is overcome by the use of either the Roe [22], or HLLC [15,26,27,5] flux.

Figs. 13 and 15 show results for the PQ approach and the HLLC flux for both 3rd and 4th order simulations, respectively, and Figs. 14 and 16 show results for the QF approach and the Roe flux for both 3rd and 4th order simulations, respectively. All plots show fourteen even contours of Mach number between 0 and 0.65. The coarse grids show some discontinuous contours and a slight asymmetry at the trailing edge. As the grid is refined, the contours become continuous and the flow becomes perfectly symmetric with respect to both axes. For the 3rd order case, grid independence is achieved on the medium grid, while for the 4th order

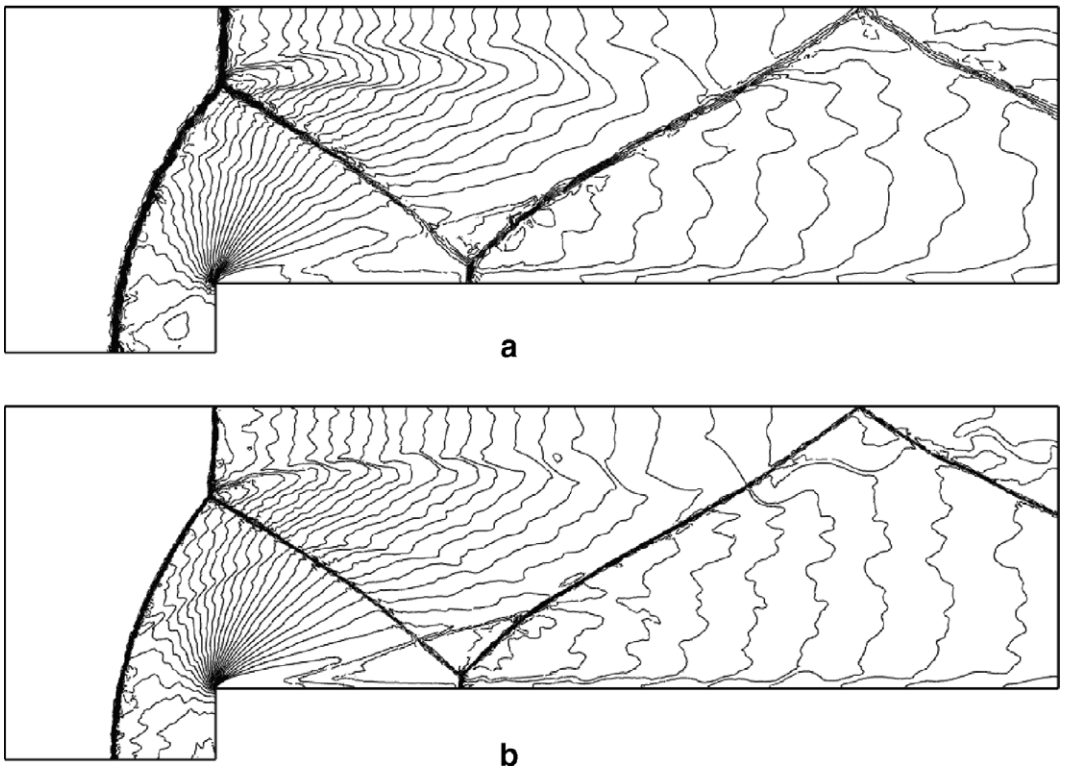

Fig. 11. Density contours for the 3rd order scheme. Thirty even contour lines between 0.09 and 4.53; (a) 52,476 DOFs and (b) 222,876 DOFs.

Fig. 12. Computational grids for flow over a circular cylinder; (a) $32 \times 8 \times 2$ cells; (b) $64 \times 16 \times 2$ cells and (c) $128 \times 32 \times 2$ cells.

case, even the coarse grid results are nearly grid independent. It is interesting to note that these results appear to be slightly better than those obtained using the full isoparametric SV approach on the same grids [34].

