A Review of Some Numerical Methods to the Euler Equations in Two-Dimensions

EDISSON SÁVIO DE GÓES MACIEL
IEA- Aeronautical Engineering Division
ITA – Aeronautical Technological Institute
Praça Mal. Eduardo Gomes, 50 – Vila das Acácias – São José dos Campos – SP – 12228-900
BRAZIL
edissonsavio@yahoo.com.br

Abstract: - This work aims to describe the numerical implementation of the Lax and Friedrichs, Lax and Wendroff TVD, Boris and Book, Beam and Warming and MacCormack, on a finite volume and structured spatial discretization contexts, to solve the Euler equations in two-dimensions. The Lax and Wendroff algorithm was implemented according to the TVD formulation of Yee. The Beam and Warming scheme was implemented only in its explicit version. Hence, it is possible to distinguish four categories of algorithms studied in this work: symmetrical (Lax and Friedrichs and Beam and Warming), FCT (Boris and Book), Predictor/Corrector (MacCormack) and TVD (Lax and Wendroff). They are applied to the solution of the steady state problem of the moderate supersonic flow along a compression corner. A spatially variable time step procedure is employed to accelerate the convergence of the numerical methods to the steady state condition. This procedure has demonstrated a meaningful gain in terms of convergence ratio, as reported by Maciel. The results have demonstrated that the Beam and Warming scheme, using the nonlinear dissipation operator, provides the best results in terms of quality (good capture of shock wave thickness and wall pressure profile) and quantity (good prediction of the oblique shock wave angle).

Key-Words: - Symmetrical schemes, FCT scheme, Predictor/Corrector scheme, TVD scheme, Finite volumes, Euler equations, Two-dimensions.

1 Introduction

This paper is a review of the initial numerical methods applied to the solution of the Euler equations. The objective of this work is to provide anyone with the knowledge employed in the initial development of the numerical methods applied to capture flow discontinuities. Symmetrical, Flux-Corrected Transport (FCT), Predictor/Corrector and TVD are the types of numerical methods studied in this research. The first works in the CFD community involving fluid dynamics, one-dimensional case, date from 1950. The works of [1] and [2] were the precursors of the modern algorithms in CFD. Some comments about important algorithms developed since 1950 to 1990 are described below:

In [1], the first representative schemes of the modern development in the field of numerical discretization of the Euler equations were presented. They were known as the schemes of Lax or Lax and Friedrichs ([1]). They are not applied in their original form any longer, due to their poor first-order accuracy, but several variants with improved accuracy are still in use. The basic idea behind the scheme of [1] was to stabilize the explicit, unstable central scheme obtained from a central differencing of the first derivative of the flux term. With this in mind, second order dissipation was introduced in the scheme, stabilizing it, but damaging severely the solution quality. A corrected viscosity scheme was constructed to improve the results and reach asymptotically second order accuracy.

[2] emphasized that the limitation of the speed and memory of calculating machines placed an upper bound on the number of mesh points that could be used in a finite difference calculation. This means that in problems involving many independent variables (and until that date, three was many) the mesh employed was necessarily coarse. Therefore in order to get reasonable accurate final results one needed to employ highly accurate difference approximations. The purpose of their work was also set up and analysis such difference schemes for solving the initial value problem for first order symmetric hyperbolic systems of partial differential equation in two space variables.

[3] proposed a new approach for numerically solving the continuity equation which yielded physically reasonable results even in circumstances where standard algorithms failed. This approach, called Flux-Corrected Transport (FCT), led to a class of Eulerian finite-difference algorithms which

strictly enforced the non-negative property of realistic mass and energy densities. As a result steep gradients and shocks could be handled particularly well. The method required no special knowledge about the solution and all internal grid points in the calculation were treated identically.

[4] developed an implicit finite-difference scheme for the numerical solution of the compressible Navier-Stokes equations in conservation-law form. The algorithm was second-order-time accurate, non-iterative and spatially factored. In order to obtain an efficient factored algorithm, the spatial cross derivatives were evaluated explicitly. However, the algorithm was unconditionally stable and, although a three-time-level scheme, requires only two time levels of data storage. The algorithm was constructed in a "delta" form (i.e., increments of the conserved variables and fluxes) that provided a direct derivation of the scheme and led to an efficient computational algorithm. In addition, the delta form had the advantageous property of a steady state (if one existed) independent of the size of the time step. Numerical results were presented for a two-dimensional shock boundary-layer interaction problem.

[5] emphasized the important developments that occurred in numerical methods in recent years. Chief among them had been the development of non-iterative implicit methods for solving the compressible Navier-Stokes equations. These methods, not subject to conventional explicit stability conditions, have significantly improved computational efficiency over the earlier explicit methods. However, their time step sizes are still frequently limited by severe accuracy and stability criteria, and their computer time per step, as well as their programming complexity, is much larger than that of the explicit methods. The goal of his research was to develop a method for solving the compressible form of the Navier-Stokes equations at high Reynolds number that is unconditionally stable, computationally more efficient than existing methods, and simple and straightforward to program. The method contained two stages. The first stage used the explicit predictor/corrector finite difference method presented by the author in 1969. The generated finite-difference equations approximated the governing equations of fluid flow to second-order accuracy in space and time, were simple to program, but were subjected to restrictive explicit stability conditions. The second stage removed these stability conditions by transforming numerically the equations of the first stage into an implicit form. The resulting matrix equations to be solved were either upper or lower block two-diagonal equations and were solved more easily than the block three-diagonal matrix equations of existing implicit methods.

[6] reformulated a one-parameter family of second-order explicit and implicit total variation diminishing (TVD) schemes so that a simpler and wider group of limiters was included. The resulting scheme could be viewed as a symmetrical algorithm with a variety of numerical dissipation terms that were designed for weak solutions of hyperbolic problems. This was a generalization of the work of [7-8] to a wider class of symmetric schemes other than [2]. The main properties of this class of schemes were that they could be implicit, and, when steady-state calculations were sought, the numerical solution was independent of the time step. Numerical experiments with two-dimensional unsteady and steady-state airfoil calculation showed that the proposed symmetric TVD schemes were quite robust and accurate.

This work aims to describe the numerical implementation of the [1], [2]/TVD, [3-5], on a finite volume and structured spatial discretization contexts, to solve the Euler equations in two-dimensions. The [2] algorithm was implemented according to the TVD formulation of [6]. The [4] scheme was implemented only in its explicit version. Hence, it is possible to distinguish four categories of algorithms studied in this work: symmetrical ([1; 4]), FCT ([3]), Predictor/Corrector ([5]) and TVD ([2]). They are applied to the solution of the steady state problem of the moderate supersonic flow along a compression corner. A spatially variable time step procedure is employed to accelerate the convergence of the numerical methods to the steady state condition. This procedure has a meaningful gain in terms of convergence ratio, as reported by [9-10]. The results have demonstrated that the [4] scheme, using the nonlinear dissipation operator, provides the best results in terms of quality (good capture of shock wave thickness and wall pressure profile) and quantity (good prediction of the oblique shock wave angle).

2 Euler Equations

The fluid movement is described by the Euler equations, which express the conservation of mass, of linear momentum and of energy to an inviscid, heat non-conductor and compressible mean, in the absence of external forces. In the integral and conservative forms, these equations can be represented by:

$$\partial/\partial t \int_V Q dV + \int_S \left(E_e n_x + F_e n_y\right) dS = 0\,, \qquad (1)$$

where Q is written to a Cartesian system, V is a cell volume, n_x and n_y are the Cartesian components of the normal unity vector to the flux face, S is the surface area and E_e and F_e represent the components of the convective flux vector. Q, E_e and F_e are represented by:

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{Bmatrix}, \quad E_e = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (e+p)u \end{Bmatrix} \quad \text{and} \quad F_e = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (e+p)v \end{Bmatrix}, \qquad (2)$$

being ρ the fluid density; u and v the Cartesian components of the velocity vector in the x and y directions, respectively; e the total energy; and p the static pressure.

In the studied problem, the Euler equations were nondimensionalized in relation to the freestream density, ρ_∞, and in relation to the freestream speed of sound, a_∞. The matrix system of the Euler equations is closed with the state equation of a perfect gas:

$$p = (\gamma - 1)\left[e - 0.5\rho(u^2 + v^2)\right] \qquad (3)$$

and γ being the ratio of specific heats, assuming the value 1.4 to "cold gas flow". The total enthalpy is determined by $H = (e+p)/\rho$.

3 [1] Algorithm

The [1] scheme is a symmetrical one, first order accurate. The convective flux vector is obtained by arithmetical average between the flux vectors that shared a flux interface. Hence, it is possible to write:

$$E_{\text{int}} = 0.5\left(E_R + E_L\right) \quad \text{and} \quad F_{\text{int}} = 0.5\left(F_R + F_L\right) \qquad (4)$$

where E_R and E_L are the right and left convective or Euler flux vectors which shared an interface. These flux vectors are obtained by Eq. (2). Hence, the convective flux vector at an interface is defined by:

$$F_{i+1/2,j} = E_{i+1/2,j} S_{x_{i+1/2,j}} + F_{i+1/2,j} S_{y_{i+1/2,j}} \qquad (5)$$

with:

$$S_{x_{i+1/2,j}} = \left(n_x S\right)_{i+1/2,j} = n_{x_{i+1/2,j}} S_{i+1/2,j}\,; \qquad (6)$$

$$S_{y_{i+1/2,j}} = \left(n_y S\right)_{i+1/2,j} = n_{y_{i+1/2,j}} S_{i+1/2,j}\,; \qquad (7)$$

$$n_x = \Delta y \Big/ \sqrt{\Delta x^2 + \Delta y^2}\,,\; n_y = -\Delta x \Big/ \sqrt{\Delta x^2 + \Delta y^2}\,; \qquad (8)$$

$$S = \sqrt{\Delta x^2 + \Delta y^2} \qquad (9)$$

and Δx and Δy are determined conform presented in Tab. 1.

