

## HIGH-ORDER SPECTRAL METHODS FOR COMPRESSIBLE FLOWS ON UNSTRUCTURED MESHES

C. Breviglieri<sup>1</sup>, F. M. Moreira<sup>2</sup> and J. L. F. Azevedo<sup>3</sup>

<sup>1</sup> Instituto Tecnológico de Aeronáutica, DCTA/ITA, (carbrevi@gmail.com)

<sup>2</sup> Instituto de Matemática, Estatística e Computação Científica, UNICAMP

<sup>3</sup> Instituto de Aeronáutica e Espaço, DCTA/IAE

**Abstract.** *The present work considers the use of high-order numerical methods for compressible aerodynamic solutions on unstructured meshes. The primary interest of the paper is to investigate the proper solution accuracy and efficiency of the Spectral Finite Volume (SFV) and Spectral Difference (SD) methods. The SFV method can easily be adapted to a finite volume solver while the SD method is designed for a finite difference framework. The mesh element support is also different for these schemes and will most likely affect the solution of a particular problem. The present work compares the order of accuracy and resolution capabilities of the two schemes for literature test cases.*

**Keywords:** *High-order methods, Unstructured mesh, Spectral Difference, Spectral Finite Volume.*

### 1. INTRODUCTION

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. Assessment of solution reliability in such cases may also require that a specified solution accuracy be met by the spatial discretization scheme. The spectral methods here considered were developed as an alternative to the discontinuous Galerkin, k-exact, ENO/WENO high-order schemes. Their common objective is to allow the implementation of a simpler and more efficient scheme, while still achieving high-order spatial accuracy.

The spectral finite volume (SFV) and spectral difference (SD) schemes are implemented in an unstructured, two dimensional Euler framework. The schemes allow for 2nd-, 3rd- and 4th-order accurate solutions, which use the Roe Riemann solver as the basic numerical flux scheme. The SFV method, as developed by Wang [1] circa 2002, was further extended to handle hyperbolic systems, two and three dimensions, smooth and discontinuous flow solutions. This method was originally incorporated by the authors into a 2-D finite volume solver

due to its similar underlying structure. However, the SFV method, at its current state, supports only triangular mesh elements. This proves to be a drawback if one is concerned with viscous flows, mainly due to the boundary layer discretization, where quadrilateral elements are better suited for the gradients discretization. The SD method, on the other hand, shares many of the SFV scheme attributes, as previously described, only that its formulation is suited for quadrilateral unstructured meshes. The present work represents an effort to evaluate the SD method as an alternate high-order method for the applications of interest to the authors, namely external compressible aerodynamic flows. The CFD tools developed by the authors and used in the present work have been verified and validated previously [2, 3, 4].

The paper is organized as follows. First, the theoretical formulation of the SFV method for the Euler equations is presented. The Spectral Difference method formulation is shown next. The order of accuracy of both schemes are measured and compared and the final section presents numerical results for well-known literature test cases.

## 2. SPECTRAL FINITE VOLUME METHOD

### 2.1. General Formulation

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

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

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

$$\frac{\partial}{\partial t} \int_V Q dV + \int_S (\vec{P} \cdot \vec{n}) dS = 0. \quad (2)$$

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

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

The standard CFD nomenclature is being used here. Hence,  $\rho$  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

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

where  $e_i$  is the specific internal energy, and the ratio of specific heats,  $\gamma$ , is set as 1.4 for all computations in this work. In the finite volume context, for fixed meshes, Eq. (2) can be rewritten for the  $i$ -th mesh element as

$$\frac{\partial Q_i}{\partial t} = -\frac{1}{V_i} \int_{SV_i} (\vec{P} \cdot \vec{n}) dS, \quad (5)$$

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  $i$ -th mesh element.

## 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. (5). In order to solve it numerically, the computational domain,  $\Omega$ , with proper initial and boundary conditions, is discretized into  $N$  non-overlapping triangles, the spectral volumes (SVs), such that

$$\Omega = \bigcup_{i=1}^N SV_i. \quad (6)$$

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  $\Gamma_K$ , such that

$$SV_i = \bigcup \Gamma_K. \quad (7)$$

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.

The boundary integral in Eq. (5) can be further discretized into the convective operator form

$$C(Q_i) \equiv \int_{SV_i} (\vec{P} \cdot \vec{n}) dS = \sum_{r=1}^{nf} \int_{A_r} (\vec{P} \cdot \vec{n}) dS, \quad (8)$$

where  $nf$  is the number of faces, or edges in 2-D, of  $SV_i$ , and  $A_r$  represents the area, or the length in 2-D, of 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. (8) can be performed numerically with a  $k$ -th order accurate Gaussian quadrature formula

$$\int_{SV_i} (\vec{P} \cdot \vec{n}) dS = \sum_{r=1}^{nf} \sum_{q=1}^{nq} w_{rq} \vec{P}(Q(x_{rq}, y_{rq})) \cdot \vec{n}_r A_r + O(A_r h^k), \quad (9)$$

where  $(x_{rq}, y_{rq})$  and  $w_{rq}$  are, respectively, the Gaussian quadrature points and the weights on the  $r$ -th edge of  $SV_i$ ,  $nq = \text{integer}((k+1)/2)$  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 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

$$G_1 = \frac{\sqrt{3}+1}{2\sqrt{3}} z_1 + \left(1 - \frac{\sqrt{3}+1}{2\sqrt{3}}\right) z_2 \quad \text{and} \quad G_2 = \frac{\sqrt{3}+1}{2\sqrt{3}} z_2 + \left(1 - \frac{\sqrt{3}+1}{2\sqrt{3}}\right) z_1, \quad (10)$$

Using the method described above, one can compute values of  $Q_i$  for instant  $t$  for each SV. 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 along the cell boundaries.