6.7. Subsonic flow over a NACA 0012 airfoil

As a final demonstration for a more realistic geometry, subsonic flow around a NACA 0012 airfoil at Mach = 0.4, and angle of attack of 5° is simulated. In this simulation, the 3rd and 4th order schemes are tested

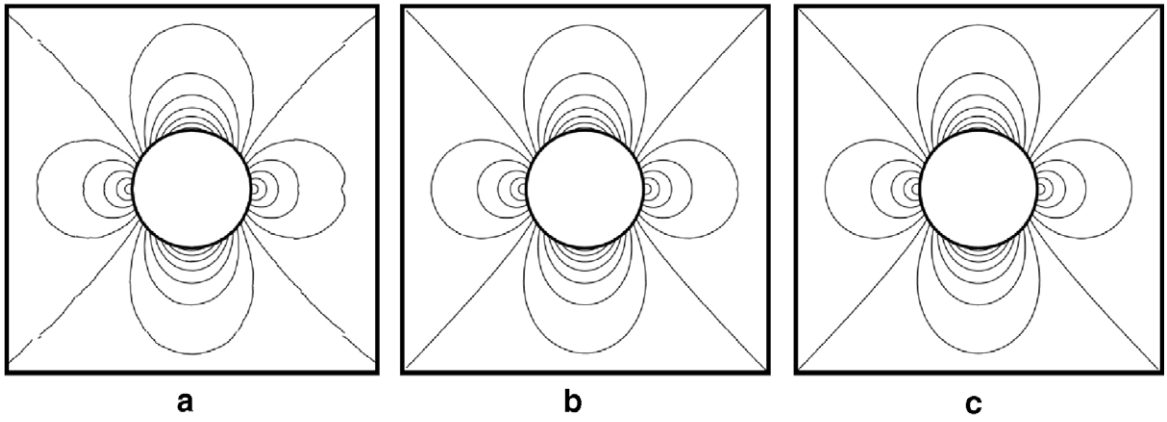

Fig. 13. Mach contours computed with 3rd order PQ approach and HLLC flux. (a) 3072 DOFs; (b) 12,288 DOFs and (c) 49,152 DOFs.

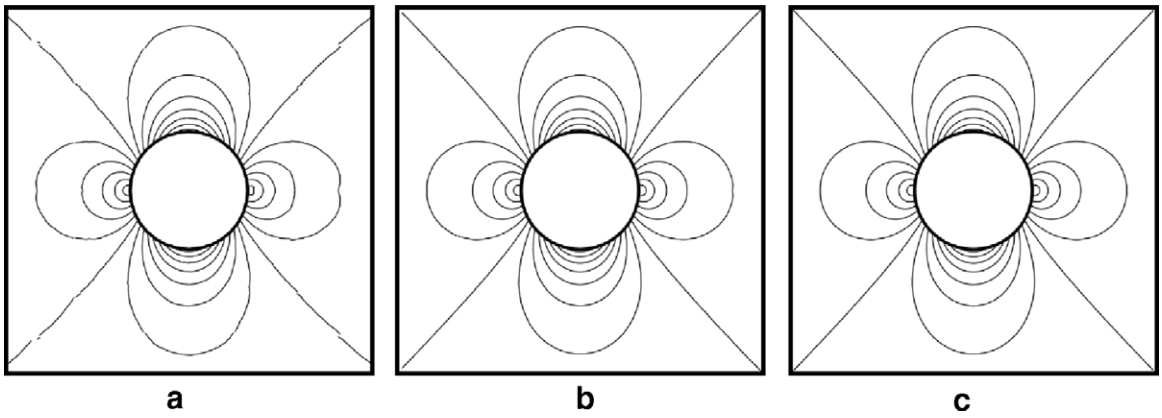

Fig. 14. Mach contours computed with 3rd order QF approach and Roe flux. (a) 3072 DOFs, (b) 12,288 DOFs and (c) 49,152 DOFs.

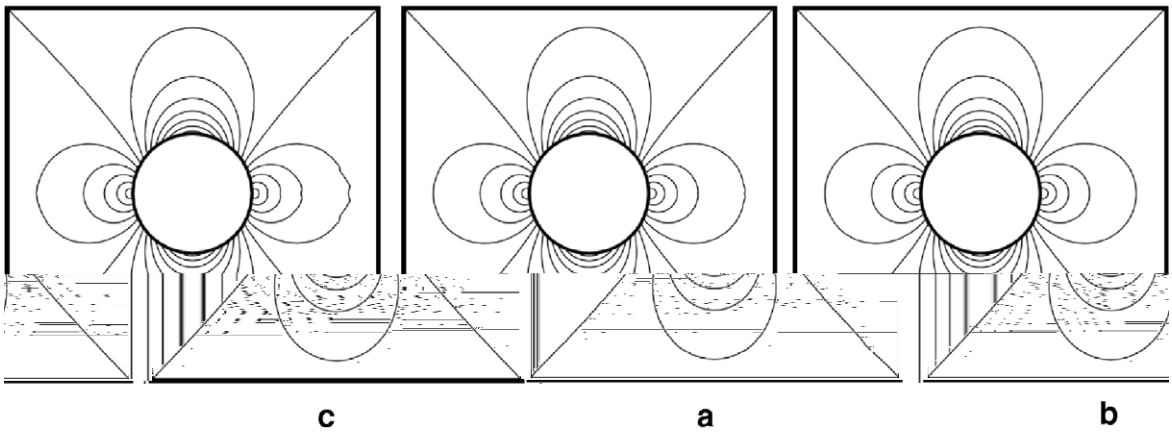

Fig. 15. Mach contours computed with 4th order PQ approach and HLLC flux. (a) 5120 DOFs; (b) 20,480 DOFs and (c) 81,920 DOFs.

