

TVD and ENO Applications to Supersonic Flows in 2D – Part II

EDISSON SÁVIO DE GÓES MACIEL⁽¹⁾ and EDUARDO MANFREDINI FERREIRA⁽²⁾

Faculdade de Engenharia (FAEN)

Fundação Universidade Federal da Grande Dourados (UFGD)

Rodovia Dourados – Itahum, km 12, Caixa Postal 533 – CEP 79804-970

Cidade Universitária – Dourados – MS – Brazil

⁽¹⁾edisavio@edissonsavio.eng.br ⁽¹⁾<http://www.edissonsavio.eng.br>

⁽²⁾eduardomanfredini@ufgd.edu.br

Abstract: - In this work, second part of this study, the high resolution numerical schemes of Yee and Harten, of Yang second order, of Yang third order, and of Yang and Hsu are applied to the solution of the Euler and Navier-Stokes equations in two-dimensions. All schemes are flux difference splitting algorithms. The Yee and Harten is a TVD (“Total Variation Diminishing”) second order accurate in space and first order accurate in time algorithm. The Yang second order is a TVD/ENO (“Essentially Nonoscillatory”) second order accurate in space and first order accurate in time algorithm. The Yang third order is a TVD/ENO third order accurate in space and first order accurate in time algorithm. Finally, the Yang and Hsu is a UNO (Uniformly Nonoscillatory) third order accurate in space and first order accurate in time algorithm. The Euler and Navier-Stokes equations, written in a conservative and integral form, are solved, according to a finite volume and structured formulations. A spatially variable time step procedure is employed aiming to accelerate the convergence of the numerical schemes to the steady state condition. It has proved excellent gains in terms of convergence acceleration as reported by Maciel. The physical problems of the supersonic shock reflection at the wall and the supersonic flow along a compression corner are solved, in the inviscid case. For the viscous case, the supersonic flow along a compression corner is solved. In the inviscid case, an implicit formulation is employed to marching in time, whereas in the viscous case, a time splitting or Strang approaches are used. The results have demonstrated that the Yang ENO third order accurate algorithm has presented the best solutions in the problems studied herein. Moreover, it is also the best as comparing with the numerical schemes of Part I of this study.

Key-Words: - Yee and Harten algorithm, Yang second order TVD/ENO algorithm, Yang third order TVD/ENO algorithm, Yang and Hsu UNO algorithm, Euler and Navier-Stokes equations, Finite Volumes.

1 Introduction

In recent years, many high resolution shock capturing finite volume schemes for the computation of the Euler equations have been developed. Of special interest are the methods that generate nonoscillatory solutions but sharp approximations to shock and contact discontinuities. This interest stems from the fact that even with the advances in high-speed supercomputing, grid generation, automatic adaptive grid procedures, etc., the lack of robust and accurate numerical schemes is a major stumbling block for the success of computational fluid dynamics. Most of these schemes [1-9] are very different in form, methodology, and design principle. However, from the standpoint of numerical analysis, these schemes are total variation diminishing (TVD) for nonlinear scalar hyperbolic conservation laws and for constant coefficient hyperbolic systems. The notion of TVD schemes was introduced by Harten [1-2]. Some of these methods can also be viewed as three-point

central difference schemes with a “smart” numerical dissipation or smoothing mechanism. “Smart” here means automatic feedback mechanism to control the amount of numerical dissipation for nonlinear problems. In general, TVD schemes can be divided into two categories, namely, upwind and symmetric TVD schemes. A way of distinguish an upwind from a symmetric TVD scheme is that the numerical dissipation term corresponding to an upwind TVD scheme is upwind-weighted [1-6] as opposed to the numerical dissipation term corresponding to a symmetric TVD scheme that is centered [7-9].

Harten’s method of constructing high resolution TVD schemes involves starting with a first order TVD scheme and applying it to a modified flux. The modified flux is chosen so that the scheme is second order at regions of smoothness and first order at points of extrema. This technique is sometimes referred to as the modified flux approach. Although the scheme is an upwind scheme, it is written in a symmetric form; i.e., central difference plus an appropriate numerical dissipation term. This special

form is especially advantageous for systems of higher than one space dimension. It results in less storage and a smaller operation count than its upwind form [10]. The modified flux approach is relatively simple to understand and easy to implement into a new or existing computing code. One can modify a standard three-point central difference code by simply changing the conventional numerical dissipation term into the one designed for the TVD scheme.

In [11], a preliminary study was completed on the implicit TVD scheme for a two-dimensional gasdynamics problem in a Cartesian coordinate. It was found that further improvement in computation efficiency and converged rate is required for practical application.

[12] proposed a modification in the work of [11], written via the modified flux approach, aiming to extend these methods to the multidimensional hyperbolic conservation laws in curvilinear coordinates. They presented various ways of linearizing the implicit operator and solution strategies to improve the computation efficiency of the implicit algorithm were discussed. Numerical experiments with some AGARD test cases for steady-state airfoil calculations showed that the proposed linearized implicit upwind TVD schemes were quite robust and accurate.

Very recently, a new class of uniformly high-order-accurate essentially nonoscillatory (ENO) schemes have been developed by [13-15]. They presented a hierarchy of uniformly high-order-accurate schemes that generalize Godunov's scheme [16], its second order accurate MUSCL extension [3; 17] and the total variation diminishing (TVD) scheme [18] to arbitrary order of accuracy. In contrast to the earlier second-order TVD schemes which drop to first-order accuracy at local extrema and maintain second-order accuracy in smooth regions, the new ENO schemes are uniformly high-order accurate throughout, even at critical points. The ENO schemes use a reconstruction algorithm that is derived from a new interpolation technique that when applied to piecewise smooth data gives high-order accuracy whenever the function is smooth but avoids a Gibbs phenomenon at discontinuities. An adaptive stencil of grid points is used; therefore, the resulting schemes are highly nonlinear even in the scalar case.