The previously described procedure follows exactly the standard finite volume method. For a given order of spatial accuracy,  $k$ , for Eq. (8), using the SFV method, each  $SV_i$  element must have at least

$$m = \frac{k(k+1)}{2} \quad (11)$$

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

$$q_{i,j} = \frac{1}{V_{i,j}} \int_{CV_{i,j}} q(x, y) dV, \quad (12)$$

where  $V_{i,j}$  is the volume of  $CV_{i,j}$ . Once the cell-averaged conservative variables, or DOFs, 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 the  $q(x, y)$  properties inside  $SV_i$ , i.e.,

$$p_i(x, y) = q(x, y) + O(h^{k-1}), \quad (x, y) \in SV_i, \quad (13)$$

where  $h$  represents the maximum edge length of all CVs within  $SV_i$ . The polynomial reconstruction process is discussed in details 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

$$\frac{dq_{i,j}}{dt} + \frac{1}{V_{i,j}} \sum_{r=1}^{nf} \int_{A_r} (\vec{f} \cdot \vec{n}) dS = 0, \quad (14)$$

where  $\vec{f}$  represents the  $E$  and  $F$  fluxes, and  $nf$  is the number of edges of  $CV_{i,j}$ . The numerical integration can be performed with a  $k$ -th order accurate Gaussian quadrature formulation, similarly to the one used for the SV elements in Eq. (9).

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

$$\vec{f}(q(x_{rq}, y_{rq})) \cdot \vec{n}_r \approx f_{Riemann}(q_L(x_{rq}, y_{rq}), q_R(x_{rq}, y_{rq}), \vec{n}_r), \quad (15)$$

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 [5] to compute the numerical flux, *i.e.*,

$$f_{Riemann} = f_{roe}(q_L, q_R, \vec{n}) = \frac{1}{2} \left[ \vec{f}(q_L) + \vec{f}(q_R) - |\overline{B}| (q_R - q_L) \right]. \quad (16)$$

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

$$B = n_x \frac{\partial E}{\partial q} + n_y \frac{\partial F}{\partial q}. \quad (17)$$

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

$$T^{-1} B T = \Lambda, \quad (18)$$

where  $\Lambda$  is the diagonal matrix composed of the eigenvalues of  $B$ , which can be written as

$$\Lambda = \text{diag} [v_n, v_n, v_n + a, v_n - a]. \quad (19)$$

$|\overline{B}|$  matrix is formed as

$$|\overline{B}| = T |\overline{\Lambda}| T^{-1}, \quad (20)$$

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

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

$$\frac{dq_{i,j}}{dt} = -\frac{1}{V_{i,j}} \sum_{r=1}^{nf} \sum_{q=1}^{nq} w_{rq} f_{Riemann}(q_L(x_{rq}, y_{rq}), q_R(x_{rq}, y_{rq}), \vec{n}_r) A_r, \quad (21)$$

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

### 2.3. Reconstruction Procedure

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 mesh element, denoted here as a Spectral Volume, or SV for short [1, 6]. 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

$$N_m = \frac{(m+1)(m+2)}{2}. \quad (22)$$

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

$$S_i = \bigcup_{j=1}^{N_m} C_{i,j}. \quad (23)$$

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

$$\int_{C_{i,j}} p_m(x, y) dS = \int_{C_{i,j}} q(x, y) dS. \quad (24)$$

With a complete polynomial basis,  $e_l(x, y) \in P_m$ , it is possible to satisfy Eq. (24). Hence,  $p_m$  can be expressed as

$$p_m = \sum_{l=1}^{N_m} b_l e_l(x, y), \quad (25)$$

where  $e$  is the base function vector,  $[e_1, \dots, e_N]$ , and  $b$  is the reconstruction coefficient vector,  $[b_1, \dots, b_N]^T$ . The substitution of Eq. (25) into Eq. (24) yields

$$\frac{1}{V_{i,j}} \sum_{l=1}^{N_m} b_l \int_{C_{i,j}} e_l(x, y) dS = \bar{q}_{i,j}. \quad (26)$$

If  $\bar{q}$  denotes the  $[\bar{q}_{i,1}, \dots, \bar{q}_{i,N_m}]^T$  column vector, Eq. (26) can be rewritten in matrix form as

$$Sb = \bar{q}, \quad (27)$$

where the  $S$  reconstruction matrix is given by

$$S = \begin{bmatrix} \frac{1}{V_{i,1}} \int_{C_{i,1}} e_1(x, y) dS & \cdots & \frac{1}{V_{i,1}} \int_{C_{i,1}} e_N(x, y) dS \\ \vdots & \cdots & \vdots \\ \frac{1}{V_{i,N}} \int_{C_{i,N}} e_1(x, y) dS & \cdots & \frac{1}{V_{i,N}} \int_{C_{i,N}} e_N(x, y) dS \end{bmatrix} \quad (28)$$

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

$$b = S^{-1} \bar{q}, \quad (29)$$

provided that  $S$  is non-singular. With the substitution of Eq. (29) into Eq. (24),  $p_m$  is, then, expressed in terms of shape functions  $L = [L_1, \dots, L_N]$ , defined as  $L = eS^{-1}$ , such that one could write

$$p_m = \sum_{j=1}^{N_m} L_j(x, y) \bar{q}_{i,j} = L \bar{q}. \quad (30)$$

Equation (30) gives the value of the conserved state variable,  $q$ , at any point within the SV and its boundaries, including the quadrature points,  $(x_{rq}, y_{rq})$ .

