First Order Unstructured Algorithms Applied to the Solution of the Euler Equations in Three-Dimensions

Edisson Sávio de Góes Maciel
Departamento de Energia Nuclear (DEN)
Universidade Federal de Pernambuco (UFPE)
Av. Prof. Luiz Freire, 1000 – Cidade Universitária – CEP 50740-540
Brazil
edissonsavio@yahoo.com.br

Abstract: - In the present work, the Roe, the Steger and Warming, the Van Leer, the Harten, the Frink, Parikh and Pirzadeh, the Liou and Steffen Jr. and the Radespiel and Kroll schemes are implemented, on a finite volume context and using an upwind and unstructured spatial discretization, to solve the Euler equations in the three-dimensional space. The Roe, the Harten, and the Frink, Parikh and Pirzadeh schemes are flux difference splitting ones, whereas the others schemes are flux vector splitting ones. All seven schemes are first order accurate in space. The time integration uses a Runge-Kutta method and is second order accurate. The physical problems of the supersonic flow along a ramp and the "cold gas" hypersonic flow along a diffuser are solved. The results have demonstrated that the Liou and Steffen Jr. scheme is the most conservative algorithm among the studied ones, whereas the Van Leer scheme is the most accurate.

Key-Words: - Flux difference splitting algorithms, Flux vector splitting algorithms, Unstructured schemes, Euler equations, Three-Dimensions, Supersonic and hypersonic flows.

1 Introduction

Conventional non-upwind algorithms have been used extensively to solve a wide variety of problems ([1] and [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], [4], [5], [6], [7], [8] and [9]. 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] developed a method that used the remarkable property that the nonlinear flux vectors of the inviscid gasdynamic equations in conservation law form were homogeneous functions of degree one of the vector of conserved variables. This property readily permitted the splitting of the flux vectors into subvectors by similarity transformations so that each subvector had associated with it a specified eigenvalue spectrum. As a consequence of flux vector splitting, new explicit and implicit dissipative finite-difference schemes were developed for first-order hyperbolic systems of equations.

[5] suggested an upwind scheme based on the flux vector splitting concept. This scheme considered the fact that the convective flux vector components could be written as flow Mach number polynomial functions, as main characteristic. Such polynomials presented the particularity of having the minor possible degree and the scheme had to satisfy seven basic properties to form such polynomials. This scheme was presented to the Euler equations in Cartesian coordinates and three-dimensions.

[6] developed a class of new finite difference schemes, explicit and with second order of spatial accuracy to 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.

[7] proposed a new scheme, unstructured and upwind, to the solution of the Euler equations. The scheme was based on the [3] flux difference splitting algorithm and was first order accurate. High resolution was obtained using a linear extrapolation process based on conserved variables. They tested the precision and the utility of this scheme in the analysis of the inviscid flows around two airplane configurations: one of transport configuration, with turbines under the wings, and the other of high speed civil configuration. Tests were accomplished at subsonic and transonic Mach numbers with the transport airplane and at transonic and low supersonic Mach numbers with the civil airplane, yielding good results.

[8] proposed a new flux vector splitting scheme. They declared that their scheme was simple and its accuracy was equivalent and, in some cases, better than the [3] scheme accuracy in the solutions of the Euler and the Navier-Stokes equations. The scheme was robust and converged solutions were obtained so fast as the [3] scheme. The authors proposed the approximated definition of an advection Mach number at the cell face, using its neighbor cell values via associated characteristic velocities. This interface Mach number was so used to determine the upwind extrapolation of the convective quantities.

[9] emphasized that the [8] scheme had its merits of low computational complexity and low numerical diffusion as compared to others methods. They also mentioned that the original method had several deficiencies. The method yielded local pressure oscillations in the shock wave proximities, adverse mesh and flow alignment problems. In the [9] work, a hybrid flux vector splitting scheme, which alternated between the [8] scheme and the [5] scheme, in the shock wave regions, is proposed, assuring that resolution of strength shocks was clear and sharply defined.

On an unstructured algorithm context, [10] and [11] have presented works involving the numerical implementation of two typical algorithms of the Computational Fluid Dynamics community in the two-dimensional space. The [12] and the [7] algorithms were implemented on an unstructured spatial discretization context. The [12] scheme was symmetrical and the [13] artificial dissipation operator was implemented aiming to guarantee the scheme stability. The [7] scheme was upwind and of flux difference splitting type based on [3] method, as commented above. The [12] scheme was second order accurate in space and time, while the [7] scheme was first order accurate in space and second order accurate in time. The Euler equations in conservative form were solved. The physical problems of the transonic flow around a NACA 0012 airfoil and the supersonic flow around a simplified version of the VLS (Brazilian Satellite Launcher Vehicle) were studied and good results were obtained, highlighting better solution quality and convergence acceleration features to the [12] scheme.

[14] have presented a work involving the [12] and the [8] unstructured algorithms applied to solution of the Euler and of the Navier-Stokes equations in three-dimensions. The governing equations in conservative form were solved, employing a finite volume formulation and an unstructured spatial discretization. The [8] algorithm was implemented in its first order version. Both schemes used a second-order Runge-Kutta method to perform time integration. The steady state problems of the supersonic flow along a ramp and of the “cold gas” hypersonic flow along a diffuser were studied. The results have demonstrated good quality and quantitative features of both tested schemes, with the [8] scheme being more robust.

In the present work, the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes are implemented, on a finite volume context and using an upwind and unstructured spatial discretization, to solve the Euler equations in the three-dimensional space. The [3], the [6], and the [7] schemes are flux difference splitting ones and more accurate solutions are expected. On the other hand, the [4], the [5], the [8], and the [9] schemes are flux vector splitting ones and more robustness properties are expected. The implemented schemes are first order accurate in space. The time integration uses a Runge-Kutta method and is second order accurate. The physical problems of the supersonic flow along a ramp and the “cold gas” hypersonic flow along a diffuser are solved. All the seven algorithms are accelerated to the steady state solution using a spatially variable time step. This technique has proved excellent gains in terms of convergence ratio as reported in [20].

The results have demonstrated that the [8] scheme is the most conservative algorithm among the studied ones, whereas the [5] scheme is the most accurate.

An unstructured discretization of the calculation domain is usually recommended to complex configurations, due to the easily and efficiency that such domains can be discretized ([13] and [15]).

However, the unstructured mesh generation question will not be studied in this work.

2 Euler Equations

The fluid movement is described by the Euler equations, which express the conservation of mass, of linear momentum and of energy to an inviscid, heat non-conductor and compressible mean, in the absence of external forces. In the integral and conservative forms, employing a finite volume formulation and using a structured spatial discretization, to three-dimensional simulations, these equations can be represented by:

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

where Q is written to a Cartesian system, V is a cell volume, which corresponds to an tetrahedron in the three-dimensional space, n_x, n_y and n_z are the components of the normal unity vector pointing outward to the flux face, S is the surface area and E_e, F_e and G_e represent the components of the convective flux vector. Q, E_e, F_e and G_e are represented by:

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

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

The quantities that appear above are described as follows: 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 per unit volume of the fluid; and p is the fluid static pressure.

The Euler equations were nondimensionalized in relation to the freestream density, ρ_∞, and the freestream speed of sound, a_∞, for the studied 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:

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

where γ is the ratio of specific heats at constant pressure and volume, respectively, which assumed a value 1.4 to the atmospheric air. The total enthalpy is determined by:

$$H = (e + p)/\rho . \quad (4)$$

3 Geometrical Characteristics of the Spatial Discretization

A given computational cell in structured 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.

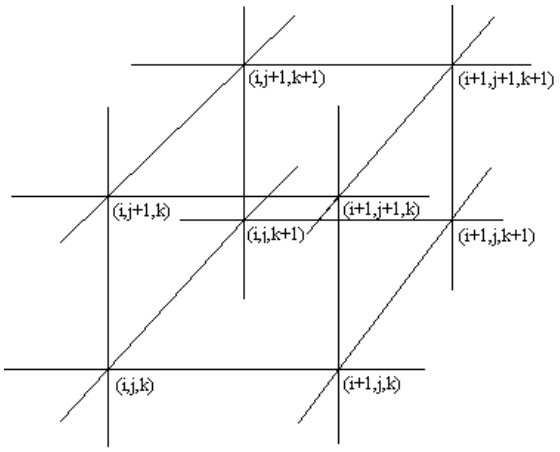


Figure 1 : Computational cell to structured discretization.

A computational cell on an unstructured context is formed by decomposing the given hexahedron in its six tetrahedral cells. Figure 2 exhibits the division of a hexahedron in its six tetrahedral components, as well the nodes of the vertices which define each tetrahedron, and Fig. 3 shows the isolated computational cell.

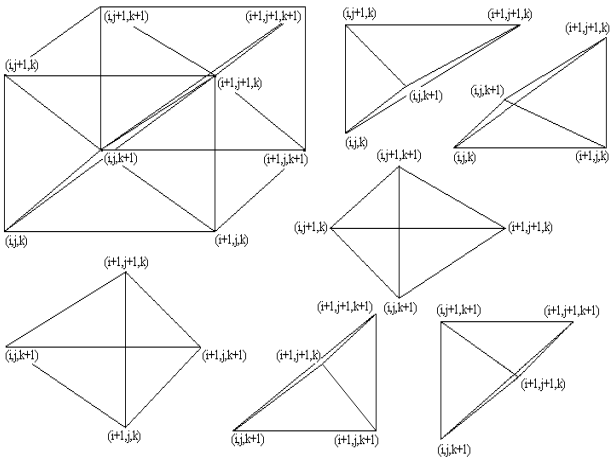


Figure 2 : Definition of a hexahedron and its six tetrahedral components (structured).

Each tetrahedron is identified by the index "i" and its four nodes n1, n2, n3 and n4. Three tables supply the necessary data to the execution of the algorithm solution. The connectivity table supplies the nodes which define a given tetrahedron; a neighboring table supplies the four neighbors which are around the volume "i", including the boundary cells, namely "ghost" cells; and the node-coordinate table which supplies the x, y and z coordinates of the mesh to each node.

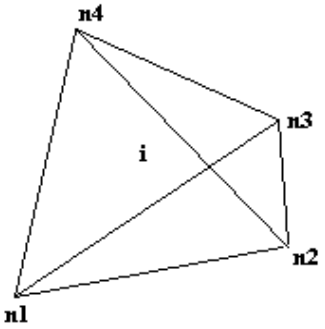


Figure 3 : Computational cell to unstructured discretization.

As the cell of the neighboring table is a ghost one, instead of its four neighbors, it is indicated the unique real neighbor cell which shares the boundary of the computational domain with this ghost cell and the type of ghost cell that is being used. The types of ghost cells vary from 1 to 6, being them: 1 – Wall ghost cell, 2 – Exit ghost cell, 3 – Far field ghost cell, 4 – Entrance ghost cell, 5 – Lateral ghost cell, and 6 – Lateral ghost cell.

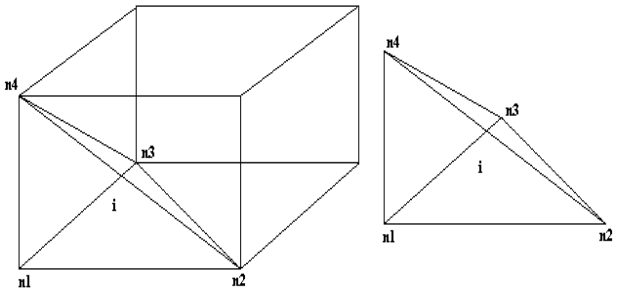
The lateral ghost cells are related to the lateral boundaries of the computational domain. In others words, there is the geometry plane (k = 1) and the parallel planes to the geometry, which in this case are the same (k = 2, k = 3, etc.). The planes k = 0 and k = kmax, according to a structured notation, only to comprehension, are the planes which encompass the ghost cells and are denominated lateral planes in the present work. These tables, as well the codification presented to the ghost cells are generated in a separated computer program as a pre-processing stage.