Theoretical results for the scalar coefficient case and numerical results for the scalar conservation law and for the one-dimensional Euler equations of gas dynamics have been reported with highly accurate results. Such high-order ENO schemes have the potential to be adapted to the current Euler/Navier-

Stokes flow solvers as one does for the second order TVD explicit and implicit schemes [11; 19-20] to further enhance the accuracy of flowfield simulation. Implementation can be either as a higher-order flow solver as in the present work or as a postprocessor to enhance the resolution.

[21] formally extended his second-order TVD schemes described in [22-23] to uniformly second-order ENO schemes for the two-dimensional Euler equations in curvilinear coordinate systems. Both explicit and implicit schemes were described. The authors emphasized in this work that TVD schemes are a special case of ENO schemes in which the TVD requirement is replaced by a less restricted essentially nonoscillatory condition, a concept advanced by Harten and co-workers [13-15]. Numerical experiments with the ENO scheme for an one-dimensional blast wave diffraction around a cylinder, shock wave collision over a circular arc, and steady transonic flow over a circular arc in a channel were reported.

[24] described a class of third-order, essentially nonoscillatory shock-capturing schemes for the Euler equations of gas dynamics. These schemes were obtained by applying the characteristic flux-difference splitting to an appropriately modified flux vector that could have high-order accuracy and nonoscillatory property. Third-order schemes were constructed using upstream interpolation and ENO interpolation. Both explicit and implicit schemes were derived. Implicit schemes to two-dimensional Euler equations in general coordinates were also given. The author applied the resulting schemes to simulate one-dimensional and two-dimensional unsteady shock tube flows and steady two-dimensional flows involving strong shocks to illustrate the performance of the schemes.

[25], following the works of [13-15], described a class of third-order (at least one-dimensional scalar case) shock capturing UNO schemes for the Euler equations of gas dynamics. Third-order schemes were constructing using UNO interpolation. The development was identical to those given in [24], except that Roe's approximate Riemann solver [26] was employed instead of the characteristic flux difference splitting method. The main difference between the approach used in [24] and in this work was that the former one operated on the difference of flux vector, whereas in this work operated on the difference of conservative state vector. It is known that the conservative vector is not continuous across the shock whereas the flux is continuous; that is, the flux vector function is one order smoother than the conservative state vector function. The Roe's averages [26] enable the Rankine-Hugoniot

relations to be satisfied across the shock. Also, the Roe's linearization technique for nonlinear system permits the use of completely different characteristic fields and is one of the most popular approximate Riemann solvers currently in use.

Traditionally, implicit numerical methods have been praised for their improved stability and condemned for their large arithmetic operation counts ([27]). 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 (Curret-Friedrichs-Lewis) 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 [28-30], and fractional step algorithms by [31]. ADI approximate factorization methods consist in approximating the Left Hand Side (LHS) of the numerical scheme by the product of one-dimensional parcels, each one associated with a different spatial coordinate direction, which retract nearly the original implicit operator. These methods have been largely applied in the CFD ("Computational Fluid Dynamics") community and, despite the fact of the error of the approximate factorization, it allows the use of large time steps, which results in significant gains in terms of convergence rate in relation to explicit methods.

In the present work, second part of this study, the [12] TVD, the [21] TVD/ENO, the [24] TVD/ENO, and the [25] UNO schemes are implemented, on a finite volume context and using a structured spatial discretization, to solve the Euler and Navier-Stokes equations in the two-dimensional space. All schemes are high resolution flux difference splitting ones, based on the concept of Harten's modified flux function. The [12] is a TVD second order accurate in space and first order accurate in time algorithm. [21] is a TVD/ENO second order accurate in space and first order accurate in time algorithm. The [24] is a TVD/ENO third order accurate in space and first order accurate in time algorithm. Finally, the [25] is a UNO (Uniformly Nonoscillatory) third order accurate in space and first order accurate in time algorithm. An implicit

formulation is employed to solve the Euler equations, whereas a time splitting or Strang methods, explicit methods, are used to solve the Navier-Stokes equations. A Linearized Nonconservative Implicit LNI form or an approximate factorization ADI method is employed by the schemes. The algorithms are accelerated to the steady state solution using a spatially variable time step, which has demonstrated effective gains in terms of convergence rate ([32-33]). All schemes are applied to the solution of physical problems of the supersonic shock reflection at the wall and the supersonic flow along a compression corner, in the inviscid case, whereas in the laminar viscous case, the supersonic flow along a compression corner is solved. The results have demonstrated that the [24] ENO algorithm, third order accurate in space, has presented the best solutions, in this study.

2 Navier-Stokes Equations

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

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

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

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

$$E_v = \frac{1}{\text{Re}} \begin{Bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xx}u + \tau_{xy}v - q_x \end{Bmatrix}, F_v = \frac{1}{\text{Re}} \begin{Bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yx}u + \tau_{yy}v - q_y \end{Bmatrix}. \quad (3)$$

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

$$\tau_{xx} = 2\mu_M \partial u / \partial x - 2/3 \mu_M (\partial u / \partial x + \partial v / \partial y); \quad (4)$$

$$\tau_{xy} = \mu_M (\partial u / \partial y + \partial v / \partial x); \quad (5)$$

$$\tau_{yy} = 2\mu_M \partial v / \partial y - 2/3 \mu_M (\partial u / \partial x + \partial v / \partial y). \quad (6)$$