Table 1. Values of Δx and Δy to each interface.

Interface	Δx	Δy
i,j-1/2	$x_{i+1,j} - x_{i,j}$	$y_{i+1,j} - y_{i,j}$
i+1/2,j	$x_{i+1,j+1} - x_{i+1,j}$	$y_{i+1,j+1} - y_{i+1,j}$
i,j+1/2	$x_{i,j+1} - x_{i+1,j+1}$	$y_{i,j+1} - y_{i+1,j+1}$
i-1/2,j	$x_{i,j} - x_{i,j+1}$	$y_{i,j} - y_{i,j+1}$

A computational cell, with its flux interfaces and nodes is presented in Fig. 1.

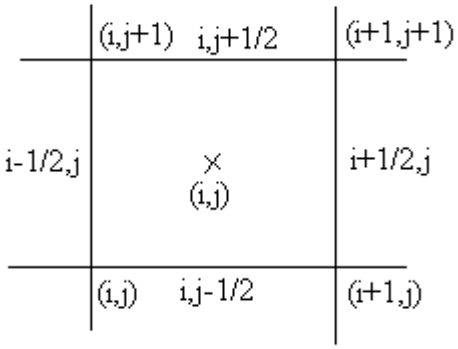

Figure 1. Computational cell: its interfaces, nodes and centroid.

As the algorithm is symmetrical, an artificial dissipation operator should be incorporated to the convective calculation. In 1954, when this algorithm was suggested, only a dissipation operator of second order was necessary to become the scheme stable. This dissipation proportioned the convergence of the algorithm, but a great amount of dissipation was provided for the algorithm. As conclusion, the scheme was only first order accurate. To improve the order of accuracy of this scheme, [1] suggested a viscosity correction to be considered in an anti-dissipative term.

3.1 Artificial Dissipation and Viscosity Correction

The artificial dissipation model employed on a finite volume context and according to the description above is:

$$D_{i,j} = d^2_{i,j} = 0.25 \times V_{i,j} / \Delta t_{i,j} \times \left(Q_{i,j-1} + Q_{i+1,j} + Q_{i,j+1} + Q_{i-1,j} - 4Q_{i,j}\right). \quad (10)$$

This term is equivalent to the dissipation provided by the finite difference model of [1], in the two-dimensional case. To eliminate the excess of dissipation conceded by this operator, a viscosity correction term considered in an anti-dissipative operator was employed to obtain asymptotically second order of accuracy. This operator is described below:

$$AD_{i,j} = \beta \times 0.25 \times V_{i,j} / \Delta t_{i,j} \times \left(Q_{i,j-1} + Q_{i+1,j} + Q_{i,j+1} + Q_{i-1,j} - 4Q_{i,j}\right), \quad (11)$$

where β is the viscosity correction term and should be less than 1.0. The dissipation operator and the anti-dissipative operator are very similar. The main differences are that: D is subtracted from the convective flux balance and AD is added to the convective flux balance. The value adopted for β is obtained through numerical experiments. The value employed in the present study was 0.85.

3.2 Numerical Convective Contribution

The numerical convective contribution of the (i,j) cell, as using the [1] scheme, is finally defined as:

$$C_{i,j} = \left(F_{i+1/2,j} - F_{i-1/2,j} + F_{i,j+1/2} - F_{i,j-1/2}\right) - D_{i,j} + AD_{i,j} \quad (12)$$

and the residual is defined as:

$$R_{i,j} = -\Delta t_{i,j} / V_{i,j} \times C_{i,j}\ , \quad (13)$$

where $\Delta t_{i,j}$ is the spatially variable time step and $V_{i,j}$ is the cell volume. The volume is determined by the following expression:

$$V_{i,j} = 0.5\left|\left(x_{i,j} - x_{i+1,j}\right)y_{i+1,j+1} + \left(x_{i+1,j} - x_{i+1,j+1}\right)y_{i,j} + \left(x_{i+1,j+1} - x_{i,j}\right)y_{i+1,j}\right| + 0.5\left|\left(x_{i,j} - x_{i+1,j+1}\right)y_{i,j+1} + \left(x_{i+1,j+1} - x_{i,j+1}\right)y_{i,j} + \left(x_{i,j+1} - x_{i,j}\right)y_{i+1,j+1}\right|. \quad (14)$$

[1] suggested that the anti-dissipative term should not be evaluated at each step. Differently, the anti-dissipative term should be evaluated only after a certain number of iterations performed by the algorithm be accomplished. This number of iterations is equal to the maximum number of points in an specific direction; Moreover, the maximum number between "imax" and "jmax", where imax is the maximum number of points in i and jmax is the maximum number of points in j, should be the number of iterations to evaluate the anti-dissipative term. This number of iterations is exactly the number of mesh points to an information propagate along the mesh, in the most distant boundary.

3.3 Time Integration

The time integration uses the forward Euler method to perform the marching of the solution to the steady state. This method is determined as follows:

$$Q^{n+1}_{i,j} = Q^{n}_{i,j} + R^{n}_{i,j}\ . \quad (15)$$

This method is first order accurate in time, but as the main interest of this work is steady state solutions, there is no problem.

4 [2] Algorithm

The [2] TVD ("Total Variation Diminishing") algorithm, second order accurate in space, is specified by the determination of the numerical flux vector at the (i+1/2,j) interface. The extension of this numerical flux to the (i,j+1/2) interface is straightforward, without any additional complications.

Initially is necessary to define the coordinate changes from the Cartesian system to a generalized coordinate system. The right and left cell volumes, as well the interface volume, necessary to coordinate change, following the finite volume formulation, which is equivalent to a generalized coordinate system, are defined as:

$$V_R = V_{i+1,j}\ ,\ V_L = V_{i,j} \quad \text{and} \quad V_{\text{int}} = 0.5\left(V_R + V_L\right), \quad (16)$$

where "R" and "L" represent right and left states, respectively. The area components at interface are defined by: $S_{x\_\text{int}} = s'_x S$ and $S_{y\_\text{int}} = s'_y S$, where s'_x and s'_y are defined as: $s'_x = s_x / S$ and $s'_y = s_y / S$, being $S = \left(s_x^2 + s_y^2\right)^{0.5}$. Expressions to s_x and s_y, which represent the S_x and S_y components always adopted in the positive orientation, are given in Tab. 2. These normalized area vectors are employed in the [2] TVD high resolution scheme.

The metric terms to this generalized coordinate system are defined as:

$$h_x = S_{x\_\text{int}} / V_{\text{int}}\ ,\ h_y = S_{y\_\text{int}} / V_{\text{int}} \quad \text{and} \quad h_n = S / V_{\text{int}}\ . \quad (17)$$

The calculated properties at the flux interface are obtained by arithmetical average or by [11] average. The [11] average was used in this work:

$$\rho_{\text{int}} = \sqrt{\rho_L \rho_R}\ ,\ u_{\text{int}} = \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (18)$$

$$v_{\text{int}} = \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (19)$$

$$H_{\text{int}} = \left(H_L + H_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (20)$$

$$a_{\text{int}} = \sqrt{(\gamma-1)\left[H_{\text{int}} - 0.5\left(u_{\text{int}}^2 + v_{\text{int}}^2\right)\right]}. \quad (21)$$

Table 2. Normalized values of s_x and s_y.

Interface	s_x	s_y
i,j-1/2	$-\left(y_{i+1,j} - y_{i,j}\right)$	$\left(x_{i+1,j} - x_{i,j}\right)$
i+1/2,j	$\left(y_{i+1,j+1} - y_{i+1,j}\right)$	$\left(x_{i+1,j} - x_{i+1,j+1}\right)$
i,j+1/2	$\left(y_{i,j+1} - y_{i+1,j+1}\right)$	$\left(x_{i+1,j+1} - x_{i,j+1}\right)$
i-1/2,j	$\left(y_{i,j+1} - y_{i,j}\right)$	$-\left(x_{i,j+1} - x_{i,j}\right)$

The eigenvalues of the Euler equations, in the ξ direction, to the convective flux are given by:

$$U_{cont} = u_{\text{int}}h_x + v_{\text{int}}h_y\ ,\ \lambda_1 = U_{cont} - a_{\text{int}}h_n\ ,\ \lambda_2 = \lambda_3 = U_{cont}; \quad (22)$$

$$\lambda_4 = U_{cont} + a_{\text{int}}h_n. \quad (23)$$

The jumps in the conserved variables, necessary to the construction of the [2] TVD dissipation function, are given by:

$$\Delta e = V_{\text{int}}\left(e_R - e_L\right),\ \Delta\rho = V_{\text{int}}\left(\rho_R - \rho_L\right); \quad (24)$$

$$\Delta(\rho u) = V_{\text{int}}\left[(\rho u)_R - (\rho u)_L\right] \text{ and } \Delta(\rho v) = V_{\text{int}}\left[(\rho v)_R - (\rho v)_L\right]. \quad (25)$$

The α vectors to the (i+1/2,j) interface are calculated by the following expressions:

$$\alpha_1 = 0.5(aa - bb),\ \alpha_2 = \Delta\rho - aa\ ,\ \alpha_3 = cc; \quad (26)$$

$$\alpha_4 = 0.5(aa + bb), \quad (27)$$

with:

$$aa = (\gamma-1)\Big/a_{\text{int}}^2\left[\Delta e + 0.5\left(u_{\text{int}}^2 + v_{\text{int}}^2\right)\Delta\rho - u_{\text{int}}\Delta(\rho u) - v_{\text{int}}\Delta(\rho v)\right]; \quad (28)$$

$$bb = 1/a_{\text{int}}\left[h_x^{'}\Delta(\rho u) - \left(h_x^{'}u_{\text{int}} + h_y^{'}v_{\text{int}}\right)\Delta\rho + h_y^{'}\Delta(\rho v)\right]; \quad (29)$$

$$cc = h_x^{'}\Delta(\rho v) + \left(h_y^{'}u_{\text{int}} - h_x^{'}v_{\text{int}}\right)\Delta\rho - h_y^{'}\Delta(\rho u); \quad (30)$$

$$h_x^{'} = h_x/h_n \quad \text{and} \quad h_y^{'} = h_y/h_n\ . \quad (31)$$

The [2] TVD dissipation function is constructed using the right eigenvector matrix of the Jacobian matrix in the normal direction to the flux face. This matrix is defined in [12].