To the calculation of the volume of each tetrahedral cell, it is necessary to use the information of the connectivity table. The connectivity table supplies the four nodes which define a given tetrahedral cell. In function of these four nodes, it is possible to determine the volume of a hexahedron composed by these four base nodes. The modulus of the mixed product $\left(\vec{a}\times\vec{b}\right)\bullet\vec{c}$ defines the volume of the hexahedron, where "×" represent external product and "•" represents the vector inner product. The vectors $\vec{a},\vec{b}$ and $\vec{c}$ are defined of the following way: $\vec{a}$ is the vector formed by nodes 1 and 2, pointing from 1 to 2; $\vec{b}$ is the vector formed by nodes 1 and 3, pointing from 1 to 3; and $\vec{c}$ is the vector formed by nodes 1 and 4, pointing from 1 to 4. Hence, one-sixth of this volume is the volume of the tetrahedron under study. In others words, the hypothesis is that this hexahedron is composed of six tetrahedra equal to the formed by the nodes 1, 2, 3 and 4. The graphical representation of this procedure is exhibited in Fig. 4. The same result to the calculation of the tetrahedron volume is obtained calculating the determinant below:

$$V_{1234}=\frac{1}{6}\begin{vmatrix} x_1 & y_1 & z_1 & 1\\ x_2 & y_2 & z_2 & 1\\ x_3 & y_3 & z_3 & 1\\ x_4 & y_4 & z_4 & 1 \end{vmatrix}, \qquad (5)$$

where x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, x_4, y_4 and z_4 are Cartesian coordinates of the nodes which define the tetrahedron represented in Fig. 4.

The flux area of a tetrahedral cell is calculated by half the external product $\vec{a}x\vec{b}$, as indicated in Fig. 5. In this figure, it is possible to perceive that the vector $\vec{a}$ is formed by the nodes 1 and 2, pointing from 1 to 2, and the vector $\vec{b}$ is formed by the nodes 2 and 3, pointing from 2 to 3.



Figur 4 : Calculation of the volume of a tetrahedron.

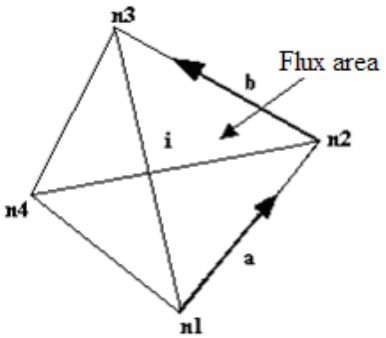


Figure 5 : Flux area (tetrahedron).

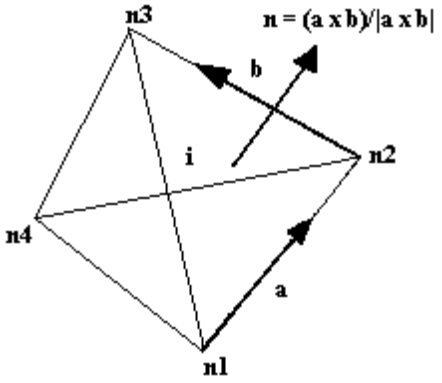


Figure 6 : Normal vector (tetrahedron).

The normal vector to each flux face is obtained by the external product $\vec{a}x\vec{b}$, divided by its norm, as indicated in Fig. 6. There is no specific rule in the determination of the sense of the unit vector, which implies that an additional test considering the node opposed to the face defined by $\vec{a}$ and $\vec{b}$ should be performed to determine the orientation of the unit vector. This test is based on the following vector mixed product $\left[(\vec{a}\,x\,\vec{b})/\left|\vec{a}\,x\,\vec{b}\right|\right]\bullet\,\vec{f}$, where $\vec{f}$ is the vector formed by one of the nodes of the flux face under study and the node of the tetrahedron that is immediately opposed to this face. The positive signal indicates that the normal vector is pointing inward the tetrahedron, what imposes that it should be changed by their opposed vector.

4 Explicit Numerical Algorithm of [3] – Flux Difference Splitting Scheme

The [3] algorithm, first order accurate in space, is specified by the determination of the numerical flux vector at the "*l*" interface.

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

$$V_R = V_{ne},\ V_L = V_i \quad \text{and} \quad V_l = 0.5(V_R + V_L), \quad (6)$$

where "*R*" and "*L*" represent right and left states, respectively, and "*ne*" represent a neighbor cell to the "*i*" cell. The subscript "*L*" is associated to properties of a given "*i*" cell and the subscript "*R*" is associated to properties of the "*ne*" neighbor cell of "*i*". The metric terms to this generalized coordinate system are defined as:

$$h_x = S_x^l/V_l,\ h_y = S_y^l/V_l,\ h_z = S_z^l/V_l,\ h_n = S^l/V_l, \quad (7)$$

with S_x^l, S_y^l and S_z^l representing the surface area components at the "*l*" interface and S^l the norm of the surface area vector at the "*l*" interface.

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

$$\rho_l = \sqrt{\rho_L \rho_R}\ ,\ u_l = \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right),$$
$$v_l = \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (8)$$

$$w_l = \left(w_L + w_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right),$$
$$H_l = \left(H_L + H_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (9)$$
$$a_l = \sqrt{(\gamma-1)\left[H_l - 0.5\left(u_l^2 + v_l^2 + w_l^2\right)\right]}, \quad (10)$$

where a_l is the speed of the sound at the interface "*l*". The eigenvalues of the Euler equations, in the normal direction to the flux face, to the convective flux are given by:

$$q_{normal} = u_l h_x + v_l h_y + w_l h_z,\ \lambda_1 = q_{normal} - a_l h_n,$$
$$\lambda_2 = \lambda_3 = \lambda_4 = q_{normal},\ \lambda_5 = q_{normal} + a_l h_n. \quad (11)$$

The jumps of the conserved variables, necessary to the construction of the [3] dissipation function, are given by:

$$\Delta e = V_l(e_R - e_L),\ \Delta\rho = V_l(\rho_R - \rho_L); \quad (12)$$
$$\Delta(\rho u) = V_l\left[(\rho u)_R - (\rho u)_L\right], \Delta(\rho v) = V_l\left[(\rho v)_R - (\rho v)_L\right]; \quad (13)$$
$$\Delta(\rho w) = V_l\left[(\rho w)_R - (\rho w)_L\right]. \quad (14)$$

The α vectors to the "*l*" interface are calculated by the following expressions:

$$\{\alpha_l\} = \left[R^{-1}\right]_l \{\Delta_l \overline{Q}\}, \quad (15)$$

with:

$$\left[R^{-1}\right] = \begin{bmatrix} 0.5\left[\left((\gamma-1)/a_l^2\right)0.5q^2 + \varphi/a_l\right] & -0.5\left[\left((\gamma-1)/a_l^2\right)u_l + h_x'/a_l\right] \\ 1-\left((\gamma-1)/a_l^2\right)0.5q^2 & \left((\gamma-1)/a_l^2\right)u_l \\ -\left(h_y'u_l + h_z'v_l + h_x'w_l\right) & h_y' \\ -\left(h_z'u_l + h_x'v_l + h_y'w_l\right) & h_z' \\ 0.5\left[\left((\gamma-1)/a_l^2\right)0.5q^2 - \varphi/a_l\right] & 0.5\left[-\left((\gamma-1)/a_l^2\right)u_l + h_x'/a_l\right] \end{bmatrix}$$

$$\begin{bmatrix} -0.5\left[\left((\gamma-1)/a_l^2\right)v_l + h_y'/a_l\right] & -0.5\left[(\gamma-1)/a_l^2 w_l + h_z'/a_l\right] & 0.5(\gamma-1)/a_l^2 \\ \left((\gamma-1)/a_l^2\right)v_l & (\gamma-1)/a_l^2 w_l & -(\gamma-1)/a_l^2 \\ h_z' & h_x' & 0 \\ h_x' & h_y' & 0 \\ 0.5\left[-\left((\gamma-1)/a_l^2\right)v_l + h_y'/a_l\right] & 0.5\left[-(\gamma-1)/a_l^2 w_l + h_z'/a_l\right] & 0.5(\gamma-1)/a_l^2 \end{bmatrix}; \quad (16)$$

$$\{\Delta_l \overline{Q}\} = \{\Delta\rho \quad \Delta(\rho u) \quad \Delta(\rho v) \quad \Delta(\rho w) \quad \Delta e\}^T, \quad (17)$$

defined by Eqs. (12), (13) and (14);

$$q^2 = u_l^2 + v_l^2 + w_l^2; \quad (18)$$

$$\varphi = u_l h_x^{'} + v_l h_y^{'} + w_l h_z^{'}; \quad (19)$$

$$h_x^{'} = h_x / h_n \ , \ h_y^{'} = h_y / h_n \ \text{ and } \ h_z^{'} = h_z / h_n \ . \quad (20)$$

The [3] dissipation function uses the right-eigenvector matrix of the normal to the flux face Jacobian matrix in generalized coordinates:

$$[R] = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ u_l - h_x^{'} a_l & u_l & h_y^{'} & h_z^{'} & u_l + h_x^{'} a_l \\ v_l - h_y^{'} a_l & v_l & h_z^{'} & h_x^{'} & v_l + h_y^{'} a_l \\ w_l - h_z^{'} a_l & w_l & h_x^{'} & h_y^{'} & w_l + h_z^{'} a_l \\ H_l - h_x^{'} u_l a_l - h_y^{'} v_l a_l - h_z^{'} w_l a_l & 0.5q^2 & h_x^{'} w_l + h_z^{'} v_l + h_y^{'} u_l & h_y^{'} w_l + h_x^{'} v_l + h_z^{'} u_l & H_l + h_x^{'} u_l a_l + h_y^{'} v_l a_l + h_z^{'} w_l a_l \end{bmatrix}. \quad (21)$$

The entropy condition is implemented using the entropy function ψ defined of the following way:

$$\psi_m = \begin{cases} |\lambda_m|, & \text{if } |\lambda_m| \geq \varepsilon_m \\ 0.5\left(\lambda_m^2 + \varepsilon_m^2\right)/\varepsilon_m, & \text{if } |\lambda_m| < \varepsilon_m \end{cases} \text{ to non-linear}$$

fields and $\psi_m = |\lambda_m|$ to linear fields, (22)

with "*m*" ranging from 1 to 5 (three-dimensional space), being the values 1 and 5 associated to non-linear fields (shock waves and expansion waves) and 2, 3 and 4 associated to linear fields (contact discontinuities), and ε_m assuming the value 0.2, recommended by [3].

