Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 194 (2004) 716–741
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems

Z.J. Wang [a,*], Laiping Zhang [a], Yen Liu [b,1]

[a] Department of Mechanical Engineering, Michigan State University, 2555 Engineering Building, East Lansing, MI 48824, USA
[b] NASA Ames Research Center, Mail Stop T27B-1, Moffett Field, CA 94035, USA

Received 28 September 2002; received in revised form 14 July 2003; accepted 29 September 2003

Abstract

In this paper, the fourth in a series, the spectral volume (SV) method is extended to multi-dimensional systems – the 2D Euler equations. The focus of this paper is to study the performance of the SV method on multidimensional non-linear systems, and to verify that high order solution accuracy up to fourth-order can be achieved for the systems of conservation laws. Implementation details including total variation diminishing (TVD) and total variation bounded (TVB) limiters are presented. An accuracy study is performed first to numerically verify that the designed order of accuracy can be obtained for smooth flow solutions. Then, solutions with both smooth features and discontinuities are utilized to demonstrate the overall capability of the SV method.
© 2003 Elsevier Inc. All rights reserved.

AMS: 65M60; 65M70; 35L65

Keywords: High-order; Unstructured grid; Spectral volume; 2D systems of conservation laws; Euler equations

1. Introduction

We continue the development of the spectral (finite) volume (SV) method for hyperbolic conservation laws on unstructured grids following the basic formulation [21], development for two-dimensional (2D) scalar conservation laws [22], and one-dimensional (1D) systems and partition optimization [23]. The ultimate goal of this research is to pursue a numerical method for conservation laws which has all of the following properties: (a) conservative; (b) high-order accuracy; (c) geometrically flexible; (d) computationally efficient. Such a method is sought after in many "real world" problems with complex geometries in computational aero-

* Corresponding author. Tel.: +1-517-432-9132; fax: +1-517-353-1750.
E-mail addresses: zjw@egr.msu.edu (Z.J. Wang), liu@nas.nasa.gov (Y. Liu).
[1] Tel.: +1-650-604-6667.

0021-9991/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2003.09.012

acoustics (CAA), large-eddy-simulation (LES) and direct numerical simulation (DNS) of turbulence, and computational electromagnetics (CEM), to name just a few. Preliminary demonstrations have indeed shown that the SV method has many advantages comparing with the current state-of-the-art numerical methods such as the high-order k-exact finite volume (FV) method [2,3], essentially non-oscillatory (ENO) and weighted ENO (WENO) methods [1,10,12,18,19], and the discontinuous Galerkin (DG) method [5–8]. For example, the SV method is more efficient than the k-exact and WENO method (with the same number of unknowns) because the reconstruction problem can be solved analytically with no memory penalty. The SV method has higher resolution for discontinuities than the DG method because of the sub-element resolution in the SV method enables limiters to be designed for the sub-cells (or *control volumes* (CVs)).

Ultimately, the SV method is a Godunov-type FV method [9], which has been under development for several decades, and has become the-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 [21], and the references therein. Similar to the Godunov-type method, the SV method has two key components. One is data reconstruction, and the other is the (approximate) Riemann solver. What distinguishes the SV method from the k-exact FV method is in the data reconstruction. Instead of using a (large) stencil of neighboring cells to perform a high-order polynomial reconstruction, the unstructured grid cell – called a SV – is partitioned into a "structured" set of sub-cells called CVs, and cell-averages on these sub-cells are then the degrees-of-freedom (DOFs). These DOFs are used to perform a high-order polynomial reconstruction inside the SV. All the SVs are partitioned in a geometrically similar manner in a simplex, and thus a single reconstruction is obtained. Next, the DOFs are updated to high-order accuracy using the usual Godunov method. Numerical tests with scalar conservation laws in both 1D and 2D and 1D systems have verified that the SV method is indeed highly accurate, conservative, and geometrically flexible [22]. It is important to point out that the total computational cost of the SV method is proportional to the number of CVs in the computational grid, which corresponds to the total number of DOFs. Therefore when the FV method is compared with the SV method, we should compare their accuracies with the same number of CVs or DOFs. In many 3D aerodynamic computations, tens of millions of CVs are required to achieve a desired order of accuracy in a second-order FV method. It is hoped that with higher order SV schemes the number of CVs can be dramatically reduced to achieve the same accuracy with much reduced cost.

In this paper, we further extend the SV method to multi-dimensional systems. In the following section, we present the basic SV method for the 2D Euler equations. In Section 3, multi-dimensional total variation diminishing (TVD) and total variation bounded (TVB) limiters are discussed in detail. In particular, a limiter which guarantees the positivity of pressure and density is presented. In Section 4, several numerical tests are used to assess the performance of the SV method. In addition, the TVD and TVB limiters are tested with problems with discontinuities. Finally, conclusions and recommendations for further investigations are summarized in Section 5.

2. Spectral volume method for the 2D Euler equations

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

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}=0, \tag{2.1a}$$

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

$$Q=\begin{Bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{Bmatrix},\quad E=\begin{Bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ u(E+p)\end{Bmatrix},\quad F=\begin{Bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ v(E+p)\end{Bmatrix}. \tag{2.1b}$$

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

$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho(u^2 + v^2) \tag{2.1c}$$

with ratio of specific heats γ. In all the simulations in this paper, γ is taken to be 1.4. The Jacobian matrix of the flux vector for a face with unit normal $\boldsymbol{n} = (n_x, n_y)$ can be written as

$$B = n_x \frac{\partial E}{\partial Q} + n_y \frac{\partial F}{\partial Q},$$

where B has 4 real eigenvalues $\lambda_{1,2} = v_n$, $\lambda_3 = v_n + c$, $\lambda_4 = v_n - c$, and a complete set of (right column) eigenvectors $\{r_1, r_2, r_3, r_4\}$, where $v_n = un_x + vn_y$ and c is the speed of sound. Let R be the matrix composed of these right eigenvectors, then the Jacobian matrix B can be diagonalized as

$$R^{-1}BR = \Lambda, \tag{2.2}$$

where Λ is the diagonal matrix containing the eigenvalues, i.e.,

$$\Lambda = \mathrm{diag}(v_n, v_n, v_n + c, v_n - c). \tag{2.3}$$

Assume that we solve (2.1a)–(2.1c) in the computational domain Ω subject to proper initial and boundary conditions. The domain Ω is discretized into N non-overlapping triangular volumes (or cells). These volumes are called SVs in the SV method, i.e.,

$$\Omega = \bigcup_{i=1}^{N} S_i. \tag{2.4}$$

Given a desired order of accuracy k for (2.1a)–(2.1c), each SV S_i is then partitioned into $m = k(k+1)/2$ subcells called CVs, and the jth CV of S_i is then denoted by $C_{i,j}$. Let q denote any of the conservative variables in Q. The cell-averaged conservative variable q at time t for CV $C_{i,j}$ is defined as

$$\bar{q}_{i,j}(t) = \frac{\int_{C_{i,j}} q(x,y,t)\,\mathrm{d}x\,\mathrm{d}y}{V_{i,j}}, \quad j = 1,\ldots,m, \;\; i = 1,\ldots,N, \tag{2.5}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. Given the cell-averaged conservative variables for all the CVs in S_i, a polynomial $p_i(x,y) \in P^{k-1}$ (the space of polynomials of degree at most $k-1$) can be reconstructed such that it is a kth order accurate approximation to the function $q(x,y)$ inside S_i:

$$p_i(x,y) = q(x,y) + \mathrm{O}(h^k), \quad (x,y) \in S_i, \;\; i = 1,\ldots,N. \tag{2.6}$$

This reconstruction can be solved analytically by satisfying the following conditions:

$$\frac{\int_{C_{i,j}} p_i(x,y)\,\mathrm{d}x\,\mathrm{d}y}{V_{i,j}} = \bar{q}_{i,j}, \quad j = 1,\ldots,m. \tag{2.7}$$

This polynomial $p_i(x,y)$ is the kth order approximation we are looking for as long as the function $q(x,y)$ is smooth in the region covered by S_i. The reconstruction can be more conveniently expressed as

$$p_i(x,y) = \sum_{j=1}^{m} L_j(x,y)\bar{q}_{i,j}, \tag{2.8}$$

where $L_j(x,y) \in P^{k-1}$ are the "shape" functions [22] which satisfy

$$\frac{\int_{C_{i,j}} L_l(x,y)\,\mathrm{d}x\,\mathrm{d}y}{V_{i,j}} = \delta_{jl}. \tag{2.9}$$

Once the basis is given, the shape functions can be obtained analytically with any software capable of symbolic manipulation such as Mathematica [25]. The values of the shape functions at the Gauss quadrature points for several partitions used in this paper are given in Appendix A. The high-order reconstruction is then used to generate high-order updates for the cell-averaged state variable on the CVs. Integrating (2.1a)–(2.1c) in $C_{i,j}$, we obtain the following integral equation for the CV-averaged mean

$$\frac{\mathrm{d}\bar{Q}_{i,j}}{\mathrm{d}t} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = 0, \tag{2.10}$$

where $\bar{Q}_{i,j}$ is the vector of the CV-averaged conservative variables in $C_{i,j}$, $f = (E,F)$, K is the number of faces in $C_{i,j}$, and A_r represents the rth face of $C_{i,j}$. The surface integration on each face can be performed with a kth order accurate Gauss quadrature formula, i.e.

$$\int_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = \sum_{q=1}^{J} w_{rq} f(Q(x_{rq},y_{rq})) \cdot \boldsymbol{n}_r A_r + \mathrm{O}(A_r h^k), \tag{2.11}$$

where $J = \text{integer}[(k+1)/2]$ is the number of quadrature points on the rth face, w_{rq} are the Gauss quadrature weights, (x_{rq}, y_{rq}) are the Gauss quadrature points, and h is the maximum edge length of all the CVs. Time t is omitted whenever there is no confusion. If $f = \text{constant}$, the following identity exists:

$$\sum_{r=1}^{K} \int_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = 0. \tag{2.12}$$

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.,

$$\sum_{r=1}^{K} \int_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = \sum_{r=1}^{K} \sum_{q=1}^{J} w_{rq} f(Q(x_{rq},y_{rq})) \cdot \boldsymbol{n}_r A_r + \mathrm{O}(A_r h^{k+1}). \tag{2.13}$$

Since $\mathrm{O}(V_i) = \mathrm{O}(A_r h)$, we therefore have

$$\frac{1}{V_{i,j}} \sum_{r=1}^{K} \oint_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = \frac{1}{V_{i,j}} \sum_{r=1}^{K} \sum_{q=1}^{J} w_{rq} f(Q(x_{rq},y_{rq})) \cdot \boldsymbol{n}_r A_r + \mathrm{O}(h^k). \tag{2.14}$$

With the polynomial distribution on each SV, the state variable is most likely discontinuous across the SV boundaries. Therefore, the flux integration involves two discontinuous state variables just to the left and right of a face on the SV boundary. This flux integration is carried out using one of the Lipschitz continuous approximate Riemann solvers or flux splitting procedures, i.e.,

$$f(Q(x_{rq},y_{rq})) \cdot \boldsymbol{n}_r \approx f_{\mathrm{Riem}}(Q_{\mathrm{L}}(x_{rq},y_{rq}), Q_{\mathrm{R}}(x_{rq},y_{rq}), \boldsymbol{n}_r), \tag{2.15}$$

where Q_{L} and Q_{R} are the vector of conservative variables just to the left and right of a face. Substituting (2.15) into (2.11), we obtain

$$\int_{A_r} (f \cdot \boldsymbol{n})\,\mathrm{d}A = \sum_{q=1}^{J} w_{rq} f_{\mathrm{Riem}}(Q_{\mathrm{L}}(x_{rq},y_{rq}), Q_{\mathrm{R}}(x_{rq},y_{rq}), \boldsymbol{n}_r) A_r + \mathrm{O}(A_r h^k). \tag{2.16}$$

It is the Riemann solver which introduces the "upwinding", and dissipation into the SV method such that the SV method is not only high-order accurate, but also stable. In this paper, we employ and evaluate the following two Riemann solvers:

- Roe's Approximate Riemann Solver [14]

$$f_{\mathrm{ROE}}(Q_{\mathrm{L}}, Q_{\mathrm{R}}, \boldsymbol{n}) = \frac{1}{2}\Big[f(Q_{\mathrm{L}}) + f(Q_{\mathrm{R}}) - |\bar{B}|(Q_{\mathrm{R}} - Q_{\mathrm{L}})\Big], \tag{2.17}$$

where $|\bar{B}|$ is the dissipation matrix computed from

$$|\bar{B}| = R|\bar{\Lambda}|R^{-1}. \tag{2.18}$$

Here $|\bar{\Lambda}|$ is the diagonal matrix composed of the absolute values of the eigenvalues of the Jacobian matrix evaluated at the so-called Roe-averages, i.e.,

$$\bar{B} = B(Q^{A}) \tag{2.19}$$

and R is also evaluated at the Roe averages [14].

