A HIGH ORDER ACCURATE UNSTRUCTURED SPECTRAL FINITE VOLUME IMPLICIT ALGORITHM FOR INVISCID COMPRESSIBLE FLOWS

Carlos Breviglieri, carbrevi@yahoo.com.br
Instituto Tecnológico de Aeronáutica, CTA/ITA/IEC, 12228-900, São José dos Campos, SP, Brazil

João Luiz F. Azevedo, azevedo@iae.cta.br
Edson Basso, basso@iae.cta.br
Instituto de Aeronáutica e Espaço, CTA/IAE/ALA, 12228-903, São José dos Campos, SP, Brazil

Abstract. *The purpose of this work is to develop a methodology that achieves high order spatial discretization for compressible aerodynamic flows based on the spectral finite volume method for hyperbolic conservation laws. High order methods are necessary on the analysis of complex flows to reduce the number of mesh elements one would otherwise need if using traditional second-order schemes. In other words, high order methods can potentially achieve a higher level of accuracy than low order ones given the same computational resources. The spectral finite volume method was developed as an alternative to k-exact high order schemes, ENO/WENO and discontinuous Galerkin methods. Its main objective is to allow the implementation of a simpler and more efficient scheme, while still achieving high order spatial accuracy. The 2-D Euler equations are solved numerically in a finite volume, cell centered context on unstructured meshes. An implicit time march algorithm is employed to advance the solution to steady-state. The treatment of discontinuities is also discussed. Several applications are performed in order to assess the method capability, which is compared to data available in the literature and also compared to results from an weighted essentially non-oscillatory (WENO) scheme. The latter comparison data can also be used to assess the present method computational performance.*

Keywords: *Spectral Finite Volume, Implicit Method, High Order Discretization, 2D Euler Equations, Unstructured Meshes*

1. INTRODUCTION

Over the past several years, the Computational Aerodynamics Laboratory of Instituto de Aeronáutica e Espaço (IAE) has been developing CFD solvers for two and three dimensional systems [Scalabrin, 2002, Basso et al., 2000]. One research area of the development effort is aimed at the implementation of high-order methods suitable for problems of interest to the Institute, i.e., external high-speed aerodynamics. Some upwind schemes such as the van Leer flux vector splitting scheme [van Leer, 1982], the Liou AUSM+ flux vector splitting scheme [Liou, 1996] and the Roe flux difference splitting scheme [Roe, 1981] were implemented and tested for second-order accuracy with a MUSCL reconstruction [Anderson et al., 1986]. However, the nominally second-order schemes presented results with an order of accuracy smaller than the expected in the solutions for unstructured grids. 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 non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes in the past [Wolf and Azevedo, 2006]. 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 three, in 2-D, or anything greater than 2nd order, in 3-D. This fact motivated the consideration of the spectral finite volume method (SFV), as proposed by Wang and co-workers [Wang, 2002, Wang and Liu, 2002, Wang and Liu, 2003, Wang et al., 2004, Liu et al., 2006, Sun et al., 2006], 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 currently extended up to 4th-order accuracy in the present work. The SFV method is already in use by the authors and, previously, numerical results have been published [Breviglieri et al., 2008]. Although the expect order of accuracy is obtained for the 2nd, 3rd and 4th order SFV methods, a significant deterioration in convergence rate for the fourth order simulations is observed, especially in the presence of shock waves. Hence, it is expected that such behaviour can be overcome by the use of an implicit time march algorithm.

The numerical solver is currently implemented for the solution of the 2-D Euler equations in a cell centered finite volume context for triangular meshes, with a lower upper symmetric Gauss-Seidel (LU-SGS) scheme for time integration. The paper, as here organized, presents the theoretical formulation of the spatial and temporal discretization methods for the Euler equations. The reconstruction process of the high-order polynomial is described next. Afterwards, the flux limiting formulation is presented followed by the numerical results and conclusions.

2. THEORETICAL FORMULATION

2.1 Governing Equations

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

$$\frac{\partial}{\partial t}\int_V QdV + \int_V (\nabla \cdot \vec{P})dV = 0\,, \tag{1}$$

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

$$\frac{\partial}{\partial t}\int_V QdV + \int_S (\vec{P}\cdot\vec{n})dS = 0\,. \tag{2}$$

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

$$Q = \left\{ \begin{array}{c} \rho \\ \rho u \\ \rho v \\ e_i \end{array} \right\}\,, \quad E = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e_i + p)u \end{array} \right\}\,, \quad F = \left\{ \begin{array}{c} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e_i + p)v \end{array} \right\}\,. \tag{3}$$

The system is closed by the equation of state for a perfect gas

$$p = (\gamma - 1)\left[e_i - \frac{1}{2}\rho(u^2 + v^2)\right], \tag{4}$$

where the ratio of specific heats, γ, is set as 1.4 for all computations in this work. The flux Jacobian matrix in the $\vec{n} = (n_x, n_y)$ face-normal direction can be written as

$$B = n_x\frac{\partial E}{\partial Q} + n_y\frac{\partial F}{\partial Q}. \tag{5}$$

The B matrix has four real eigenvalues $\lambda_1 = \lambda_2 = v_n, \lambda_3 = v_n + a, \lambda_4 = v_n - a$, and a complete set of right eigenvectors (r_1, r_2, r_3, r_4), where $v_n = un_x + vn_y$ and a is the speed of sound. Let R be the matrix composed of these right eigenvectors, then the Jacobian matrix, B, can be diagonalized as

$$R^{-1}BR = \Lambda, \tag{6}$$

where Λ is the diagonal matrix containing the eigenvalues,

$$\Lambda = diag(v_n, v_n, v_n + a, v_n - a). \tag{7}$$

In the finite volume context, Eq. (2) can be rewritten for the i-th control volume as

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{V_i}\int_{S_i} (\vec{P}\cdot\vec{n})dS\,, \tag{8}$$

where Q_i is the cell averaged value of Q at time t and V_i is the volume, or area in 2-D, on the i-th control volume.

2.2 Spatial Discretization

The spatial discretization process determines a k-th order discrete approximation to the integral in the right-hand side of Eq. (8). In order to solve it numerically, the computational domain, Ω, with proper initial and boundary conditions, is discretized into N non-overlapping triangles, the spectral volumes (SVs), such that

$$\Omega = \bigcup_{i=1}^{N} S_i. \tag{9}$$

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

$$S_i = \bigcup \Gamma_K\,. \tag{10}$$

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.