Finally, the [3] dissipation function to the "*l*" interface is constructed by the following matrix-vector product:

$$\{D_{Roe}\}_l = [R]_l \{-\psi\alpha\}_l \ . \quad (23)$$

The convective numerical flux vector to the "*l*" interface is described by:

$$F_l^{(m)} = \left(E_l^{(m)} h_x + F_l^{(m)} h_y + G_l^{(m)} h_z\right) V_l + 0.5 D_{Roe}^{(m)}, \quad (24)$$

where:

$$E_l^{(m)} = 0.5\left(E_R^{(m)} + E_L^{(m)}\right) \ , \quad F_l^{(m)} = 0.5\left(F_R^{(m)} + F_L^{(m)}\right)$$

$$\text{and } G_l^{(m)} = 0.5\left(G_R^{(m)} + G_L^{(m)}\right). \quad (25)$$

The time integration is performed by an explicit method, second order accurate, Runge-Kutta type of five stages and can be represented of generalized form by:

$$\begin{aligned} Q_i^{(0)} &= Q_i^{(n)} \\ Q_i^{(k)} &= Q_i^{(0)} - \alpha_k \, \Delta t_i / V_i \times C\left(Q_i^{(k-1)}\right), \\ Q_i^{(n+1)} &= Q_i^{(k)} \end{aligned} \quad (26)$$

with "*k*" = 1,...,5; $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, $\alpha_3 = 3/8$, $\alpha_4 = 1/2$ and $\alpha_5 = 1$. The contribution of the convective numerical flux vectors is determined by the C_i vector:

$$C_i^{(m)} = F_1^{(m)} + F_2^{(m)} + F_3^{(m)} + F_4^{(m)} \ . \quad (27)$$

5 Explicit Numerical Algorithm of [4] – Flux Vector Splitting Scheme

5.1 Theory for the one-dimensional case

If the homogeneous Euler equations are put in characteristic form

$$\partial W / \partial t + \Lambda \, \partial W / \partial x = 0 \, , \quad (28)$$

where *W* is the vector of characteristic variables (defined in [16]) and Λ is the diagonal matrix of eigenvalues, the upwind scheme:

$$u_i^{n+1} - u_i^n = -\Delta t / \Delta x \left[\hat{a}^+ \left(u_i^n - u_{i-1}^n\right) + \hat{a}^- \left(u_{i+1}^n - u_i^n\right)\right], \quad (29)$$

where *u* is a scalar property, $\hat{a}^+ = 0.5\left(\hat{a} + |\hat{a}|\right)$ and $\hat{a}^- = 0.5\left(\hat{a} - |\hat{a}|\right)$, can be applied to each of the three characteristic variables separately, with the definitions

$$\lambda_m^+ = 0.5\left(\lambda_m + |\lambda_m|\right) \quad \text{and} \quad \lambda_m^- = 0.5\left(\lambda_m - |\lambda_m|\right) \quad (30)$$

for each of the eigenvalues of Λ

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} = \begin{bmatrix} u & & \\ & u + a & \\ & & u - a \end{bmatrix}. \quad (31)$$

This defines two diagonal matrices $\Lambda^{\pm}$:

$$\Lambda^{\pm} = \begin{bmatrix} \lambda_1^{\pm} & & \\ & \lambda_2^{\pm} & \\ & & \lambda_3^{\pm} \end{bmatrix} = \begin{bmatrix} 0.5\left(u \pm |u|\right) & & \\ & 0.5\left(u + a \pm |u + a|\right) & \\ & & 0.5\left(u - a \pm |u - a|\right) \end{bmatrix}, \quad (32)$$

where Λ^+ has only positive eigenvalues, Λ^- only negative eigenvalues, and such that

$$\Lambda = \Lambda^+ + \Lambda^- \quad \text{and} \quad |\Lambda| = \Lambda^+ - \Lambda^- \quad \text{or}$$

$$\lambda_m = \lambda_m^+ + \lambda_m^- \quad \text{and} \quad |\lambda_m| = \lambda_m^+ - \lambda_m^- \, . \quad (33)$$

The quasi-linear coupled equations are obtained from the characteristic form by the transformation

matrix P (defined in [16]), with the Jacobian A satisfying

$$A = P\Lambda P^{-1}, \text{ resulting in } \partial Q/\partial t + A\partial Q/\partial x = 0. \quad (34)$$

Hence an upwind formulation can be obtained with the Jacobians

$$A^+ = P\Lambda^+ P^{-1} \quad \text{and} \quad A^- = P\Lambda^- P^{-1}, \text{ with:}$$

$$A = A^+ + A^- \text{ and } |A| = A^+ - A^-. \quad (35)$$

The fluxes associated with these split Jacobians are obtained from the remarkable property of homogeneity of the flux vector *f(Q)*. *f(Q)* is a homogeneous function of degree one of *Q*. Hence, $f = AQ$ and the following flux splitting can be defined:

$$f^+ = A^+Q \quad \text{and} \quad f^- = A^-Q, \text{ with: } f = f^+ + f^-. \quad (36)$$

This flux vector splitting, based on Eq. (30), has been introduced by [4]. The split fluxes f^+ and f^- are also homogeneous functions of degree one in *Q*.

5.2 Arbitrary meshes

In practical computations one deal mostly with arbitrary meshes, considering either in a finite volume approach or in a curvilinear coordinate system. In both cases, the upwind characterization is based on the signs of the eigenvalues of the matrix

$$K^{(n)} = \vec{A} \bullet \vec{n} = An_x + Bn_y + Cn_z, \quad (37)$$

where *A*, *B* and *C* are the Jacobian matrices written to the Cartesian system. The fluxes will be decomposed by their components

$$\tilde{F}^{(n)} = \vec{\tilde{F}} \bullet \vec{n} = En_x + Fn_y + Gn_z \quad (38)$$

and separated into positive and negative parts according to the sign of the eigenvalues of $K^{(n)}$ as described above, considering the normal direction as a local coordinate direction.

For a general eigenvalue splitting, as Eq. (30), the normal flux projection, Eq. (38), is decomposed by a [4] flux splitting as

$$\tilde{F}_{\pm}^{(n)} = \frac{\rho}{2\gamma}\begin{Bmatrix} \alpha \\ \alpha u + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_x \\ \alpha v + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_y \\ \alpha w + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_z \\ \alpha\dfrac{u^2+v^2+w^2}{2} + av_n\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right) + a^2\dfrac{\lambda_2^{\pm} + \lambda_3^{\pm}}{\gamma - 1} \end{Bmatrix}, \quad (39)$$

where the eigenvalues of the matrix *K* are defined as

$$\lambda_1 = \vec{v} \bullet \vec{n} \equiv v_n, \ \lambda_2 = \vec{v} \bullet \vec{n} + a \text{ and } \lambda_3 = \vec{v} \bullet \vec{n} - a, \quad (40)$$

with $\vec{v}$ being the flow velocity vector, ± sign indicates the positive or negative parts respectively, and the speed of sound defined by $a = \sqrt{\gamma p/\rho}$. The parameter α is defined as

$$\alpha = 2(\gamma - 1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}. \quad (41)$$

5.3 Numerical scheme

The numerical scheme of [4] implemented in this work is based on an unstructured finite volume formulation, where the convective numerical fluxes at interface are calculated as

$$\tilde{F}_l^{(m)} = \left[\left(\tilde{F}_R^-\right)^{(m)} + \left(\tilde{F}_L^+\right)^{(m)}\right] S^l. \quad (42)$$

The Runge-Kutta explicit method, second order accurate in time, described by Eq. (26), is employed to perform the time integration. The contribution of the convective numerical flux vectors is determined by the C_i vector:

$$C_i^{(m)} = \tilde{F}_1^{(m)} + \tilde{F}_2^{(m)} + \tilde{F}_3^{(m)} + \tilde{F}_4^{(m)}. \quad (43)$$

This version of the flux vector splitting algorithm of [4] is first order accurate in space.

6 Explicit Numerical Algorithm of [5] – Flux Vector Splitting Scheme

The approximation of the integral equation (1) to a tetrahedron finite volume yields a system of ordinary differential equations with respect to time:

$$V_i \, dQ_i/dt = -C_i, \quad (44)$$

with C_i representing the net flux (residual) of conservation of mass, of linear momentum and of energy in the V_i volume. The residual is calculated as:

$$C_i^{(m)} = F_1^{(m)} + F_2^{(m)} + F_3^{(m)} + F_4^{(m)}, \quad (45)$$

with $F_1^{(m)} = F_1^{e(m)}$, being "*e*" related to the flow convective contribution at the $l = 1$ interface.

As shown in [8], the discrete convective flux calculated by the AUSM scheme ("Advection Upstream Splitting Method") can be interpreted as a sum involving the arithmetical average between the right (*R*) and the left (*L*) states of the "*l*" cell face, related to cell *i* and its neighbor, respectively, multiplied by the interface Mach number, and a scalar dissipative term. Hence, to the "*l*" interface:

$$F_l = |S|_l \left(\frac{1}{2} M_l \left(\begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_L + \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_R \right) - \frac{1}{2}\phi_l \left(\begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_R - \begin{bmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a w \\ \rho a H \end{bmatrix}_L \right) \right) + \begin{bmatrix} 0 \\ S_x p \\ S_y p \\ S_z p \\ 0 \end{bmatrix}_l, \quad (46)$$

where $S_l = \begin{bmatrix} S_x & S_y & S_z \end{bmatrix}_l^T$ defines the normal area vector to the "*l*" surface. M_l defines the advection Mach number at the "*l*" face of the "*i*" cell, which is calculated according to [8] as:

$$M_l = M_L^+ + M_R^-, \tag{47}$$

where the separated Mach numbers $M^{+/-}$ are defined by [5]:

$$M^+ = \begin{cases} M, & if\ M \geq 1; \\ 0.25(M+1)^2, & if\ |M| < 1; \\ 0, & if\ M \leq -1; \end{cases} \quad \text{and}$$

$$M^- = \begin{cases} 0, & if\ M \geq 1; \\ -0.25(M-1)^2, & if\ |M| < 1; \\ M, & if\ M \leq -1. \end{cases} \tag{48}$$

M_L and M_R represent the Mach number associated with the left and right states, respectively. The advection Mach number is defined by:

$$M = \left(S_x u + S_y v + S_z w\right) / \left(|S| a\right). \tag{49}$$

The pressure at the "*l*" face of the "*i*" cell is calculated by a similar way:

$$p_l = p_L^+ + p_R^-, \tag{50}$$

with $p^{+/-}$ denoting the pressure separation defined according to [5]:

$$p^+ = \begin{cases} p, & if\ M \geq 1; \\ 0.25p(M+1)^2(2-M), & if\ |M| < 1 \\ 0, & if\ M \leq -1; \end{cases} \quad \text{and}$$

$$p^- = \begin{cases} 0, & if\ M \geq 1; \\ 0.25p(M-1)^2(2+M), & if\ |M| < 1; \\ p, & if\ M \leq -1. \end{cases} \tag{51}$$

The definition of the dissipative term ϕ determines the particular formulation of the convective fluxes. According to [9], the choice below corresponds to the [5] scheme:

$$\varphi_l = \varphi_l^{VL} = \begin{cases} |M_l|, & if\ |M_l| \geq 1; \\ |M_l| + 0.5(M_R - 1)^2, & if\ 0 \leq M_l < 1; \\ |M_l| + 0.5(M_L + 1)^2, & if\ -1 < M_l \leq 0. \end{cases} \tag{52}$$

The equations above clearly show that to a supersonic Mach number at the cell face, the [5] scheme represents a purely upwind discretization, using either the left state or the right state to the convective and pressure terms, depending of the Mach number signal. The time integration follows the Runge-Kutta method described in the [3] scheme (Eq. 26). The [5] scheme presented in this work is first order accurate in space.

7 Explicit Numerical Algorithm of [6] – Flux Difference Splitting Scheme

The [6] algorithm, first order accurate in space, is specified by the determination of the numerical flux vector at "*l*" interface. This scheme uses Equations (6) to (21) of [3] scheme, also using the [3] average to determine the interface properties. The next step consists in determining the entropy condition. The entropy condition is implemented of the following way:

$$\upsilon_m = \Delta t_i \lambda_m = Z_m \quad \text{and}$$

$$\psi_m = \begin{cases} |Z_m|, & if\ |Z_m| \geq \delta_f \\ 0.5\left(Z_m^2 + \delta_f^2\right)/\delta_f, & if\ |Z_m| < \delta_f \end{cases}, \tag{53}$$

with "*m*" varying from 1 to 5 (three-dimensional space), as defined in the [3] scheme, and δ_f assuming values between 0.1 and 0.5, being 0.2 the value suggested by [6].

