TVD Algorithms Applied to the Solution of the Euler and Navier-Stokes Equations in Three-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: - In the present work, the Yee, Warming and Harten, the Harten, the Yee and Kutler, and the Hughson and Beran schemes are implemented, on a finite volume context and using a structured spatial discretization, to solve the Euler and the Navier-Stokes equations in three-dimensions. All four schemes are TVD ("Total Variation Diminishing") high resolution flux difference splitting ones, second order accurate. An implicit formulation is employed to solve the Euler equations, whereas the Navier-Stokes equations are solved by an explicit formulation. Turbulence is taken into account considering the algebraic models of Cebeci and Smith and of Baldwin and Lomax. The physical problems of the transonic flow along a convergent-divergent nozzle and the supersonic flow along a compression corner in the inviscid case are studied. In the viscous case, the supersonic flow along a ramp is solved. The results have demonstrated that the most severe results are obtained with the Hughson and Beran TVD high resolution scheme, whereas the Yee, Warming and Harten and the Yee and Kutler schemes present more accurate results.

Key-Words: - Yee, Warming and Harten algorithm, Harten algorithm, Yee and Kutler algorithm, Hughson and Beran algorithm, TVD high resolution schemes, Finite Volumes, Three-dimensions.

1 Introduction

Conventional non-upwind algorithms have been used extensively to solve a wide variety of problems ([1-2]). Conventional algorithms are somewhat unreliable in the sense that for every different problem (and sometimes, every different case in the same class of problems) artificial dissipation terms must be specially tuned and judicially chosen for convergence. Also, complex problems with shocks and steep compression and expansion gradients may defy solution altogether.

Upwind schemes are in general more robust but are also more involved in their derivation and application. Some upwind schemes that have been applied to the Euler equations are: [3-7]. Some comments about these methods are reported below:

[3] presented a work that emphasized that several numerical schemes to the solution of the hyperbolic conservation equations were based on exploring the information obtained in the solution of a sequence of Riemann problems. It was verified that in the existent schemes the major part of these information was degraded and that only certain solution aspects were solved. It was demonstrated that the information could be preserved by the construction of a matrix with a certain "U property". After the construction of this matrix, its eigenvalues could be considered as wave velocities of the Riemann problem and the U_L-U_R projections over the matrix's eigenvectors would be the jumps which occur between intermediate stages.

[4] implemented a high resolution second order explicit method based on Harten's ideas. The method had the following properties: (a) the scheme was developed in conservation form to ensure that the limit was a weak solution; (b) the scheme satisfied a proper entropy inequality to ensure that the limit solution would have only physically relevant discontinuities; and (c) the scheme was designed such that the numerical dissipation produced highly accurate weak solutions. The method was applied to the solution of a quasi-one-dimensional nozzle problem and to the two-dimensional shock reflection problem, yielding good results. An implicit formulation was also investigated to one- and two-dimensional cases.

[5] developed a class of new finite difference schemes, explicit and with second order spatial accuracy to the calculation of weak solutions of the hyperbolic conservation laws. These schemes highly non-linear were obtained by the application of a first order non-oscillatory scheme to an appropriated modified flux function. The so derived second order schemes reached high resolution, while preserved the robustness property of the original non-oscillatory first order scheme.

[6] presented a work which extended the [5] scheme to a generalized coordinate system, in two-dimensions. The method called “TVD scheme” by the authors was tested to the physical problem of a moving shock impinging a cylinder. The numerical results were compared with the [8] scheme, presenting good results.

[7] proposed an explicit, second order accurate in space, TVD scheme to solve the Euler equations in axis-symmetrical form, applied to the studies of the supersonic flow around a sphere and the hypersonic flow around a blunt body. The scheme was based on the modified flux function approximation of [5] and its extension from the two-dimensional space to the axis-symmetrical treatment was developed. Results were compared to the [8] algorithm’s solutions. High resolution aspects, capability of shock capturing and robustness properties of this TVD scheme were investigated.

Traditionally, implicit numerical methods have been praised for their improved stability and condemned for their large arithmetic operation counts ([9]). On the one hand, the slow convergence rate of explicit methods become they so unattractive to the solution of steady state problems due to the large number of iterations required to convergence, in spite of the reduced number of operation counts per time step in comparison with their implicit counterparts. Such problem is resulting from the limited stability region which such methods are subjected (the Courant condition). On the other hand, implicit schemes guarantee a larger stability region, which allows the use of CFL numbers above 1.0, and fast convergence to steady state conditions. Undoubtedly, the most significant efficiency achievement for multidimensional implicit methods was the introduction of the Alternating Direction Implicit (ADI) algorithms by [10-12] and fractional step algorithms by [13]. ADI approximate factorization methods consist in approximating the Left Hand Side (LHS) of the numerical scheme by the product of one-dimensional parcels, each one associated with a different spatial coordinate direction, which retract nearly the original implicit operator. These methods have been largely applied in the CFD community and, despite the fact of the error of the approximate factorization, it allows the use of large time steps, which results in significant gains in terms of convergence rate in relation to explicit methods.

There is a practical necessity in the aeronautical industry and in other fields of the capability of calculating separated turbulent compressible flows. With the available numerical methods, researches seem able to analyze several separated flows, three-dimensional in general, if an appropriated turbulence model is employed. Simple methods as the algebraic turbulence models of [14-15] supply satisfactory results with low computational cost and allow that the main features of the turbulent flow be detected.

[16] performed a comparison between the [8, 17] schemes implemented coupled with the [14-15] models to accomplish turbulent flow simulations in three-dimensions. The Navier-Stokes equations in conservative and integral forms were solved, employing a finite volume formulation and a structured spatial discretization. The [8] scheme is a predictor/corrector method which performs coupled time and space discretizations, whereas the [17] algorithm is a symmetrical scheme and its time discretization is performed by a Runge-Kutta method. Both schemes are second order accurate in space and time and require artificial dissipation to guarantee stability. The steady state problem of the supersonic turbulent flow along a ramp was studied. The results have demonstrated that both turbulence models predicted appropriately the boundary layer separation region formed at the compression corner, reducing, however, its extension in relation to the laminar solution, as expected.

In the present work, the [4-7] schemes are implemented, on a finite volume context and using a structured spatial discretization, to solve the Euler and the laminar/turbulent Navier-Stokes equations in the three-dimensional space. All four schemes are TVD high resolution flux difference splitting ones, based on the concept of Harten’s modified flux function. They are second order accurate in space. An implicit formulation is employed to solve the Euler equations in the inviscid problems. An approximate factorization in Linearized Nonconservative Implicit LNI form is employed by the flux difference splitting schemes. To solve the laminar/turbulent Navier-Stokes equations, an explicit formulation based on a time splitting procedure is employed. All schemes are first order accurate in time in their implicit and explicit versions. Turbulence is taken into account considering two algebraic models, namely: the [14-15] ones. The algorithms are accelerated to the steady state solution using a spatially variable time step, which has demonstrated effective gains in terms of convergence rate ([18-19]). All four schemes are applied to the solution of the physical problems of the transonic flow along a convergent-divergent nozzle and of the supersonic flow along a compression corner in the inviscid case (Euler equations). To the laminar/turbulent viscous case, the supersonic flow along a ramp is solved. The results have demonstrated that the most severe results are obtained with the [7] TVD high resolution scheme, whereas the [4] and the [6] algorithms present more accurate results.

The main contribution of this work to the CFD (Computational Fluid Dynamics) community is the extension of the TVD high resolution algorithms of [4-7] to the three-dimensional space, following a finite volume formulation, and their implementation coupled with two different algebraic turbulence models to simulate viscous turbulent flows, which characterizes an original contribution in the field of high resolution structured numerical algorithms. The implicit implementation in three-dimensions of these algorithms is also a meaningful contribution.

2 Navier-Stokes Equations

As the Euler equations can be obtained from the Navier-Stokes ones by disregarding the viscous vectors, only the formulation to the latter will be presented. The Navier-Stokes equations in integral conservative form, employing a finite volume formulation and using a structured spatial discretization, to three-dimensional simulations, can be written as:

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

where V is the cell volume, which corresponds to an hexahedron in the three-dimensional space; Q is the vector of conserved variables; and $\vec{P} = (E_e - E_v)\vec{i} + (F_e - F_v)\vec{j} + (G_e - G_v)\vec{k}$ is the complete flux vector in Cartesian coordinates, with the subscript "e" related to the inviscid contributions or the Euler contributions and "v" related to the viscous contributions. These components of the complete flux vector, as well the vector of conserved variables, are described below:

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

$$G_e = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{Bmatrix}, \qquad (2)$$

$$E_v = \frac{1}{\mathrm{Re}}\begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ \tau_{xx}u + \tau_{xy}v + \tau_{xz}w - q_x \end{Bmatrix},\quad F_v = \frac{1}{\mathrm{Re}}\begin{Bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ \tau_{yx}u + \tau_{yy}v + \tau_{yz}w - q_y \end{Bmatrix}$$

$$G_v = \frac{1}{\mathrm{Re}}\begin{Bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ \tau_{zx}u + \tau_{zy}v + \tau_{zz}w - q_z \end{Bmatrix}. \qquad (3)$$

In these equations, the components of the viscous stress tensor are defined as:

$$\tau_{xx} = 2(\mu_M + \mu_T)\partial u/\partial x - 2/3(\mu_M + \mu_T)(\partial u/\partial x + \partial v/\partial y + \partial w/\partial z),$$
$$\tau_{xy} = (\mu_M + \mu_T)(\partial u/\partial y + \partial v/\partial x); \qquad (4)$$

$$\tau_{xz} = (\mu_M + \mu_T)(\partial u/\partial z + \partial w/\partial x),$$
$$\tau_{yy} = 2(\mu_M + \mu_T)\partial v/\partial y - 2/3(\mu_M + \mu_T)(\partial u/\partial x + \partial v/\partial y + \partial w/\partial z); \qquad (5)$$

$$\tau_{yz} = (\mu_M + \mu_T)(\partial v/\partial z + \partial w/\partial y),$$
$$\tau_{zz} = 2(\mu_M + \mu_T)\partial w/\partial z - 2/3(\mu_M + \mu_T)(\partial u/\partial x + \partial v/\partial y + \partial w/\partial z). \qquad (6)$$

The components of the conductive heat flux vector are defined as follows:

$$q_x = -\gamma(\mu_M/\mathrm{Pr}d + \mu_T/\mathrm{Pr}d_T)\partial e_i/\partial x,$$
$$q_y = -\gamma(\mu_M/\mathrm{Pr}d + \mu_T/\mathrm{Pr}d_T)\partial e_i/\partial y,$$
$$q_z = -\gamma(\mu_M/\mathrm{Pr}d + \mu_T/\mathrm{Pr}d_T)\partial e_i/\partial z. \qquad (7)$$

The quantities that appear above are described as: ρ is the fluid density, u, v and w are the Cartesian components of the flow velocity vector in the x, y and z directions, respectively; e is the total energy; p is the fluid static pressure; e_i is the fluid internal energy, defined as:

$$e_i = e/\rho - 0.5(u^2 + v^2 + w^2); \qquad (8)$$

the τ's represent the components of the viscous stress tensor; Prd is the laminar Prandtl number, which assumed a value of 0.72 in the present simulations; Prd$_T$ is the turbulent Prandtl number, which assumed a value of 0.9; the q's represent the components of the conductive heat flux; μ_M is the fluid molecular viscosity; μ_T is the flow turbulent viscosity; γ is the ratio of specific heats at constant pressure and volume, respectively, equal to 1.4 to the atmospheric air; and Re is the Reynolds number of the viscous simulation, defined by:

$$Re = \rho u_{REF}\, l_{REF} / \mu_M \ , \qquad (9)$$

where u_{REF} is a characteristic flow velocity and l_{REF} is a configuration characteristic length. The molecular viscosity is estimated by the empiric Sutherland formula:

$$\mu_M = bT^{1/2} / \left(1 + S/T\right), \qquad (10)$$

where T is the absolute temperature (K), b = 1.458x10^{-6} Kg/(m.s.K$^{1/2}$) and S = 110.4 K, to the atmospheric air in the standard atmospheric conditions ([20]).

The Navier-Stokes equations were dimensionless in relation to the stagnation density, ρ_*, the critical speed of sound, a_*, and the stagnation viscosity, μ_*, for the nozzle problem, whereas in relation to the freestream density, ρ_∞, the freestream speed of sound, a_∞, and the freestream molecular viscosity, μ_∞, for the compression corner and ramp problems. To allow the solution of the matrix system of five equations to five unknowns described by Eq. (1), it is employed the state equation of perfect gases presented below:

$$p = (\gamma - 1)\left[e - 0.5\rho(u^2 + v^2 + w^2)\right]. \qquad (11)$$

The total enthalpy is determined by $H = (e + p)/\rho$.

3 Geometrical Characteristics of the Spatial Discretization

Adopting in Equation (1) Q as a constant on a computational cell and using a structured mesh notation to the fluid and flow quantities, it is possible to write:

$$\partial Q_{i,j,k} / \partial t = -1/V_{i,j,k} \int_{S_{i,j,k}} \left[(E_e - E_v)n_x + (F_e - F_v)n_y + (G_e - G_v)n_z\right]_{i,j,k} dS_{i,j,k} \ . \qquad (12)$$

A given computational cell in this notation is composed by the following nodes: (i,j,k), (i+1,j,k), (i+1,j+1,k), (i,j+1,k), (i,j,k+1), (i+1,j,k+1), (i+1,j+1,k+1) and (i,j+1,k+1). Figure 1 shows a representation of the computational cell, which is a hexahedron in three-dimensions.

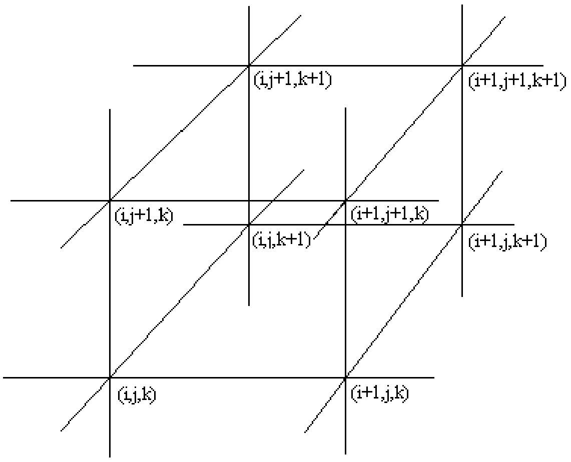

Figure 1 : Computational Cell.

The calculation of the volume of the computational cells is based, in the more general case, in the determination of the volume of a deformed hexahedron in the three-dimensional space. This volume is determined by the summation of the volumes of the six tetrahedrons which composes the given hexahedron. Figure 2 exhibits the division of a hexahedron in its six tetrahedral components, as well the nodes of the vertices which define each tetrahedron.

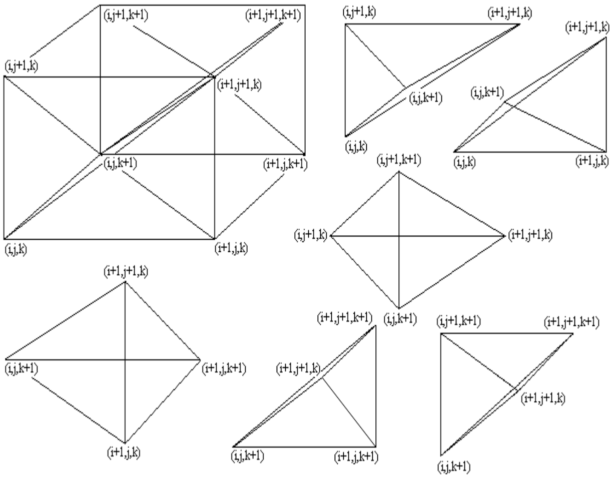

Figure 2 : Hexahedron and its Components.

The volume of a tetrahedron is given by the calculation of the following determinant:

$$V_{PABC} = \frac{1}{6}\begin{vmatrix} x_P & y_P & z_P & 1 \\ x_A & y_A & z_A & 1 \\ x_B & y_B & z_B & 1 \\ x_C & y_C & z_C & 1 \end{vmatrix}, \qquad (13)$$

where x_P, y_P, z_P, x_A, y_A, z_A, x_B, y_B, z_B, x_C, y_C and z_C are Cartesian coordinates of the nodes which define the tetrahedron represented in Fig. 3.