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

$$q_x = -\gamma(\mu_M / Prd) \partial e_i / \partial x; \quad (7)$$

$$q_y = -\gamma(\mu_M / Prd) \partial e_i / \partial y. \quad (8)$$

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

$$e_i = e / \rho - 0.5(u^2 + v^2); \quad (9)$$

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

$$Re = \rho u_{REF} l / \mu_M, \quad (10)$$

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

$$\mu_M = bT^{1/2} / (1 + S/T), \quad (11)$$

where T is the absolute temperature (K), $b = 1.458 \times 10^{-6} \text{ Kg}/(\text{m.s.K}^{1/2})$ and $S = 110.4 \text{ K}$, to the atmospheric air in the standard atmospheric conditions ([34]). The Navier-Stokes equations were nondimensionalized in relation to the freestream density, ρ_∞ , and the freestream speed of sound, a_∞ ,

for the all problems. For the viscous compression corner problem it is also considered the freestream molecular viscosity, μ_∞ . To allow the solution of the matrix system of four equations to four unknowns described by Eq. (1), it is employed the state equation of perfect gases presented below:

$$p = (\gamma - 1) [e - 0.5\rho(u^2 + v^2)]. \quad (12)$$

The total enthalpy is determined by:

$$H = (e + p) / \rho. \quad (13)$$

3 Yee and Harten TVD Second Order Algorithm

The [12] TVD algorithm, second order accurate in space, is specified by the determination of the numerical flux vector at the $(i+1/2, j)$ interface. The extension of this numerical flux to the $(i, j+1/2)$ interface is straightforward, without any additional complications.

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

$$V_R = V_{i+1, j}, \quad V_L = V_{i, j} \quad \text{and} \quad V_{int} = 0.5(V_R + V_L), \quad (14)$$

where "R" and "L" represent right and left, respectively. The cell volume is defined by:

$$V_{i, j} = 0.5 \left[(x_{i, j} - x_{i+1, j}) y_{i+1, j+1} + (x_{i+1, j} - x_{i+1, j+1}) y_{i, j} + (x_{i+1, j+1} - x_{i, j}) y_{i+1, j} \right] + 0.5 \left[(x_{i, j} - x_{i+1, j+1}) y_{i, j+1} + (x_{i+1, j+1} - x_{i, j+1}) y_{i, j} + (x_{i, j+1} - x_{i+1, j+1}) y_{i+1, j+1} \right], \quad (15)$$

where a computational cell, with its nodes and flux surfaces are defined in Fig. 1.

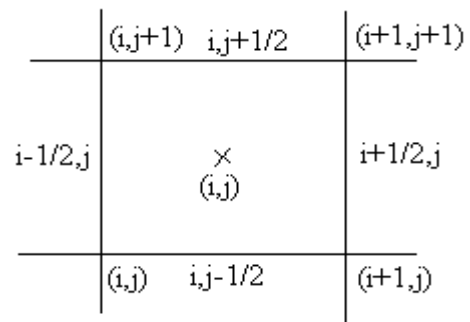


Figure 1. Computational cell, interfaces and nodes.

The area components at interface are defined by: $S_{x_int} = s'_x S$ and $S_{y_int} = s'_y S$, where s'_x and s'_y are defined as: $s'_x = s_x/S$ and $s'_y = s_y/S$, being $S = (s_x^2 + s_y^2)^{0.5}$. Expressions to s_x and s_y , which represent the S_x and S_y components always adopted in the positive orientation, are given in Tab. 1.

Table 1. Normalized values of s_x and s_y .

Surface:	s_x :	s_y :
i,j-1/2	$-(y_{i+1,j} - y_{i,j})$	$(x_{i+1,j} - x_{i,j})$
i+1/2,j	$(y_{i+1,j+1} - y_{i+1,j})$	$(x_{i+1,j} - x_{i+1,j+1})$
i,j+1/2	$(y_{i,j+1} - y_{i+1,j+1})$	$(x_{i+1,j+1} - x_{i,j+1})$
i-1/2,j	$(y_{i,j+1} - y_{i,j})$	$-(x_{i,j+1} - x_{i,j})$

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

$$h_x = S_{x_int}/V_{int}, h_y = S_{y_int}/V_{int} \text{ and } h_n = S/V_{int}. \quad (16)$$

The calculated properties at the flux interface are obtained by arithmetical average or by [26] average. The [26] average was used in this work:

$$\rho_{int} = \sqrt{\rho_L \rho_R}, u_{int} = (u_L + u_R \sqrt{\rho_R/\rho_L}) / (1 + \sqrt{\rho_R/\rho_L}), \quad (17)$$

$$v_{int} = (v_L + v_R \sqrt{\rho_R/\rho_L}) / (1 + \sqrt{\rho_R/\rho_L}); \quad (18)$$

$$H_{int} = (H_L + H_R \sqrt{\rho_R/\rho_L}) / (1 + \sqrt{\rho_R/\rho_L}); \quad (19)$$

$$a_{int} = \sqrt{(\gamma-1)[H_{int} - 0.5(u_{int}^2 + v_{int}^2)]}. \quad (20)$$

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

$$U_{cont} = u_{int} h_x + v_{int} h_y, \lambda_1 = U_{cont} - a_{int} h_n, \quad (21)$$

$$\lambda_2 = \lambda_3 = U_{cont} \text{ and } \lambda_4 = U_{cont} + a_{int} h_n. \quad (22)$$