The [6] dissipation function to the "*l*" interface is constructed by the following matrix-vector product:

$$\{D_{Harten}\}_l = [R]_l \{-\psi\alpha/\Delta t_i\}_l. \tag{54}$$

The convective numerical flux vector to the "*l*" interface is described by:

$$F_l^{(m)} = \left(E_l^{(m)} h_x + F_l^{(m)} h_y + G_l^{(m)} h_z\right) V_l + 0.5 D_{Harten}^{(m)}, \tag{55}$$

with $E_l^{(m)}$, $F_l^{(m)}$ and $G_l^{(m)}$ defined according to Eq. (25). The time integration is performed by the Runge-Kutta explicit method, second order accurate, of five stages, described in Eq. (26). The contribution of the convective numerical flux vectors is determined by the C_i vector:

$$C_i^{(m)} = F_1^{(m)} + F_2^{(m)} + F_3^{(m)} + F_4^{(m)}. \tag{56}$$

8 Explicit Numerical Algorithm of [7] – Flux Difference Splitting Scheme

In this scheme, the numerical flux vector is calculated applying the flux difference splitting procedure of [3]. The flux which crosses each "*l*" cell face is calculated using the [3] formula:

$$F_l = 1/2\left[F(Q_L) + F(Q_R) - \left|\tilde{A}\right|(Q_R - Q_L)\right]_l. \tag{57}$$

In this equation Q_R and Q_L are right and left state variables of the "*l*" flux interface, respectively. The Roe matrix $\tilde{A}$ is determined by the evaluation of $A = \partial F/\partial Q$ with the flow properties obtained by the [3] average [Eqs. (8), (9) and (10)] of such way that $F(Q_R) - F(Q_L) = \tilde{A}(Q_R - Q_L)$ is exactly satisfied. Introducing the diagonalization matrices $[R]$ and $[R^{-1}]$ evaluated with the [3] average, defined by Eqs. (21) and (16), respectively, and the

eigenvalue diagonal matrix Λ, the $\left|\tilde{A}\right|$ matrix is defined as $\left|\tilde{A}\right| = [R]\Lambda\left[R^{-1}\right]$. The term

$$\left|\tilde{A}\right|(Q_R - Q_L) = [R]\Lambda\left[R^{-1}\right]\Delta Q \quad (58)$$

in the formula of the numerical flux vector of [3], can be rewritten in terms of three flux components, each one associated with a distinct eigenvalue, and the dissipation function of the [7] scheme is defined by:

$$D_{FPP} = [R]\Lambda\left[R^{-1}\right]\Delta Q = \left|\Delta F_1\right| + \left|\Delta F_4\right| + \left|\Delta F_5\right|, \quad (59)$$

where:

$$\left|\Delta F_1\right| = \left|\psi_1\right|\left\{\left(\Delta\rho - \frac{\Delta p}{a_l^2}\right)\begin{bmatrix}1\\ u_l\\ v_l\\ w_l\\ \frac{u_l^2+v_l^2+w_l^2}{2}\end{bmatrix} + \rho_l\begin{bmatrix}0\\ \Delta u - n_x\Delta U_l\\ \Delta v - n_y\Delta U_l\\ \Delta w - n_z\Delta U_l\\ u_l\Delta u + v_l\Delta v + w_l\Delta w - U_l\Delta U_l\end{bmatrix}\right\}; \quad (60)$$

$$\left|\Delta F_{4,5}\right| = \left|\psi_{4,5}\right|\left(\frac{\Delta p \pm \rho_l a_l \Delta U_l}{2a_l^2}\right)\begin{bmatrix}1\\ u_l \pm n_x a_l\\ v_l \pm n_y a_l\\ w_l \pm n_z a_l\\ H_l \pm U_l a_l\end{bmatrix}, \quad (61)$$

with $U_l = u_l n_x + v_l n_y + w_l n_z$, $\Delta U_l = n_x\Delta u + n_y\Delta v + n_z\Delta w$ and $\Delta(\cdot) = (\cdot)_{ne} - (\cdot)_i$.

The present author introduced the entropy function ψ aiming to avoid zero values to the contributions of the system eigenvalues to the dissipation function. This entropy condition is implemented in the eigenvalues $\lambda_1 = U_l$, $\lambda_4 = U_l + a_l$ and $\lambda_5 = U_l - a_l$ as follows:

$$\psi_l = \begin{cases}\left|Z_l\right|, & \text{if } \left|Z_l\right| \geq \varepsilon\\ 0.5\left(Z_l^2 + \varepsilon^2\right)/\varepsilon, & \text{if } \left|Z_l\right| < \varepsilon\end{cases}, \text{ with: } Z_l = \lambda_l, \quad (62)$$

where the ε parameter assumes the value 0.01, recommended by the present author. In the original work of [7], the value used to ε is equal to zero, which corresponds to the non-use of the entropy condition.

The numerical flux vector at the "*l*" interface is determined by:

$$F_l^{(m)} = \left(E_l^{(m)} n_x + F_l^{(m)} n_y + G_l^{(m)} n_z - 0.5 D_{FPP}^{(m)}\right)S^l, \quad (63)$$

with $E_l^{(m)}$, $F_l^{(m)}$ and $G_l^{(m)}$ defined according to Eq. (25). The time integration is performed by the Runge-Kutta explicit method, second order accurate, of five stages, described in Eq. (26). The contribution of the convective numerical flux vectors is determined by the C_i vector:

$$C_i^{(m)} = F_1^{(m)} + F_2^{(m)} + F_3^{(m)} + F_4^{(m)}. \quad (64)$$

The [7] scheme implemented in this work is first order accurate in space.

9 Explicit Numerical Algorithm of [8] – Flux Vector Splitting Scheme

The algorithm of [8] is specified by the determination of the numerical flux vector at the "*l*" interface. This scheme employs Equations (44) to (51) of the [5] scheme to determine interface properties. The next step consists in determining the dissipative term ϕ. The definition of the this term determines the particular formulation of the convective fluxes. According to [9], the choice below corresponds to the [8] scheme:

$$\phi_l = \phi_l^{LS}, \text{ with: } \phi_l^{LS} = \left|M_l\right|. \quad (65)$$

The time integration employs the Runge-Kutta method described by Eq. (26). The [8] scheme presented in this work is first order accurate in space.

10 Explicit Numerical Algorithm of [9] – Flux Vector Splitting Scheme

The algorithm of [9] is specified by the determination of the numerical flux vector at the "*l*" interface. This scheme employs Equations (44) to (51) of the [5] scheme to determine interface properties. The next step consists in determining the dissipative term ϕ. The definition of the this term determines the particular formulation of the convective fluxes. A hybrid scheme is proposed by [9], which combines the [5] scheme and the [8] (AUSM) scheme. Hence,

$$\varphi_l = (1-\omega)\varphi_l^{VL} + \omega\varphi_l^{LS} \quad (66)$$

with:

$$\varphi_l^{VL} = \begin{cases}\left|M_l\right|, & \text{if } \left|M_l\right| \geq 1;\\ \left|M_l\right| + \frac{1}{2}(M_R - 1)^2, & \text{if } 0 \leq M_l < 1;\\ \left|M_l\right| + \frac{1}{2}(M_L + 1)^2, & \text{if } -1 < M_l \leq 0;\end{cases} \quad (67)$$

$$\varphi_l^{LS} = \begin{cases}\left|M_l\right|, & \text{if } \left|M_l\right| \geq \tilde{\delta}\\ \frac{(M_l)^2 + \tilde{\delta}^2}{2\tilde{\delta}}, & \text{if } \left|M_l\right| < \tilde{\delta}\end{cases}, \quad (68)$$

where $\tilde{\delta}$ is a small parameter, $0 < \tilde{\delta} \leq 0.5$, and ω is a constant, $0 \leq \omega \leq 1$. In this work, the values used to $\tilde{\delta}$ and ω were: 0.2 and 0.5, respectively. The

time integration follows the Runge-Kutta method of five stages, second order accurate, described by Eq. (26). This scheme is first order accurate in space.

11 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:

$$\Delta t_i = \frac{CFL(\Delta s)_i}{(|q| + a)_i}, \quad (69)$$

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

$$(\Delta s)_i = [MIN(l_{MIN}, C_{MIN})]_i , \quad (70)$$

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 $(|q| + a)_i$, with $q = \sqrt{u^2 + v^2 + w^2}$.

12 Initial and Boundary Conditions

12.1 Initial condition

The initial condition adopted for the problems is the freestream flow in all calculation domain ([12] and [17]). The vector of conserved variables is expressed as follows:

$$Q_\infty = \begin{Bmatrix} 1 \\ M_\infty \cos\theta \\ M_\infty \sin\theta \cos\psi \\ M_\infty \sin\theta \sin\psi \\ \left[\frac{1}{\gamma(\gamma-1)} + \frac{M_\infty^2}{2}\right] \end{Bmatrix}, \quad (71)$$

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.

12.2 Boundary conditions

The different types of implemented boundary conditions are described as follows. They are implemented in the "ghost" cells.

a) Wall - 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 neighbor 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 neighbor real cell. [18] suggests that these procedures lead to the following expressions to the velocity components *u*, *v* and *w* of the ghost cells:

$$u_g = (1 - 2n_x n_x)u_{real} + (-2n_x n_y)v_{real} + (-2n_x n_z)w_{real}; \quad (72)$$

$$v_g = (-2n_y n_x)u_{real} + (1 - 2n_y n_y)v_{real} + (-2n_y n_z)w_{real}; \quad (73)$$

$$w_g = (-2n_z n_x)u_{real} + (-2n_z n_y)v_{real} + (1 - 2n_z n_z)w_{real}. \quad (74)$$

The fluid pressure gradient in the direction normal to the wall is equal to zero for the inviscid case. The temperature gradient is equal to zero along the whole wall, according to the condition of adiabatic wall. 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, Eq. (3).

b) Far field - In the implementation of the boundary conditions in the mesh limit external region to the ramp problem (external flow), it is necessary to identify four possible situations: entrance with subsonic flow, entrance with supersonic flow, exit with subsonic flow and exit with supersonic flow. These situations are described below.

b.1) Entrance with subsonic flow – Considering the one-dimensional characteristic relation concept in the normal direction of flow penetration, the entrance with subsonic flow presents four characteristic velocities of information propagation which have direction and orientation point inward the calculation domain, which implies that the variables associated with these waves cannot be extrapolated ([17]). It is necessary to specify four conditions to these four information. [12] indicate as appropriated quantities to be specified the freestream density and the freestream Cartesian velocity components *u*, *v* and *w*. Just the last characteristics,"*(q_n-a)*",which transports information from inside to outside of the calculation domain, cannot be specified and will have to be determined by interior information of the calculation domain. In this work, a zero order extrapolation to the pressure is performed, being the total energy defined by the state equation of a perfect gas.

b.2) Entrance with supersonic flow - All variables are specified at the entrance boundary, adopting freestream values.
b.3) Exit with subsonic flow - Four characteristics which govern the Euler equations proceed from the internal region of the calculation domain. So, the density and the Cartesian velocity components are extrapolated from the interior domain ([17]). One condition should be specified to the boundary. In this case, the pressure is fixed in the calculation domain exit, keeping its respective value of freestream flow. Total energy is specified by the equation of state to a perfect gas.
b.4) Exit with supersonic flow - The five characteristics which govern the Euler equations proceed from the internal region of the calculation domain. It is not possible to specify variable values at the exit. The zero order extrapolation is applied to density, Cartesian velocity components and pressure. Total energy is specified by the equation of state to a perfect gas.
c) Entrance and exit – The entrance and exit boundaries are applied to the ramp and diffuser problems. Boundary conditions which involve flow entrance in the calculation domain had the flow properties fixed with freestream values. Boundary conditions which involve flow exit of the computational domain used simply the zero order extrapolation to the determination of properties in this boundary. This procedure is correct because the entrance flow and the exit flow are no minimal supersonic to both studied examples.

13 Results

Tests were performed in a microcomputer with processor AMD SEMPRON (tm) 2600+, 1.83GHz, and 512 Mbytes of RAM memory. As the interest of this work is steady state problems, one needs to define a criterion which guarantees that such condition was reached. The criterion adopted in this work was to consider a reduction of 4 orders in the magnitude of the maximum residual in the domain, a typical criterion in the CFD community. The residual to each cell was defined as the numerical value obtained from the discretized conservation equations. As there are five conservation equations to each cell, the maximum value obtained from these equations is defined as the residual of this cell. Thus, this residual is compared with the residual of the others cells, calculated of the same way, to define the maximum residual in the domain. The configuration upstream and the configuration longitudinal plane angles were set equal to 0.0°.

The physical problems to be studied are the supersonic flow along a ramp with 20° of inclination and the “cold gas” hypersonic flow along a diffuser also with 20° of inclination at the contraction region. The ramp and diffuser configurations in the xy plane are described in Figs. 7 and 8. The ramp spanwise length is 0.25m, while the diffuser spanwise length is 0.10m.

