AIAA JOURNAL
Vol. 48, No. 10, October 2010

Implicit High-Order Spectral Finite Volume Method for Inviscid Compressible Flows

Carlos Breviglieri*
Instituto Tecnológico de Aeronáutica, 12228-900 São José dos Campos, São Paulo, Brazil
João Luiz F. Azevedo† and Edson Basso‡
Instituto de Aeronáutica e Espaço, 12228-903 São José dos Campos, São Paulo, Brazil
and
Maximiliano A. F. Souza*
Instituto Tecnológico de Aeronáutica, 12228-900 São José dos Campos, São Paulo, Brazil

DOI: 10.2514/1.J050395

The present work has the objective of demonstrating the capabilities of a spectral finite volume scheme implemented in a cell-centered finite volume context for unstructured meshes. The two-dimensional Euler equations are considered to represent the flows of interest. The spatial discretization scheme is developed to achieve high resolution for flow problems governed by hyperbolic conservation laws. Roe's flux difference splitting method is used as the numerical approximate Riemann solver. Several applications are performed in order to assess the method capability compared to data available in the literature and also compared to an weighted essentially nonoscillatory scheme. There is good agreement with the comparison data, and efficiency improvements over the weighted essentially nonoscillatory method are observed. The features of the present methodology include an implicit time-marching algorithm; second-, third-, and fourth-order spatial resolution; exact high-order domain boundary representation; and a hierarchical moment limiter to treat flow solution discontinuities.

Nomenclature

a	=	speed of sound
C	=	convective operator
E, F	=	flux vectors in the (x, y) Cartesian directions
e_t	=	total energy per unit of volume
$\mathbf{f}$	=	analytical flux vector
f	=	numerical flux vector
$\mathbf{n}$	=	unit normal vector to the face, positive outward
nf	=	number of faces/edges
Q	=	vector of spectral volume conserved properties
q	=	vector of control volume conserved properties
u, v	=	velocity components in the (x, y) Cartesian directions
ρ	=	density

Subscripts

i	=	ith spectral volume
j	=	jth control volume
nb	=	nbth neighbor
r	=	rth edge of the mesh

Superscript

n	=	nth iteration

Presented as Paper 2009-4119 at the 27th AIAA Applied Aerodynamics Conference, San Antonio, TX, 22–25 June 2009; received 19 December 2009; revision received 11 May 2010; accepted for publication 12 May 2010.

*Graduate Student, Department of Computer and Electronic Engineering, Departamento de Ciência e Tecnologia Aeroespacial, DCTA/ITA.

†Senior Research Engineer, Departamento de Ciência e Tecnologia Aeroespacial, DCTA/IAE/ALA; azevedo@iae.cta.br. Associate Fellow AIAA

‡Senior Research Engineer, Departamento de Ciência e Tecnologia Aeroespacial, DCTA/IAE/ALA; basso@iae.cta.br.

I. Introduction

OVER the past several years, the Computational Aerodynamics Laboratory of Instituto de Aeronáutica e Espaço has been developing computational fluid dynamics (CFD) solvers for two and three-dimensional systems [1,2]. One research area of the development effort is aimed at the implementation of high-order methods suitable for steady and unsteady, subsonic, transonic and supersonic flow problems, i.e., external high-speed aerodynamics. Some flux vector splitting schemes such as the van Leer [3] and Liou AUSM+ schemes [4] are implemented along with Roe's flux difference splitting scheme [5] and tested for second-order accuracy with a MUSCL reconstruction [6]. However, the nominally second-order schemes presented results with an order of accuracy smaller than the expected in the solutions for unstructured grids [7]. Aside from this fact, it is well known that total-variation-diminishing (TVD) schemes have their order of accuracy reduced to first order in the presence of discontinuities, due to the effect of limiters.

This observation has motivated the group to study and to implement essentially nonoscillatory (ENO) and weighted essentially nonoscillatory (WENO) schemes in the past [8]. However, as the intrinsic reconstruction model of these schemes relies on gathering neighboring cells for polynomial reconstructions for each cell at each time step, both were found to be very demanding on computational resources for resolution orders greater than 3 in 2-D or anything greater than second order in 3-D. This fact motivated the consideration of the spectral finite volume (SFV) method, as proposed in [9–14], as a more efficient alternative. Such method is expected to perform better than ENO and WENO schemes, compared to the overall cost of the simulation, since it differs on the reconstruction model applied and it is extended up to fourth-order accuracy in the present work. Although the expect order of accuracy is obtained for the second-, third-, and fourth-order SFV methods, a significant deterioration in convergence rate to steady state for the fourth-order simulations is observed, especially in the presence of shock waves. Hence, it is expected that such behavior can be overcome by the use of an implicit time-marching algorithm suitable for high-order methods [15]. Furthermore, in order to keep the high-order method competitive with the lower-order methods, a high-order representation of the geometry boundaries is necessary to reduce the total number of mesh elements. Actually, such high-order boundary

representation is also important for stability and convergence aspects and it is discussed in the present work.

Another important aspect of a compressible flow solver is the limiter technique employed. The use of limiters is necessary when the flow solution contains discontinuities, in order to remove spurious oscillations that may eventually lead to divergence of the numerical solution. Previous work on limiter implementations for high-order methods [16] was based on problem-dependent parameters to determine which elements need limiting and which can limit too many or too few elements of the solution. In the first case, the order of the method is seriously reduced, and in the second case, divergence can occur. To circumvent this drawback, the present work uses a parameter-free generalized moment limiter [17] along with an ENO based reconstruction, to deal with discontinuities. The new limiter does not require input constants from the user, rendering the code more robust. The original formulation for the limiter was developed for the spectral difference method in [17]. The contribution of the present work was in the extension of the limiter formulation for the SFV method, which represents original work by the authors. The numerical solver is implemented for the solution of the 2-D Euler equations in a cell-centered finite volume context for triangular meshes, with options for explicit and implicit schemes for time integration.

The remainder of the paper is organized as follows. In Sec. II, the theoretical formulation is detailed, including spatial and time integration methods. In Sec. III, the general SFV method reconstruction process is presented. Details are given for linear, quadratic and cubic polynomial reconstructions for triangular mesh elements. Also this Section presents the details of the high-order boundary and limiter formulation. Moreover, a new technique employing exact geometry representation is proposed, which is another contribution of the work. In Sec. IV, numerical results are presented. Conclusions and some remarks are given in Sec. V.

II. Theoretical Formulation

A. Governing Equations

In the present work, the 2-D Euler equations are solved in integral form as

$$\frac{\partial}{\partial t}\int_V Q\,\mathrm{d}V + \int_V (\nabla\cdot\mathbf{P})\,\mathrm{d}V = 0 \tag{1}$$

where $\mathbf{P} = E\hat{i} + F\hat{j}$. The application of the divergence theorem to Eq. (1) yields

$$\frac{\partial}{\partial t}\int_V Q\,\mathrm{d}V + \int_S (\mathbf{P}\cdot\mathbf{n})\,\mathrm{d}S = 0 \tag{2}$$

The vector of conserved variables Q and the convective flux vectors E and F are given by

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e_t \end{Bmatrix}, \qquad E = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e_t + p)u \end{Bmatrix}, \qquad F = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e_t + p)v \end{Bmatrix} \tag{3}$$

The standard CFD nomenclature is being used here. Hence, ρ is the density; u and v are the Cartesian velocity components in the x and y directions, respectively; p is the pressure; and e_t is the total energy per unit volume. The system is closed by the equation of state for a perfect gas:

$$p = (\gamma - 1)[e_t - \tfrac{1}{2}\rho(u^2 + v^2)] \tag{4}$$

where e_t is the total energy per unit volume, and the ratio of specific heats γ is set as 1.4 for all computations in this work. In the finite volume context, for stationary meshes, Eq. (2) can be rewritten for the ith mesh element as

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{V_i}\int_S (\mathbf{P}\cdot\mathbf{n})\,\mathrm{d}S \tag{5}$$

where Q_i is the cell-averaged value of Q at time t and V_i is the volume, or area in 2-D, of the ith mesh element.

B. Spatial Discretization

The spatial discretization process determines a kth-order discrete approximation to the integral in the right-hand side of Eq. (5). To solve it numerically, the computational domain Ω, with proper initial and boundary conditions, is discretized into N nonoverlapping triangles, the spectral volumes (SVs), such that

$$\Omega = \bigcup_{i=1}^{N} SV_i \tag{6}$$

One should observe that the spectral volumes could be composed by any type of polygon, given that it is possible to decompose its bounding edges into a finite number of line segments Γ_K, such that

$$SV_i = \bigcup \Gamma_K \tag{7}$$

In the present paper, however, the authors assume that the computational mesh is always composed of triangular elements. Hence, although the theoretical formulation is presented for the general case, the actual SV partition schemes are only implemented for triangular grids.

For a given order of spatial accuracy using the SFV method, each SV_i element must be partitioned in

$$N_m = \frac{k(k+1)}{2} \tag{8}$$

subelements or control volumes (CVs). If one denotes $CV_{i,j}$ as the jth control volume of SV_i, the cell-averaged conserved variables q at time t for $CV_{i,j}$ are computed as

$$q_{i,j} = \frac{1}{V_{i,j}}\int_{CV_{i,j}} q(x,y)\,\mathrm{d}V \tag{9}$$

where $V_{i,j}$ is the volume of $CV_{i,j}$. Once the CV cell-averaged conserved variables are available for all CVs within SV_i, a polynomial, $p_i(x,y) \in P^{k-1}$, with degree $k-1$, can be reconstructed to approximate each component of the CV conserved variable vector q inside SV_i, i.e.,

$$p_i(x,y) = q(x,y) + O(h^{k-1}), \qquad (x,y) \in SV_i \tag{10}$$

where h represents the maximum edge length of all CVs within SV_i. The polynomial reconstruction process is discussed in detail in the following section. For now, it is sufficient to say that this high-order reconstruction is used to update the cell-averaged state variables at the next time step for all the CVs within the computational domain. Note that this polynomial approximation is valid within SV_i and the use of numerical fluxes are necessary across SV boundaries.