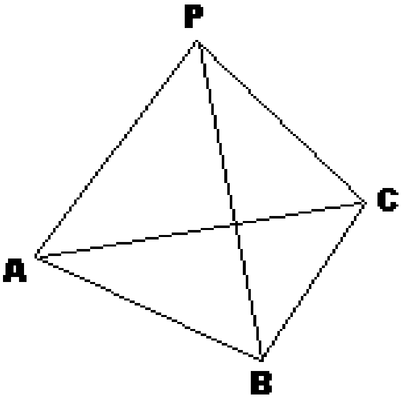

Figure 3 : Reference Tetrahedron.

The flux area of the hexahedron is calculated by the sum of half areas defined by the vector external products $\left|\vec{a}\,x\vec{b}\right|$ and $\left|\vec{c}\,x\vec{d}\right|$, where $\vec{a}$, $\vec{b}$, $\vec{c}$ and $\vec{d}$ are vectors formed by the nodes which define a given flux surface, conform exhibited in Fig. 4. The physical quantity $0.5\left(\left|\vec{a}\,x\vec{b}\right|+\left|\vec{c}\,x\vec{d}\right|\right)$ determines the flux area of each face, which is nothing more than the area of a deformed rectangle.

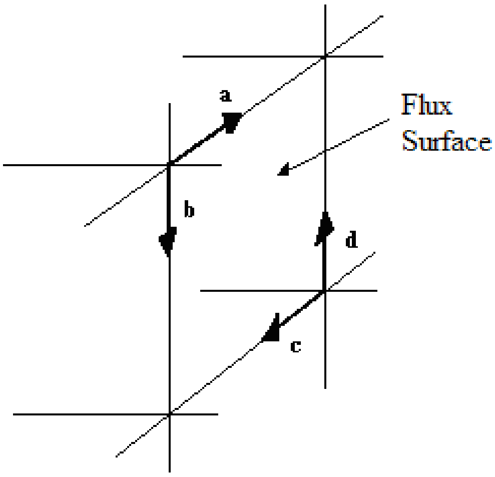

Figure 4 : Flux Area (hexahedron).

The normal unit vector pointing outward at each flux face is calculated taken into account the vector external product $\vec{n}=\vec{a}\,x\vec{b}\Big/\left|\vec{a}\,x\vec{b}\right|$, as shown in Fig. 5. An additional test is necessary to verify if this unit vector is pointing inward or outward of the hexahedron. This test is based on the following vector mixed product $\left[(\vec{a}\,x\vec{b}\,)\Big/\left|\vec{a}\,x\vec{b}\right|\right]\bullet\vec{f}$, where $\vec{f}$ is the vector formed by one of the nodes of the flux face under study and one node of the hexahedron that be contained at the face immediately opposed, and "•" represents the vector inner product . The positive signal indicates that the normal unit vector is pointing inward the hexahedron, what imposes that it should be changed by their opposed vector.

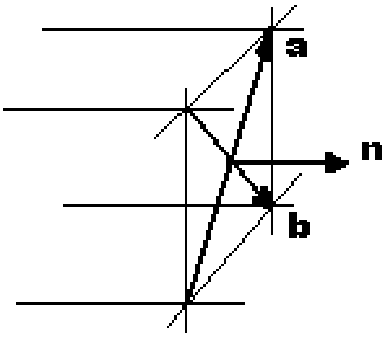

Figure 5 : Normal Unit Vector (Hexahedron).

4 Numerical Scheme of [4]

The [4] algorithm, second order accurate in space, is specified by the determination of the numerical flux vector at *(i+½,j,k)* interface. The implementation of the other numerical flux vectors at the other interfaces is straightforward.

Following a finite volume formalism, which is equivalent to a generalized system, the right and left cell volumes, as well the interface volume, necessary to coordinate change, are defined by:

$$V_R=V_{i+1,j,k}\,,\ V_L=V_{i,j,k}\ \text{and}\ V_{int}=0.5\left(V_R+V_L\right). \tag{14}$$

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

$$h_x=S_{x\_int}\big/V_{int}\,,\ h_y=S_{y\_int}\big/V_{int}\,,$$
$$h_z=S_{z\_int}\big/V_{int}\quad\text{and}\quad h_n=S\big/V_{int}\,, \tag{15}$$

where $S_{x\_int}=n_xS$, $S_{y\_int}=n_yS$, $S_{z\_int}=n_zS$ are the Cartesian components of the flux area and S is the flux area, calculated as described in section 3.

The properties calculated at the flux interface are obtained either by arithmetical average or by [3] average. In this work, the [3] average was used:

$$\rho_{\text{int}} = \sqrt{\rho_L \rho_R}\ ,$$
$$u_{\text{int}} = \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right),$$
$$v_{\text{int}} = \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right); \qquad (16)$$

$$w_{int} = \left(w_L + w_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right),$$
$$H_{\text{int}} = \left(H_L + H_R\sqrt{\rho_R/\rho_L}\right)\Big/\left(1+\sqrt{\rho_R/\rho_L}\right); \ (17)$$

$$a_{int} = \sqrt{(\gamma-1)\left[H_{int} - 0.5\left(u_{int}^2 + v_{int}^2 + w_{int}^2\right)\right]},\ (18)$$

where a_{int} is the speed of sound at the flux interface. The eigenvalues of the Euler equations, in the ξ direction, are given by:

$$U_{cont} = u_{int}h_x + v_{int}h_y + w_{int}h_z,\ \lambda_1 = U_{cont} - a_{int}h_n,$$

$$\lambda_2 = \lambda_3 = \lambda_4 = U_{cont}; \qquad (19)$$

$$\lambda_5 = U_{cont} + a_{int}h_n. \qquad (20)$$

The jumps of the conserved variables, necessary to the construction of the [4] dissipation function, are given by:

$$\Delta\rho = V_{\text{int}}\left(\rho_R - \rho_L\right),\quad \Delta(\rho u) = V_{\text{int}}\left[(\rho u)_R - (\rho u)_L\right],$$
$$\Delta(\rho v) = V_{\text{int}}\left[(\rho v)_R - (\rho v)_L\right]; \qquad (21)$$

$$\Delta(\rho w) = V_{int}\left[(\rho w)_R - (\rho w)_L\right],$$

$$\Delta e = V_{\text{int}}\left(e_R - e_L\right). \qquad (22)$$

The α vectors, which are the jumps of the characteristic variables, at the *(i+½,j,k)* interface are calculated by the following expressions:

$$\left\{\alpha_{i+1/2,j,k}\right\} = \left[R^{-1}\right]_{i+1/2,j,k}\left\{\Delta_{i+1/2,j,k}\overline{Q}\right\}, \qquad (23)$$

with:

$$\left\{\Delta_{i+1/2,j,k}\overline{Q}\right\} = \left\{\Delta\rho \quad \Delta(\rho u) \quad \Delta(\rho v) \quad \Delta(\rho w) \quad \Delta e\right\}^T,$$
defined by Eqs. (21) and (22); (24)

$$\left[R^{-1}\right] = \begin{bmatrix} 0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 + \phi/a_{\text{int}}\right] & -0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} + h_x^{'}/a_{\text{int}}\right] & -0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} + h_y^{'}/a_{\text{int}}\right] \\ 1-\left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 & \left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} & \left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} \\ -\left(h_y^{'}u_{\text{int}} + h_z^{'}v_{\text{int}} + h_x^{'}w_{\text{int}}\right) & h_y^{'} & h_z^{'} \\ -\left(h_z^{'}u_{\text{int}} + h_x^{'}v_{\text{int}} + h_y^{'}w_{\text{int}}\right) & h_z^{'} & h_x^{'} \\ 0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 - \phi/a_{\text{int}}\right] & 0.5\left[-\left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} + h_x^{'}/a_{\text{int}}\right] & 0.5\left[-\left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} + h_y^{'}/a_{\text{int}}\right] \end{bmatrix}$$
$$\begin{bmatrix} -0.5\left[(\gamma-1)/a_{\text{int}}^2 w_{\text{int}} + h_z^{'}/a_{\text{int}}\right] & 0.5(\gamma-1)/a_{\text{int}}^2 \\ (\gamma-1)/a_{\text{int}}^2 w_{\text{int}} & -(\gamma-1)/a_{\text{int}}^2 \\ h_x^{'} & 0 \\ h_y^{'} & 0 \\ 0.5\left[-(\gamma-1)/a_{\text{int}}^2 w_{\text{int}} + h_z^{'}/a_{\text{int}}\right] & 0.5(\gamma-1)/a_{\text{int}}^2 \end{bmatrix}; \quad (25)$$

$$q^2 = u_{int}^2 + v_{int}^2 + w_{int}^2; \qquad (26)$$

$$\phi = u_{\text{int}}h_x^{'} + v_{\text{int}}h_y^{'} + w_{\text{int}}h_z^{'}; \qquad (27)$$

$$h_x^{'} = h_x/h_n\ ,\ h_y^{'} = h_y/h_n \quad \text{and} \quad h_z^{'} = h_z/h_n\ . \ (28)$$

The Yee, Warming and Harten (1982) dissipation function uses the right-eigenvector matrix of the normal to the flux face Jacobian matrix in generalized coordinates:

$$[R] = \begin{bmatrix} 1 & 1 & 0 \\ u_{\text{int}} - h_x^{'}a_{\text{int}} & u_{\text{int}} & h_y^{'} \\ v_{\text{int}} - h_y^{'}a_{\text{int}} & v_{\text{int}} & h_z^{'} \\ w_{\text{int}} - h_z^{'}a_{\text{int}} & w_{\text{int}} & h_x^{'} \\ H_{\text{int}} - h_x^{'}u_{\text{int}}a_{\text{int}} - h_y^{'}v_{\text{int}}a_{\text{int}} - h_z^{'}w_{\text{int}}a_{\text{int}} & 0.5q^2 & h_x^{'}w_{\text{int}} + h_z^{'}v_{\text{int}} + h_y^{'}u_{\text{int}} \end{bmatrix}$$
$$\begin{bmatrix} 0 & 1 \\ h_z^{'} & u_{\text{int}} + h_x^{'}a_{\text{int}} \\ h_x^{'} & v_{\text{int}} + h_y^{'}a_{\text{int}} \\ h_y^{'} & w_{\text{int}} + h_z^{'}a_{\text{int}} \\ h_y^{'}w_{\text{int}} + h_x^{'}v_{\text{int}} + h_z^{'}u_{\text{int}} & H_{\text{int}} + h_x^{'}u_{\text{int}}a_{\text{int}} + h_y^{'}v_{\text{int}}a_{\text{int}} + h_z^{'}w_{\text{int}}a_{\text{int}} \end{bmatrix}. \ (29)$$

Two options to the ψ_l entropy function, responsible to guarantee only relevant physical solutions, are implemented aiming an entropy satisfying algorithm:

$$\nu_l = \Delta t_{i,j,k}\lambda_l = Z_l \quad \text{and} \quad \psi_l = Z_l^2 + 0.25;\ (30)$$

or:

$$\psi_l = \begin{cases} \left|Z_l\right|, & if\ \left|Z_l\right| \geq \delta_f \\ 0.5\left(Z_l^2 + \delta_f^2\right)/\delta_f\ , & if\ \left|Z_l\right| < \delta_f \end{cases},\quad (31)$$

where “*l*” varies from 1 to 5 (three-dimensional space) and δ_f assuming values between 0.1 and 0.5, being 0.2 the value recommended by [4]. In the present studies, Eq. (30) was used to perform the inviscid numerical experiments and Eq. (31) was used to perform the viscous numerical experiments.

The $\tilde{g}$ function at the *(i+½,j,k)* interface is defined by:

$$\tilde{g}^l = 0.5\left(\psi_l - Z_l^2\right)\alpha^l, \tag{32}$$

with α^l being the *l*th component of the alpha vector (Eq. 23).

The *g* numerical flux function, which is a limited function to avoid the formation of new extrema in the solution and is responsible to scheme second order accuracy, is given by:

$$\mathrm{g}^{\mathrm{l}}_{\mathrm{i,j,k}} = \mathrm{signal}_\mathrm{l} \times \mathrm{MAX}\left[0.0; \mathrm{MIN}\left(\left|\tilde{\mathrm{g}}^{\mathrm{l}}_{\mathrm{i+1/2,j,k}}\right|, \tilde{\mathrm{g}}^{\mathrm{l}}_{\mathrm{i-1/2,j,k}} \times \mathrm{signal}_\mathrm{l}\right)\right], \tag{33}$$

where *signal*$_l$ is equal to 1.0 if $\tilde{g}^l_{i+1/2,j,k} \geq 0.0$ and -1.0 otherwise.

The θ term, responsible to the artificial compression, which enhances the resolution of the scheme at discontinuities, is defined as follows:

$$\theta^{\mathrm{l}}_{\mathrm{i,j,k}} = \begin{cases} \left|\alpha^{\mathrm{l}}_{\mathrm{i+1/2,j,k}} - \alpha^{\mathrm{l}}_{\mathrm{i-1/2,j,k}}\right| \Big/ \left(\left|\alpha^{\mathrm{l}}_{\mathrm{i+1/2,j,k}}\right| + \left|\alpha^{\mathrm{l}}_{\mathrm{i-1/2,j,k}}\right|\right), & \text{if } \left|\alpha^{\mathrm{l}}_{\mathrm{i+1/2,j,k}}\right| + \left|\alpha^{\mathrm{l}}_{\mathrm{i-1/2,j,k}}\right| \neq 0.0 \\ 0.0, & \text{if } \left|\alpha^{\mathrm{l}}_{\mathrm{i+1/2,j,k}}\right| + \left|\alpha^{\mathrm{l}}_{\mathrm{i-1/2,j,k}}\right| = 0.0 \end{cases}. \tag{34}$$

The β parameter at the *(i+½,j,k)* interface, which introduces the artificial compression term in the algorithm, is given by the following expression:

$$\beta_l = 1.0 + \omega_l MAX(\theta^l_{i,j,k}, \theta^l_{i+1,j,k}), \tag{35}$$

in which ω_l assumes the following values: $\omega_1 = 0.25$ (non-linear field), $\omega_2 = \omega_3 = \omega_4 = 1.0$ (linear field) and $\omega_5 = 0.25$ (non-linear field). The numerical characteristic speed, φ_l, at the *(i+½,j,k)* interface, which is responsible to transport the numerical information associated to the g numerical flux function, is defined by:

$$\varphi_l = \begin{cases} \left(g^l_{i+1,j,k} - g^l_{i,j,k}\right)\big/\alpha^l, & if\ \alpha^l \neq 0.0 \\ 0.0, & if\ \alpha^l = 0.0 \end{cases}. \tag{36}$$

The entropy function is redefined considering φ_l and β_l: $Z_l = \nu_l + \beta_l\varphi_l$, and ψ_l is recalculated according to Eq. (30) or to Eq. (31). Finally, the [4] dissipation function, to second order of spatial accuracy, is constructed by the following matrix-vector product:

$$\left\{D_{[4]}\right\}_{i+1/2,j,k} = \left[R\right]_{i+1/2,j,k}\left\{\left(\beta\left(g_{i,j,k} + g_{i+1,j,k}\right) - \psi\alpha\right)\big/\Delta t_{i,j,k}\right\}_{i+1/2,j,k}. \tag{37}$$

The convective numerical flux vector to the *(i+½,j,k)* interface is described by:

$$F^{(l)}_{i+1/2,j,k} = \left(E^{(l)}_{\mathrm{int}}h_x + F^{(l)}_{\mathrm{int}}h_y + G^{(l)}_{\mathrm{int}}h_z\right)V_{\mathrm{int}} + 0.5D^{(l)}_{[4]}, \tag{38}$$

with:

$$E^{(l)}_{int} = 0.5\left(E^{(l)}_R + E^{(l)}_L\right),\ F^{(l)}_{int} = 0.5\left(F^{(l)}_R + F^{(l)}_L\right),$$

$$G^{(l)}_{int} = 0.5\left(G^{(l)}_R + G^{(l)}_L\right). \tag{39}$$

The right-hand-side of the [4] scheme, necessaries to the resolution of the implicit version of this algorithm, is determined by:

$$RHS([4])^n_{i,j,k} = -\Delta t_{i,j,k}/V_{i,j,k}\left(F^n_{i+1/2,j,k} - F^n_{i-1/2,j,k} + F^n_{i,j+1/2,k} + F^n_{i,j-1/2,k} - F^n_{i,j,k+1/2} + F^n_{i,j,k-1/2}\right). \tag{40}$$