The boundary integral in Eq. (8) can be further discretized into the convective operator form

$$C(Q_i) \equiv \int_{S_i} (\vec{P} \cdot \vec{n}) dS = \sum_{r=1}^{K} \int_{A_r} (\vec{P} \cdot \vec{n}) dS, \tag{11}$$

where K is the number of faces, or edges in 2-D, of S_i, and A_r represents the $r-th$ edge of the SV. Given the fact that $\vec{n}$ is constant for each line segment, the integration on the right side of Eq. (11) can be performed numerically with a $k-th$ order accurate Gaussian quadrature formula

$$\int_{A_r} (\vec{P} \cdot \vec{n}) dS = \sum_{r=1}^{K} \sum_{q=1}^{J} w_{rq} \vec{P}(Q(x_{rq}, y_{rq})) \cdot \vec{n}_r A_r + O(A_r h^k). \tag{12}$$

where (x_{rq}, y_{rq}) and w_{rq} are, respectively, the Gaussian points and the weights on the r-th edge of S_i, $J = integer((k+1)/2)$ is the number of quadrature points required on the $r-th$ edge, and h will be defined in the forthcoming discussion. For the second-order schemes, one Gaussian 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 Gaussian point as the middle point of the segment connecting the two end points, $G_1 = \frac{1}{2}(z_1 + z_2)$. For this case, the weight is $w_1 = 1$. For the third and fourth order schemes, two Gaussian points are necessary along each line segment. Their values are given by

$$G_1 = \frac{\sqrt{3}+1}{2\sqrt{3}} z_1 + (1 - \frac{\sqrt{3}+1}{2\sqrt{3}}) z_2 \quad \text{and} \quad G_2 = \frac{\sqrt{3}+1}{2\sqrt{3}} z_2 + (1 - \frac{\sqrt{3}+1}{2\sqrt{3}}) z_1, \tag{13}$$

and the respective weights, w_1 and w_2, are set as $w_1 = w_2 = \frac{1}{2}$. Using the method described above, one can compute values of Q_i for instant t for each SV. From these averaged values, it is possible to reconstruct polynomials that represent the conserved variables, ρ, ρu, ρv and e. Due to 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 on the cell boundaries.

The above procedures follow exactly the standard finite volume method. For a given order of spatial accuracy, k, for Eq. (11), using the SFV method, each S_i element must have at least

$$m = \frac{k(k+1)}{2} \tag{14}$$

degrees of freedom (DOFs). This corresponds to the number of control volumes that S_i shall be partitioned into. If one denotes by $C_{i,j}$ the j-th control volume of S_i, the cell-averaged conservative variables, $\overline{q}$, at time t, for $C_{i,j}$ is computed as

$$\overline{q}_{i,j}(t) = \frac{1}{V_{i,j}} \int_{C_{i,j}} q(x, y, t) dx dy, \tag{15}$$

where $V_{i,j}$ is the volume of $C_{i,j}$. Once the cell-averaged conservative variables, or DOFs, are available for all CVs within S_i, a polynomial, $p_i(x, y) \in P^{k-1}$, with degree $k-1$, can be reconstructed to approximate the $q(x, y)$ function inside S_i, i.e.,

$$p_i(x, y) = q(x, y) + O(h^{k-1}), \qquad (x, y) \in S_i, \tag{16}$$

where h represents the maximum edge length of all CVs within S_i. The polynomial reconstruction process is discussed in details in the following section. For now, it is enough 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 S_i and some numerical flux coupling is necessary across SV boundaries.

Integrating Eq. (8) in $C_{i,j}$, one can obtain the integral form for the CV averaged mean state variable

$$\frac{d\overline{q}_{i,j}}{dt} + \frac{1}{V_{i,j}} \sum_{r=1}^{K} \int_{A_r} (f \cdot \vec{n}) dS = 0, \tag{17}$$

where f represents the E and F fluxes, K is the number of edges of $C_{i,j}$ and A_r represents the $r-th$ edge of the CV. The numerical integration can be performed with a $k-th$ order accurate Gaussian quadrature formulation, similarly to the one for the SV elements in Eq. (12).

As previously stated, the flux integration across SV boundaries involves two discontinuous states, to the left and to the right of the edge. This flux computation can be carried out using an exact or approximate Riemann solver, or even a flux splitting procedure, which can be written in the form

$$f(q(x_{rq}, y_{rq})) \cdot \vec{n}_r \approx f_{Riemann}(q_L(x_{rq}, y_{rq}), q_R(x_{rq}, y_{rq}), \vec{n}_r), \tag{18}$$

where q_L is the conservative 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 which introduces the upwind and artificial dissipation effects into the method, making it stable and accurate. In this work, the authors have used the Roe flux difference splitting method [Roe, 1981] to compute the numerical flux, i.e.,

$$f_{Riemann} = f_{roe}(q_L, q_R, \vec{n}) = \frac{1}{2}\left[f(q_L) + f(q_R) - \left|\overline{B}\right|(q_R - q_L)\right], \tag{19}$$

where $\left|\overline{B}\right|$ is Roe's dissipation matrix computed in the direction normal to the edges as

$$\left|\overline{B}\right| = R\left|\overline{\Lambda}\right|R^{-1}. \tag{20}$$

Here, $\left|\overline{\Lambda}\right|$ is the diagonal matrix composed of the absolute values of the eigenvalues of the Jacobian matrix, as defined in Eq. (7), evaluated using the Roe averages.

Finally, one ends up with the semi-discrete SFV scheme for updating the DOFs at control volumes, which can be written as

$$\frac{d\overline{q}_{i,j}}{dt} = -\frac{1}{V_{i,j}}\sum_{r=1}^{K}\sum_{q=1}^{J} w_{rq} f_{Riemann}(q_L(x_{rq}, y_{rq}), q_R(x_{rq}, y_{rq}), \vec{n}_r) A_r. \tag{21}$$

where the right hand side of Eq. (21) is the equivalent convective operator, $C(q_{i,j})$, for the j-th control volume of S_i. It is worth mentioning 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.