Integrating Eq. (5) in $CV_{i,j}$, one can obtain the integral form for the CV mean state variable

$$\frac{\mathrm{d}q_{i,j}}{\mathrm{d}t} + \frac{1}{V_{i,j}}\sum_{r=1}^{nf}\int_{A_r} (\mathbf{f}\cdot\mathbf{n})\,\mathrm{d}S = 0 \tag{11}$$

where $\mathbf{f} = E\hat{i} + F\hat{j}$, at the CV level, and nf is the number of edges of $CV_{i,j}$.

The boundary integral in Eq. (11) can be further discretized into the convective operator form:

$$C(q_{i,j}) \equiv \int_S (\mathbf{f}\cdot\mathbf{n})\,\mathrm{d}S = \sum_{r=1}^{nf}\int_{A_r} (\mathbf{f}\cdot\mathbf{n})\,\mathrm{d}S \tag{12}$$

where nf is the number of faces, or edges in 2-D, of $CV_{i,j}$, and A_r represents the area, or the length in 2-D, of the rth edge of the CV.

Given the fact that $\mathbf{n}$ is constant for each line segment, the integration on the right side of Eq. (12) can be performed numerically with a kth-order-accurate Gaussian quadrature formula:

$$\int_S (\mathbf{f}\cdot\mathbf{n})\,\mathrm{d}S = \sum_{r=1}^{nf}\sum_{q=1}^{nq} w_{rq}\mathbf{f}(q(x_{rq},y_{rq}))\cdot\mathbf{n}_r A_r + O(A_r h^k) \quad (13)$$

where (x_{rq}, y_{rq}) and w_{rq} are, respectively, the Gaussian quadrature point coordinates and the weights on the rth edge of SV_i, and $nq = \text{integer}\,[(k+1)/2]$ is the number of quadrature points required on the rth edge. For the second-order schemes, one Gaussian quadrature point is used in the integration. Given the coordinates of the end points of the element edge, z_1 and z_2, one can obtain the quadrature point as the middle point of the segment connecting the two end points, $G_1 = \frac{1}{2}(z_1 + z_2)$. For the third- and fourth-order schemes, two Gaussian points are necessary along each line segment. Their values are given by

$$G_1 = \frac{\sqrt{3}+1}{2\sqrt{3}}z_1 + \left(1 - \frac{\sqrt{3}+1}{2\sqrt{3}}\right)z_2 \quad \text{and}$$
$$G_2 = \frac{\sqrt{3}+1}{2\sqrt{3}}z_2 + \left(1 - \frac{\sqrt{3}+1}{2\sqrt{3}}\right)z_1 \quad (14)$$

Using the method described above, one can compute values of $q_{i,j}$ for instant t for each CV. Because of the discontinuity of the reconstructed values of the conserved variables over SV boundaries, one must use a numerical flux function to approximate the flux values along the cell boundaries. At any moment during the simulation, one can compute the SV-averaged conserved variable vector, Q_i, for the ith spectral volume, as

$$Q_i = \frac{1}{V_i}\sum_{j=1}^{N_m} q_{i,j}V_{i,j} \quad (15)$$

The calculation of the SV-averaged values is important at the end of the computation in order to analyze the high-order numerical solution at the original grid level. The average is also used to recover the conserved variable vectors for the SVs, which are required for the limited reconstruction process as discussed in Sec. III.

As previously stated, the flux integration across CV boundaries that lies on the SV edges involves two discontinuous states, to the left and to the right of the edge. This flux computation can be carried out using a flux vector splitting procedure or an exact or approximate Riemann solver. The latter procedure is adopted in the present work and the numerical flux can be written as

$$\mathbf{f}\,(q(x_{rq},y_{rq}))\cdot\mathbf{n}_r \approx f_{\text{Riemann}}(q_L(x_{rq},y_{rq}), q_R(x_{rq},y_{rq}), \mathbf{n}_r) \quad (16)$$

where q_L is the conserved variable vector obtained by the p_i polynomial applied at the (x_{rq}, y_{rq}) coordinates and q_R is the same vector obtained with the p_{nb} polynomial in the same coordinates of the edge. Note that the nb subscript represents the element to the right of the edge, whereas the i subscript denotes the CV to its left. As the numerical flux integration in the present paper is based on one of the forms of a Riemann solver, this is the mechanism that introduces the upwind and artificial dissipation effects into the method, making it stable and accurate. In the present work, the authors have used the Roe flux difference splitting method [5] to compute the numerical flux, i.e.,

$$f_{\text{Riemann}} = f_{\text{roe}}(q_L, q_R, \mathbf{n}) = \tfrac{1}{2}[\mathbf{f}(q_L) + \mathbf{f}(q_R)]\cdot\mathbf{n} - \tfrac{1}{2}|\bar{B}|(q_R - q_L) \quad (17)$$

Here, $|\bar{B}|$ is the Roe matrix in the direction normal to the edge. The B matrix in the edge-normal direction is defined as

$$B = n_x\frac{\partial E}{\partial q} + n_y\frac{\partial F}{\partial q} \quad (18)$$

The B matrix has four real eigenvalues, namely, $\lambda_1 = \lambda_2 = v_n$, $\lambda_3 = v_n + a$, $\lambda_4 = v_n - a$, where v_n is the velocity component normal to the edge and a is the speed of sound. Let T be the matrix composed of the right eigenvectors of B. Then this matrix can be diagonalized as

$$T^{-1}BT = \Lambda \quad (19)$$

where Λ is the diagonal matrix composed of the eigenvalues of B, which can be written as

$$\Lambda = \text{diag}[v_n, v_n, v_n + a, v_n - a] \quad (20)$$

The $|\bar{B}|$ matrix is formed as

$$|\bar{B}| = T|\bar{\Lambda}|T^{-1} \quad (21)$$

where $\bar{\Lambda}$ and T are calculated as a function of the Roe-averaged properties [5]. Furthermore, $|\bar{\Lambda}|$ uses the magnitude of the eigenvalues.

Finally, one ends up with the semidiscrete SFV scheme for updating the CVs, which can be written as

$$\frac{\mathrm{d}q_{i,j}}{\mathrm{d}t} = -\frac{1}{V_{i,j}}\sum_{r=1}^{nf}\sum_{q=1}^{nq} w_{rq}f_{\text{Riemann}}(q_L(x_{rq},y_{rq}), q_R(x_{rq},y_{rq}), \mathbf{n}_r)A_r \quad (22)$$

where the right-hand side of Eq. (22) is the equivalent convective operator, $C(q_{i,j})$, for the jth control volume of SV_i. It is important to emphasize that some edges of the CVs, resulting from the partition of the SVs, lie inside the SV element in the region where the polynomial is continuous. For such edges, there is no need to compute the numerical flux, as described above. Instead, one uses analytical formulas for the flux computation, i.e., without numerical dissipation.

C. Temporal Discretization

To obtain the steady-state solution of the flow from an initial condition, a relaxation scheme is necessary. The convergence behavior to steady state of high-order methods, such as the SFV method, is generally poor with explicit time-marching approaches. The approach typically used in the present research group has been to resort to explicit multistage Runge–Kutta time-stepping methods. The main advantages of such an approach are that it is easy to implement and the memory requirements are quite modest. The code uses a three-stage TVD Runge–Kutta scheme for explicit time integration [8], i.e.,

$$q^{(1)} = q^n + \Delta t C(q^n), \qquad q^{(2)} = \alpha_1 q^n + \alpha_2[q^{(1)} + \Delta t C(q^{(1)})]$$
$$q^{(n+1)} = \alpha_3 q^n + \alpha_4[q^{(2)} + \Delta t C(q^{(2)})]$$

where the n and $n+1$ superscripts denote, respectively, the values of the properties at the beginning and at the end of the nth time step. The α coefficients are $\alpha_1 = 3/4, \alpha_2 = 1/4, \alpha_3 = 1/3$, and $\alpha_4 = 2/3$. The C operator represents the discretized convective operator computed as the right-hand side (RHS) of Eq. (22).

However, adequate solution convergence rate, especially for the higher-order implementations, dictate that an implicit time integrator should be employed. Therefore, an implicit LU-SGS scheme is also implemented in the context of the present work. Equation (22) can be rewritten as

$$V_{i,j}\frac{\mathrm{d}q_{i,j}}{\mathrm{d}t} = -R_{i,j} \quad (23)$$

where $R_{i,j}$ is the right-hand side residual for the jth CV of the ith SV and it tends to zero as the simulation converges to a steady-state solution. Using the implicit Euler method for time integration, Eq. (23) can be written in discrete form as

$$V_{i,j}\frac{\Delta q^n_{i,j}}{\Delta t} = -R^{n+1}_{i,j} \quad (24)$$

where Δt is the time increment and $\Delta q^n = q^{n+1} - q^n$. The previous equation can be linearized in time as

$$V_{i,j}\frac{\Delta q^n_{i,j}}{\Delta t} = -R^n_{i,j} - \frac{\partial R^n_{i,j}}{\partial q}\Delta q^n_{i,j} \tag{25}$$

The term $\partial R/\partial q$ represents the Jacobian matrix. Writing the equation for all CVs, one obtains the delta form of the backward Euler scheme:

$$A\Delta q = -R \tag{26}$$

where

$$A = \frac{V}{\Delta t}\mathbf{I} + \frac{\partial R^n}{\partial q} \tag{27}$$

and $\mathbf{I}$ is the identity matrix.