The jumps in the conserved variables, necessary to the construction of the [12] TVD dissipation function, are given by:

$$\Delta e = V_{int}(e_R - e_L), \Delta \rho = V_{int}(\rho_R - \rho_L), \Delta(\rho u) = V_{int}[(\rho u)_R - (\rho u)_L]; \quad (23)$$

$$\Delta(\rho v) = V_{int}[(\rho v)_R - (\rho v)_L]. \quad (24)$$

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

$$\alpha_1 = 0.5(aa - bb), \alpha_2 = \Delta\rho - aa, \alpha_3 = cc; \quad (25)$$

$$\alpha_4 = 0.5(aa + bb), \quad (26)$$

with:

$$aa = (\gamma-1)/a_{int}^2 [\Delta e + 0.5(u_{int}^2 + v_{int}^2)\Delta\rho - u_{int}\Delta(\rho u) - v_{int}\Delta(\rho v)]; \quad (27)$$

$$bb = 1/a_{int} [h'_x \Delta(\rho u) - (h'_x u_{int} + h'_y v_{int})\Delta\rho + h'_y \Delta(\rho v)]; \quad (28)$$

$$cc = h'_x \Delta(\rho v) + (h'_y u_{int} - h'_x v_{int})\Delta\rho - h'_y \Delta(\rho u); \quad (29)$$

$$h'_x = h_x/h_n \text{ and } h'_y = h_y/h_n. \quad (30)$$

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

$$R_{i+1/2,j} = \begin{bmatrix} 1 & 1 & 0 \\ u_{int} - h'_x a_{int} & u_{int} & -h'_y \\ v_{int} - h'_y a_{int} & v_{int} & h'_x \\ H_{int} - h'_x u_{int} a_{int} - h'_y v_{int} a_{int} & 0.5(u_{int}^2 + v_{int}^2) & h'_x v_{int} - h'_y u_{int} \\ 1 & & \\ u_{int} + h'_x a_{int} & & \\ v_{int} + h'_y a_{int} & & \\ H_{int} + h'_x u_{int} a_{int} + h'_y v_{int} a_{int} & & \end{bmatrix}. \quad (31)$$

The numerical flux function or modified function of [1], g , responsible to the second order accuracy of the [12] scheme, is defined as:

$$g^l_{i,j} = S \times \text{MAX} \left[0, \text{MIN} \left(|\alpha^l_{i+1/2,j}|, S \times \alpha^l_{i-1/2,j} \right) \right] \quad (32)$$

with:

$$S = \text{sign}(\alpha^l_{i+1/2,j}) \quad (33)$$

The entropy function to avoid non-physical solutions, is defined as:

$$\Psi(z) = \begin{cases} |z| & \text{if } |z| \geq \varepsilon \\ (z^2 + \varepsilon^2)/(2\varepsilon) & \text{if } |z| < \varepsilon \end{cases} \quad (34)$$

with $\varepsilon = 0.2$, as recommended by [11].

The numerical information propagation velocity, ϕ , responsible to transport the numerical information to the algorithm, is determined by:

$$\phi_{i+1/2,j}^l = \frac{1}{2} \Psi(\lambda_{i+1/2,j}^l) \times \begin{cases} (g_{i+1,j}^l + g_{i,j}^l) / \alpha_{i+1/2,j}^l & \text{if } \alpha_{i+1/2,j}^l \neq 0.0 \\ 0.0 & \text{if } \alpha_{i+1/2,j}^l = 0.0 \end{cases} \quad (35)$$

The ϕ dissipation function to form the numerical dissipation operator is written as

$$\phi_{i+1/2,j}^l = \frac{1}{2} \Psi(\lambda_{i+1/2,j}^l) (g_{i,j}^l + g_{i+1,j}^l) - \Psi(\lambda_{i+1/2,j}^l + \phi_{i+1/2,j}^l) \alpha_{i+1/2,j}^l. \quad (36)$$

In all definitions, Eqs. (32-35), “ P ” varies from 1 to 4 (two-dimensional space).

Finally, the [12] TVD dissipation function is constructed by the following matrix-vector product:

$$\{D_{YH}\}_{i+1/2,j} = [R]_{i+1/2,j} \{\Phi\}_{i+1/2,j}. \quad (37)$$

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

$$F_{i+1/2,j}^{(l)} = (E_{int}^{(l)} h_x + F_{int}^{(l)} h_y) V_{int} + 0.5 D_{YH}^{(l)}, \quad (38)$$

with:

$$E_{int}^{(l)} = 0.5 [(E_R^{(l)} + E_L^{(l)})_e] - (E_v^{(l)})_{int}; \quad (39)$$

$$F_{int}^{(l)} = 0.5 [(F_R^{(l)} + F_L^{(l)})_e] - (F_v^{(l)})_{int}. \quad (40)$$

The viscous vectors at the flux interface are obtained by arithmetical average between the primitive variables at the left and at the right states of the flux interface, as also arithmetical average of the primitive variable gradients also considering the left and the right states of the flux interface.

The right-hand-side (RHS) of the [12] TVD scheme, necessities to the resolution of the implicit version of this algorithm, is determined by:

$$RHS(YH)_{i,j}^n = -\Delta t_{i,j} / V_{i,j} (F_{i+1/2,j}^{YH} - F_{i-1/2,j}^{YH} + F_{i,j+1/2}^{YH} - F_{i,j-1/2}^{YH})^n. \quad (41)$$

The time integration to the viscous simulations follows the time splitting method, first order accurate, which divides the integration in two steps, each one associated with a specific spatial direction.