2.3 Temporal Discretization

The convergence behavior of high-order methods, such as the SFV method, is generally poor with explicit time marching approaches. In order to obtain the steady state solution of the flow from an initial condition, a relaxation scheme is necessary. The approach typically used in the present research group has been to resort to explicit, multi-stage, 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. Hence, the current "production" version of the code uses a 3-stage TVD Runge-Kutta scheme for time integration [Wolf and Azevedo, 2006]. However, adequate solution convergence characteristics, especially for the higher-order implementations, dictate that an implicit time integrator should be implemented. Therefore, an implicit LU-SGS scheme is implemented in the context of the present work.

Equation (2) can be recast in the semi-discrete form as

$$V_i \frac{\partial q_i}{\partial t} = -R_i \tag{22}$$

where R_i is the right-hand side residual and it tends to zero as the simulation converges to a steady-state solution. Using Euler implicit time-integration, Eq. (22) can be written in discrete form as

$$V_i \frac{\delta q_i^n}{\Delta t} = -R_i^{n+1} \tag{23}$$

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

$$V_i \frac{\delta q_i^n}{\Delta t} = -R_i^n - \frac{\partial R_i^n}{\partial q}\delta q_i^n. \tag{24}$$

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

$$A\delta q = R \tag{25}$$

where

$$A = \frac{V}{\Delta t}\mathbf{I} + \frac{\partial R^n}{\partial q} \tag{26}$$

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

In order to reduce the number of non-zero entries in the Jacobian matrix and to simplify the linearization process, only a first-order representation of the numerical fluxes is linearized. This results in the fact that the graph of the sparse matrix is identical to the graph of the unstructured mesh. Hence, the Jacobian matrix entries can be computed and stored over a loop on the mesh edges. Therefore, the residual operator can be written as

$$R_i(q_i, q_j, \vec{n}_{ij}) = \frac{1}{2}\left[f(q_i, \vec{n}_{ij}) + f(q_j, \vec{n}_{ij}) - |\lambda_{ij}|(q_j - q_i)\right] \tag{27}$$

for an edge that shares volume i and j. A scalar dissipation model is used,

$$|\lambda_{ij}| = |\vec{v}_{ij} \cdot \vec{n}_{ij}| + a_{ij} \tag{28}$$

where $\vec{n}_{ij}$ is the unit vector normal to the edge, $\vec{v}_{ij}$ is the velocity vector normal to the edge and a is the speed of sound. One should note that the dissipation on the flux function is approximated by the Jacobian matrix spectral radius. The linearization of Eq. (27) yields

$$\begin{aligned}
\frac{\partial R_i}{\partial q_i} &= \frac{1}{2}\left(J(q_i) + |\lambda_{ij}|\mathbf{I}\right) \\
\frac{\partial R_i}{\partial q_j} &= \frac{1}{2}\left(J(q_j) - |\lambda_{ij}|\mathbf{I}\right)
\end{aligned} \tag{29}$$

where $J = \partial F/\partial q$ is the Jacobian of the inviscid flux vector.

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

$$\begin{aligned}
L &= \frac{1}{2}\left[-J(q_i, \vec{n}_{ij}) - |\lambda_{ij}|\mathbf{I}\right] \\
U &= \frac{1}{2}\left[J(q_j, \vec{n}_{ij}) - |\lambda_{ij}|\mathbf{I}\right] \\
D &= \frac{V}{\Delta t}\mathbf{I} + \sum_j \frac{1}{2}\left[J(q_i, \vec{n}_{ij}) + |\lambda_{ij}|\mathbf{I}\right].
\end{aligned} \tag{30}$$

Note that L, U and D represent the strict lower, upper and diagonal matrices, respectively. Equation (25) 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 [Cuthill and McKee, 1969], and the system is solved in two steps, a forward and backward sweep

$$\begin{aligned}
(D + L)\delta q^* &= R \\
(D + U)\delta q &= D\delta q^*.
\end{aligned} \tag{31}$$

It is found that the CPU cost of one LU-SGS step is very close, if not cheaper, than that of the 3-stage Runge Kutta explicit step for these inviscid analyses.

3. SPECTRAL FINITE VOLUME RECONSTRUCTION

3.1 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 edges. These evaluations can be achieved by reconstructing conserved variables in terms of some base functions using the DOFs within a SV. The present work has carried out such reconstructions using polynomial functions. Let P_m denote the space of m-th degree polynomials in two dimensions. Then, the minimum dimension of the approximation space that allows P_m to be complete is

$$N_m = \frac{(m+1)(m+2)}{2}. \tag{32}$$

In order to reconstruct q in P_m, it is necessary to partition the SV into N_m non-overlapping CVs, such that

$$S_i = \bigcup_{j=1}^{N_m} C_{i,j}. \tag{33}$$

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

$$\int_{C_{i,j}} p_m(x,y)dS = \int_{C_{i,j}} q(x,y)dS. \tag{34}$$

With a complete polynomial basis, $e_l(x,y) \in P_m$, it is possible to satisfy Eq. (34). Hence, p_m can be expressed as

$$p_m = \sum_{l=1}^{N_m} b_l e_l(x,y), \tag{35}$$

where e is the base function vector, $[e_1, \cdots, e_N]$, and b is the reconstruction coefficient vector, $[b_1, \cdots, b_N]^T$. The substitution of Eq. (35) into Eq. (34) yields

$$\frac{1}{V_{i,j}} \sum_{l=1}^{N_m} b_l \int_{C_{i,j}} e_l(x,y)dS = \overline{q}_{i,j}. \tag{36}$$

If $\overline{q}$ denotes the $[\overline{q}_{i,1}, \cdots, \overline{q}_{i,Nm}]^T$ column vector, Eq. (36) can be rewritten in matrix form as

$$Sb = \overline{q}, \tag{37}$$

where the S reconstruction matrix is given by

$$S = \begin{bmatrix} \frac{1}{V_{i,1}} \int_{C_{i,1}} e_1(x,y)dS & \cdots & \frac{1}{V_{i,1}} \int_{C_{i,1}} e_N(x,y)dS \\ \vdots & \cdots & \vdots \\ \frac{1}{V_{i,N}} \int_{C_{i,N}} e_1(x,y)dS & \cdots & \frac{1}{V_{i,N}} \int_{C_{i,N}} e_N(x,y)dS \end{bmatrix} \tag{38}$$

and, then, the reconstruction coefficients b can be obtained as

$$b = S^{-1}\overline{q}, \tag{39}$$

provided that S is non-singular. With the substitution of Eq. (39) into Eq. (34), p_m is, then, expressed in terms of shape functions $L = [L_1, \cdots, L_N]$, defined as $L = eS^{-1}$, such that one could write

$$p_m = \sum_{j=1}^{N_m} L_j(x,y)\overline{q}_{i,j} = L\overline{q}. \tag{40}$$

Equation (40) 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}).