To the viscous simulations, it was implemented the explicit version. In this case, the time integration is replaced by a time splitting method, first order accurate, which divides the temporal integration in three steps, each one associated with a different spatial direction. Considering the initial step associated with the ξ direction, one has:

$$\Delta Q^*_{i,j,k} = -\Delta t_{i,j,k}/V_{i,j,k}\left(F^n_{i+1/2,j,k} - F^n_{i-1/2,j,k}\right),$$
$$Q^*_{i,j,k} = Q^n_{i,j,k} + \Delta Q^*_{i,j,k}; \tag{41}$$

in the intermediate step, considering the η direction, one has:

$$\Delta Q^{**}_{i,j,k} = -\Delta t_{i,j,k}/V_{i,j,k}\left(F^*_{i,j+1/2,k} - F^*_{i,j-1/2,k}\right),$$
$$Q^{**}_{i,j,k} = Q^*_{i,j,k} + \Delta Q^{**}_{i,j,k}; \tag{42}$$

and, in the final step, considering the ζ direction, one has:

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

The viscous vectors at the flux interface are obtained by arithmetical average between the primitive variables at the left and at the right states of the flux interface, as also arithmetical average of the primitive variable gradients, also considering the left and the right states of the flux interface. The gradients of the primitive variables present in the viscous flux vectors are calculated employing the Green Theorem which considers that the gradient of a primitive variable is constant in the volume and that the volume integral which defines this gradient is replaced by a surface integral ([21]); For instance, to $\partial u/\partial x$:

$$\frac{\partial u}{\partial x} = \frac{1}{V}\int_V \frac{\partial u}{\partial x} dV = \frac{1}{V}\int_V u\left(\vec{n} \bullet d\vec{S}\right) = \frac{1}{V}\int_V u dS_x \cong \frac{1}{V_{i,j,k}}\left[0.5\left(u_{i,j,k} + u_{i,j-1,k}\right)S_{x_{i,j-1/2,k}} + \right.$$
$$0.5\left(u_{i,j,k} + u_{i+1,j,k}\right)S_{x_{i+1/2,j,k}} + 0.5\left(u_{i,j,k} + u_{i,j+1,k}\right)S_{x_{i,j+1/2,k}} + 0.5\left(u_{i,j,k} + u_{i-1,j,k}\right)S_{x_{i-1/2,j,k}} +$$
$$\left. 0.5\left(u_{i,j,k} + u_{i,j,k-1}\right)S_{x_{i,j,k-1/2}} + 0.5\left(u_{i,j,k} + u_{i,j,k+1}\right)S_{x_{i,j,k+1/2}} \right]. \qquad (44)$$

5 Numerical Scheme of [5]

The [5] algorithm, second order accurate in space, follows the Eqs. (14) to (29). The next step is the definition of the entropy condition, which is defined by Eq. (30), ν_l, and Eq. (31).

The $\tilde{g}$ function at the *(i+½,j,k)* interface is defined according to Eq. (32) and the g numerical flux function is given by Eq. (33). The numerical characteristic speed φ_l at the *(i+½,j,k)* interface is defined according to Eq. (36).

The entropy function is redefined considering φ_l: $Z_l = \nu_l + \varphi_l$, and ψ_l is recalculated according to Eq. (31). Finally, the [5] dissipation function, to second order spatial accuracy, is constructed by the following matrix-vector product:

$$\left\{D_{[5]}\right\}_{i+1/2,j,k} = \left[R\right]_{i+1/2,j,k}\left\{\left(g_{i,j,k} + g_{i+1,j,k} - \psi\alpha\right)/\Delta t_{i,j,k}\right\}_{i+1/2,j,k}. \qquad (45)$$

The convective numerical flux vector of the [5] scheme is defined by:

$$F_{i+1/2,j,k}^{(l)} = \left(E_{\text{int}}^{(l)} h_x + F_{\text{int}}^{(l)} h_y + G_{\text{int}}^{(l)} h_z\right)V_{\text{int}} + 0.5 D_{[5]}^{(l)}, \qquad (46)$$

with $E_{int}^{(l)}$, $F_{int}^{(l)}$ and $G_{int}^{(l)}$ determined by Eq. (39). The right-hand-side of the [5] scheme, necessaries to the resolution of the implicit version of this algorithm, is defined by:

$$RHS([5])_{i,j,k}^{n} = -\Delta t_{i,j,k}/V_{i,j,k}\left(F_{i+1/2,j,k}^{n} - F_{i-1/2,j,k}^{n} + F_{i,j+1/2,k}^{n} - \right.$$
$$\left. F_{i,j-1/2,k}^{n} + F_{i,j,k+1/2}^{n} - F_{i,j,k-1/2}^{n}\right). \qquad (48)$$

The explicit version of this algorithm to perform the viscous simulations is described by Eqs. (41) to (43). The implementation of the viscous terms follows the same procedure as described in section **4**.

6 Numerical Scheme of [6]

The [6] algorithm, second order accurate in space, follows Eqs. (14) to (29). The next step consists in determining the θ function. This function is defined in terms of the differences of the gradients of the characteristic variables to take into account discontinuities effects and is responsible to artificial compression:

$$\theta_{i,j,k}^{l} = \begin{cases} \dfrac{\left|\alpha_{i+1/2,j,k}^{l} - \alpha_{i-1/2,j,k}^{l}\right|}{\alpha_{i+1/2,j,k}^{l} + \alpha_{i-1/2,j,k}^{l}}, & if\ \left(\alpha_{i+1/2,j,k}^{l} + \alpha_{i-1/2,j,k}^{l}\right) \neq 0.0 \\ 0.0, & if\ \left(\alpha_{i+1/2,j,k}^{l} + \alpha_{i-1/2,j,k}^{l}\right) = 0.0 \end{cases}. \qquad (48)$$

The κ function at the *(i+½,j,k)* interface is defined as follows:

$$\kappa_l = 1/8\left(1 + \omega_l MAX\left(\theta_{i,j,k}^{l}, \theta_{i+1,j,k}^{l}\right)\right), \qquad (49)$$

The *g* numerical flux function is determined by:

$$g_{i,j,k}^{l} = signal_l \times MAX\left[0.0; MIN\left(\left|\alpha_{i+1/2,j,k}^{l}\right|, \alpha_{i-1/2,j,k}^{l} \times signal_l\right)\right], \qquad (50)$$

where $signal_l$ assumes value 1.0 if $\alpha_{i+1/2,j,k}^{l} \geq 0.0$ and -1.0 otherwise. The numerical characteristic speed φ_l at the *(i+½,j,k)* interface is calculated by the following expression:

$$\varphi_l = \begin{cases} \left(g_{i+1,j,k}^{l} - g_{i,j,k}^{l}\right)/\alpha^{l}, & if\ \alpha^{l} \neq 0.0 \\ 0.0, & if\ \alpha^{l} = 0.0 \end{cases}. \qquad (51)$$

The ψ_l entropy function at the *(i+½,j,k)* interface is defined by:

$$\psi_l = \left(\nu_l + \varphi_l\right)^2 + 0.25; \tag{52}$$

with ν_l defined according to Eq. (30). Finally, the [6] dissipation function, to second order spatial accuracy, is constructed by the following matrix-vector product:

$$\left\{D_{[6]}\right\}_{i+1/2,j,k} = \left[R\right]_{i+1/2,j,k} \left\{\left(\kappa\left(g_{i,j,k} + g_{i+1,j,k}\right) - \psi\alpha\right) / \Delta t_{i,j,k}\right\}_{i+1/2,j,k}. \tag{53}$$

The convective numerical flux vector of the [6] scheme is defined by:

$$F^{(l)}_{i+1/2,j,k} = \left(E^{(l)}_{\text{int}} h_x + F^{(l)}_{\text{int}} h_y + G^{(l)}_{\text{int}} h_z\right) V_{\text{int}} + 0.5 D^{(l)}_{[6]}, \tag{54}$$

with $E^{(l)}_{int}$, $F^{(l)}_{int}$ and $G^{(l)}_{int}$ determined by Eq. (39). The right-hand-side of the [6] scheme is defined by:

$$RHS([6])^n_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left(F^n_{i+1/2,j,k} - F^n_{i-1/2,j,k} + F^n_{i,j+1/2,k} - F^n_{i,j-1/2,k} + F^n_{i,j,k+1/2} - F^n_{i,j,k-1/2}\right). \tag{55}$$

The explicit version of this algorithm to perform the viscous simulations is described by Eqs. (41) to (43). The implementation of the viscous terms follows the same procedure as described in section **4**.

7 Numerical Scheme of [7]

The [7] algorithm, second order accurate in space, follows Eqs. (14) to (29). The next step consists in determining the g numerical flux function. To non-linear fields (l = 1 and 5), it is possible to write:

$$g^l_{i,j,k} = \begin{cases} \dfrac{\alpha^l_{i+1/2,j,k}\alpha^l_{i-1/2,j,k} + \left|\alpha^l_{i+1/2,j,k}\alpha^l_{i-1/2,j,k}\right|}{\alpha^l_{i+1/2,j,k} + \alpha^l_{i-1/2,j,k}}, & if\left(\alpha^l_{i+1/2,j,k} + \alpha^l_{i-1/2,j,k}\right) \neq 0.0 \\ 0.0, & if\left(\alpha^l_{i+1/2,j,k} + \alpha^l_{i-1/2,j,k}\right) = 0.0 \end{cases}. \tag{56}$$

To linear fields (l = 2 to 4), it is possible to write:

$$g^l_{i,j,k} = signal_l \times MAX\left[0.0; MIN\left(\left|\alpha^l_{i-1/2,j,k}\right|, \alpha^l_{i+1/2,j,k} \times signal_l\right)\right], \tag{57}$$

where $signal_l$ is equals to 1.0 if $\alpha^l_{i-1/2,j,k}$ ≥ 0.0 and -1.0 otherwise. After that, Equations (30), ν_l term, and (31) are used and the σ_l term at the *(i+½,j,k)* interface is defined by:

$$\sigma_l = 0.5\left(\psi_l - Z_l^2\right). \tag{58}$$

The φ_l numerical characteristic speed at the *(i+½,j,k)* interface is defined by:

$$\varphi_l = \begin{cases} \sigma_l\left(g^l_{i+1,j,k} - g^l_{i,j,k}\right) / \alpha^l, & if\ \alpha^l \neq 0.0 \\ 0.0, & if\ \alpha^l = 0.0 \end{cases}. \tag{59}$$

The entropy function is redefined considering the φ_l term: $Z_l = \nu_l + \varphi_l$ and ψ_l is recalculated according to Eq. (31). The [7] dissipation function, to second order accuracy in space, is constructed by the following matrix-vector product:

$$\left\{D_{[7]}\right\}_{i+1/2,j,k} = \left[R\right]_{i+1/2,j,k} \left\{\left(\sigma\left(g_{i,j,k} + g_{i+1,j,k}\right) - \psi\alpha\right) / \Delta t_{i,j,k}\right\}_{i+1/2,j,k}. \tag{60}$$

The convective numerical flux vector of the [7] scheme is defined by:

$$F^{(l)}_{i+1/2,j,k} = \left(E^{(l)}_{\text{int}} h_x + F^{(l)}_{\text{int}} h_y + G^{(l)}_{\text{int}} h_z\right) V_{\text{int}} + 0.5 D^{(l)}_{[7]}, \tag{61}$$

with $E^{(l)}_{int}$, $F^{(l)}_{int}$ and $G^{(l)}_{int}$ determined by Eq. (39). The right-hand-side of the [7] scheme is defined by:

$$RHS([7])^n_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left(F^n_{i+1/2,j,k} - F^n_{i-1/2,j,k} + F^n_{i,j+1/2,k} - F^n_{i,j-1/2,k} + F^n_{i,j,k+1/2} - F^n_{i,j,k-1/2}\right). \tag{62}$$

The explicit version of this algorithm to perform the viscous simulations is described by Eqs. (41) to (43). The implementation of the viscous terms follows the same procedure as described in section **4**.

8 Implicit Formulation

All implicit schemes implemented in this work used backward Euler in time and LNI approximate factorization to solve a three-diagonal system in each direction.

To the flux difference splitting algorithms tested in this work, a Linearized Nonconservative Implicit (LNI) form is applied that, although the resulting schemes loss the conservative property, preserve their unconditionally TVD property. Moreover, the LNI form is mainly useful to steady state calculations, since the schemes are only conservative after the solution reaches steady state. This LNI form to the solution of the studied TVD implicit schemes was proposed by [22]. The LNI form presents three stages as described below:

$$\left[I - \Delta t_{i,j,k} J^{-}_{i+1/2,j,k} \Delta_{i+1/2,j,k} + \Delta t_{i,j,k} J^{+}_{i-1/2,j,k} \Delta_{i-1/2,j,k}\right] \Delta Q^{*}_{i,j,k} = RHS_{i,j,k}\,; \tag{63}$$

$$\left[I - \Delta t_{i,j,k} K^{-}_{i,j+1/2,k} \Delta_{i,j+1/2,k} + \Delta t_{i,j,k} K^{+}_{i,j-1/2,k} \Delta_{i,j-1/2,k}\right] \Delta Q^{**}_{i,j,k} = \Delta Q^{*}_{i,j,k}\,; \tag{64}$$

$$\left[I - \Delta t_{i,j,k} L^{-}_{i,j,k+1/2} \Delta_{i,j,k+1/2} + \Delta t_{i,j,k} L^{+}_{i,j,k-1/2} \Delta_{i,j,k-1/2}\right] \Delta Q^{n+1}_{i,j,k} = \Delta Q^{**}_{i,j,k}\,, \tag{65}$$

where: $RHS_{i,j,k}$ is defined by Eq. (40) or (47) or (55) or (62) depending if the [4] or the [5] or the [6] or [7] scheme is being solved, respectively; the difference operators are defined as:

$$\begin{aligned} &\Delta_{i+1/2,j,k}(\cdot) = (\cdot)_{i+1,j,k} - (\cdot)_{i,j,k} \\ &\Delta_{i-1/2,j,k}(\cdot) = (\cdot)_{i,j,k} - (\cdot)_{i-1,j,k} \\ &\Delta_{i,j+1/2,k}(\cdot) = (\cdot)_{i,j+1,k} - (\cdot)_{i,j,k}\,; \end{aligned} \tag{66}$$

$$\begin{aligned} &\Delta_{i,j-1/2,k}(\cdot) = (\cdot)_{i,j,k} - (\cdot)_{i,j-1,k}\,, \\ &\Delta_{i,j,k+1/2}(\cdot) = (\cdot)_{i,j,k+1} - (\cdot)_{i,j,k}\,, \\ &\Delta_{i,j,k-1/2}(\cdot) = (\cdot)_{i,j,k} - (\cdot)_{i,j,k-1}\,; \end{aligned} \tag{67}$$

and the update of the conserved variable vector is proceeded as follows:

$$Q^{n+1}_{i,j,k} = Q^{n}_{i,j,k} + \Delta Q^{n+1}_{i,j,k}\,. \tag{68}$$

This system of 5x5 block three-diagonal linear equations is solved using LU decomposition and the Thomas algorithm applied to systems of block matrices.