In this work, according to [6], five different limiters are implemented which incorporate the TVD properties to the original [2] scheme. The limited dissipation function Q is defined to the five options as:

$$Q(r^-, r^+) = \text{minmod}\left(1, r^-\right) + \text{minmod}\left(1, r^+\right) - 1; \quad (32)$$

$$Q(r^-, r^+) = \text{minmod}\left(1, r^-, r^+\right); \quad (33)$$

$$Q(r^-, r^+) = \text{minmod}\left[2, 2r^-, 2r^+, 0.5\left(r^- + r^+\right)\right]; \quad (34)$$

$$Q(r^-, r^+) = \max\left[0, \min\left(2r^-, 1\right), \min\left(r^-, 2\right)\right] + \max\left[0, \min\left(2r^+, 1\right), \min\left(r^+, 2\right)\right] - 1 \quad (35)$$

$$Q(r^-, r^+) = \frac{r^- + \left|r^-\right|}{1 + r^-} + \frac{r^+ + \left|r^+\right|}{1 + r^+} - 1, \quad (36)$$

where:

$$\left(r^-_{i+1/2,j}\right) = \alpha^l_{i-1/2,j}\Big/\alpha^l_{i+1/2,j} \quad \text{and} \quad \left(r^+_{i+1/2,j}\right) = \alpha^l_{i+3/2,j}\Big/\alpha^l_{i+1/2,j}, \quad (37)$$

"l" assuming values from 1 to 4. Equations (32)-(34) are referenced by this author as minmod1 (Min1), minmod2 (Min2) and minmod3 (Min3), respectively. Equation (35) is referred in the CFD ("Computational Fluid Dynamics") literature as the "Super Bee" limiter due to [13] and Eq. (31) is the Van Leer limiter due to[14].

The [2] TVD dissipation function is finally constructed by the following matrix-vector product:

$$\left\{D_{LW}\right\}_{i+1/2,j} = \left[R\right]_{i+1/2,j}\left\{\left[\Delta t_{i,j}\lambda^2 Q + |\lambda|(1-Q)\right]\alpha\right\}_{i+1/2,j}. \quad (38)$$

The complete numerical flux vector to the (i+1/2,j) interface is described by:

$$F^{(l)}_{i+1/2,j} = \left(E^{(l)}_{\text{int}}h_x + F^{(l)}_{\text{int}}h_y\right)V_{\text{int}} - 0.5D^{(l)}_{LW}, \quad (39)$$

with:

$$E^{(l)}_{\text{int}} = 0.5\left(E^{(l)}_R + E^{(l)}_L\right) \quad \text{and} \quad F^{(l)}_{\text{int}} = 0.5\left(F^{(l)}_R + F^{(l)}_L\right). \quad (40)$$

The time integration follows the time splitting method, first order accurate, which divides the integration in two steps, each one associated with a specific spatial direction. In the initial step, it is possible to write for the ξ direction:

$$\Delta Q_{i,j}^{*} = -\Delta t_{i,j} / V_{i,j} \left(F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n} \right); \quad (41)$$

$$Q_{i,j}^{*} = Q_{i,j}^{n} + \Delta Q_{i,j}^{*}; \quad (42)$$

and at the end step, η direction:

$$\Delta Q_{i,j}^{n+1} = -\Delta t_{i,j} / V_{i,j} \left(F_{i,j+1/2}^{*} - F_{i,j-1/2}^{*} \right); \quad (43)$$

$$Q_{i,j}^{n+1} = Q_{i,j}^{*} + \Delta Q_{i,j}^{n+1}. \quad (44)$$

5 [3] FCT Algorithm

The [3] algorithm can be understood as composed of two phases: (1) Generation of an initial solution using a simple and practical base algorithm; (2) Perform the anti-dissipative phase in the initial solution.

To the base algorithm required in the first phase of the [3] FCT scheme, the authors recommend the [15] algorithm. Hence, the base algorithm can be described in finite volumes by:

Predictor step:

$$Q_{init} = Q_{i,j}^{n}; \quad (45)$$

$$C_{i,j}^{n} = F_{i,j}^{n} + F_{i+1,j}^{n} + F_{i,j+1}^{n} + F_{i,j}^{n}; \quad (46)$$

$$Q_{i,j}^{*} = Q_{i,j}^{n} - \Delta t_{i,j} / V_{i,j} \times C_{i,j}^{n}, \quad (47)$$

where:

$$F_{i,j}^{n} = \begin{Bmatrix} \rho q_s \\ \rho u q_s + p S_x \\ \rho v q_s + p S_y \\ (e+p) q_s \end{Bmatrix} \quad \text{and} \quad q_s = u S_x + v S_y, \quad (48)$$

with S_x and S_y described by Eqs. (6)-(7).

Corrector step:

$$C_{i,j}^{*} = F_{i,j-1}^{*} + F_{i,j}^{*} + F_{i,j}^{*} + F_{i-1,j}^{*}; \quad (49)$$

$$Q_{i,j}^{n+1} = 0.5 \left(Q_{init} + Q_{i,j}^{*} - \Delta t_{i,j} / V_{i,j} \times C_{i,j}^{*} \right). \quad (50)$$

The residual is defined as the arithmetical average between $-C_{i,j}^{n} / V_{i,j}$ and $-C_{i,j}^{*} / V_{i,j}$. The base solution is therefore:

$$Q_{i,j}^{**} = Q_{i,j}^{n+1} = Q^{Mac}. \quad (51)$$

Defined the base solution, the next step is construct the anti-dissipative phase. The anti-dissipative phase follows the following sequence of steps:

(1) Define the constants: $\eta_0 = 1/6$, $\eta_1 = 1/3$ and $\eta_2 = -1/6$;

(2) Assume $Q_{init} = Q_{i,j}^{n}$;

(3) Call the MacCormack algorithm to determine the base solution;

(4) Define u_{int} and v_{int} at the four interfaces of a computational cell. *u* and *v* are determined using the values of Q_{init} and u_{int} and v_{int} are obtained by arithmetical average between the cell under study and its neighbors;

(5) Define the speed of sound at the four neighbors;

(6) Determine the maximum eigenvalue of the Euler equations at the four interfaces:

$$q_{\text{int}}^{\max} = \sqrt{u_{\text{int}}^2 + v_{\text{int}}^2} + a; \quad (52)$$

(7) Determine $v_{\text{int}} = \eta_0 + \eta_1 \times \left(q_{\text{int}}^{\max} \Delta t_{i,j} \right)^2$ to the four interfaces;

(8) Calculate the diffusive flux vector:

$$F_D^{(i,j-1/2)} = v_{(i,j-1/2)} \left(Q_{i,j}^{\text{int}} - Q_{i,j-1}^{\text{int}} \right); \quad (53)$$

$$F_D^{(i+1/2,j)} = v_{(i+1/2,j)} \left(Q_{i+1,j}^{\text{int}} - Q_{i,j}^{\text{int}} \right); \quad (54)$$

$$F_D^{(i,j+1/2)} = v_{(i,j+1/2)} \left(Q_{i,j+1}^{\text{int}} - Q_{i,j}^{\text{int}} \right); \quad (55)$$

$$F_D^{(i-1/2,j)} = v_{(i-1/2,j)} \left(Q_{i,j}^{\text{int}} - Q_{i-1,j}^{\text{int}} \right). \quad (56)$$

(9) Diffuse the solution:

$$Q_{i,j}^{***} = Q_{i,j}^{**} + \left[F_D^{(i+1/2,j)} - F_D^{(i-1/2,j)} + F_D^{(i,j+1/2)} - F_D^{(i,j-1/2)} \right] \quad (57)$$

(10) Do $Q_{i,j}^{n} = Q_{i,j}^{***}$ and imposes the boundary conditions;

(11) Do $Q_{i,j}^{***} = Q_{i,j}^{n}$ and calculate the anti-diffusive flux – FCT;

(12) Repeat steps (4)-(6);

(13) Determine $\mu_{\text{int}} = \eta_0 + \eta_2 \times \left(q_{\text{int}}^{\max} \Delta t_{i,j} \right)^2$;