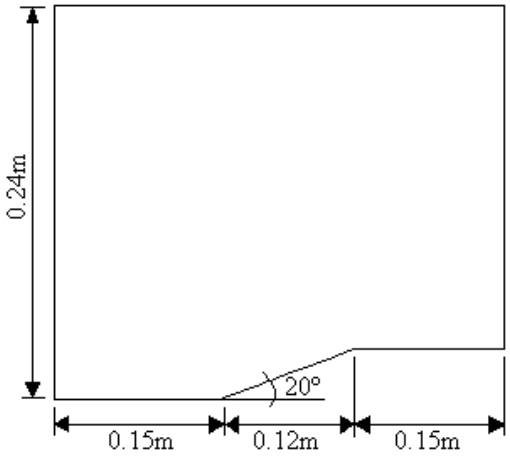


Figure 7 : Ramp configuration in the xy plane.

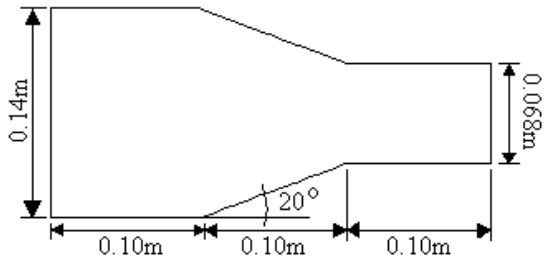


Figure 8 : Diffuser configuration in the xy plane.

In the ramp problem, an oblique shock wave and an expansion fan are formed, at the ramp and after the ramp, respectively. In the diffuser problem, the two oblique shock waves originated by the convergent walls suffer interference after the throat and expansions waves are formed after the contraction region in both upper and lower walls.

The meshes employed in this work were generated on a structured context, obtaining cells of hexahedra, and through the generation of the connectivity, neighboring and node-coordinate tables such meshes were transformed in meshes of tetrahedra. Although this procedure of mesh generation does not produce meshes with the best spatial discretization, meshes with reasonable quality have been obtained for the present problems. The ramp and diffuser meshes were generated in the

xy plane and projected in the z direction, in x'y' planes parallel to the original xy plane. The computational data of the generated meshes are presented in Tab. 1.

Table 1 : Computational data of the ramp and diffuser meshes.

Data	Ramp	Diffuser
Finite Difference Representation	61(ξ)x50(η)x 10(ζ)	61(ξ)x41(η)x 10(ζ)
Cells (Finite Volumes)	158,760	129,600
Nodes (Finite Volumes)	30,500	25,010

13.1 Ramp physical problem

The freestream Mach number adopted as initial condition to this simulation was 4.0, characterizing a supersonic flow regime

Figures 9 to 15 show the density contours obtained by the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes, respectively. All solutions present good quality characteristics with the shock being well captured by all schemes. The [8] scheme presents the densest field in comparison with the others schemes.

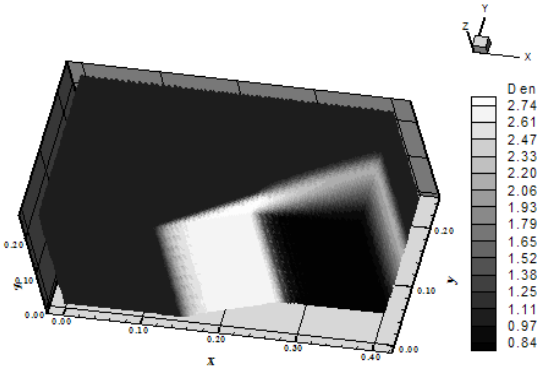


Figure 9 : Density contours (R).

Figures 16 to 22 exhibit the pressure contours obtained by the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes, respectively. The most severe pressure field was obtained by the [8] scheme. Good qualitative characteristics are observed in all solutions with the oblique shock wave appropriately captured.

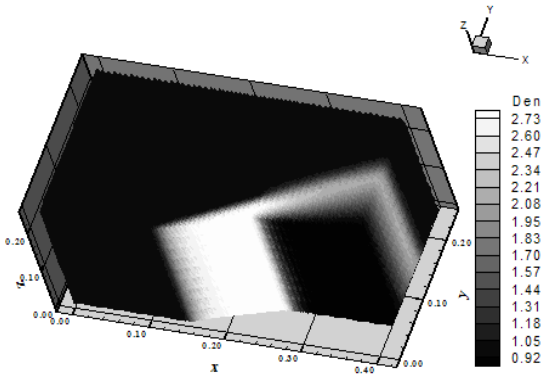


Figure 10 : Density contours (SW).

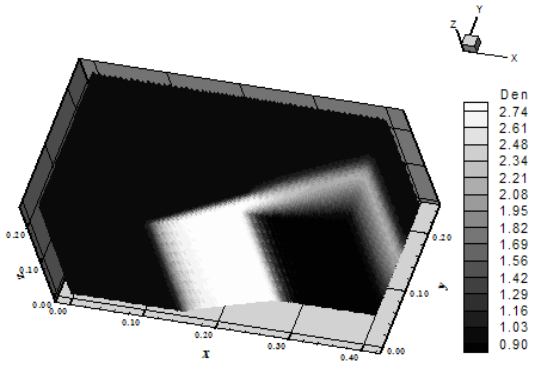


Figure 11 : Density contours (VL).

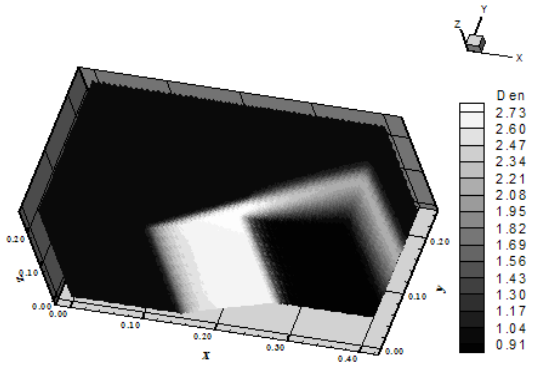


Figure 12 : Density contours (H).

Figures 23 to 29 show the Mach number contours obtained by the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes, respectively. The [5] scheme presents the most intense Mach number field in comparison with the others schemes.

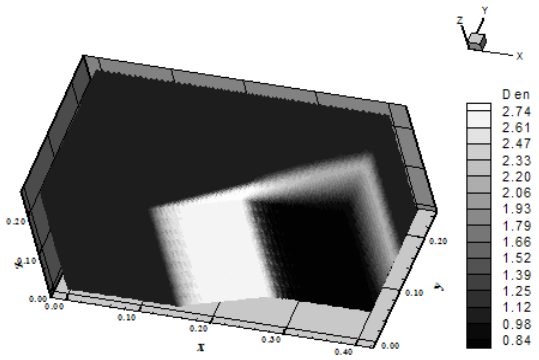


Figure 13 : Density contours (FPP).

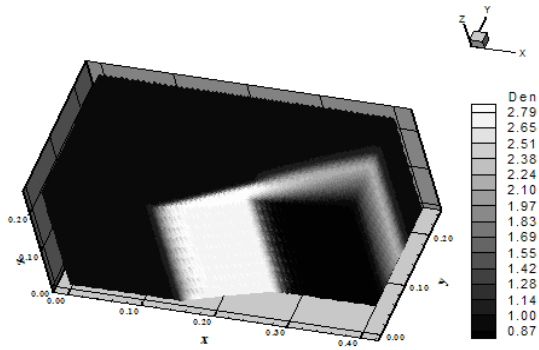


Figure 14 : Density contours (LS).

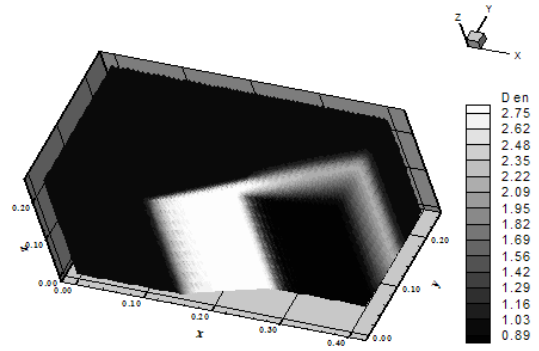


Figure 15 : Density contours (RK).

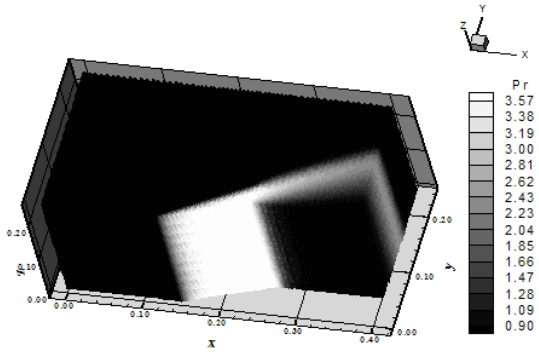


Figure 16 : Pressure contours (R).

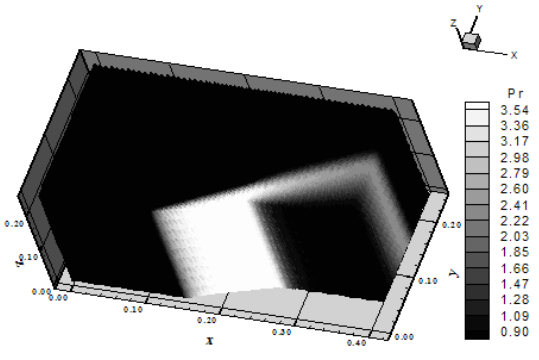


Figure 17 : Pressure contours (SW).

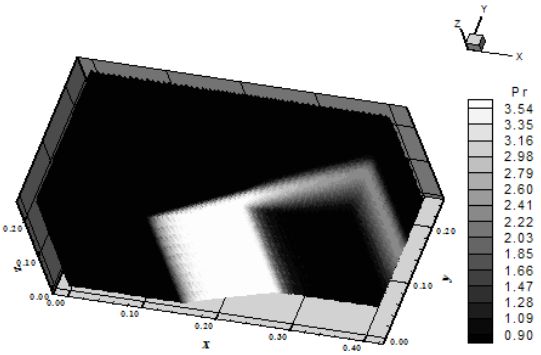


Figure 18 : Pressure contours (VL).

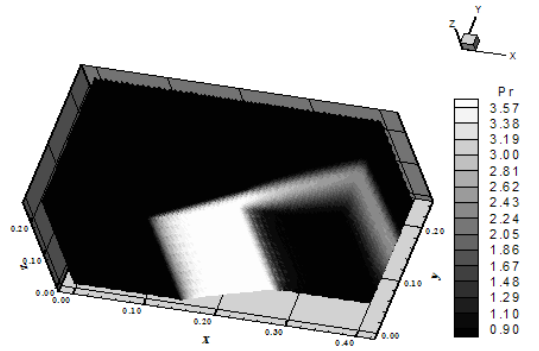


Figure 19 : Pressure contours (H).

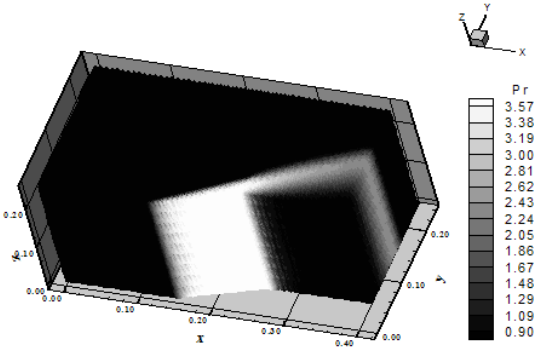


Figure 20 : Pressure contours (FPP).