using a very course mesh with $48 \times 16 \times 2$ triangles, as shown in Fig. 17, to further demonstrate the effectiveness of the curved boundary treatment. The outer boundary is 20 chords away from the center of the airfoil. Thirty-three even pressure contours between 0.5 and 0.8 computed using both a linear and quadratic bound-

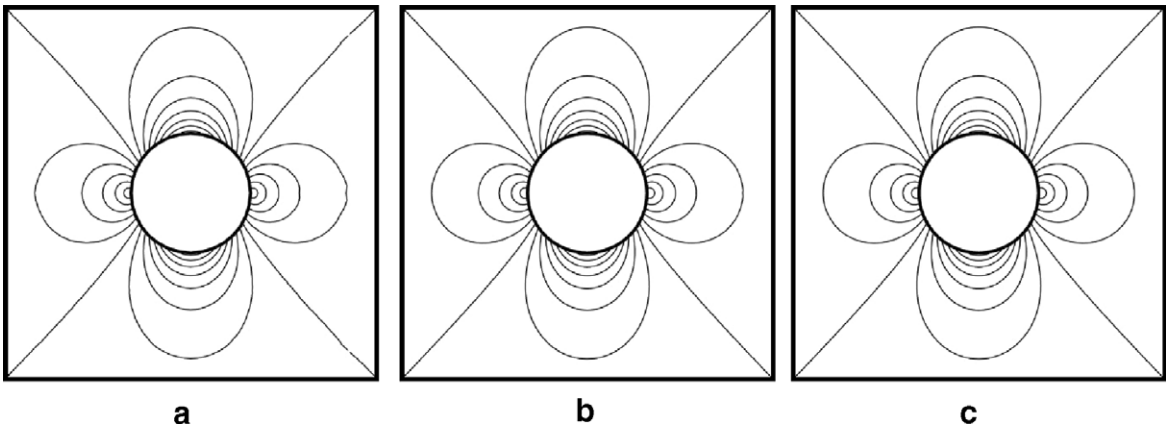

Fig. 16. Mach contours computed with 4th order QF approach and Roe flux. (a) 5120 DOFs; (b) 20,480 DOFs and (c) 81,920 DOFs.

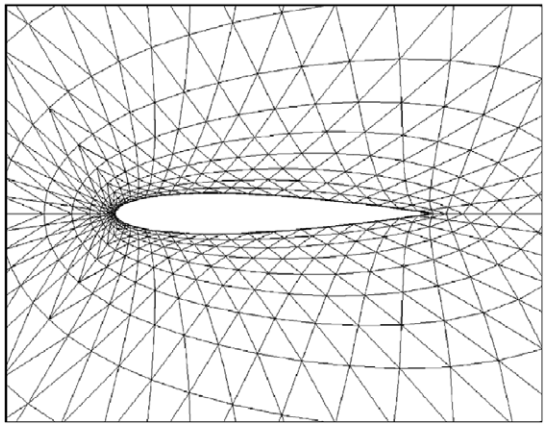

Fig. 17. Computational grid for flow over a NACA 0012 airfoil ($48 \times 16 \times 2$ triangles).

