Explicit and Implicit Solutions of First Order Algorithms Applied to the Euler Equations in Three-Dimensions

Edisson Sávio de Góes Maciel
Aeronautical Engineering Division (IEA)
Aeronautical Technological Institute (ITA)
Pça. Mal. do Ar Eduardo Gomes, 50 – Vila das Acácias – São José dos Campos – SP – 12228-900
Brazil
edisavio@edissonsavio.eng.br

Abstract: - In the present work, the Roe, the Steger and Warming, the Van Leer, the Chakravarthy and Osher, 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 structured spatial discretization, to solve the Euler equations in the three-dimensional space. The Roe, the Harten, the Chakravarthy and Osher and the Frink, Parikh and Pirzadeh schemes are flux difference splitting ones, whereas the others schemes are flux vector splitting ones. All eight schemes are first order accurate in space and their explicit and implicit versions are implemented in three-dimensions. The explicit time integration uses a Runge-Kutta, a time splitting or an Euler method. The former is second order accurate in time, whereas the others are first order accurate in time. In the implicit case, an ADI approximate factorization is employed, which is first order accurate in time. The physical problems of the supersonic flow along a ramp, in the implicit case, and the "cold gas" hypersonic flows around a blunt body and along an air inlet, in the explicit case, 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, Structured 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 ([1-2]). 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: [3-10]. 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] presented an upwind, shock capturing algorithm generalized to arbitrary coordinate systems. It could be applied to essentially all hyperbolic systems of conservation laws arising in physics, but became especially simple for the Euler equations. The method did not require any special properties of the Euler equations such as homogeneity. The [6] scheme was based on a Riemann problem solver, where compression waves are used to approximate shocks. This leads to a cleaner and smoother algorithm.

[7] 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.

[8] 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.

[9] 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.

[10] emphasized that the [9] 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 [10] work, a hybrid flux vector splitting scheme, which alternated between the [9] scheme and the [5] scheme, in the shock wave regions, is proposed, assuring that resolution of strength shocks was clear and sharply defined.

Traditionally, implicit numerical methods have been praised for their improved stability and condemned for their large arithmetic operation counts ([11]). 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 [12-14]. 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 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.

Considering the two-dimensional case, [15-16] studied the [3-10] and [17] algorithms, first order accurate in space, implemented employing explicit and implicit formulations to solve the Euler equations. Such algorithms were implemented according to a finite volume methodology and using structured spatial discretization. The [3, 6-8] algorithms were flux difference splitting ones, whereas the others were flux vector splitting algorithms. The implicit schemes employed an ADI approximate factorization or Symmetric Line Gauss-Seidel to solve implicitly the Euler equations. Explicit and implicit results were compared, as also the computational costs, trying to emphasize the advantages and disadvantages of each formulation. The schemes were accelerated to the steady state solution using a spatially variable time step, which had demonstrated effective gains in terms of convergence rate according to [18-19]. The algorithms were applied to the solution of the physical problem of the moderate supersonic flow along a compression corner. The results had demonstrated that the most accurate solutions were obtained with the [7] first order scheme, when implemented in its explicit version. The best wall pressure distribution was obtained by the [10] first order scheme, in both explicit and implicit cases.

In the present work, the [3-10] schemes are implemented, on a finite volume context and using an upwind and structured spatial discretization, to solve the Euler equations in the three-dimensional space. The [3] and [6-8] schemes are flux difference splitting ones and more accurate solutions are expected. On the other hand, the [4-5] and [9-10], schemes are flux vector splitting ones and more robustness properties are expected. The implemented schemes are first order accurate in space. The explicit time integration uses a Runge-Kutta, a time splitting or an Euler method. The former is second order accurate in time, whereas the others are first order accurate in time. In the implicit case, an ADI approximate factorization is employed, which is first order accurate in time. The physical problems of the supersonic flow along a ramp, in the implicit case, and the “cold gas” hypersonic flows around a blunt body and along an air inlet, in the explicit case, are solved. All the eight 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 [18-19].

The results have demonstrated that the [9] scheme is the most conservative algorithm among the studied ones, whereas the [5] scheme is the most accurate.

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:

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

where Q is written to a Cartesian system, V is a cell volume, which corresponds to an hexahedron 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:

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

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

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:

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

The geometrical domain with the description of a structured cell, its nodes, interfaces and neighbors, as also the calculation of the cell volume, cell surfaces and normal unity vectors are found in [20-21].

3 Roe Numerical Algorithm

The [3] algorithm, first order accurate in space, is specified by the determination of the numerical flux vector at the (i+1/2,j,k) interface. Its extension to the (i,j+1/2,k) and the (i,j,k+1/2) interfaces is straightforward, without any complication.

Following a finite volume formulation, 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:

$$V_R = V_{i+1,j,k}\,,\ V_L = V_{i,j,k}\ \text{and}\ V_{\text{int}} = 0.5\left(V_R + V_L\right), \quad (6)$$

where "*R*" and "*L*" represent right and left states in relation to the interface, respectively. The metric terms to this coordinate system are defined as:

$$h_x = S_{x,\text{int}}/V_{\text{int}}\,,\ h_y = S_{y,\text{int}}/V_{\text{int}}\,,\ h_z = S_{z,\text{int}}/V_{\text{int}}\,;$$
$$h_n = S_{\text{int}}/V_{\text{int}}\,, \quad (7)$$

with, $S_{x,\text{int}}$, $S_{y,\text{int}}$ and $S_{z,\text{int}}$ representing the surface area components and S_{int} is the norm of the surface area vector.

The properties calculated at the flux interface are obtained by arithmetical average or by Roe's average. In the present work, the Roe's average was used:

$$\rho_{\text{int}} = \sqrt{\rho_L\rho_R}\,,\ u_{\text{int}} = \left(u_L + u_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right);$$
$$v_{\text{int}} = \left(v_L + v_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right);$$
$$w_{\text{int}} = \left(w_L + w_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (8)$$
$$H_{\text{int}} = \left(H_L + H_R\sqrt{\rho_R/\rho_L}\right)/\left(1+\sqrt{\rho_R/\rho_L}\right); \quad (9)$$
$$a_{\text{int}} = \sqrt{(\gamma-1)\left[H_{\text{int}} - 0.5\left(u_{\text{int}}^2 + v_{\text{int}}^2 + w_{\text{int}}^2\right)\right]}, \quad (10)$$

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

$$q_{normal} = u_{\text{int}}h_x + v_{\text{int}}h_y + w_{\text{int}}h_z\,,\ \lambda_1 = q_{normal} - a_{\text{int}}h_n\,;$$
$$\lambda_2 = \lambda_3 = \lambda_4 = q_{normal},\ \lambda_5 = q_{normal} + a_{\text{int}}h_n\,. \quad (11)$$

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

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

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

$$\left\{\alpha_{\text{int}}\right\} = \left[R^{-1}\right]_I\left\{\Delta_{\text{int}}\overline{Q}\right\}, \quad (15)$$

with:

$$\left[R^{-1}\right] = \begin{bmatrix} 0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 + \phi/a_{\text{int}}\right] & -0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} + h_x'/a_{\text{int}}\right] & -0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} + h_y'/a_{\text{int}}\right] & -0.5\left[(\gamma-1)/a_{\text{int}}^2 w_{\text{int}} + h_z'/a_{\text{int}}\right] & 0.5(\gamma-1)/a_{\text{int}}^2 \\ 1 - \left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 & \left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} & \left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} & (\gamma-1)/a_{\text{int}}^2 w_{\text{int}} & -(\gamma-1)/a_{\text{int}}^2 \\ -\left(h_y'u_{\text{int}} + h_z'v_{\text{int}} + h_x'w_{\text{int}}\right) & h_y' & h_z' & h_x' & 0 \\ -\left(h_z'u_{\text{int}} + h_x'v_{\text{int}} + h_y'w_{\text{int}}\right) & h_z' & h_x' & h_y' & 0 \\ 0.5\left[\left((\gamma-1)/a_{\text{int}}^2\right)0.5q^2 - \phi/a_{\text{int}}\right] & 0.5\left[-\left((\gamma-1)/a_{\text{int}}^2\right)u_{\text{int}} + h_x'/a_{\text{int}}\right] & 0.5\left[-\left((\gamma-1)/a_{\text{int}}^2\right)v_{\text{int}} + h_y'/a_{\text{int}}\right] & 0.5\left[-(\gamma-1)/a_{\text{int}}^2 w_{\text{int}} + h_z'/a_{\text{int}}\right] & 0.5(\gamma-1)/a_{\text{int}}^2 \end{bmatrix}; \quad (16)$$

$\left\{\Delta_{\text{int}}\overline{Q}\right\} = \left\{\Delta\rho \quad \Delta(\rho u) \quad \Delta(\rho v) \quad \Delta(\rho w) \quad \Delta e\right\}^T$, defined by Eqs. (12), (13) and (14); (17)

$$q^2 = u_{\text{int}}^2 + v_{\text{int}}^2 + w_{\text{int}}^2; \quad (18)$$

$$\phi = u_{\text{int}}h_x' + v_{\text{int}}h_y' + w_{\text{int}}h_z'; \quad (19)$$

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

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

$$[R] = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ u_{\text{int}} - h_x'a_{\text{int}} & u_{\text{int}} & h_y' & h_z' & u_{\text{int}} + h_x'a_{\text{int}} \\ v_{\text{int}} - h_y'a_{\text{int}} & v_{\text{int}} & h_z' & h_x' & v_{\text{int}} + h_y'a_{\text{int}} \\ w_{\text{int}} - h_z'a_{\text{int}} & w_{\text{int}} & h_x' & h_y' & w_{\text{int}} + h_z'a_{\text{int}} \\ H_{\text{int}} - h_x'u_{\text{int}}a_{\text{int}} - h_y'v_{\text{int}}a_{\text{int}} - h_z'w_{\text{int}}a_{\text{int}} & 0.5q^2 & h_x'w_{\text{int}} + h_z'v_{\text{int}} + h_y'u_{\text{int}} & h_y'w_{\text{int}} + h_x'v_{\text{int}} + h_z'u_{\text{int}} & H_{\text{int}} + h_x'u_{\text{int}}a_{\text{int}} + h_y'v_{\text{int}}a_{\text{int}} + h_z'w_{\text{int}}a_{\text{int}} \end{bmatrix}. \quad (21)$$

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

$$\psi_m = \begin{cases} \left|\lambda_m\right|, & if\ \left|\lambda_m\right| \geq \varepsilon_m \\ 0.5\left(\lambda_m^2 + \varepsilon_m^2\right)/\varepsilon_m, & if\ \left|\lambda_m\right| < \varepsilon_m \end{cases}$$ to non-linear fields and $\psi_m = \left|\lambda_m\right|$ 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 interface is constructed by the following matrix-vector product:

$$\{D_{Roe}\}_{\text{int}} = [R]_{\text{int}}\{-\psi\alpha\}_{\text{int}} . \qquad (23)$$

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

$$F^{(m)}_{i+1/2,j,k} = \left(E^{(m)}_{\text{int}} h_x + F^{(m)}_{\text{int}} h_y + G^{(m)}_{\text{int}} h_z\right) V_{\text{int}} + 0.5 D^{(m)}_{Roe} , \qquad (24)$$

where:

$$E^{(m)}_{\text{int}} = 0.5\left(E^{(m)}_R + E^{(m)}_L\right),\; F^{(m)}_{\text{int}} = 0.5\left(F^{(m)}_R + F^{(m)}_L\right);$$
$$G^{(m)}_{\text{int}} = 0.5\left(G^{(m)}_R + G^{(m)}_L\right). \qquad (25)$$

The right-hand-side (RHS) of the [3] scheme, necessaries to the resolution of the implicit version of this algorithm, is determined by:

$$RHS^n_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left[F^{Roe}_{i+1/2,j,k} - F^{Roe}_{i-1/2,j,k} + F^{Roe}_{i,j+1/2,k} - F^{Roe}_{i,j-1/2,k} + F^{Roe}_{i,j,k+1/2} - F^{Roe}_{i,j,k-1/2}\right]^n . \qquad (26)$$

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

$$\Delta Q^{*}_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left(F^{n}_{i+1/2,j,k} - F^{n}_{i-1/2,j,k}\right);$$
$$Q^{*}_{i,j,k} = Q^{n}_{i,j,k} + \Delta Q^{*}_{i,j,k} ; \qquad (27)$$

in the intermediate step, η direction:

$$\Delta Q^{**}_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left(F^{*}_{i,j+1/2,k} - F^{*}_{i,j-1/2,k}\right);$$
$$Q^{**}_{i,j,k} = Q^{*}_{i,j,k} + \Delta Q^{**}_{i,j,k} . \qquad (28)$$

and at the final step, ζ direction:

$$\Delta Q^{n+1}_{i,j,k} = -\Delta t_{i,j,k} / V_{i,j,k} \left(F^{**}_{i,j,k+1/2} - F^{**}_{i,j,k-1/2}\right);$$
$$Q^{n+1}_{i,j,k} = Q^{**}_{i,j,k} + \Delta Q^{n+1}_{i,j,k} . \qquad (29)$$

4 Steger and Warming Numerical Algorithm

4.1 Theory for the One-Dimensional Case

If the homogeneous Euler equations are put in characteristic form

$$\partial W / \partial t + \Lambda\, \partial W / \partial x = 0 , \qquad (30)$$

where W is the vector of characteristic variables (defined in [22]) and Λ is the diagonal matrix of eigenvalues, the upwind scheme:

$$u^{n+1}_i - u^n_i = -\Delta t / \Delta x \left[\hat{a}^{+}\left(u^n_i - u^n_{i-1}\right) + \hat{a}^{-}\left(u^n_{i+1} - u^n_i\right)\right], \qquad (31)$$

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

$$\lambda^{+}_m = 0.5\left(\lambda_m + |\lambda_m|\right) \quad \text{and} \quad \lambda^{-}_m = 0.5\left(\lambda_m - |\lambda_m|\right) \qquad (32)$$

for each of the eigenvalues of Λ

$$\Lambda = \begin{bmatrix} \lambda_1 & & \\ & \lambda_2 & \\ & & \lambda_3 \end{bmatrix} = \begin{bmatrix} u & & \\ & u + a & \\ & & u - a \end{bmatrix} . \qquad (33)$$

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

$$\Lambda^{\pm} = \begin{bmatrix} \lambda^{\pm}_1 & & \\ & \lambda^{\pm}_2 & \\ & & \lambda^{\pm}_3 \end{bmatrix} = \begin{bmatrix} 0.5\left(u \pm |u|\right) & & \\ & 0.5\left(u + a \pm |u + a|\right) & \\ & & 0.5\left(u - a \pm |u - a|\right) \end{bmatrix} , \qquad (34)$$

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

$$\Lambda = \Lambda^{+} + \Lambda^{-} \quad \text{and} \quad |\Lambda| = \Lambda^{+} - \Lambda^{-} \quad \text{or}$$
$$\lambda_m = \lambda^{+}_m + \lambda^{-}_m \quad \text{and} \quad |\lambda_m| = \lambda^{+}_m - \lambda^{-}_m . \qquad (35)$$

The quasi-linear coupled equations are obtained from the characteristic form by the transformation matrix P (defined in [22]), with the Jacobian A satisfying

$$A = P\Lambda P^{-1}, \text{ resulting in } \partial Q / \partial t + A\,\partial Q / \partial x = 0 . \qquad (36)$$

Hence an upwind formulation can be obtained with the Jacobians

$$A^{+} = P\Lambda^{+}P^{-1} \quad \text{and} \quad A^{-} = P\Lambda^{-}P^{-1}, \text{ with:}$$
$$A = A^{+} + A^{-} \text{ and } |A| = A^{+} - A^{-}. \qquad (37)$$

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:

$$f^{+} = A^{+}Q \quad \text{and} \quad f^{-} = A^{-}Q, \text{ with: } f = f^{+} + f^{-}. \qquad (38)$$

This flux vector splitting, based on Eq. (32), has been introduced by [4]. The split fluxes f^{+} and f^{-} are also homogeneous functions of degree one in *Q*.