In the initial step, it is possible to write for the ξ direction:

$$\begin{aligned} \Delta Q_{i,j}^* &= -\Delta t_{i,j} / V_{i,j} (F_{i+1/2,j}^n - F_{i-1/2,j}^n); \\ Q_{i,j}^* &= Q_{i,j}^n + \Delta Q_{i,j}^*; \end{aligned} \quad (42)$$

and at the end step, η direction:

$$\begin{aligned} \Delta Q_{i,j}^{n+1} &= -\Delta t_{i,j} / V_{i,j} (F_{i,j+1/2}^* - F_{i,j-1/2}^*); \\ Q_{i,j}^{n+1} &= Q_{i,j}^* + \Delta Q_{i,j}^{n+1}. \end{aligned} \quad (43)$$

4 Yang TVD/ENO Second Order Algorithm

A typical conservative numerical scheme, using a finite volume formulation, for solving Eq. (1) can be expressed in terms of numerical fluxes as follows:

$$Q_{i,j}^{n+1} = Q_{i,j}^n - \frac{\Delta t_{i,j}}{V_{i,j}} (E_{i+1/2,j}^N - E_{i-1/2,j}^N) - \frac{\Delta t_{i,j}}{V_{i,j}} (F_{i,j+1/2}^N - F_{i,j-1/2}^N), \quad (44)$$

where $E_{i+1/2,j}^N$ and $F_{i,j+1/2}^N$ are the numerical fluxes.

For a first order upwind scheme, $E_{i+1/2,j}^N$ is given by:

$$E_{i+1/2,j}^N = E_{i+1/2,j}^n - \hat{A}_{i+1/2,j}^+ \Delta_{i+1/2,j} E_{i,j}^n, \quad (45)$$

with: $\Delta_{i+1/2,j} = (\cdot)_{i+1,j} - (\cdot)_{i,j}$, $E_{i+1/2,j}^n$ and $E_{i,j}^n$ defined by:

$$E_{i+1/2,j}^n = V_{int} \begin{Bmatrix} \rho U_{cont} \\ \rho u U_{cont} + p h_x \\ \rho v U_{cont} + p h_y \\ (e + p) \mathcal{U}_{cont} \end{Bmatrix}_{i+1/2,j} \quad \text{and} \quad (46)$$

$$E_{i,j}^n = V_{int} \begin{Bmatrix} \rho U_{cont} \\ \rho u U_{cont} + p h_x \\ \rho v U_{cont} + p h_y \\ (e + p) \mathcal{U}_{cont} \end{Bmatrix}_{i,j}$$

and $\hat{A}_{i+1/2,j}^+$ defined as follows:

$$\hat{A}_{i+1/2,j}^+ = (R_{\xi} \hat{\Lambda}_{\xi}^+ R_{\xi}^{-1})_{i+1/2,j}, \quad (47a)$$

$$\hat{\Lambda}_\xi^+ = \text{diag}\{\hat{\lambda}_1^+\} \text{ and } \hat{\lambda}_i^+ = 0.5[1 + \text{sign}(\lambda_i^\xi)], \quad (47b)$$

where: R_ξ and R_ξ^{-1} defined by Eqs. (31) and (48), respectively, to the ξ direction;

$$R^{-1} = \begin{bmatrix} \frac{1}{2} \left[\frac{\gamma-1}{a_{\text{int}}^2} \frac{(u_{\text{int}}^2 + v_{\text{int}}^2)}{2} + \frac{1}{a_{\text{int}}} (u_{\text{int}} h'_x + v_{\text{int}} h'_y) \right] & & & & \\ & 1 - \frac{\gamma-1}{a_{\text{int}}^2} \frac{(u_{\text{int}}^2 + v_{\text{int}}^2)}{2} & & & \\ & & - (h'_x v_{\text{int}} - h'_y u_{\text{int}}) & & \\ & & & \frac{1}{2} \left[\frac{\gamma-1}{a_{\text{int}}^2} \frac{(u_{\text{int}}^2 + v_{\text{int}}^2)}{2} - \frac{1}{a_{\text{int}}} (u_{\text{int}} h'_x + v_{\text{int}} h'_y) \right] & \\ \frac{1}{2} \left(-\frac{\gamma-1}{a_{\text{int}}^2} u_{\text{int}} - \frac{h'_x}{a_{\text{int}}} \right) & \frac{1}{2} \left(-\frac{\gamma-1}{a_{\text{int}}^2} v_{\text{int}} - \frac{h'_y}{a_{\text{int}}} \right) & \frac{\gamma-1}{2a_{\text{int}}^2} & & \\ & \frac{\gamma-1}{a_{\text{int}}^2} u_{\text{int}} & \frac{\gamma-1}{a_{\text{int}}^2} v_{\text{int}} & -\frac{\gamma-1}{a_{\text{int}}^2} & \\ & -h'_y & h'_x & 0 & \\ \frac{1}{2} \left(-\frac{\gamma-1}{a_{\text{int}}^2} u_{\text{int}} + \frac{h'_x}{a_{\text{int}}} \right) & \frac{1}{2} \left(-\frac{\gamma-1}{a_{\text{int}}^2} v_{\text{int}} + \frac{h'_y}{a_{\text{int}}} \right) & \frac{\gamma-1}{2a_{\text{int}}^2} & & \end{bmatrix}, \quad (48)$$

with h'_x and h'_y defined according to Eq. (30); $\text{diag}(\cdot)$ represents a diagonal matrix, as for instance:

$$\hat{\Lambda}_\xi^+ = \begin{bmatrix} \hat{\lambda}_1^+ & & & & \\ & \hat{\lambda}_2^+ & & & \\ & & \hat{\lambda}_3^+ & & \\ & & & \hat{\lambda}_4^+ & \\ & & & & \hat{\lambda}_5^+ \end{bmatrix}; \quad (49)$$

λ_i^ξ are defined by Eqs. (21-22) to the ξ direction; $\text{sign}(\lambda_i^\xi)$ is equal to 1.0 if $\lambda_i^\xi \geq 0.0$ and -1.0 otherwise.