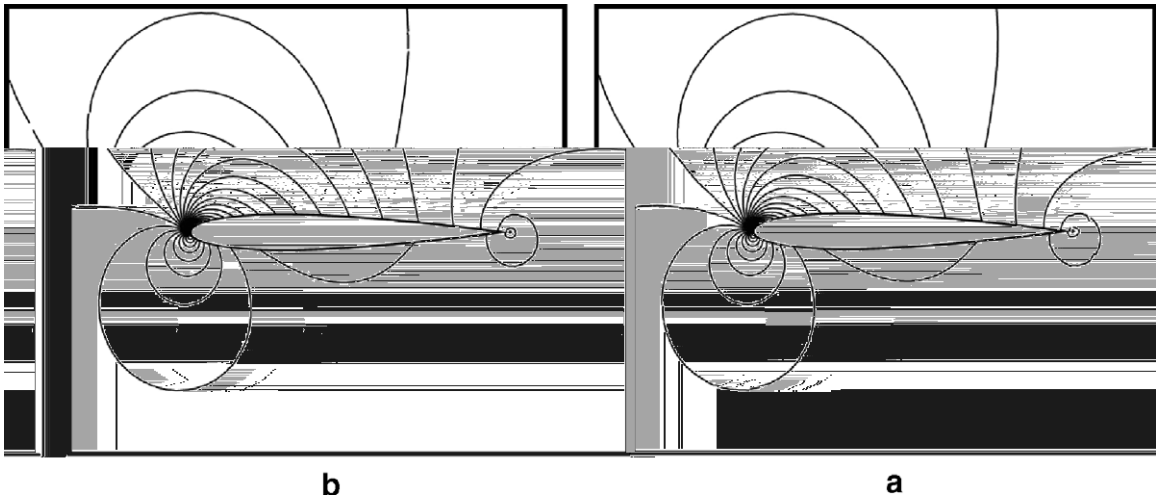

Fig. 18. Pressure contours for QF approach and Roe flux: (a) 3rd order and (b) 4th order.

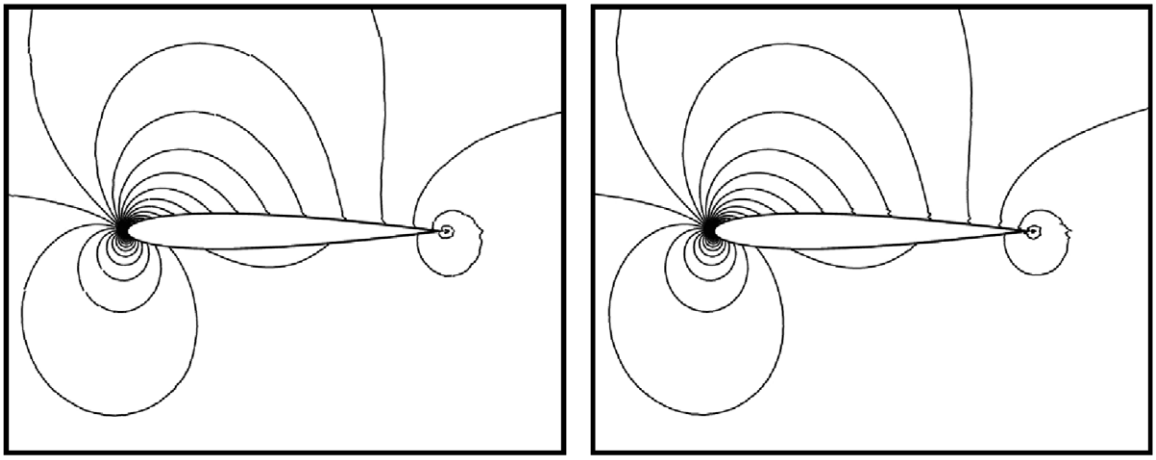

Fig. 19. Mach contours for QF approach and Roe flux: (a) 3rd order and (b) 4th order.

ary representation are shown in Fig. 18a and b, respectively. Fig. 19a and b show 33 even contours of Mach number between 0 and 0.8. All results are for the QF approach and Roe flux. As can be seen in Fig. 18, the pressure contours become very smooth and continuous under p-refinement. Some spurious entropy production has caused a slight irregularity in the Mach contours at the airfoil surface, which can most likely be attributed to the extreme coarseness of the grid.

7. Conclusions

An efficient implementation of the spectral volume method has been successfully carried out for the 2D scalar and Euler equations. Two different approaches to compute the flux integrals more efficiently, the partial quadrature (PQ) and quadrature-free (QF) approaches, have been developed and evaluated for both 3rd and 4th order SV schemes. For a 3rd order SV scheme, the PQ and QF approaches require 22% and 72% fewer flux calculations than the traditional approach, respectively, and for a 4th order SV scheme, the PQ and QF approaches require 28% and 72% fewer flux calculations than the traditional approach, respectively. The savings is obviously significant in 2D, but it does not end here. In 3D, the hundreds or thousands of flux calculations per SV required in the traditional approach will literally be reduced to only dozens. For all approaches, it has been found that the nearly optimum order of accuracy can be obtained in both the L_1 and L_∞ norms with respect to density errors. A total variation diminishing (TVD) limiter has been implemented in both the PQ and QF framework, and demonstrated for an unsteady supersonic problem with strong shocks. A simplified curved boundary treatment, which significantly reduces complexity of the numerical implementation, has been developed for both the PQ and QF framework and employed with success. The 3rd and 4th order schemes using the QF approach and Roe flux have been implemented for the case of subsonic flow around a NACA 0012 airfoil to further demonstrate the effectiveness of the simplified high-order boundary treatment. The linear boundary representation was also used with 3rd and 4th order SV schemes for the case of subsonic flow over a cylinder, but we were only able to obtain convergent numerical solutions using the high-order boundary treatment for this case. The extension of the QF approach to the 3D scalar and Euler equations is now under way, and will be reported in a future publication.