Table 1. Polynomial base functions.

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

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.

For most of the proposed applications in compressible aerodynamics, the use of a limiter technique is necessary to render the scheme stable. The authors conducted several studies regarding different limiter formulations and their performance. The interested reader is referred to Ref. [3] for a throughout discussion of this subject. The cited reference also presents the curved boundary formulation. This technique is responsible to maintain the methods design order of accuracy near the domain boundaries and also helps to reduce the overall number of mesh elements. The extension of such techniques for the Spectral Difference method have already been carried out elsewhere in the literature and is currently under development by the authors on their implementation.

## 2.4. Linear Reconstruction

For the linear SFV method reconstruction,  $m = 1$ , one needs to partition a SV in three sub-elements as in Eq. (22) 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 [7].

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. (28) are obtained analytically [8] for fundamental shapes. The shape functions, in the sense of Eq. (30), 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.

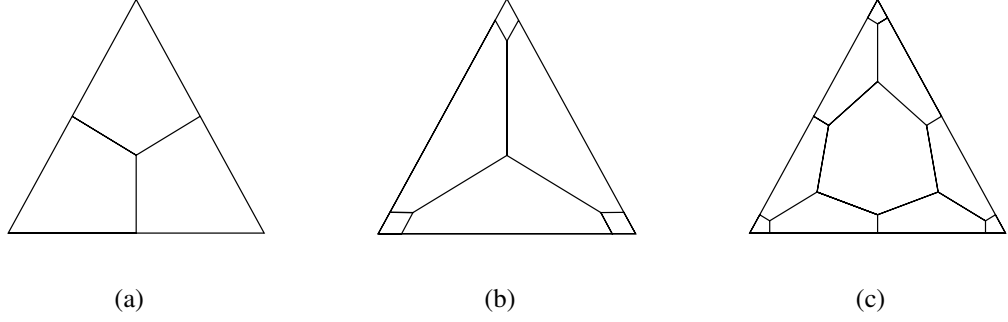


Figure 1. Triangular spectral volume partitions for (a) linear, (b) quadratic and (c) cubic reconstructions.

## 2.5. 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 [9] is shown in Fig. 1(b). It has a total of 13 points, 18 edges (9 external edges and 9 internal ones) and 36 quadrature points. The shape functions, in the sense of Eq. (30), 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 [8]. In this case, two quadrature points are required per CV edge.

## 2.6. 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 [9], presented in Fig. 1(c) and it has a total of 21 points, 30 edges (12 external edges and 18 internal ones) and 60 quadrature points. The shape functions, in the sense of Eq. (30), are obtained as in the linear partition in a pre-processing step. As with the quadratic reconstruction, each CV edge has two quadrature points [2].



### 3. SPECTRAL DIFFERENCE METHOD

The Spectral Difference method (SD) formulation in the present work considers the unsteady compressible 2-D Navier Stoker equations in conservative form

$$\frac{\partial Q}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} = 0 \quad (31)$$

where  $Q$  is the vector of conserved variables,  $E$  and  $F$  are the fluxes as described in the SFV method section. The implementation follows the formulation presented in Refs. [10, 11]. The SD method employs a Finite Difference like scheme rather than a Finite Volume formulation. In this sense, it represents a different approach to the solver programming and algorithm considerations. For instance, to achieve an efficient implementation, all elements in the physical domain  $(x, y)$  are transformed into a unit square element in the computational domain. Such transformation can be written as

$$\begin{pmatrix} x \\ y \end{pmatrix} = \sum_{i=1}^K M_i(\xi, \eta) \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (32)$$

where  $K$  is the number of points used to define the physical element,  $(x_i, y_i)$  are the Cartesian coordinates of those points, and  $M_i(\xi, \eta)$  are the shape functions of the geometric transformation. The metrics and the Jacobian of the transformation can be computed in a pre-processing step and kept in memory given the stationary aspect of the problems here considered. The governing equations in the physical domain are then transferred into the computation domain, and can be rewritten as

$$\frac{\partial \tilde{Q}}{\partial t} + \frac{\partial \tilde{E}}{\partial \xi} + \frac{\partial \tilde{F}}{\partial \eta} = 0 \quad (33)$$

where  $\tilde{Q} = |J| \cdot Q$  and

$$\begin{pmatrix} \tilde{E} \\ \tilde{F} \end{pmatrix} = |J| \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} \begin{pmatrix} E \\ F \end{pmatrix} \quad (34)$$

In the standard element, two sets of points are defined, namely the solution points and the flux points, illustrated in Fig. 2. In order to construct a degree  $(N - 1)$  polynomial in each coordinate direction, solution at  $N$  points are required. The solution points in 1-D are chosen to be the Gauss points defined by

$$X_s = \frac{1}{2} \left[ 1 - \cos \left( \frac{2s-1}{2N} \cdot \pi \right) \right], s = 1, 2, \dots, N. \quad (35)$$

The flux points are selected to be the Gauss-Lobatto points given by

$$X_{s+\frac{1}{2}} = \frac{1}{2} \left[ 1 - \cos \left( \frac{s}{N} \cdot \pi \right) \right], s = 0, 1, \dots, N. \quad (36)$$

Using the solution at  $N$  solution points, a degree  $(N - 1)$  polynomial can be built using the following Lagrange basis defined as

$$h_i(X) = \prod_{s=0, s \neq i}^N \left( \frac{X - X_s}{X_i - X_s} \right) \quad (37)$$

similarly, using the fluxes at  $(N + 1)$  flux points, a degree  $N$  polynomial can be built for the flux using a similar Lagrange basis defined as

$$l_{i+\frac{1}{2}}(X) = \prod_{s=0, s \neq i}^N \left( \frac{X - X_{s+\frac{1}{2}}}{X_{i+\frac{1}{2}} - X_{s+\frac{1}{2}}} \right) \quad (38)$$