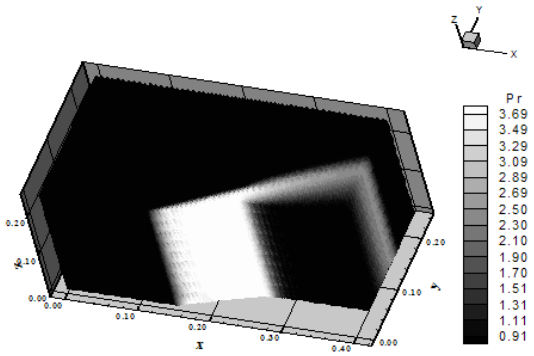


Figure 21 : Pressure contours (LS).

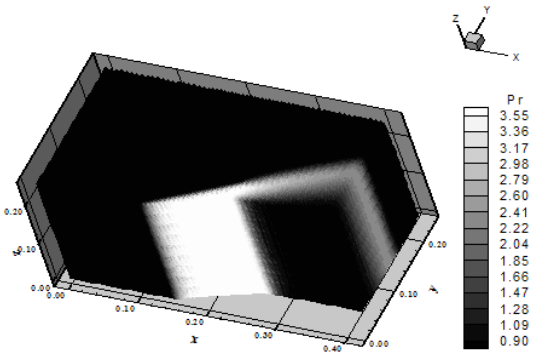


Figure 22 : Pressure contours (RK).

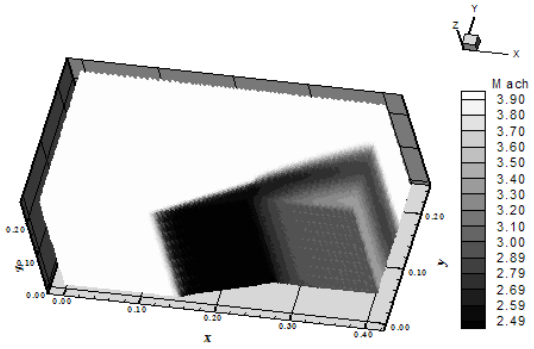


Figure 23 : Mach contours (R).

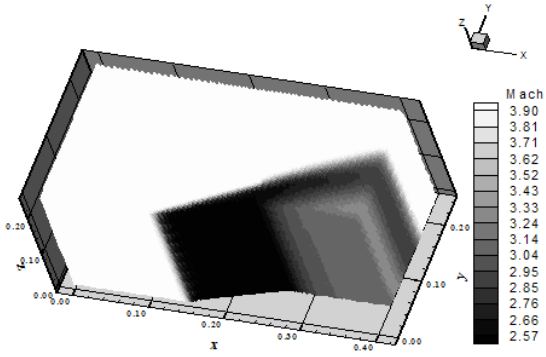


Figure 24 : Mach contours (SW).

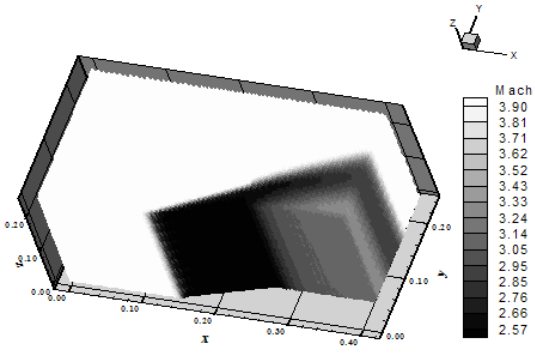


Figure 25 : Mach contours (VL).

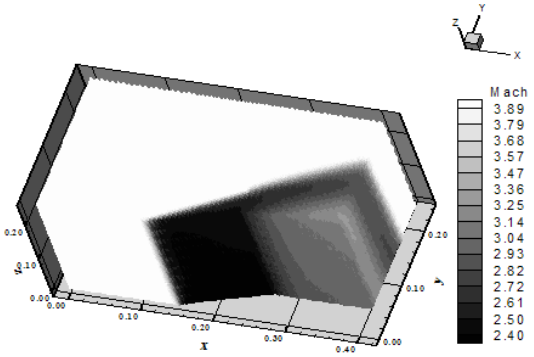


Figure 26 : Mach contours (H).

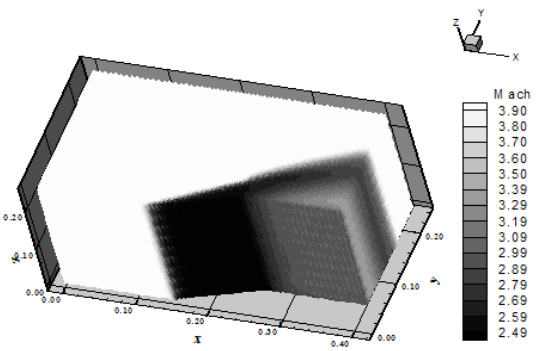


Figure 27 : Mach contours (FPP).

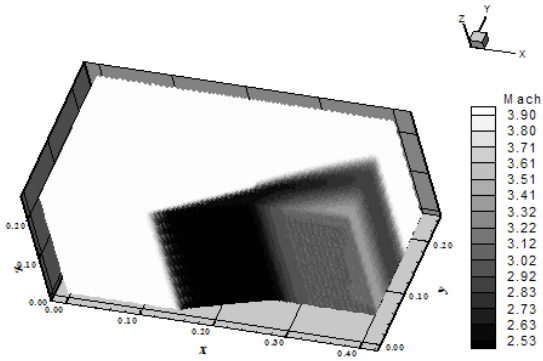


Figure 28 : Mach contours (LS).

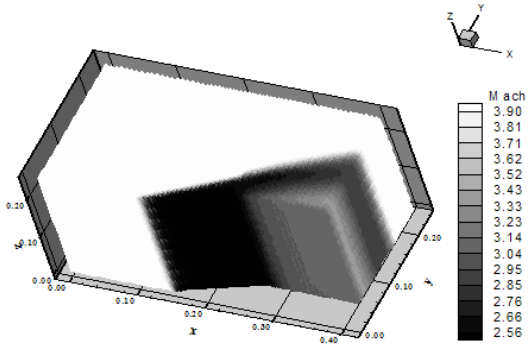


Figure 29 – Mach contours (RK).

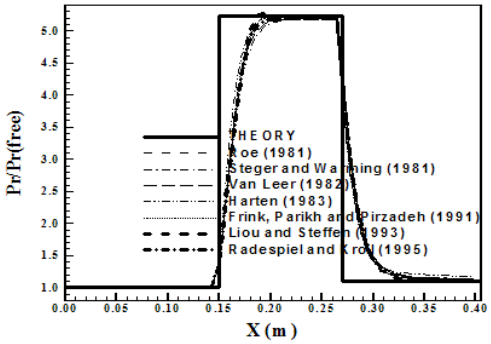


Figure 30 : Wall pressure distributions.

Figure 30 presents the wall pressure distributions obtained with the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes, evaluated at the computational plane k = 1, where "k" is the index of points at the z direction. They are compared with exact solutions of the oblique shock wave and the Prandtl-Meyer expansion wave theories. With the exception of the [8] scheme, which presents an overshoot at the pressure plateau at the ramp, all others schemes represent accurately the pressure plateau at the ramp, agreeing with the oblique shock wave theory. However, the best prediction of the shock pressure and width of the pressure plateau is determined by the [3] scheme, as expected due to this scheme resolves exactly the jump discontinuities as the Roe average is used. All schemes detect appropriately the pressure at the end of the expansion fan, after the ramp, with the exception of the [6] scheme, with little increase in relation to the others solutions.

One way to quantitatively verify if the solutions generated by each scheme are satisfactory consists in determining the shock angle of the oblique shock wave, β, measured in relation to the initial direction of the flow field. [19] (pages 352 and 353) presents a diagram with values of the shock angle, β, to oblique shock waves. The value of this angle is determined as function of the freestream Mach number and of the deflection angle of the flow after the shock wave, ϕ. To ϕ = 20º (ramp inclination angle) and to a freestream Mach number equals to 4.0, it is possible to obtain from this diagram a value to β equals to 32.5º. Using a transfer in Figures 16 to 22, 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.

Table 2 : Shock angle of the oblique shock wave at the ramp and percentage error to each scheme.

Algorithm	β (°)	Error (%)
[3]	32.2	0.923
[4]	32.8	0.923
[5]	32.7	0.615
[6]	35.2	8.308
[7]	32.3	0.615
[8]	32.0	1.538
[9]	32.2	0.923

Five of the seven schemes predicted accurately the shock angle of the oblique shock wave, with errors less than 1.0%. Only, the [6], the worst, and the [8] schemes presented errors above 1.0%. The best results were obtained by the [5] and the [7] schemes.

13.2 Diffuser physical problem

The freestream Mach number adopted as initial condition to this simulation was 10.0, characterizing a "cold gas" hypersonic flow regime. The [3] and the [7] schemes were not so robust to simulate this more severe physical problem. Results obtained with these schemes to a freestream Mach number equals to 4.0 (supersonic flow) are presented, but the author major interest is in the hypersonic case

Figures 31 to 35 exhibit the density contours obtained by the [4], the [5], the [6], the [8] and the [9] schemes, respectively, to a freestream Mach number equals to 10.0 . All solutions present good quality characteristics with the shock interference being well captured by all schemes. The most appropriated symmetry and homogeneity properties are observed in the [8] scheme. The [8] scheme presents the densest field in comparison with the others schemes.

Figures 36 and 37 show the density contours obtained with the [3] and the [7] schemes, respectively, to a freestream Mach number of 4.0. It is possible to observe shock reflections at the upper and lower walls close to the exit boundary, after 0.15m. It originates a pressure peak at the wall pressure distribution in both solutions. Good qualitative features are observed in both solutions. The [7] presents denser field than the [3] scheme.

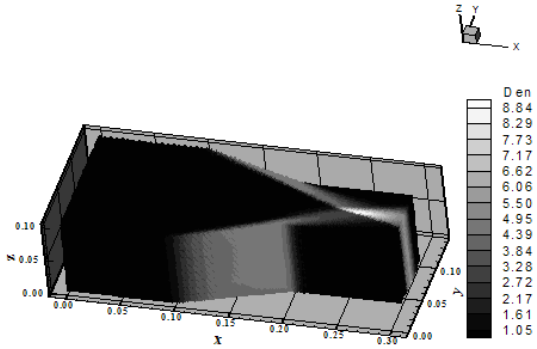


Figure 31 : Density contours (SW).

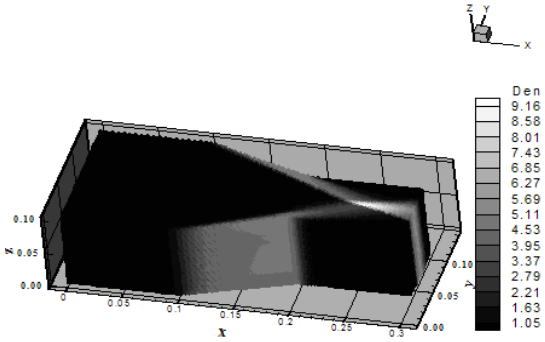


Figure 32 : Density contours (VL).

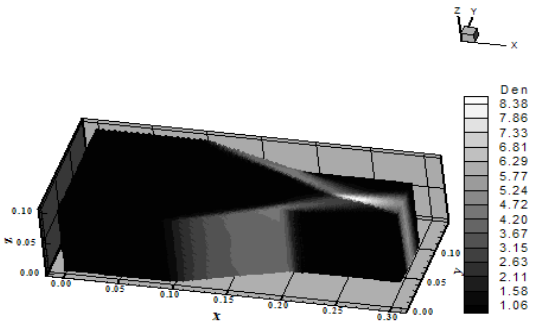


Figure 33 : Density contours (H).

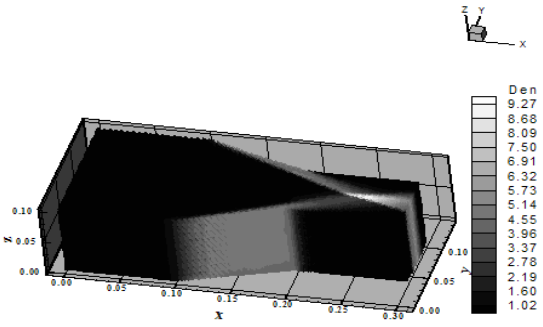


Figure 34 : Density contours (LS).