Acknowledgements

This study has been supported by the Department of Energy (DOE) and the Air Force Office of Scientific Research (AFOSR). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the DOE and AFOSR.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[2] H.L. Atkins, Chi-Wang Shu, Quadrature-free implementation of the discontinuous Galerkin method for hyperbolic equations, AIAA J. 96 (1996) 1683.
[3] T.J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.
[4] F. Bassi, S. Rebay, High-order accurate discontinuous finite element solution of the 2D Euler equations, J. Comput. Phys. 138 (1997) 251–285.
[5] P. Batten, N. Clarke, C. Lambert, D.M. Causon, On the choice of wavespeeds for the HLLC Riemann solver, SIAM J. Sci. Comput. 18 (6) (1997) 1553–1570.
[6] Q.Y. Chen, Partitions for spectral (finite) volume reconstruction in the tetrahedron, SIAM J. Sci. Comput. (2005).
[7] B. Cockburn, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.
[8] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
[9] B. Cockburn, S. Hou, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.
[10] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[11] M. Delanaye, Yen Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.
[12] S.K. Godunov, A finite-difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 271.
[13] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, SIAM Rev. 43 (1) (2001) 89–112.
[14] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.
[15] A. Harten, P.D. Lax, B. Van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35–61.
[16] J.S. Hesthaven, From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex, SIAM J. Numer. Anal. 35 (2) (1998) 655–676.
[17] J.S. Hesthaven, C.H. Teng, Stable spectral methods on tetrahedral elements, SIAM J. Sci. Comput. 21 (6) (2000) 2352–2380.
[18] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[19] L. Krivodonova, M. Berger, High-order accurate implementation of solid wall boundary conditions in curved geometries, J. Comput. Phys. 211 (2006) 492–512.
[20] Y. Liu, M. Vinokur, Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids V: extension to three-dimensional systems, J. Comput. Phys. 212 (2006) 454–472.
[21] H. Luo, J. Baum, R. Löhner, On the computation of steady-state compressible flows using a discontinuous Galerkin method, presented at the Fourth International Conference on Computational Fluid Dynamics, July 10–14, Ghent, Belgium, 2006.
[22] P.L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[23] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.
[24] Y. Sun, Z.J. Wang, Y. Liu, High-order multidomain spectral difference method for the Navier–Stokes equations on unstructured hexahedral grids, Commun. Comput. Phys. 2 (2007) 310–333.
[25] Y. Sun, Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow, J. Comput. Phys. 215 (2006) 41–58.
[26] E.F. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL Riemann solver, Shock Waves 4 (1994) 25–34.
[27] S. Tu, S. Aliabadi, A slope limiting procedure in discontinuous Galerkin finite element method for gasdynamics applications, Int. J. Numer. Anal. Mod. 2 (2) (2005) 163–178.
[28] B. van Leer, Towards the ultimate conservative difference scheme V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[29] B. van Leer, Upwind and high-resolution methods for compressible flow: From donor cell to residual-distribution schemes, Commun. Comput. Phys. 1 (2006) 192–206.
[30] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.
[31] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665.
[32] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: extension to one-dimensional systems, J. Sci. Comput. 20 (2004) 137.
[33] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2004) 716.

[34] Z.J. Wang, Y. Liu, Extension of the spectral volume method to high-order boundary representation, J. Comput. Phys. 211 (2006) 154–178.
[35] S. Wolfram, Mathematica Book, fourth ed., Wolfram Media and Cambridge University Press, New York, 1999.
[36] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.
[37] Y. Xing, C.W. Shu, A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms, Commun. Comput. Phys. 1 (2006) 100–134.
[38] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 32 (1979) 335–362.
[39] M. Zhang, C.W. Shu, An analysis and a comparison between the discontinuous Galerkin method and the spectral finite volume methods, Comput. Fluids 34 (4–5) (2005) 581–592.