The major advantage of the SFV method is that the reconstruction process does not need to be carried out for every mesh element S_i. One can compute these coefficients as a pre-processing step and they do not change along the simulation. This is a major difference when compared to methods such as ENO and WENO, for which every mesh element has a different reconstruction process at each time step. The polynomial base functions for the linear, quadratic and cubic reconstructions are listed in Table 1. Clearly, the linear, quadratic and cubic polynomial reconstructions will yield, respectively, 2nd-, 3rd- and 4th-order spatial discretization numerical schemes.

3.2 Linear Reconstruction

For the linear SFV method reconstruction, $m = 1$, one needs to partition a SV in three sub-elements, as in Eqs. (14) and (32) and use the base vector as defined in Table 1. The partition scheme is given for a standard element. The partition for this case is uniquely defined. 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 [Knuth, 1998].

The linear partition is presented in Fig. 1-a. It yields a total of 7 points, 9 edges (6 are external edges and 3 are internal ones) and 9 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. (38) are obtained analytically [Liu and Vinokur, 1998] for fundamental shapes. The shape functions, in the sense of Eq. (40), are calculated and stored in memory for the quadrature points, (x_{rq}, y_{rq}), of the standard element. Such shape functions have the exact same value for the quadratures points of any other SV of the mesh, provided they all have the same partition. There is one quadrature point located at the middle of the every CV edge.

Table 1. Polynomial base functions.

Reconstruction Order	e
linear	$[\,1\ x\ y\,]$
quadratic	$[\,1\ x\ y\ x^2\ xy\ y^2\,]$
cubic	$[\,1\ x\ y\ x^2\ xy\ y^2\ x^3\ x^2y\ xy^2\ y^3\,]$

3.3 Quadratic Reconstruction

For the quadratic reconstruction, $m = 2$, one needs to partition a SV in six sub-elements and use the base vector as defined in Table 1. The partition scheme is also given in this work for a right 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 this work [van den Abeele and Lacor, 2007] is shown in Fig. 1-b. It has a total of 13 points, 18 edges (9 external edges and 9 internal ones), 36 quadrature points and it has a Lebesgue constant value of 3.075. The shape functions, in the sense of Eq. (40), 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 and kept in memory [Liu and Vinokur, 1998]. In this case, two quadrature points are required per CV edge.

3.4 Cubic Reconstruction

For the cubic reconstruction, $m = 3$, one needs to partition the SV in ten sub-elements 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 utilized to perform these tasks can be applied to higher order reconstructions. The partition used in this work is the improved cubic partition [van den Abeele and Lacor, 2007], presented in Fig. 1-c and it has a total of 21 points, 30 edges (12 external edges and 18 internal ones), 60 quadrature points and it has a Lebesgue constant value of 4.2446. The shape functions, in the sense of Eq. (40), are obtained as in the linear partition in a pre-processing step. As with the quadratic reconstruction, each CV edge has two quadrature points [Breviglieri et al., 2008].

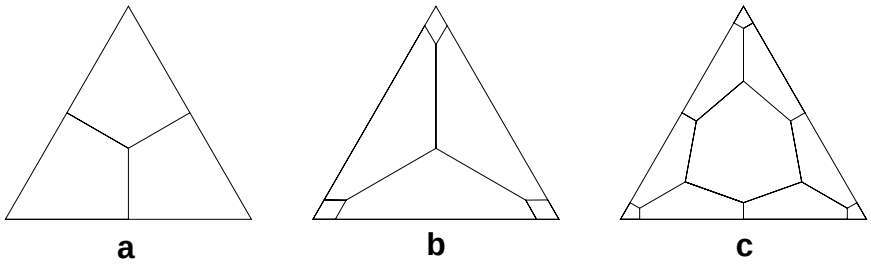


Figure 1. Triangular spectral volume partitions for (a) linear, (b) quadratic, and (c) cubic reconstructions.

4. LIMITER FORMULATION

For the Euler equations, it is necessary to limit some reconstructed properties in order to maintain stability and convergence of the simulation, if the resulting flowfield contains discontinuities. The limiters are applied in each component of the primitive variable vector $(\rho, u, v, p)^T$, derived from the conserved variable vector evaluated at quadrature points. For each CV, the following numerical monotonicity criterion is prescribed:

$$\overline{q}_{i,j}^{min} \leq q_{i,j}(x_{rq}, y_{rq}) \leq \overline{q}_{i,j}^{max}, \tag{41}$$

where $\overline{q}_{i,j}^{min}$ and $\overline{q}_{i,j}^{max}$ are the minimum and maximum cell averaged property values among all neighboring CVs that share a edge with $C_{i,j}$. If Eq. (41) is strictly enforced, the method becomes TVD [Leveque, 2002]. This method, however, is first-order accurate and it may compromise the general accuracy of the solution. To maintain high-order accuracy away from discontinuities, small oscillations are allowed in the simulation, as in TVB methods [Shu, 1987]. If one expresses

the reconstruction for the quadrature points as a difference with respect to the cell averaged mean,

$$\Delta q_{rq} = p_i(x_{rq}, y_{rq}) - \overline{q}_{i,j}, \tag{42}$$

then no limiting is necessary if $|\Delta q_{rq}|$ satisfies Eq. (41) for every quadrature point of the CV edges. If it does not, then, the solution is limited for this CV and linearly reconstructed as

$$q_{i,j}(x, y) = \overline{q}_{i,j} + \Phi \nabla q_{i,j} \cdot \mathbf{r}, \tag{43}$$

where $\nabla q_{i,j}$ is the gradient at the CV centroid, and $\mathbf{r}$ is the position vector of the quadrature point with regard to the CV centroid. The original high order polynomial in the CV is used to compute the gradient, i.e.,

$$\nabla q_{i,j} = \left(\frac{\partial p_i}{\partial x}, \frac{\partial p_i}{\partial y} \right). \tag{44}$$

The reconstructed property value from Eq. (43) may not satisfy Eq.(41) and, therefore, it is limited by multiplying the increment in the CV average value by a scalar $\Phi \in [0, 1]$, that can be computed following the general orientation of the literature, such that it satisfies the monotonicity constraint. In this work, the $superbee$ limiter is used [Hirsch, 1990].