The splitting matrices J^+, J^-, K^+, K^-, L^+ and L^- are defined as:

$$J^{+} = R_{\xi}\, diag\left(D^{+}_{\xi}\right) R^{-1}_{\xi}\,,$$

$$J^{-} = R_{\xi}\, diag\left(D^{-}_{\xi}\right) R^{-1}_{\xi}\,,\ K^{+} = R_{\eta}\, diag\left(D^{+}_{\eta}\right) R^{-1}_{\eta}\,; \tag{69}$$

$$\begin{aligned} &K^{-} = R_{\eta}\, diag\left(D^{-}_{\eta}\right) R^{-1}_{\eta}\,,\ L^{+} = R_{\zeta}\, diag\left(D^{+}_{\zeta}\right) R^{-1}_{\zeta}\,, \\ &L^{-} = R_{\zeta}\, diag\left(D^{-}_{\zeta}\right) R^{-1}_{\zeta}\,, \end{aligned} \tag{70}$$

where R_ξ, R_η, R_ζ, R^{-1}_{ξ}, R^{-1}_{η} and R^{-1}_{ζ} are defined by Eqs. (25) and (29) applied to each coordinate direction; $diag(\cdot)$ represents a diagonal matrix, as for instance:

$$D^{+}_{\xi} = \begin{bmatrix} D^{\xi,+}_{1} & & & & \\ & D^{\xi,+}_{2} & & & \\ & & D^{\xi,+}_{3} & & \\ & & & D^{\xi,+}_{4} & \\ & & & & D^{\xi,+}_{5} \end{bmatrix}$$

$$D^{-}_{\xi} = \begin{bmatrix} D^{\xi,-}_{1} & & & & \\ & D^{\xi,-}_{2} & & & \\ & & D^{\xi,-}_{3} & & \\ & & & D^{\xi,-}_{4} & \\ & & & & D^{\xi,-}_{5} \end{bmatrix} \tag{71}$$

and the terms D are defined as:

$$\begin{aligned} D^{\pm}_{\xi} &= 0.5\left[Q\left(\lambda^{l}_{\xi} + \gamma^{l}_{\xi}\right) \pm \left(\lambda^{l}_{\xi} + \gamma^{l}_{\xi}\right)\right], \\ D^{\pm}_{\eta} &= 0.5\left[Q\left(\lambda^{l}_{\eta} + \gamma^{l}_{\eta}\right) \pm \left(\lambda^{l}_{\eta} + \gamma^{l}_{\eta}\right)\right]; \\ D^{\pm}_{\zeta} &= 0.5\left[Q\left(\lambda^{l}_{\zeta} + \gamma^{l}_{\zeta}\right) \pm \left(\lambda^{l}_{\zeta} + \gamma^{l}_{\zeta}\right)\right], \end{aligned} \tag{72}$$

with:

$$Q(x_l) = \begin{cases} |x_l|, & if\ |x_l| \geq \delta_f \\ 0.5\left(x^2_l + \delta^2_f\right)/\delta_f\,, & if\ |x_l| < \delta_f \end{cases},\quad \delta_f$$

defined according to Eq. (31); (73)

λ^{l}_{ξ}, λ^{l}_{η} and λ^{l}_{ζ} are the eigenvalues of the Euler equations defined by Eqs. (19) and (20) in each coordinate direction;

$$\left(\gamma^{l}_{\xi}\right)_{i+1/2,j,k} = \begin{cases} \left[\left(g^{'}_{\xi}\right)^{l}_{i+1,j,k} - \left(g^{'}_{\xi}\right)^{l}_{i,j,k}\right] \Big/ \left(\alpha^{l}_{\xi}\right)_{i+1/2,j,k}, & if\ \left(\alpha^{l}_{\xi}\right)_{i+1/2,j,k} \neq 0.0 \\ 0.0, & if\ \left(\alpha^{l}_{\xi}\right)_{i+1/2,j,k} = 0.0 \end{cases}; \tag{74}$$

$$\left(\gamma^{l}_{\eta}\right)_{i,j+1/2,k} = \begin{cases} \left[\left(g^{'}_{\eta}\right)^{l}_{i,j+1,k} - \left(g^{'}_{\eta}\right)^{l}_{i,j,k}\right] \Big/ \left(\alpha^{l}_{\eta}\right)_{i,j+1/2,k}, & if\ \left(\alpha^{l}_{\eta}\right)_{i,j+1/2,k} \neq 0.0 \\ 0.0, & if\ \left(\alpha^{l}_{\eta}\right)_{i,j+1/2,k} = 0.0 \end{cases}; \tag{75}$$

$$\left(\gamma^{l}_{\zeta}\right)_{i,j,k+1/2} = \begin{cases} \left[\left(g^{'}_{\zeta}\right)^{l}_{i,j,k+1} - \left(g^{'}_{\zeta}\right)^{l}_{i,j,k}\right] \Big/ \left(\alpha^{l}_{\zeta}\right)_{i,j,k+1/2}, & if\ \left(\alpha^{l}_{\zeta}\right)_{i,j,k+1/2} \neq 0.0 \\ 0.0, & if\ \left(\alpha^{l}_{\zeta}\right)_{i,j,k+1/2} = 0.0 \end{cases}; \tag{76}$$

$$\left(g'_\xi\right)^l_{i,j,k} = signal^l_\xi MAX\left[0.0, MIN\left(\sigma^l_{i+1/2,j,k}\left|\left(\alpha^l_\xi\right)_{i+1/2,j,k}\right|,\right.\right.$$

$$\left.\left. signal^l_\xi \sigma^l_{i-1/2,j,k}\left(\alpha^l_\xi\right)_{i-1/2,j,k}\right)\right]; \qquad (77)$$

$$\left(g'_\eta\right)^l_{i,j,k} = signal^l_\eta MAX\left[0.0, MIN\left(\sigma^l_{i,j+1/2,k}\left|\left(\alpha^l_\eta\right)_{i,j+1/2,k}\right|,\right.\right.$$

$$\left.\left. signal^l_\eta \sigma^l_{i,j-1/2,k}\left(\alpha^l_\eta\right)_{i,j-1/2,k}\right)\right]; \qquad (78)$$

$$\left(g'_\zeta\right)^l_{i,j,k} = signal^l_\zeta MAX\left[0.0, MIN\left(\sigma^l_{i,j,k+1/2}\left|\left(\alpha^l_\zeta\right)_{i,j,k+1/2}\right|,\right.\right.$$

$$\left.\left. signal^l_\zeta \sigma^l_{i,j,k-1/2}\left(\alpha^l_\zeta\right)_{i,j,k-1/2}\right)\right]; \qquad (79)$$

$\sigma^l = 0.5Q^l\left(\lambda^l\right)$ to steady state simulations. (80)

Finally, $signal^l_\xi$ = 1.0 if $\left(\alpha^l_\xi\right)_{i+1/2,j,k} \geq 0.0$ and -1.0 otherwise; $signal^l_\eta$ = 1.0 if $\left(\alpha^l_\eta\right)_{i,j+1/2,k} \geq 0.0$ and -1.0 otherwise; and $signal^l_\zeta$ = 1.0 if $\left(\alpha^l_\zeta\right)_{i,j,k+1/2} \geq 0.0$ and -1.0 otherwise.

This implicit formulation to the LHS of the [4], of the [5], of the [6], and of the [7] schemes is first order accurate in time and second in space due to the presence of the numerical characteristic speed γ associated to the numerical flux function g'. In this case, the solution accuracy in space is definitively of second order because both LHS and RHS are also of second order.

It is important to emphasize that as the right-hand-side of the implicit flux difference splitting schemes tested in this work presents steady state solutions which depends of the time step, the use of large time steps with the implicit schemes can affect the steady solutions, as mentioned in [4]. This is an initial study with implicit schemes and improvements of the implementation of these schemes with steady state solutions independent of the time step is a goal to be aimed in future works by this author.

9 Turbulence Models

9.1 Turbulence model of [14]

The problem of the turbulent simulation is in the calculation of the Reynolds stress. Expressions involving velocity fluctuations, originating from the average process, represent six new unknowns. However, the number of equations keeps the same and the system is not closed. The modeling function is to develop approximations to these correlations. To the calculation of the turbulent viscosity according to the [14] model, the boundary layer is divided in internal and external.

Initially, the (ν_w) kinematic viscosity at wall and the ($\tau_{xy,w}$) shear stress at wall are calculated. After that, the (δ) boundary layer thickness, the (δ_{LM}) linear momentum thickness and the (Vt_{BL}) boundary layer tangential velocity are calculated. So, the (N) normal distance from the wall to the studied cell is calculated. The N^+ term is obtained from:

$$N^+ = \sqrt{\mathrm{Re}}\sqrt{\tau_{xy,w}/\rho_w}\, N/\nu_w\,, \qquad (81)$$

where ρ_w is the wall density. The van Driest damping factor is calculated by:

$$D = 1 - e^{\left(-N^+\sqrt{\rho/\rho_w}\,\mu_w/\mu/A^+\right)}, \qquad (82)$$

with $A^+ = 26$ and μ_w is the wall molecular viscosity. After that, the (dVt/dN) normal to the wall gradient of the tangential velocity is calculated and the internal turbulent viscosity is given by:

$$\mu_{Ti} = \mathrm{Re}\,\rho(\kappa ND)^2\, dVt/dN\,, \qquad (83)$$

where κ is the von Kárman constant, which has the value 0.4. The intermittent function of Klebanoff is calculated to the external viscosity by:

$$g_{Kleb}(N) = \left[1 + 5.5(N/\delta)^6\right]^{-1}. \qquad (84)$$

With it, the external turbulent viscosity is calculated by:

$$\mu_{Te} = \mathrm{Re}(0.0168)\rho Vt_{BL}\delta_{LM}\, g_{Kleb}\,. \qquad (85)$$

Finally, the turbulent viscosity is chosen from the internal and the external viscosities: $\mu_T = MIN(\mu_{Ti}, \mu_{Te})$.

9.2 Turbulence model of [15]

To the calculation of the turbulent viscosity according to the [15] model, the boundary layer is again divided in internal and external. In the internal layer,

$$\mu_{Ti} = \rho l^2_{mix}\|\omega\| \quad \text{and} \quad l_{mix} = \kappa N\left(1 - e^{-N^+/A_0^+}\right). \qquad (86)$$

In the external layer,

$$\mu_{Te} = \rho\alpha C_{cp} F_{wake} F_{Kleb}(N; N_{\max}/C_{Kleb}) , \qquad (87)$$

with:

$$F_{wake} = MIN\left[N_{\max}F_{\max}; C_{wk}N_{\max}U_{dif}^{2}/F_{\max}\right],$$
$$F_{\max} = 1/\kappa\left[\underset{N}{MAX}\left(l_{mix}\|\omega\|\right)\right]. \qquad (88)$$

Hence, $N_{\max}$ is the value of N where $l_{mix}\|\omega\|$ reached its maximum value and l_{mix} is the Prandtl mixture length. The constant values are: $\kappa = 0.4$, $\alpha = 0.0168$, $A_0^{+} = 26$, $C_{cp} = 1.6$, $C_{Kleb} = 0.3$ and $C_{wk} = 1$. F_{Kleb} is the intermittent function of Klebanoff given by:

$$F_{Kleb}(N) = \left[1 + 5.5\left(C_{Kleb}N/N_{\max}\right)^{6}\right]^{-1}, \qquad (89)$$

$\|\omega\|$ is the magnitude of the vortex vector and U_{dif} is the maximum velocity value in the boundary layer case. To free shear layers,

$$U_{dif} = \left(\sqrt{u^2+v^2+w^2}\right)_{\max} - \left(\sqrt{u^2+v^2+w^2}\right)_{N=N_{\max}} . \qquad (90)$$

10 Spatially Variable Time Step

The idea of a spatially variable time step consists in keeping constant a CFL number in the calculation domain and to guarantee time steps appropriated to each mesh region during the convergence process. The spatially variable time step can be defined by:

$$\Delta t_{i,j,k} = \frac{CFL(\Delta s)_{i,j,k}}{\left(|q| + a\right)_{i,j,k}} , \qquad (91)$$

where CFL is the Courant-Friedrichs-Lewis number to method stability; $(\Delta s)_{i,j,k}$ is a characteristic length of information transport; and $\left(|q| + a\right)_{i,j,k}$ is the maximum characteristic speed of information transport, where a is the speed of sound. The characteristic length of information transport, $(\Delta s)_{i,j,k}$, can be determined by:

$$(\Delta s)_{i,j,k} = \left[MIN\left(l_{MIN}, C_{MIN}\right)\right]_{i,j,k} , \qquad (92)$$

where l_{MIN} is the minimum side length which forms a computational cell and C_{MIN} is the minimum distance of baricenters among the computational cell and its neighbors. The maximum characteristic speed of information transport is defined by $\left(|q| + a\right)_{i,j,k}$, with $q = \sqrt{u^2 + v^2 + w^2}$.

11 Initial and Boundary Conditions

11.1 Initial Condition

Stagnation values are used as initial condition to the nozzle problem. Only at the exit boundary is imposed a reduction of 1/3 to the density and to the pressure to start the flow along the nozzle ([23]). The vector of conserved variables is defined as:

(a) Domain except the nozzle exit:

$$Q_{i,j,k} = \{1 \quad 0 \quad 0 \quad 0 \quad (\gamma+1)/[2\gamma(\gamma-1)]\}^{T}; \qquad (93)$$

(b) Nozzle exit:

$$Q_{i,j,k} = \{1/3 \quad 0 \quad 0 \quad 0 \quad (\gamma+1)/[6\gamma(\gamma-1)]\}^{T}. \qquad (94)$$

To the compression corner and ramp problems, values of freestream flow are adopted for all properties as initial condition, in the whole calculation domain ([17, 23]):

$$Q_{i,j,k} = \left\{\begin{array}{c} 1 \\ M_\infty\cos\theta \\ M_\infty\sin\theta\cos\psi \\ M_\infty\sin\theta\sin\psi \\ \left[\dfrac{1}{\gamma(\gamma-1)} + \dfrac{M_\infty^2}{2}\right] \end{array}\right\}, \qquad (95)$$

where M_∞ represents the freestream Mach number, θ is the flow incidence angle upstream the configuration under study and ψ is the angle in the configuration longitudinal plane.

11.2 Boundary Conditions

Three types of boundary conditions are implemented in this work: wall, entrance and exit. They are implemented in special cells named “ghost cells”, as referred in the CFD community.

(a) Wall condition - The Euler case requires the flux tangency condition. On the context of finite volumes, this imposition is done considering that the tangent flow velocity component to the wall of the ghost cell be equal to the tangent flow velocity component to the wall of the neighbour real cell. At the same time, the normal flow velocity component to the wall of the ghost cell should be equal to the negative of the normal flow velocity component to the wall of the neighbour real cell. [24] suggests that these procedures lead to the following expressions to the velocity components *u*, *v* and *w* of the ghost cells:

$$u_g = (1-2n_x n_x)u_{real} + (-2n_x n_y)v_{real} + (-2n_x n_z)w_{real}; \quad (96)$$

$$v_g = (-2n_y n_x)u_{real} + (1-2n_y n_y)v_{real} + (-2n_y n_z)w_{real}; \quad (97)$$

$$w_g = (-2n_z n_x)u_{real} + (-2n_z n_y)v_{real} + (1-2n_z n_z)w_{real}. \quad (98)$$

In the viscous case, however, the velocity components of the ghost cells are set equal to corresponding values of the velocity components of the real neighbour, with opposite signal. In other words:

$$u_g = -u_{real},\ v_g = -v_{real} \ \text{ and } \ w_g = -w_{real}. \quad (99)$$

The fluid pressure gradient in the direction normal to the wall is equal to zero for the inviscid case and also equalled to zero in the viscous case due to the boundary layer theory. The temperature gradient is equal to zero along the whole wall, according to the condition of adiabatic wall, for both cases (viscous and non-viscous). With these two conditions, a zero order extrapolation is performed to the fluid pressure and to the temperature. It is possible to conclude that the fluid density will also be obtained by zero order extrapolation. The energy conserved variable is obtained from the state equation to a perfect gas.

(b) Entrance condition:

(b.1) Subsonic flow: Four properties are specified and one is extrapolated, based on analysis of information propagation along characteristic directions in the calculation domain ([23]). In other words, four characteristic directions of information propagation point inward the computational domain and should be specified. Only the characteristic direction associated to the "$(q_{normal}\text{-}a)$" velocity cannot be specified and should be determined from the interior information of the calculation domain. The *u* velocity component was the extrapolated variable from the real neighbour volume to the nozzle problem, whereas the pressure was the extrapolated variable to the compression corner and ramp problems. Density, pressure, and the *v* and *w* components of velocity had their values determined by isentropic and geometrical relations in the nozzle problem, whereas density and velocity components had their values determined by the freestream flow properties in the compression corner and ramp problems. 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: Four characteristic directions of information propagation point outward the computational domain and should be extrapolated from interior information ([23]). The characteristic direction associated to the "$(q_{normal}\text{-}a)$" 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 five characteristic directions of information propagation of the Euler equations point outward the calculation domain and, with it, nothing can be fixed.