To reduce the number of nonzero entries in the Jacobian matrix and to simplify the linearization process, only a first-order representation of the numerical fluxes $\mathbf{f}$ is linearized. This results in a graph for the sparse matrix that is identical to the graph of the unstructured mesh. Hence, the Jacobian matrix entries can be computed and stored in a loop over the mesh edges. The ideal scenario would be to linearize the implicit operator with the same order as the RHS operator, to better approximate the ideal preconditioner and improve the convergence rate. However, this would prohibitively increase the storage requirements and, also, the complexity of the scheme formulation. Therefore, the simplified numerical flux function, which is actually used to form the residual operator R^n, which appears in Eq. (27), is defined as

$$\bar{f}_i(q_i, q_{nb}, \mathbf{n}_r) = \tfrac{1}{2}[\mathbf{f}(q_i, \mathbf{n}_r) + \mathbf{f}(q_{nb}, \mathbf{n}_r) - |\lambda|(q_{nb} - q_i)] \tag{28}$$

for the rth edge that shares the i and nb control volumes. A scalar dissipation model is used, where

$$|\lambda| = |v_n| + a \tag{29}$$

One should note that the dissipation in the flux function is approximated by the Jacobian matrix spectral radius. Moreover, one should also observe that the artificial dissipation is included for every edge of the CVs under consideration. In other words, there is no distinction of whether a particular edge is internal to the SV or it is a boundary edge of that SV, for the implementation of the implicit operator. Since a significant approximation is already performed by linearizing the implicit operator to first order, the code to distinguish between internal and external faces on the left-hand side would hamper the vectorized characteristic of this simplified numerical flux. Furthermore, such a database structure would make the implementation particular to the SFV scheme. The simplified numerical flux function is used to obtain the left-hand side Jacobian matrices. Hence, the linearization of Eqs. (27) and (28) yields

$$\frac{\partial R_i}{\partial q_i} = \frac{1}{2}[J(q_i) + |\lambda|\mathbf{I}] \qquad \frac{\partial R_i}{\partial q_{nb}} = \frac{1}{2}[J(q_{nb}) - |\lambda|\mathbf{I}] \tag{30}$$

where J is the Jacobian of the inviscid flux vectors in the direction normal to the edge.

Using an edge-based data structure, the Jacobian matrix is stored in lower, upper, and diagonal components, which are computed as

$$L = \frac{1}{2}[-J(q_{nb}, \mathbf{n}_r) - |\lambda|\mathbf{I}] \qquad U = \frac{1}{2}[J(q_{nb}, \mathbf{n}_r) - |\lambda|\mathbf{I}]$$
$$D = \frac{V}{\Delta t}\mathbf{I} + \sum_{nb}\frac{1}{2}[J(q_i, \mathbf{n}_r) + |\lambda|\mathbf{I}] \tag{31}$$

Note that L, U, and D represent the strict lower, upper, and diagonal matrices, respectively. Equation (26) represents a system of linear simultaneous algebraic equations that needs to be solved at each time step. The iterative LU-SGS solution method is employed, along with a mesh renumbering algorithm [18], and the system is solved in two steps, a forward and backward sweep:

$$(D + L)\Delta q^* = R \qquad (D + U)\Delta q = D\Delta q^* \tag{32}$$

This time-marching algorithm can be used independently of the spatial discretization scheme and, in fact, it is also applied with the WENO scheme to compare the CPU time with that of the SFV scheme. It should be observed, however, that a fourth-order SFV scheme, for example, requires a cubic polynomial reconstruction and, hence, each SV element must be partitioned into 10 CVs, rendering the actual computational mesh 10 times larger than the original SV grid. Such a fact must be considered on the choice of the implicit algorithm. For the scope of the present work though, which consists of a 2-D inviscid formulation, and the problems here considered, the implicit scheme memory requirement is acceptable and its formulation is easy to implement. Moreover, such an approach is not specific for a particular spatial discretization scheme, as previously discussed, which is important for the comparison studies carried out in the present work.

III. Numerical Formulation

A. General Formulation

The evaluation of the conserved variables at the quadrature points is necessary in order to perform the flux integration over the mesh element faces. These evaluations can be achieved by reconstructing conserved variables in terms of some base functions using the degrees of freedom within a SV. The present work has carried out such reconstructions using polynomial base functions, although one can choose any linearly independent set of functions. Let P_m denote the space of mth degree polynomials in two dimensions. Then the minimum dimension of the approximation space that allows P_m to be complete is

$$N_m = \frac{(m+1)(m+2)}{2} \tag{33}$$

To reconstruct q in P_m, it is necessary to partition the SV into N_m nonoverlapping CVs, such that

$$SV_i = \bigcup_{j=1}^{N_m} CV_{i,j} \tag{34}$$

The reconstruction problem, for a given continuous function in SV_i and a suitable partition, can be stated as finding $p_m \in P_m$ such that

$$\int_{CV_{i,j}} p_m(x, y)\,\mathrm{d}S = \int_{CV_{i,j}} q(x, y)\,\mathrm{d}S \tag{35}$$

With a complete polynomial basis, $e_\ell(x, y) \in P_m$, it is possible to satisfy Eq. (35). Hence, p_m can be expressed as

$$p_m = \sum_{\ell=1}^{N_m} b_\ell e_\ell(x, y) \tag{36}$$

where e is the base function vector, $[e_1, \ldots, e_{N_m}]$, and b is the reconstruction coefficient vector, $[b_1, \ldots, b_{N_m}]^T$. The substitution of Eq. (36) into Eq. (35) yields

$$\frac{1}{V_{i,j}}\sum_{\ell=1}^{N_m} b_\ell \int_{CV_{i,j}} e_\ell(x, y)\,\mathrm{d}S = q_{i,j} \tag{37}$$

If q denotes the $[q_{i,1}, \ldots, q_{i,N_m}]^T$ column vector, Eq. (37) can be rewritten in matrix form as

$$Sb = q \tag{38}$$

where the S reconstruction matrix is given by

$$S = \begin{bmatrix} \frac{1}{V_{i,1}}\int_{CV_{i,1}} e_1(x, y)\,\mathrm{d}S & \cdots & \frac{1}{V_{i,1}}\int_{CV_{i,1}} e_{N_m}(x, y)\,\mathrm{d}S \\ \vdots & \cdots & \vdots \\ \frac{1}{V_{i,N_m}}\int_{CV_{i,N_m}} e_1(x, y)\,\mathrm{d}S & \cdots & \frac{1}{V_{i,N_m}}\int_{CV_{i,N_m}} e_{N_m}(x, y)\,\mathrm{d}S \end{bmatrix} \tag{39}$$

and then the reconstruction coefficients b can be obtained as

$$b = S^{-1}q \tag{40}$$

provided that S is nonsingular. Substituting Eq. (40) into Eq. (36), p_m is then expressed in terms of shape functions $L = [L_1, \ldots, L_{N_m}]$, defined as $L = eS^{-1}$, such that one could write

$$p_m = \sum_{j=1}^{N_m} L_j(x, y) q_{i,j} = Lq \tag{41}$$

Equation (41) gives the value of the conserved state variable, q, at any point within the SV and its boundaries, including the quadrature points, (x_{rq}, y_{rq}). Note that q in the equation takes the form as a column vector, as presented in Eq. (38). The above equation can be interpreted as an interpolation of a property at a point using a set of cell-averaged values and the respective weights, which are set equal to the corresponding cardinal base value evaluated at that point.

Once the polynomial base functions e_ℓ are chosen, the L shape functions are uniquely defined by the partition of the spectral volume. The shape and partition of the SV can be arbitrary, as long as the S matrix is nonsingular. The major advantage of the SFV method is that the reconstruction process does not need to be carried out for every mesh element SV_i. Once the SV partition is defined, the same partition can be applied to all mesh elements and it results in the same reconstruction matrix. That is, the shape functions at similar points over different SVs have the same values. One can compute these coefficients as a preprocessing step and they do not change along the simulation. This single reconstruction is carried out only once for a standard element, for instance, an equilateral triangle, and it can be read by the numerical solver as input. This is a major difference when compared to k-exact methods, or ENO and WENO schemes, for which every mesh element has a different reconstruction stencil at every time step. Clearly, the SFV is more efficient in this step. Recently, several partitions for both 2-D and 3-D SFV reconstructions were studied and refined [12,19]. For the present work, the partition schemes are presented in the following sections. Moreover, the polynomial base functions for the linear, quadratic and cubic reconstructions are listed in Table 1. For more details regarding partition quality and stability analysis the interested reader is referred to [16,20].

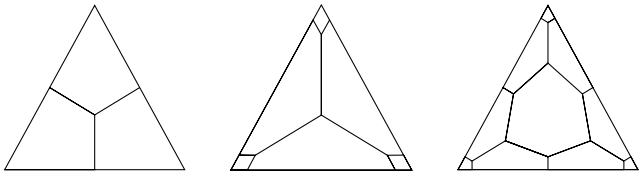
B. Linear Reconstruction

For the linear SFV method reconstruction, $m = 1$, one needs to partition a SV into three subelements, as in Eq. (33), and use the appropriate base vector as defined in Table 1. The partition scheme is given for a standard element and it is uniquely defined for this case. The structured aspect of the CVs within the SVs is used to determine neighborhood information and generate the global connectivity data considering a hash table search algorithm [21].

The linear partition is presented in Fig. 1a. It yields a total of seven points, nine edges (six are external edges and three are internal edges), and nine quadrature points. The linear polynomial for the SFV method depends only on the base functions and on the partition shape. The integrals of the reconstruction matrix in Eq. (39) are obtained analytically for the triangular mesh elements [22]. The shape functions, as defined in Eq. (41), are calculated and stored in memory for the quadrature points (x_{rq}, y_{rq}) of the standard element. There is one quadrature point located at the middle of the every CV edge. Moreover, such shape functions have the exact same value for the corresponding quadratures points of any other SV of the mesh, provided they all have the same partition.