(14) Calculate the anti-diffusive flux:

$$F_{AD}^{(i,j-1/2)} = \mu_{(i,j-1/2)}\left(Q_{i,j}^{**} - Q_{i,j-1}^{**}\right); \quad (58)$$

$$F_{AD}^{(i+1/2,j)} = \mu_{(i+1/2,j)}\left(Q_{i+1,j}^{**} - Q_{i,j}^{**}\right); \quad (59)$$

$$F_{AD}^{(i,j+1/2)} = \mu_{(i,j+1/2)}\left(Q_{i,j+1}^{**} - Q_{i,j}^{**}\right); \quad (60)$$

$$F_{AD}^{(i-1/2,j)} = \mu_{(i-1/2,j)}\left(Q_{i,j}^{**} - Q_{i-1,j}^{**}\right). \quad (61)$$

(15) Determine the corrective flux at each interface:

Interface (i,j-1/2):

$$\Delta Q_{i,j-3/2} = Q_{i,j-1}^{**} - Q_{i,j-2}^{***}; \quad (62)$$

$$\Delta Q_{i,j+1/2} = Q_{i,j+1}^{**} - Q_{i,j}^{***}. \quad (63)$$

The corrective anti-diffusive or anti-dissipative flux is calculated in vector form by:

$$F_{cad}^{(i,j-1/2)} = signal \times MAX\left[0.0, MIN\left(\Delta Q_{i,j-3/2} \times signal, \left|F_{AD}^{(i,j-1/2)}\right|, \Delta Q_{i,j+1/2} \times signal\right)\right], \quad (64)$$

where: *signal* is a vector assuming values 1.0 or -1.0: If $F_{AD,l}^{(i,j-1/2)} \geq 0.0$ => $signal_l$ = 1.0 or -1.0 otherwise; "*l*" is an index referring to the vector components.

Interface (i+1/2,j):

$$\Delta Q_{i-1/2,j} = Q_{i,j}^{**} - Q_{i-1,j}^{***}; \quad (65)$$

$$\Delta Q_{i+3/2,j} = Q_{i+2,j}^{**} - Q_{i+1,j}^{***}. \quad (66)$$

The corrective anti-diffusive or anti-dissipative flux is calculated in vector form by:

$$F_{cad}^{(i+1/2,j)} = signal \times MAX\left[0.0, MIN\left(\Delta Q_{i-1/2,j} \times signal, \left|F_{AD}^{(i+1/2,j)}\right|, \Delta Q_{i+3/2,j} \times signal\right)\right], \quad (67)$$

where: *signal* is a vector assuming values 1.0 or -1.0: If $F_{AD,l}^{(i+1/2,j)} \geq 0.0$ => $signal_l$ = 1.0 or -1.0 otherwise; "*l*" is an index referring to the vector components.

Interface (i,j+1/2):

$$\Delta Q_{i,j-1/2} = Q_{i,j}^{**} - Q_{i,j-1}^{***}; \quad (68)$$

$$\Delta Q_{i,j+3/2} = Q_{i,j+2}^{**} - Q_{i,j+1}^{***}; \quad (69)$$

The corrective anti-diffusive or anti-dissipative flux is calculated in vector form by:

$$F_{cad}^{(i,j+1/2)} = signal \times MAX\left[0.0, MIN\left(\Delta Q_{i,j-1/2} \times signal, \left|F_{AD}^{(i,j+1/2)}\right|, \Delta Q_{i,j+3/2} \times signal\right)\right], \quad (70)$$

where: *signal* is a vector assuming values 1.0 or -1.0: If $F_{AD,l}^{(i,j+1/2)} \geq 0.0$ => $signal_l$ = 1.0 or -1.0 otherwise; "*l*" is an index referring to the vector components.

Interface (i-1/2,j):

$$\Delta Q_{i-3/2,j} = Q_{i-1,j}^{**} - Q_{i-2,j}^{***}; \quad (71)$$

$$\Delta Q_{i+1/2,j} = Q_{i+1,j}^{**} - Q_{i,j}^{***}. \quad (72)$$

The corrective anti-diffusive or anti-dissipative flux is calculated in vector form by:

$$F_{cad}^{(i-1/2,j)} = signal \times MAX\left[0.0, MIN\left(\Delta Q_{i-3/2,j} \times signal, \left|F_{AD}^{(i-1/2,j)}\right|, \Delta Q_{i+1/2,j} \times signal\right)\right], \quad (73)$$

where: *signal* is a vector assuming values 1.0 or -1.0: If $F_{AD,l}^{(i-1/2,j)} \geq 0.0$ => $signal_l$ = 1.0 or -1.0 otherwise; "*l*" is an index referring to the vector components.

(16) Determine the new value of $Q_{i,j}$:

The new value of $Q_{i,j}$ is obtained from the following scheme:

$$Q_{i,j}^{n+1} = Q_{i,j}^{***} - \left[F_{cad}^{(i+1/2,j)} - F_{cad}^{(i-1/2,j)} + F_{cad}^{(i,j+1/2)} - F_{cad}^{(i,j-1/2)}\right]. \quad (74)$$

6 [4] Algorithm

The [4] scheme is a symmetrical algorithm. Its discretization, on a finite volume context, is centered and artificial dissipation is needed to guarantee convergence to the steady state solution. Two dissipation operators are studied in this work: a linear and a non-linear, both defined according to [16] work.

On a finite volume context, it is necessary to determine the convective flux vectors E_e and F_e written to a generalized system at the centroid of the cell under study. Hence, the metric terms are defined as arithmetical average between the metric

terms in each flux interface. With this in mind, one can write the convective flux vectors as:

$$E = V_{average}\begin{Bmatrix} \rho v_n \\ \rho u v_n + p h_x \\ \rho v v_n + p h_y \\ (e+p) v_n \end{Bmatrix}, \quad (75)$$

where, for instance, $V_{average} = 0.5\left(V_{i-1/2,j} + V_{i+1/2,j}\right)$ to the (i+1/2,j) flux interface. The same analysis is valid to the *F* flux interface, only changing the metric terms at the cell(i,j).

6.1 Dissipation Models

Two types of dissipation models are analyzed in this work: one linear and one non-linear, both isotropic. They were suggested by [16].

6.1.1 Linear Dissipation Model

The linear dissipation model of [16] distributes dissipation of uniform way in the field. As the dissipation is provided uniformly in each direction, without have a weighting coefficient to better evaluate the effects of non-linearity in the field, it is isotropic. The operator is a fourth order in the interior domain, being second order at the boundaries. To the internal domain, its format is described below:

$$D^L_{i,j} = -ee \times V_{i,j} \times \left[\left(Q^n_{i-2,j} - 4Q^n_{i-1,j} + 6Q^n_{i,j} - 4Q^n_{i+1,j} + Q^n_{i+2,j}\right) + \left(Q^n_{i,j-2} - 4Q^n_{i,j-1} + 6Q^n_{i,j} - 4Q^n_{i,j+1} + Q^n_{i,j+2}\right)\right] \quad (76)$$

To the boundaries, considering for instance the entrance boundary ("i" direction), one has:

$$D^L_{i,j} = ee \times V_{i,j} \times \left(Q^n_{i-1,j} - 2Q^n_{i,j} + Q^n_{i+1,j}\right), \quad (77)$$

with an equivalent expression to the "j" direction.

6.1.2 Non-Linear Dissipation Model

The non-linear model is employed specifically to determine non-linear instabilities, like shock waves, and is also isotropic. It needs the determination of a pressure sensor to detect shock waves. Hence, for the ξ direction, one has:

$$\nu^{\xi}_{i,j} = \left|p_{i-1,j} - 2p_{i,j} + p_{i+1,j}\right| \Big/ \left(p_{i-1,j} + 2p_{i,j} + p_{i+1,j}\right), \quad (78)$$

with an equivalent expression obtained by the η direction. The next step is to define ε's terms that determine the amount of dissipation to be provided by the dissipation model. To the ξ direction, it is possible to write:

$$\varepsilon^{(2)}_{i,j} = k^{(2)} \times \Delta t_{i,j} \times MAX\left(\nu^{\xi}_{i-1,j}, \nu^{\xi}_{i,j}, \nu^{\xi}_{i+1,j}\right); \quad (79)$$

$$\varepsilon^{(4)}_{i,j} = MAX\left(0.0, k^{(4)}\Delta t_{i,j} - \varepsilon^{(2)}_{i,j}\right); \quad (80)$$

an equivalent expression to η is valid. After that, the spectral ratio of the Euler equations in a generalized coordinate system is determined:

$$\sigma_{i,j} = \left|v^{\xi}_n\right| + a_{i,j} h^{\xi}_n + \left|v^{\eta}_n\right| + a_{i,j} h^{\eta}_n, \quad (81)$$

with:

$$v^{\xi}_n = \left(u h^{\xi}_x\right)_{i,j} + \left(v h^{\xi}_y\right)_{i,j} \quad \text{and} \quad v^{\eta}_n = \left(u h^{\eta}_x\right)_{i,j} + \left(v h^{\eta}_y\right)_{i,j}, \quad (82)$$

h_x, h_y and h_n defined according to Eq. (17). The next step consists in determine the non-linear dissipation operators. Considering the ξ direction, one has:

$$\omega_1 = 0.5 \times \left(\sigma_{i,j} V_{i,j} + \sigma_{i+1,j} V_{i+1,j}\right); \quad (83)$$

$$\omega_2 = 0.5 \times \left(\sigma_{i,j} V_{i,j} + \sigma_{i-1,j} V_{i-1,j}\right). \quad (84)$$