12 Configurations of the Physical Problems and Employed Meshes

The geometry of the convergent-divergent nozzle at the xy plane is described in Fig. 6. The total length of the nozzle is 0.38ft (0.116m) and the throat height is equal to 0.090ft (0.027m). The throat is located at 0.19ft (0.058m) from the entrance boundary. The throat curvature ratio is equal to 0.090ft. The nozzle convergence angle is 22.33° and the nozzle divergence angle is 1.21°. An exponential stretching of 10% in both ξ and η directions was used. An algebraic mesh of 61 points in the ξ direction, 71 points in the η direction and 10 points in the ζ direction was generated, which corresponds in finite volumes to 37,800 hexahedrons and 43,310 nodes. Its spanwise length is 0.10ft (0.0305m). Figure 7 exhibits the mesh employed in the simulations.

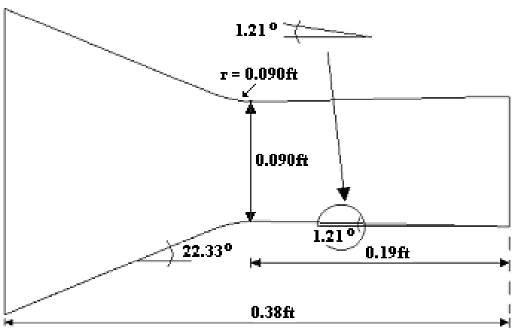

Figure 6 : Nozzle Configuration in the xy Plane.

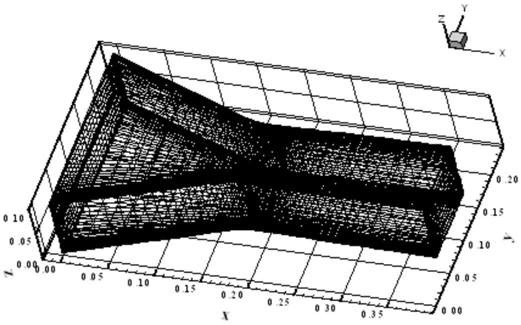

Figure 7 : Nozzle Mesh in Three-Dimensions.

The compression corner configuration at the xy plane is described in Fig. 8. The corner inclination angle is 10°. An algebraic mesh of 70 points in the ξ direction, 50 points in the η direction and 10 points in the ζ direction was generated, which corresponds in finite volumes to 30,429 hexahedrons and 35,000 nodes. Its spanwise length is 0.5m. Figure 9 exhibits such mesh.

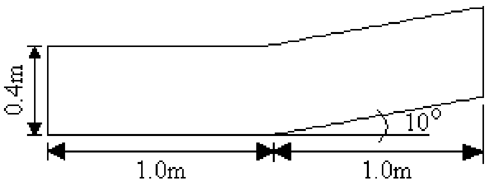

Figure 8 : Corner Configuration in the xy Plane.

Finally, the ramp configuration at the xy plane is described in Fig. 10. The compression corner has 20° of inclination. Its spanwise length is 0.25m. The mesh used in the simulations has 31,860 hexahedrons and 36,600 nodes to a structured discretization of the calculation domain. This mesh is equivalent, in finite differences, of being composed of 61 points in the ξ direction, 60 points in the η direction and 10 points in the ζ direction. An exponential stretching of 10% in the η direction was employed. Figure 11 shows such mesh. Table 1 presents a summary of the computational meshes.

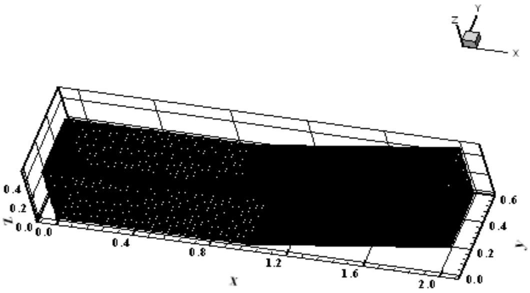

Figure 9 : Corner Mesh in Three-Dimensions.

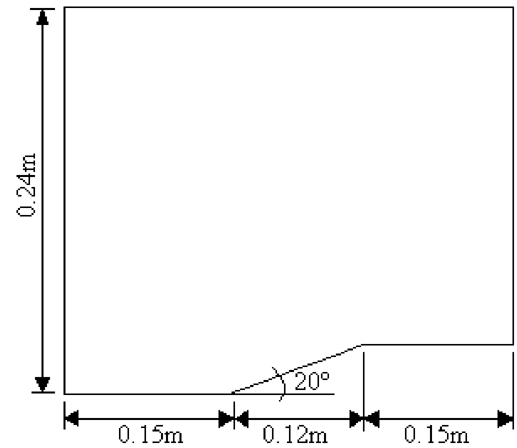

Figure 10 : Ramp Configuration in the xy Plane.

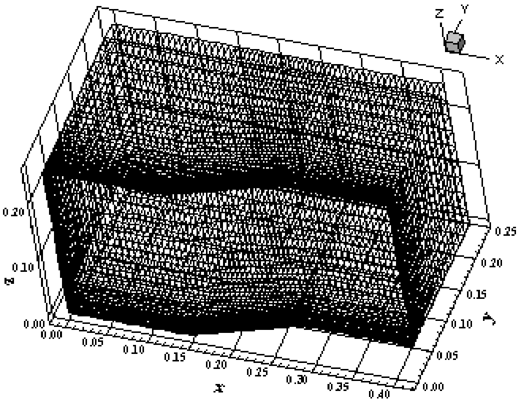

Figure 11 : Ramp Mesh in Three-Dimensions.

Table 1 : Computational data of the meshes.

	Nozzle	Corner	Ramp
	61x71x10	70x50x10	61x60x10
Cells	37,800	30,429	31,860
Nodes	43,310	35,000	36,600

13 Results

Tests were performed in a microcomputer with processor AMD SEMPRON (tm) 2600+, 1.83GHz, and 512 Mbytes of RAM memory. Converged results occurred to 3 orders of reduction in the maximum residual value. The configuration upstream and the configuration longitudinal plane angles were set equal to 0.0°. All pressure distributions were determined at the plane corresponding to k = KMAX/2, where “KMAX” is the maximum number of points in the z direction, and j = 1, corresponding to the configuration wall.

13.1 Inviscid results – Convergent-divergent nozzle

Stagnation flow was adopted as initial condition to this problem, with only a small reduction of the density and the pressure at the nozzle exit to initialize the flow.

Figures 12 to 15 show the pressure contours obtained by the [4], the [5], the [6], and the [7] schemes, respectively. The [7] scheme presented the most severe pressure field in relation to the other schemes, representing a more conservative scheme to this problem. Good symmetry characteristics are observed in all solution.

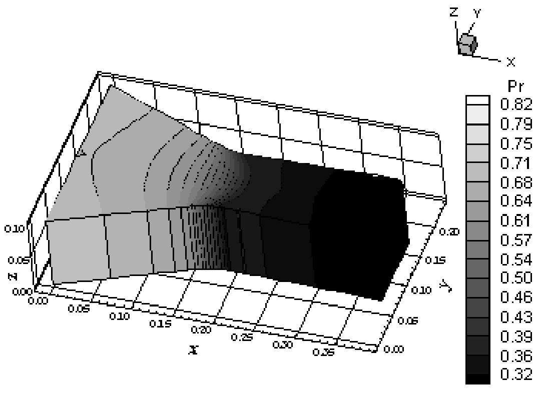

Figure 12 : Pressure Contours ([4]).

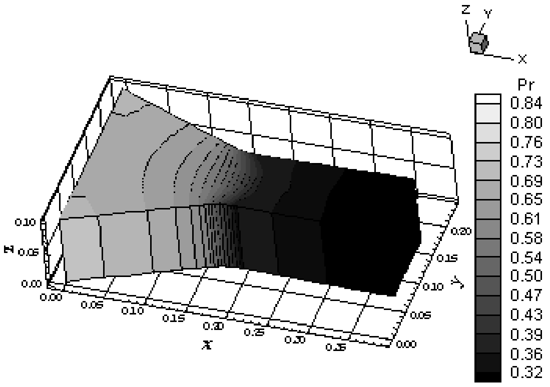

Figure 13 : Pressure Contours ([5]).

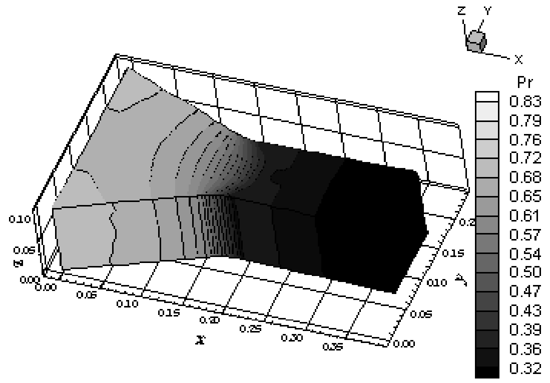

Figure 14 : Pressure Contours ([6]).

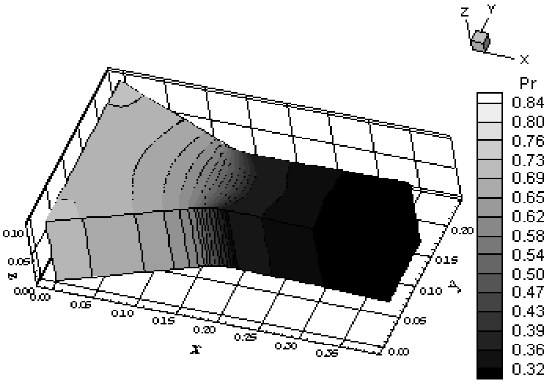

Figure 15 : Pressure Contours ([7]).

Figures 16 to 19 present the Mach number contours obtained by the schemes of [4], of [5], of [6], and of [7], respectively. The solution generated by the [6] scheme is the most intense, but there is loss of symmetry in the Mach field, which damages the solution. Disregarding this solution due to the error in the description of the Mach field, the most intense Mach number field is due to the [4] scheme. Despite of the [6] solution, all other results present good symmetry properties.

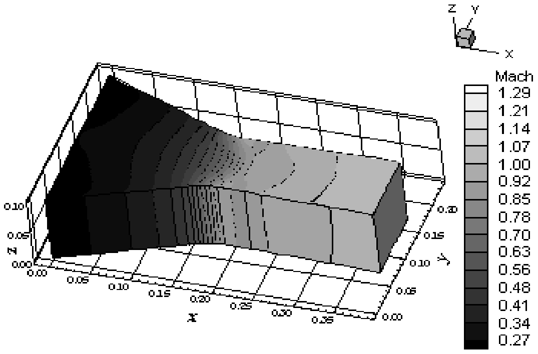

Figure 16 : Mach Contours ([4]).

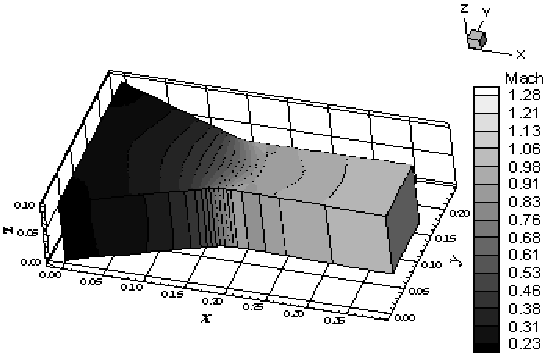

Figure 17 : Mach Contours ([5]).

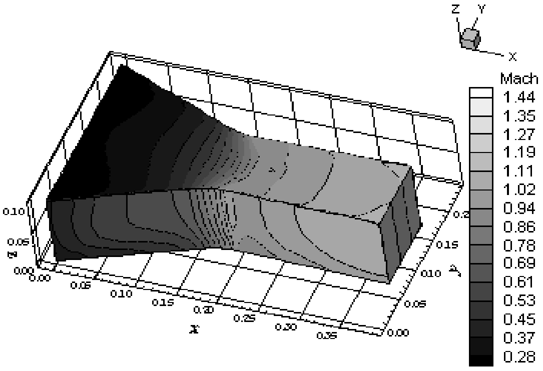

Figure 18 : Mach Contours ([6]).

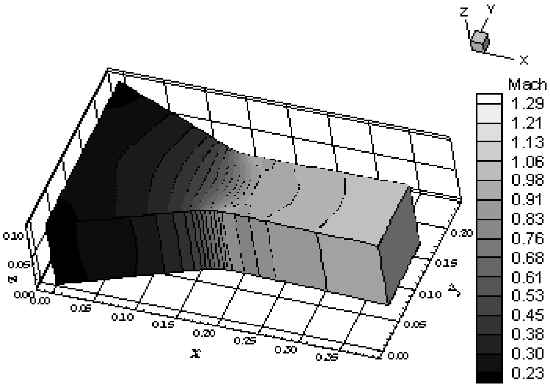

Figure 19 : Mach Contours ([7]).

Figure 20 exhibits the lower wall pressure distributions along the convergent-divergent nozzle. They are compared with the experimental results of [25]. As can be observed, the [4] scheme presents the best pressure distribution (closer to the experimental results).

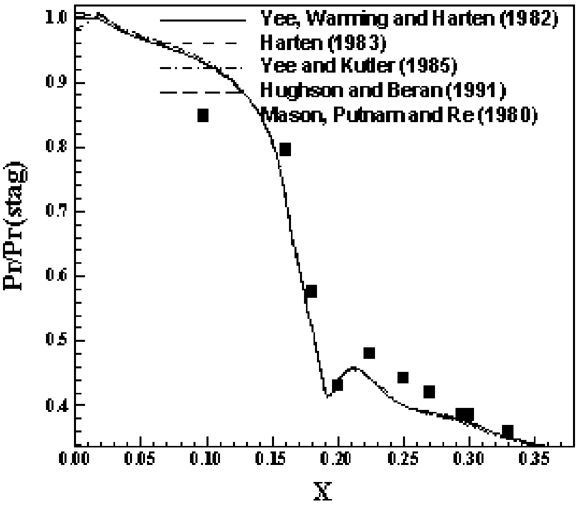

Figure 20 : Lower Wall Pressure Distributions.

13.2 Inviscid results – Compression corner

A freestream Mach number of 3.0, characterizing a moderate supersonic flow regime, was adopted as initial condition to this problem. The flow reaches the compression corner, generating an oblique shock wave along the corner.

Figures 21 to 24 exhibit the pressure contours obtained by the schemes of [4], of [5], of [6], and of [7], respectively. All solutions are clear, without pressure oscillations, and all pressure fields are equal in qualitative and quantitative terms.

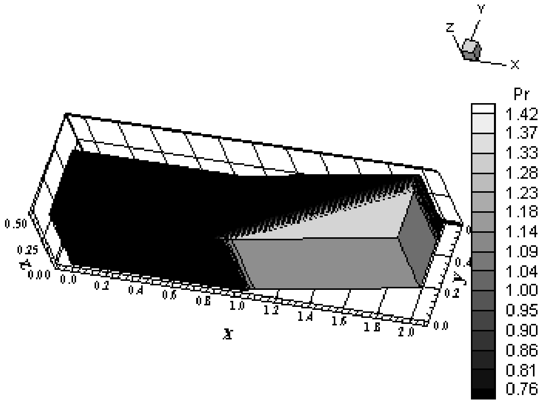

Figure 21 : Pressure Contours ([4]).

Figures 25 to 28 show the Mach number contours obtained by the [4], the [5], the [6], and the

[7] schemes, respectively. The most intense Mach number field is obtained by the [6] algorithm. All solutions are clear and without oscillations.

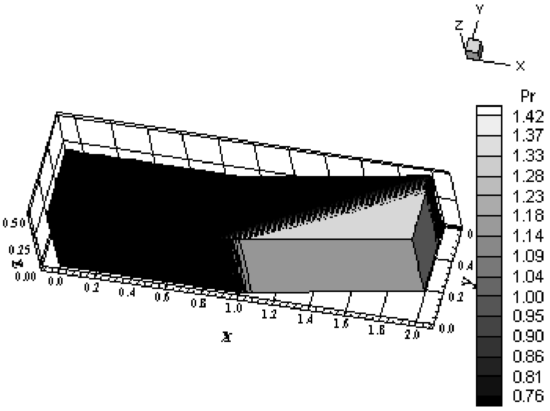

Figure 22 : Pressure Contours ([5]).