4.2 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

$$K^{(n)} = \vec{A}\bullet\vec{n} = An_x + Bn_y + Cn_z, \qquad (39)$$

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

$$\tilde{F}^{(n)} = \tilde{\tilde{F}}\bullet\vec{n} = En_x + Fn_y + Gn_z \qquad (40)$$

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. (32), the normal flux projection, Eq. (40), is decomposed by the [4] flux splitting as

$$\tilde{F}^{(n)}_{\pm} = \frac{\rho}{2\gamma}\left\{\begin{array}{c} \alpha \\ \alpha u + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_x \\ \alpha v + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_y \\ \alpha w + a\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right)n_z \\ \alpha\dfrac{u^2+v^2+w^2}{2} + av_n\left(\lambda_2^{\pm} - \lambda_3^{\pm}\right) + a^2\dfrac{\lambda_2^{\pm}+\lambda_3^{\pm}}{\gamma-1} \end{array}\right\}, \qquad (41)$$

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

$$\lambda_1 = \vec{v}\bullet\vec{n} \equiv v_n, \ \lambda_2 = \vec{v}\bullet\vec{n} + a \text{ and } \lambda_3 = \vec{v}\bullet\vec{n} - a, \qquad (42)$$

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

$$\alpha = 2(\gamma-1)\lambda_1^{\pm} + \lambda_2^{\pm} + \lambda_3^{\pm}. \qquad (43)$$

4.3 Numerical scheme

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

$$\tilde{F}_{i,j-1/2,k} = \left(\tilde{F}^{-}_{i,j-1,k} + \tilde{F}^{+}_{i,j,k}\right)S_{i,j-1/2,k},$$
$$\tilde{F}_{i+1/2,j,k} = \left(\tilde{F}^{-}_{i+1,j,k} + \tilde{F}^{+}_{i,j,k}\right)S_{i+1/2,j,k}; \qquad (44)$$
$$\tilde{F}_{i,j+1/2,k} = \left(\tilde{F}^{-}_{i,j+1,k} + \tilde{F}^{+}_{i,j,k}\right)S_{i,j+1/2,k},$$
$$\tilde{F}_{i-1/2,j,k} = \left(\tilde{F}^{-}_{i-1,j,k} + \tilde{F}^{+}_{i,j,k}\right)S_{i-1/2,j,k}; \qquad (45)$$
$$\tilde{F}_{i,j,k+1/2} = \left(\tilde{F}^{-}_{i,j,k+1} + \tilde{F}^{+}_{i,j,k}\right)S_{i,j,k+1/2},$$
$$\tilde{F}_{i,j,k-1/2} = \left(\tilde{F}^{-}_{i,j,k-1} + \tilde{F}^{+}_{i,j,k}\right)S_{i,j,k-1/2}; \qquad (46)$$

The Euler explicit method, first order accurate in time, is employed to perform the time integration:

$$Q^{n+1}_{i,j,k} = Q^{n}_{i,j,k} - \Delta t_{i,j,k}/V_{i,j,k}\left(\tilde{F}^{SW}_{i,j-1/2,k} + \tilde{F}^{SW}_{i+1/2,j,k} + \tilde{F}^{SW}_{i,j+1/2,k} + \tilde{F}^{SW}_{i-1/2,j,k} + \tilde{F}^{SW}_{i,j,k+1/2} + \tilde{F}^{SW}_{i,j,k-1/2}\right)^{n}. \qquad (47)$$

The RHS to the implicit simulations is determined by:

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

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

5 Van Leer Numerical Algorithm

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

$$V_{i,j,k}\, dQ_{i,j,k}/dt = -R_{i,j,k}, \qquad (49)$$

with $R_{i,j,k}$ representing the neat flux (residue) of the conservation of mass, of linear momentum and of energy in the $V_{i,j,k}$ volume. The residue is calculated as:

$$R_{i,j,k} = R_{i+1/2,j,k} - R_{i-1/2,j,k} + R_{i,j+1/2,k} - R_{i,j-1/2,k} + R_{i,j,k+1/2} - R_{i,j,k-1/2}\,, \qquad (50)$$

where $R_{i+1/2,j} = R^{c}_{i+1/2,j}$, in which "c" is related to the flow convective contribution.

As shown in [9], 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 (i+1/2,j,k) cell face, related to cell (i,j,k) and its neighbor, respectively, multiplied by the interface Mach number, and a scalar dissipative term. Hence, to the (i+1/2,j,k) interface:

$$R_{i+1/2,j,k} = |S|_{i+1/2,j,k}\left(\frac{1}{2}M_{i+1/2,j,k}\left(\begin{bmatrix}\rho a\\ \rho au\\ \rho av\\ \rho aw\\ \rho aH\end{bmatrix}_L + \begin{bmatrix}\rho a\\ \rho au\\ \rho av\\ \rho aw\\ \rho aH\end{bmatrix}_R\right) - \frac{1}{2}\phi_{i+1/2,j,k}\left(\begin{bmatrix}\rho a\\ \rho au\\ \rho av\\ \rho aw\\ \rho aH\end{bmatrix}_R - \begin{bmatrix}\rho a\\ \rho au\\ \rho av\\ \rho aw\\ \rho aH\end{bmatrix}_L\right)\right) + \begin{bmatrix}0\\ S_x p\\ S_y p\\ S_z p\\ 0\end{bmatrix}_{i+1/2,j,k}, \qquad (51)$$

where $S_{i+1/2,j,k} = \begin{bmatrix}S_x & S_y & S_z\end{bmatrix}^T_{i+1/2,j,k}$ defines the normal area vector to the (i+1/2,j,k) surface. $M_{i+1/2,j,k}$ defines the advection Mach number at the (i+1/2,j,k) face of the (i,j,k) cell, which is calculated according to [9] as:

$$M_l = M_L^+ + M_R^-\,, \qquad (52)$$

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

$$M^+ = \begin{bmatrix} M, & if\ M \geq 1; \\ 0.25(M+1)^2, & if\ |M| < 1; \\ 0, & if\ M \leq -1; \end{bmatrix} \qquad (53a)$$

and

$$M^- = \begin{bmatrix} 0, & if\ M \geq 1; \\ -0.25(M-1)^2, & if\ |M| < 1; \\ M, & if\ M \leq -1. \end{bmatrix} \qquad (53b)$$

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

$$M = \left(S_x u + S_y v + S_z w\right)/\left(|S|a\right). \qquad (54)$$

The pressure at the (i+1/2,j,k) face of the (i,j,k) cell is calculated by a similar way:

$$p_{i+1/2,j,k} = p_L^+ + p_R^-\,, \qquad (55)$$

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

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

And

$$p^- = \begin{bmatrix} 0, & if\ M \geq 1; \\ 0.25p(M-1)^2(2+M), & if\ |M| < 1; \\ p, & if\ M \leq -1. \end{bmatrix} \qquad (56)$$

The definition of the dissipative term ϕ determines the particular formulation of the convective fluxes. According to [10], the choice below corresponds to the [5] scheme:

$$\phi_l = \phi_l^{VL} = \begin{pmatrix} |M_l|, & if\ |M_l| \geq 1; \\ |M_l| + 0.5(M_R - 1)^2, & if\ 0 \leq M_l < 1; \\ |M_l| + 0.5(M_L + 1)^2, & if\ -1 < M_l \leq 0. \end{pmatrix} \qquad (57)$$

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 explicit time integration follows the time splitting method described in the [3] scheme [Eqs. (27-29)]. To the implicit time integration, it is necessary the following RHS definition:

$$RHS^n_{i,j,k} = -\Delta t_{i,j,k}/V_{i,j,k}\left[R^{VL}_{i+1/2,j,k} - R^{VL}_{i-1/2,j,k} + R^{VL}_{i,j+1/2,k} - R^{VL}_{i,j-1/2,k} + R^{VL}_{i,j,k+1/2} - R^{VL}_{i,j,k-1/2}\right]^n. \qquad (58)$$

The [5] scheme presented in this work is first order accurate in space.

6 Chakravarthy and Osher Numerical Algorithm

6.1 Eigenvalues

The Jacobian matrices in generalized coordinates, necessary to define the system's eigenvalues according to the [6] scheme, are defined by:

$$\hat{A}_{i+1/2,j,k} = \left(h_x A + h_y B + h_z C\right)_{i+1/2,j,k};$$
$$\hat{B}_{i,j+1/2,k} = \left(h_x A + h_y B + h_z C\right)_{i,j+1/2,k};$$
$$\hat{C}_{i,j,k+1/2} = \left(h_x A + h_y B + h_z C\right)_{i,j,k+1/2}, \quad (59)$$

where $A = \partial E/\partial Q$, $B = \partial F/\partial Q$ and $C = \partial G/\partial Q$ are the Cartesian Jacobian matrices. Remembering that the sound speed is determined by $a = \sqrt{\gamma p/\rho}$, the eigenvalues of $\hat{A}$ are defined by:

$$\lambda_1 = (U + a)h_n,\ \lambda_{2,3,4} = Uh_n \text{ and } \lambda_5 = (U - a)h_n, \quad (60)$$

with $U = h'_x u + h'_y v + h'_z w$.

6.2 Riemann Invariants

Riemann invariants are the building blocks for the [6] algorithm applied to Euler equations. Riemann invariants are associated with the eigenvalues of the generalized Jacobian matrices and are obtained from the corresponding right eigenvectors. For the [6] scheme, the Riemann invariants ψ corresponding to the *l*th eigenvalue are obtained by solving

$$\nabla_Q \psi \cdot r_l(Q) = 0, \quad (61)$$

where ∇_Q is the gradient operator with respect to the vector of dependent variables denoted by *Q* and r_l is the *l*th right eigenvector. It may easily be verified that the following are Riemann invariants:

For $\lambda_1 = (U + a)h_n$:

$$\psi_2^1 = U - 2a/(\gamma - 1);$$
$$\psi_3^1 = p/\rho^\gamma = S = \text{entropy};$$
$$\psi_4^1 = vh'_x - uh'_y = V;$$
$$\psi_5^1 = wh'_x - uh'_z = W; \quad (62)$$

For $\lambda_{2,3,4} = Uh_n$:

$$\psi_1^{2,3,4} = p \quad \text{and} \quad \psi_5^{2,3,4} = U; \quad (63)$$

For $\lambda_5 = (U - a)h_n$:

$$\psi_1^5 = U + 2a/(\gamma - 1),\ \psi_2^5 = p/\rho^\gamma = S = \text{entropy};$$
$$\psi_3^5 = vh'_x - uh'_y = V,\ \psi_4^5 = wh'_x - uh'_z = W. \quad (64)$$

The superscript denotes the eigenvalue to which the Riemann invariants correspond.

6.3 Intermediate States

In finite volumes the variable of interest are defined at the (i,j,k) cell centroid, where the vector of conserved variables is denoted as $Q_{i,j,k}$. The coordinate direction ξ is treated in details in this work and the extension to the η and ζ coordinates is straightforward. By simplicity, the indexes *j* and *k* are suppressed in the present notation.

Conventional finite volume schemes employ values at the cell centroids of the conserved variables, or dependents, of a simple manner. Such schemes are generally symmetric, what simplify their numerical implementation. The [6] algorithm is more sophisticated. Fundamental to the [6] scheme are the intermediate states of the dependent variables, which are defined from the values of the cell states of the computational domain. While states in the computational cells are defined by Q_{i-1}, Q_i, etc., the correspondent intermediate states are defined by $Q_{i-2/3}$ and $Q_{i-1/3}$. The rest of this sublevel describes how these intermediate states are defined and their meanings.

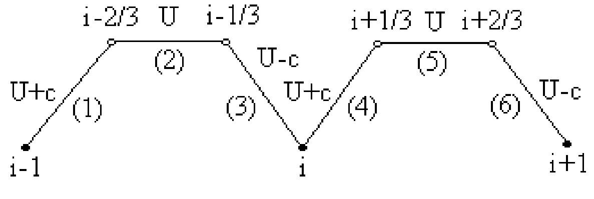

Figure 1. Schematic Representation of [6] Scheme in Terms of Intermediate and Cell Values of Dependent Variables.

Figure 1 serves as a guide to the construction of the intermediate values. The states *i-1* and *i* are connected through a curve in the state space which is made up of three subpaths. The first path connects Q_{i-1} and $Q_{i-2/3}$ and is associated with λ_1. Path 2 connecting $Q_{i-2/3}$ and $Q_{i-1/3}$ is associated with $\lambda_{2,3,4}$

and path 3 connecting $Q_{i-1/3}$ and Q_i is associated with λ_5. Thus $\psi^1_{2,3,4,5}$ are constant between Q_{i-1} and $Q_{i-2/3}$; $\psi^{2,3,4}_{1,5}$ are constant between $Q_{i-2/3}$ and $Q_{i-1/3}$; and $\psi^5_{1,2,3,4}$ are constant between $Q_{i-1/3}$ and Q_i. Equating Riemann invariants between the end points of each subpath it is possible to find 4+2+4 = 10 equations to obtain the 10 unknown values of $Q_{i-2/3}$ and $Q_{i-1/3}$ from the known values at Q_{i-1} and Q_i. Thus the dependent variables at *i-2/3* and *i-1/3* are defined by the following formulas:

$$\rho_{i-1/3}^{(\gamma-1)/2} = \left[(\gamma-1)(U_i - U_{i-1})/2 + a_i + a_{i-1}\right] / \left\lfloor a_i \left\lfloor 1 + \left(S_{i-1}/S_i\right)^{1/(2\gamma)} \right\rfloor\right\rfloor \rho_i^{(\gamma-1)/2}; \qquad (65)$$

$$\rho_{i-1/3}^{(\gamma-1)/2} = \left[(\gamma-1)(U_i - U_{i-1})/2 + a_i + a_{i-1}\right] / \left\lfloor a_{i-1} \left\lfloor 1 + \left(S_i/S_{i-1}\right)^{1/(2\gamma)} \right\rfloor\right\rfloor \rho_{i-1}^{(\gamma-1)/2}; \qquad (66)$$

$$p_{i-2/3} = p_{i-1/3} = S_{i-1}\rho_{i-2/3}^{\gamma}; \qquad (67)$$

$$U_{i-2/3} = U_{i-1} - 2/(\gamma-1)\left(a_{i-1} - a_{i-2/3}\right) = U_{i-1/3} = U_i + 2/(\gamma-1)\left(a_i - a_{i-1/3}\right); \qquad (68)$$

$$V_{i-2/3} = V_{i-1} \quad \text{and} \quad V_{i-1/3} = V_i. \qquad (69)$$

$$W_{i-2/3} = W_{i-1} \quad \text{and} \quad W_{i-1/3} = W_i. \qquad (70)$$

Once $Q_{i-1/3}$ and $Q_{i-2/3}$ are known, λ_1 may be computed at *i-1* and *i-2/3* and λ_5 at *i* and *i-1/3*. It can be shown that λ_1 and λ_5 can at most change sign only once along paths 1 and 3, respectively. If these eigenvalues do indeed change sign [if $\lambda_1(i-1)\cdot\lambda_1(i-2/3) < 0$, for example], it becomes necessary to compute the dependent variables at the points along paths 1 and 3 where the respective eigenvalues λ_1 and λ_5 vanish. These "sonic" points are defined as $\overline{Q}_{i-2/3}$ and $\overline{Q}_{i-1/3}$ and are given by the formulas that follow below:

$$\overline{U}_{i-1/3} = (\gamma-1)/(\gamma+1)\left(U_i + 2/(\gamma-1)\,a_i\right), \quad \overline{\rho}_{i-1/3}^{(\gamma-1)/2} = \overline{U}_{i-1/3}/(\gamma S_i)^{1/2}; \qquad (71)$$

$$\overline{p}_{i-1/3} = S_i\overline{\rho}_{i-1/3}^{\gamma},\ \overline{V}_{i-1/3} = V_i,\ \overline{W}_{i-1/3} = W_i; \qquad (72)$$

$$\overline{U}_{i-2/3} = (\gamma-1)/(\gamma+1)\left(U_{i-1} - 2/(\gamma-1)a_{i-1}\right), \quad \overline{\rho}_{i-2/3}^{(\gamma-1)/2} = -\overline{U}_{i-2/3}/(\gamma S_{i-1})^{1/2}; \qquad (73)$$

$$\overline{p}_{i-2/3} = S_{i-1}\overline{\rho}_{i-2/3}^{\gamma},\ \overline{V}_{i-2/3} = V_{i-1},\ \overline{W}_{i-2/3} = W_{i-1}. \qquad (74)$$

Along path 2, the Riemann invariant $\psi_5^{2,3,4}$ is equal to λ_2/h_n. Thus λ_2 does not change either magnitude or sign. The λ_1 and λ_5 fields are called genuinely nonlinear and the fields corresponding to $\lambda_{2,3,4}$ are termed linearly degenerate.