5. NUMERICAL 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 WENO scheme implementation. For the presented results, density is made dimensionless with respect to the free stream condition and pressure is made dimensionless with respect to the density times the speed of sound squared. For the steady case simulations, the CFL number is set as a constant value and the local time step is computed using the local grid spacing and characteristic speeds. For both test cases, the CFL number is set to $1.0e + 6$.

All numerical simulations were 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[1] is used. For all performance comparisons which 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.

5.1 Wedge Flow

The computation of the supersonic flow field past a wedge with half-angle $\theta = 10$ deg is considered. The computational mesh has 816 nodes and 1504 volumes and it is shown in Fig. 2, along with the density contours obtained with 4th-order SFV method. For comparison purposes, the second, third and fourth order SFV methods were utilized along with WENO schemes. The leading edge of the wedge is located at coordinates $x = 0.25$ and $y = 0.0$. The computational domain is bounded along the bottom by the wedge surface and by an outflow section before the leading edge. The inflow boundary is located at the left and top of the domain, while the outflow boundary is located ahead of the wedge and at the right of the domain. The analytical solution gives the change in properties across the oblique shock as a function of the free stream Mach number and shock angle, which is obtained from the $\theta - \beta - Mach$ relation. For this case, a free stream Mach number of $M_1 = 5.0$ was used, and the oblique shock angle β is obtained as 19.5 deg. For the analytical solution, the pressure ratio is $p_2/p_1 \approx 3.083$ and the Mach number past the shock wave is $M_2 \approx 3.939$. For these simulations the use of the limiter was necessary in order to keep the high order reconstruction away from the shock wave.

The numerical solutions of the SFV method are in good agreement with the analytical solution. In Fig. 3 we compare the numerical solutions of the SFV and WENO schemes, in terms of pressure coefficient values, with the analytical one. Note that the SFV scheme is the one that better approximates the jump in pressure on the leading edge. The pressure ratio and Mach number after the shock wave for the fourth order SFV scheme were computed as $p_2/p_1 \approx 3.047$ and $M_2 \approx 3.901$. As expected, the fourth order SFV scheme achieved results closer to the analytical one. Also, the Cp results for the second and third order WENO solutions are very similar. On the other hand, only the SFV method achieved a solution, for the fourth-order methods, because the 4th-order WENO scheme diverged.

5.2 NACA 0012 Airfoil

For the NACA 0012 airfoil simulation, the mesh is shown in Fig. 4, along with the Mach number contours. The mesh has 8414 elements and 4369 nodes. Flow conditions are the same as in the experimental data [McDevitt and Okuno, 1985], that is, freestream Mach number of $M_\infty = 0.8$ and 0 deg. angle-of-attack. Figure 5 shows the Cp plots of the numerical simulations for both WENO and SFV methods of different orders. Their agreement with the experimental data, in terms of

[1] Compiler flags: -O3 -assume buffered_io -parallel

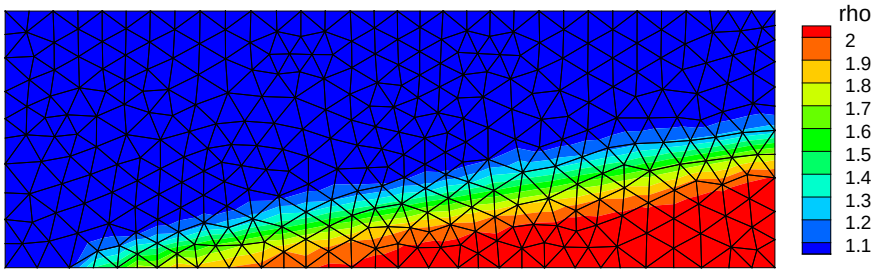


Figure 2. Supersonic wedge flow unstructured mesh with density distribution obtained with 4th order SFV.

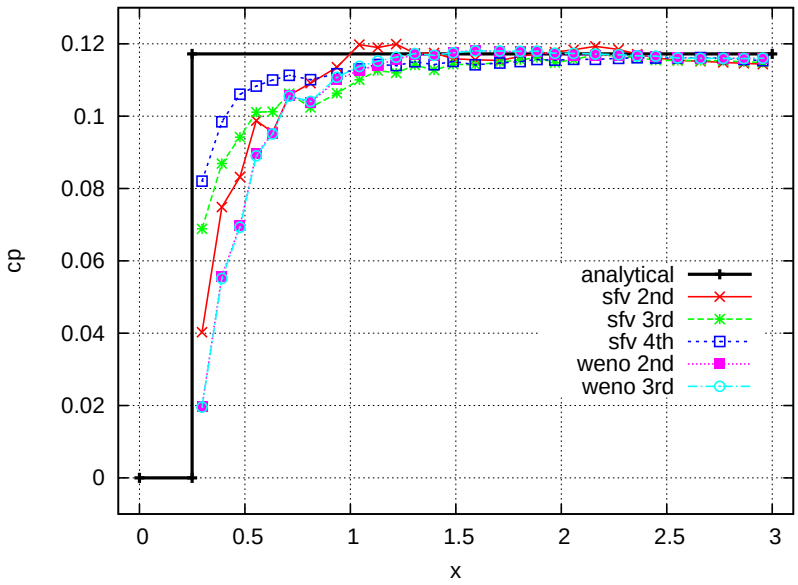


Figure 3. Supersonic wedge flow analytical and numeric pressure coefficient.

shock position and pressure coefficient (Cp) values, is very reasonable. The main difference between the methods occurs for the fourth order simulation where the SFV method better approaches the experimental data and gives more consistent values for Cp after the shock wave. For these simulations, the use of limiters is also necessary.