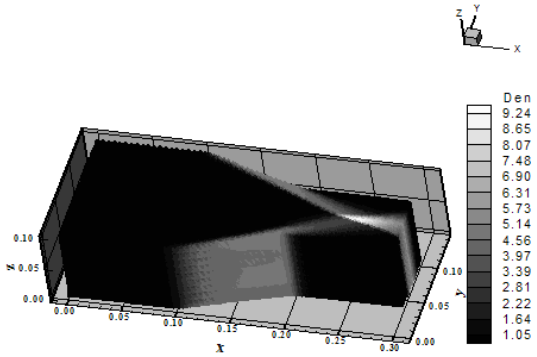


Figure 35 : Density contours (RK).

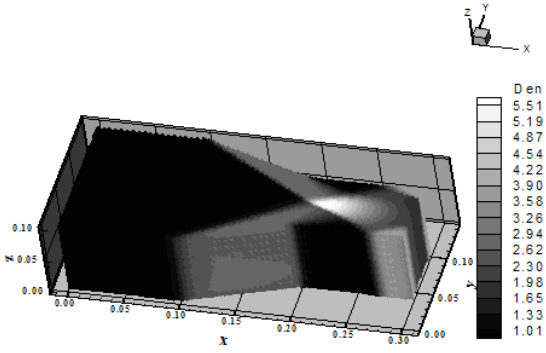


Figure 36 : Density contours (R).

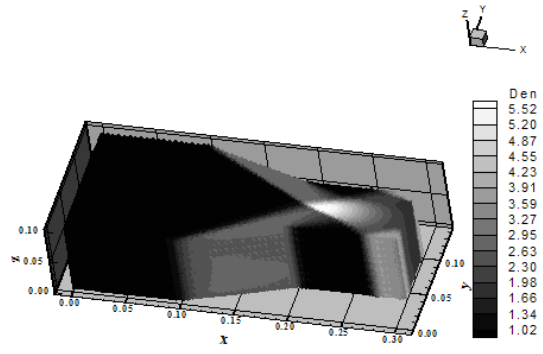


Figure 37 : Density contours (FPP).

Figures 38 to 42 exhibit the pressure contours obtained by the [4], the [5], the [6], the [8] and the [9] schemes, respectively, to the hypersonic case. The [8] scheme again presented the most severe pressure field, which characterizes this one as the most conservative among the studied schemes.

Figures 43 and 44 show the pressure contours obtained by the [3] and by the [7] schemes, respectively. The [3] scheme presents a more severe pressure field than the [7].

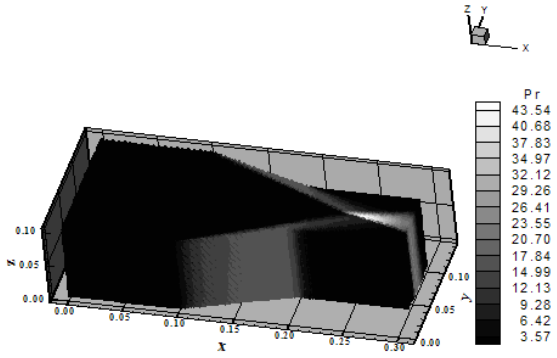


Figure 38 : Pressure contours (SW).

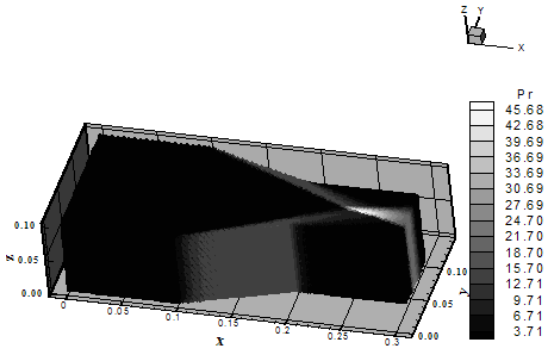


Figure 39 : Pressure contours (VL).

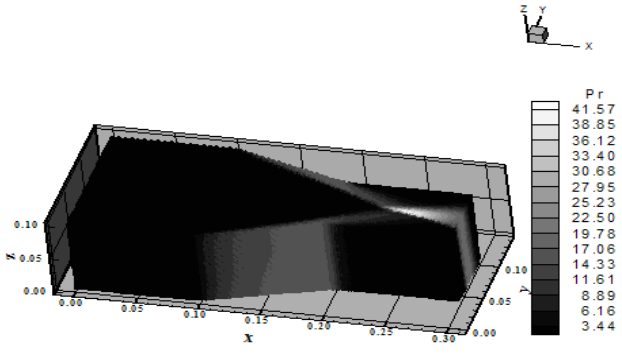


Figure 40 : Pressure contours (H).

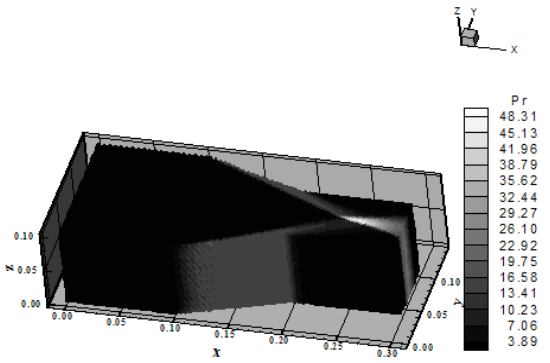


Figure 41 : Pressure contours (LS).

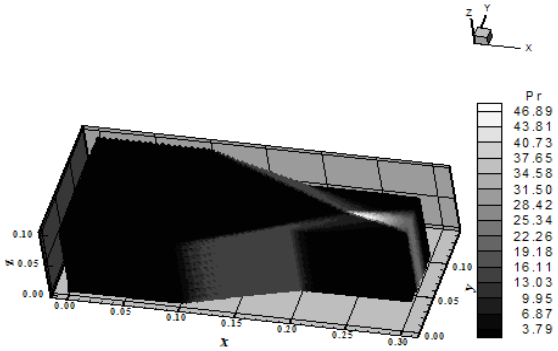


Figure 42 : Pressure contours (RK).

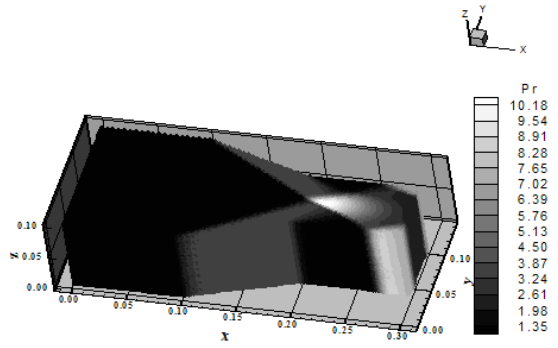


Figure 43 : Pressure contours (R).

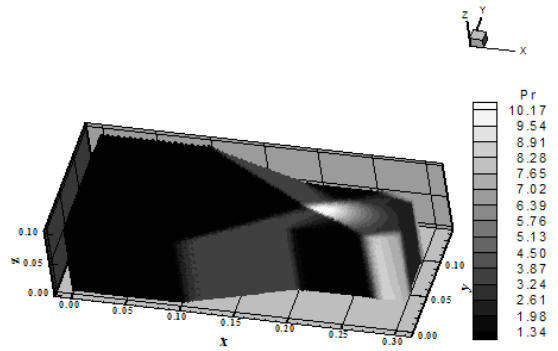


Figure 44 : Pressure contours (FPP).

Figures 45 to 49 exhibit the Mach number contours obtained by the [4], the [5], the [6], the [8] and the [9] schemes, respectively, to the hypersonic case. The most intense Mach number filed is obtained with the [4] scheme. Good homogeneity and symmetry properties are observed in all schemes, except in the [6] scheme.

Figures 50 and 51 show the Mach number field obtained with the [3] and the [7] schemes, respectively. The [3] scheme presents more intense Mach number field than the [7] scheme.

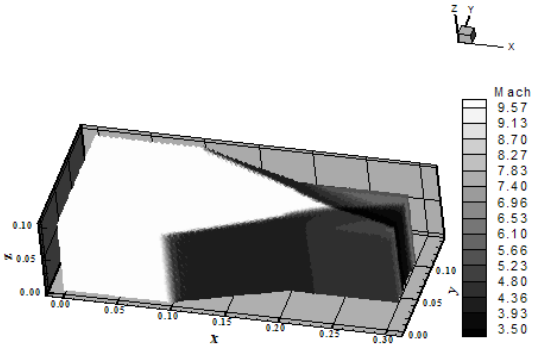


Figure 45 : Mach contours (SW).

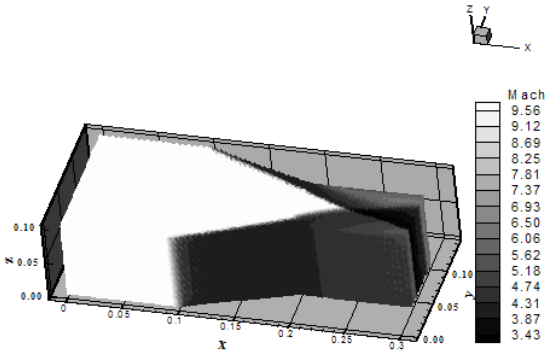


Figure 46 : Mach contours (VL).

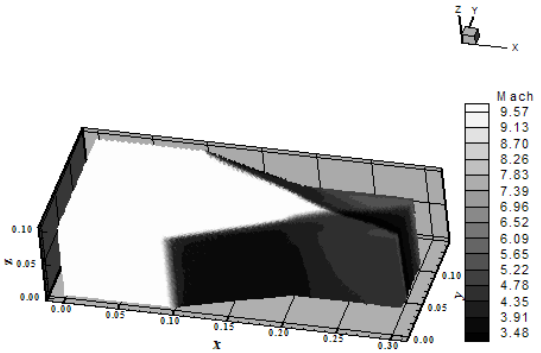


Figure 47 : Mach contours (H).

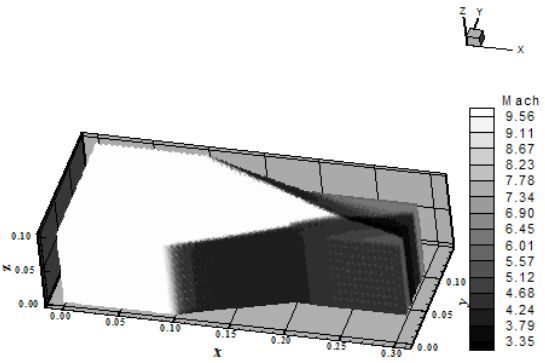


Figure 48 : Mach contours (LS).

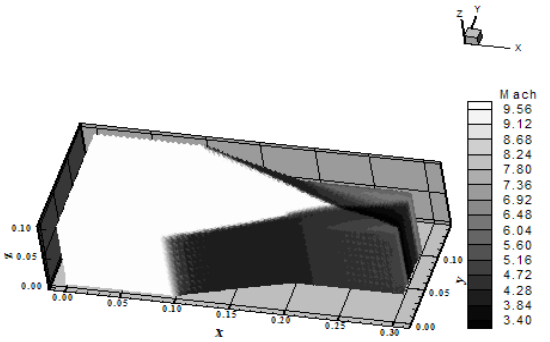


Figure 49 : Mach contours (RK).

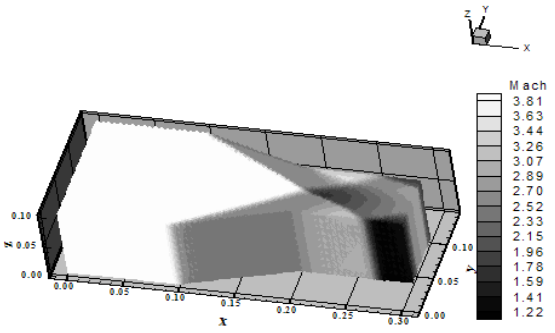


Figure 50 : Mach contours (R).