To the internal domain, it is possible to write:

$$D^{NL}_{i,j} = \omega_1 \times \left[\varepsilon^{(2)}_{i,j} \times \left(Q^n_{i+1,j} - Q^n_{i,j}\right) - \varepsilon^{(4)}_{i,j} \times \left(Q^n_{i+2,j} - 3Q^n_{i+1,j} + 3Q^n_{i,j} + - Q^n_{i-1,j}\right)\right] - \omega_2 \times \left[\varepsilon^{(2)}_{i-1,j} \times \left(Q^n_{i,j} - Q^n_{i-1,j}\right) - \varepsilon^{(4)}_{i-1,j}\left(Q^n_{i+1,j} - 3Q^n_{i,j} + 3Q^n_{i-1,j} + - Q^n_{i-2,j}\right)\right] \quad (85)$$

To the boundaries, one has:

$$D^{NL}_{i,j} = \omega_1 \times \left[\varepsilon^{(2)}_{i,j} \times \left(Q^n_{i+1,j} - Q^n_{i,j}\right)\right] - \omega_2 \times \left[\varepsilon^{(2)}_{i-1,j} \times \left(Q^n_{i,j} - Q^n_{i-1,j}\right)\right]. \quad (86)$$

6.2 Numerical Algorithm

The residual of the [4] scheme is defined as:

$$R^n_{i,j} = \left[0.5 \times \left(E^n_{i+1,j} - E^n_{i-1,j} + F^n_{i,j+1} - F^n_{i,j-1}\right) - D^L_{i,j}\right] \Big/ V_{i,j}, \quad (87)$$

using the linear dissipation operator Eqs. (76)-(77).

$$R^n_{i,j} = \left[0.5 \times \left(E^n_{i+1,j} - E^n_{i-1,j} + F^n_{i,j+1} - F^n_{i,j-1}\right) - D^{NL}_{i,j} / \Delta t_{i,j}\right] \Big/ V_{i,j}, \quad (88)$$

using the non-linear dissipation operator Eqs. (85-86). The time marching method is the first order forward Euler one and results in:

$$Q_{i,j}^{n+1} = Q_{i,j}^{n} - \Delta t_{i,j} \times R_{i,j}^{n}, \quad (89)$$

7 [5] Algorithm

The [5] algorithm is a hybrid scheme that alternate between an explicit and implicit methods, depending of the employed CFL number. For CFL number less than 1.0, the scheme employs the [15] algorithm and when the CFL number is more than 1.0, the scheme is a two-diagonal implicit algorithm, where the RHS ("Right Hand Side") is constructed by the [15] scheme. Hence, the first step is to calculate the [15] convective flux balance. This balance is described by Eqs. (45)-(50).

The [5] algorithm starts defining the explicit correction in the predictor step:

$$dq_e^* = -\Delta t_{i,j} / V_{i,j} \times C_{i,j}^{n}. \quad (90)$$

After that, it calculates the implicit RHS to the ξ direction:

$$W = \left\{dq_e^*\right\}_{i,j} + \Delta t_{i,j} \times \left[A^{Mac}\right]_{+1/2,j} \times \left\{dq_i^*\right\}_{i+1,j}, \quad (91)$$

where the matrix $\left[A^{Mac}\right]$ is defined as:

$$\left[A^{Mac}\right] = [T] \times [DA] \times \left[T^{-1}\right] \quad (92)$$

with:

T and T^{-1} being similarity matrices and DA being a diagonal matrix composed of the Euler eigenvalues in ξ direction; Expressions to T and T^{-1} are given as follows:

$$T = \begin{bmatrix} 1 & 0 & \alpha & \alpha \\ u_{\text{int}} & h_y' \rho_{\text{int}} & \alpha\left(u_{\text{int}} + h_x' a_{\text{int}}\right) & \alpha\left(u_{\text{int}} - h_x' a_{\text{int}}\right) \\ v_{\text{int}} & -h_x' \rho_{\text{int}} & \alpha\left(v_{\text{int}} + h_y' a_{\text{int}}\right) & \alpha\left(v_{\text{int}} - h_y' a_{\text{int}}\right) \\ \dfrac{\varphi^2}{\gamma - 1} & \rho_{\text{int}}\left(h_y' u_{\text{int}} - h_x' v_{\text{int}}\right) & \alpha\left(\dfrac{\varphi^2 + a_{\text{int}}^2}{\gamma - 1} + a_{\text{int}}\tilde{\theta}\right) & \alpha\left(\dfrac{\varphi^2 + a_{\text{int}}^2}{\gamma - 1} - a_{\text{int}}\tilde{\theta}\right) \end{bmatrix}; \quad (93)$$

$$\alpha = \rho_{\text{int}} / \left(\sqrt{2} a_{\text{int}}\right), \beta = 1 / \left(\sqrt{2} \rho_{\text{int}} a_{\text{int}}\right), \varphi^2 = (\gamma - 1)\frac{u_{\text{int}}^2 + v_{\text{int}}^2}{2}, \quad (94)$$

$$\tilde{\theta} = h_x' u_{\text{int}} + h_y' v_{\text{int}}; \quad (95)$$

$$T^{-1} = \begin{bmatrix} 1 - \dfrac{\varphi^2}{a_{\text{int}}^2} & (\gamma - 1)\dfrac{u_{\text{int}}}{a_{\text{int}}^2} & (\gamma - 1)\dfrac{v_{\text{int}}}{a_{\text{int}}^2} & -\dfrac{\gamma - 1}{a_{\text{int}}^2} \\ -\dfrac{h_y' u_{\text{int}} - h_x' v_{\text{int}}}{\rho_{\text{int}}} & \dfrac{h_y'}{\rho_{\text{int}}} & -\dfrac{h_x'}{\rho_{\text{int}}} & 0 \\ \beta\left(\varphi^2 - a_{\text{int}}\tilde{\theta}\right) & \beta\left[h_x' a_{\text{int}} - (\gamma - 1)u_{\text{int}}\right] & \beta\left[h_y' a_{\text{int}} - (\gamma - 1)v_{\text{int}}\right] & \beta(\gamma - 1) \\ \beta\left(\varphi^2 + a_{\text{int}}\tilde{\theta}\right) & -\beta\left[h_x' a_{\text{int}} + (\gamma - 1)u_{\text{int}}\right] & -\beta\left[h_y' a_{\text{int}} + (\gamma - 1)v_{\text{int}}\right] & \beta(\gamma - 1) \end{bmatrix}. \quad (96)$$

The properties defined at interface are calculated by arithmetical average. The normalized metric terms are defined in Section 4. The expression to DA is as follows:

$$DA = \begin{bmatrix} \left|v_{n\text{int}}\right| & 0 & 0 & 0 \\ 0 & \left|v_{n\text{int}}\right| & 0 & 0 \\ 0 & 0 & \left|v_{n\text{int}} + a_{\text{int}} h_n\right| & 0 \\ 0 & 0 & 0 & \left|v_{n\text{int}} - a_{\text{int}} h_n\right| \end{bmatrix}, \quad (97)$$

with $v_{n_{\text{int}}} = u_{\text{int}} h_x + v_{\text{int}} h_y$ being the normal velocity at interface. The metric terms are defined in Section 4. Continuing the algorithm, the next step is to determine the solution of the two-diagonal system. Defining a vector {*Prod*} and a matrix [*Term*], one has:

$$\{\text{Prod}\}_{i,j} = \left[T^{-1}\right]_{i+1/2,j} \times \{W\}_{i,j}; \quad (98)$$

$$[Term]_{i,j} = [I] + \Delta t_{i,j} \times [DA]_{i+1/2,j}. \quad (99)$$

Now, it is possible to determine the vector *Y* as:

$$\{Y\}_{i,j} = \left[Term^{-1}\right]_{i,j} \times \{\text{Prod}\}_{i,j}, \quad (100)$$

where:

$\left[Term^{-1}\right]_{i,j}$ is a diagonal matrix of easy inversion.

The implicit correction is obtained by the following matrix-vector product:

$$\left\{dq_i^*\right\}_{i,j} = [T]_{i+1/2,j} \times \{Y\}_{i,j} \quad (101)$$

As the sweep along the mesh starts at the i = imax-1, it is adopted that $\left\{dq_i^*\right\}_{i\max,j} = 0.0$ at this cell, which is a ghost cell. Finished the ξ flux, it calculates now the implicit RHS to the η direction:

$$W = \left\{dq_i^*\right\}_{i,j} + \Delta t_{i,j} \times \left[B^{Mac}\right]_{,j+1/2} \times \left\{dq_i^{**}\right\}_{i,j+1}, \quad (102)$$

where the matrix $\left[B^{Mac}\right]$ is defined as:

$$\left[B^{Mac}\right] = [T] \times [DB] \times \left[T^{-1}\right], \quad (103)$$

with T and T^{-1} being similarity matrices and DB being a diagonal matrix composed of the Euler eigenvalues in the η direction. The same matrices defined by Eqs. (93-97) are employed considering the η direction. Continuing the algorithm, the next step is to determine the solution of the two-diagonal system. Defining a vector {*Prod*} and a matrix [*Term*], one has:

$$\{\mathrm{Prod}\}_{i,j} = \left[T^{-1}\right]_{,j+1/2} \times \{W\}_{i,j}; \quad (104)$$

$$[Term]_{i,j} = [I] + \Delta t_{i,j} \times [DB]_{i,j+1/2}. \quad (105)$$

Now, it is possible to determine the vector *Y* as:

$$\{Y\}_{i,j} = \left[Term^{-1}\right]_{,j} \times \{\mathrm{Prod}\}_{i,j}, \quad (106)$$