The reconstructed solution for the conserved variables in the standard element is just the tensor products of the two one-dimensional polynomial,

$$Q(\xi, \eta) = \sum_{j=1}^N \sum_{i=1}^N \frac{\tilde{Q}_{i,j}}{|J_{i,j}|} h_i(\xi) \cdot h_j(\eta) \quad (39)$$

Similarly, the reconstructed flux polynomials take the following form

$$\tilde{E}(\xi, \eta) = \sum_{j=1}^N \sum_{i=0}^N \tilde{E}_{i+\frac{1}{2},j} l_{i+\frac{1}{2}}(\xi) \cdot h_j(\eta), \quad (40)$$

$$\tilde{F}(\xi, \eta) = \sum_{j=0}^N \sum_{i=1}^N \tilde{F}_{i,j+\frac{1}{2}} h_i(\xi) \cdot l_{j+\frac{1}{2}}(\eta). \quad (41)$$

The reconstructed fluxes are only element-wise continuous, but discontinuous across cell interfaces. For the inviscid flux, a Riemann solver is employed to compute a common flux at interfaces to ensure conservation and stability. In the present work, the Roe approximate Riemann solver is considered, as described in the previous sections. The numerical flux includes the artificial dissipation and upwind characteristic of the method.

In order to compute the inviscid flux derivatives the method computes the conservative variables at the flux points. The inviscid fluxes are then updated followed by the numerical flux calculations.

The derivatives of the fluxes are computed at the solution points using the derivatives of Lagrange operators  $l$  as

$$\begin{aligned} \left( \frac{\partial \tilde{E}}{\partial \xi} \right)_{i,j} &= \sum_{r=0}^N \tilde{E}_{r+\frac{1}{2},j} \cdot l'_{r+\frac{1}{2}}(\xi_i), \\ \left( \frac{\partial \tilde{F}}{\partial \eta} \right)_{i,j} &= \sum_{r=0}^N \tilde{F}_{i,r+\frac{1}{2}} \cdot l'_{r+\frac{1}{2}}(\eta_j) \end{aligned} \quad (42)$$

At the moment of writing, the SD method implementation is currently under way on the available numerical framework. A second, third and fourth-order reconstruction procedure will be considered for benchmark simulations against the SFV method.

#### 4. Temporal Discretization

The temporal discretization is concerned with solving a system of ordinary differential equations. In the present work, the authors use a third-order, TVD Runge-Kutta scheme [12]. Rewriting Eq. (21) in a concise ODE form, one obtains

$$\frac{dQ}{dt} = -\frac{1}{V_i} C(Q). \quad (43)$$

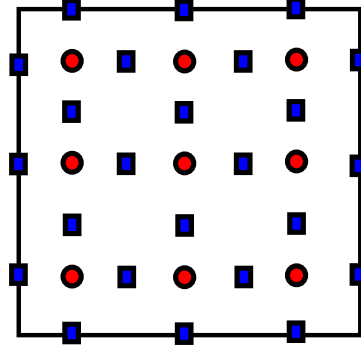


Figure 2. Possible flux points (blue squares) and solution points (red circles) distribution for the SD method.

where

$$Q = \begin{Bmatrix} Q_1 \\ \dots \\ Q_i \\ \dots \\ Q_N \end{Bmatrix}, \quad C(Q) = \begin{Bmatrix} C(Q_1) \\ \dots \\ C(Q_i) \\ \dots \\ C(Q_N) \end{Bmatrix}, \quad (44)$$

and

$$C(Q_i) = -\frac{1}{V_i} \sum_{r=1}^K \sum_{q=1}^J w_{rq} f_{Roe}(q_L(x_{rq}, y_{rq}), q_R(x_{rq}, y_{rq}), \vec{n}_r) A_r. \quad (45)$$

Hence, the time marching scheme can be written as

$$\begin{aligned} Q^{(1)} &= Q^n + \Delta t C(Q^n), \\ Q^{(2)} &= \alpha_1 Q^n + \alpha_2 [Q^{(1)} + \Delta t C(Q^{(1)})], \\ Q^{(n+1)} &= \alpha_3 Q^n + \alpha_4 [Q^{(2)} + \Delta t C(Q^{(2)})], \end{aligned}$$

where the  $n$  and  $n + 1$  superscripts denote, respectively, the values of the properties at the beginning and at the end of the  $n$ -th time step. The  $\alpha$  coefficients are  $\alpha_1 = 3/4$ ,  $\alpha_2 = 1/4$ ,  $\alpha_3 = 1/3$  and  $\alpha_4 = 2/3$ .

## 5. Accuracy Assessment

The accuracy of the SFV and SD methods is tested for the linear wave equation,

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \quad (46)$$

for  $-1 \leq x \leq 1$  and  $u(x, 0) = u_0(x)$  with periodic boundary condition at the domain extremes. For this setup, the initial condition is  $u_0(x) = \sin(\pi x)$ . No limiters were used in this study. A third-order TVD Runge-Kutta method is employed for time integration with a  $\Delta t$  value of  $10^{-4}$ , in order to make the discretization error time-step independent. Table 2

Table 2. Accuracy assessment of the 1-D SFV method for the wave equation. Gauss-Lobatto CV partition.