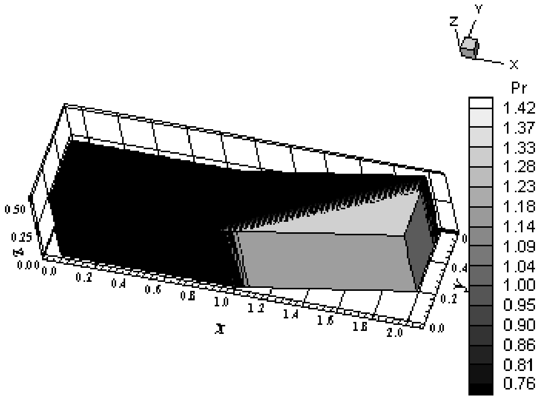

Figure 23 : Pressure Contours ([6]).

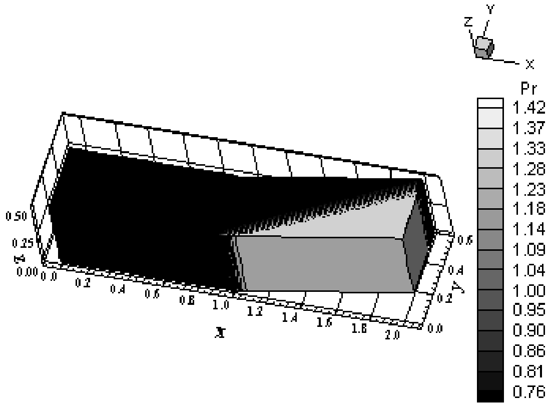

Figure 24 : Pressure Contours ([7]).

Figure 29 exhibits the wall pressure distributions along the compression corner obtained by all schemes. They are compared with the oblique shock wave theory results. As can be observed, all TVD schemes do not present oscillations at the discontinuity. Only the [6] solution presents a pressure distribution slightly under-predicted in relation to the other schemes. All schemes slightly under-predict the shock plateau. Figure 30 shows the same pressure distributions plotted using symbols to identify how much cells are necessary to capture the shock discontinuity. All schemes detect the shock discontinuity using four cells, which is an acceptable value to high resolution schemes.

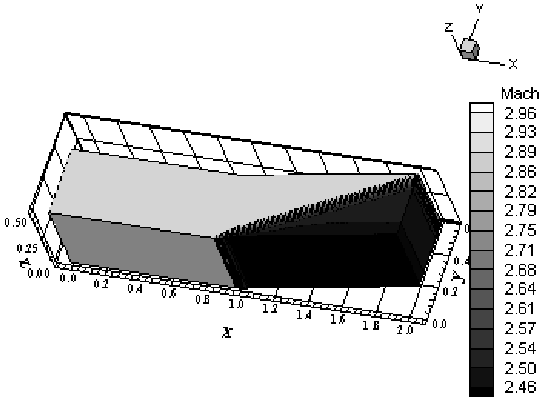

Figure 25 : Mach Contours ([4]).

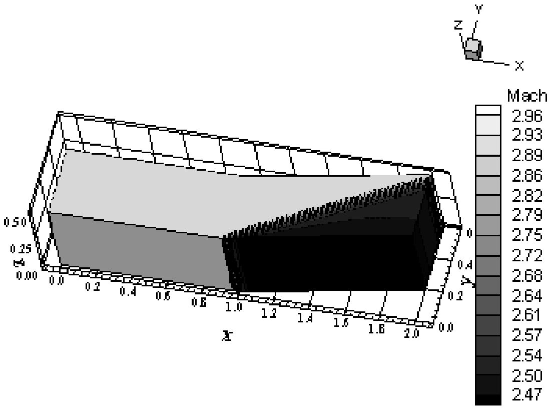

Figure 26 : Mach Contours ([5]).

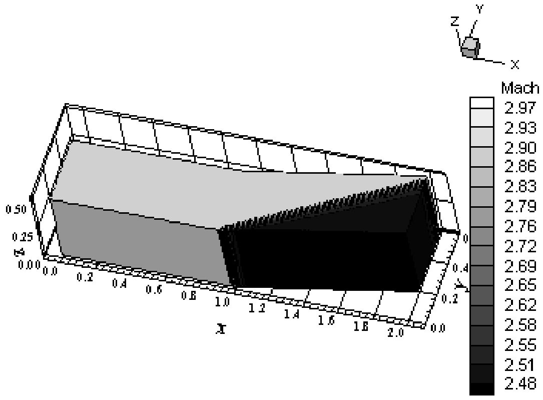

Figure 27 : Mach Contours ([6]).

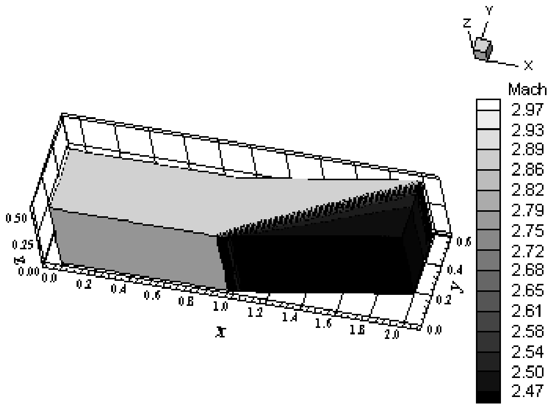

Figure 28 : Mach Contours ([7]).

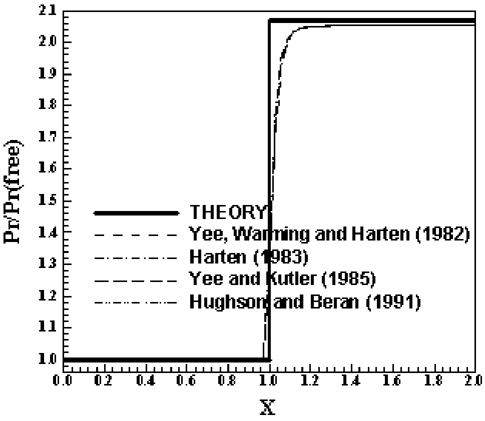

Figure 29 : Wall Pressure Distributions.

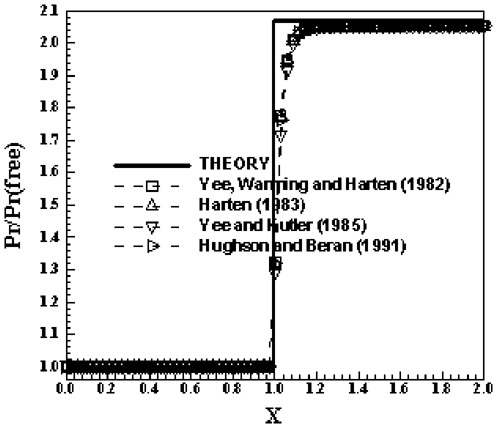

Figure 30 : Wall Pressure Distributions.

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. [26] (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. To the compression corner problem, ϕ = 10º (corner 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 21 to 24, considering the xy plane, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 2. As can be observed, the best scheme was the [6] one, with a percentage error of 0.00%.

Table 2 : Shock angle and respective percentage errors to the corner problem.

Algorithm	β(°)	Error (%)
[4]	27.90	1.45
[5]	27.80	1.09
[6]	27.50	0.00
[7]	28.00	1.82

13.3 Viscous results – Ramp

The physical problem studied in the viscous laminar and turbulent simulations is the flow along a ramp. This problem is a supersonic flow hitting a ramp with 20° of inclination. It generates a shock and an expansion fan. The freestream Mach number adopted as initial condition to this simulation was 5.0, characterizing a high supersonic flow. The Reynolds number was estimated to be 4.031x10^5 at a flight altitude of 20,000m and l_{REF} = 0.0437m, based on the work of [20].

13.2.1 Laminar results

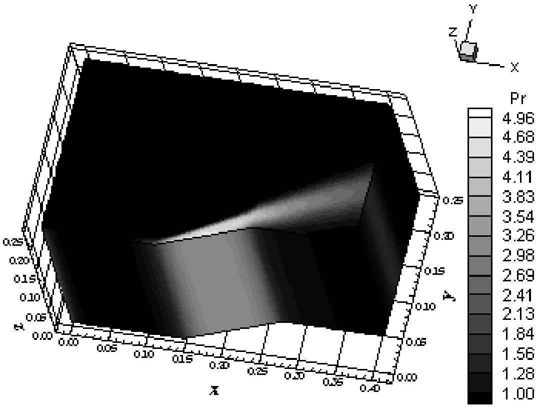

Figure 31 : Pressure Contours ([4]).

Figures 31 to 34 exhibit the pressure contours obtained by the [4], the [5], the [6], and the [7] schemes, respectively. The most severe pressure field, which characterizes the most conservative solution, is obtained by the [7] TVD scheme. Good symmetry and homogeneity properties at the k planes are observed in all solutions. The shock wave is well captured.

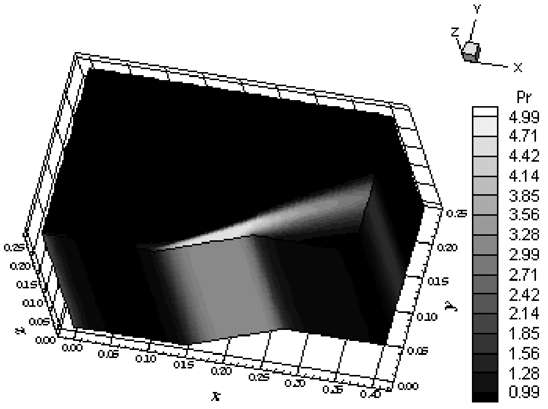

Figure 32 : Pressure Contours ([5]).

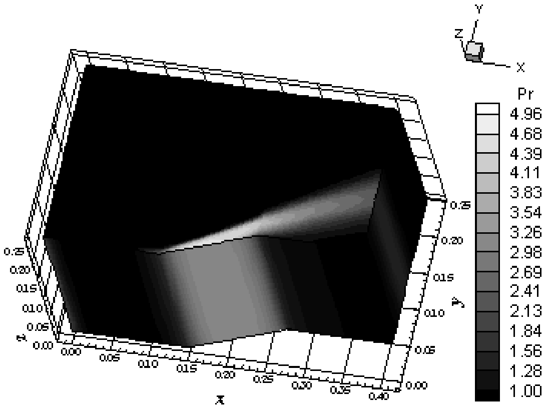

Figure 33 : Pressure Contours ([6]).

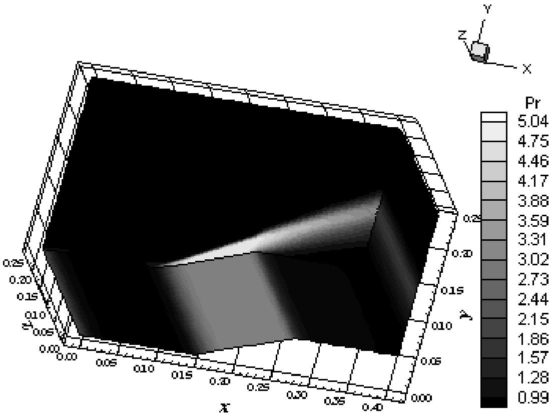

Figure 34 : Pressure Contours ([7]).

Figures 35 to 38 show the Mach number contours obtained by the schemes of [4], of [5], of [6], and of [7], respectively. The Mach number contours generated by the [7] TVD scheme is the most intense. No regions of separation flow are observed.

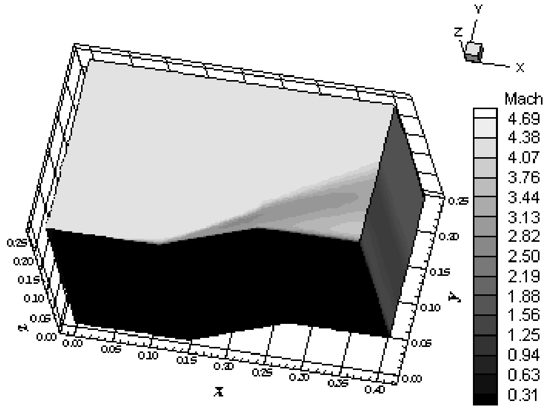

Figure 35 : Mach Contours ([4]).

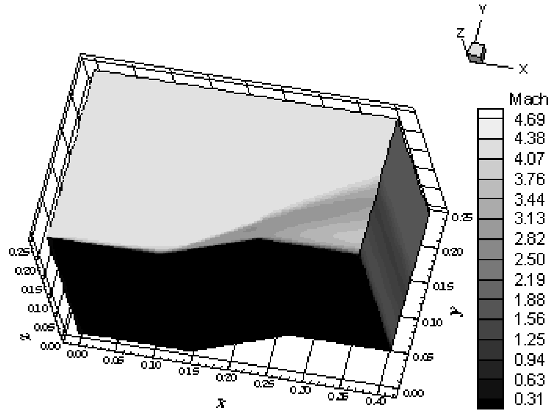

Figure 36: Mach Contours ([5]).

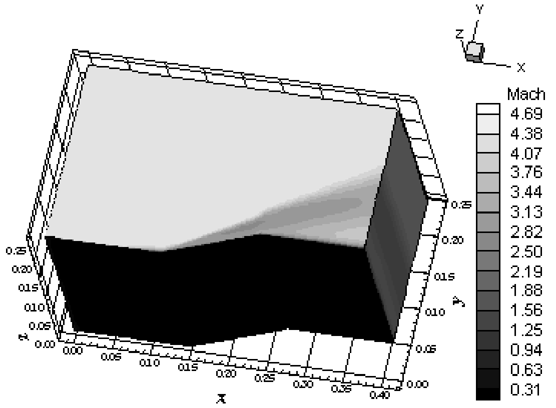

Figure 37: Mach Contours ([6]).

Again, 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. To the ramp problem, ϕ = 20º (ramp inclination angle) and the freestream Mach number is 5.0, resulting from this diagram a value to β equals to 30.0º. Using a transfer in Figures 31 to 34, considering the xy plane, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 3. As can be observed, the best scheme was the [7] TVD one, with an error of 0.00%.

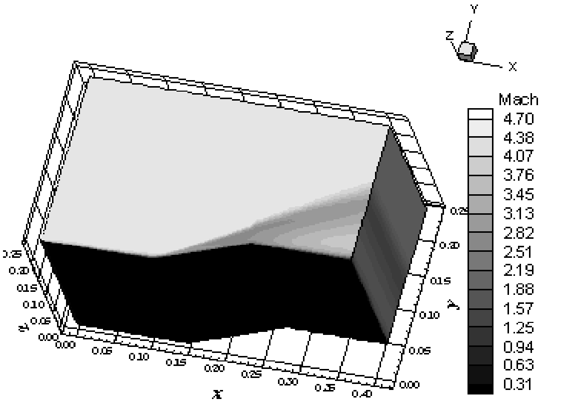

Figure 38 : Mach Contours ([7]).

Table 3 : Shock angle and respective percentage errors to the ramp problem (Laminar).

Algorithm	**β(°)**	**Error (%)**
[4]	29.50	1.67
[5]	30.40	1.33
[6]	29.50	1.67
[7]	30.00	0.00

13.2.2 Turbulent results – [14] model

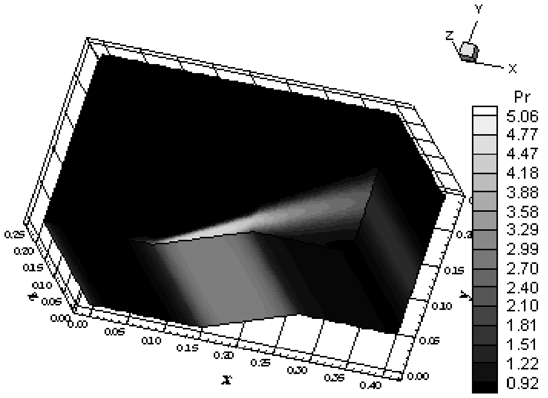

Figure 39: Pressure Contours ([4]).