The implicit correction is obtained by the following matrix-vector product:

$$\left\{dq_i^{**}\right\}_{i,j} = [T]_{i,j+1/2} \times \{Y\}_{i,j} \quad (107)$$

As the sweep along the mesh starts at the j = jmax-1, it is adopted that $\left\{dq_i^{**}\right\}_{i,j\max} = 0.0$ at this cell, which is a ghost cell. Finally, the predictor correction is estimated as $\{\Delta Q_p\} = \left\{dq_i^{**}\right\}_{i,j}$ and the new vector of conserved variables is determined as:

$$Q_{i,j}^* = Q_{i,j}^n + \left(\Delta Q_p\right)_{i,j}. \quad (108)$$

This finishes the predictor step. Now, it is necessary to determine the correction in the corrector step. It is done as follows:

$$dq_e^* = -\Delta t_{i,j} / V_{i,j} \times C_{i,j}^*. \quad (109)$$

After that, it calculates the implicit RHS to the ξ direction:

$$W = \left\{dq_e^*\right\}_{i,j} + \Delta t_{i,j} \times \left[A^{Mac}\right]_{-1/2,j} \times \left\{dq_i^*\right\}_{i-1,j}, \quad (110)$$

where the matrix $\left[A^{Mac}\right]$ is defined as done in the predictor step, Eq. (92), calculated now in the (i-1/2,j) interface. Defining a vector {*Prod*} and a matrix [*Term*], one has:

$$\{\mathrm{Prod}\}_{i,j} = \left[T^{-1}\right]_{-1/2,j} \times \{W\}_{i,j}; \quad (111)$$

$$[Term]_{i,j} = [I] + \Delta t_{i,j} \times [DA]_{i-1/2,j}. \quad (112)$$

Now, it is possible to determine the vector *Y* as:

$$\{Y\}_{i,j} = \left[Term^{-1}\right]_{,j} \times \{\mathrm{Prod}\}_{i,j}. \quad (113)$$

The implicit correction is obtained by the following matrix-vector product:

$$\left\{dq_i^*\right\}_{i,j} = [T]_{i-1/2,j} \times \{Y\}_{i,j}. \quad (114)$$

As the sweep along the mesh starts at the i = 1, it is adopted that $\left\{dq_i^*\right\}_{0,j} = 0.0$ at this cell, which is a ghost cell. Finished the ξ flux, it calculates now the implicit RHS to the η direction:

$$W = \left\{dq_i^*\right\}_{i,j} + \Delta t_{i,j} \times \left[B^{Mac}\right]_{,j-1/2} \times \left\{dq_i^{**}\right\}_{i,j-1}, \quad (115)$$

where the matrix $\left[B^{Mac}\right]$ is defined as done in the predictor step [Eq. (103)], calculated now in the (i,j-1/2) interface. Continuing the algorithm, the next step is to determine the solution of the two-diagonal system. Defining a vector {*Prod*} and a matrix [*Term*], one has:

$$\{\mathrm{Prod}\}_{i,j} = \left[T^{-1}\right]_{,j-1/2} \times \{W\}_{i,j}; \quad (116)$$

$$[Term]_{i,j} = [I] + \Delta t_{i,j} \times [DB]_{i,j-1/2}. \quad (117)$$

Now, it is possible to determine the vector *Y* as:

$$\{Y\}_{i,j} = \left[Term^{-1}\right]_{,j} \times \{\mathrm{Prod}\}_{i,j}. \quad (118)$$

The implicit correction is obtained by the following matrix-vector product:

$$\left\{dq_i^{**}\right\}_{i,j} = [T]_{i,j-1/2} \times \{Y\}_{i,j}. \quad (119)$$

As the sweep along the mesh starts at the j = 1, it is adopted that $\left\{dq_i^{**}\right\}_{i,0} = 0.0$ at this cell, which is a ghost cell. Finally, the corrector value is estimated as $\{\Delta Q_c\} = \left\{dq_i^{**}\right\}_{i,j}$ and the new vector of conserved variables is determined as:

$$Q_{i,j}^{n+1} = 0.5 \times \left[Q_{init} + Q_{i,j}^* + \left(\Delta Q_c\right)_{i,j}\right]. \quad (120)$$

The residual is defined as the arithmetical average between $\left(\Delta Q_p\right)_{i,j}$ and $\left(\Delta Q_c\right)_{i,j}$.

8 Spatially Variable Time Step

The basic idea of this procedure consists in keeping constant the CFL number in all calculation domain, allowing, hence, the use of appropriated time steps to each specific mesh region during the convergence process. Hence, according to the definition of the CFL number, it is possible to write:

$$\Delta t_{i,j} = CFL\left(\Delta s\right)_{i,j} / c_{i,j}\,, \qquad (121)$$

where: CFL is the "Courant-Friedrichs-Lewy" number to provide numerical stability to the scheme; $c_{i,j} = \left[\left(u^2+v^2\right)^{0.5} + a\right]_{i,j}$ is the maximum characteristic speed of information propagation in the calculation domain; and $\left(\Delta s\right)_{i,j}$ is a characteristic length of information transport. On a finite volume context, $\left(\Delta s\right)_{i,j}$ is chosen as the minor value found between the minor centroid distance, involving the (i,j) cell and a neighbor, and the minor cell side length.

9 Initial and Boundary Conditions

9.1 Initial Conditions

Values of freestream flow are adopted for all properties as initial condition, in the whole calculation domain, to the studied problem in this work ([17-18]):

$$Q_\infty = \left\{1 \quad M_\infty \cos\theta \quad M_\infty \sin\theta \quad \left[1/\left[\gamma(\gamma-1)\right] + 0.5M_\infty^2\right]\right\}^T, \qquad (122)$$

where M_∞ represents the freestream Mach number and θ is the flow attack angle.

9.2 Boundary Conditions

The boundary conditions are basically of three types: solid wall, entrance and exit. These conditions are implemented in special cells named ghost cells.

(a) Wall condition: This condition imposes the flow tangency at the solid wall. This condition is satisfied considering the wall tangent velocity component of the ghost volume as equals to the respective velocity component of its real neighbor cell. At the same way, the wall normal velocity component of the ghost cell is equaled in value, but with opposite signal, to the respective velocity component of the real neighbor cell.

The pressure gradient normal to the wall is assumed be equal to zero, following an inviscid formulation. The same hypothesis is applied to the temperature gradient normal to the wall, considering adiabatic wall. The ghost volume density and pressure are extrapolated from the respective values of the real neighbor volume (zero order extrapolation), with these two conditions. The total energy is obtained by the state equation of a perfect gas.

(b) Entrance condition:

(b.1) Subsonic flow: Three properties are specified and one is extrapolated, based on analysis of information propagation along characteristic directions in the calculation domain [18]. In other words, three characteristic directions of information propagation point inward the computational domain and should be specified. Only the characteristic direction associated to the "(q_{normal}-a)" velocity cannot be specified and should be determined by interior information of the calculation domain. Pressure was the extrapolated variable to the present problem. Density and velocity components had their values determined from the freestream flow properties. The total energy per unity fluid volume is determined by the state equation of a perfect gas.

(b.2) Supersonic flow: All variables are fixed with their freestream flow values.

(c) Exit condition:

(c.1) Subsonic flow: Three characteristic directions of information propagation point outward the computational domain and should be extrapolated from interior information [18]. The characteristic direction associated to the "(q_{normal}-a)" velocity should be specified because it penetrates the calculation domain. In this case, the ghost volume's pressure is specified by its freestream value. Density and velocity components are extrapolated and the total energy is obtained by the state equation of a perfect gas.

(c.2) Supersonic flow: All variables are extrapolated from the interior domain due to the fact that all four characteristic directions of information propagation of the Euler equations point outward the calculation domain and, with it, nothing can be fixed.

10 Results

Tests were performed in a notebook with processor Pentium Dual Core, 2.3GHz of clock and 3,0 Gbytes of RAM memory. The criterion adopted in this work to obtain convergence considers a reduction of 4 orders in the magnitude of the maximum residual in the domain, a typical criterion in the CFD community. In the compression corner problem, the entrance angle was set equal to 0.0°.

Figure 1 shows the compression corner configuration, whereas the Figure 2 exhibits the compression corner mesh. This mesh is composed of 3,381 rectangular cells and 3,500 nodes or, in finite differences, 70x50 points. The freestream Mach number was adopted with the value of 3.0, a moderate supersonic flow, where the perfect gas hypothesis is valid.

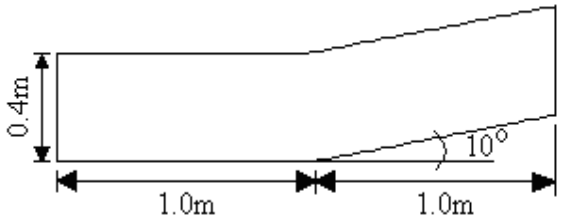

Figure 2. Compression corner configuration.