Order	NDOF	h	$L_\infty$ error	$L_\infty$ order	$L_1$ error	$L_1$ order
3	30	3.33E-002	2.67E-003	–	1.24E-003	–
	60	1.67E-002	3.65E-004	2.87	1.61E-004	2.95
	120	8.33E-003	4.67E-005	2.97	2.05E-005	2.97
	240	4.17E-003	5.90E-006	2.98	2.59E-006	2.98
	480	2.08E-003	7.41E-007	2.99	3.23E-007	3.00
4	20	3.33E-002	2.26E-003	–	7.30E-004	–
	40	1.67E-002	1.60E-004	3.82	5.07E-005	3.85
	80	8.33E-003	9.72E-006	4.04	3.18E-006	3.99
	160	4.17E-003	6.15E-007	3.98	2.00E-007	3.99
	320	2.08E-003	3.85E-008	4.00	1.26E-008	3.99
	640	1.04E-003	2.41E-009	4.00	7.87E-010	4.00
5	20	5.00E-002	5.30E-004	–	1.46E-004	–
	40	2.50E-002	1.96E-005	4.76	4.58E-006	4.99
	80	1.25E-002	6.50E-007	4.91	1.49E-007	4.94
	160	6.25E-003	2.13E-008	4.93	4.91E-009	4.92
	320	3.13E-003	6.13E-010	5.12	1.57E-010	4.97
6	30	5.00E-002	1.28E-005	–	2.57E-006	–
	60	2.50E-002	1.88E-007	6.09	4.08E-008	5.98
	120	1.25E-002	2.98E-009	5.98	6.49E-10	5.97
	240	6.25E-003	4.51E-011	6.05	1.04E-011	5.96

shows the  $L_1$  and  $L_\infty$  error norms produced using the SFV method with the Gauss-Lobatto point distribution. One can note that all schemes are able to achieve their expected order of accuracy. The same study is carried out for the SD method, presented in Table 3 and Fig. 3. In these tables,  $NDOF$  represents the number of degrees of freedom for the problem. One can observe that the expected order of accuracy, even for coarse grids, is achieved for both methods. However, the performance of the fifth and sixth-order SD scheme is questionable, specially for the finer meshes. This is mostly likely related to the oscillatory behavior of the polynomial interpolation, due to the internal points distribution chosen. As the grid is refined, the errors actually increase in both norms, which give a negative order of accuracy.

Other test cases are considered for the 1-D SFV and SD methods. A Gaussian pulse with a half width equal to 0.05 is used as an initial condition,

$$u(x, 0) = \exp \left[ - \left( \frac{x - 0.5}{0.05} \right)^2 \right] \quad (47)$$

again with periodic boundary conditions. This simulation was carried out for various orders of accuracy,  $k = 1, 2, 3, 4$  and 6 for  $t = 2$  with 100 SVs. The Gauss-Lobatto points distribution was used for CV partitioning. The results are presented in Fig. 4 and 5. The first-order

Table 3. Accuracy assessment of the 1-D SD method for the wave equation.

Order	NDOF	h	$L_\infty$ error	$L_\infty$ order	$L_1$ error	$L_1$ order
2	4	1.000E+00	6.021E-01	—	5.413E-01	—
	8	5.000E-01	4.215E-01	0.51	3.108E-01	0.80
	32	1.250E-01	4.090E-02	1.88	2.578E-02	1.88
	128	3.125E-02	2.618E-03	2.00	1.659E-03	1.99
	512	7.812E-03	1.637E-04	2.00	1.041E-04	2.00
	2048	1.953E-03	1.023E-05	2.00	6.510E-06	2.00
	4096	9.766E-04	2.557E-06	2.00	1.628E-06	2.00
3	6	1.000E+00	3.387E-01	—	2.029E-01	—
	12	5.000E-01	5.234E-02	2.69	1.937E-02	3.39
	48	1.250E-01	5.535E-04	3.27	2.402E-04	3.12
	192	3.125E-02	9.059E-06	2.90	3.634E-06	2.99
	768	7.812E-03	1.480E-07	2.98	5.763E-08	2.99
	3072	1.953E-03	2.338E-09	2.99	9.047E-10	3.00
	6144	9.766E-04	2.928E-10	3.00	1.132E-10	3.00
4	8	1.000E+00	6.080E-02	—	2.709E-02	—
	32	2.500E-01	2.073E-04	3.94	8.377E-05	4.13
	128	6.250E-02	8.314E-07	3.99	3.141E-07	4.01
	512	1.562E-02	3.260E-09	4.00	1.224E-09	4.00
	2048	3.906E-03	1.263E-11	4.01	4.808E-12	3.99
	8192	9.766E-04	5.157E-13	0.57	3.065E-13	0.27
5	10	1.000E+00	9.320E-03	—	3.630E-03	—
	40	2.500E-01	1.153E-05	5.10	6.207E-06	4.86
	160	6.250E-02	1.142E-08	4.97	6.103E-09	5.00
	640	1.562E-02	1.127E-11	4.99	5.982E-12	4.99
	2560	3.906E-03	2.104E-13	1.02	1.298E-13	0.89
	10240	9.766E-04	7.930E-13	-0.99	4.995E-13	-1.00
6	12	1.000E+00	1.672E-03	—	6.279E-04	—
	48	2.500E-01	4.068E-07	5.80	1.415E-07	6.02
	192	6.250E-02	9.464E-11	6.03	3.641E-11	5.98
	768	1.562E-02	1.679E-13	3.10	9.192E-14	2.65
	3072	3.906E-03	6.010E-13	-0.98	3.786E-13	-1.00
	12288	9.766E-04	2.386E-12	-0.99	1.515E-12	-1.00