- Rusanov Flux [2] [15]

$$f_{\mathrm{RUSANOV}}(Q_{\mathrm{L}}, Q_{\mathrm{R}}, \boldsymbol{n}) = \frac{1}{2}[f(Q_{\mathrm{L}}) + f(Q_{\mathrm{R}}) - \alpha(Q_{\mathrm{R}} - Q_{\mathrm{L}})], \tag{2.20a}$$

where α is the maximum absolute eigenvalue computed from

$$\alpha = |\bar{v}_n| + \bar{c}, \tag{2.20b}$$

where $\bar{v}_n$ and $\bar{c}$ are the averaged normal velocity and speed of sound computed from Q_{L} and Q_{R}. Both the Roe and Rusanov fluxes will be evaluated to assess their performances. Finally we obtain the following semi-discrete SV scheme for $C_{i,j}$

$$\frac{\mathrm{d}\bar{Q}_{i,j}}{\mathrm{d}t} + \frac{1}{V_{i,j}}\sum_{r=1}^{K}\sum_{q=1}^{J} w_{rq} f_{\mathrm{Riem}}(Q_{\mathrm{L}}(x_{rq}, y_{rq}), Q_{\mathrm{R}}(x_{rq}, y_{rq}), \boldsymbol{n}_r) A_r = 0. \tag{2.21}$$

Note that the sub-cell-averages over the CVs are updated independently. That's why the cost of the SV method is proportional to the total number of CVs. For time integration, we use either the second or the third-order TVD Runge–Kutta scheme from [16].

3. Multi-dimensional TVD and TVB limiters

For the non-linear Euler equations, it is necessary to perform data limiting to maintain stability if the solution contains discontinuities. TVD limiters enforce strict monotonicity by sacrificing solution accuracy at local extrema, while TVB limiters relax the monotonicity requirement to achieve uniform accuracy away from discontinuities. In this paper, both limiters are presented in a uniform TVB limiter form, with TVD being a special case. There are two different ways of applying limiters in the system setting. One way is to apply a limiter to each characteristic variable. The other is to apply a limiter to each component of the vector of the conservative variables. The former has the nice property of naturally degenerating to the scalar case if the hyperbolic system is linear, but the latter is much more efficient. In this paper, we choose

[2] This flux has been called the local Lax-Friedrichs flux by many researchers and the current authors in previous publications. Thanks are due to Professor van Leer for pointing out that Rusanov published the flux first in [15].

the component-wise approach because of its efficiency. To this end, we first establish the following numerical monotonicity criterion for each CV

$$\bar{q}_{i,j}^{\min} \leqslant q_{i,j}(x_{rq}, y_{yq}) \leqslant \bar{q}_{i,j}^{\max}, \tag{3.1}$$

where $\bar{q}_{i,j}^{\min}$ and $\bar{q}_{i,j}^{\max}$ are the minimum and maximum cell-averaged solutions among all the neighboring CVs sharing a face (called face neighbors) or a node (called node neighbors) with $C_{i,j}$, and $q_{i,j}(x_{rq}, y_{yq})$ is the reconstructed solution at any of the quadrature points of the CV under consideration. Obviously, the face neighbors are included in the node neighbors as shown in Fig. 1. The face neighbors are used to define $\bar{q}_{i,j}^{\min}$ and $\bar{q}_{i,j}^{\max}$ for the limiter in the second-order SV scheme, while the node neighbors are used to define $\bar{q}_{i,j}^{\min}$ and $\bar{q}_{i,j}^{\max}$ for the limiter in the third- and fourth-order SV schemes. This choice is due to the fact that one quadrature point is used for the second-order SV scheme, but two quadrature points are required for the third- and fourth-order SV schemes. The use of face neighbors in the third- and fourth-order SV schemes causes too many cells to be limited, essentially degrading the schemes to a second-order FV scheme. Note that the definition of $\bar{q}_{i,j}^{\min}$ and $\bar{q}_{i,j}^{\max}$ for the third-order SV scheme is different from that of an earlier publication [24]. Numerical experiments have shown that the new definition produces dramatically better resolution for discontinuities and complex flow features than the previous one. It was proven by Liu [13] in the scalar case that if (3.1) is satisfied by the reconstruction, the numerical solution satisfies a maximum principle, i.e., the solution is monotonic. If (3.1) is strictly enforced, the resultant numerical scheme for the scalar case is TVD. However, it is well known that TVD schemes are locally first-order at extrema, and may pollute the global accuracy of the solution. In order to maintain the order of accuracy away from discontinuities, the TVB idea [17] is followed here, i.e., small oscillations are allowed in the solution. If we express the reconstruction for the quadrature points in the following form

$$\Delta q_{rq} = p_i(x_{rq}, y_{rq}) - \bar{q}_{i,j}, \tag{3.2}$$

then no data limiting is necessary if

$$\left|\Delta q_{rq}\right| \leqslant 4 M_q h_{rq}^2, \tag{3.3}$$

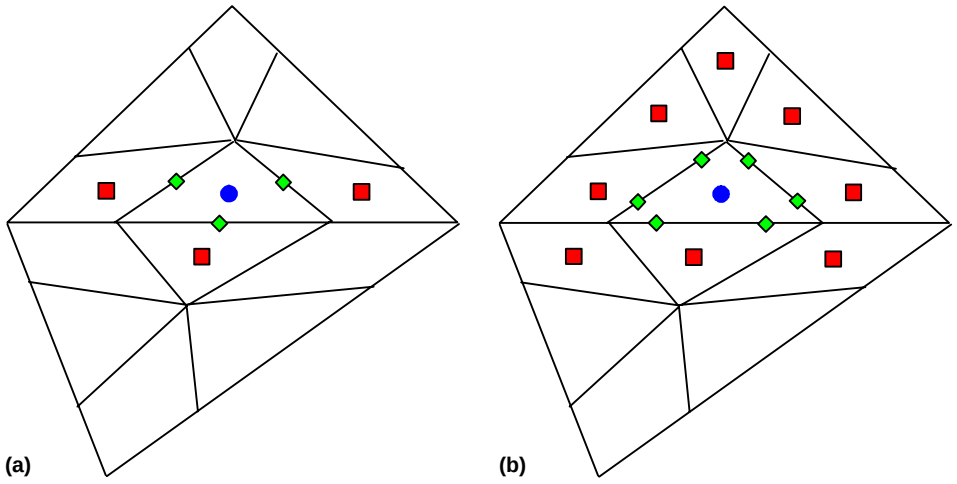

Fig. 1. Face and node neighbors for the CV with a circular dot: (a) face neighbours; (b) node neighbors.

where $h_{rq} = |r_{i,j} - r_{rq}|$ is the distance from the CV centroid to the quadrature point. In other words, no data limiting is necessary if (3.3) is satisfied, even if (3.1) is not. Usually the constant M_q is chosen to be the maximum second derivative of the solution. However M_q is a user chosen parameter if there is a discontinuity in the solution. Note that a different M_q should be used for a different conservative variable because the conservative variables generally have very different scales. In this paper, M_q is scaled according to the minimum and maximum of the component, i.e.,

$$M_q = M(\bar{q}^{\max} - \bar{q}^{\min}), \tag{3.4}$$

where M is a constant independent of the component, and $\bar{q}^{\max}$ and $\bar{q}^{\min}$ are the maximum and minimum of the CV-averaged solution over the entire computational domain. If (3.3) and (3.1) are violated for any quadrature point, then it is assumed that the CV is close to a discontinuity, and the solution in the CV is locally linear, i.e.,

$$q_{i,j}(x,y) = \bar{q}_{i,j} + \nabla q_{i,j} \cdot (r - r_{i,j}), \quad \forall \boldsymbol{r} \in C_{i,j}. \tag{3.5}$$

In addition the solutions are assumed linear for all the CVs inside a SV if any of the CVs is limited. The magnitude of the solution gradient is maximized subject to the monotonicity condition given in (3.3). The initial guess of the gradient $\nabla q_{i,j}$ for each CV is computed with the cell-averaged solution at its face neighbors using a least squares linear reconstruction algorithm [20]. This gradient may not satisfy (3.3). Therefore it is limited by multiplying a scalar $\varphi \in [0,1]$ so that the following solution satisfies (3.3)

$$q_{i,j}(x,y) = \bar{q}_{i,j} + \varphi \nabla q_{i,j} \cdot (\boldsymbol{r} - \boldsymbol{r}_{i,j}). \tag{3.6}$$

The scalar is computed from

$$\varphi = \begin{cases} \min\left(1, \frac{\Delta q_{rq}}{\bar{q}_{i,j}^{\max} - \bar{q}_{i,j}}\right) & \text{if } \Delta q_{rq} > 0, \\ \min\left(1, \frac{\Delta q_{rq}}{\bar{q}_{i,j}^{\min} - \bar{q}_{i,j}}\right) & \text{if } \Delta q_{rq} < 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3.7}$$

In the case of $M = 0$, the limiter becomes TVD. The availability of cell-averaged data on the CVs inside a SV makes this CV-based data limiting possible, whereas in the DG method, one can only do an element based data limiting. Due to the increased local resolution, the SV method was shown to have better resolutions for discontinuities than the DG method in one dimension [23]. This advantage should also carry over to the multi-dimensional systems case.

Note that at the interior CV boundaries inside a SV, the reconstructed conservative variables are continuous if no limiter is imposed. Then the flux is just the analytical flux, which is cheaper to compute than the Riemann flux. When the flux is to be computed, the differences between the left and right state variables are first evaluated. If they are all zero, then the analytical flux is computed. Otherwise, the Riemann flux is calculated.

For problems with strong discontinuities, even if the monotonicity condition (3.1) is strictly enforced for each component of the conserved variables, the positivity of pressure is not guaranteed although the density is always positive. This is because the pressure is computed from the reconstructed conserved variables. To ensure that the pressure is always positive near a discontinuity, the pressure is limited instead of the total energy if it is necessary to limit the reconstruction. In this case, the pressure is assumed linear, and should satisfy the monotonicity condition (3.1). Then the total energy at the quadrature points is computed using the reconstructed density, momentums and pressure. This positivity-guaranteeing limiting has been found to be very robust.

4. Numerical tests

In this section, SV schemes of various orders of accuracy are evaluated for the 2D Euler equations. Based on the study performed for 2D scalar conservation laws [22], one particular partition for a given order of accuracy is selected. The partitions for the second–fourth-order SV schemes are shown in Fig. 2. The reconstruction coefficients for these partitions are presented in Appendix A. Note that these partitions are by no means optimal. Numerical tests have shown that these partitions yield accurate and convergent SV schemes for the 2D scalar conservation laws. They are tested for 2D Euler equations in this paper. The optimization of partitions for various orders of interpolations in a triangle will be studied in a future publication by following ideas presented by Chen and Babuska [4] and by Hesthaven and Teng [11].