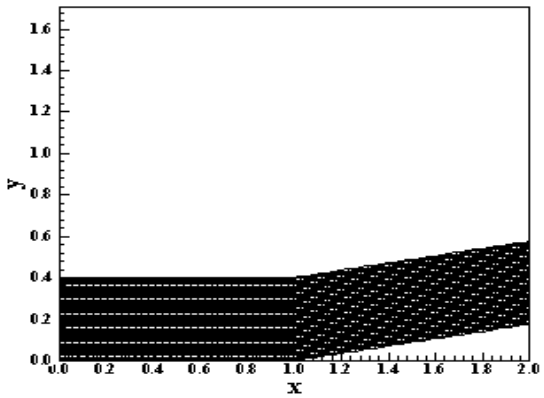

Figure 3. Compression corner mesh.

Figures 4 to 13 shows the pressure contours obtained by the [1] scheme, the [2] TVD scheme, in its five variants, the [3] scheme, the [4] scheme, in its two variants, and the [5] scheme. For convention, the present author adopts the following abbreviation to the schemes: [1] (LF), [2] using minmod1 limiter (LW-Min1), [2] using minmod2 limiter (LW-Min2), [2] using minmod3 limiter (LW-Min3), [2] using Super Bee limiter (LW-SB), [2] using Van Leer limiter (LW-VL), [3] (BB), [4] using the linear dissipation operator (BW-L), [4] using the nonlinear dissipation operator (BW-NL), and [5] (M). Examining these figures, the most severe pressure field is that obtained by the LW-SB scheme. The lowest shock wave thickness is observed in the LW-SB, BB and BW-NL schemes. The LW-SB, BW-L and M present pressure oscillations in their solutions, which damage severely their behavior.

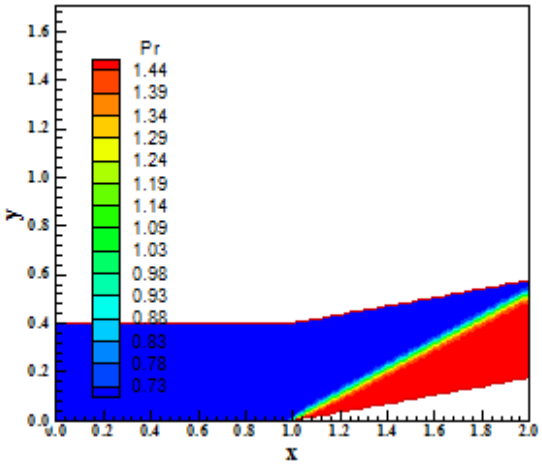

Figure 4. Pressure contours (LF).

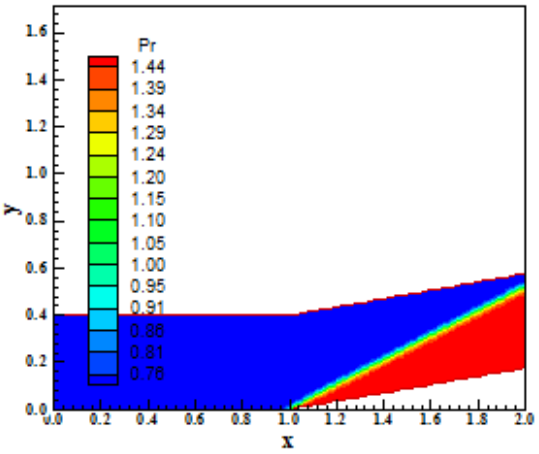

Figure 5. Pressure contours (LW-Min1).

Figure 14 exhibits the wall pressure distributions of all schemes tested in this work. They are compared with the oblique shock wave theoretical solution. It is possible to highlight that the best pressure distribution, close to the theoretical result is obtained by the BW-NL scheme. Figure 15 shows the same wall pressure distributions, plotted now with symbols to illustrate in number of points necessary to capture the shock discontinuity. The BW-NL scheme captured the shock discontinuity using two cells. Other schemes also capture the shock discontinuity using two cells, but the best pressure distribution was due to the BW-NL.

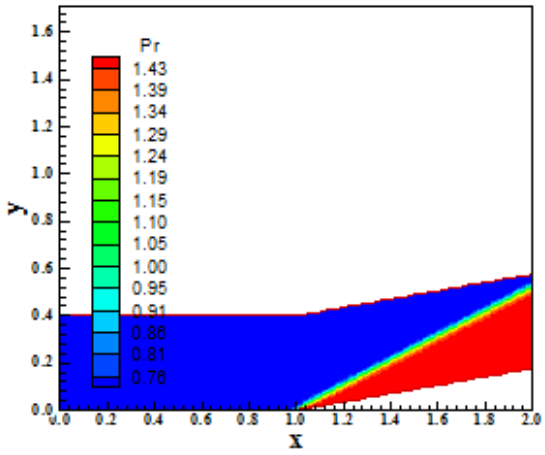

Figure 6. Pressure contours (LW-Min2).

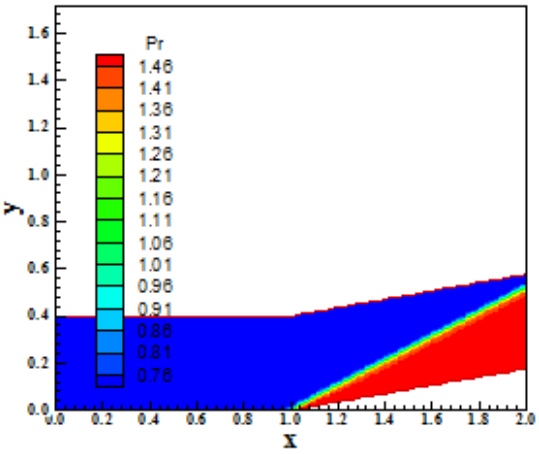

Figure 7. Pressure contours (LW-Min3).

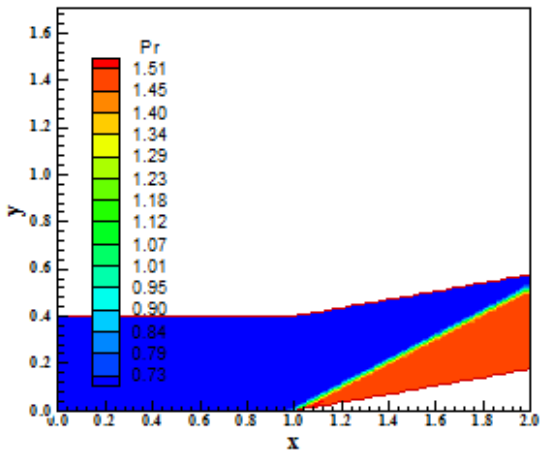

Figure 8. Pressure contours (LW-SB).

One way to quantitatively verify if the solutions generated by each scheme are satisfactory consists in determining the shock angle of the oblique shock wave, β, measured in relation to the initial direction of the flow field. [19] (pages 352 and 353) presents a diagram with values of the shock angle, β, to oblique shock waves. The value of this angle is determined as function of the freestream Mach number and of the deflection angle of the flow after the shock wave, ϕ.

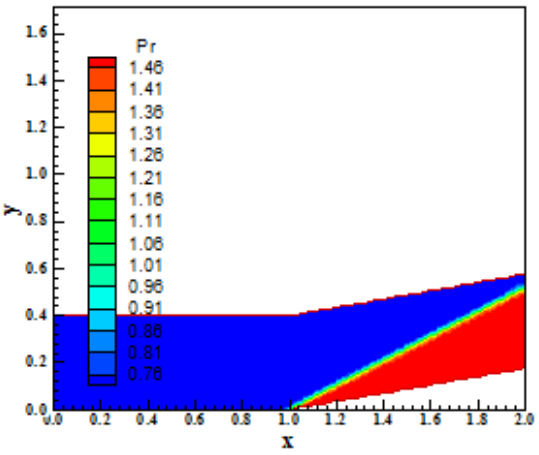

Figure 9. Pressure contours (LW-VL).

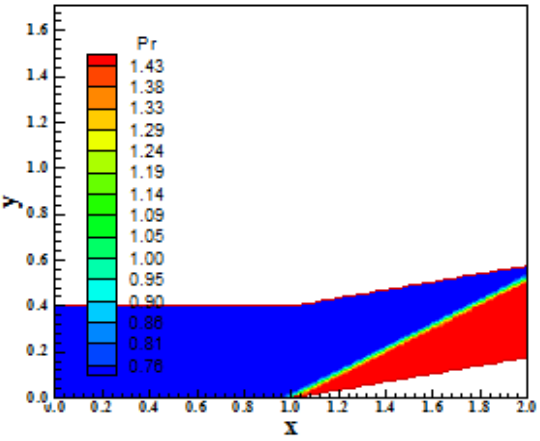

Figure 10. Pressure contours (BB).

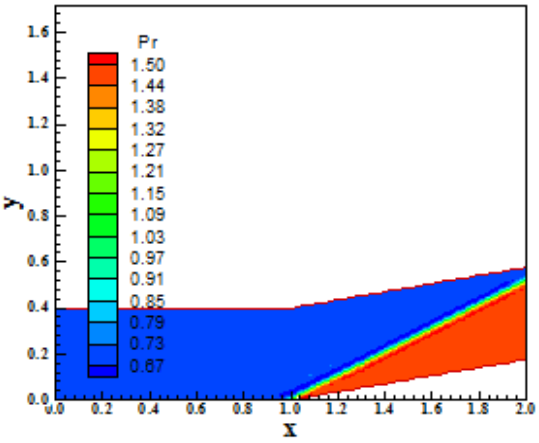

Figure 11. Pressure contours (BW-L).