4.1 Uniformly second order essentially non-oscillatory scheme

[1] proposed to construct second order accurate TVD schemes by applying a first order approximate Riemann solver to a modified flux. Following [1], [21] proposed to define a modified numerical flux function with the definition of the modified fluxes:

$$E^n = E^M = E^n + \bar{E}^n; \quad (50a)$$

$$F^n = F^M = F^n + \bar{F}^n, \quad (50b)$$

where E^M and F^M are the modified fluxes which have essentially non-oscillatory property yet to be defined. In the following, a numerical method of uniformly second order accuracy in time and space which combines both characteristic and conversion features of Eq. (1) is discussed.

For the present ENO scheme, the numerical flux E^N is described by:

$$E_{i+1/2,j}^N = E_{i+1,j}^M - \hat{A}_{\xi_{i+1/2,j}}^+ \Delta_{i+1/2,j} E_{i,j}^M = E_{i,j}^M + \hat{A}_{\xi_{i+1/2,j}}^- \Delta_{i+1/2,j} E_{i,j}^M. \quad (51)$$

The components of the additional vector \bar{E} are given by:

$$\bar{e}_{i,j} = m \left[\tilde{e}_{i-1/2,j} + \beta \bar{m} (\Delta_- \tilde{e}_{i-1/2,j}, \Delta_+ \tilde{e}_{i-1/2,j}), \tilde{e}_{i+1/2,j} - \beta \bar{m} (\Delta_- \tilde{e}_{i+1/2,j}, \Delta_+ \tilde{e}_{i+1/2,j}) \right], \quad (52)$$

where:

$$\Delta_- = (\cdot)_{i,j} - (\cdot)_{i-1,j}, \quad \Delta_+ = (\cdot)_{i+1,j} - (\cdot)_{i,j}; \quad (53)$$

and $\tilde{e}_{i+1/2,j}$ are components of the following column vector:

$$\tilde{E}_{i+1/2,j} = \text{sign}(A_{i+1/2,j}) (I - \Delta t_{i,j} |A_{i+1/2,j}|) \Lambda_{i+1/2,j} E_{i,j} / 2. \quad (54)$$

The $\text{sign}(A)$ and $|A|$ in Eq. (54) are given by:

$$\text{sign}(A) = R_\xi \text{diag}\{\text{sign}(\lambda_i^\xi)\} R_\xi^{-1}; \quad |A| = R_\xi \text{diag}\{|\lambda_i^\xi|\} R_\xi^{-1}. \quad (55)$$

Similar derivations can be given for the \bar{F} vector in the η direction. In Equation (52), m is the minmod function:

$$m(a,b) = \begin{cases} s \times \min(|a|, |b|) & \text{if } \text{sign}(a) = \text{sign}(b) = s; \\ 0.0 & \text{otherwise} \end{cases}; \quad (56)$$

and the \bar{m} function is defined by:

$$\bar{m} = \begin{cases} a & \text{if } |a| \leq |b| \\ b & \text{if } |a| \geq |b| \end{cases}. \quad (57)$$

For $\beta = 0.0$, one has a second order TVD scheme. For $\beta = 0.5$, one has a uniformly second order non-oscillatory scheme. The original numerical scheme of [21] is thus formed by Eq. (44) using the definition (51) to the numerical flux function.

The first author introduced some modifications in the [21] scheme in this work. Equation (51) is redefined as:

$$E_{i+1/2,j}^N = E_{i+1,j}^M - A_{\xi_{i+1/2,j}}^+ \Delta_{i+1/2,j} Q_{i,j}^M = E_{i,j}^M + A_{\xi_{i+1/2,j}}^- \Delta_{i+1/2,j} Q_{i,j}^M, \quad (58)$$

with:

$$E_{i+1,j}^M = E_{i+1,j}^n + A_{\xi_{i,j}}^n \bar{E}_{i+1,j}^n; \quad (59)$$

$$Q_{i,j}^M = V_{int} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{Bmatrix}_{i,j} + \bar{E}_{i,j}^n. \quad (60)$$

The positive splitting matrix $A_{\xi_{i+1/2,j}}^+$ is defined as

$$A_{\xi_{i+1/2,j}}^+ = R_{\xi} \text{diag}\{\lambda_l^+\} R_{\xi}^{-1}, \quad (61)$$

with:

$$\lambda_l^+ = 0.5(\lambda_l^{\xi} + |\lambda_l^{\xi}|), \quad (62)$$

and the Jacobian matrix at the ξ direction is described by

$$A_{\xi_{i,j}}^n = R_{\xi} \text{diag}\{\lambda_l^{\xi}\} R_{\xi}^{-1}. \quad (63)$$

The vector $\tilde{E}_{i+1/2,j}$ is also redefined as:

$$\tilde{E}_{i+1/2,j} = \text{sign}(A_{i+1/2,j}) \left(I - \Delta t_{i,j} |A_{i+1/2,j}| \right) A_{i+1/2,j} Q_{i,j} / 2, \quad (64)$$

where:

$$Q_{i,j} = V_{int} \{ \rho \quad \rho u \quad \rho v \quad e \}_{i,j}^T. \quad (65)$$

Observe that the resulting scheme is equivalent to the original of [21], with the unique difference that the difference of fluxes in Eq. (54) is changed by the difference of conserved variables. With this new definition, the solutions present better behaviour,

free of oscillations, undershoots and overshoots. The other expressions maintain the same structure.

The right-hand-side of the [21] scheme, necessary to the resolution of the implicit version of this algorithm, is defined by:

$$RHS(Y)_{i,j}^n = -\frac{\Delta t_{i,j}}{V_{i,j}} (E_{i+1/2,j}^N - E_{i-1/2,j}^N) - \frac{\Delta t_{i,j}}{V_{i,j}} (F_{i,j+1/2}^N - F_{i,j-1/2}^N). \quad (66)$$

The viscous formulation obeys the same procedure described in section 3. For explicit methods in two-dimensions, the Strang-type directional splitting [35] was employed

$$Q_{i,j}^{n+2} = L_{\xi}(\Delta t) L_{\eta}(\Delta t) L_{\eta}(\Delta t) L_{\xi}(\Delta t) Q_{i,j}^n \quad (67)$$

The L_{ξ} operator is defined by

$$L_{\xi}(\Delta t) Q_{i,j}^n = Q_{i,j}^n - \Delta t (E_{i+1/2,j}^N - E_{i-1/2,j}^N). \quad (68)$$

Similar expressions can be given for $F_{i+1/2,j}^N$ and the L_{η} operator.