In Equations (68), (69) and (70), (71) and (72) and (73) and (74), it is straightforward to decode for *u*, *v* and *w* from *U* , *V* and *W* (in generalized coordinates, *U*, *V* and *W* would be the normalized velocity contravariants). From these definitions, it is possible to write for *u*, *v* and *w*:

$$u = \left[U\left(h_x^{'}\right)^2 - Vh_x^{'}h_y^{'} - Wh_x^{'}h_z^{'}\right]\Big/h_x^{'};$$

$$v = \left[Uh_x^{'}h_y^{'} + V\left(h_x^{'}h_x^{'} + h_z^{'}h_z^{'}\right) - Wh_y^{'}h_z^{'}\right]\Big/h_x^{'};$$

$$w = \left[Uh_x^{'}h_z^{'} - Vh_y^{'}h_z^{'} + W\left(h_x^{'}h_x^{'} + h_y^{'}h_y^{'}\right)\right]\Big/h_x^{'}. \qquad (75)$$

6.4 Paths of integration

In this sublevel, the values of the variables of the intermediate states and the values of the dependent variables at the cells of the computational domain are employed to form the [6] algorithm to the Euler equations. Initially, the net numerical flux vector in the ξ direction is approximated by:

$$F_{i+1/2,j,k} - F_{i-1/2,j,k} \approx \left[\int_{Q_{i-1}}^{Q_i} X\hat{A}dQ + \int_{Q_i}^{Q_{i+1}} (I-X)\hat{A}dQ\right]. \qquad (76)$$

The matrix *X(Q)* and the paths of integration are what define the scheme. The sub-paths of integration were recently defined (curves 1-3 of Fig. 1). The matrix *X(Q)* is defined to be:

$$X(Q) = R(Q)diag\{1/2 + 1/2signal[\lambda_l(Q)]\}R^{-1}(Q), \qquad (77)$$

where *R(Q)* is the right-eigenvector matrix or the matrix of column eigenvectors of the Jacobian matrix in generalized coordinates $\hat{A}$ and $R^{-1}(Q)$ its inverse. Hence,

$$X\hat{A} = Rdiag\left[\max(\lambda_l, 0)\right]R^{-1} = \hat{A}^+$$

and

$$(I-X)\hat{A} = Rdiag\left[\min(\lambda_l, 0)\right]R^{-1} = \hat{A}^-. \qquad (78)$$

The asymmetric character of the scheme is apparent from this definition. The Equation (76) can be more simplified by the partition of the original integration interval through the sub-paths of integration:

$$\left[\int_{Q_{i-1}}^{Q_i} X\hat{A}dQ + \int_{Q_i}^{Q_{i+1}} (I-X)\hat{A}dQ\right] = F^{(n)}(h, Q_{i+1}) - F^{(n)}(h, Q_i) + \int_{i-1}^{i-2/3}\hat{A}^+dQ + \int_{i-2/3}^{i-1/3}\hat{A}^+dQ + \int_{i-1/3}^{i}\hat{A}^+dQ \qquad (79a)$$

$$-\int_{i}^{i+1/3}\hat{A}^{+}dQ-\int_{i+1/3}^{i+2/3}\hat{A}^{+}dQ-\int_{i+2/3}^{i+1}\hat{A}^{+}dQ\,, \quad (79b)$$

with the normal flux to the interface defined by:

$$F^{(n)}(h,Q)=V_{\text{int}}\begin{Bmatrix}\rho Uh_n\\ \rho uUh_n+ph_x\\ \rho vUh_n+ph_y\\ \rho wUh_n+ph_z\\ (e+p)Uh_n\end{Bmatrix}. \quad (80)$$

It is obvious so that the blocks of construction of the [6] scheme are the sub-integrals along the sub-paths connecting all pair of neighbor cells. To the interval between the pair of cells i-1 and i, for example, it is possible to define:

$$D1=\int_{Q_{i-1}}^{Q_{i-2/3}}\hat{A}^{+}dQ,\ D2=\int_{Q_{i-2/3}}^{Q_{i-1/3}}\hat{A}^{+}dQ,\ D3=\int_{Q_{i-1/3}}^{Q_i}\hat{A}^{+}dQ. \quad (81)$$

While the integral formulae in the above equations seem very complex to be evaluated, they simplify considerably and each sub-integral is reduced to the flux difference $F^{(n)}(Q)$ in each mesh cell and in each cell of the intermediate and sonic states. Therefore, it is possible to write:

$$D1=F^{(n)}(h,Q_{i-2/3})-F^{(n)}(h,Q_{i-1}),$$
$$\text{if } \lambda_1(Q_{i-1})>0 \text{ and } \lambda_1(Q_{i-2/3})>0; \quad (82)$$

$$D1=F^{(n)}(h,Q_{i-2/3})-F^{(n)}(h,\overline{Q}_{i-2/3}),$$
$$\text{if } \lambda_1(Q_{i-1})\le 0 \text{ and } \lambda_1(Q_{i-2/3})>0; \quad (83)$$

$$D1=F^{(n)}(h,\overline{Q}_{i-2/3})-F^{(n)}(h,Q_{i-1}),$$
$$\text{if } \lambda_1(Q_{i-1})>0 \text{ and } \lambda_1(Q_{i-2/3})\le 0; \quad (84)$$

$$D1=0,\ \text{if } \lambda_1(Q_{i-1})\le 0 \text{ and } \lambda_1(Q_{i-2/3})\le 0; \quad (85)$$

$$D2=F^{(n)}(h,Q_{i-1/3})-F^{(n)}(h,Q_{i-2/3}),$$
$$\text{if } \lambda_{2,3,4}(Q_{i-1/3})>0; \quad (86)$$

$$D2=0,\ \text{if } \lambda_{2,3,4}(Q_{i-1/3})\le 0; \quad (87)$$

$$D3=F^{(n)}(h,Q_i)-F^{(n)}(h,Q_{i-1/3}),$$
$$\text{if } \lambda_5(Q_{i-1/3})>0 \text{ and } \lambda_5(Q_i)>0; \quad (88)$$

$$D3=F^{(n)}(h,Q_i)-F^{(n)}(h,\overline{Q}_{i-1/3}),$$
$$\text{if } \lambda_5(Q_{i-1/3})\le 0 \text{ and } \lambda_5(Q_i)>0; \quad (89)$$

$$D3=F^{(n)}(h,\overline{Q}_{i-1/3})-F^{(n)}(h,Q_{i-1/3}),$$
$$\text{if } \lambda_5(Q_{i-1/3})>0 \text{ and } \lambda_5(Q_i)\le 0; \quad (90)$$

$$D3=0,\ \text{if } \lambda_5(Q_{i-1/3})\le 0 \text{ and } \lambda_5(Q_i)\le 0. \quad (91)$$

6.5 Chakravarthy and Osher algorithm

The complete algorithm of [6] to update the dependent variables at the (i,j,k) cell of the n temporal level to the next n+1 level can be simply write as a concise sequence of steps.

ξ contribution:

1) Evaluate the dependent variables at the intermediate cells between i-1 and i using Eqs. (65) to (70). The metric terms are calculated at i-1/2 interface (pointing to inside the cell);

2) Using Eqs. (71) to (74), evaluate the sonic cells which appear between i-1 and i (if the eigenvalues change signal). The metric terms are calculated at i-1/2 interface (pointing inside the cell);

3) Evaluate the sub-integrals D1, D2 and D3 between cells i-1 and i using Eqs. (82) to (91).

The fluxes $F^{(n)}$ at the cells of the computational domain and at the intermediate states are evaluated as necessary;

4) Repeat steps 1-3 between cells i and i+1. The metric terms are calculated at the i+1/2 interface (pointing outside the cell);

5) Substitute the sub-integrals and fluxes in Eq. (79) to evaluate the ξ contribution.

η contribution:

6) Repeat steps 1-5 to the cells j-1, j and j+1 and the metric terms calculated at the j-1/2 and j+1/2 interfaces [the formulae to η are obtained substituting areas and volumes at the interfaces (i-1/2,j,k) and (i+1/2,j,k) by areas and volumes at interfaces (i,j-1/2,k) e (i,j+1/2,k)].

ζ contribution:

7) Repeat steps 1-5 to the cells k-1, k and k+1 and the metric terms calculated at the k-1/2 and k+1/2

interfaces [the formulae to ζ are obtained substituting areas and volumes at the interfaces (i-1/2,j,k) and (i+1/2,j,k) by areas and volumes at interfaces (i,j,k-1/2) e (i,j,k+1/2)].

Update:

8) Update the conserved variables using the explicit Euler method to the time march with first order of accuracy:

$$Q_{i,j,k}^{n+1}=Q_{i,j,k}^{n}-\Delta t_{i,j,k}/V_{i,j,k}\left(\int_{Q_{i-1,j,k}}^{Q_{i,j,k}}\hat{A}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i+1,j,k}}\hat{A}^{-}dQ+\int_{Q_{i,j-1,k}}^{Q_{i,j,k}}\hat{B}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i,j+1,k}}\hat{B}^{-}dQ+\int_{Q_{i,j,k-1}}^{Q_{i,j,k}}\hat{C}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i,j,k+1}}\hat{C}^{-}dQ\right), \tag{92}$$

with the terms multiplying $\Delta t_{i,j,k}$ being evaluated in steps 1-6. This version of the algorithm of flux difference splitting of [6] is first order accurate in space. The RHS to an implicit simulation is dictate by:

$$RHS_{i,j,k}^{n}=-\Delta t_{i,j,k}/V_{i,j,k}\left(\int_{Q_{i-1,j,k}}^{Q_{i,j,k}}\hat{A}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i+1,j,k}}\hat{A}^{-}dQ+\int_{Q_{i,j-1,k}}^{Q_{i,j,k}}\hat{B}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i,j+1,k}}\hat{B}^{-}dQ+\int_{Q_{i,j,k-1}}^{Q_{i,j,k}}\hat{C}^{+}dQ+\int_{Q_{i,j,k}}^{Q_{i,j,k+1}}\hat{C}^{-}dQ\right). \tag{93}$$

7 Harten Numerical Algorithm

The [7] algorithm, first order accurate in space, is specified by the determination of the numerical flux vector at (i+1/2,j,k) interface. This scheme uses Equations (6-21) of [3] scheme, also using the Roe average to determine the interface properties. The next step consists in determining the entropy condition. The entropy condition is implemented of the following way:

$$\upsilon_m=\Delta t_{i,j,k}\lambda_m=Z_m$$

and

$$\psi_m=\begin{cases}|Z_m|, & if\ |Z_m|\geq\delta_f\\ 0.5\left(Z_m^2+\delta_f^2\right)/\delta_f, & if\ |Z_m|<\delta_f\end{cases}, \tag{94}$$

with "*m*" varying from 1 to 5 (three-dimensional space), as defined in [3] scheme, and δ_f assuming values between 0.1 and 0.5, being 0.2 the value suggested by [7].

The [7] dissipation function to the (i+1/2,j,k) interface is constructed by the following matrix-vector product:

$$\{D_{Harten}\}_{\rm int}=[R]_{\rm int}\{-\psi\alpha/\Delta t_{i,j,k}\}_{\rm int}. \tag{95}$$

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

$$F_{i+1/2,j,k}^{(m)}=\left(E_{\rm int}^{(m)}h_x+F_{\rm int}^{(m)}h_y+G_{\rm int}^{(m)}h_z\right)V_{\rm int}+0.5D_{Harten}^{(m)}, \tag{96}$$

with $E_{\rm int}^{(m)}$, $F_{\rm int}^{(m)}$ and $G_{\rm int}^{(m)}$ defined according to Eq. (25). The time integration is performed by the time splitting method, described ins Eq. (27-29).

8 Frink, Parikh and Pirzadeh Numerical Algorithm

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

$$F_{i+1/2,j,k}=1/2\left[F(Q_L)+F(Q_R)-\left|\tilde{A}\right|(Q_R-Q_L)\right]_{\rm int}. \tag{97}$$

In this equation, Q_R and Q_L are right and left state variables of the (i+1/2,j,k) 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 Roe's 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 Roe's average, defined by Eqs. (21) and (16), respectively, and the eigenvalue diagonal matrix Λ, the $|\tilde{A}|$ matrix is defined as $|\tilde{A}|=[R]|\Lambda|[R^{-1}]$. The term

$$\left|\tilde{A}\right|(Q_R-Q_L)=[R]|\Lambda|[R^{-1}]\Delta Q \tag{98}$$

in the formula of the numerical flux vector of [3], it can be rewritten in terms of three flux components, each one associated with a distinct eigenvalue, and the dissipation function of the [8] scheme is defined by:

$$D_{FPP}=[R]|\Lambda|[R^{-1}]\Delta Q=|\Delta F_1|+|\Delta F_4|+|\Delta F_5|, \tag{99}$$

where:

$$\left|\Delta F_{\text{int}}\right| = \left|\psi_{\text{int}}\right| \left\{ \left(\Delta\rho - \frac{\Delta p}{a_{\text{int}}^2}\right) \begin{bmatrix} 1 \\ u_{\text{int}l} \\ v_{\text{int}} \\ w_{\text{int}} \\ \dfrac{u_{\text{int}}^2 + v_{\text{int}}^2 + w_{\text{int}}^2}{2} \end{bmatrix} + \rho_{\text{int}} \begin{bmatrix} 0 \\ \Delta u - n_x \Delta U_{\text{int}} \\ \Delta v - n_y \Delta U_{\text{int}} \\ \Delta w - n_z \Delta U_{\text{int}} \\ u_{\text{int}}\Delta u + v_{\text{int}}\Delta v + w_{\text{int}}\Delta w - U_{\text{int}}\Delta U_{\text{int}} \end{bmatrix} \right\}; \quad (100)$$

$$\left|\Delta F_{4,5}\right| = \left|\psi_{4,5}\right| \left(\frac{\Delta p \pm \rho_l a_l \Delta U_{\text{int}}}{2a_{\text{int}}^2} \right) \begin{bmatrix} 1 \\ u_{\text{int}} \pm n_x a_{\text{int}} \\ v_{\text{int}} \pm n_y a_{\text{int}} \\ w_{\text{int}} \pm n_z a_{\text{int}} \\ H_{\text{int}} \pm U_{\text{int}} a_{\text{int}} \end{bmatrix}, \quad (101)$$

with $U_{\text{int}} = u_{\text{int}} n_x + v_{\text{int}} n_y + w_{\text{int}} n_z$, $\Delta U_{\text{int}} = n_x \Delta u + n_y \Delta v + n_z \Delta w$ and $\Delta(\cdot) = (\cdot)_{i+1,j,k} - (\cdot)_{i,j,k}$.

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 $\lambda_1 = U_{\text{int}}$, $\lambda_4 = U_{\text{int}} + a_{\text{int}}$ and $\lambda_5 = U_{\text{int}} - a_{\text{int}}$ as follows:

$$\psi_l = \begin{cases} \left|Z_l\right|, & if\ \left|Z_l\right| \ge \varepsilon \\ 0.5\left(Z_l^2 + \varepsilon^2\right)/\varepsilon, & if\ \left|Z_l\right| < \varepsilon \end{cases}, \text{ with: } Z_l = \lambda_l, \quad (102)$$

where l assumes values 1, 4 and 5 and ε is a parameter which assumes the value 0.01, recommended by the present author. In the original work of [8], 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 (i+1/2,j,k) interface is determined by:

$$F_{i+1/2,j,k}^{(m)} = \left(E_{\text{int}}^{(m)} n_x + F_{\text{int}}^{(m)} n_y + G_{\text{int}}^{(m)} n_z - 0.5 D_{FPP}^{(m)}\right) S_{\text{int}}, \quad (103)$$

with $E_{\text{int}}^{(m)}$, $F_{\text{int}}^{(m)}$ and $G_{\text{int}}^{(m)}$ defined according to Eq. (25). The time integration is performed by the Runge-Kutta explicit method, second order accurate, of five stages, described below. The contribution of the convective numerical flux vectors is determined by the $C_{i,j,k}$ vector:

$$C_{i,j,k}^{(m)} = F_{i,j-1/2,k}^{(m)} + F_{i+1/2,j,k}^{(m)} + F_{i,j+1/2,k}^{(m)} + F_{i-1/2,j,k}^{(m)} + F_{i,j,k+1/2}^{(m)} + F_{i,j,k-1/2}^{(m)}. \quad (104)$$

The Runge-Kutta can be represented of generalized form by:

$$\begin{aligned} Q_{i,j,k}^{(0)} &= Q_{i,j,k}^{(n)} \\ Q_{i,j,k}^{(m)} &= Q_{i,j,k}^{(0)} - \alpha_m\, \Delta t_{i,j,k} / V_{i,j,k} \times C\left(Q_{i,j,k}^{(m-1)}\right), \quad (105) \\ Q_{i,j,k}^{(n+1)} &= Q_{i,j,k}^{(m)} \end{aligned}$$

with “m” = 1,...,5; $\alpha_1 = 1/4$, $\alpha_2 = 1/6$, $\alpha_3 = 3/8$, $\alpha_4 = 1/2$ and $\alpha_5 = 1$. The [8] scheme implemented in this work is first order accurate in space. The RHS of this scheme is given by:

$$RHS_{i,j,k}^n = -\Delta t_{i,j,k} / V_{i,j,k} C_{i,j,k}^n. \quad (106)$$

9 Liou and Steffen Jr. Numerical Algorithm

The [9] algorithm is specified by the determination of the numerical flux vector at the (i+1/2,j,k) interface. This scheme employs Equations (49-56) to determine interface properties. The next step consists in determining the dissipative term ϕ. The definition of this term determines the particular formulation of the convective fluxes. According to [10], the choice below corresponds to the [9] scheme:

$$\phi_{\text{int}} = \phi_{\text{int}}^{LS}, \text{ with: } \phi_{\text{int}}^{LS} = \left|M_{\text{int}}\right|. \quad (107)$$

The explicit time integration employs the time splitting method described by Eqs. (27-29). The implicit time integration is performed with the definition of the RHS. To this scheme, the RHS is determined by Eq. (58), with the definition of Eq. (107) to the dissipative term. The [9] scheme presented in this work is first order accurate in space.