Table 1 Polynomial base functions

Reconstruction order	e
Linear	$[1 \quad x \quad y]$
Quadratic	$[1 \quad x \quad y \quad x^2 \quad xy \quad y^2]$
Cubic	$[1 \quad x \quad y \quad x^2 \quad xy \quad y^2 \quad x^3 \quad x^2y \quad xy^2 \quad y^3]$



a) b) c)

Fig. 1 Triangular spectral volume partitions for a) linear, b) quadratic, and c) cubic reconstructions.

C. Quadratic Reconstruction

For the quadratic reconstruction, $m = 2$, one needs to partition a SV into six subelements and use the base vector as defined in the second line of Table 1. The partition scheme is also given in this work for an equilateral triangle. The nodes of the partition are obtained in terms of barycentric coordinates of the SV element nodes in the same manner as the linear partition. The structured aspect of the CVs within the SVs is used to determine neighborhood information and generate the connectivity table. The ghost-creation process and edge-based data structure is the same as for the linear reconstruction case. The partition used in the present work follows the orientations given in [20] and it is shown in Fig. 1b. It has a total of 13 points, 18 edges (nine external edges and nine internal edges), and 36 quadrature points. In this case, two quadrature points are required per CV edge for numerical flux integration. The shape functions, in the sense of Eq. (41), are obtained as in the linear partition. The reader should note that in this case, the base functions have six terms that shall be integrated. Again, these terms are obtained exactly [22] and kept in memory.

D. Cubic Reconstruction

For the cubic reconstruction, $m = 3$, one needs to partition the SV into 10 subelements and to use the base vector as defined in Table 1. The ghost-creation process and edge-based data structure is the same as for the linear and quadratic reconstruction cases. As a matter of fact, the same algorithm used to perform these tasks can be applied to higher-order reconstructions. The partition used in this work is the improved cubic partition [20], presented in Fig. 1c, and it has a total of 21 points, 30 edges (12 external and 18 internal edges), and 60 quadrature points The shape functions, in the sense of Eq. (41), are obtained as in the linear partition in a preprocessing step. As with the quadratic reconstruction, each CV edge has two quadrature points for numerical flux integration.

E. High-Order Boundary Representation

An advantage of high-order methods is that fewer unknowns are required to achieve a given level of accuracy than with lower-order methods. Therefore, the computational grids used in high-order simulations can be considered coarse when compared to those in second-order simulations. In second-order CFD simulations, all curved boundaries are represented with line segments or planar facets, since such representation is compatible with the linear interpolations used in second-order flow solvers. To minimize the solution errors produced by this "crude" approximation of curved boundaries, many elements may be required to simply preserve the geometry with a reasonable accuracy. If this crude geometric representation is used in high-order simulations, unnecessarily fine grids may be required near curved boundaries to represent the boundary with high fidelity. Obviously, this practice can waste significant computational resources. A much more desirable approach is to represent curved boundaries with higher-order polynomials compatible with the order of the reconstruction. For example, quadratic or cubic polynomials should be used to approximate boundaries in third- and fourth-order SFV schemes, which employ quadratic and cubic polynomials to represent the solution variables.

Following the usual practice in high-order finite element method [23], isoparametric SVs can be used to map SVs with curved boundaries into standard SVs. Hence, one can assume that a

one-to-one transformation exists between a general SV in the physical space, (x, y), and the standard triangle in the computational domain, (ξ, η); i.e.,

$$\xi = \xi(x, y) \qquad \eta = \eta(x, y) \tag{42}$$

The partition of the SV is performed in the standard triangle, and the partition in the physical domain is the result of the inverse transformation from the computational domain back to the physical domain. Therefore, a necessary condition for a valid transformation is that the inverse transformation exists. The degrees of freedom for the general SV are the CV-averaged state variables in the physical domain:

$$q_{i,j} = \frac{\int_{CV_{i,j}} q(x, y)\, \mathrm{d}x\, \mathrm{d}y}{\int_{CV_{i,j}} \mathrm{d}x\, \mathrm{d}y} = \frac{\int_D q(\xi, \eta)|J|\, \mathrm{d}\xi\, \mathrm{d}\eta}{\int_D |J|\, \mathrm{d}\xi\, \mathrm{d}\eta} \tag{43}$$

where J is the Jacobian matrix of the transformation, i.e., $J = \partial(x, y)/\partial(\xi, \eta)$. The reconstruction problem can be stated as constructing a $(k-1)$-degree polynomial, $p_i(\xi, \eta)$, in the computational domain, given the CV-averaged state variables in all the CVs of a SV, such that

$$\int_D p_i(\xi, \eta)|J|\, \mathrm{d}\xi\, \mathrm{d}\eta = q_{i,j} \int_D |J|\, \mathrm{d}\xi\, \mathrm{d}\eta \tag{44}$$

Given a basis set and a valid transformation, the reconstruction problem can be solved either analytically or numerically to obtain the reconstruction coefficients. However, different curved SVs have different reconstruction coefficients. Therefore, it is necessary to store these coefficients for SVs with curved boundaries. Since the number of curved SVs is expected to be small, when compared to the total number of SVs in any simulation, the memory cost for storing these coefficients is also small.

Geometry information is necessary in order to correctly position the nodes of the high-order SV element. The authors have chosen to use exact geometry data instead of approximate data to represent all boundaries as splines in IGES geometry standard [24].

F. Limited Reconstruction

For the Euler equations, it is necessary to limit some reconstructed properties at flux integration points in order to maintain stability and convergence of the simulation, if the flow solution contains discontinuities. The present limiter technique involves two stages. First, the solver must find and mark "troubled cells, " which are limited in the second stage. For the detection and limiting process, the limiter employs a Taylor series expansion for the reconstruction [25] with regard to the cell-averaged derivatives. The troubled cells are then limited in a hierarchical manner, i.e., from the highest-order derivative to the lowest-order derivative. If the highest derivative is not limited, the original polynomial is preserved and so is the order of the method at the element level. This limiter technique is capable of suppressing spurious oscillations near solution discontinuities without loss of accuracy at local extrema in smooth regions. Originally, this limiter methodology was developed for the spectral difference method in [17]. In the present work, the formulation is extended for the SFV method. The extension here reported applies some ideas from previous work on ENO and WENO schemes [8] in the calculation of the limited properties. It should be emphasized that there are other approaches that implement a similar hierarchical moment limiter formulation for the SFV scheme [26], but with slight differences, in comparison to the present approach, on the calculation of the derivatives for the limited reconstruction.

Several markers (or sensors) were developed and employed for unstructured meshes over the past decades. For an in-depth review, the interested reader is referred to [27]. The limiter marker used in the present work is termed accuracy-preserving TVD marker [17]. One important aspect is that the troubled-cell and limited properties are inherent to the SV element, and not to the CVs in which the flux calculations are performed. Once the SV element is confirmed as a troubled cell, its polynomial, based on CV cell-averaged variables, can no longer be used in any flux integration point, because the property function is no longer smooth within such an SV. Hence, it is of utmost importance to limit as few SVs as possible. To that end, the marker is designed to first check for the flux integration points in each SV and mark those that do not satisfy the monotonicity criterion. However, if an extremum is smooth, the first derivative of the solution, at such a point, should be locally monotonic. Hence, on a second moment and using derivative information, as described in the forthcoming discussion, the limiter sensor unmarks those SVs at local extrema that were unnecessarily marked as troubled cells. Therefore, for instance, for a quadratic reconstruction, the limiting process can be summarized in the following stages:

1) For a given spectral volume SV_i, compute the minimum and maximum cell averages using a local stencil that includes its immediate face neighbors, i.e.,

$$Q_{\min,i} = \min(Q_i, \min_{1 \le r \le nf} Q_r) \qquad Q_{\max,i} = \max(Q_i, \max_{1 \le r \le nf} Q_r) \tag{45}$$

2) The ith cell is considered as a possible troubled cell if

$$p_i(x_{rq}, y_{rq}) > 1.001 Q_{\max,i} \quad \text{or} \quad p_i(x_{rq}, y_{rq}) < 0.999 Q_{\min,i} \tag{46}$$

The 1.001 and 0.999 constants are not problem-dependent. They are simply used to overcome machine error, when comparing two real numbers, and to avoid the trivial case of when the solution is constant in the neighborhood of the spectral volume considered. This step is performed in order to check the monotonicity criterion.

3) Since the previous steps may flag more SVs than strictly necessary, the next operations attempt to unmark SVs in smooth regions of the flow. Hence, for a given marked spectral volume, a minmod TVD function is applied to verify whether the cell-averaged second derivative is bounded by an estimate of the second derivative obtained using cell-averaged first derivatives of the neighboring spectral volumes. Such test is performed as

a) If the unit vector in the direction connecting the centroids of the ith and nbth cells is denoted $\boldsymbol{l} = l_x\hat{i} + l_y\hat{j}$, where nb indicates the face-neighbor of a marked SV_i, the second derivative in such a direction is defined as

$$Q_{ll,i} = Q_{xx,i} l_x^2 + 2Q_{xy,i} l_x l_y + Q_{yy,i} l_y^2 \tag{47}$$

b) In a similar fashion, the first derivatives in the same $\boldsymbol{l}$ direction, for both ith and nbth cells, can be computed as

$$Q_{l,i} = Q_{x,i} l_x + Q_{y,i} l_y, \qquad Q_{l,nb} = Q_{x,nb} l_x + Q_{y,nb} l_y \tag{48}$$

c) Another estimate of the second derivative, in the $\boldsymbol{l}$ direction, can be obtained as

$$\tilde{Q}_{ll,i} = \frac{Q_{l,nb} - Q_{l,i}}{|\mathbf{r}_i - \mathbf{r}_{nb}|} \tag{49}$$

where $\mathbf{r}$ is the centroid coordinate vector;

d) A scalar limiter for this face is computed according to

$$\phi_{i,nb}^{(2)} = \min \mathrm{mod}\left(1, \frac{\tilde{Q}_{ll,i}}{Q_{ll,i}}\right) \tag{50}$$

e) The process is repeated for the other faces of SV_i, and the scalar limiter for the SV is the minimum among those computed for the faces, i.e.,

$$\phi_i^{(2)} = \min_{nb}(\phi_{i,nb}^{(2)}) \tag{51}$$

f) If $\phi_i^{(2)} = 1$, the second derivatives are bounded, as previously defined, and, hence, SV_i is actually in a smooth region of the flow. Therefore, SV_i is unmarked.