5 Yang TVD/ENO Third Order Algorithm

5.1 TVD formulation

In [36], second- and third-order upwind schemes have been described by one-dimensional cases. It was found that such high-order schemes can be constructed by using a more accurate flux representation (in the discrete sense) at each nodal point. [24] has taken such an approach following Harten's work [18] in which he applied a three-point first-order upwind scheme to a modified flux to yield second-order TVD scheme. Therefore, [24] calls it the modified flux approach.

[24] has considered a high-order extension of the Euler equations in one-dimensional case. The extension to the two-dimensional case is as follows:

$$\frac{\partial Q}{\partial t} + (\hat{A}^+ + \hat{A}^-) \frac{\partial E^M(Q)}{\partial x} + (\hat{B}^+ + \hat{B}^-) \frac{\partial F^M(Q)}{\partial y} = 0. \quad (69)$$

Here, $E_{i,j}^M$ and $F_{i,j}^M$ are called the modified flux vectors at nodal point (i,j) and is consisted of the original flux vectors $E_{i,j}$ and $F_{i,j}$ and additional terms of high-order accuracy that usually have some

nonlinear control terms to avoid oscillatory solutions.

In terms of numerical flux vector, a conservative scheme for Eq. (69) could be Eq. (44), with the numerical flux, in the ξ direction, defined as:

$$E_{i+1/2,j}^N = E_{i+1,j}^M - \hat{A}_{i+1/2,j}^+ \Delta_+ E_{i,j}^M = E_{i,j}^M + \hat{A}_{i+1/2,j}^- \Delta_+ E_{i,j}^M. \quad (70)$$

A third-order scheme for Eq. (44) can be expressed in terms of numerical flux of the form Eq. (70) with

$$E_{i,j}^M = E_{i,j}^{TVD3} = E_{i,j}^n + D_{i,j}^n. \quad (71)$$

The components of $D_{i,j}$ are given by

$$d_{i,j}^l = [1 - S(\theta_{i,j}^l)] \tilde{d}_{i+1/2,j}^l + [1 + S(\theta_{i,j}^l)] \bar{d}_{i-1/2,j}^l, \quad (72)$$

where $\tilde{d}_{i+1/2,j}^l$ and $\bar{d}_{i+1/2,j}^l$ are components of $\tilde{D}_{i+1/2,j}$ and $\bar{D}_{i+1/2,j}$ given, respectively, by

$$\tilde{D}_{i+1/2,j} = \text{sign} A_{i+1/2,j} \left(\Delta t_{i,j}^2 |A_{i+1/2,j}|^2 - 3\Delta t_{i,j} |A_{i+1/2,j}| + 2I \right) \Delta_+ E_{i,j} / 6; \quad (73)$$

$$\bar{D}_{i+1/2,j} = \text{sign} A_{i+1/2,j} \left(I - \Delta t_{i,j}^2 |A_{i+1/2,j}|^2 \right) \Delta_+ E_{i,j} / 6, \quad (74)$$

and $S(\theta_{i,j}^l)$ is the smoothness monitor given by [37] as

$$S(\theta_{i,j}^l) = \begin{cases} 0, & \text{if } |\Delta_+ q_{i,j}^l| + |\Delta_- q_{i,j}^l| = 0 \\ \frac{|\Delta_+ q_{i,j}^l| - |\Delta_- q_{i,j}^l|}{|\Delta_+ q_{i,j}^l| + |\Delta_- q_{i,j}^l|}, & \text{otherwise} \end{cases}; \quad (75)$$

where $q_{i,j}^l$ are components of the conservative state vector $Q_{i,j}$; and the Δ 's defined as follows to the ξ direction:

$$\Delta_+ = (\cdot)_{i+1,j} - (\cdot)_{i,j} \quad \text{and} \quad \Delta_- = (\cdot)_{i,j} - (\cdot)_{i-1,j}. \quad (76)$$

The first author of this work introduces some modifications in the original scheme of [24]. These modifications are as follows:

$$E_{i+1/2,j}^N = E_{i+1,j}^M - A_{\xi_{i+1/2,j}} \Delta_+ Q_{i,j}^M, \quad (76)$$

where:

$$Q_{i,j}^M = V_{int} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{Bmatrix} + D_{i,j}^n; \quad (77)$$

$$E_{i,j}^M = E_{i,j}^n + A_{\xi_{i,j}}^n D_{i,j}^n; \quad (78)$$

$$\tilde{D}_{i+1/2,j} = \text{sign} A_{i+1/2,j} \left(\Delta t_{i,j}^2 |A_{i+1/2,j}|^2 - 3\Delta t_{i,j} |A_{i+1/2,j}| + 2I \right) \Delta_+ Q_{i,j} / 6; \quad (79)$$

$$\bar{D}_{i+1/2,j} = \text{sign} A_{i+1/2,j} \left(I - \Delta t_{i,j}^2 |A_{i+1/2,j}|^2 \right) \Delta_+ Q_{i,j} / 6. \quad (80)$$

Observe that the resulting scheme is equivalent to the original of [24], with the unique difference that the difference of fluxes in Eqs. (73-74) is changed by the difference of conserved variables. With this new definition, the solutions present better behaviour, free of oscillations, undershoots and overshoots. The other expressions maintain the same structure.