Figures 39 to 42 show the pressure contours obtained by the [4], the [5], the [6], and the [7] schemes, respectively. The most severe pressure field was obtained by the [7] TVD scheme using the [14] model. Good symmetry and homogeneity properties are observed in all solutions. The shock wave is well captured.

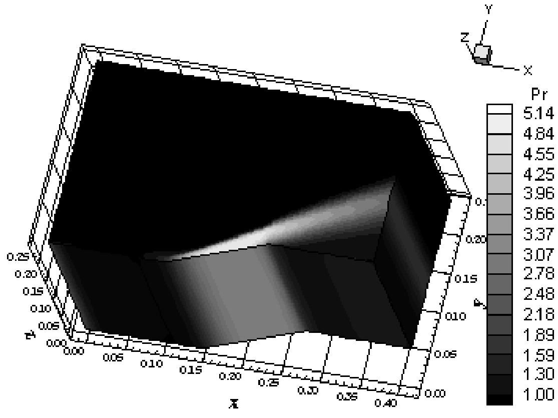

Figure 40 : Pressure Contours ([5]).

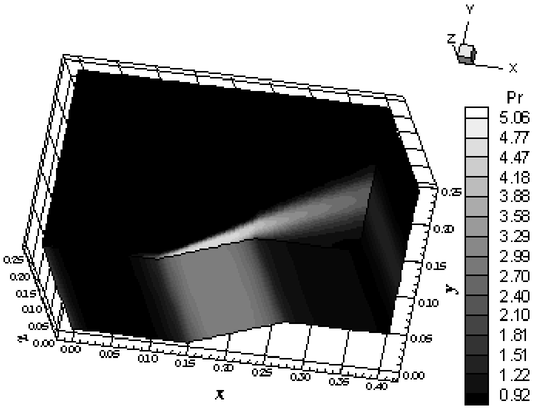

Figure 41 : Pressure Contours ([6]).

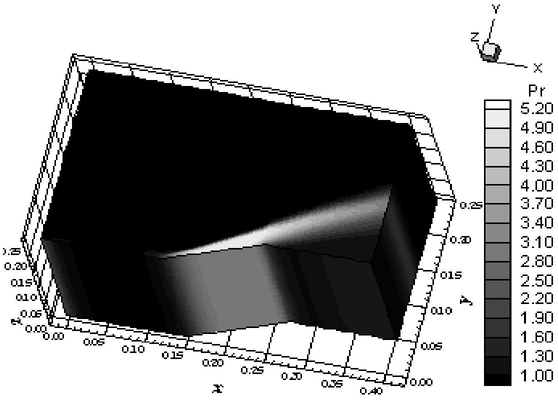

Figure 42 : Pressure Contours ([7]).

Figures 43 to 46 exhibit the Mach number contours obtained by the schemes of [4], of [5], of

[6], and of [7], respectively. The most intense Mach number fields are obtained by the [4, 6] schemes using the [14] model. However, it is important to note that these fields are more intense due to pre-shock oscillations that occur close to the wall, which characterizes an unphysical solution. Hence, considering the most correct value to the Mach number field, the [7] TVD scheme presents the most severe field.

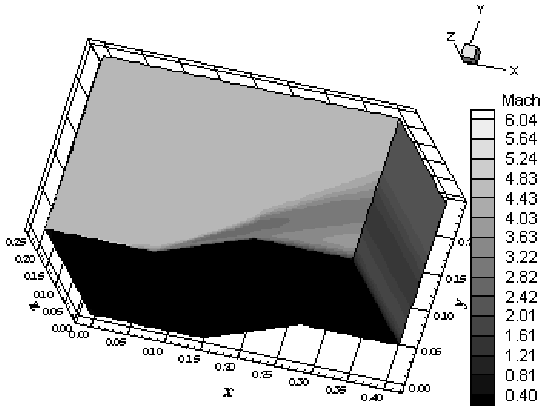

Figure 43 : Mach Contours ([4]).

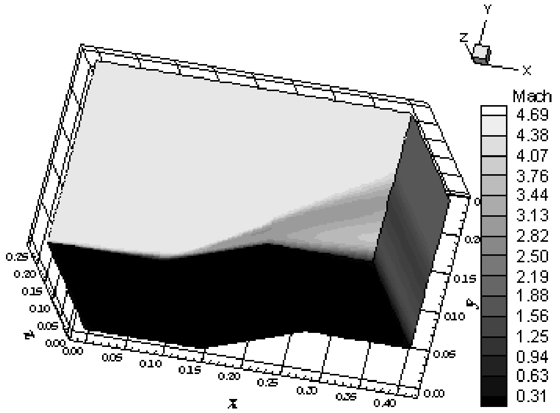

Figure 44 : Mach Contours ([5]).

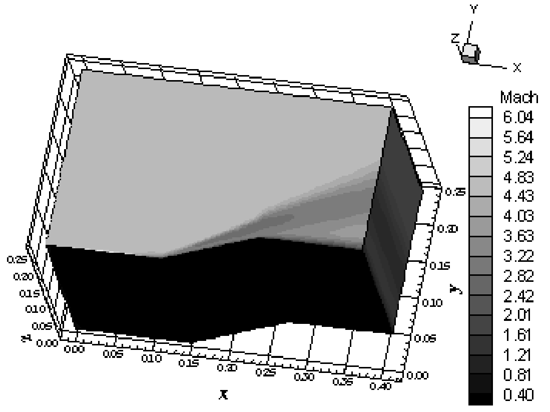

Figure 45 : Mach Contours ([6]).

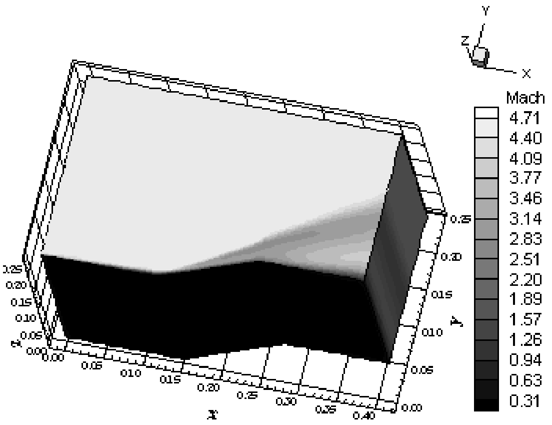

Figure 46 : Mach Contours ([7]).

Again, it is possible to determine by each scheme the shock angle of the oblique shock wave, β, measured in relation to the initial direction of the flow field. Using a transfer in Figures 39 to 42, considering the xy plane, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 4. As can be observed, the best scheme was the [4] TVD scheme, with an error of 0.00%.

Table 4 : Shock angle and respective percentage errors to the ramp problem ([14]).

Algorithm	β(°)	Error (%)
[4]	30.00	0.00
[5]	29.90	0.33
[6]	30.50	1.67
[7]	31.00	3.33

13.2.3 Turbulent results – [15] model

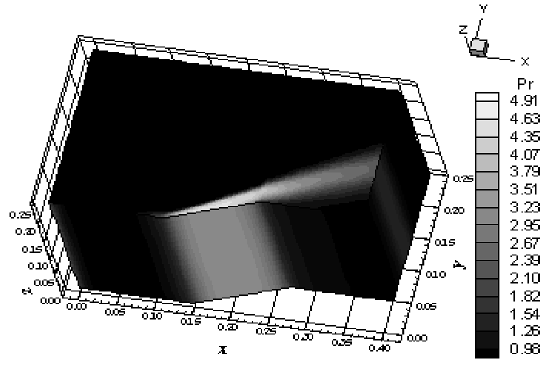

Figure 47 : Pressure Contours ([4]).

Figures 47 to 50 exhibit the pressure contours obtained by the [4], the [5], the [6], and the [7] schemes, respectively. The most severe pressure field is again obtained with the [7] scheme, using now the [15] model. All solutions present good symmetry and homogeneity properties. The shock wave is well captured.

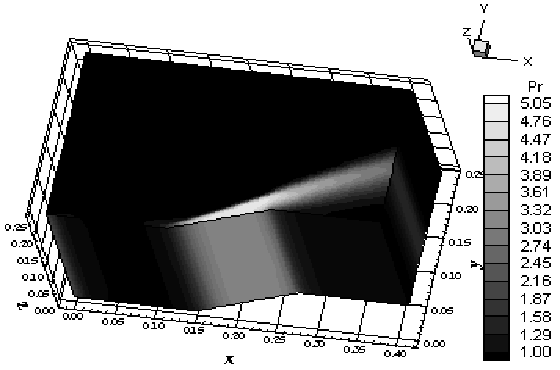

Figure 48 : Pressure Contours ([5]).

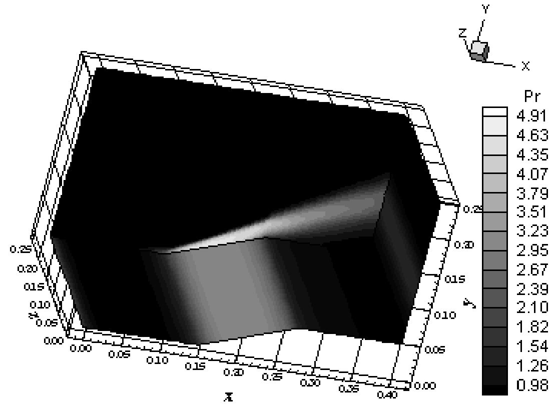

Figure 49 : Pressure Contours ([6]).

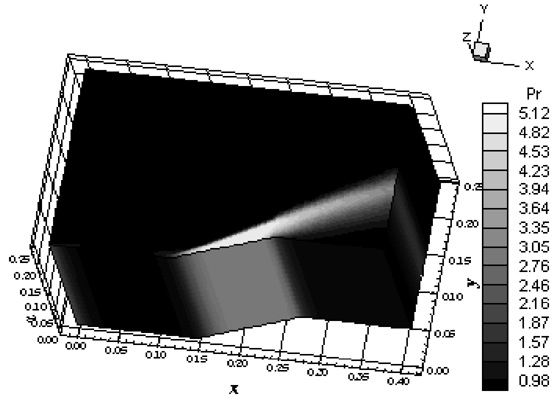

Figure 50 : Pressure Contours ([7]).

Figures 51 to 54 show the Mach number contours obtained by the schemes of [4], of [5], of [6], and of [7], respectively. The most intense Mach number fields are due to the [4] and the [6] TVD schemes. No pre-shock oscillations are observed with the [15] model, opposed to the behaviour observed with the [14] model.

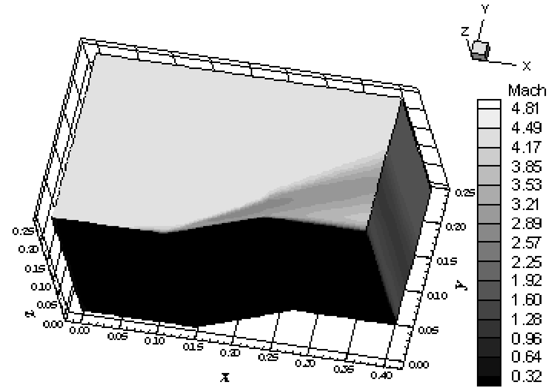

Figure 51 : Mach Contours ([4]).

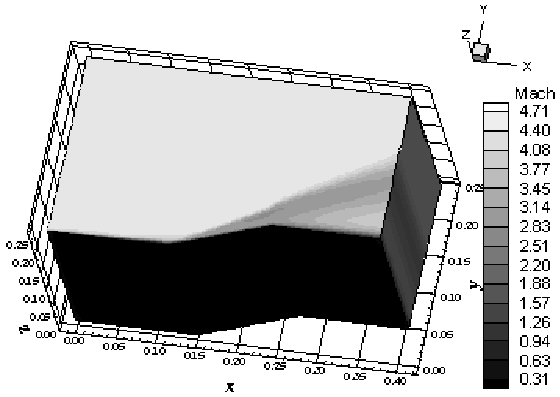

Figure 52 : Mach Contours ([5]).

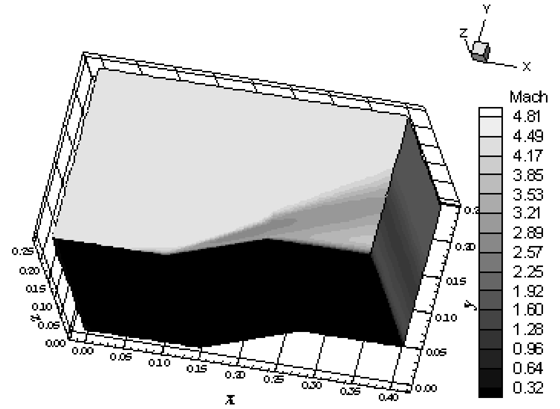

Figure 53 : Mach Contours ([6]).

Again, it is possible to determine by each scheme the shock angle of the oblique shock wave, β, measured in relation to the initial direction of the flow field. Using a transfer in Figures 47 to 50, considering the xy plane, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 5. As can be observed, the [6] scheme is the best one to this case.

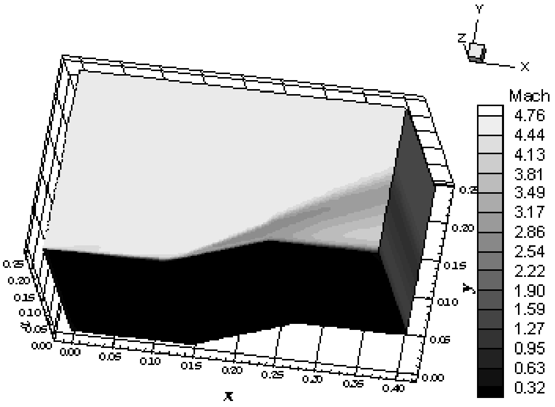

Figure 54 : Mach Contours ([7]).

Table 5 : Shock angle and respective percentage errors to the ramp problem ([15]).

Algorithm	β(°)	Error (%)
[4]	30.30	1.00
[5]	30.60	2.00
[6]	30.00	0.00
[7]	31.00	3.33

13.2.4 Comparison among wall pressure distributions, oblique shock angles and simulation data

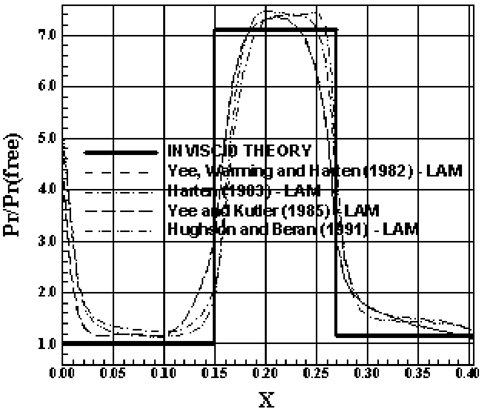

Figure 55 : Wall Pressure Distributions (Lam).

Figure 55 shows the laminar wall pressure distributions obtained by the [4-7] TVD schemes. These wall pressure distributions are compared with the inviscid solution, which is the true solution according to the boundary layer theory. As can be observed, the [7] TVD scheme presents the most intense pressure field. All solutions over-predict the pressure plateau at the ramp.

In Figure 56, the wall pressure distributions obtained by the all four schemes using the [14] model present the most intense pressure field generated again by the [7] scheme. Again, all solutions over-predict the pressure plateau at the ramp.

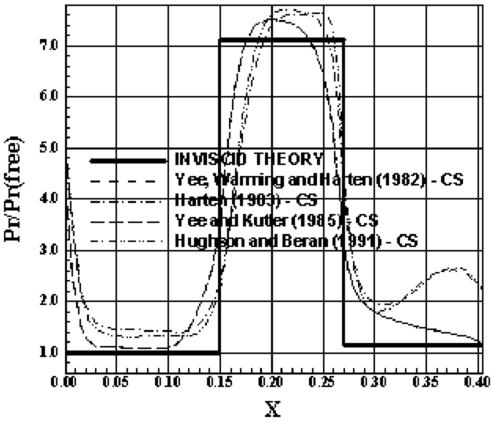

Figure 56 : Wall Pressure Distributions ([14]).

Figure 57 shows the wall pressure distributions generated by the all four schemes using the [15] model. Again, the [7] scheme presents the most severe pressure field. The best pressure distribution, in accordance with the boundary layer theory, which is represented by the inviscid theory, is due to [4] and [6] TVD schemes.