4) If the previous test is not satisfied, this means that the particular SV_i spectral volume should indeed be limited. In this case, the limiter for the first derivative reconstruction must also be computed. The

calculation procedure follows the same approach as for the second derivatives, and it can be summarized as follows:

a) An estimate of the first derivative in the $\boldsymbol{l}$ direction is calculated as

$$\tilde{Q}_{l,i} = \frac{Q_{nb} - Q_i}{|\mathbf{r}_i - \mathbf{r}_{nb}|} \tag{52}$$

b) Such estimate is compared to the cell-average first derivative in the $\boldsymbol{l}$ direction, computed according to Eq. (48), in order to obtain the scalar limiter for the face as

$$\phi^{(1)}_{i,nb} = \min \mathrm{mod}\left(1, \frac{\tilde{Q}_{l,i}}{Q_{l,i}}\right) \tag{53}$$

c) As before, the scalar limiter for the cell is the minimum of those limiters computed for the faces, i.e.,

$$\phi^{(1)}_i = \min_{nb}(\phi^{(1)}_{i,nb}) \tag{54}$$

The cell-averaged derivatives for the ith cell, necessary to perform the above calculations, are obtained by solving a quadratic least-squares reconstruction problem, for a third-order scheme, or a cubic least-squares reconstruction problem, for a fourth-order scheme. For the quadratic reconstruction presented here, one would impose the mean conservation constraint in the first row of the least-squares system and solve the following linear system:

$$\begin{bmatrix} M_{x^1y^0}|_{nb1} & M_{x^0y^1}|_{nb1} & M_{x^2y^0}|_{nb1} & M_{x^1y^1}|_{nb1} & M_{x^0y^2}|_{nb1} \\ M_{x^1y^0}|_{nb2} & M_{x^0y^1}|_{nb2} & M_{x^2y^0}|_{nb2} & M_{x^1y^1}|_{nb2} & M_{x^0y^2}|_{nb2} \\ M_{x^1y^0}|_{nb3} & M_{x^0y^1}|_{nb3} & M_{x^2y^0}|_{nb3} & M_{x^1y^1}|_{nb3} & M_{x^0y^2}|_{nb3} \\ M_{x^1y^0}|_{nb4} & M_{x^0y^1}|_{nb4} & M_{x^2y^0}|_{nb4} & M_{x^1y^1}|_{nb4} & M_{x^0y^2}|_{nb4} \\ M_{x^1y^0}|_{nb5} & M_{x^0y^1}|_{nb5} & M_{x^2y^0}|_{nb5} & M_{x^1y^1}|_{nb5} & M_{x^0y^2}|_{nb5} \end{bmatrix} \times \begin{Bmatrix} Q_x \\ Q_y \\ Q_{xx} \\ Q_{xy} \\ Q_{yy} \end{Bmatrix} = \begin{Bmatrix} Q_{nb1} - Q_i \\ Q_{nb2} - Q_i \\ Q_{nb3} - Q_i \\ Q_{nb4} - Q_i \\ Q_{nb5} - Q_i \end{Bmatrix} \tag{55}$$

The matrix terms are the SV area moments and can be computed, up to the desired order of accuracy, by numerical integration as

$$M_{x^m y^n}|_i = \int_{SV} (x_{rq} - x_i)^m (y_{rq} - y_i)^n \, \mathrm{d}V \tag{56}$$

The SV area moments are computed during an initial stage of the numerical solver and kept in memory for efficiency. The $nb1$ to $nb5$ subscripts represent the neighbors of SV_i that form the computational stencil to compute the averaged derivatives. This stencil is determined by an ENO based search, as in [8], for the smoothest SV set. Since only a small number of SVs is selected for limited reconstruction, the overhead of this search does not adversely affect the overall performance of the scheme. It is also important to observe that, considering all the information already available in a SFV implementation, there are other possible approaches to compute the averaged derivatives. Actually, such approaches can be more computationally efficient than the one here adopted. The interested reader is referred, for instance, to the work presented in [17] for further details of one such alternative approach.

Finally, the quadratic limited polynomial, which is used in order to obtain property values at the quadrature points for a troubled SV_i spectral volume, is given by

$$p_i^{\mathrm{limited}}(x_{rq}, y_{rq}) = Q_i + \phi_i^{(1)}\left[\frac{1}{V_i}(Q_x M_x + Q_y M_y)_i\right] + \phi_i^{(2)}\left[\frac{1}{V_i}\left(\frac{1}{2}Q_{xx}M_{x^2} + Q_{xy}M_{xy} + \frac{1}{2}Q_{yy}M_{y^2}\right)_i\right] \tag{57}$$

The limited reconstruction is based on primitive variables $\{\rho, u, v, p\}^T$, instead of conserved variables, as suggested in [28]. Once these properties are available from the limited reconstruction, the vector of conserved variables is easily obtained to resume the numerical flux integration.

IV. Results

The results presented here attempt to validate both the implementation of the data structure, temporal integration, numerical stability and resolution of the SFV method. The overall performance of the method is compared with that of a third-order WENO scheme implementation, employing an oscillation indicator proposed by Jiang and Shu [29], with the modification of Friedrich [30]. The Roe numerical flux is also used with the WENO scheme. Moreover, the geometric coefficients for the WENO reconstructions are computed in a preprocessing step and kept in memory during the computation. For more details on the WENO scheme formulation used, the interested reader is referred to the work in [8].

For the results here reported, density is made dimensionless with respect to the freestream condition and pressure is made dimensionless with respect to the freestream density times the freestream speed of sound squared. For the steady case simulations, the Courant–Friedrichs–Lewy (CFL) number is set as a constant value and the local time step is computed using the local grid spacing and characteristic speeds. For all test cases, the CFL number is set to 10^6.

All numerical simulations are carried out on a dual-core 1.6 GHz PC Intel64 architecture, with Linux OS. The code is written in FORTRAN 95 language and the Intel FORTRAN compiler with optimization flags[§] is used. For the performance comparisons that are presented in this section, all residuals are normalized by the first iteration residue. Moreover, the L_2 norm is used in all residuals here reported, except when explicitly noted otherwise.

A. Wedge Flow

The first test case is the computation of the supersonic flowfield past a wedge with half-angle $\theta = 15$ deg.[¶] The computational mesh has 2409 nodes and 4613 volumes. Numerical pressure contours obtained with the third-order SFV method are shown in Fig. 2. For comparison purposes, the third-order SFV and WENO methods are used for this simulation. The leading edge of the wedge is located at coordinates $(x, y) = (0, 0)$. The computational domain is bounded along the bottom by the symmetry plane of the wedge and the wedge surface. The inflow boundary is located at $x = -0.5$. The outflow boundary is placed at $x = 1.0$. The far-field boundary is placed at $y = 1.0$, in order to guarantee that it is located above the oblique shock. The analytical solution [31,32] gives the change in properties across the oblique shock as a function of the freestream Mach number and shock angle, which is obtained from the θ–β–Mach relation. For this case, a freestream Mach number of $M_1 = 2.5$ is used, and the oblique shock angle β is obtained as 36.94 deg. For the present simulations, the limiter is turned on, and only elements in the shock wave region are marked for limited reconstruction, as one can observe in Fig. 3. This figure shows, in solid black, the spectral volumes in which the limiter is active for pressure reconstruction in the converged solution for the present test case.

The numerical solution of the SFV method is in good agreement with the analytical result for this case, as observed in Table 2, which presents the analytical and numerical ratio of properties before (subscript 1) and after (subscript 2) the oblique shock wave. The

[§]Compiler flags: -O3 -assume buffered_io -parallel.

[¶]Data available online at http://www.grc.nasa.gov/WWW/wind/valid/wedge/wedge.html [retrieved 28 May 2010].

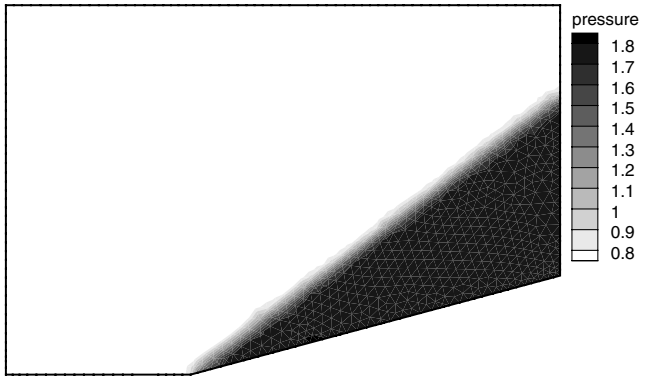


Fig. 2 Supersonic wedge flow pressure contours for the third-order SFV method.

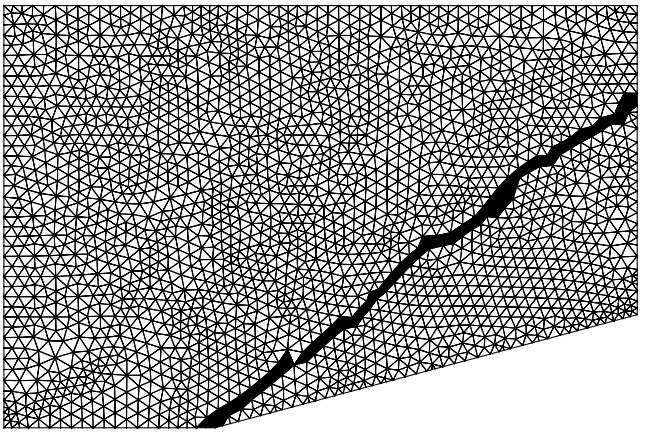


Fig. 3 Marked SVs for pressure-limiting considering a third-order SFV simulation of the supersonic wedge flow.

numerical results reported in Table 2 are obtained by performing some averaging of the computational results right after the shock in the interior of the domain. The third-order results are compared in Fig. 4, in terms of Mach number distribution, with the analytical values. Note that the SFV scheme is the one that better approximates the jump in Mach number at the leading edge.