4.1. Accuracy study with vortex evolution problem

This is an idealized problem for the Euler equations in 2D used by Shu [18]. 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 is infinitely big, the exact solution of the Euler equations with the above initial conditions is just the passive convection of the isotropic vortex with the 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 simulations were carried out until $t = 2$ on two different grids, one regular and one irregular as shown in Fig. 3. The finer grids are generated recursively by cutting each coarser grid cell into four finer grid cells. No limiters were employed in this study since the problem is smooth, and the Rusanov

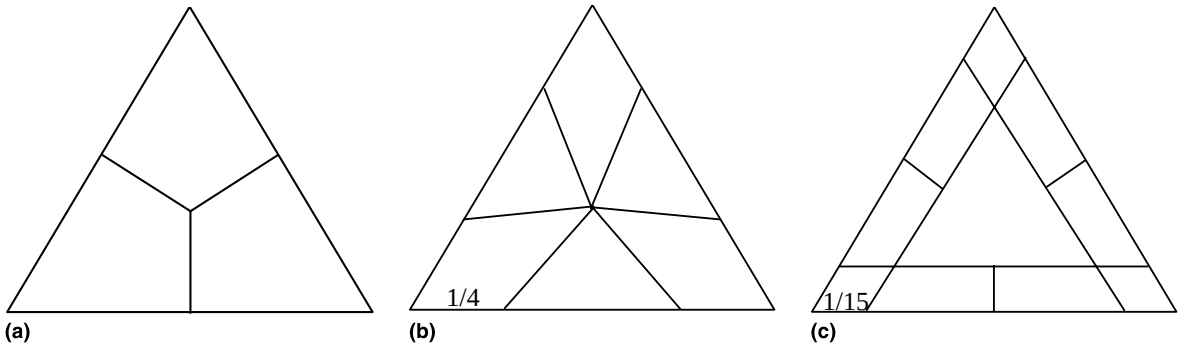

Fig. 2. Partitions of various orders in a triangular SV: (a) linear; (b) quadratic; (c) cubic.

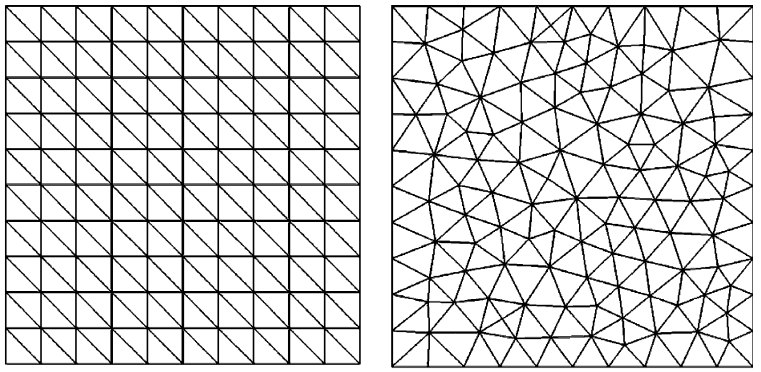

Fig. 3. Regular and irregular grids for the vortex case: (a) regular ($10 \times 10 \times 2$); (b) irregular ($10 \times 10 \times 2$).

Table 1
Accuracy study of SV schemes with propagating vortex problem regular grid

Order of accuracy	Grid	L_∞ error	L_∞ order	L_1 error	L_1 order
2	$10 \times 10 \times 2$	1.1910e − 1	–	7.4323e − 3	–
	$20 \times 20 \times 2$	4.5718e − 2	1.38	1.7502e − 3	2.09
	$40 \times 40 \times 2$	1.0832e − 2	2.07	4.2363e − 4	2.05
	$80 \times 80 \times 2$	2.3293e − 3	2.22	9.9231e − 5	2.09
3	$10 \times 10 \times 2$	5.2871e − 2	–	2.6626e − 3	–
	$20 \times 20 \times 2$	8.2271e − 3	2.68	3.4387e − 4	3.07
	$40 \times 40 \times 2$	1.3659e − 3	2.59	5.3779e − 5	2.49
	$80 \times 80 \times 2$	2.1575e − 4	2.66	8.3061e − 6	2.88
4	$10 \times 10 \times 2$	1.0647e − 2	–	5.4272e − 4	–
	$20 \times 20 \times 2$	1.1254e − 3	3.24	5.0390e − 5	3.43
	$40 \times 40 \times 2$	1.1051e − 4	3.35	3.0832e − 6	4.03
	$80 \times 80 \times 2$	7.2922e − 6	3.92	1.9354e − 7	3.99

Table 2
Accuracy study of SV schemes with propagating vortex problem irregular grid

Order of accuracy	Grid	L_∞ error	L_∞ order	L_1 error	L_1 order
2	$10 \times 10 \times 2$	8.4826e − 2	–	5.4913e − 3	–
	$20 \times 20 \times 2$	2.7206e − 2	1.64	1.3843e − 3	1.99
	$40 \times 40 \times 2$	7.3694e − 3	1.88	3.7917e − 4	1.87
	$80 \times 80 \times 2$	2.2179e − 3	1.73	9.9125e − 5	1.94
3	$10 \times 10 \times 2$	2.7389e − 2	–	1.5259e − 3	–
	$20 \times 20 \times 2$	6.0315e − 3	2.18	2.7834e − 4	2.46
	$40 \times 40 \times 2$	8.5093e − 4	2.83	4.3581e − 5	2.68
	$80 \times 80 \times 2$	1.6079e − 4	2.40	6.7872e − 6	2.68
4	$10 \times 10 \times 2$	7.8981e − 3	–	3.4651e − 4	–
	$20 \times 20 \times 2$	4.5335e − 4	4.12	3.0005e − 5	3.53
	$40 \times 40 \times 2$	4.7422e − 5	3.26	2.1245e − 6	3.82
	$80 \times 80 \times 2$	3.8195e − 6	3.63	1.4503e − 7	3.87

flux was used in all simulations. In Tables 1 and 2, the L_1 and L_∞ norms in the CV-averaged density are presented for the regular and irregular grids respectively, for SV schemes of second–fourth order. The errors presented in the tables were made time step independent by using sufficiently small time steps. Note that all the simulations have reached the desired order of accuracy in the L_1 norm. The L_∞ norms degrade about half an order on the irregular mesh probably because the mesh is not smooth.

In order to compare the SV method with a second-order FV method [20], simulations were also performed on the *sub-cell CV grids* generated by the various SV partitions, as shown in Fig. 4. These grids are named SV2, SV3 and SV4 grids respectively. In other words, the numbers of unknowns in the FV simulations are the same as those of the corresponding SV simulations. The L_1 and L_∞ errors computed using a second-order FV method on the sub-cell grids are presented in Table 3. It is obvious that second-order accuracy was achieved in both the L_1 and L_∞ norms by the FV solver on all the grids. Note that the solution errors on the SV3 grids are only slightly lower than those on the SV2 grids although the SV3 grids have twice as many CVs as the SV2 grids. Furthermore, the errors on the SV4 grids are larger than those on the SV2 grids although the former have more than three times the number of control volumes. This can be

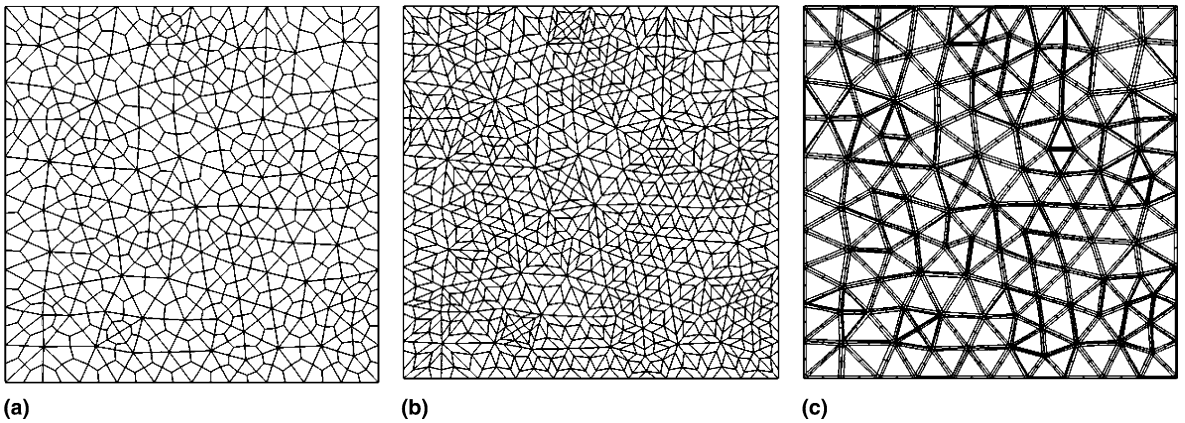

Fig. 4. Control volumes for the second–fourth-order SV schemes: (a) SV2 grid; (b) SV3 grid; (c) SV4 grid.

Table 3
Accuracy study of a second-order FV scheme on the CVs generated using various SV partitions with the propagating vortex problem on the irregular grid

Grid	DOFs	L_∞ error	L_∞ order	L_1 error	L_1 order
SV2	$10\times10\times2\times3$	7.16e−2	–	3.79e−3	–
	$20\times20\times2\times3$	1.80e−2	1.99	8.84e−4	2.10
	$40\times40\times2\times3$	4.57e−3	1.98	1.94e−4	2.19
	$80\times80\times2\times3$	1.15e−3	1.99	4.41e−5	2.14
SV3	$10\times10\times2\times6$	3.51e−2	–	2.05e−3	–
	$20\times20\times2\times6$	1.20e−2	1.55	4.66e−4	2.14
	$40\times40\times2\times6$	2.76e−3	2.12	1.03e−4	2.18
	$80\times80\times2\times6$	1.06e−3	1.38	2.59e−5	1.99
SV4	$10\times10\times2\times10$	1.06e−1	–	3.95e−3	–
	$20\times20\times2\times10$	2.82e−2	1.91	9.54e−4	2.05
	$40\times40\times2\times10$	7.27e−3	1.96	2.14e−4	2.16
	$80\times80\times2\times10$	1.93e−3	1.91	6.31e−5	1.76

attributed to the non-smoothness of the control volumes in the SV4 grid. It is interesting to see that the second-order FV method is more accurate than the second-order SV method with the same number of DOFs. This is because the linear reconstruction is for the macro-cell in the SV method, but for the sub-cell in the FV method. The third- and fourth-order SV schemes are more accurate than the second-order FV scheme with the same DOFs. For example, the third-order SV method is a factor of 3.81 more accurate (in terms of L_1 error) than the second-order FV scheme on the finest mesh, while the fourth-order SV scheme is a factor of 435 more accurate than the second-order FV scheme with the same number of DOFs (on the finest mesh).

It is also interesting to compare the relative cost of the SV and FV schemes. In this paper, the relative cost is defined roughly as the relative CPU times needed to achieve the same solution L_1 error. The following CPU times were measured on a 2.0 GHz Intel Pentium 4 computer running the Linux operating system. One residual computation on the finest mesh with a second-order SV scheme took 106