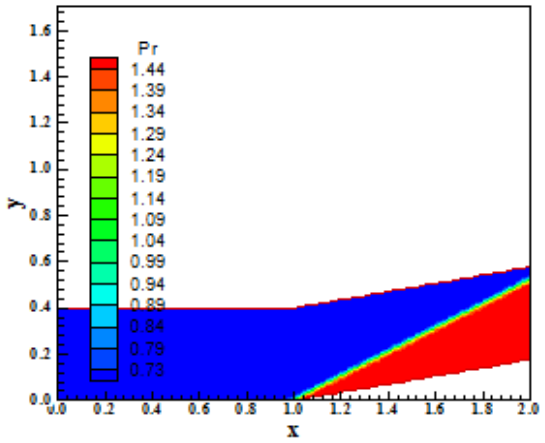

Figure 12. Pressure contours (BW-NL).

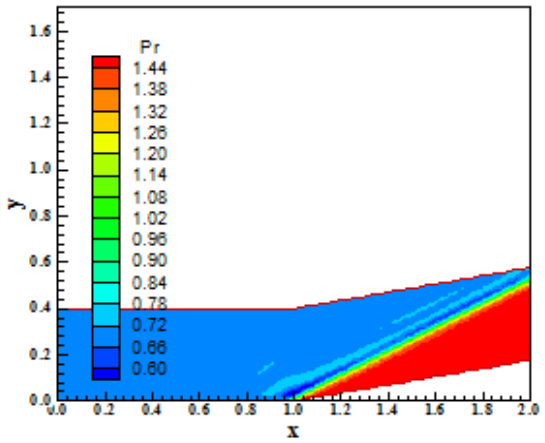

Figure 13. Pressure contours (M).

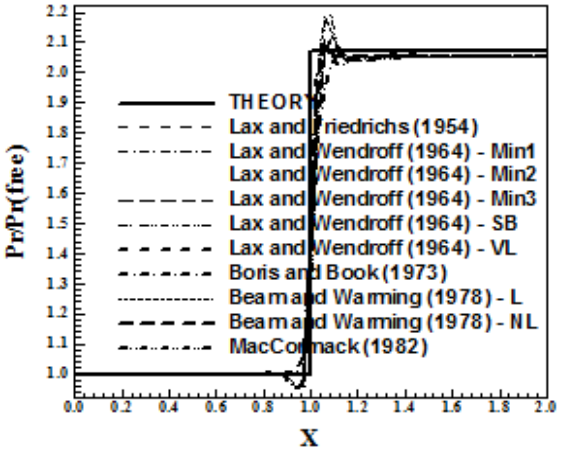

Figure 14. Wall pressure distributions.

To the compression corner problem, $\phi = 10^{o}$ (ramp inclination angle) and the freestream Mach number is 3.0, resulting from this diagram a value to β equals to 27.5º. Using a transfer in Figures 4 to 13, it is possible to obtain the values of β to each scheme, as well the respective percentage errors, shown in Tab. 3.

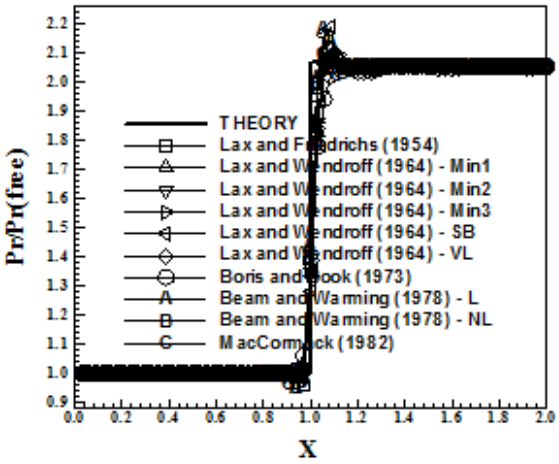

Figure 15. Wall pressure distributions.

Table 3. Shock angle of the oblique shock wave and percentage errors.

Algorithm	β (°)	Error (%)
LF	28.0	1.82
LW-Min1	27.4	0.36
LW-Min2	27.2	1.09
LW-Min3	26.3	4.36
LW-SB	26.2	4.73
LW-VL	27.1	1.45
BB	27.2	1.09
BW-L	27.3	0.73
BW-NL	27.6	0.36
M	26.4	4.00

As can be seen, the best value of the shock angle is obtained with the LW-Min1 and BW-NL schemes. Both schemes presented errors inferior to 0,50%. As the BW-NL scheme had the best behavior in the wall pressure distribution, it is chosen as the best scheme in quality and quantity terms.

11 Conclusions

This work aims to describe the numerical implementation of the [1], [2] TVD, [3], [4] and [5] algorithms, on a finite volume and structured spatial discretization contexts, to solve the Euler equations in two-dimensions. The [2] algorithm was implemented according to the TVD formulation of [6]. The [4] scheme was implemented only in its explicit version. Hence, it is possible to distinguish four categories of algorithms studied in this work: symmetrical ([1]; [4]), FCT ([3]), Predictor / Corrector ([5]) and TVD ([2]). They are applied to the solution of the steady state problem of the

moderate supersonic flow along a compression corner. A spatially variable time step procedure is employed to accelerate the convergence of the numerical methods to the steady state condition. This procedure has a meaningful gain in terms of convergence ratio, as reported by [9-10]. The results have demonstrated that the [4] scheme, using the nonlinear dissipation operator, provides the best results in terms of quality (good capture of shock wave thickness and wall pressure profile) and quantity (good prediction of the oblique shock wave angle). An error of 0.50% is considered excellent.

12 Acknowledgments

The present author acknowledges the CNPq by the financial support conceded under the form of a DTI (Industrial Technological Development) scholarship no. 384681/2011-5. He also acknowledges the infra-structure of the ITA that allowed the realization of this work.

References:

[1] P. D. Lax, Weak Solutions of Non Linear Hyperbolic Equations and Their Numerical Computation, *Comm. Pure and Applied Mathematics*, Vol. 7, 1954, pp. 159-193.

[2] P. D. Lax, and B. Wendroff, Difference Schemes for Hyperbolic Equations with High Order of Accuracy, *Communications on Pure and Applied Mathematics*, Vol. XVII, 1964, pp. 381-398.

[3] J. P. Boris, and D. L. Book, Flux-Corrected Transport. I. SHASTA, A Fluid Transport Algorithm That Works, *Journal of Computational Physics*, Vol. 11, 1973, pp. 38-69.

[4] R. M. Beam, and R. F. Warming, An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, *AIAA Journal*, Vol. 16, No. 4, 1978, pp. 393-402.

[5] R. W. MacCormack, A Numerical Method for Solving the Equations of Compressible Viscous Flow, *AIAA Journal*, Vol. 20, No. 9, 1982, pp.1275-1281.

[6] H. C. Yee, Construction of Explicit and Implicit Symmetric TVD Schemes and Their Applications, *Journal of Computational Physics*, Vol. 68, 1987, pp. 151-179.

[7] P. L. Roe, Generalized Formulation of TVD Lax-Wendroff Schemes, *ICASE Report No. 84-53*, 1984.

[8] S. F. Davis, TVD Finite Difference Schemes and Artificial Viscosity, *ICASE Report No. 84-20*, 1984.

[9] E. S. G. Maciel, Analysis of Convergence Acceleration Techniques Used in Unstructured Algorithms in the Solution of Aeronautical Problems – Part I, *Proceedings of the XVIII International Congress of Mechanical Engineering (XVIII COBEM)*, Ouro Preto, MG, Brazil, 2005. [CD-ROM]

[10] E. S. G., Maciel, Analysis of Convergence Acceleration Techniques Used in Unstructured Algorithms in the Solution of Aerospace Problems – Part II, *Proceedings of the XII Brazilian Congress of Thermal Engineering and Sciences (XII ENCIT)*, Belo Horizonte, MG, Brazil, 2008. [CD-ROM]

[11] P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

[12] E. S. G. Maciel, Comparison Among the First Order Upwind Algorithms of Roe, of Steger and Warming, of Van Leer and of Chakravarthy and Osher in the Solution of the Euler Equations in 2D – Theory, *Proceedings of the 8th Symposium of Computational Mechanics (VIII SIMMEC)*, Belo Horizonte, MG, Brazil, 2008. [CD-ROM]

[13] P. L. Roe, In Proceedings of the AMS-SIAM Summer Seminar on Large-Scale Computation in Fluid Mechanics, Edited by B. E. Engquist *et al*, *Lectures in Applied Mathematics*, Vol. 22, 1983, p. 163.

[14] B. Van Leer, Towards the Ultimate Conservative Difference Scheme. II. Monotonicity and Conservation Combined in a Second-Order Scheme, *Journal of Computational Physics*, Vol. 14, 1974, pp. 361-370.

[15] R. W. MacCormack, The Effect of Viscosity in Hypervelocity Impact Cratering, *AIAA Paper 69-354*, 1969.

[16] T. H. Pulliam, Artificial Dissipation Models for the Euler Equations, *AIAA Journal*, Vol. 24, No. 12, 1986, pp. 1931-1940.

[17] A. Jameson, D. J. Mavriplis, Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh, *AIAA Journal*, Vol. 24, No. 4, 1986, pp. 611-618.

[18] Maciel, E. S. G., Simulação Numérica de Escoamentos Supersônicos e Hipersônicos Utilizando Técnicas de Dinâmica dos Fluidos Computacional, *Doctoral Thesis*, ITA, São José dos Campos, SP, Brazil, 258p, 2002.

[19] J. D. Anderson Jr., *Fundamentals of Aerodynamics*, McGraw-Hill, Inc., EUA, 563p, 1984.