10 Radespiel and Kroll Numerical Algorithm

The [10] algorithm is specified by the determination of the numerical flux vector at the (i+1/2,j,k) interface. This scheme employs Equations (49-56) to determine interface properties. The next step consists in determining the dissipative term ϕ. The definition of this term determines the particular formulation of the convective fluxes. A hybrid scheme is proposed by [10], which combines the [5] scheme and the [9] (AUSM) scheme. Hence,

$$\phi_{\text{int}} = (1-\omega)\phi_{\text{int}}^{VL} + \omega\phi_{\text{int}}^{LS}, \qquad (108)$$

with:

$$\phi_{\text{int}}^{VL} = \begin{cases} |M_{\text{int}}|, & if\ |M_{\text{int}}| \geq 1; \\ |M_{\text{int}}| + \frac{1}{2}(M_R - 1)^2, & if\ 0 \leq M_{\text{int}} < 1; \\ |M_{\text{int}}| + \frac{1}{2}(M_L + 1)^2, & if\ -1 < M_{\text{int}} \leq 0; \end{cases} \qquad (109)$$

$$\phi_{\text{int}}^{LS} = \begin{cases} |M_{\text{int}}|, & if\ |M_{\text{int}}| \geq \tilde{\delta} \\ \dfrac{(M_{\text{int}})^2 + \tilde{\delta}^2}{2\tilde{\delta}}, & if\ |M_{\text{int}}| < \tilde{\delta} \end{cases}, \qquad (110)$$

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 explicit time integration is described by Eqs. (27-29). The implicit time integration is defined by Eq. (58), using Eq. (108). This scheme is first order accurate in space.

11 Implicit Formulations

All implicit schemes studied in this work used an ADI formulation to solve the algebraic nonlinear system of equations. In these cases, the nonlinear system of equations is linearized considering the implicit operator evaluated at the time "*n*" and, posteriorly, the heptadiagonal system of linear algebraic equations is factored in three tridiagonal systems of linear algebraic equations, each one associated with a particular spatial direction. Thomas algorithm is employed to solve these three tridiagonal systems. All the implicit schemes studied in this work were only applicable to the solution of the Euler equations, which implies that only the convective contributions were considered in the RHS operator.

11.1 Implicit Formulation to the Flux Difference Splitting Schemes

The ADI form to the [3, 6-8] first order schemes is defined by the following three step algorithm:

$$\left\{I + \Delta t_{i,j,k}\Delta_\xi^- K_{i+1/2,j,k}^+ + \Delta t_{i,,kj}\Delta_\xi^+ K_{i+1/2,j,k}^-\right\}\Delta Q_{i,j,k}^* = \left[RHS_{(FDS)}\right]_{i,j,k}^n,$$

to the ξ direction; (111)

$$\left\{I + \Delta t_{i,j,k}\Delta_\eta^- J_{i,j+1/2,k}^+ + \Delta t_{i,j,k}\Delta_\eta^+ J_{i,j+1/2,k}^-\right\}\Delta Q_{i,j,k}^{**} = \Delta Q_{i,j,k}^*,$$

to the η direction; (112)

$$\left\{I + \Delta t_{i,j,k}\Delta_\zeta^- L_{i,j,k+1/2}^+ + \Delta t_{i,j,k}\Delta_\zeta^+ L_{i,j,k+1/2}^-\right\}\Delta Q_{i,j,k}^{n+1} = \Delta Q_{i,j,k}^{**},$$

to the ζ direction; (113)

$$Q_{i,j,k}^{n+1} = Q_{i,j,k}^n + \Delta Q_{i,j,k}^{n+1}, \qquad (114)$$

where:

$$K_{i\pm1/2,j,k}^{\pm} = [R]_{i\pm1/2,j,k}^n \Omega_{i\pm1/2,j,k}^{\pm}\left[R^{-1}\right]_{i\pm1/2,j,k}^n;$$
$$J_{i,j\pm1/2,k}^{\pm} = [R]_{i,j\pm1/2,k}^n \Phi_{i,j\pm1/2,k}^{\pm}\left[R^{-1}\right]_{i,j\pm1/2,k}^n;$$
$$L_{i,j,k\pm1/2}^{\pm} = [R]_{i,j,k\pm1/2}^n \Psi_{i,j,k\pm1/2}^{\pm}\left[R^{-1}\right]_{i,j,k\pm1/2}^n; \qquad (115)$$

$$\Omega_{i\pm1/2,j,k}^{\pm} = diag\left[\left(\lambda_\xi^l\right)^{\pm}\right]_{i\pm1/2,j,k}^n;$$
$$\Phi_{i,j\pm1/2,k}^{\pm} = diag\left[\left(\lambda_\eta^l\right)^{\pm}\right]_{i,j\pm1/2,k}^n;$$
$$\Psi_{i,j,k\pm1/2}^{\pm} = diag\left[\left(\lambda_\zeta^l\right)^{\pm}\right]_{i,j,k\pm1/2}^n; \qquad (116)$$

$$\left(\lambda_\xi^l\right)^{\pm} = 0.5\left(\lambda_\xi^l \pm \left|\lambda_\xi^l\right|\right),\ \left(\lambda_\eta^l\right)^{\pm} = 0.5\left(\lambda_\eta^l \pm \left|\lambda_\eta^l\right|\right);$$
$$\left(\lambda_\zeta^l\right)^{\pm} = 0.5\left(\lambda_\zeta^l \pm \left|\lambda_\zeta^l\right|\right); \qquad (117)$$

$$\Delta_\xi^- = (\cdot)_{i,j,k} - (\cdot)_{i-1,j,k},\ \Delta_\xi^+ = (\cdot)_{i+1,j,k} - (\cdot)_{i,j,k}; \qquad (118)$$

$$\Delta_\eta^- = (\cdot)_{i,j,k} - (\cdot)_{i,j-1,k},\ \Delta_\eta^+ = (\cdot)_{i,j+1,k} - (\cdot)_{i,j,k}. \qquad (120)$$

$$\Delta_\zeta^- = (\cdot)_{i,j,k} - (\cdot)_{i,j,k-1},\ \Delta_\zeta^+ = (\cdot)_{i,j,k+1} - (\cdot)_{i,j,k}. \qquad (121)$$

In Equation (115), the R matrix is defined by Eq. (21); *diag*[·] is a diagonal matrix; in Eqs. (116-117), "*l*" assumes values from 1 to 5 (three-dimensional space) and λ's are the eigenvalues of the Euler equations, defined by Eq. (11). The matrix R^{-1} is defined by Eq. (16). The $RHS_{(FDS)}$ is defined as the residual of the flux difference splitting schemes, which is defined, for instance, by Eq. (26). The other schemes follow similar formulae.

This implementation is first order accurate in time due to the definition of Ω, of Φ and of Ψ, as reported in [23], and is first order accurate in space due to the RHS of the numerical schemes.

11.2 Implicit Formulation to the Flux Vector Splitting Schemes

The ADI form to the [4-5, 9-10] first order schemes is defined by the following three step algorithm:

$$\left\{I + \Delta t_{i,j,k}\Delta_\xi^- A_{i+1/2,j,k}^+ + \Delta t_{i,,kj}\Delta_\xi^+ A_{i+1/2,j,k}^-\right\}\Delta Q_{i,j,k}^* = \left[RHS_{(FVS)}\right]_{i,j,k}^n,$$

to the ξ direction; (122)

$$\left\{I+\Delta t_{i,j,k}\Delta_\eta^- B^+_{i,j+1/2,k}+\Delta t_{i,j,k}\Delta_\eta^+ B^-_{i,j+1/2,k}\right\}\Delta Q^{**}_{i,j,k}=\Delta Q^{*}_{i,j,k},$$

to the η direction; (123)

$$\left\{I+\Delta t_{i,j,k}\Delta_\zeta^- C^+_{i,j,k+1/2}+\Delta t_{i,j,k}\Delta_\zeta^+ C^-_{i,j,k+1/2}\right\}\Delta Q^{n+1}_{i,j,k}=\Delta Q^{**}_{i,j,k},$$

to the ζ direction; (124)

$$Q^{n+1}_{i,j,k}=Q^{n}_{i,j,k}+\Delta Q^{n+1}_{i,j,k}, \qquad (125)$$

where the matrices $A^\pm$, $B^\pm$ and $C^\pm$ are defined as:

$$A^{\pm}_{i\pm1/2,j,k}=[T]^n_{i\pm1/2,j,k}\,\Omega^{\pm}_{i\pm1/2,j,k}\left[T^{-1}\right]^n_{i\pm1/2,j,k};$$
$$B^{\pm}_{i,j\pm1/2,k}=[T]^n_{i,j\pm1/2,k}\,\Phi^{\pm}_{i,j\pm1/2,k}\left[T^{-1}\right]^n_{i,j\pm1/2,k};$$
$$C^{\pm}_{i,j,k\pm1/2}=[T]^n_{i,j,k\pm1/2}\,\Psi^{\pm}_{i,j,k\pm1/2}\left[T^{-1}\right]^n_{i,j,k\pm1/2}; \qquad (126)$$

$$\Omega^{\pm}_{i\pm1/2,j,k}=diag\left[\left(\lambda^l_\xi\right)^{\pm}\right]^n_{i\pm1/2,j,k};$$
$$\Phi^{\pm}_{i,j\pm1/2,k}=diag\left[\left(\lambda^l_\eta\right)^{\pm}\right]^n_{i,j\pm1/2,k};$$
$$\Psi^{\pm}_{i,j,k\pm1/2}=diag\left[\left(\lambda^l_\zeta\right)^{\pm}\right]^n_{i,j,k\pm1/2}, \qquad (127)$$

with the similarity transformation matrices defined by:

$$T=\begin{bmatrix} h'_x & h'_y & h'_z & \alpha & \alpha \\ h'_x u_{int} & h'_y u_{int}-h'_z\rho_{int} & h'_z u_{int}+h'_y\rho_{int} & \alpha\left(u_{int}+h'_x a_{int}\right) & \alpha\left(u_{int}-h'_x a_{int}\right) \\ h'_x v_{int}+h'_z\rho_{int} & h'_y v_{int} & h'_z v_{int}-h'_x\rho_{int} & \alpha\left(v_{int}+h'_y a_{int}\right) & \alpha\left(v_{int}-h'_y a_{int}\right) \\ h'_x w_{int}-h'_y\rho_{int} & h'_y w_{int}+h'_x\rho_{int} & h'_z w_{int} & \alpha\left(w_{int}+h'_z a_{int}\right) & \alpha\left(w_{int}-h'_z a_{int}\right) \\ \frac{h'_x\phi^2}{\gamma-1}+\rho_{int}\left(h'_z v_{int}-h'_y w_{int}\right) & \frac{h'_y\phi^2}{\gamma-1}+\rho_{int}\left(h'_x w_{int}-h'_z u_{int}\right) & \frac{h'_z\phi^2}{\gamma-1}+\rho_{int}\left(h'_y u_{int}-h'_x v_{int}\right) & \alpha\left[\frac{\left(\phi^2+a^2_{int}\right)}{\gamma-1}+a_{int}\tilde{\theta}\right] & \alpha\left[\frac{\left(\phi^2+a^2_{int}\right)}{\gamma-1}-a_{int}\tilde{\theta}\right] \end{bmatrix}; \qquad (128)$$

$$\alpha=\rho_{int}/\left(\sqrt{2}a_{int}\right),\ \beta=1/\left(\sqrt{2}\rho_{int}a_{int}\right);$$
$$\phi^2=(\gamma-1)\frac{u^2_{int}+v^2_{int}+w^2_{int}}{2};$$
$$\tilde{\theta}=h'_x u_{int}+h'_y v_{int}+h'_z w_{int}; \qquad (129)$$

$$T^{-1}=\begin{bmatrix} h'_x\left(1-\frac{\phi^2}{\gamma-1}\right)-\frac{\left(h'_z v_{int}-h'_y w_{int}\right)}{\rho_{int}} & h'_x\frac{(\gamma-1)u_{int}}{a^2_{int}} & \frac{h'_z}{\rho_{int}}+h'_x\frac{(\gamma-1)v_{int}}{a^2_{int}} & -\frac{h'_y}{\rho_{int}}+h'_x\frac{(\gamma-1)w_{int}}{a^2_{int}} & -h'_x\frac{(\gamma-1)}{a^2_{int}} \\ h'_y\left(1-\frac{\phi^2}{\gamma-1}\right)-\frac{\left(h'_x w_{int}-h'_z u_{int}\right)}{\rho_{int}} & -\frac{h'_x}{\rho_{int}}+h'_y\frac{(\gamma-1)u_{int}}{a^2_{int}} & h'_y\frac{(\gamma-1)v_{int}}{a^2} & \frac{h'_x}{\rho_{int}}+h'_y\frac{(\gamma-1)w_{int}}{a^2_{int}} & -h'_y\frac{(\gamma-1)}{a^2_{int}} \\ h'_z\left(1-\frac{\phi^2}{\gamma-1}\right)-\frac{\left(h'_y u_{int}-h'_x v_{int}\right)}{\rho_{int}} & \frac{h'_y}{\rho_{int}}+h'_z\frac{(\gamma-1)u_{int}}{a^2_{int}} & -\frac{h'_x}{\rho_{int}}+h'_z\frac{(\gamma-1)v_{int}}{a^2_{int}} & h'_z\frac{(\gamma-1)w_{int}}{a^2} & -h'_z\frac{(\gamma-1)}{a^2_{int}} \\ \beta\left(\phi^2-a_{int}\tilde{\theta}\right) & \beta\left[h'_x a_{int}-(\gamma-1)u_{int}\right] & \beta\left[h'_y a_{int}-(\gamma-1)v_{int}\right] & \beta\left[h'_z a_{int}-(\gamma-1)w_{int}\right] & \beta(\gamma-1) \\ \beta\left(\phi^2+a_{int}\tilde{\theta}\right) & -\beta\left[h'_x a_{int}+(\gamma-1)u_{int}\right] & -\beta\left[h'_y a_{int}+(\gamma-1)v_{int}\right] & -\beta\left[h'_z a_{int}+(\gamma-1)w_{int}\right] & \beta(\gamma-1) \end{bmatrix}, \qquad (130)$$

with ρ_{int} defined as the interface density. The properties defined at interface are calculated by arithmetical average. The $RHS_{(FVS)}$ is defined as the residual of the flux vector splitting schemes, similar to Eq. (111). This implementation is first order accurate in time.