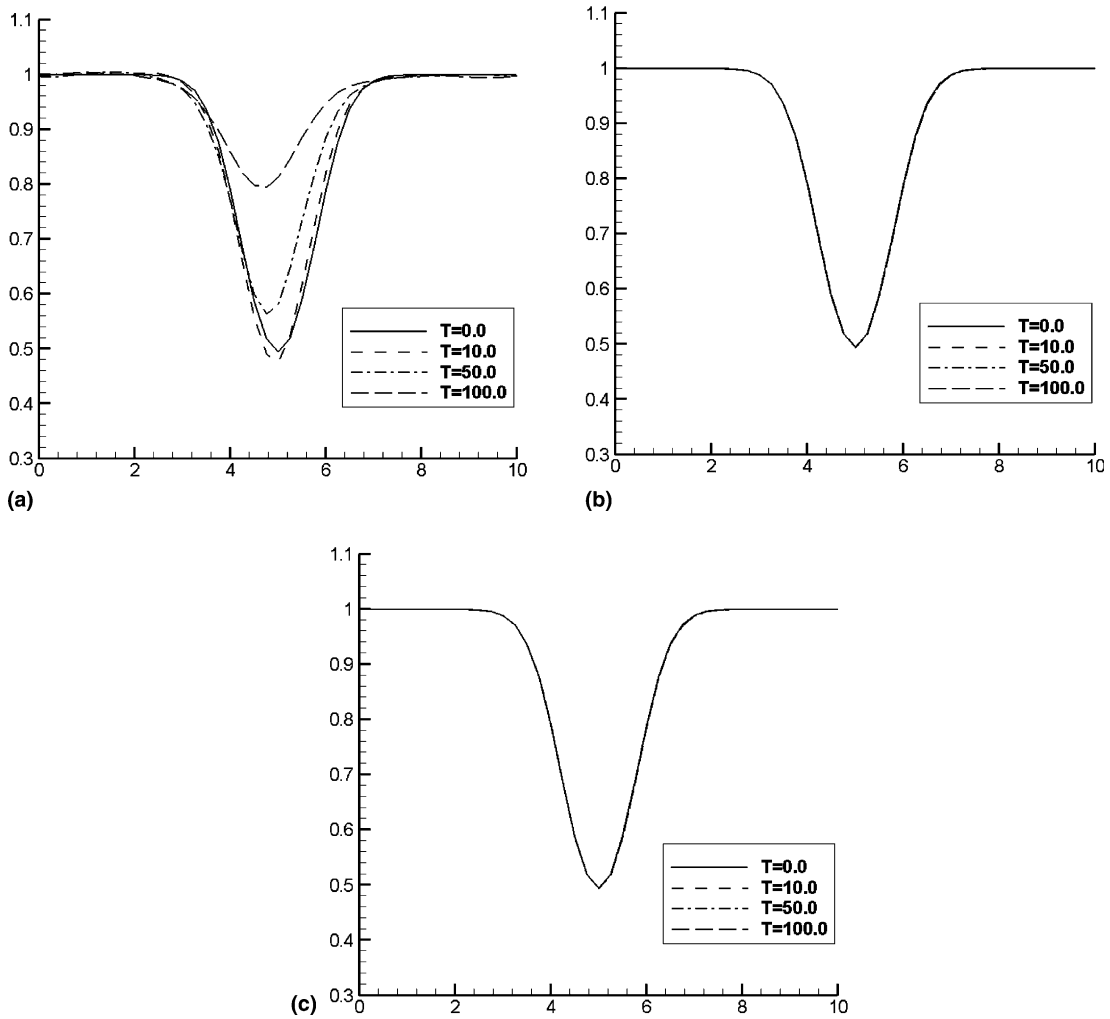

Fig. 5. Density profiles along $x = 5$ at $t = 0$, 10, 50 and 100 for the second-, third- and fourth-order SV schemes: (a) second order; (b) third order (c) fourth order.

microseconds. The second-order FV scheme took 128 microseconds, the third-order SV scheme took 370 microseconds and the fourth-order SV scheme took 880 microseconds. For a second-order FV scheme in 2D, the error should decrease linearly with the increase in the number of cells. The relative cost is estimated assuming that the SV2 grid is chosen for the second-order FV scheme because it gives the minimum error with a fixed number of control volumes. Therefore, the second-order SV scheme has a cost of 1.86 (106/128*9.91e – 5/4.41e – 5) relative to the second-order FV scheme (i.e., takes 1.86 times the CPU time to achieve the same error). The third-order SV scheme has a relative cost of 0.44 (370/128*6.79e – 6/4.41e – 5), and the fourth-order SV has a relative cost of 0.023 (880/128*1.45e – 7/4.41e – 5). Clearly the high-order SV schemes are much more efficient than the second-order FV scheme in achieving the same solution quality.

Next we test the SV schemes for handling the long time evolution of the vortex. For this purpose, periodic boundary conditions were employed at all the boundaries. The simulation was performed on the medium regular grid with $20 \times 20 \times 2$ SVs. Fig. 5 shows the density profiles along $x = 0$ at $t = 0$, $t = 10$, 50 and 100 for the second-, third- and fourth-order SV schemes. Note that the second-order SV scheme displays significant dispersion and dissipation errors, especially for the long term simulations. In contrast, both the third- and fourth-order schemes give excellent results.

4.2. Mach 3 wind tunnel with a step

This problem was studied extensively by Woodward and Colella [26], 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 [26], 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.

In the first test, we focus on comparing the various schemes with different orders of accuracy, and the Riemann solvers. For this purpose, a very coarse mesh with $h = 1/20$ was used to illustrate the differences, and is shown in Fig. 6. The mesh has 1763 triangles. Therefore the number of unknowns is 5289 for the second-order SV scheme, 10,578 for the third-order SV scheme and 17,630 for the fourth-order SV scheme. All the simulations were carried out until $t = 4$. The "baseline" simulation was computed with the Rusanov flux and the TVD limiter, i.e., $M = 0$ in (3.4). The computed density contours using the second–fourth-

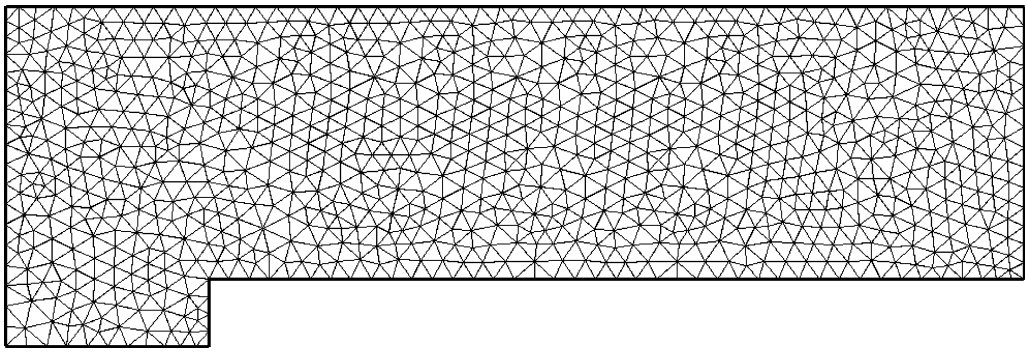

Fig. 6. A coarse mesh for the supersonic wind tunnel problem with size 1/20.

order SV schemes are displayed in Fig. 7. Clearly the third-order SV scheme performed the best for this case since the spurious Mach stem is the weakest. The fourth-order SV scheme performed worse than the third-order SV scheme because many reconstructions are CV-wise linear due to data limiting, and the non-uniform subcell mesh produced large errors. For comparison purposes, the computed density contours using Roe's approximate Riemann solver and TVD limiter are presented in Fig. 8. Obviously, Roe's flux splitting produced weaker Mach stems than the Rusanov flux, which confirms that it has smaller numerical dissipation. In addition, it appears that the third-order SV again produced the best result. Next, the density contours computed using the Rusanov flux and a TVB limiter with $M = 10$ are displayed in Fig. 9.

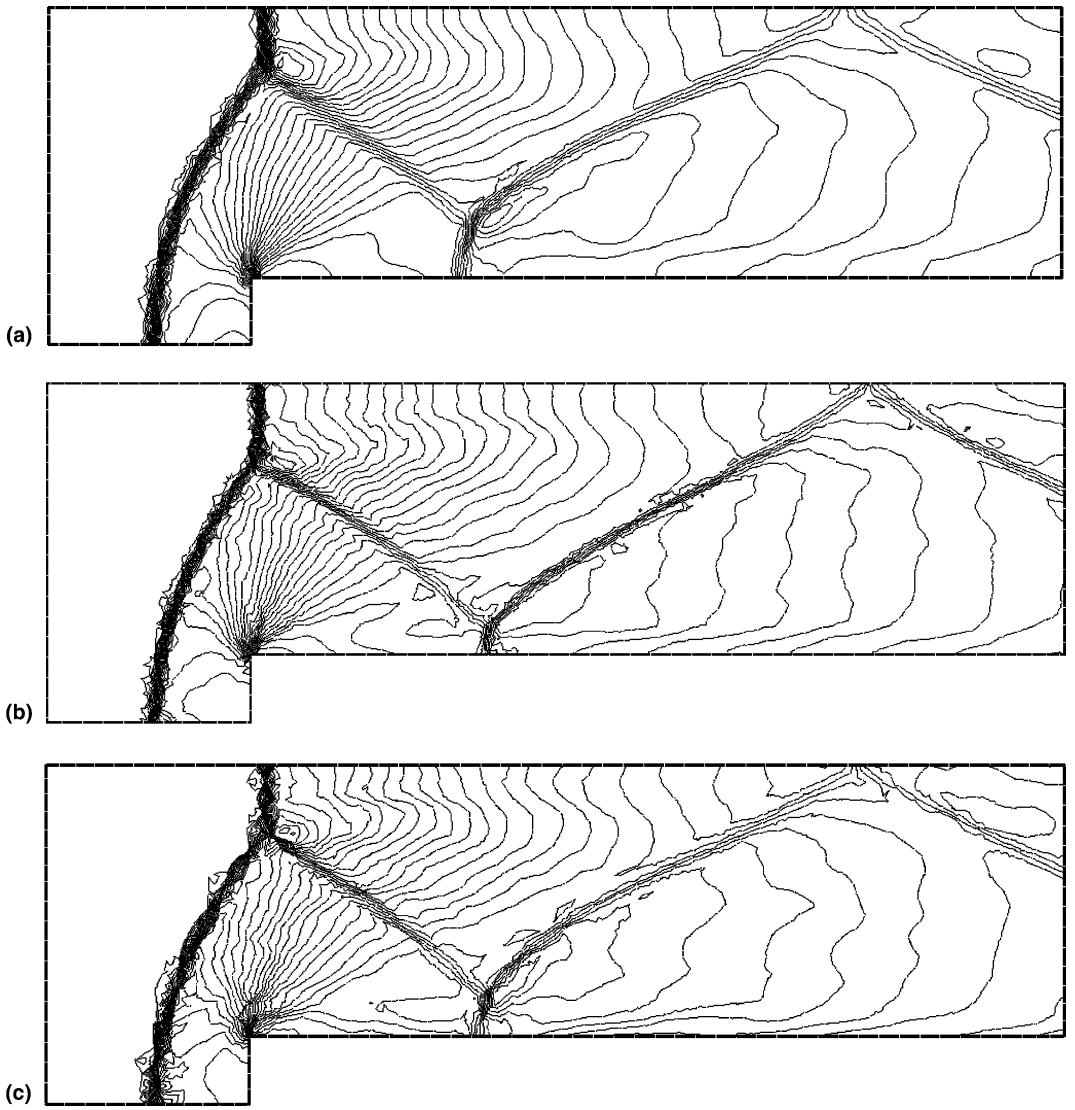

Fig. 7. Density contours computed on the coarse mesh using the Rusanov flux and TVD limiter. Thirty even contour lines between 0.09 and 4.53: (a) second-order SV scheme; (b) third-order SV scheme; (c) fourth-order SV scheme.