The Cp curves indicate that the SFV method captures the shock wave, over the airfoil, usually with a single SV element in it, as shown in Fig. 6, which validates the limiter formulation and the suitability of the method to our needs. The Cp distributions in the post-shock region show that the influence of the limiter operator reduced the fourth order scheme resolution. Also, the fourth order simulation of the WENO scheme presented large oscillations along the airfoil chord, as seen in Fig. 6(c). This can be explained given the fact that the cubic polynomial reconstruction process of the WENO scheme, which involves neighbor data processing, is limited near boundaries and it must work with the available data. Hence, it produces an oscillatory interpolation polynomial. It is important to emphasize that the present computations are performed assuming inviscid flow. One should observe, however, that the pressure rise across the shock wave, in the experimental results, is spread over a larger region due to the presence of the boundary layer and the consequent shock-boundary layer interaction that necessarily occurs in the experiment. For the numerical solutions, the shock presents a sharper resolution, as one can expect from an Euler solution.

The performance analysis is carried out for this test. The time for solution of the SFV 3rd order implicit and explicit methods can be seen in Fig. 7 along with the number of iterations. For the explicit run, a CFL value of 0.2 was used. The total iteration number is limited to thirty thousand iterations. Despite the relative low residual drop, for the explicit simulation, the lift coefficient of the airfoil reached a steady value at about 10000 iterations. Next, a comparison of the implicit WENO and SFV schemes is presented for the third order spatial resolution case. As one can observe in Fig. 8, the SFV method is able to reduce the residual several orders, whereas the WENO case seems to stall the convergence. Nevertheless, for both schemes, the lift coefficient plot shows a constant, or zero value for this case, after the first 500 iterations on the SFV scheme and after 1000 iterations for the WENO scheme.

The performance achieved on the third order case is carried over to the fourth order simulation as numerical experimentation showed. However, as should expect, for flows with discontinuities such as the present test case, there is a small performance degradation for the 4th-order scheme due to the limitation process. There are more limited control volumes, which increases the overall cost of the method, since these elements must be linearly reconstructed and, then, limited.

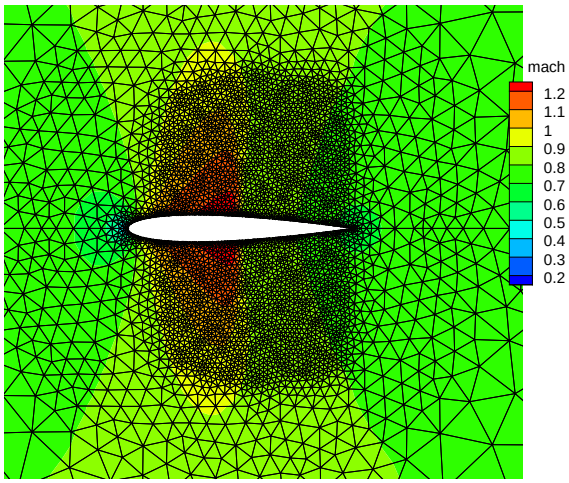


Figure 4. Mesh and Mach number visualization for 4th-order SFV numerical simulation of flow over NACA 0012 airfoil ($M_\infty = 0.8$).

6. CONCLUSIONS

The second, third and fourth order spectral finite volume methods are successfully implemented and validated with the proposed numerical tests. The method behavior for resolution orders greater than second order is shown to be in good agreement with both experimental and analytical data. Furthermore, the results obtained show that the current method can yield solutions with the same or better quality, at a much lower computational resource usage, than other high order schemes, as demonstrated by the comparison with computations performed with a WENO scheme. Further improvements in the SFV method capabilities are achieved by addition of an implicit time march algorithm.

The method seems suitable for the aerospace applications of interest to IAE in the sense that it is compact, given the fact that the stencil for polynomial reconstruction is always known, geometrically flexible, by supporting unstructured meshes, and computationally efficient.