It should be observed that the computational results shown in Fig. 4 are slightly different than those reported on Table 2 because these are results at the wedge surface whereas the computed value of 1.88 for M_2 is obtained in the interior of the domain, as noted. The data in Fig. 4 indicates that the SFV method yields results that are slightly better than those calculated by the WENO scheme, especially downstream of the shock wave. Moreover, the SFV method is much less expensive than the WENO calculation, for the same test case, as indicated in Figs. 5a and 5b. It should be further emphasized that the third-order SFV method achieved a residual drop of 12 orders of magnitude, whereas the WENO method stalled convergence after the first few hundred iterations, even with the use of an implicit time integration. One should note that the calculation with the SFV method has 6 times more CVs than the WENO calculation has control volumes, for the same mesh. Therefore, the

Table 2 Comparison of analytical and computational results for the oblique shock wave properties

Property	Analytical	Numerical
M_2	1.8735	1.88
ρ_2/ρ_1	1.8663	1.87
p_2/p_1	2.4675	2.47
T_2/T_1	1.3218	1.32

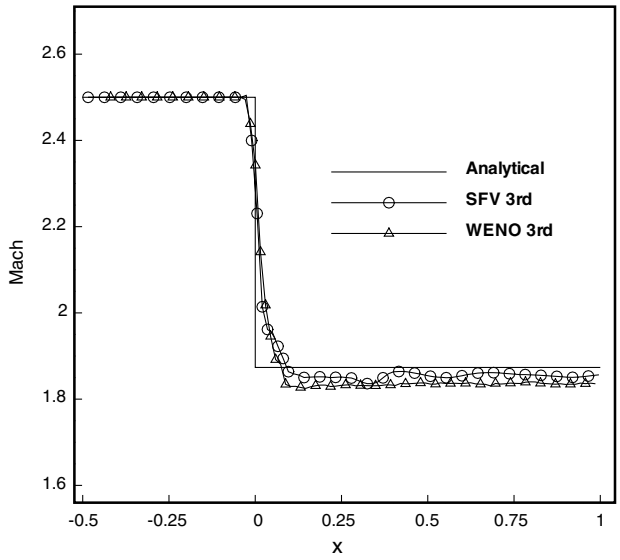


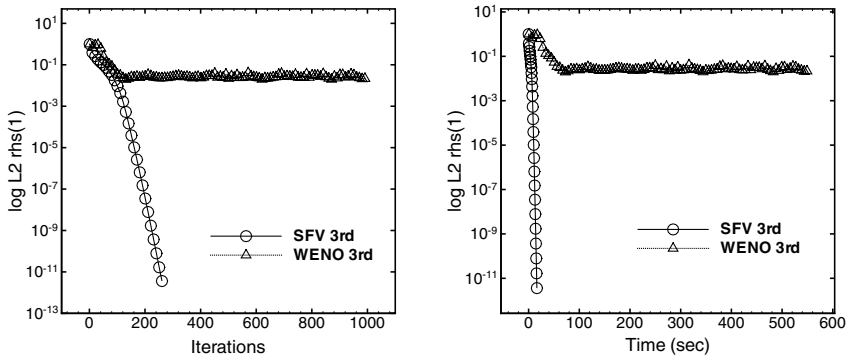
Fig. 4 Analytical and numerical supersonic wedge flow Mach number distributions.

results highlight the improved computational efficiency achieved by the SFV approach. On the other hand, such a result should not be a surprise, because a third-order WENO scheme on a triangular grid has to perform the reconstruction for a large number of possible computational molecules for each cell at every time step with the current implementation.

B. Ringleb Flow

The Ringleb flow simulation consists of an internal flow, which has an analytical solution for the Euler equations derived with the hodograph transformation [33]. The analytical solution is used as initial condition for all simulations here discussed. The flow depends on the inverse of the stream function, k, and the velocity magnitude, v_t. In the present simulations, these parameters are chosen as $k = 0.4$ and 0.6 to define the bounding walls and $v_t = 0.35$ to define the inlet and outlet boundaries. For such a configuration, the test case represents an irrotational and isentropic flow around a symmetric blunt obstacle. An interesting property of the Ringleb test case is that transition of flow regime, from subsonic to supersonic, for example, is shockless [34]. To measure the order of the implemented SFV method, four meshes are considered for the mesh refinement study, corresponding to 128, 512, 2048, and 8192 spectral volume elements. The analytical solution is computed for all meshes in order to measure how close the numerical results are to the exact solution. The error with respect to the analytical solution is computed using the L_1 and L_∞ norms of the density. Figure 6 shows the 2048-element grid and the Mach number contours computed in this grid with the fourth-order SFV method, using the corresponding high-order boundary representation.

It should be pointed out that the same numerical test case was studied in [16], considering only the linear boundary representation. It was observed in that effort that the low-order boundary treatment causes a shock wave to develop close to the inner boundary, which then makes the limiter active. Eventually, the shock wave propagates and it causes the simulation to diverge. In the present work, however, which considers the higher-order boundary representation, reasonable results are always obtained for this test case, including the simulations with the fourth-order SFV method. As previously discussed, for the third-order scheme, a quadratic polynomial is used to represent the SV edges that lie along the geometry boundaries. In a similar fashion, for the fourth-order scheme, a cubic polynomial is employed instead, which is compatible with the internal polynomial order of each SV. Table 3 presents the L_1 and L_∞ error norms of the density for the present calculations with the high-order boundary representation. The table also shows the actual measured order of accuracy for the third- and fourth-order SFV methods. The orders of



a) b)

Fig. 5 Convergence history for wedge flow with implicit third-order SFV and WENO schemes.

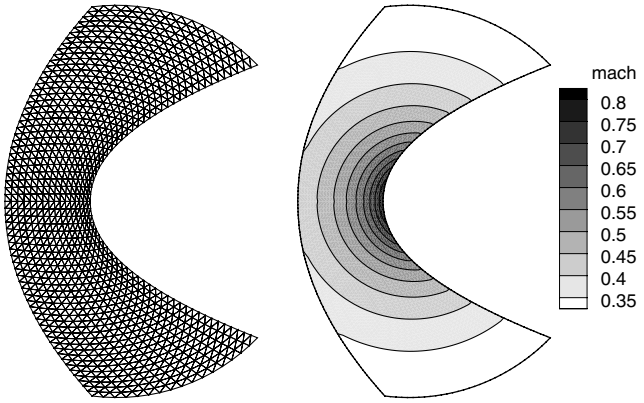


Fig. 6 Ringleb flow mesh and Mach number contour results for fourth-order SFV method.

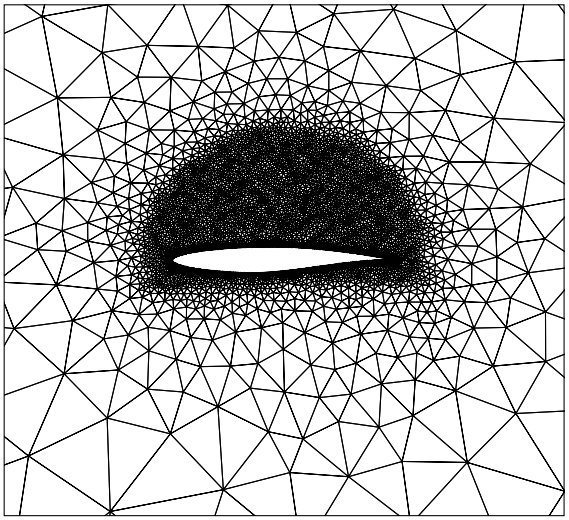


Fig. 7 Fine mesh used for second-order WENO computation.

accuracy in the results shown in Table 3 are calculated as indicated in [34]. The actual orders of accuracy here obtained are in good agreement with those shown in the cited reference.

C. RAE 2822 Flow

The transonic flow over a RAE 2822 airfoil with 2.31 deg. angle of attack and freestream Mach number $M_\infty = 0.729$ is also considered. To evaluate the present implementation against computations performed with a WENO scheme [8], two different simulations are performed for this test case. Initially, a simulation with a second-order WENO scheme is performed on a standard finite volume mesh that uses linear boundary representation. The mesh for such a calculation has 16,383 nodes and 32,399 elements, of which 340 elements lie on the airfoil surface, as necessary to properly describe the airfoil geometry, especially on the leading edge. This mesh is shown in Fig. 7. The second simulation is carried out for the third-order SFV method with quadratic boundary representation. The approach does not require a very fine mesh, as in the first test case, because the geometry definition file is directly available to the solver. Moreover, only the edges of SVs that are on the airfoil surface are mapped and stored for high-order boundary computation. As discussed, only the boundary spectral volumes with high-order curved edges need a special treatment and require the additional storage. Spectral volumes in the interior of the domain are always treated through the use of the standard partition scheme. The coarse mesh, used for this second test case, has 2,700 nodes and 5,265 elements, and it is shown in Fig. 8. This mesh has only 85 spectral volume elements along the airfoil surface. It should be noted, however, that although the coarse mesh has much fewer elements, the computation is actually carried over on 31,590 CV elements for the third-order SFV method. Therefore, the two test cases are comparable.