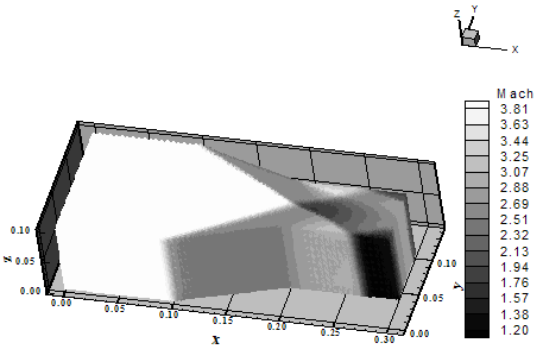


Figure 51 : Mach contours (FPP).

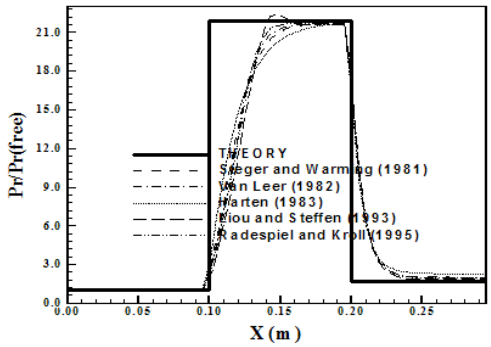


Figure 52 : Wall pressure distributions (HC).

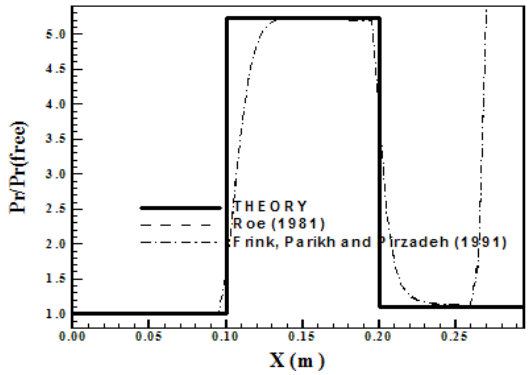


Figure 53 : Wall pressure distributions (Roe and FPP).

Figure 52 exhibits the lower wall pressure distributions obtained with the [4], the [5], the [6], the [8] and the [9] schemes to the hypersonic case. They are again compared with exact solutions of the oblique shock wave and the Prandtl-Meyer expansion wave theories. The [9] scheme presents better pressure width at the pressure plateau, best estimative of the pressure at the ramp (convergent region of the diffuser), and better pressure at the end of the expansion fan. Figure 53 shows the lower wall pressure distributions obtained only with the [3] and the [7] schemes to the supersonic case. They are compared with the oblique shock wave and the Prandtl-Meyer expansion fan theories. Both solutions are identical, without meaningful differences.

Another way to check the accuracy of the schemes to this problem consists in determining the shock angle β of the oblique shock waves at the lower and upper walls of the diffuser. Following the same analysis described in the ramp problem, to ϕ = 20°, angle of inclination of the convergent region of the diffuser and of the deflection of the flow after the shock wave, and to a freestream Mach number equals to 10.0, it is possible to find from [19] the value β = 26.0°. Using a transfer in Figures 38 to 42, in the xy plane, it is possible to obtain the values of the angles of the oblique shock waves at the upper and lower walls of the diffuser and respective percentage errors shown in Tab. 3.

The same analysis is performed to the [3] and the [7] schemes. To ϕ = 20° and to a freestream Mach number equals to 4.0, it is possible to find from [19] the value β = 32.5°, as obtained in the ramp problem. Using a transfer in Figures 43 and 44, in the xy plane, it is possible to obtain the values of the angles of the oblique shock waves at the upper and lower walls of the diffuser and respective percentage errors also shown in Tab. 3. In this table, was adopted the following convention:
R81 = [3], SW81 = [4], VL82 = [5], H83 = [6], FPP91 = [7], LS93 = [8] and RK95 = [9].

Table 3 : Shock angles of the oblique shock waves at lower and upper walls of the diffuser and percentage errors to each scheme.

	Diffuser Lower wall		Diffuser Upper wall	
Scheme	**β (°)**	**Error (%)**	**β (°)**	**Error (%)**
R81[(1)]	33.2	2.154	32.2	0.923
SW81	26.3	1.154	25.5	1.923
VL82	26.0	0.000	25.7	1.154
H83	27.0	3.846	25.7	1.154
FPP91[(1)]	32.2	0.923	32.5	0.000
LS93	25.5	1.923	25.4	2.308
RK95	25.5	1.923	25.6	1.538

[(1)] Diffuser problem to M_∞ = 4.0.

As can be observed from Tab. 3, to the hypersonic conditions, the [5] scheme presents the best values to the shock angles of the oblique shock waves at the lower and upper wall of the diffuser in comparison with the others schemes, resulting in the best choice to this problem. To the supersonic conditions, the [3] and the [7] schemes are equivalent.

13.3 Numerical data of the simulations

Table 4 shows the numerical data of the simulations: maximum CFL number and the number of iterations to convergence of each scheme

analyzed in this work. It is possible to note that all schemes used a CFL number less than 0.5, which is a typical behavior of CFD codes in supersonic and hypersonic flows. Supersonic and hypersonic flows require more robustness of the numerical methods due to the severity of high Mach numbers. It becomes these flow regimes very difficult to simulate. Moreover, in this paper was supposed the "cold gas" flow, which becomes the flow conditions also less severe. The best behavior was due to the [8] scheme, which converged in both cases with a minimum number of iterations.

Table 4 : Numerical data of the simulations.

	Ramp		Diffuser	
Scheme	**CFL**	**Iteration**	**CFL**	**Iteration**
R81	0.2	980	0.2[(1)]	1,874
SW81	0.2	979	0.3	736
VL82	0.2	975	0.3	725
H83	0.2	1,018	0.1	2,356
FPP91	0.2	980	0.3[(1)]	1,260
LS93	0.2	972	0.3	720
RK95	0.2	973	0.3	724

[(1)] Diffuser problem to $M_\infty = 4.0$.

Table 5 : Computational costs.

Algorithm	Computational Cost[(1)]
R81	0.0000588
SW81	0.0000157
VL82	0.0000166
H83	0.0000659
FPP91	0.0000450
LS93	0.0000158
RK95	0.0000164

[(1)] Measured in seconds/per cell/per iterations.

Table 5 exhibits the computational cost of each scheme. As can be observed from Table 5, the first order scheme of [4] is the cheapest, while the [6] scheme is the most expensive. It is approximately 319.7% more expensive than the [4] scheme. The expected behavior was that the flux vector splitting schemes were less expensive than the flux difference splitting ones. It is due to the matrix-vector product that occurs in the latters. In the formers, the dissipation function is defined without this type of product, what mark down these schemes.

14 Conclusions

In the present work, the [3], the [4], the [5], the [6], the [7], the [8] and the [9] schemes are implemented, on a finite volume context and using an upwind and unstructured spatial discretization, to solve the Euler equations in the three-dimensional space. The [3], the [6], and the [7] schemes are flux difference splitting ones and more accurate solutions are expected. On the other hand, the [4], the [5], the [8], and the [9] schemes are flux vector splitting ones and more robustness properties are expected. The implemented schemes are first order accurate in space. The time integration uses a Runge-Kutta method and is second order accurate. The physical problems of the supersonic flow along a ramp and the "cold gas" hypersonic flow along a diffuser are solved. All the seven algorithms are accelerated to the steady state solution using a spatially variable time step. This technique has proved excellent gains in terms of convergence ratio as reported in [20].

The results have demonstrated that the [8] scheme is the most conservative algorithm among the studied ones, whereas the [5] scheme is the most accurate. The [8] scheme yielded the most severe pressure field in the ramp problem, which indicates this one as a more conservative scheme to the prediction of moderate design conditions. The pressure distribution along the ramp was well predicted by all schemes, with the exception of the [8] scheme which presented an overshoot at the ramp. The [3] scheme presented better shock capturing properties due to the use of [3] average. In the estimation of the angle of the oblique shock wave, five of the seven schemes presented appropriate predictions (errors less than 1.0%). The exceptions were the [6], the worst, and the [8] schemes. The best results were obtained by the [5] and the [7] schemes. In the diffuser problem, the [3] and the [7] schemes were not so robust as the others schemes and simulated a less severe initial condition, what characterized a supersonic case. The others five schemes simulated the intended "cold gas" hypersonic flow. The following comments are related only with the hypersonic case. The most severe pressure field was again estimated by the [8] scheme, which also indicates this scheme to more severe design conditions of aerospace vehicles. The

lower wall pressure distribution was more appropriately described by the [9] scheme. In the prediction of the shock angles of the oblique shock waves at the lower and upper walls of the diffuser, the [5] scheme was the best. In terms of computational cost, the [4] scheme was the cheapest, whereas the [6] scheme was the most expensive. It was 319.7% more expensive than the [4] scheme, which penalizes its use.

As final conclusion, the present author recommends the [5] scheme, among the studied algorithms, to obtain more accurate solutions in the three-dimensional space. The [8] scheme, due to its confirmed robustness and more conservative properties, could be used in the initial design phase of aerospace vehicles, where less refined results are characteristics.

References:

[1] P. Kutler, Computation of Three Dimensional, Inviscid Supersonic Flows, *Lecture Notes in Physics*, Springer Verlag, Berlin, 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, 1981, pp. 357-372.

[4] J. L. Steger, R. F. Warming, Flux Vector Splitting of the Inviscid Gasdynamic Equations with Application to Finite-Difference Methods. *Journal of Computational Physics*, Vol. 40, 1981, pp. 263-293.

[5] B. Van Leer, Flux-Vector Splitting for the Euler Equations, *Lecture Notes in Physics*, Springer Verlag, Berlin, Vol. 170, 1982, pp. 507-512.

[6] A. Harten, High Resolution Schemes for Hyperbolic Conservation Laws, *Journal of Computational Physics*, Vol. 49, 1983, pp. 357-393.

[7] N. T. Frink, P. Parikh, S. Pirzadeh, Aerodynamic Analysis of Complex Configurations Using Unstructured Grids, *AIAA 91-3292-CP*, 1981.

[8] M. Liou, C. J. Steffen Jr., A New Flux Splitting Scheme, *Journal of Computational Physics*, Vol. 107, 1993, pp. 23-39.

[9] R. Radespiel, N. Kroll, Accurate Flux Vector Splitting for Shocks and Shear Layers, *Journal of Computational Physics*, Vol. 121, 1995, pp. 66-78.

[10] E. S. G. Maciel, Comparison Between a Centered and a Flux Difference Split Schemes Using Unstructured Strategy, *JBSMSE – Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Brazil, Vol. XXVII, No. 3, 2005, pp. 223-235.

[11] E. S. G. Maciel, Comparison Between a Centered and a High Resolution Upwind Schemes in the Solution of Aerospace Problems Using Unstructured Strategy, *Proceedings of the XVIII International Congress of Mechanical Engineering (XVIII COBEM)*, Ouro Preto, MG, Brazil, 2005.

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

[13] D. J. Mavriplis, Accurate Multigrid Solution of the Euler Equations on Unstructured and Adaptive Meshes, *AIAA Journal*, Vol. 28, No. 2, 1990, pp. 213-221.

[14] E. S. G. Maciel, Solutions of the Euler and the Navier-Stokes Equations Using the Jameson and Mavriplis and the Liou and Steffen Unstructured Algorithms in Three-Dimensions, *Engineering Applications of Computational Fluid Mechanics*, China, Vol. 1, No. 4, 2007, pp. 238-252.

[15] S. Pirzadeh, Structured Background Grids for Generation of Unstructured Grids by Advancing Front Method. *AIAA Paper 91-3233-CP*, 1991.

[16] C. Hirsch, *Numerical Computation of Internal and External Flows – Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons Ltd, 691p, 1990.

[17] 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.

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

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

[20] 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.