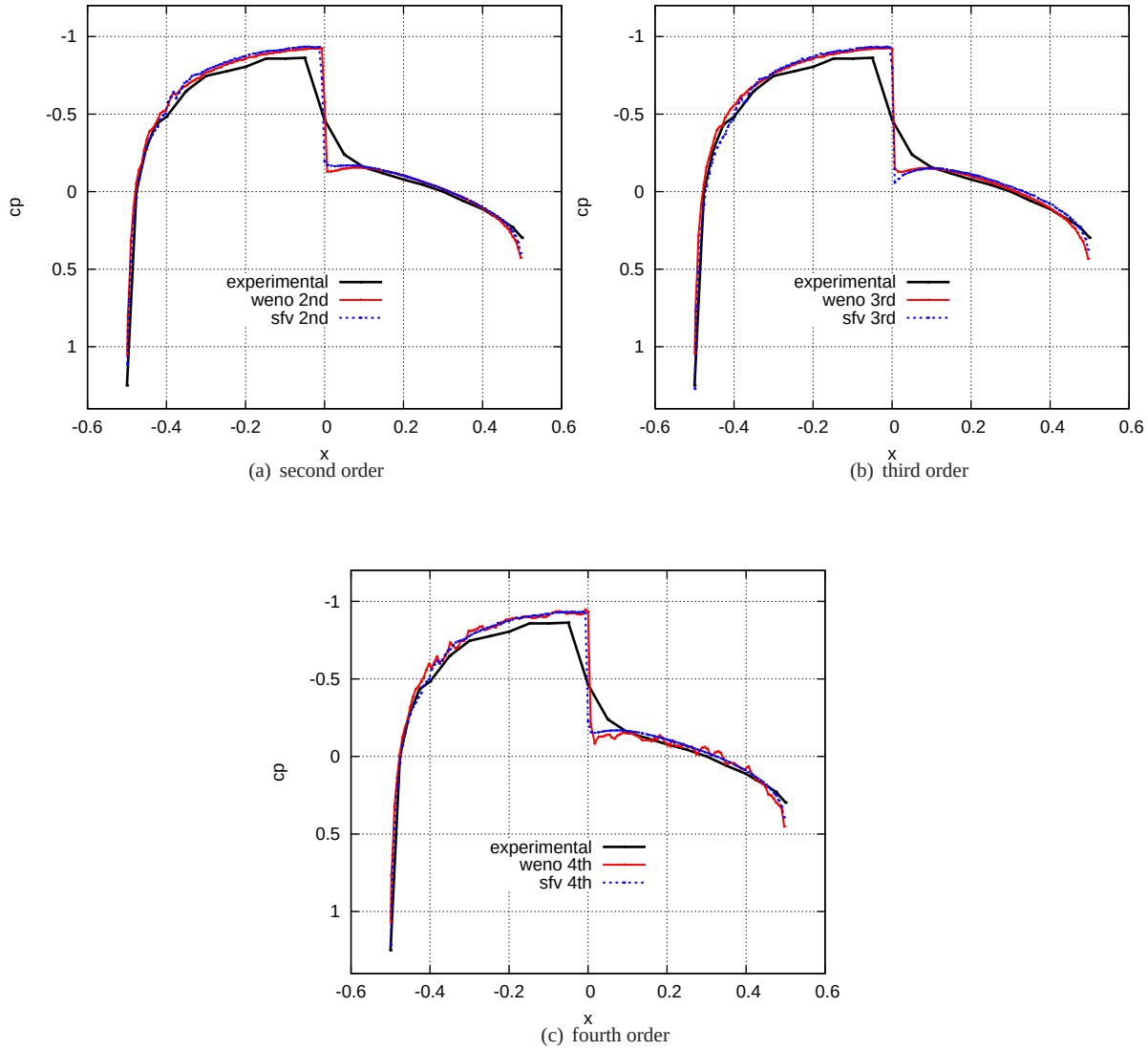
7. ACKNOWLEDGMENTS

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.

8. REFERENCES

Anderson, W. K., Thomas, J. L., and van Leer, B., 1986, Comparison of Finite Volume Flux Vector Splittings for the Euler Equations, “AIAA Journal”, Vol. 24, No. 9, pp. 1453–1460.

Basso, E., Antunes, A. P., and Azevedo, J. L. F., 2000, Three Dimensional Flow Simulations Over a Complete Satellite Launcher with a Cluster Configuration, “Proceedings of the 18th AIAA Applied Aerodynamics Conference and Exhibit”, pp. 805–813, Denver, CO. AIAA Paper No. 2008-7182.



(a) second order (b) third order

(c) fourth order

Figure 5. Experimental and numerical Cp distribution for NACA 0012 airfoil.

Breviglieri, C., Basso, E., and Azevedo, J. L. F., 2008, High-Order Unstructured Spectral Finite Volume Scheme for Aerodynamic Applications, “26th AIAA Applied Aerodynamics Conference”, Honolulu, HI. AIAA Paper No. 2008-7182.

Cuthill and McKee, J., 1969, Reducing the Bandwidth of Sparse Symmetric Matrices, “Proceedings of the 24th National Conference of the ACM”. Brandon Systems Press.

Hirsch, C., 1990, “Numerical Computation of Internal and External Flows”, John Wiley and Sons, New York.

Knuth, D., 1998, “The Art of Computer Programming. 3:Sorting and Searching (2nd ed.)”, Addison-Wesley.

Leveque, R. J., 2002, “Finite Volume Methods For Hyperbolic Problems”, Cambridge University Press.

Liou, M. S., 1996, A Sequel to AUSM: AUSM+, “Journal of Computational Physics”, Vol. 129, No. 2, pp. 364–382.

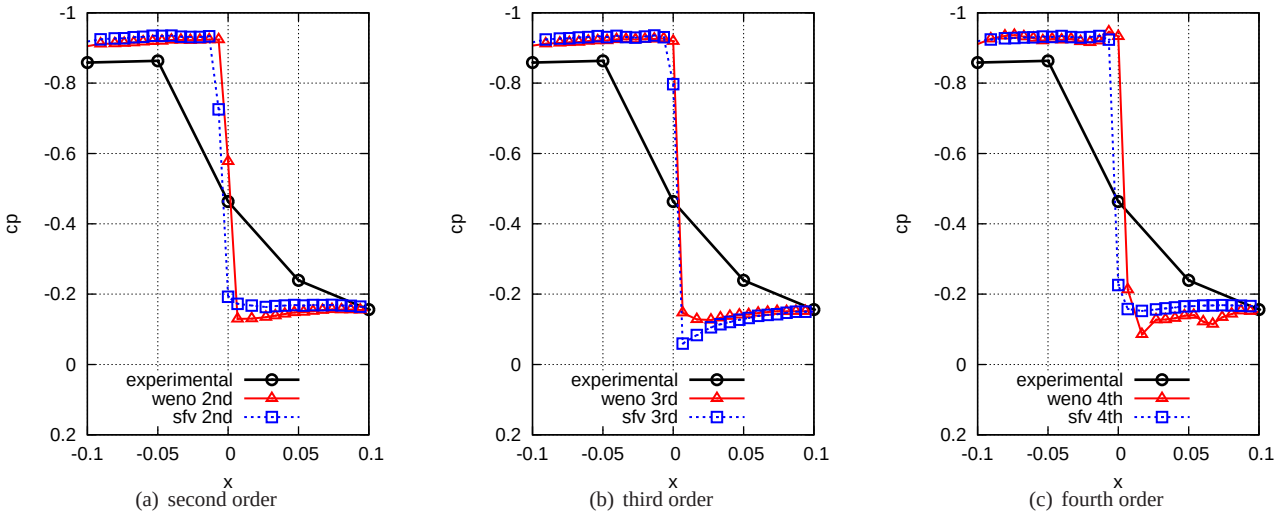
Liu, Y. and Vinokur, M., 1998, Exact Integrations of Polynomials and Symmetric Quadrature Formulas over Arbitrary Polyhedral Grids, “Journal of Computational Physics”, Vol. 140, No. 1, pp. 122–147.

Liu, Y., Vinokur, M., and Wang, Z. J., 2006, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids V: Extension to Three-Dimensional Systems, “Journal of Computational Physics”, Vol. 212, No. 2, pp. 454–472.

McDevitt, J. and Okuno, A. F., 1985, Static and Dynamic Pressure Measurements on a NACA 0012 Airfoil in the Ames High Reynolds Number Facility, NASA TP-2485.