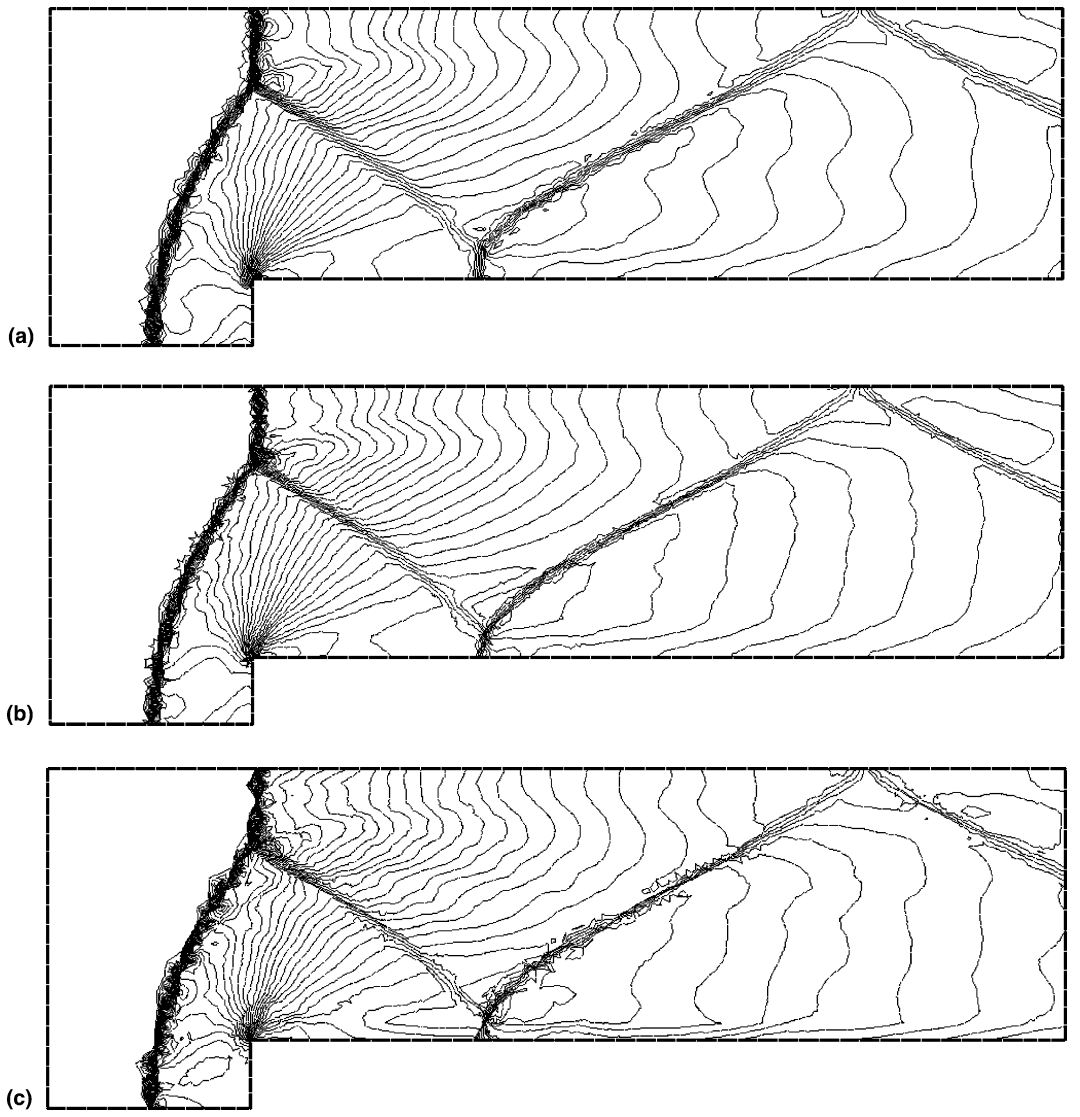

Fig. 8. Density contours computed on the coarse mesh using the Roe flux and TVD limiter. Thirty even contour lines between 0.09 and 4.53: (a) second-order SV scheme; (b) third-order SV scheme; (c) fourth-order SV scheme.

Comparing Figs. 7 and 9, we can see that they are very similar except that the TVB limiter produced slightly more oscillations.

Next we compare the FV and SV methods, again using the coarse mesh shown in Fig. 6. The density contours computed with a second-order FV scheme on the SV2 and SV3 grids are displayed in Fig. 10. Therefore, the same numbers of unknowns are used to obtain the contours in Figs. 7(a) and 10(a), 7(b) and 10(b). Note that the second-order FV scheme produced a weaker Mach stem than the second-order SV scheme, while the third-order SV scheme generated a slightly weaker Mach stem and entropy layers, and sharper shock waves and contact than the second-order FV scheme.

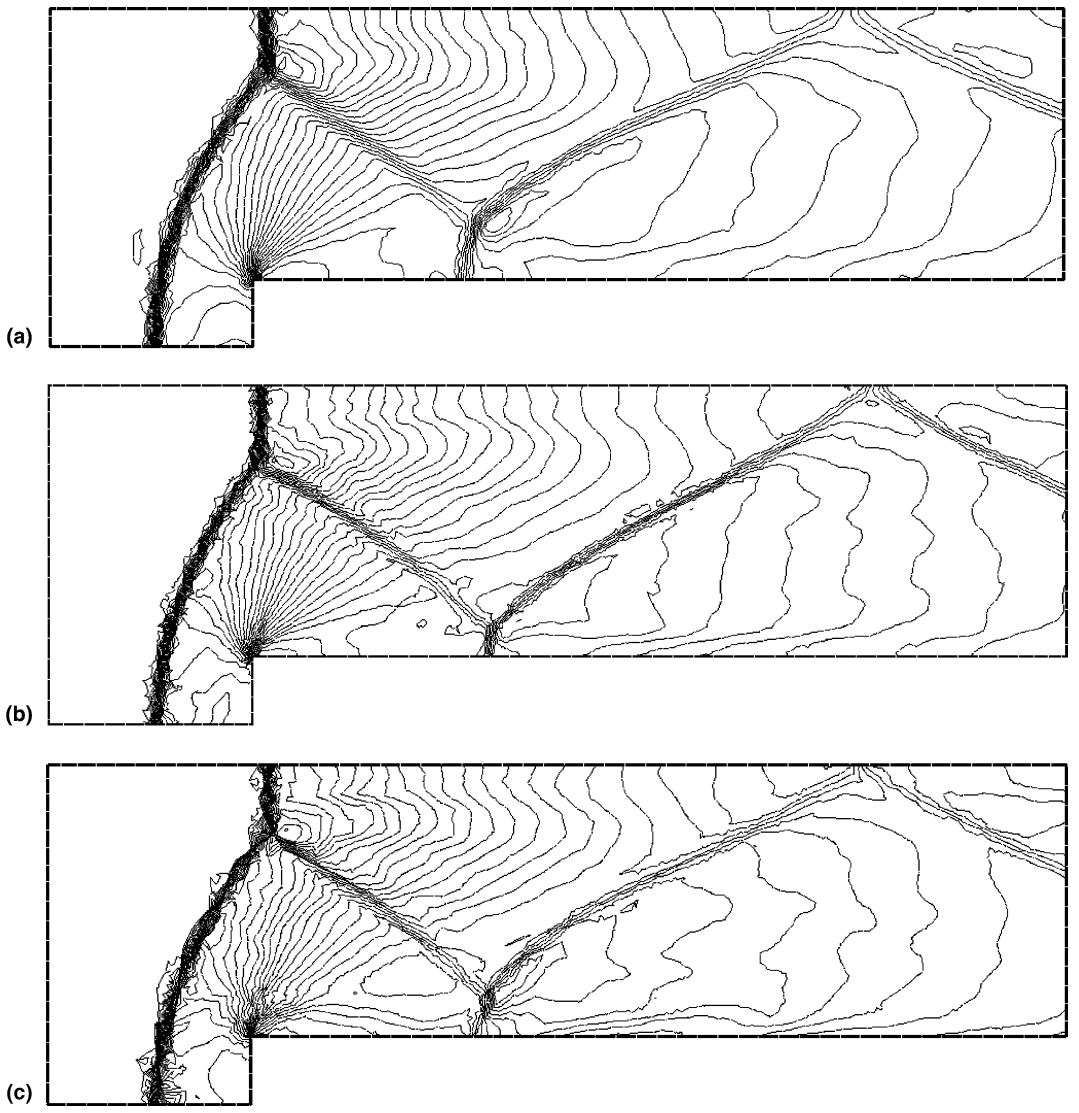

Fig. 9. Density contours computed on the coarse mesh using the Rusanov flux and TVB limiter with $M = 10$. Thirty even contour lines between 0.09 and 4.53: (a) second-order SV scheme; (b) third-order SV scheme; (c) fourth-order SV scheme.

A grid refinement study is then carried out using the "baseline" solver (*Rusanov* flux and $M = 0$). Two finer grids with mesh sizes of $h = 1/40$ (medium), 1/80 (fine) were employed in the simulations. The medium and fine meshes have 8746 and 37,146 triangles respectively. Therefore, the medium mesh has 26,238 and 52,476 unknowns for the second- and third-order SV schemes, and the fine mesh has 111,438 and 222,876 unknowns for the second- and third-order SV schemes, respectively. The computed density contours with the second-order SV scheme are shown in Fig. 11, and the results computed with the third-order SV scheme are presented in Fig. 12. Obviously, the resolution of the third-order results is better than that of the second-order results while the third-order results are more oscillatory than the second-

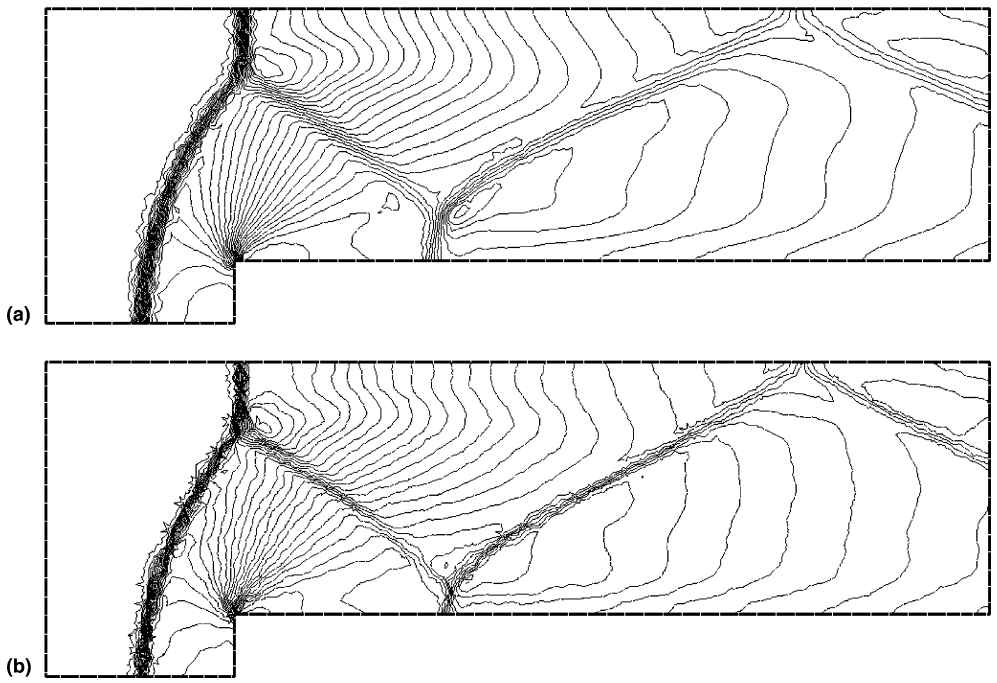

Fig. 10. Density contours computed on the sub-cell grids using a second-order FV scheme with the Rusanov flux and TVD limiter. Thirty even contour lines between 0.09 and 4.53: (a) second-order FV scheme on the SV2 grid; (b) second-order FV scheme on the SV3 grid.

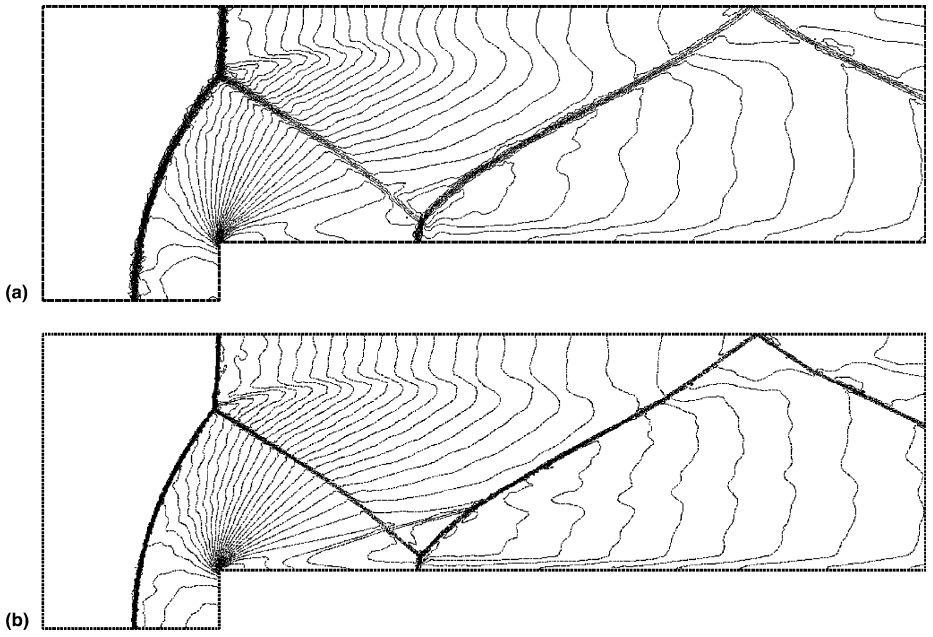

Fig. 11. Density contours computed using a second-order SV scheme with the Rusanov flux and TVD limiter. Thirty even contour lines between 0.09 and 4.53: (a) $h = 1/40$ (26,238 DOFs); (b) $h = 1/80$ (111,438 DOFs).