Table 3 Accuracy assessment of SFV method for the Ringleb flow test case

Mesh elements	L_1 error	L_1 order	L_∞ error	L_∞ order
	Third-order SFV			
128	2.41E − 02	——	2.14E − 01	——
512	4.14E − 03	2.54	2.45E − 02	3.13
2048	6.27E − 04	2.72	3.13E − 03	2.97
8192	8.67E − 05	2.85	3.60E − 04	3.12
	Fourth-order SFV			
128	5.77E − 04	——	6.11E − 03	——
512	6.48E − 05	3.16	4.52E − 04	3.76
2048	6.15E − 06	3.39	5.68E − 05	2.99
8192	6.87E − 07	3.16	5.62E − 06	3.33

The present test cases evidence the large benefit of using a high-order method for spatial discretization. The ability to obtain an equivalent numerical solution with much less stringent mesh requirements than would be necessary to standard second-order methods is an important advantage. The numerical C_p results for the two test cases are presented in Fig. 9, together with the corresponding experimental data [35]. The numerical solution of the SFV method is in overall good agreement with the experimental data and with the calculations from the WENO scheme with the finer mesh, as shown in Fig. 9. Clearly, the experimental data shows a smoothed transonic shock, due to shock-wave/boundary-layer interaction, which is an important phenomenon for such supercritical airfoils. The present computations cannot reproduce such physics, since the Euler equations are used. Furthermore, Fig. 10 presents the L_2 norm of the density residues as a function of the number of iterations and of the actual wall-clock time. Such results indicate that the third-order SFV

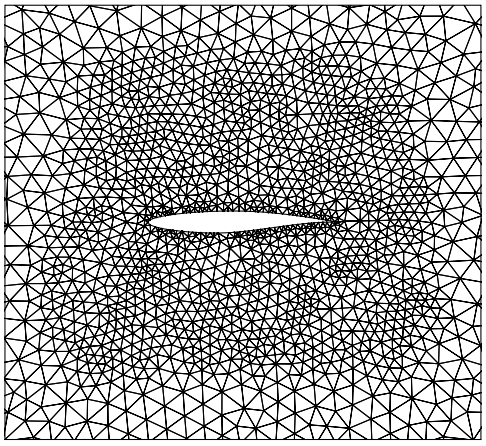


Fig. 8 Coarse mesh used for third-order SFV computation.

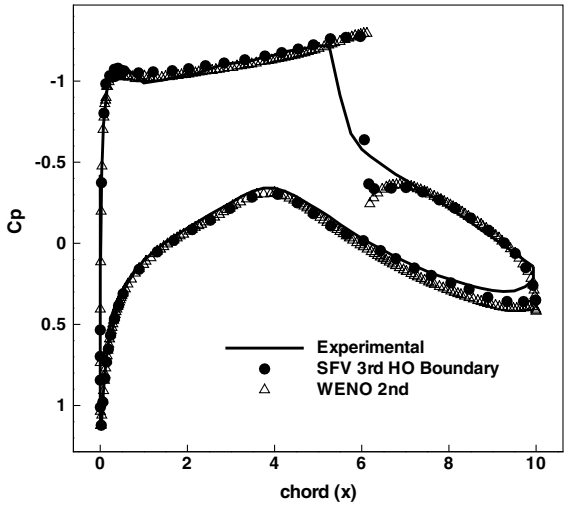
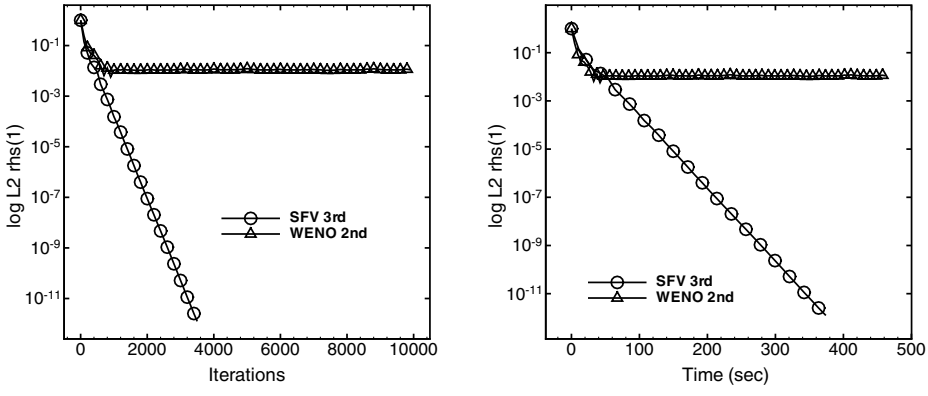


Fig. 9 C_p distributions for RAE 2822 airfoil at $M_\infty = 0.729$ and $\alpha = 2.31$ deg.

method, with proper boundary treatment, outperforms the second-order simulation, especially in terms of the number of iterations. It should be emphasized, however, that although the residues in the WENO calculations do not drop to machine zero, the solution indeed converges. It is clear that one would typically feel more comfortable with residue histories that go to machine zero, as the present SFV method calculations. One should further observe that the comparison in terms of wall-clock time indicates that the computational costs for the two simulations are actually comparable. In this particular case, the added cost in the third-order SFV method calculation comes about due to the need of limited reconstructions around the shock wave and the additional operations associated with high-order boundary flux integrations. Nevertheless, the comparison here considers a second-order WENO scheme and a third-order SFV scheme. If such a comparison had considered third-order schemes in both cases, previous experience [16] indicates that the SFV method would easily outperform the WENO calculation even in terms of wall-clock time.

The limiter ability to mark only those SV elements strictly along the discontinuity is another important aspect explored by the present test case. The limited SVs in the final iteration, for the solution with the third-order SFV method, are identified (in solid black) in Fig. 11. One can observe from this figure that the marked SVs only correspond to those in the numerical shock wave structure, as previously discussed in the present paper. The process of identifying SVs that need limiting and of actually performing the reconstruction of limited properties can have a significant impact on the computational cost of the simulation. For some test cases, such procedures can double the computational time per iteration. For instance, for every limited SV and considering a third-order accuracy SFV scheme, there are 36 numerical flux calculations at quadrature points that must be reevaluated with the limited properties and using the appropriate (approximate) Riemann solver. Such additional calculations already represent almost twice the computational cost for a nonlimited SV, because only 18 numerical flux evaluations are required per SV in the standard (nonlimited) application of the third-order SFV method. Therefore, by the present limiter implementation, only a few SV elements are actually limited during the computation, rendering the method competitive. Moreover, the limiter implementation here discussed is free from user-dependent input parameters, which is always a good feature for a robust flow solver.

The entropy error along the airfoil surface for both numerical results is shown in Fig. 12. The solution for the airfoil lower surface has very small entropy errors. On the upper surface, however, one can observe that the high-order boundary, third-order SFV method solution still has very small entropy errors, but the second-order WENO solution already indicates a much higher level of entropy. The presence of the shock wave can be clearly identified by the sharp increase in the entropy errors for both solutions. Entropy values observed in Fig. 12 for the second-order WENO scheme are relatively close to those observed in the literature, for instance, in [36]. This reference presents entropy contours for a fully subsonic flow over a NACA 0012 airfoil, computed with second- and fourth-order spatial resolutions. As discussed, entropy error levels for the second-order solution in the present case are similar to those seen in [36]. For the third-order SFV scheme, however, there is no direct comparison with the cited reference, but it seems that the present results are giving entropy error levels slightly higher than the literature. It is possible that the entropy error behavior in the present case is related to the smoothness of the grids used. The literature, considering for instance [34,36], seems to use meshes that are



a) **b)**

Fig. 10 Convergence histories for RAE 2822 airfoil with implicit third-order SFV and second-order WENO schemes.

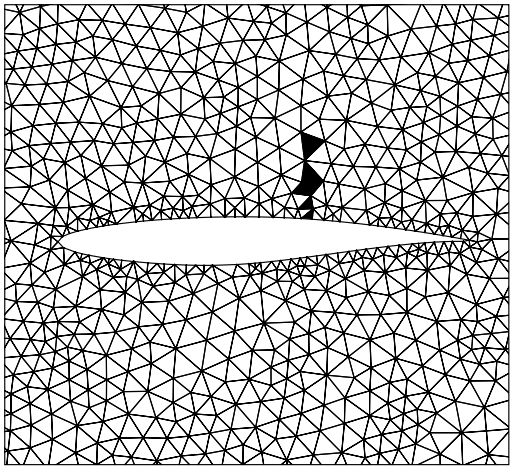


Fig. 11 Limited SVs in the final iteration for third-order SFV simulation of the RAE 2822 airfoil flow.

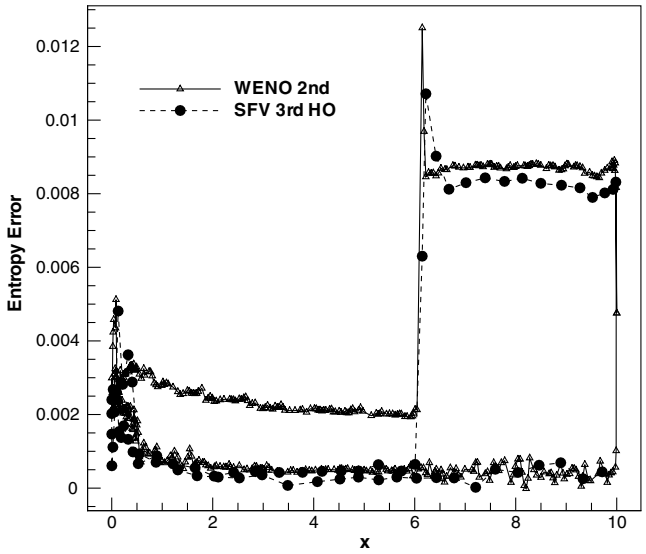


Fig. 12 Entropy errors along the RAE 2822 airfoil surface for WENO and SFV method solutions.

smoother than the one considered in the present paper, as shown in Figs. 8 and 11. In any event, as expected, there are clear advantages in terms of entropy errors in the use of the high-order boundary treatment and the high-order spatial discretization.