12 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:

$$\Delta t_{i,j,k}=\frac{CFL(\Delta s)_{i,j,k}}{\left(|q|+a\right)_{i,j,k}}, \qquad (131)$$

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:

$$(\Delta s)_{i,j,k}=\left[MIN\left(l_{MIN},C_{MIN}\right)\right]_i, \qquad (132)$$

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

12 Initial and Boundary Conditions

12.1 Initial Condition

The initial condition adopted for the problems is the freestream flow in all calculation domain ([24-25]). The vector of conserved variables is expressed as follows:

$$Q_\infty = \begin{Bmatrix} 1 \\ M_\infty \cos\theta \\ M_\infty \sin\theta\cos\psi \\ M_\infty \sin\theta\sin\psi \\ \left[\dfrac{1}{\gamma(\gamma-1)} + \dfrac{M_\infty^2}{2}\right] \end{Bmatrix}, \qquad (133)$$

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 special cells named "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. [26] suggests that these procedures lead to the following expressions to the velocity components u, v and w of the ghost cells:

$$u_g = (1-2n_x n_x)u_{real} + (-2n_x n_y)v_{real} + (-2n_x n_z)w_{real}; \quad (134)$$

$$v_g = (-2n_y n_x)u_{real} + (1-2n_y n_y)v_{real} + (-2n_y n_z)w_{real}; \quad (135)$$

$$w_g = (-2n_z n_x)u_{real} + (-2n_z n_y)v_{real} + (1-2n_z n_z)w_{real}. \quad (136)$$

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 external region of the mesh to external flow problems, 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 ([25]). It is necessary to specify four conditions to these four data. [24] 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 ([25]). 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 state equation of 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 stae equation of a perfect gas.

c) Entrance and exit – The entrance and exit boundaries are applied to the ramp and air inlet 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 notebook with processor Intel core i7, 2.20GHz of clock, and 8Gbytes 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 flows along an air inlet and around a cylindrical blunt body. The ramp configuration is described in Fig. 2. An algebraic mesh of 61x60x10 points or composed of 31,860 hexahedrons and 36,600 nodes was used as shown in Fig. 3. The points are equally spaced in both directions. The second configuration is the cylindrical blunt body. It has a nose ratio of 1.0m and the far field is located at twenty times the nose ratio in relation to the body's leading edge. It is composed of 32,922 hexahedrons and 37,800 nodes, which corresponds in finite differences to a 63x60x10 mesh. Figure 4 shows the blunt body configuration and Fig. 5 exhibits the blunt body's mesh.

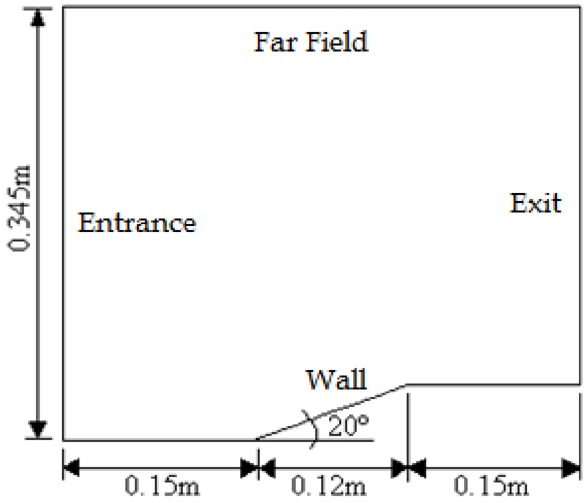

Figure 2. Ramp configuration.

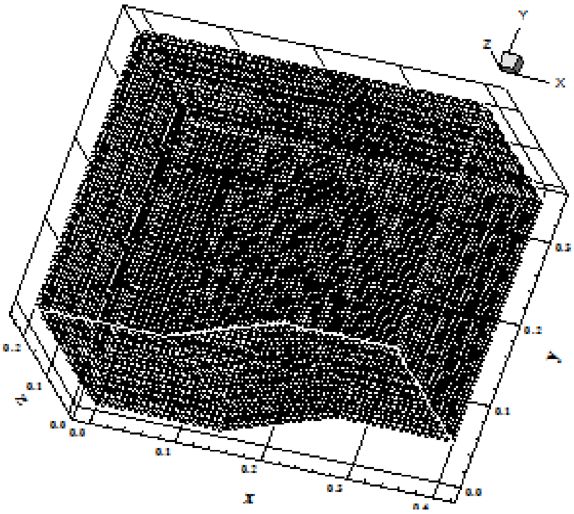

Figure 3. Ramp mesh (61x60x10).

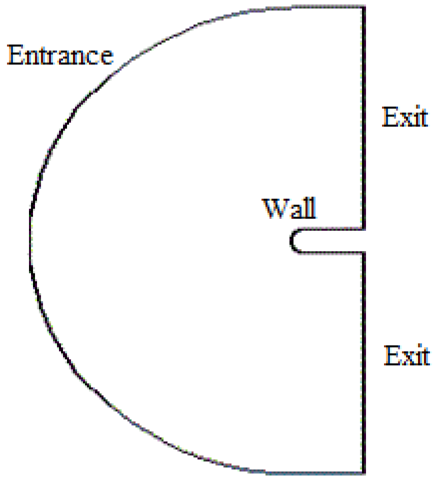

Figure 4. Blunt body configuration.

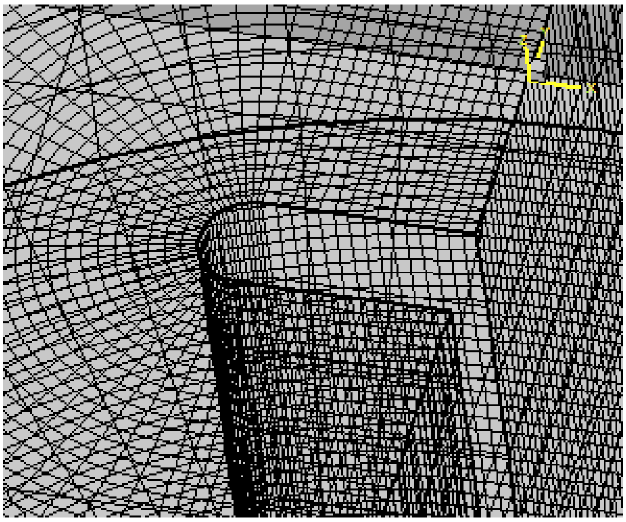

Figure 5. Blunt body mesh.

Finally, the last configuration is the air inlet geometry. It is composed of 20,709 hexahedrons and 24,000 nodes, which corresponds to a mesh of 60x40x10 in finite differences. Figure 6 exhibits the air inlet configuration, whereas Fig. 7 presents the air inlet mesh.

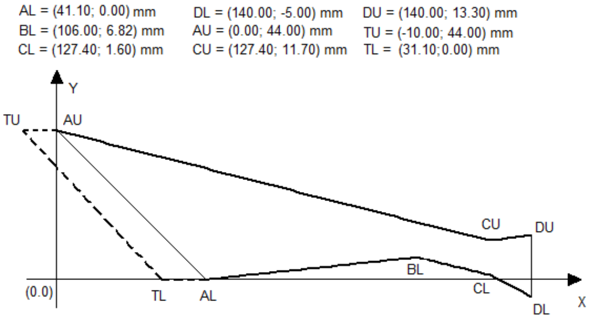

Figure 6. Air inlet configuration.

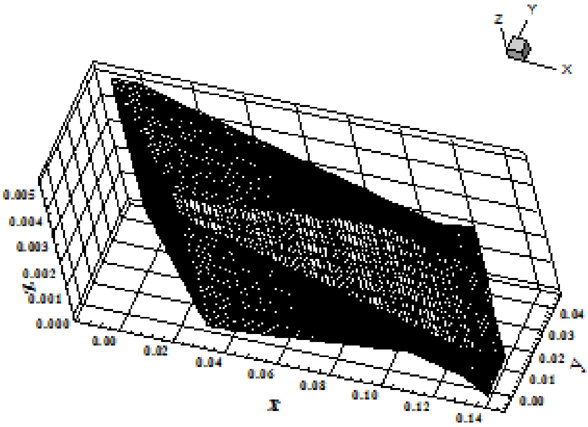

Figure 7. Air inlet mesh.

13.1 Ramp Problem – Implicit Simulation

The ramp problem is described as a low supersonic flow impinging a compression corner, generating an oblique shock wave, and expanding in an expansion corner, as shown in Fig. 2. The freestream Mach number flow, which defines the initial condition, was adopted equal to 2.0 (supersonic flow). All schemes generated converged results to this implicit problem.

Figures (8-15) show the pressure contours obtained by [3-10] numerical algorithms. As can be seen, only the [6] and [9] schemes yielded bad results. The [6] scheme presents a unphysical solution, whereas [9] scheme exhibits a pressure oscillation due to the shock, even being a low Mach number flow. The rest of the solutions presents good behavior, capturing appropriately the shock discontinuity.

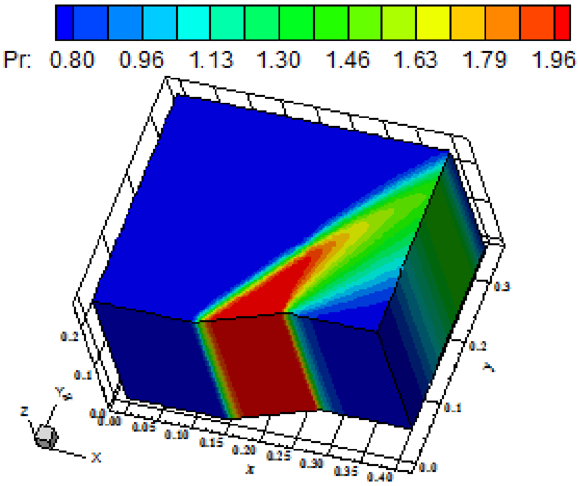

Figure 8. Pressure contours ([3]).

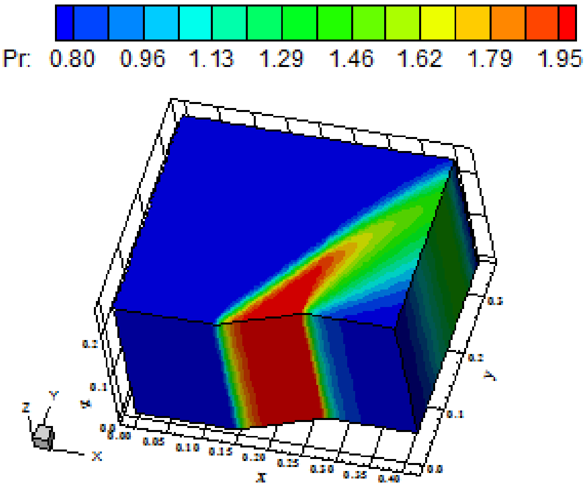

Figure 9. Pressure contours ([4]).

Figures (16-23) show the Mach number contours obtained by all studied schemes. The problems with the [6] and [9] schemes appear again. The [6] solution is an unphysical one and is not due to the implemented algorithm. In other words, the problem is not numerical, but physical. The [6] scheme seems unable to capture appropriately the shock wave.

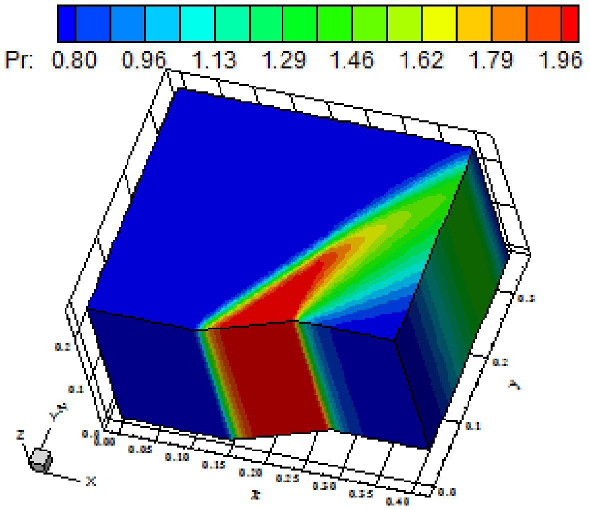

Figure 10. Pressure contours ([5]).

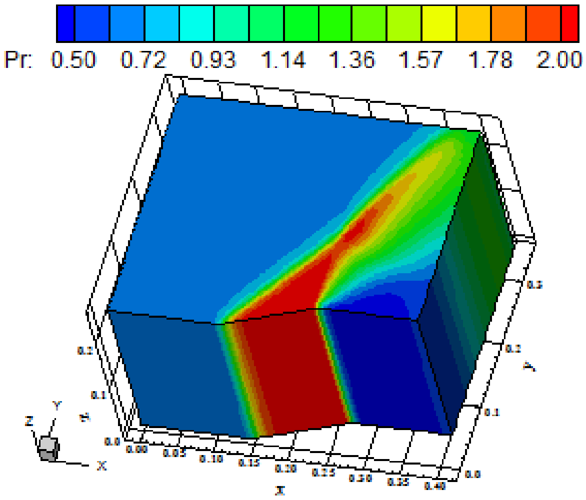

Figure 11. Pressure contours ([6]).

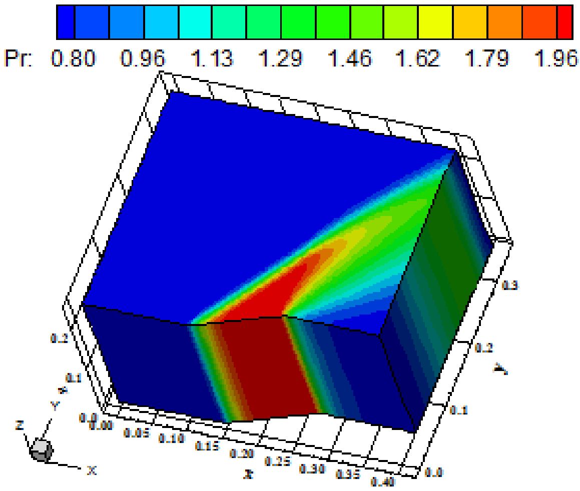

Figure 12. Pressure contours ([7]).

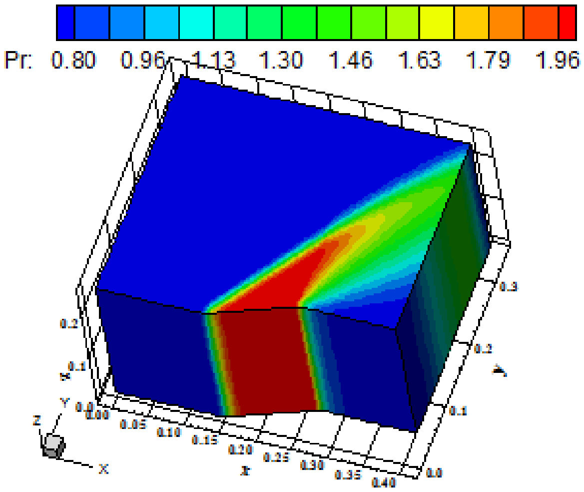

Figure 13. Pressure contours ([8]).

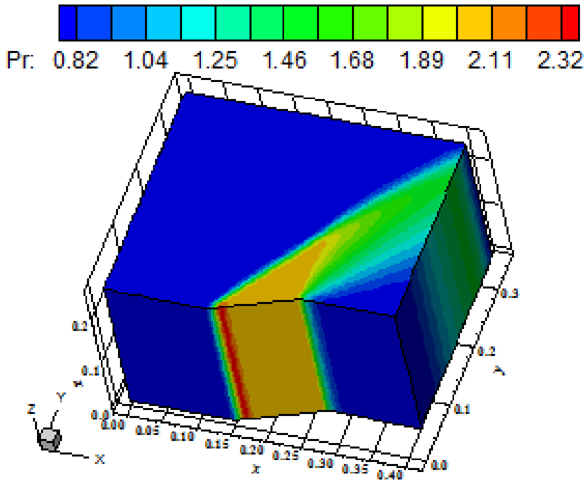

Figure 14. Pressure contours ([9]).

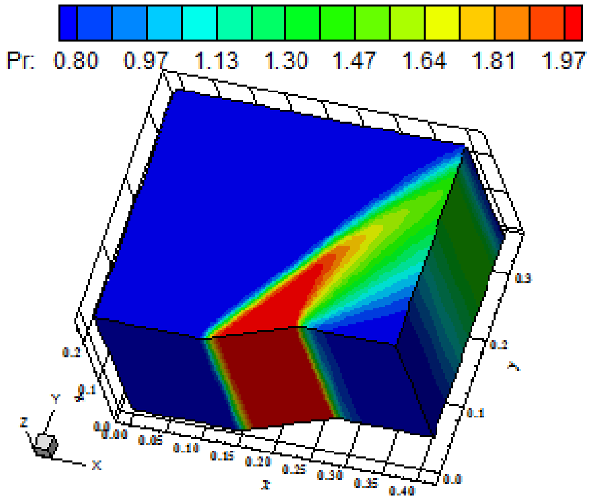

Figure 15. Pressure contours ([10]).

The [9] solution only highlights that the problem with this scheme is the non-capacity of avoid pre-shock oscillations at compression regions. Although the shock profile is captured, the oscillation in the wall pressure distribution becomes such scheme limited in its capacity to predict discontinuities.