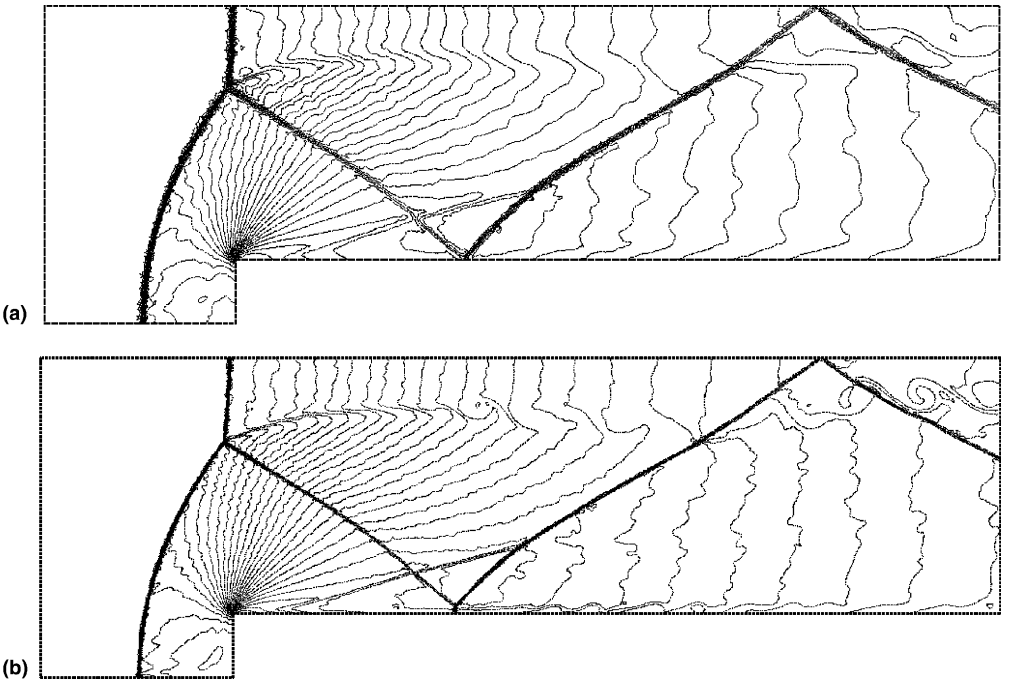

Fig. 12. Density contours computed using a third-order SV scheme with the Rusanov flux and TVD limiter. Thirty even contour lines between 0.09 and 4.53: (a) $h = 1/40$ (52,476 DOFs); (b) $h = 1/80$ (222,876 DOFs).

order results. For example, the spurious Mach stems in the third-order results are visibly weaker than those in the second-order results. The instability generated by the slip line is visible in the third-order results on the fine mesh. It seems the third-order SV scheme produced a much weaker spurious Mach stem and higher resolution for the slip line than published results computed with the third-order DG scheme on comparable grids [8].

Finally to demonstrate that indeed the singular corner point is the cause of the spurious Mach stem, we employed computational grids refined only near the corner point. The first grid is nearly uniform with a mesh size of $h = 1/40$. The other two grids are successively refined by halving the grid size near the corner. The three grids are shown in Fig. 13. The computed density distributions at $t = 4$ with the second order SV scheme with the Rusanov flux and TVD limiter is shown in Fig. 14, and the corresponding third-order results are displayed in Fig. 15. First it is evident that the spurious Mach stem weakened considerably with increased grid resolution near the corner point in both the second- and third-order results. In addition, the entropy layer downstream the shock reflection point is also weakened. Note that the third-order results have higher resolutions for the shock waves and the contact, as expected.

4.3. Double Mach reflection

This problem is also a standard test case for high-resolution schemes [26], and has been studied extensively by many researchers [8,18]. The computational domain for this problem is chosen to be $[0, 4] \times [0, 1]$. The reflecting wall lies at the bottom of the computational domain starting from $x = 1/6$. Initially a right-moving Mach 10 shock is positioned at $x = 1/6$, $y = 0$ and makes a 60° angle with the

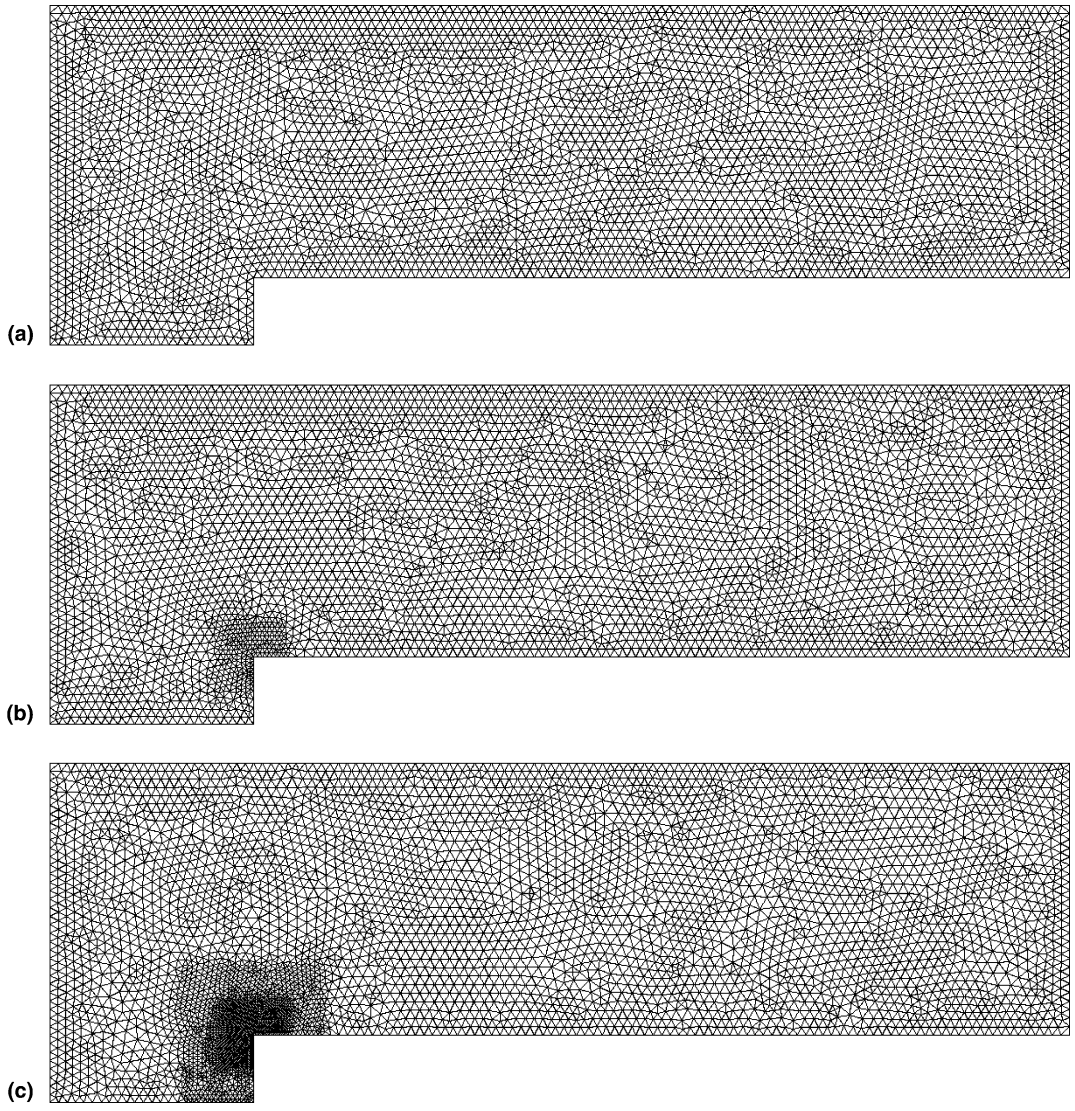

Fig. 13. Three computational grids with different degrees of refinement near the singular corner point.

x-axis. For the bottom boundary, the exact post-shock condition is imposed for the region from $x = 0$ to $x = 1/6$ and a solid wall boundary condition is used for the rest. For the top boundary of the computational domain, the solution is set to describe the exact motion of the Mach 10 shock. The left boundary is set as the exact post-shock condition, while the right boundary is set as outflow boundary. Three unstructured triangular grids were generated with approximate mesh sizes of 1/30, 1/60 and 1/120. These meshes have 8,200, 32,936 and 130,828 triangular cells respectively, corresponding to 24,600, 98,808, and 392,484 unknowns for the second-order SV scheme, and 49,200, 197,616 and 784,968 unknowns for the third-order SV scheme. The coarsest mesh is displayed in Fig. 16. All the simulations were carried until $t = 0.2$ using the Rusanov flux and TVD limiter. Fig. 17 shows the density contours

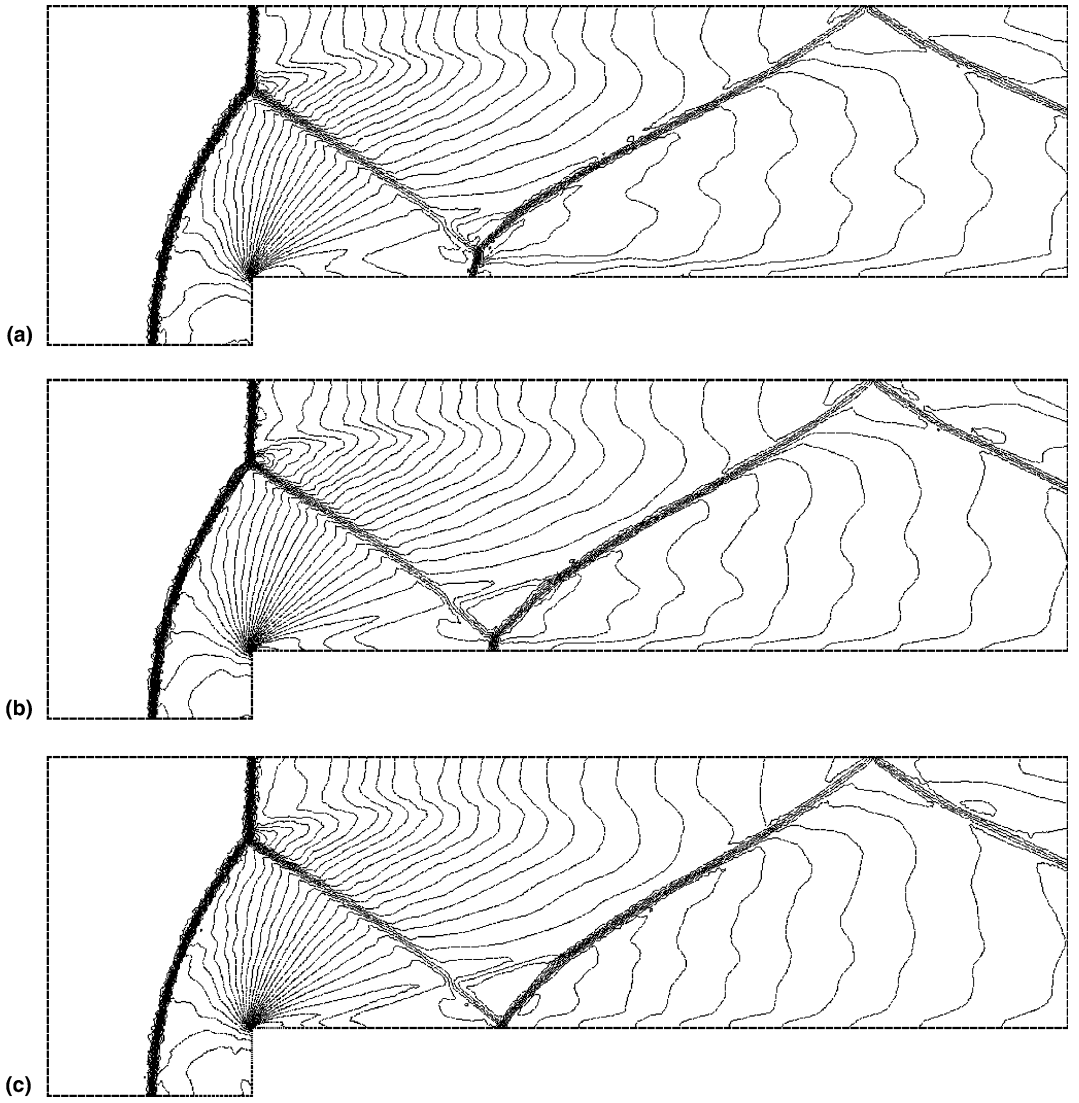

Fig. 14. Computed density contours with the second-order SV scheme using the Rusanov flux and TVD limiter.

computed with the second-order SV scheme on the coarse, medium and fine grids. The corresponding results for the third-order SV scheme are shown in Fig. 18. It is obvious that the third-order SV scheme has much higher resolution than the second-order SV scheme for the complex flow structures near the double Mach stem. In fact, the density contours computed with the third-order scheme on the medium mesh display finer structures than those computed with the second-order scheme on the fine mesh, as shown in Fig. 19. The present results also compare favorably with published results computed with the DG method [8]. It appears that the results computed with the third-order SV scheme on a mesh of size 1/120 have a better resolution for the complex structures near the double Mach stem than the results computed with the third-order DG scheme on a finer mesh of size 1/240 [8].