V. Conclusions

The application of the high-order SFV method to inviscid compressible flow simulations is presented. An efficient implementation with regard to CPU time usage is achieved by employing several techniques. One such technique consists of the use of an implicit time-marching scheme, which allows for large time steps for the steady-state calculations here considered. The high-order boundary treatment is also an important aspect, because it allows the use of a much coarser mesh, when compared to that required by second-order schemes to achieve a similar level of geometry resolution. The present implementation of the limiter technique (which extends, to SFV methods, ideas that have been previously tested on spectral difference schemes) is an important ingredient of the method efficiency. The present limiter reduces the number of limited spectral volumes to a bare minimum, which reduces computational costs and, at the same time, allows for a more uniform high-order solution. Furthermore, a user-input-free limiter implementation contributes to enhance the robustness of the flow solver.

The test cases addressed in the present work have indicated that the present SFV method implementation is able to accurately predict the flowfields of interest. In particular, the transonic airfoil simulation is representative of the actual flow features present in the relevant applications. In this test case, all three features previously indicated have been brought to bear in the calculations and they have indeed yielded the expected behavior to the method. The techniques added to the present implementation have resolved the problems the authors experienced in the past with high-order methods, mainly due to the implicit time-marching algorithm and reduction of the mesh elements count, due to the high-order boundary representation, as shown in the RAE 2822 test case.

Acknowledgments

The authors gratefully acknowledge the partial support provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), under the Integrated Project Research grant no. 312064/2006-3. The authors are also grateful to Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, which also partially supported the present research through project no. 2004/16064-9.

References

[1] Scalabrin, L. C., "Numerical Simulation of Three-Dimensional Flows over Aerospace Configurations," M.S. Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2002.

[2] Basso, E., Antunes, A. P., and Azevedo, J. L. F., "Chimera Simulations of Supersonic Flows over a Complex Satellite Launcher Configuration," *Journal of Spacecraft and Rockets*, Vol. 40, No. 3, May–June 2003, pp. 345–355. doi:10.2514/2.3969

[3] van Leer, B., "Flux-Vector Splitting for the Euler Equations," *Lecture Notes in Physics*, Vol. 170, 1982, pp. 507–512. doi:10.1007/3-540-11948-5_66

[4] Liou, M. S., "A Sequel to AUSM: AUSM+," *Journal of Computational Physics*, Vol. 129, No. 2, Dec. 1996, pp. 364–382. doi:10.1006/jcph.1996.0256

[5] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372. doi:10.1016/0021-9991(81)90128-5

[6] Anderson, W. K., Thomas, J. L., and van Leer, B., "Comparison of Finite Volume Flux Vector Splittings for the Euler Equations," *AIAA Journal*, Vol. 24, No. 9, 1986, pp. 1453–1460. doi:10.2514/3.9465

[7] Azevedo, J. L. F., Figueira da Silva, L. F., and Strauss, D., "Order of Accuracy Study of Unstructured Grid Finite Volume Upwind Schemes," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Vol. 32, No. 1, Jan.–March 2010, pp. 78–93.

[8] Wolf, W. R., and Azevedo, J. L. F., "High-Order Unstructured Essentially Non Oscillatory and Weighted Essentially Non Oscillatory Schemes for Aerodynamic Flows," *AIAA Journal*, Vol. 44, No. 10, Oct. 2006, pp. 2295–2310. doi:10.2514/1.19373

[9] Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids: Basic Formulation," *Journal of Computational Physics*, Vol. 178, No. 1, May 2002, pp. 210–251. doi:10.1006/jcph.2002.7041

[10] Wang, Z. J., and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids II: Extension to Two-Dimensional Scalar Equation," *Journal of Computational Physics*, Vol. 179, No. 2, July 2002, pp. 665–698. doi:10.1006/jcph.2002.7082

[11] Wang, Z. J., and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids III: One-Dimensional Systems and Partition Optimization," *Journal of Scientific Computing*, Vol. 20, No. 1, Feb. 2004, pp. 137–157. doi:10.1023/A:1025896119548

[12] Wang, Z. J., Liu, Y., and Zhang, L., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids IV: Extension to Two-Dimensional Systems," *Journal of Computational Physics*, Vol. 194, No. 2, March 2004, pp. 716–741. doi:10.1016/j.jcp.2003.09.012

[13] Liu, Y., Vinokur, M., and Wang, Z. J., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems," *Journal of Computational Physics*, Vol. 212,

No. 2, March 2006, pp. 454–472.
doi:10.1016/j.jcp.2005.06.024

[14] Sun, Y., Wang, Z. J., and Liu, Y., "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow," *Journal of Computational Physics*, Vol. 215, No. 1, June 2006, pp. 41–58.
doi:10.1016/j.jcp.2005.10.019

[15] Kannan, R., Sun, Y., and Wang, Z. J., "A Study of Viscous Flux Formulations for an Implicit *P*-Multigrid Spectral Volume Navier Stokes Solver," 46th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2008-783, Reno, NV, Jan. 2008.

[16] Breviglieri, C., Basso, E., and Azevedo, J. L. F., "High-Order Unstructured Spectral Finite Volume Scheme for Aerodynamic Applications," 26th AIAA Applied Aerodynamics Conference, AIAA Paper 2008-7182, Honolulu, HI, Aug. 2008.

[17] Yang, M., and Wang, Z. J., "A Parameter-Free Generalized Moment Limiter for High-Order Methods on Unstructured Grids," *Advances in Applied Mathematics and Mechanics*, Vol. 1, No. 4, 2009, pp. 451–480.
doi:10.4208/aamm.09-m0913

[18] Cuthill, E., and McKee, J., "Reducing the Bandwidth of Sparse Symmetric Matrices," *Proceedings of the 24th National Conference of the ACM*, Brandon Systems, New York, 1969.

[19] Chen, Q. Y., *Partitions for Spectral (Finite) Volume Reconstruction in the Tetrahedron*, Preprint Series, No. 2035, Inst. for Mathematics and Its Applications, Univ. of Minnesota, Minneapolis, MN, April 2005.

[20] van den Abeele, K., and Lacor, C., "An Accuracy and Stability Study of the 2D Spectral Volume Method," *Journal of Computational Physics*, Vol. 226, No. 1, Sept. 2007, pp. 1007–1026.
doi:10.1016/j.jcp.2007.05.004

[21] Knuth, D. E., *The Art of Computer Programming. 3: Sorting and Searching*, 2nd ed., Addison-Wesley, Reading, MA, 1998.

[22] Liu, Y., and Vinokur, M., "Exact Integrations of Polynomials and Symmetric Quadrature Formulas over Arbitrary Polyhedral Grids," *Journal of Computational Physics*, Vol. 140, No. 1, Feb. 1998, pp. 122–147.
doi:10.1006/jcph.1998.5884

[23] Zienkiewicz, O., and Taylor, R. L., *The Finite Element Method*, 5th ed., Butterworth-Heinemann, Oxford, 2000.

[24] IGES Ver. 6.0, Software Package, NIST Standards Services Div., Gaithersburg, MD, 1995.

[25] van Altena, M., "High-Order Finite-Volume Discretisations for Solving a Modified Advection-Diffusion Problem on Unstructured Triangular Meshes," M.S. Thesis, Univ. of British Columbia, Vancouver, VC, Canada, 1999.

[26] Xu, Z., Liu, Y., and Shu, C. W., "Hierarchical Reconstruction for Spectral Volume Method on Unstructured Grids," *Journal of Computational Physics*, Vol. 228, No. 16, Sept. 2009, pp. 5787–5802.
doi:10.1016/j.jcp.2009.05.001

[27] Qiu, J., and Shu, C.-W., "A Comparison of Troubled-Cell Indicators for Runge–Kutta Discontinuous Galerkin Methods Using Weighted Essentially Nonoscillatory Limiters," *SIAM Journal on Scientific Computing*, Vol. 27, No. 3, 2005, pp. 995–1013.
doi:10.1137/04061372X

[28] Bigarella, E., "Advanced Turbulence Modelling for Complex Aerospace Applications," Ph.D. Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil, 2007.

[29] Jiang, G. S., and Shu, C. W., "Efficient Implementation of Weighted ENO Schemes," *Journal of Computational Physics*, Vol. 126, No. 1, June 1996, pp. 77–99.
doi:10.1006/jcph.1996.0121

[30] Friedrich, O., "Weighted Essentially Non-Oscillatory Schemes for the Interpolation of Mean Values on Unstructured Grids," *Journal of Computational Physics*, Vol. 144, No. 1, July 1998, pp. 194–212.
doi:10.1006/jcph.1998.5988

[31] Anderson, J. D., *Modern Compressible Flow, with Historical Perspective*, 2nd ed., McGraw–Hill, Boston, 1982.

[32] "Equations, Tables, and Charts for Compressible Flow," NACA Ames Research Center, Rept. 1135, 1953.

[33] Shapiro, A. H., *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Wiley, New York, 1953.

[34] Wang, Z. J., and Liu, Y., "Extension of the Spectral Volume Method to High-Order Boundary Representation," *Journal of Computational Physics*, Vol. 211, No. 1, Jan. 2006, pp. 154–178.
doi:10.1016/j.jcp.2005.05.022

[35] Cook, P. H., McDonald, M. A., and Firmin, M. C. P., "Aerofoil RAE 2822—Pressure Distributions and Boundary Layer and Wake Measurements," *Experimental Data Base for Computer Program Assessment*, AGARD Rept. AR 138, Neuilly-sur-Seine, France, 1979.

[36] Michalak, K., and Ollivier-Gooch, C., "Limiters for Unstructured Higher-Order Accurate Solutions of the Euler Equations," 46th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2008-0776, Reno, NV, Jan. 2008.

Z. Wang
Associate Editor