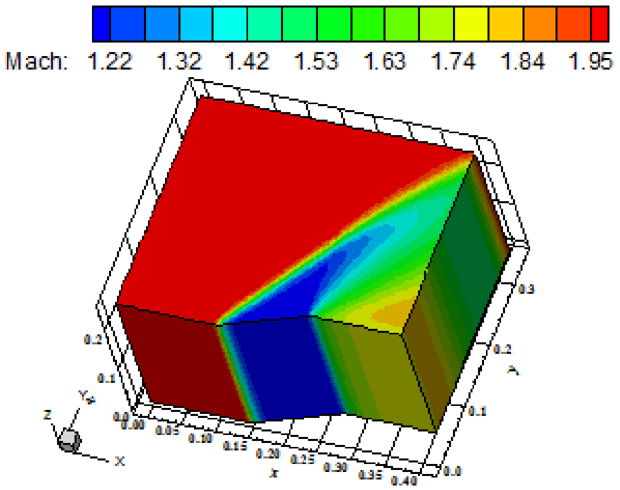

Figure 16. Mach number contours ([3]).

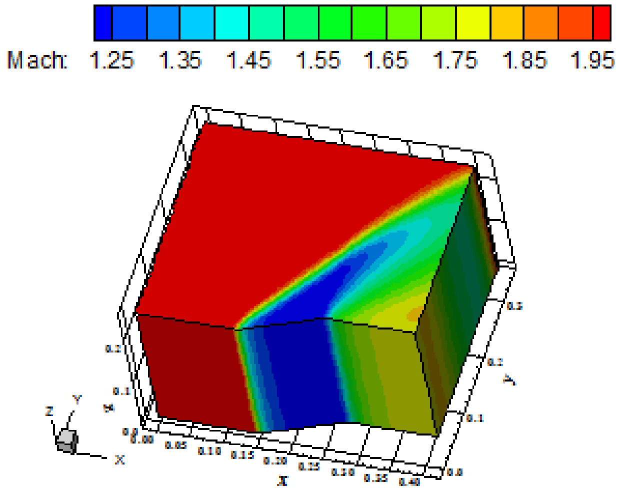

Figure 17. Mach number contours ([4]).

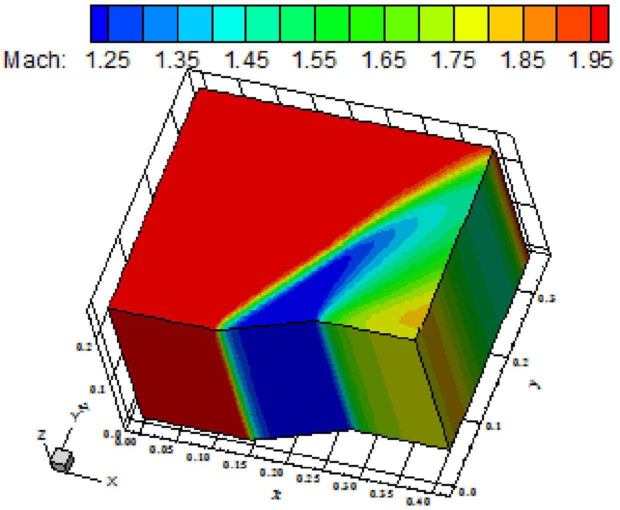

Figure 18. Mach number contours ([5]).

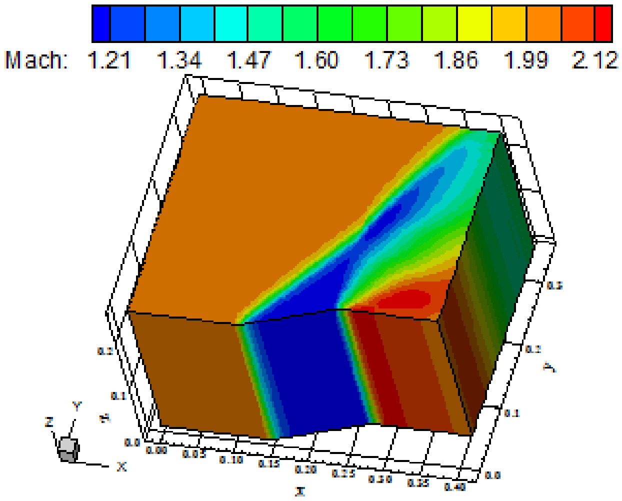

Figure 19. Mach number contours ([6]).

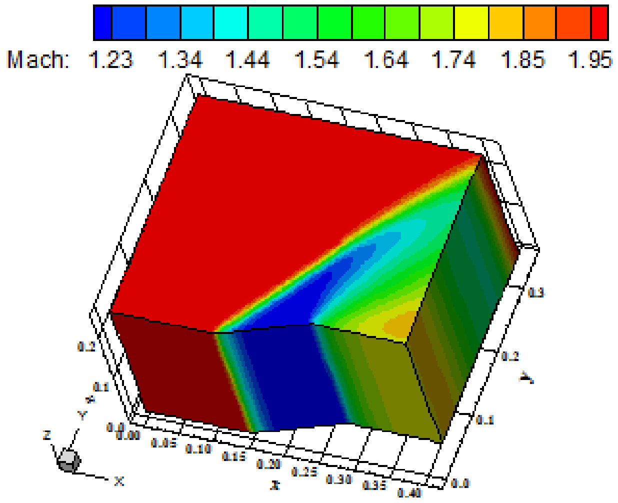

Figure 20. Mach number contours ([7]).

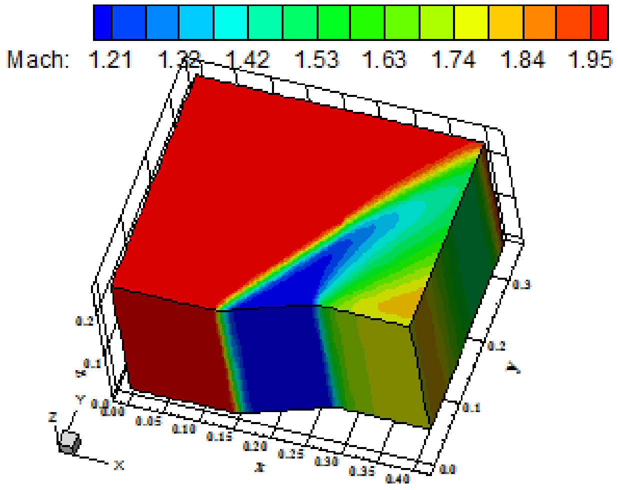

Figure 21. Mach number contours ([8]).

Figures (24-27) exhibit the wall pressure distributions at wall. Figure 24 shows the flux difference splitting solutions, obtained by the [3, 6-

8] algorithms. In this plot, the [3] and [8] schemes have presented the best choice for this type of formulation.

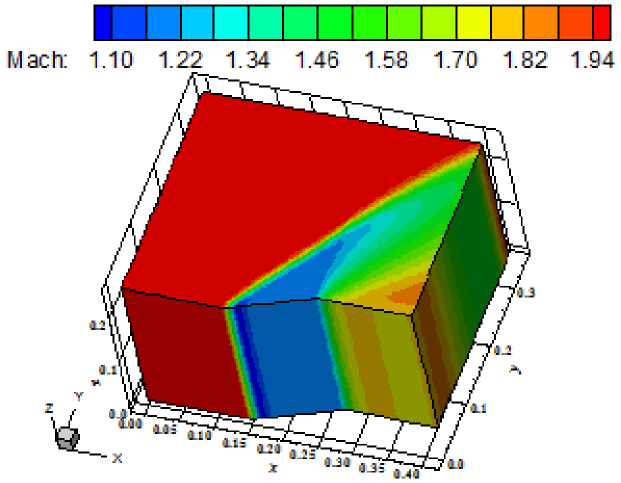

Figure 22. Mach number contours ([9]).

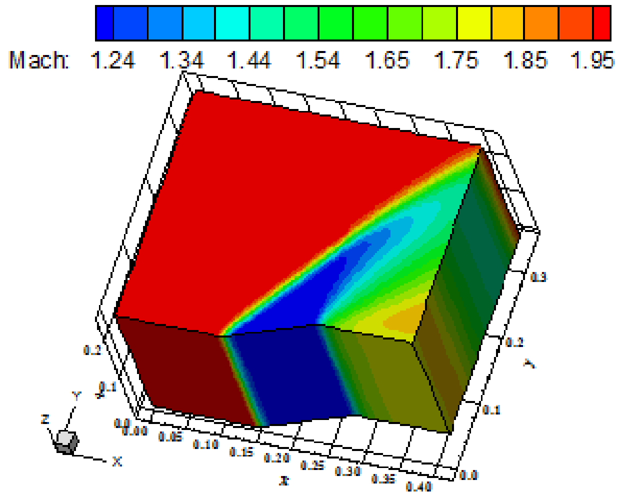

Figure 23. Mach number contours ([10]).

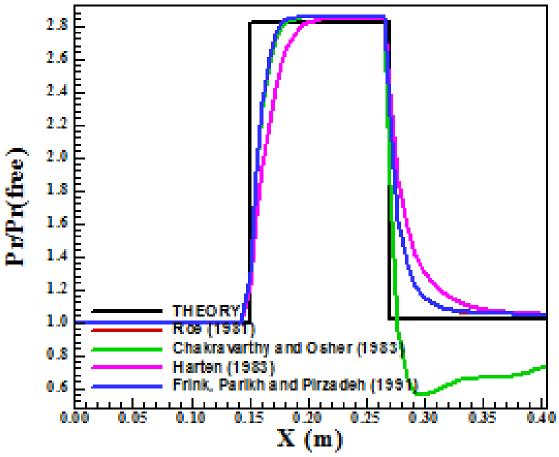

Figure 24. Wall pressure distributions (FDS).

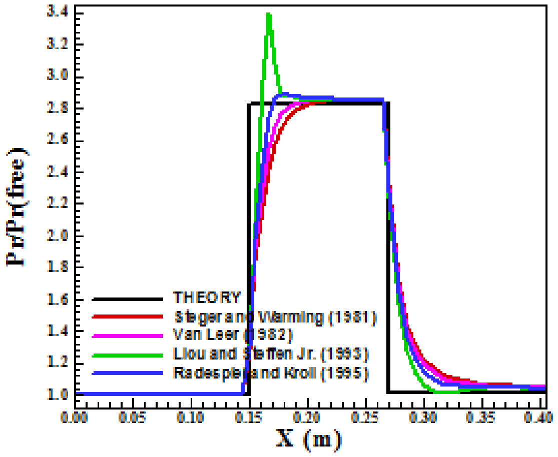

Figure 25. Wall pressure distributions (FVS).

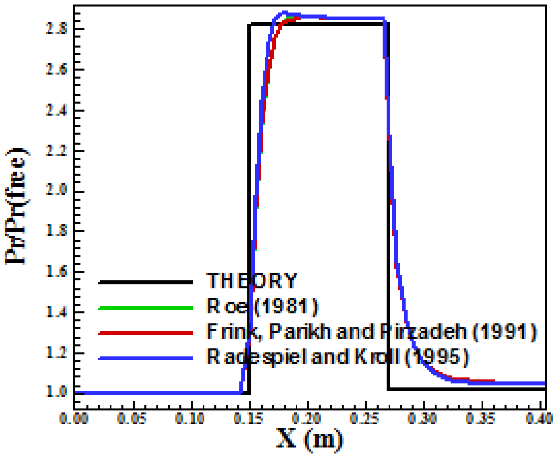

Figure 26. Choosing the best distribution.

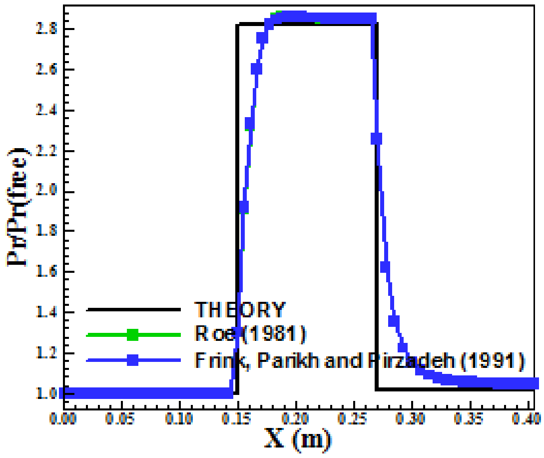

Figure 27. Best profiles.

Figure 25 shows the wall pressure distributions obtained by the flux vector splitting schemes [4-5, 9-10]. The best distribution for this type of formulation was obtained by the [10] scheme.

Figure 26 represents the comparison among the three solutions that were chosen as the best in each formulation. The best global solution was due to [3, 8]. Finally, Figure 27 exhibits the best profiles of pressure with symbols to identify in how many cells the discontinuity is captured. In this study, the best results captured the discontinuity using six (6) cells, which gives us a feeling in how important is the use of high resolution algorithms.

A 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. [27] (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 $\phi = 20°$ (ramp inclination angle) and to a freestream Mach number equals to 2.0, it is possible to obtain from this diagram a value to β equals to 53.0 °. Using a transfer in all pressure contours figures, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 1. As can be noted, the best result is due to [5]. As the best wall pressure distribution was due to [3, 8] and the best shock angle has the [3] algorithm with an error of only 0.94% (less than 1.0%), the [3] algorithm is the best in this problem.

Table 1. Shock angle and percentage errors.

Algorithm	β (º)	Error (%)
[3]	53.5	0.94
[4]	53.4	0.75
[5]	53.0	0.00
[6]	53.9	1.70
[7]	52.9	0.19
[8]	54.0	1.89
[9]	54.0	1.89
[10]	54.3	2.45

13.2 Blunt Body Problem – Explicit Simulation

In this problem, a high "cold gas" hypersonic flow is simulated. A freestream Mach number of 36.0 is studied. The [3, 6, 8] schemes did not yield converged results. The results are compared with normal shock wave theory ones.

Figures (28-32) show the pressure contours obtained by [4-5, 7, 9-10] schemes, respectively. The most severe pressure field is due to [9], which characterizes this scheme as the most conservative.

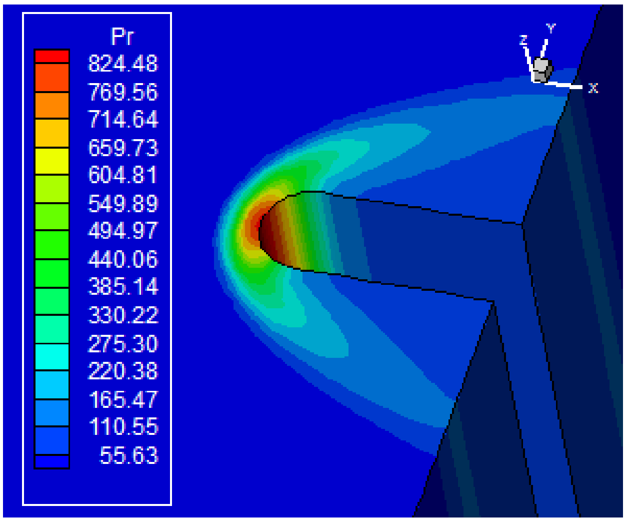

Figure 28. Pressure contours ([4]).

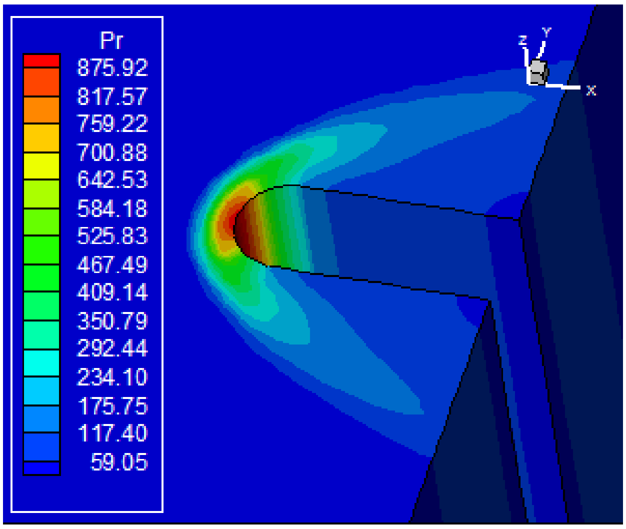

Figure 29. Pressure contours ([5]).