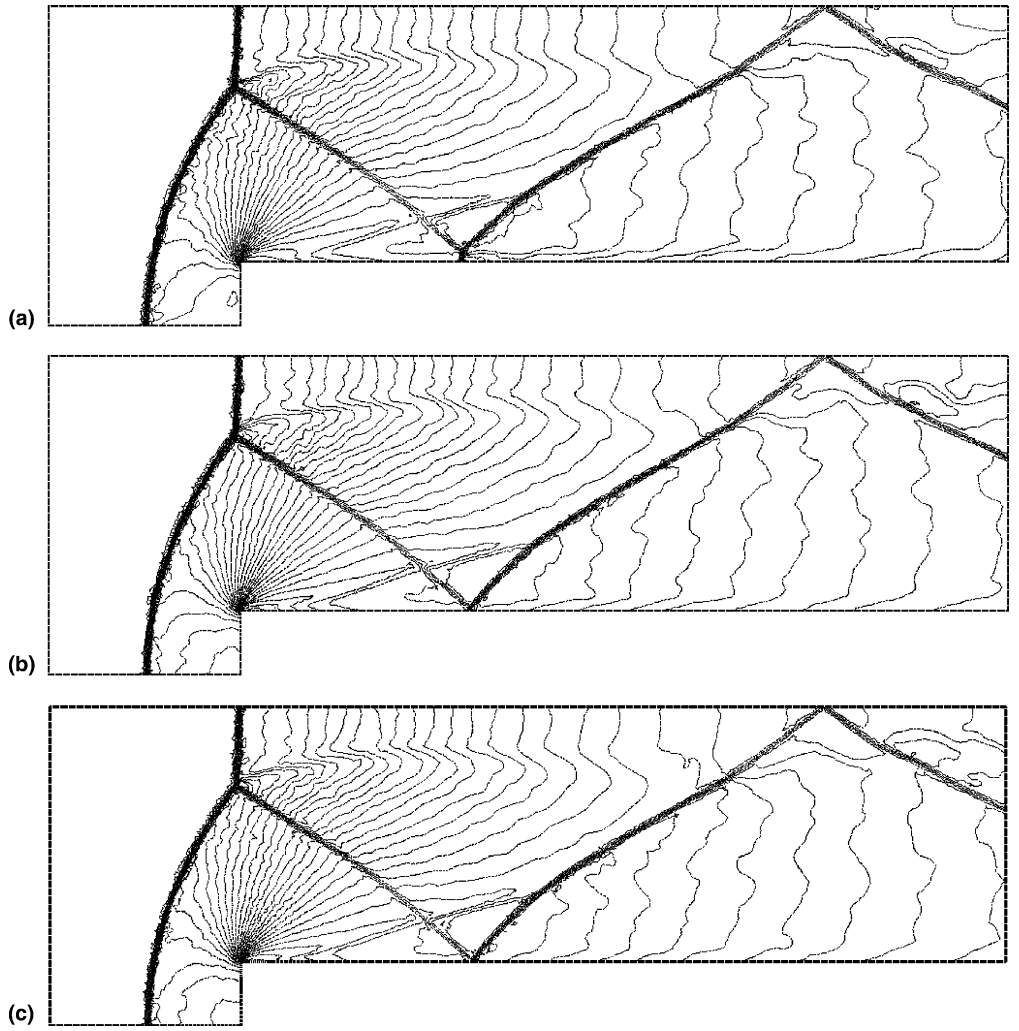

Fig. 15. Computed density contours with the third-order SV scheme using the Rusanov flux and TVD limiter.

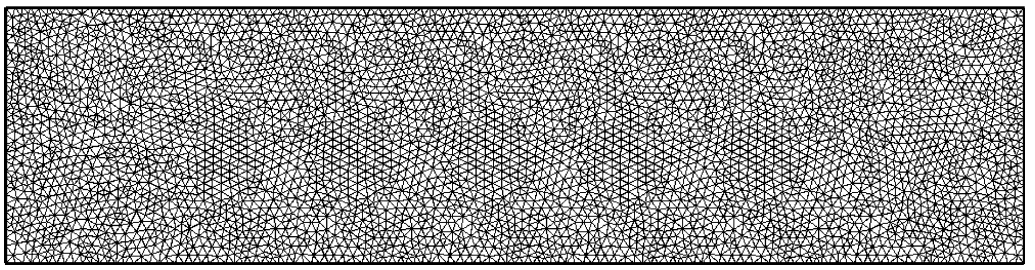

Fig. 16. A coarse mesh for the double Mach reflection problem with size 1/30.

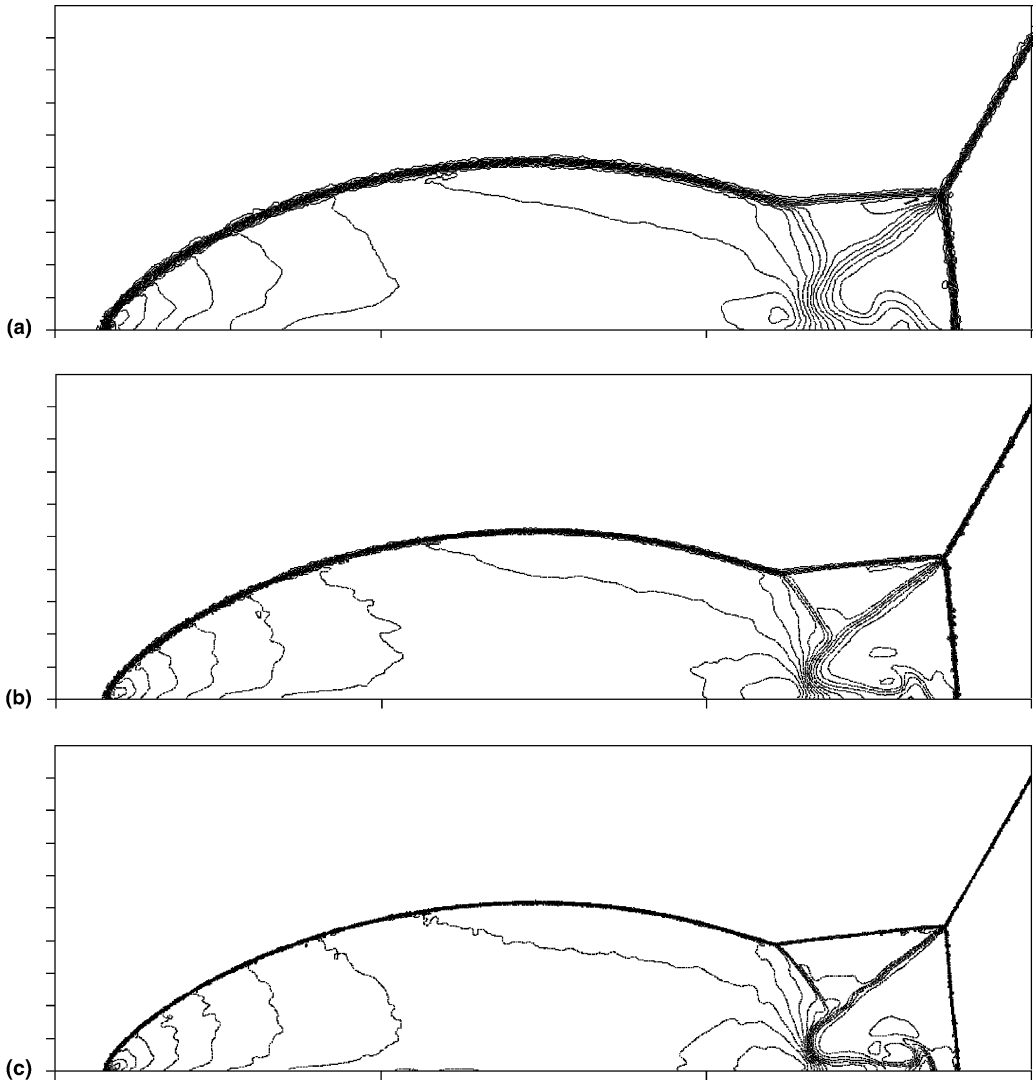

Fig. 17. Density contours computed using a second-order SV scheme with the Rusanov flux and TVD limiter. Thirty even contours between 1.25 and 21.5. (a) $h = 1/30$ (24,600 DOFs) (b) $h = 1/60$ (98,808 DOFs) (c) $h = 1/120$ (392,484 DOFs).

5. Conclusions

The spectral volume method has been successfully extended to two-dimensional hyperbolic systems of conservation laws. The extension to the system has been performed in a component-wise manner. Therefore, the mean conservative variables at the CVs (sub-cells) are used in the polynomial reconstruction. Both TVD and TVB limiters are also implemented in a component-wise manner. In addition, a limiter guaranteeing the positivity of pressure and density has been developed, and shown to be very robust. The limiter for the third-order SV method has been improved to achieve much higher resolution for smooth flow features. An accuracy study has been carried out, and the expected numerical orders of accuracy of the SV schemes have been achieved in both the L_1 and L_∞ norms. The SV method has also

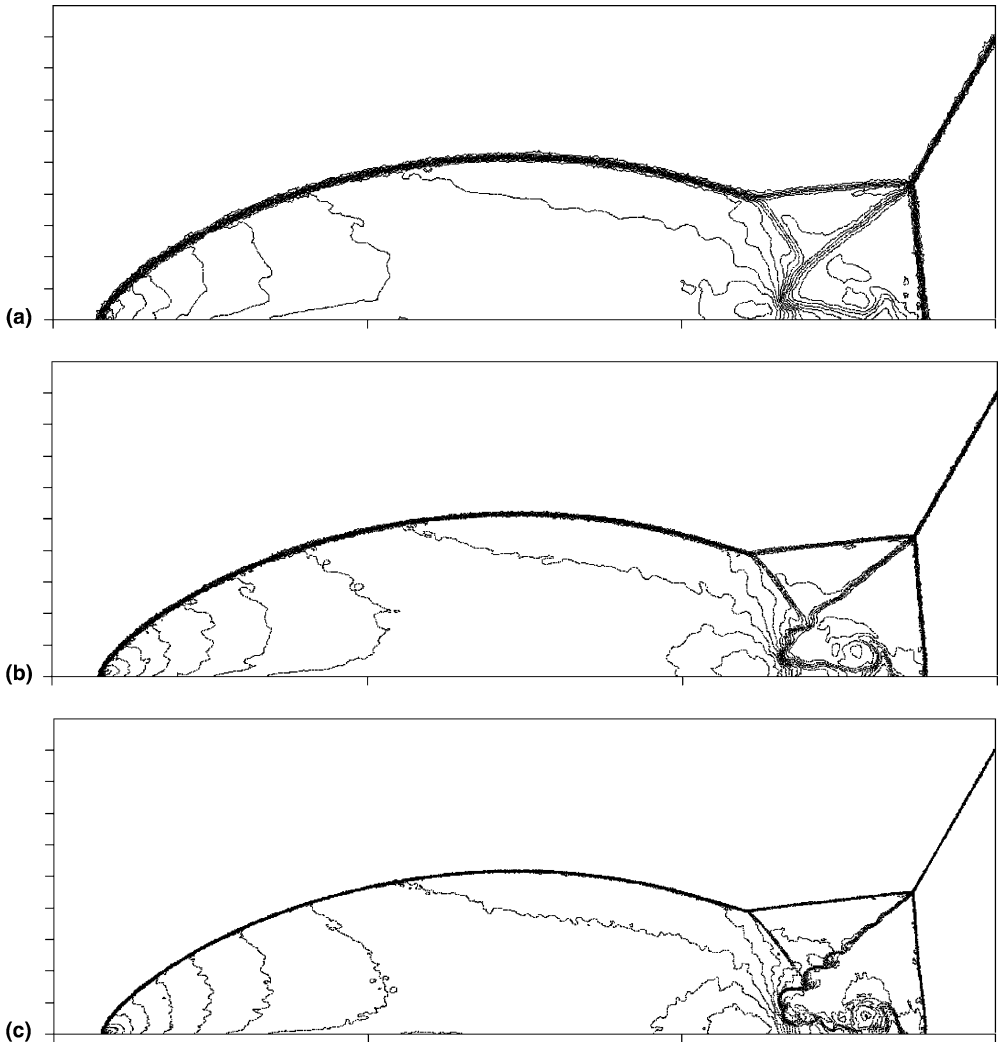

Fig. 18. Density contours computed using a third-order SV scheme with the Rusanov flux and TVD limiter. Thirty even contours between 1.25 and 21.5: (a) $h = 1/30$ (49,200 DOFs); (b) $h = 1/60$ (197,616 DOFs); (c) $h = 1/120$ (784,968 DOFs).

been compared with a second-order FV method, and the advantage of high-order SV schemes has been demonstrated. The extension to 2D Navier–Stokes equations is now under way, and will be reported in a future publication.