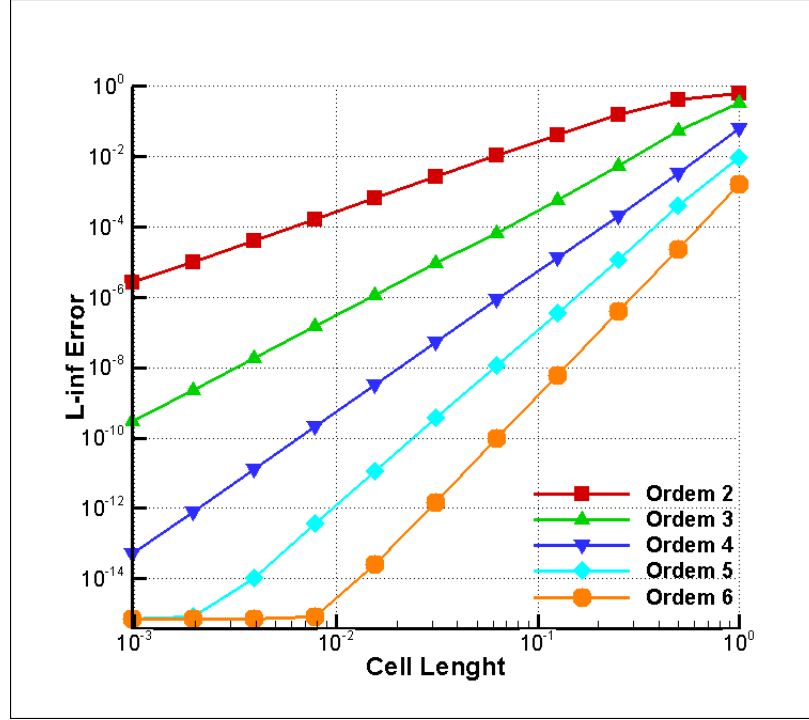


Figure 3. Spectral Difference solution error versus mesh spacing plot order of accuracy measurement of a sine wave convection.

simulation smeared the pulse to the point that it is hardly recognizable. This is due to the extra amount of dissipation associated with such low-order schemes. The second-order simulation retained the shape of the initial condition but also smeared the pulse and produced an oscillatory behavior, which, in turn, is associated with the dispersion properties of the method. The third, fourth and sixth-order simulation yielded nice results just like the analytical solution, also shown in the figure.

### 5.1. 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 [13]. The analytical solution is used as initial condition for the SFV 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  $k = 0.6$ , in order to define the bounding walls, and  $v_t = 0.35$  to define the inlet and outlet boundaries. For such 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 [14]. In order 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  $L_\infty$  norms of the density.

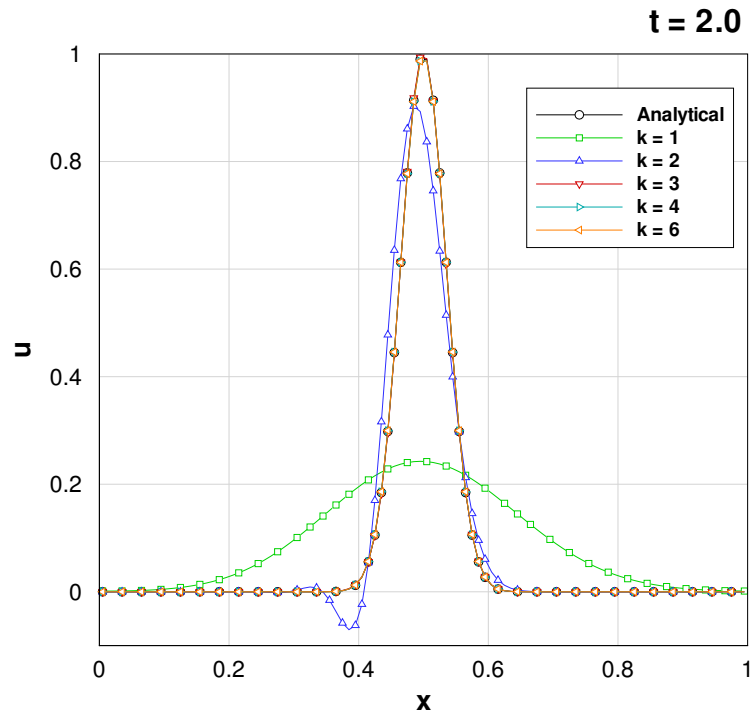


Figure 4. Simulation of a traveling Gaussian pulse with SFV schemes of various orders at  $t = 2$ .