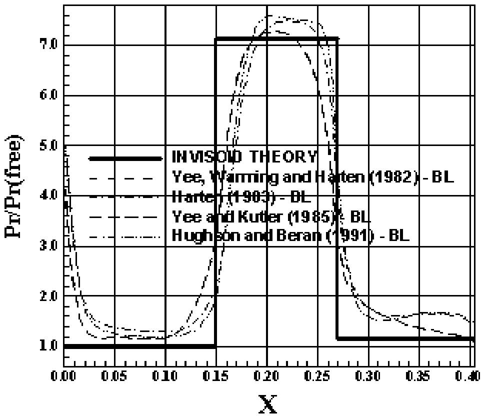

Figure 57 : Wall Pressure Distributions ([15]).

Figure 58 shows all wall pressure distributions obtained by the [4] TVD scheme, in all cases, laminar and turbulent. As can be observed, the best distribution, in relation to the boundary layer theory, occurred as the laminar and the [15] model were employed, whereas the most severe pressure distribution occurred as the [14] was used.

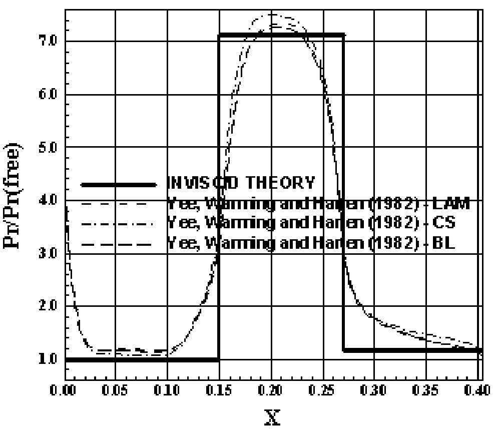

Figure 58 : Wall Pressure Distributions ([4]).

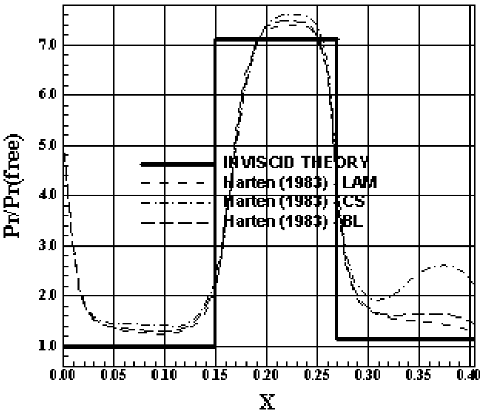

Figure 59 : Wall Pressure Distributions ([5]).

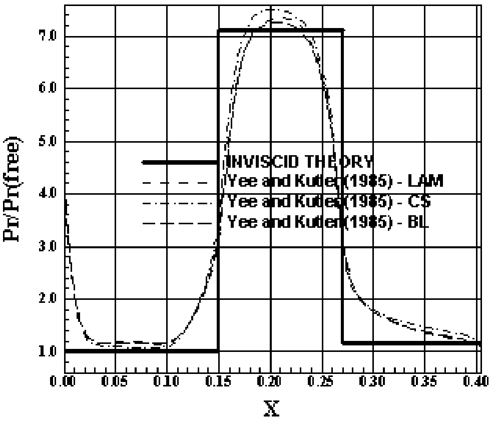

Figure 60 : Wall Pressure Distributions ([6]).

Figure 59 exhibits the wall pressure distributions generated by the [5] TVD scheme to the three cases, laminar and using the two turbulence models, compared with the theory of boundary layer and the best solution was again obtained by the laminar solution, whereas the [14] model predicts the most severe field. Figure 60 presents the wall pressure distribution obtained by the [6] TVD scheme to the three studied cases. The [14] model predicts the most severe pressure field, whereas the laminar and the [15] solutions are closer to the theoretical result. Finally, Figure 61 presents better pressure distribution generated by the [7] TVD scheme in the laminar case, whereas the most intense pressure field was obtained by using the [14] model.

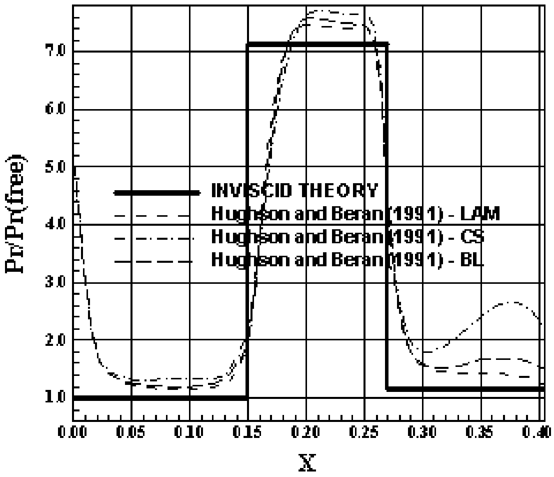

Figure 61 : Wall Pressure Distributions ([7]).

Aiming a global comparison involving the shock angle of the oblique shock waves estimated by the schemes in these viscous simulations, Tab. 6 exhibits the values calculated to these angles and respective errors.

Table 6 : Shock angle obtained in the laminar and turbulent cases to the ramp problem.

Laminar	[4]	[5]	[6]	[7]
β (°)	29.50	30.40	29.50	30.00
Error (%)	1.67	1.33	1.67	0.00
[14]	[4]	[5]	[6]	[7]
β (°)	30.00	29.90	30.50	31.00
Error (%)	0.00	0.33	1.67	3.33
[15]	[4]	[5]	[6]	[7]
β (°)	30.30	30.60	30.00	31.00
Error (%)	1.00	2.00	0.00	3.33

As can be observed, the [7] scheme presented the best estimative in the laminar case, the [4] scheme in the turbulent case using the [14] model, whereas the [6] scheme in the turbulent case using the [15] model. The global error was less than 3.5% to all schemes. As the most severe pressure fields are the most interesting results in terms of data to the project phase of airplanes and space vehicles, the solutions obtained with the [14] model, which in comparison with the laminar and the [15] model always presented the most severe field, are the intended results to an experienced aerodynamicist. Hence, the best value to be considered to the angle of the oblique shock wave, in terms of project analysis, is due to [4], to be, in general, a more conservative scheme in all three studied cases.

Table 7 presents the computational data of the numerical simulations in the inviscid case. All algorithms were implemented and tested in an implicit way and present converged results in the two problems. All schemes employed a CFL number of 1.5 in the nozzle problem, with the exception of the [6] scheme, which converged using a CFL number of 1.3. In the compression corner problem, all schemes converged using a CFL number of 4.3, with the exception again of the [6] scheme, which employed a CFL number of 3.2. The cheapest scheme is the [5] TVD scheme, whereas the most expensive is the [4] TVD scheme. The [4] TVD scheme is approximately 2.02% more expensive than the [5] TVD scheme, which represent a negligible difference.

Table 7 : Numerical data of the inviscid implicit simulations.

	Nozzle		Corner		
S(1)	CFL	Iterations	CFL	Iterations	Cost(2)
[4]	1.5	4,000	4.3	100	0.0002823
[5]	1.5	5,000	4.3	98	0.0002767
[6]	1.3	4,895	3.2	125	0.0002803
[7]	1.5	4,991	4.3	97	0.0002782

(1): S = Scheme; (2): Measured in seconds/per cell/per iterations.

Table 8 shows the computational data of the numerical simulations in the viscous laminar and turbulent cases in the ramp problem. All schemes to the viscous laminar and turbulent cases used an explicit formulation to the simulations. As can be observed, the fastest scheme is due to [7] TVD algorithm in two of the three cases.

Table 8 : Computational data of the explicit algorithms to the ramp viscous cases.

	Laminar		[14]		[15]	
S(1)	CFL	Iter.	CFL	Iter.	CFL	Iter.
[4]	0.2	9,251	0.2	7,459	0.2	10,677
[5]	0.3	4,395	0.3	4,095	0.3	4,440
[6]	0.2	9,251	0.2	7,459	0.2	10,677
[7]	0.3	4,212	0.3	3,904	0.3	4,475

(1) S = Scheme.

Table 9 exhibits the computational costs of the numerical algorithms obtained in the viscous laminar and turbulent cases. The cheapest algorithm in the laminar case is due to [5] TVD, while the most expensive is due to [4] TVD. In the turbulent case, using the [14] model, the cheapest algorithm is again due to [5] TVD, while the most expensive is again due to [4] TVD. Finally, using the [15] model, the cheapest algorithm is again the [5] TVD scheme, whereas the most expensive is again the [4] TVD scheme. As conclusion, in general the [5] TVD scheme yields the cheapest one in terms of viscous laminar and turbulent simulations.

Table 9 : Computational cost of the numerical algorithms to the ramp viscous laminar and turbulent cases.

	Computational Cost(1)		
Scheme	Laminar	[14]	[15]
[4]	0.0000495	0.0001287	0.0000615
[5]	0.0000463	0.0001231	0.0000562
[6]	0.0000476	0.0001274	0.0000603
[7]	0.0000465	0.0001232	0.0000566

(1): Measured in seconds/per cell/per iterations.

14 Conclusions

In the present work, the [4-7] schemes are implemented, on a finite volume context and using a structured spatial discretization, to solve the Euler and the laminar/turbulent Navier-Stokes equations in the three-dimensional space. All four schemes are TVD high resolution flux difference splitting ones, based on the concept of Harten's modified flux function. They are second order accurate in space. An implicit formulation is employed to solve the Euler equations in the inviscid problems. An approximate factorization in Linearized Nonconservative Implicit LNI form is employed by the flux difference splitting schemes. To solve the laminar/turbulent Navier-Stokes equations, an explicit formulation based on a time splitting procedure is employed. All schemes are first order accurate in time in their implicit and explicit versions. Turbulence is taken into account considering two algebraic models, namely: the [14-15] ones. The algorithms are accelerated to the steady state solution using a spatially variable time step, which has demonstrated effective gains in terms of convergence rate ([18-19]). All four schemes are applied to the solution of the physical problems of the transonic flow along a convergent-divergent nozzle and the supersonic flow along a compression corner in the inviscid case (Euler equations). To the laminar/turbulent viscous case, the supersonic flow along a ramp is solved.

The results have demonstrated that the most severe results are obtained with the [7] TVD high resolution scheme, whereas the [4] and the [6] schemes present more accurate results. In the inviscid case, it is possible to highlight that the [4] TVD scheme yields the best pressure distribution along the nozzle lower wall. In the compression corner, all schemes present good wall pressure distributions, slightly under-predicting the pressure plateau. Only the [6] scheme presents a wall pressure distribution under-predicted in relation to the other numerical schemes. The shock angle of the oblique shock wave that is formed at the compression corner is best estimated by the [6] TVD algorithm. The most expensive tested implicit scheme was due to [4] TVD scheme, whereas the cheapest was the [5] TVD scheme. The former is approximately 2.02% more expensive than the latter.

In the ramp viscous case, the laminar results present the [7] TVD scheme as yielding the best value to the shock angle at the ramp. The [14] model presents the [4] TVD scheme as yielding the best estimation, whereas in the [15] model, the [6] TVD scheme produces the best value to the shock angle. Considering the values estimated by the shock angle of the oblique shock wave, the [6] algorithm presents the best estimative to this parameter in two of the four cases – in the inviscid case and using the [15] turbulence model. Considering the nozzle lower wall pressure distribution, the [4] scheme predicts approximately this distribution and accurately the shock angle of the oblique shock wave, in the ramp problem, using the [14] model. Hence, in terms of accuracy, the [4] and the [5] algorithms are the most accurate among the tested schemes. As general conclusion in terms of viscous simulations, all algorithms present the most severe solution considering wall pressure distribution as using the [14] model. The [5] scheme was the cheapest considering laminar and turbulent cases, whereas the [4] scheme was the most expensive.

With these results, the main contributions of this work were accomplished, allowing that three-dimensional simulations, solving the Euler and the laminar/turbulent Navier-Stokes equations, were performed, employing four different TVD high resolution algorithms. Good results were provided, as expected, mainly in the capture of shock waves, as occurred in the nozzle and compression corner problems, and in the accurate prediction of the properties of the oblique shock wave in the viscous cases, as occurred in the ramp problem.

References:

[1] P. Kutler, Computation of Three-Dimensional, Inviscid Supersonic Flows, *Lecture Notes in Physics*, Vol. 41, 1975, pp. 287-374.

[2] J. L. Steger, Implicit Finite-Difference Simulation of Flow About Arbitrary Two-Dimensional Geometries, *AIAA Journal*, Vol. 16, No. 7, 1978, pp. 679-686.

[3] P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357-372.

[4] H. C. Yee, R. F. Warming, and A. Harten, A High Resolution Numerical Technique for Inviscid Gas-Dynamic Problems with Weak Solutions, *Lecture Notes in Physics*, Vol. 170, 1982, pp. 546-552, Springer Verlag, Berlin.

[5] A. Harten, High Resolution Schemes for Hyperbolic Conservation Laws, *Journal of Computational Physics*, Vol. 49, No. 2, 1983, pp. 357-393.

[6] H. C. Yee, and P. Kutler, Application of Second-Order-Accurate Total Variation Diminishing (TVD) Schemes to the Euler Equations in General Geometries, *NASA-TM-85845*, 1985.

[7] M. C. Hughson, and P. S. Beran, Analysis of Hyperbolic Blunt-Body Flows Using a Total Variation Diminishing (TVD) Scheme and the

MacCormack Scheme, *AIAA 91-3206-CP*, 1991.

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

[9] 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.

[10] J. Douglas, On the Numerical Integration of $u_{xx}+u_{yy}=u_t$ by Implicit Methods, *Journal of the Society of Industrial and Applied Mathematics*, Vol. 3, 1955, pp. 42-65.

[11] D. W. Peaceman, and H. H. Rachford, The Numerical Solution of Parabolic and Elliptic Differential Equations, *Journal of the Society of Industrial and Applied Mathematics*, Vol. 3, 1955, pp. 28-41.

[12] J. Douglas, and J. E. Gunn, A General Formulation of Alternating Direction Methods, *Numerische Mathematiks*, Vol. 6, 1964, pp. 428-453.

[13] N. N. Yanenko, *The Method of Fractional Steps*, Springer Verlag, NY, EUA, 1971.

[14] T. Cebeci, and A. M. O. Smith, A Finite-Difference Method for Calculating Compressible Laminar and Turbulent Boundary Layers, *Journal of Basic Engineering*, Trans. ASME, Series B, Vol. 92, No. 3, 1970, pp. 523-535.

[15] B. D. Baldwin, and H. Lomax, Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows, *AIAA Paper 78-257*, 1978.

[16] E. S. G. Maciel, Turbulent Flow Simulations Using the MacCormack and the Jameson and Mavriplis Algorithms Coupled with the Cebeci and Smith and the Baldwin and Lomax Models in Three-Dimensions, *Engineering Applications of Computational Fluid Mechanics*, Vol. 1, No. 3, 2007, pp. 147-163.

[17] A. Jameson and 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] 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.

[19] 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.

[20] R. W. Fox, and A. T. McDonald, *Introdução à Mecânica dos Fluidos*, Ed. Guanabara Koogan, Rio de Janeiro, RJ, Brazil, 632p, 1988.

[21] L. N. Long, M. M. S. Khan, and H. T. Sharp, Massively Parallel Three-Dimensional Euler / Navier-Stokes Method, *AIAA Journal*, Vol. 29, No. 5, 1991, pp. 657-666.

[22] H. C. Yee, R. F. Warming, and A. Harten, Implicit Total Variation Diminishing (TVD) Schemes for Steady-State Calculations, *Journal of Computational Physics*, Vol. 57, No. 3, 1985, pp. 327-360.

[23] E. S. G. Maciel, 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.

[24] J. T. Batina, Implicit Upwind Solution Algorithms for Three-Dimensional Unstructured Meshes, *AIAA Journal*, Vol. 31, No. 5, 1993, pp. 801-805.

[25] M. L. Mason, L. E. Putnam, and R. J. Re, The Effect of Throat Contouring on Two-Dimensional Converging-Diverging Nozzles at Sonic Conditions, *NASA Technical Paper 1704*, 1980.

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