Acknowledgements

The first author grateful acknowledges an Intramural Research Grant Project from Michigan State University, and a start-up fund provided by the Department of Mechanical Engineering, College of Engineering of Michigan State University.

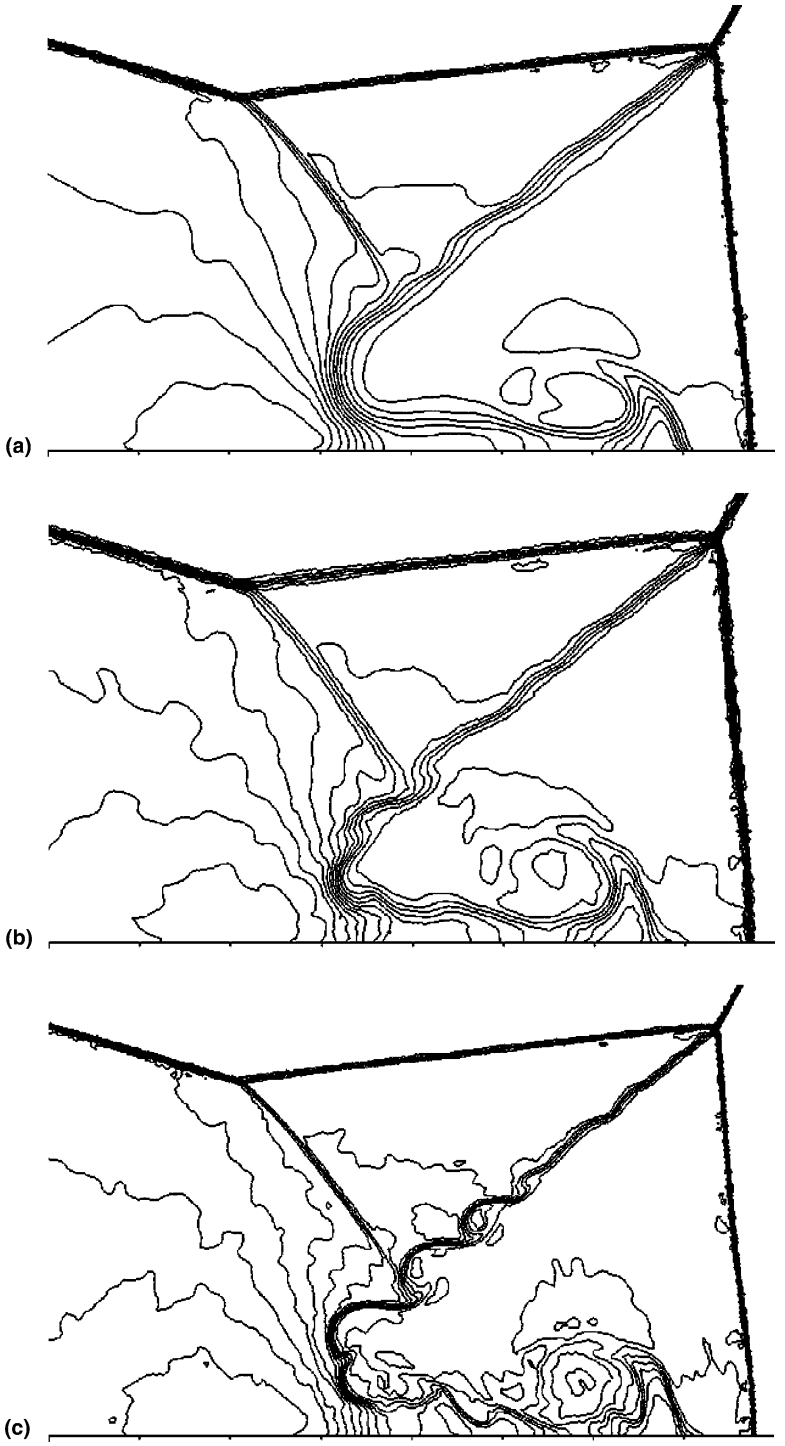

Fig. 19. Close-up view of the density contours near the double Mach stem: (a) second-order SV scheme, $h = 1/120$ (392,384 DOFs); (b) third-order SV scheme, $h = 1/60$ (197,616 DOFs); (c) third-order SV scheme, $h = 1/120$ (784,968 DOFs).

Appendix A. Reconstruction coefficients

A.1. Linear partition

The CVs (sub-cells) inside a spectral volume (the macro-cell) are numbered based on the local node numbering as shown in Fig. 20. Due to the symmetry in a triangle, the reconstruction coefficients at two quadrature points (a and b) are required. The coefficients for point a and b are

$$C_a = \{4/3, 2/15, -7/15\},$$

$$C_b = \{8/15, 8/15, -1/15\}.$$

The solution at any quadrature point a can be computed with

$$q_a = C_a \begin{Bmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{Bmatrix}.$$

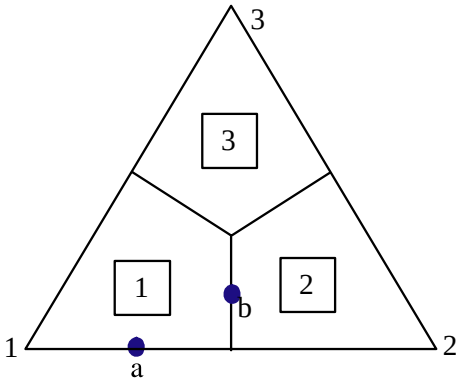

Fig. 20.

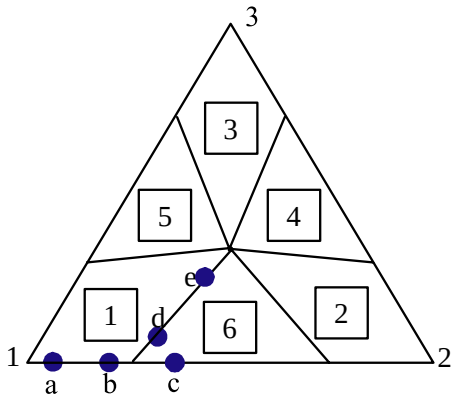

Fig. 21.

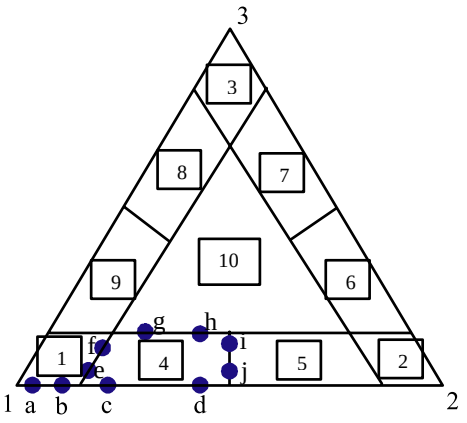

Fig. 22.

A.2. Quadratic partition

The CVs are numbered as shown in Fig. 21. The reconstruction coefficients at five quadrature points are required. The coefficients for these points are

$$C_a = \left\{ \frac{1603+253\sqrt{3}}{738}, \frac{127+89\sqrt{3}}{738}, \frac{361+27\sqrt{3}}{738}, \frac{-89-2\sqrt{3}}{369}, \frac{-458-43\sqrt{3}}{369}, \frac{-259-279\sqrt{3}}{738} \right\},$$

$$C_b = \left\{ \frac{1603-253\sqrt{3}}{738}, \frac{127-89\sqrt{3}}{738}, \frac{361-27\sqrt{3}}{738}, \frac{-89+2\sqrt{3}}{369}, \frac{-458+43\sqrt{3}}{369}, \frac{-259+279\sqrt{3}}{738} \right\},$$

$$C_c = \left\{ \frac{181}{738}+\frac{2}{3\sqrt{3}}, \frac{181}{738}-\frac{2}{3\sqrt{3}}, \frac{253}{738}, \frac{-367+82\sqrt{3}}{738}, \frac{-367-82\sqrt{3}}{738}, \frac{857}{738} \right\},$$

$$C_d = \left\{ \frac{7(83+79\sqrt{3})}{2214}, \frac{-511+61\sqrt{3}}{2214}, \frac{-193+247\sqrt{3}}{2214}, \frac{203-158\sqrt{3}}{1107}, \frac{146-281\sqrt{3}}{1107}, \frac{1639+17\sqrt{3}}{2214} \right\},$$

$$C_e = \left\{ \frac{-7(-83+79\sqrt{3})}{2214}, \frac{-511-61\sqrt{3}}{2214}, \frac{-193-247\sqrt{3}}{2214}, \frac{203+158\sqrt{3}}{1107}, \frac{146+281\sqrt{3}}{1107}, \frac{1639-17\sqrt{3}}{2214} \right\}.$$

A.3. Cubic partition

The CVs are numbered as shown in Fig. 22. The reconstruction coefficients at ten quadrature points are required. Since these coefficients are not easy to express, they are not listed here. Interested readers can send the first author an e-mail to request a text file containing these coefficients.

References

[1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
[2] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper No. 89-0366.

[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] Q. Chen, I. Babuska, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle, Comput. Meth. Appl. Mech. Eng. 128 (1995) 405–417.
[5] 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.
[6] 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.
[7] 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.
[8] B. Cockburn, C.-W. Shu, The Runge–Kutta discontinuous Garlerkin method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[9] 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.
[10] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.
[11] J.S. Hesthaven, C.H. Teng, Stable spectral methods on tetrahedral elements, SIAM J. Sci. Comput. 21 (2000) 2352–2380.
[12] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[13] X.D. Liu, A maximum principle satisfying modification of triangle based adaptive stencil for the solution of scalar hyperbolic conservation laws, SIAM J. Numer. Anal. 30 (1993) 701–716.
[14] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[15] V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.
[16] C.-W. Shu, Total-variation-diminishing time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.
[17] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comp. 49 (1987) 105–121.
[18] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations (Editor: A. Quarteroni), Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, 1998, pp. 325–432.
[19] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II, J. Comput. Phys. 83 (1989) 32.
[20] Z.J. Wang, A fast nested multi-grid viscous flow solver for adaptive Cartesian/quad grids, Int. J. Numer. Meth. Fluids 33 (2000) 657–680.
[21] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.
[22] Z.J. Wang, Yen Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665–697.
[23] Z.J. Wang and Yen Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization, J. Sci. Comput. 20 (2004) 137–157.
[24] Z.J. Wang, Yen Liu, L. Zhang, High-order spectral volume method for 2D Euler equations, in: Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, Orlando, FL, June 2003.
[25] S. Wolfram, The Mathematica Book, fourth ed., Wolfram Media and Cambridge University Press, New York, 1999.
[26] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.