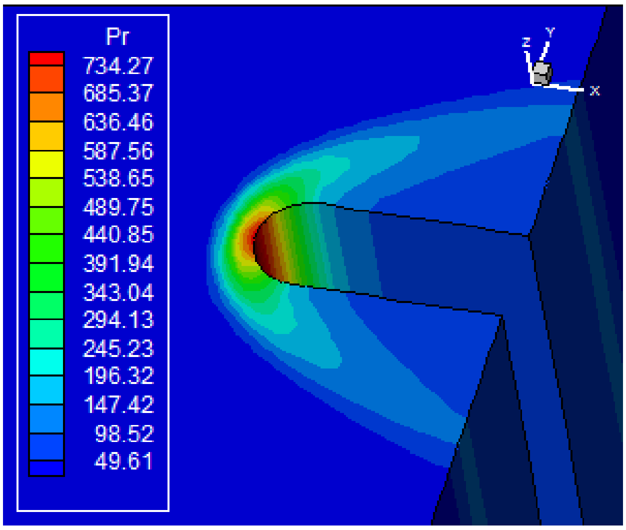

Figure 30. Pressure contours ([7]).

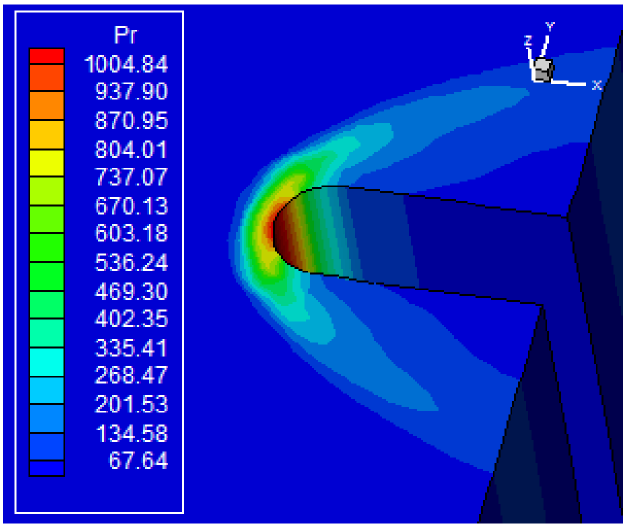

Figure 31. Pressure contours ([9]).

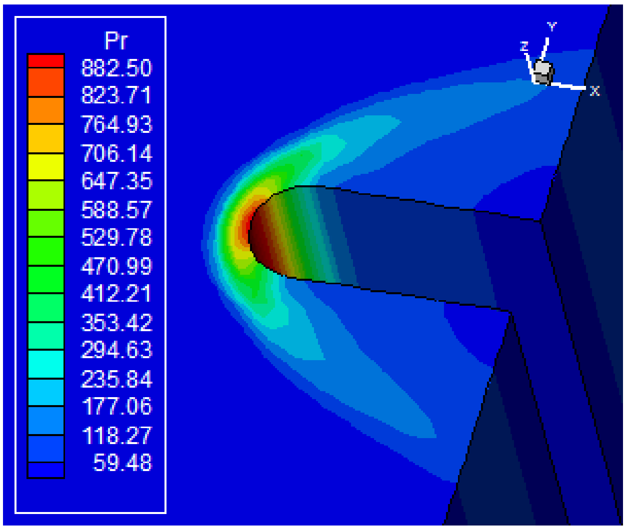

Figure 32. Pressure contours ([10]).

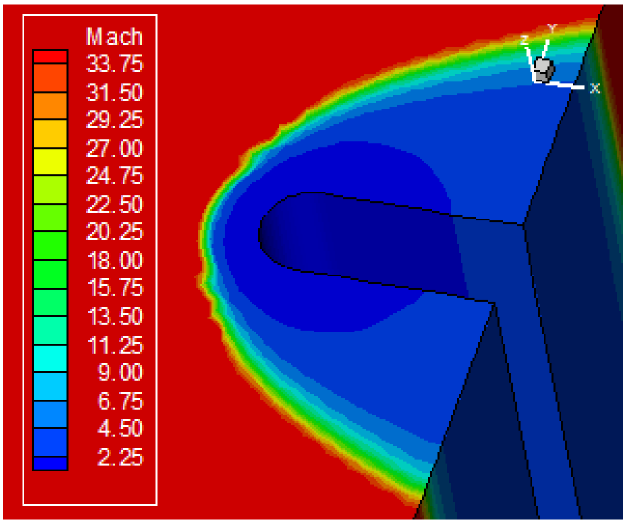

Figure 33. Mach number contours ([4]).

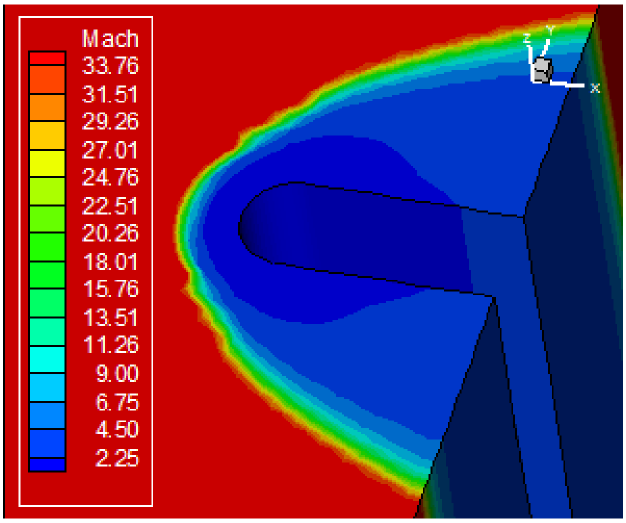

Figure 34. Mach number contours ([5]).

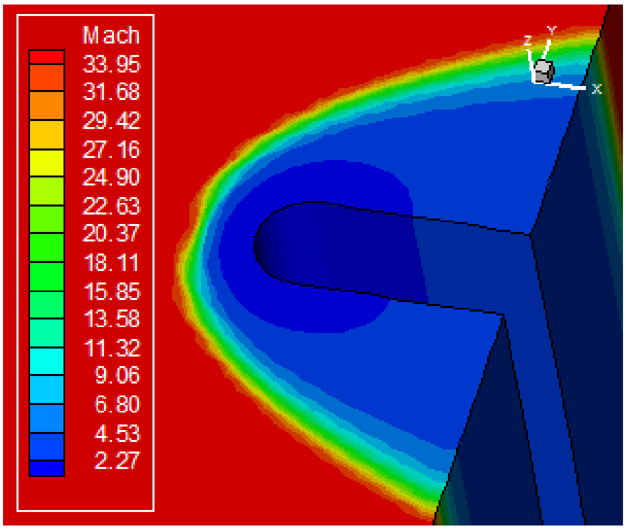

Figure 35. Mach number contours ([7]).

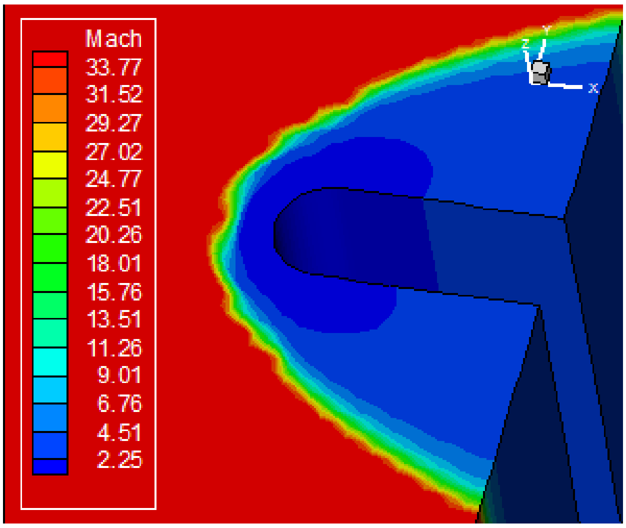

Figure 36. Mach number contours ([9]).

Figures (33-37) exhibit the Mach number contours obtained by the [4-5, 7, 9-10] schemes, respectively. All schemes predict a freestream Mach number inferior to 36.0. The Mach number field generated by the [7] algorithm is the closest in relation to the original freestream flow. All solutions present good symmetry properties. The [5] and [10] results present the biggest extension of the low supersonic region, ahead of the body's configuration.

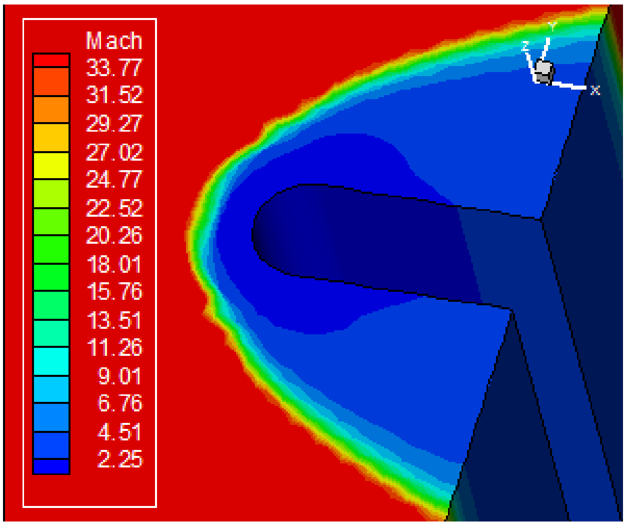

Figure 37. Mach number contours ([10]).

Figure 38 exhibits the –Cp distributions at wall of the blunt body, generated by the five schemes under study in this problem. The maximum Cp was obtained by [9] and reaches the value 1.66.

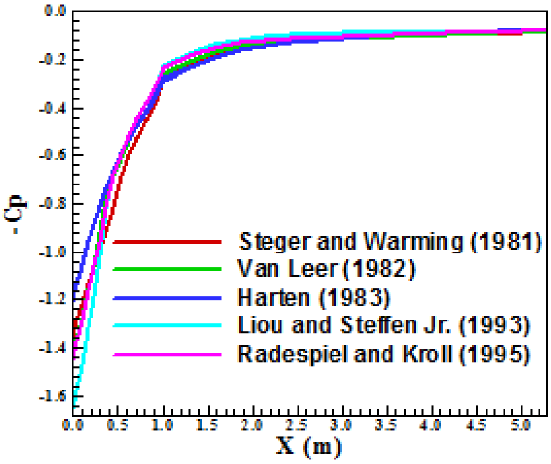

Figure 38. –Cp distributions at wall.

Figure 39 shows the temperature distribution at wall generated by the five schemes. The maximum peak of temperature is obtained by [7] and assumed the value of 70,000K, which is a solution typical of "cold gas" flows and is incorrect in reality. Values of temperature around 20,000K are more realistic and are obtained with a "hot gas" formulation. To more details about "hot gas" formulation, see [21, 28-33].

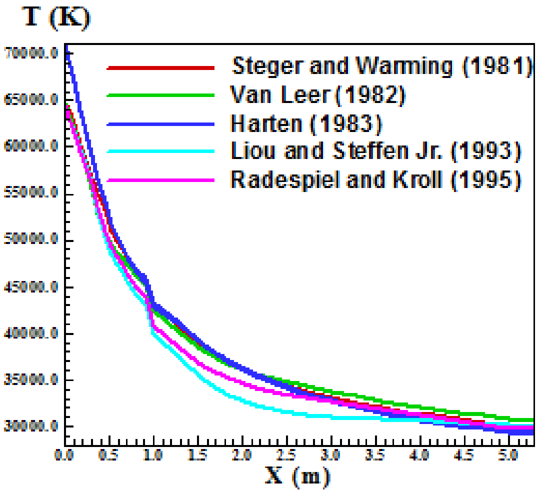

Figure 39. Temperature distributions at wall.

In terms of quantitative results, the present author compared the stagnation pressure at the blunt body nose assuming the perfect gas formulation. To calculate the stagnation pressure ahead of the blunt body, [27] presents in its B Appendix values of the normal shock wave properties ahead of the configuration. The ratio pr_0/pr_∞ is estimated as function of the normal Mach number and the stagnation pressure pr_0 can be determined from this parameter. Hence, to a freestream Mach number of 36.0, the ratio pr_0/pr_∞ assumes the value 1669.0. The value of pr_∞ is 0.714 by the present dimensionless. Using the ratio obtained from [27], the stagnation pressure ahead of the configuration nose is estimated as 1,191.66 unities. Table 2 compares the values obtained from the simulations with this theoretical parameter and presents the numerical percentage errors. As can be observed, all solutions present percentage errors less than 40.0%, which is a reasonable estimation of the stagnation pressure. The best estimation was due to the [9] scheme, first order accurate, with an error of 15.68%.

Table 2. Stagnation pressure and errors.

Algorithm	pr_0	Error (%)
[4]	824.48	30.81
[5]	875.92	26.50
[7]	734.27	38.38
[9]	1,004.84	15.68
[10]	882.50	25.94

13.3 Air Inlet Problem – Explicit Simulation

The freestream Mach number of this high “cold gas” air inlet flow was 20.0. The [3, 6, 8] schemes did not yield converged results.

Figures (40-44) exhibit the pressure contours of this simulation to the five algorithms that were robust enough to perform this study. As can be seen, only the [7] did not produce good results. Moreover, none of the flux difference splitting schemes yielded reasonable results for the two “cold gas” high hypersonic flow simulations, ratifying the expected behavior that the flux vector splitting algorithms are more robust. The pressures contours are well captured by the FVS algorithms, mainly the shock interference at the air inlet throat.

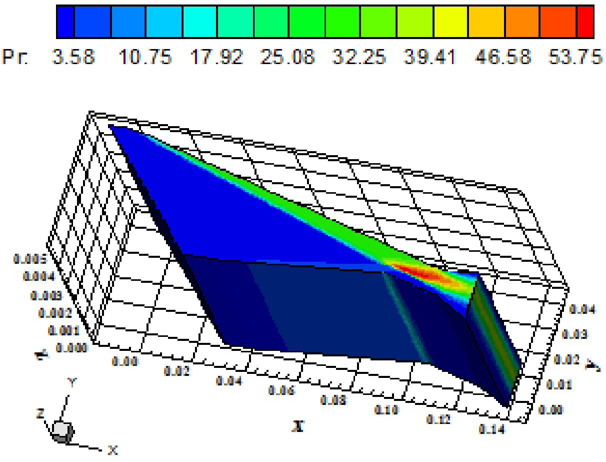

Figure 40. Pressure contours ([4]).

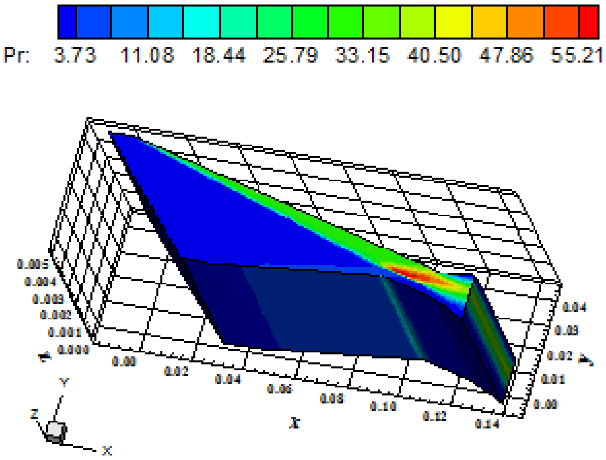

Figure 41. Pressure contours ([5]).

Figures (45-49) present the Mach number contours obtained by the [4-5, 7, 9-10] schemes, respectively. With the exception of the [7] solution, which seems very diffusive, the other solutions are good and capture the shock interference at the throat. Particularly, the [5] and [10] solutions are very close.

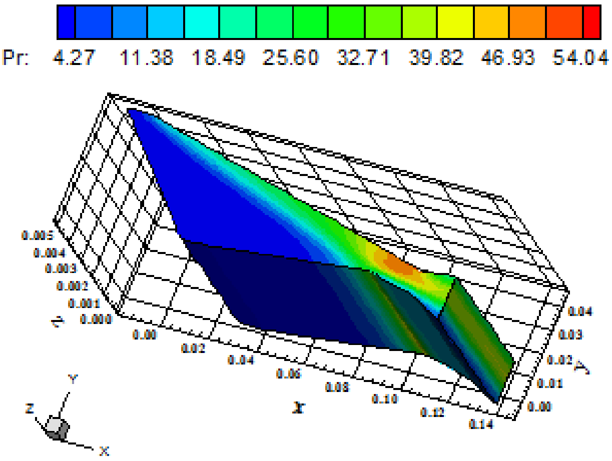

Figure 42. Pressure contours ([7]).