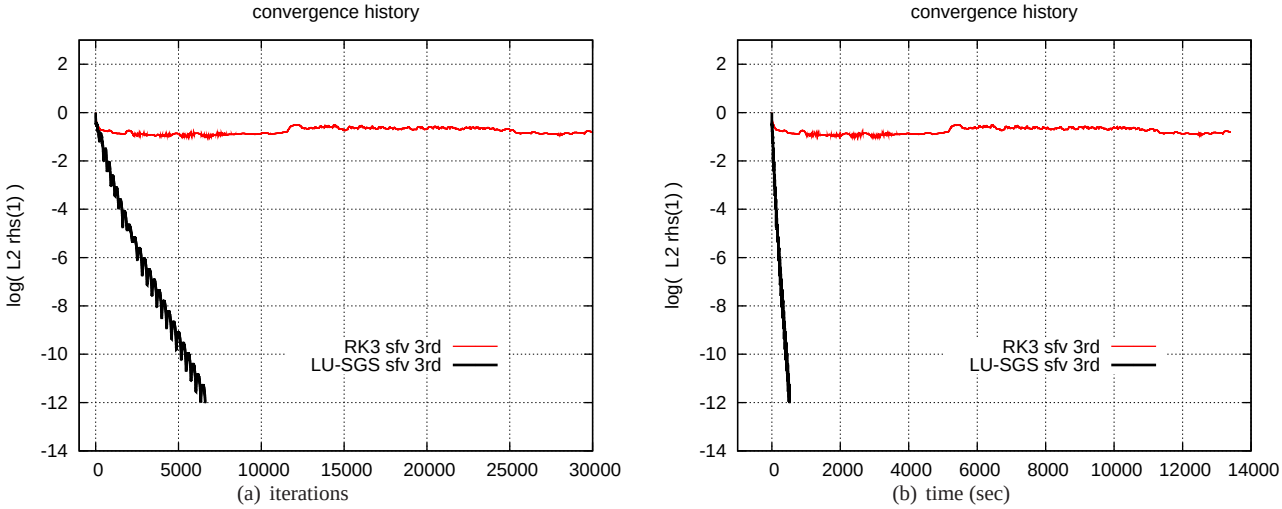
Roe, P. L., 1981, Approximatte Riemann Solvers, Parameter Vectors, and Difference Schemes, “Journal of Computational Physics”, Vol. 43, No. 2, pp. 357–372.

Scalabrin, L. C., 2002, “Numerical Simulation of Three-Dimensional Flows over Aerospace Configurations”, Master Thesis, Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil.



(a) second order (b) third order (c) fourth order

Figure 6. Detail of shock region.



(a) iterations (b) time (sec)

Figure 7. Convergence history for NACA 0012 airfoil with implicit and explicit 3rd-order SFV methods.

Shu, W. C., 1987, TVB Uniformly High-Order Schemes for Conservation Laws, "Mathematics of Computation", Vol. 49, pp. 105–121.

Sun, Y., Wang, Z. J., and Liu, Y., 2006, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids VI: Extension to Viscous Flow, "Journal of Computational Physics", Vol. 215, No. 1, pp. 41–58.

van den Abeele, K. and Lacor, C., 2007, An Accuracy and Stability Study of the 2D Spectral Volume Method, "Journal of Computational Physics", Vol. 226, No. 1, pp. 1007–1026.

van Leer, B., 1982, Flux-Vector Splitting for the Euler Equations, "Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics", pp. 507–512.

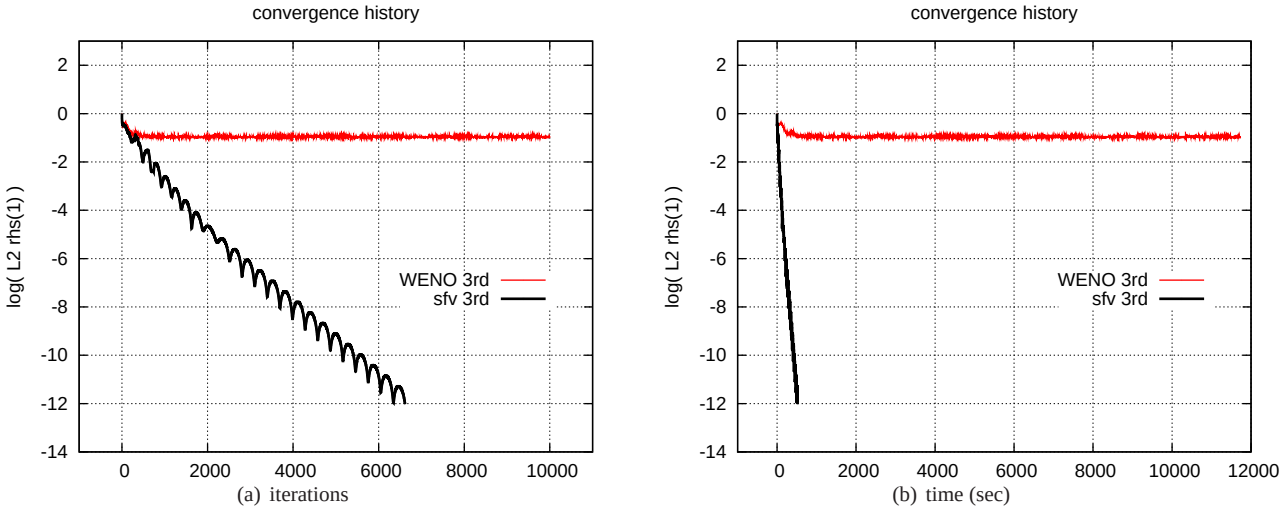
Wang, Z. J., 2002, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids: Basic Formulation, "Journal of Computational Physics", Vol. 178, No. 1, pp. 210–251.

Wang, Z. J. and Liu, Y., 2002, 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, pp. 665–698.

Wang, Z. J. and Liu, Y., 2003, Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids III: One-Dimensional Systems and Partition Optimization, "Journal of Scientific Computing", Vol. 20, pp. 137–157.

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

Wolf, W. R. and Azevedo, J. L. F., 2006, High-Order Unstructured Essentially Non Oscillatory and Weighted Essentially



(a) iterations (b) time (sec)

Figure 8. Convergence history for NACA 0012 airfoil simulation with implicit 3rd-order schemes.

Non Oscillatory Schemes for Aerodynamic Flows, “AIAA Journal”, Vol. 44, No. 10, pp. 2295–2310.

9. Responsibility notice

The authors are the only responsible for the printed material included in this paper