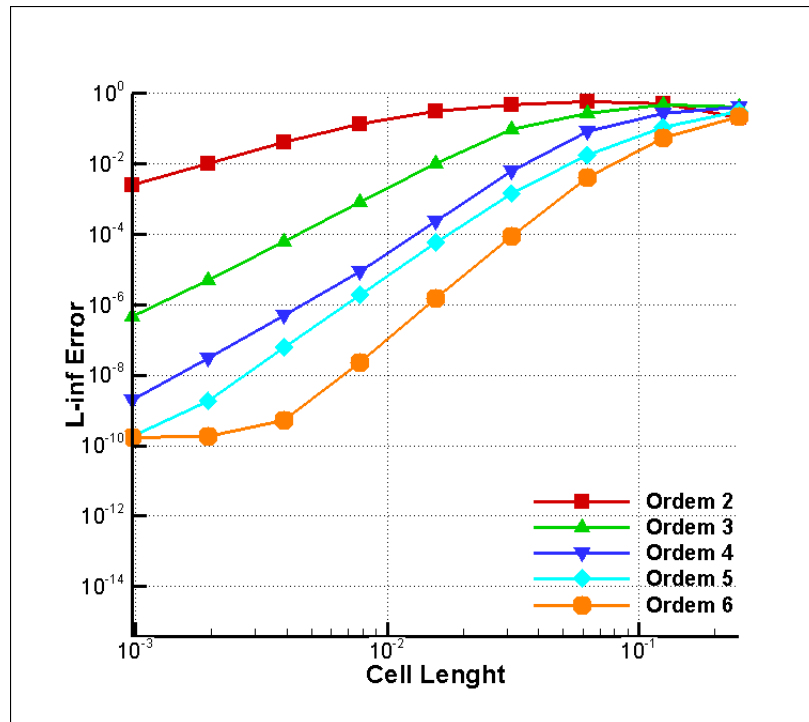


Figure 5. Spectral Difference error versus mesh spacing plot order of accuracy measurement of a Gaussian pulse convection.

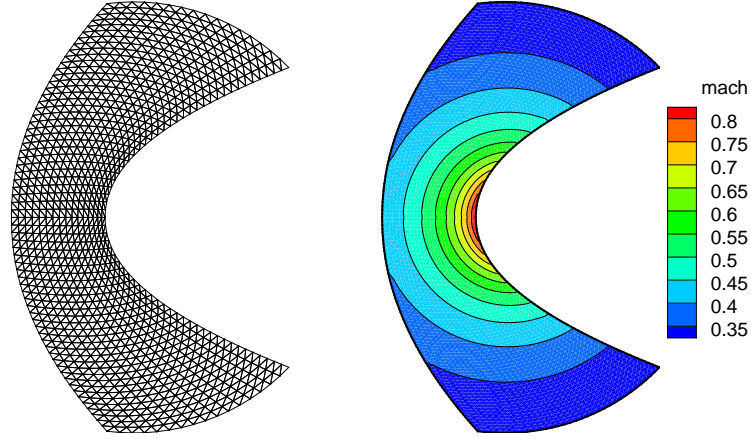


Figure 6. Ringleb flow mesh and Mach number contour results for fourth-order SFV method.

Table 4. Accuracy assessment of SFV method for the Ringleb flow test case.

Method	Mesh elements	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order
3rd 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
4th 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

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. Table 4 presents the  $L_1$  and  $L_\infty$  error norms of the density for the present calculations with the high-order boundary representation for the SFV method.

## 6. Airfoil Simulation

A simulation with the NACA 0012 airfoil is considered for the SFV method. The Euler equations are solved on a coarse mesh with 716 cells and 358 nodes, from which 40 define the airfoil wall. This  $O$ -grid mesh is presented in Fig. 7. The airfoil profile itself is collapsed on the trailing edge. The far field boundary radius is 50 chord units.

The free stream flow replicate the conditions of the experimental data [15], that is, free stream Mach number value of  $M_\infty = 0.8$  and 0 deg angle-of-attack. Simulations with the second, third and fourth order SFV schemes are performed, along with the first-order Roe scheme. Figure 8 shows the computed  $C_p$  values obtained with with the first-order Roe scheme and third-order SFV method, considering quadratic curved boundaries. It is clear



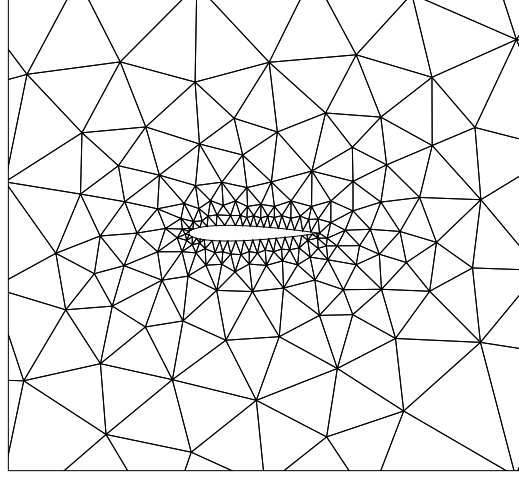


Figure 7. Coarse mesh employed on the NACA 0012 airfoil simulation.

from the  $C_p$  distribution that the first-order scheme introduces too much dissipation and essentially smears out the shock wave. The high-order  $C_p$  distribution, on the other hand, is remarkably close to the experimental results in Fig. 8, particularly for the shock position, considering the crude mesh discretization. These show the potential for high-order methods, on such applications, to ease the mesh generation process. One should observe, however, that the experimental results consider the presence of the boundary layer and the consequent shock-boundary layer interaction that necessarily occurs in the experiment. For the numerical solution, the shock presents a sharper resolution, as one can expect for an Euler simulation.