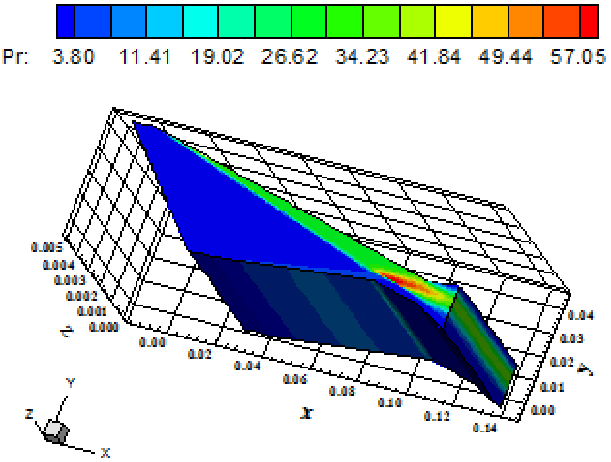

Figure 43. Pressure contours ([9]).

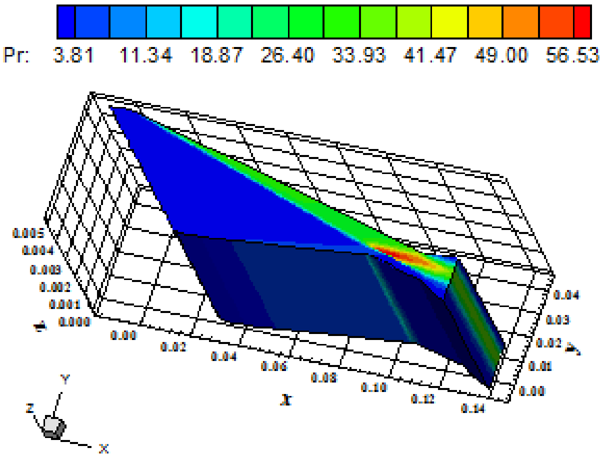

Figure 44. Pressure contours ([10]).

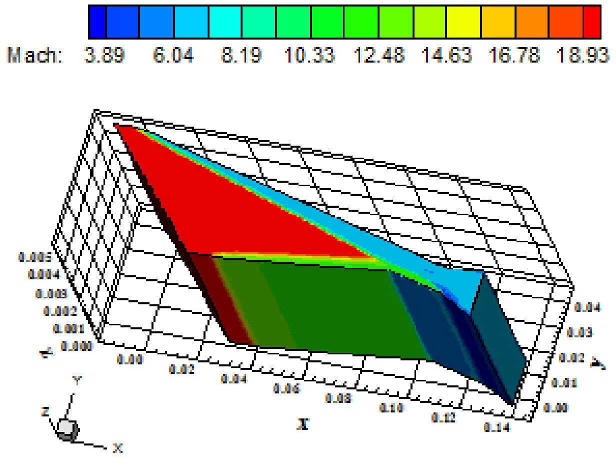

Figure 45. Mach number contours ([4]).

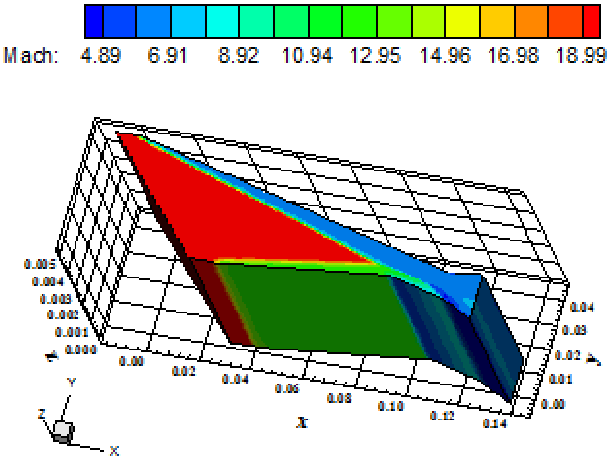

Figure 46. Mach number contours ([5]).

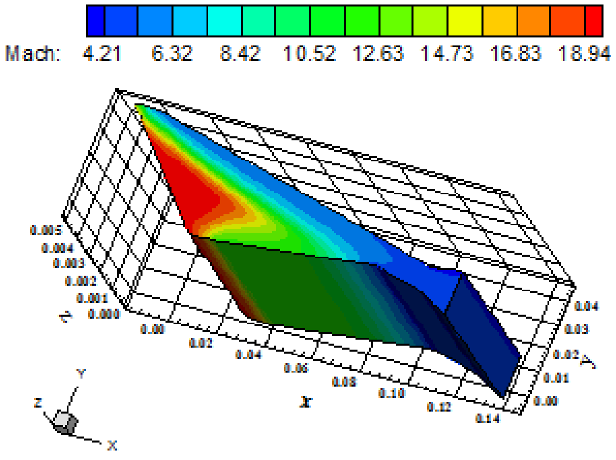

Figure 47. Mach number contours ([7]).

Figure 50 shows the pressure distributions along the air inlet. Only the [7] solution is not plotted because it presents a peak of pressure at the shock interference that is very high to be considered altogether. The best solution is that originated by [10]: it captures the pressure plateau at the beginning region (> 10mm), after that the peak of pressure from the shock interference, the shock and the expansion wave (> 12mm).

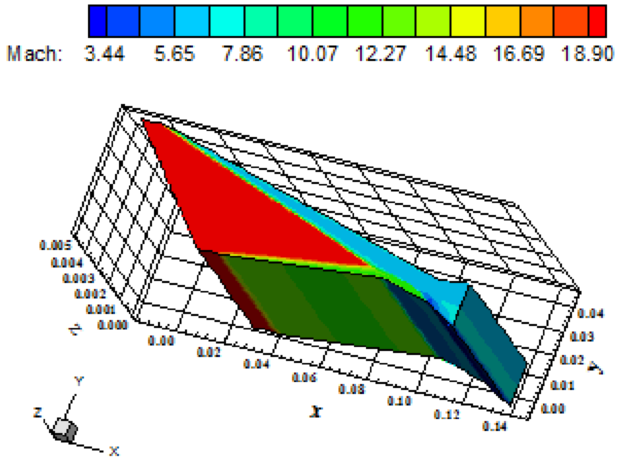

Figure 48. Mach number contours ([9]).

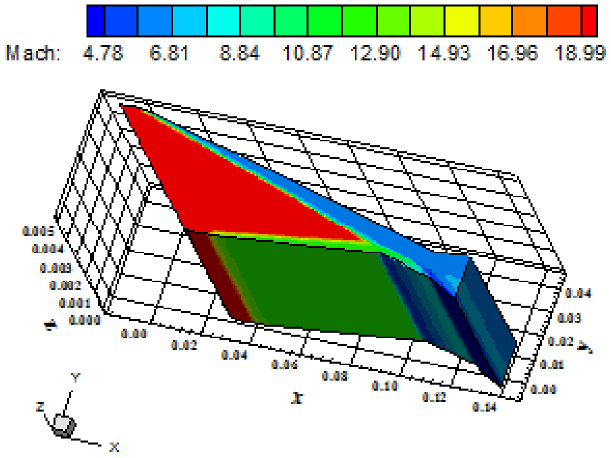

Figure 49. Mach number contours ([10]).

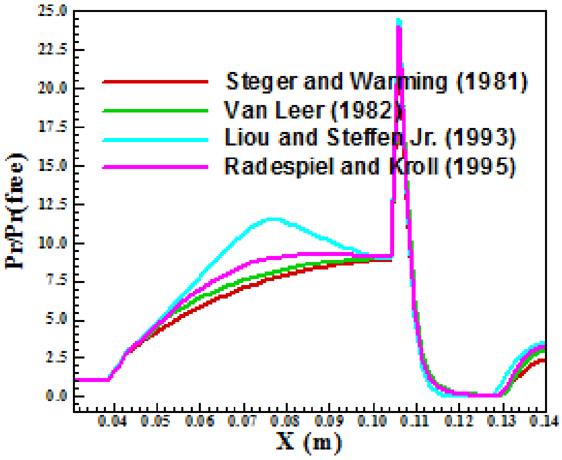

Figure 50. Wall pressure distributions.

A parameter which can be analyzed to evaluate the accuracy of the four schemes, without considering [7], is the shock angle of the oblique shock wave that appear at the lower and upper air inlet walls. With a transfer were measured the inclination angles of the lower and upper ramp of the entrance device. To the lower wall this angle was of 6.5^o in relation to the horizontal and to the upper wall was of 13.5^o in relation to the horizontal. With these angles and with the freestream Mach number was possible to determine the theoretical shock angles of the oblique shock waves. These angles are disposed in Tab. 3, joined with the measured values of them to each scheme and the respective percentage error. Figures (40-44), at the xy plane, were used to evaluate the measured angles.

Table 3. Measured values of the shock angles of the oblique shock waves.

Surface	Scheme	$\beta^{(Theory)}$	$\beta^{(Measured)}$	Error (%)
Lower	[4]	10.0	10.0	0.00
	[5]	10.0	10.0	0.00
	[9]	10.0	9.4	6.00
	[10]	10.0	10.0	0.00
Upper	[4]	17.5	18.8	7.43
	[5]	17.5	18.0	2.86
	[9]	17.5	17.4	0.57
	[10]	17.5	18.0	2.86

As can be seen, the [5, 10] schemes gave the best results considering a global analyses; in other words, the results of the two walls. Although the [9] scheme had determined with excellent accuracy the value of the shock angle of the oblique shock wave at the air inlet upper wall, its behavior at the lower wall was reasonable, with an error of 6.0%. The [5, 10] operators had errors inferiors to 3.0% in both walls and, therefore, presented the best solutions.

13.4 Computational Data

Table 4 presents the computational data of the simulations. All simulations converged in four (4) orders of reduction of the initial residue. It is interesting to note the behavior of the [4] scheme in the air inlet problem. The [5, 7, 9-10] algorithms converged using a CFL of 0.2 at maximum. However, the [4] converged with CFL of 0.9, which indicates that a more strength initial condition (freestream Mach number) could be used. The robustness of the [4] scheme is a remarkable property of this algorithm. For an initial phase of airplanes design, this algorithm is highly recommended by its low cost, simple numerical implementation and good solution quality. Table 4 highlights the excellent performance of the [4] scheme, being the fastest to convergence and using bigger CFL number than the other schemes.

Table 4. Computational Data.

	Ramp		Blunt Body		Air Inlet	
Scheme	**CFL**	**Iter.**	**CFL**	**Iter.**	**CFL**	**Iter.**
[3]	1.5	299	-	-	-	-
[4]	2.5	173	0.9	305	0.9	561
[5]	1.5	291	0.4	819	0.1	5196
[6]	2.0	400	-	-	-	-
[7]	2.0	227	0.5	562	0.1	8192
[8]	1.5	299	-	-	-	-
[9]	1.5	325	0.9	726	0.1	5320
[10]	1.5	303	0.9	440	0.2	2590

14 Conclusions

In the present work, the [3-10] schemes are implemented, on a finite volume context and using an upwind and structured spatial discretization, to solve the Euler equations in the three-dimensional space. The [3, 6-8] schemes are flux difference splitting ones and more accurate solutions are expected. On the other hand, the [4-5, 9-10] are flux vector splitting ones and more robustness properties are expected. The implemented schemes are first order accurate in space. The explicit time integration uses a time splitting method, an Euler method or a Runge-Kutta method. The implicit time integration uses an ADI approximate factorization method. The physical problems of the supersonic flow along a ramp and the high "cold gas" hypersonic flows around a blunt body and along an air inlet are solved. All the eight 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 [18-19].

The results have demonstrated that the [9-10] schemes are the most conservative algorithms among the studied ones and that the [3, 5, 9] schemes are the most accurate. The [9] scheme yielded the most severe pressure field in the blunt body problem, which indicates this one as a more conservative scheme to the prediction of high "cold gas" hypersonic design conditions. The pressure distribution along the ramp was better predicted by [3, 8] schemes. The [3, 8] scheme presented better shock capturing properties due to the use of Roe's average. In the estimation of the angle of the oblique shock wave, only the [5] scheme presented appropriate predictions (error equal to 0.00%). The best algorithm for the ramp problem was a compromise between quality and quantity and resulted in the [3] scheme as the best. In the blunt body problem, the [3, 6, 8] 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 high "cold gas" hypersonic flow. The following comments are related only with the hypersonic case. The most severe pressure field was estimated by the [9] scheme, which also indicates this scheme to more severe design conditions of aerospace vehicles. The stagnation pressure ahead of the configuration is better predicted by the [9] scheme, which indicates it for high hypersonic airplanes design. In the air inlet configuration, only five schemes simulate this problem. However, the [7] scheme did not yield reasonable results and was excluded from such analysis. The more conservative scheme was the [10] one. Moreover, the upper wall pressure distribution was more appropriately described by [10]. In the prediction of the shock angles of the oblique shock waves at the lower and upper walls of the air inlet, the [5, 10] schemes were the best.

As final conclusion, the present author recommends the [5, 9] schemes, among the studied algorithms, to obtain more accurate solutions in the three-dimensional space. The [5] scheme, due to its confirmed robustness and accuracy, could be used in the initial and final design phases of aerospace vehicles.

15 Concluding Remarks

As a final comment is the excellent computational performance of the [4] scheme, as was demonstrated in section 13.4. Although not present comparative accuracy in relation to the [5] scheme, its algorithm has presented significant numerical behavior, also being indicated to the initial phase of airplane design.

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] S. R. Chakravarthy, and S. Osher, Numerical Experiments with the Osher Upwind Scheme for the Euler Equations, *AIAA Journal*, Vol. 21, No. 9, 1983, pp. 1241-1248.

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

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

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

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

[11] 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.

[12] 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.

[13] 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.

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

[15] E. S. G. Maciel, Explicit and Implicit Solutions of First Order Algorithms Applied to the Euler Equations in Two-Dimensions – Part I – Explicit Formulations, *Proceedings of the IX Symposium of Computational Mechanics (IX SIMMEC)*, São João Del Rey, MG, Brazil, 2010. [available in CD-ROM]

[16] E. S. G. Maciel, Explicit and Implicit Solutions of First Order Algorithms Applied to the Euler Equations in Two-Dimensions – Part II – Implicit Formulations and Results, *Proceedings of the IX Symposium of Computational Mechanics (IX SIMMEC)*, São João Del Rey, MG, Brazil, 2010. [available in CD-ROM]

[17] R. W. MacCormack, Current Status of Numerical Solutions of the Navier-Stokes Equations, *AIAA Paper 85-0032*, 1985.

[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. [available in CD-ROM]

[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 Science (XII ENCIT)*, Belo Horizonte, MG, Brazil, 2008. [available in CD-ROM]

[20] E. S. G. Maciel, TVD Algorithms Applied to the Solution of the Euler and Navier-Stokes Equations in Three-Dimensions, *WSEAS Transactions on Mathematics*, Vol. 11, June, Issue 6, 2012, pp. 546-572.

[21] E. S. G. Maciel, and A. P. Pimenta, Reentry Flows in Chemical Non-Equilibrium in Three-Dimensions, *WSEAS Transactions on Mathematics*, Vol. 11, March, Issue 3, pp. 227-256.

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

[23] 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, 1985, pp. 327-360.

[24] 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.

[25] 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.

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

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

[28] E. S. G. Maciel, and A. P. Pimenta, Thermochemical Non-Equilibrium Reentry Flows in Three-Dimensions – Part I – Structured Solutions, Submitted to *WSEAS Transactions on Applied and Theoretical Mechanics* (under review).

[29] E. S. G. Maciel, and A. P. Pimenta, Thermochemical Non-Equilibrium Reentry Flows in Three-Dimensions: Seven Species Model – Part I – Structured Solutions, Submitted to *WSEAS Transactions on Mathematics* (under review).

[30] E. S. G. Maciel, and A. P. Pimenta, Reentry Flows in Chemical Non-Equilibrium in Two-Dimensions, *Proceedings of the 10th International Symposium on Combustion and Energy Utilisation (10th ICCEU)*, Mugla, Turkey, 2010.

[31] E. S. G. Maciel, Reentry Flows in Chemical Non-Equilibrium in Two-Dimensions – Part II, *Mecánica Computational Journal*, Vol. XXX, No. 5, 2011, pp. 323-347.

[32] E. S. G. Maciel, and A. P. Pimenta, Thermochemical Non-Equilibrium Reentry Flows in Two-Dimensions – Part I, *WSEAS Transactions on Mathematics*, Vol. 11, June, Issue 6, 2012, pp. 520-545.

[33] E. S. G. Maciel, and A. P. Pimenta, Thermochemical Non-Equilibrium Reentry Flows in Two-Dimensions – Part II, Submitted and Accepted by *WSEAS Transactions on Mathematics* (to be published).