Another relevant simulation is performed to assess the benefits of the curved boundary implementation, namely, the measure of entropy error  $\epsilon_s$  levels at the airfoil boundary. Because the diffusive flux vectors are zero, there is no physical dissipation mechanism that produces heat in regions of smooth flow, away from shocks. If no external heat is added into the flow, then it is called adiabatic and, from the first law of thermodynamics, it follows that entropy, give by

$$s = C_v \ln \left( \frac{p}{\rho^\gamma} \right), \quad (48)$$

is constant throughout the field if no shocks are present. Therefore, the entropy error  $\epsilon_s$ , defined as

$$\epsilon_s = \frac{p}{p_\infty} \left( \frac{\rho_\infty}{\rho} \right)^\gamma - 1, \quad (49)$$

is a good measure of the accuracy of a numerical solution obtained with a method to approximately solve the Euler equations. Figure 9 presents the entropy error generate by the third-order SFV method with linear and curved boundary cells for the coarse mesh, which has only 40 cells to represent the whole airfoil geometry. The curved boundary approach is able to produce the smaller error levels than the linear approach. One can even note, form the figure, that at position  $x = 0$  there is an increase of entropy error, due to the presence of the

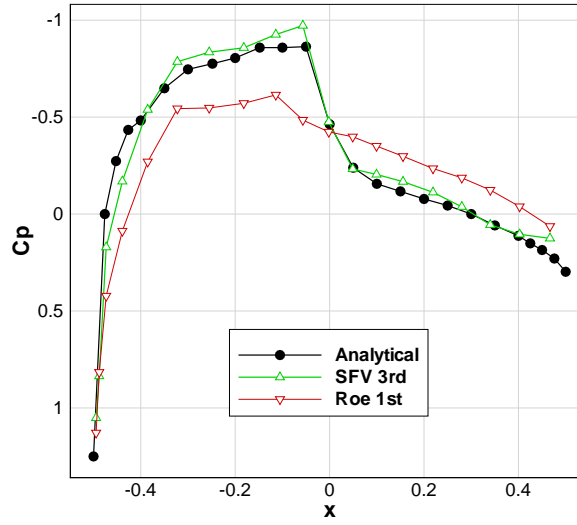


Figure 8. NACA 0012  $C_p$  distribution for coarse mesh solutions.

shock wave in this region. This, again, demonstrates that such extension indeed improves the overall accuracy of the SFV method.

## 7. CONCLUSIONS

The present work compares two high-order methods for several test cases. The formal order of accuracy of the SFV and SD methods is achieved for the model convection equation. Moreover, solutions of the Euler equations for the SFV method are in good agreement with the available analytical and experimental data. Even though additional numerical techniques are required to handle the desired flow regime of compressible aerodynamics, such as the limiter and high-order boundary representation, the SFV method is only capable to work with triangular mesh elements. The ability to handle quadrilateral meshes is a desirable feature of the SD method, specially for RANS simulations. Current work is in place to extend the SD method for the 2-D formulation and proper results and benchmark simulations against the SFV method will be presented at the conference.

## Acknowledgements

The authors gratefully acknowledge the partial support for this research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grants No. 312064/2006-3 and No. 471592/2011-0. This work is also supported by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, through a Ph.D. Scholarship for the first author.

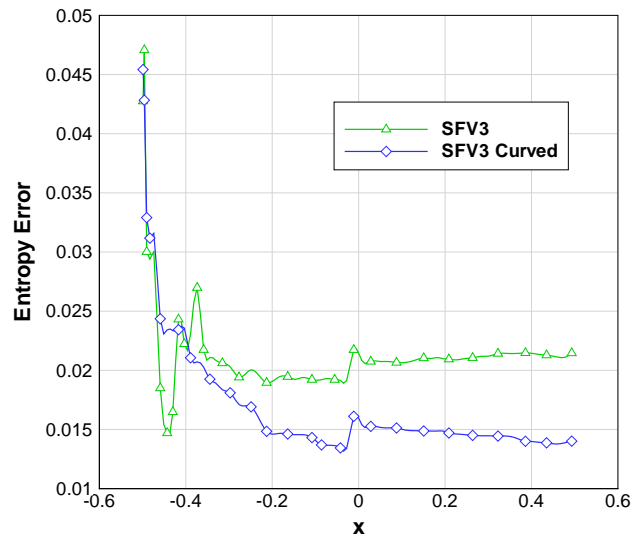


Figure 9. Entropy error for third-order SFV method.

## 8. REFERENCES

- [1] 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.
- [2] 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 No. 2008-7182, Honolulu, HI, Aug. 2008.
- [3] Breviglieri, C., Azevedo, J. L. F., Basso, E., and Souza, M. A. F., “Implicit High-Order Spectral Finite Volume Method for Inviscid Compressible Flows,” *AIAA Journal*, Vol. 48, No. 10, Oct. 2010, pp. 2365–2376.
- [4] Breviglieri, C. and Azevedo, J. L. F., “Unsteady Aerodynamic Applications Using High-Order Unstructured Grid Methods,” *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA Paper No. 2012-0701, Nashville, TN, Jan. 2012.
- [5] Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes,” *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.
- [6] 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, Jul. 2002, pp. 665–698.
- [7] Knuth, D. E., *The Art of Computer Programming. 3: Sorting and Searching (2nd ed.)*, Addison-Wesley, Reading, MA, 1998.

- [8] 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.
- [9] 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.
- [10] Wang, Z. J., Liu, Y., May, G., and Jameson, A., “Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations,” *Journal of Scientific Computing*, Vol. 32, No. 1, 2007.
- [11] May, G. and Jameson, A., “A Spectral Difference Method for the Euler and Navier-Stokes Equations on Unstructured Meshes,” *44th AIAA Aerospace Sciences Meeting*, AIAA Paper No. 2006-304, Reno, NV, Jan. 2006.
- [12] Shu, W. C., “TVB Uniformly High-Order Schemes for Conservation Laws,” *Mathematics of Computation*, Vol. 49, 1987, pp. 105–121.
- [13] Shapiro, A. H., *The Dynamics and Thermodynamics of Compressible Fluid Flow*, Wiley, New York, 1953.
- [14] 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.
- [15] McDevitt, J. and Okuno, A. F., “Static and Dynamic Pressure Measurements on a NACA 0012 Airfoil in the Ames High Reynolds Number Facility,” NASA TP-2485, NASA, Jun. 1985.