The same equations, considered in the η direction, can be developed, without any additional complexity. The algorithm described above is referred by [24] as TVD3 and the present authors remain this nomenclature.

The definition of the RHS to the implicit formulation is done as follows:

$$RHS(Y)_{i,j}^n = -\frac{\Delta t_{i,j}}{V_{i,j}} \left(E_{i+1/2,j}^{TVD} - E_{i-1/2,j}^{TVD} \right) - \frac{\Delta t_{i,j}}{V_{i,j}} \left(F_{i,j+1/2}^{TVD} - F_{i,j-1/2}^{TVD} \right). \quad (81)$$

The Strang method is applied to the viscous simulations.

5.2 ENO formulation

A third-order ENO scheme for Eq. (69) can be constructed using reconstruction by primitive variable. Here, it is adopted:

$$E_{i,j}^M = E_{i,j}^{ENO3} = E_{i,j}^n + G_{i,j}^n + D_{i,j}^n; \quad (82)$$

$$F_{i,j}^M = F_{i,j}^{ENO3} = F_{i,j}^n + H_{i,j}^n + D_{i,j}^n,$$

and G, H, and D are the terms that make up for the higher order accuracy that also depends either the

TVD or the ENO property to avoid Gibbs phenomena.

In Eq. (82) the components of the column vector $G_{i,j}$ are given by

$$g_{i,j}^l = m(\tilde{g}_{i+1/2,j}^l, \tilde{g}_{i-1/2,j}^l), \quad (83)$$

where $\tilde{g}_{i+1/2,j}^l$ is given by

$$\tilde{G}_{i+1/2,j} = \text{sign}A_{i+1/2,j} (I - \Delta t_{i,j} |A_{i+1/2,j}|) \Delta_+ E_{i,j} / 2. \quad (84)$$

And the components of column vector $D_{i,j}$ are given by

$$d_{i,j}^l = m(\Delta_- \tilde{d}_{i-1/2,j}^l, \Delta_+ \tilde{d}_{i-1/2,j}^l) \text{ if } |\Delta_{i-1/2,j} q^l| \leq |\Delta_{i+1/2,j} q^l|;$$

or,

$$d_{i,j}^l = m(\Delta_- \hat{d}_{i-1/2,j}^l, \Delta_+ \hat{d}_{i-1/2,j}^l) \text{ if } |\Delta_{i-1/2,j} q^l| > |\Delta_{i+1/2,j} q^l|, \quad (85)$$

where $\tilde{d}_{i+1/2,j}^l$ and $\hat{d}_{i-1/2,j}^l$ are components of $\tilde{D}_{i+1/2,j}$ and $\hat{D}_{i+1/2,j}$, respectively.

$\tilde{D}_{i+1/2,j}$ is given by Eq. (73) and $\hat{D}_{i+1/2,j}$ is given by

$$\hat{D}_{i+1/2,j} = \text{sign}A_{i+1/2,j} (\Delta t_{i,j}^2 |A_{i+1/2,j}|^2 - I) \Delta_+ E_{i,j} / 6. \quad (86)$$

The first author also introduces a modification in the original algorithm of [24] in its ENO3 (ENO third-order) version. This modification is the same as that applied to the TVD3 scheme. Hence,

$$E_{i,j}^M = E_{i,j}^{ENO3} = E_{i,j}^n + A_{\xi_{i+1/2,j}} G_{i,j}^n + A_{\xi_{i+1/2,j}} D_{i,j}^n; \\ F_{i,j}^M = F_{i,j}^{ENO3} = F_{i,j}^n + B_{\eta_{i+1/2,j}} H_{i,j}^n + B_{\eta_{i+1/2,j}} D_{i,j}^n. \quad (87)$$

$$\tilde{G}_{i+1/2,j} = \text{sign}A_{i+1/2,j} (I - \Delta t_{i,j} |A_{i+1/2,j}|) \Delta_+ Q_{i,j} / 2; \\ \hat{D}_{i+1/2,j} = \text{sign}A_{i+1/2,j} (\Delta t_{i,j}^2 |A_{i+1/2,j}|^2 - I) \Delta_+ Q_{i,j} / 6 \quad (88)$$

and $\tilde{D}_{i+1/2,j}$ defined as in Eq. (79). The other equations maintain the same aspect. The extension to the η direction is straightforward.

The definition of the RHS to the implicit formulation is done as follows:

$$RHS(Y)_{i,j}^n = -\frac{\Delta t_{i,j}}{V_{i,j}} (E_{i+1/2,j}^{ENO} - E_{i-1/2,j}^{ENO}) - \frac{\Delta t_{i,j}}{V_{i,j}} (F_{i,j+1/2}^{ENO} - F_{i,j-1/2}^{ENO}). \quad (89)$$

The Strang method is applied to the viscous simulations.

6 Yang and Hsu UNO Third Order Algorithm

Unlike TVD schemes, nonoscillatory algorithms are not required to damp the values of each local extremum at every single time step, but are allowed to occasionally accentuate a local extremum. The design involves an essentially nonoscillatory piecewise polynomial reconstruction of the solution from its cell averages, time evolution through an approximate solution of the resulting initial value problem, and averaging of this approximate solution over each cell.

A third-order UNO scheme for Eq. (1), based on [25] work, can be expressed by the following numerical flux, in ξ direction, for instance:

$$E_{i+1/2,j}^{UNO3} = \frac{1}{2} (E_{i,j}^n + E_{i+1,j}^n + R_{i+1/2,j}^{\xi,n} \Phi_{i+1/2,j}^{UNO3}). \quad (90)$$

The components of $\Phi_{i+1/2,j}^{UNO3}$ are defined as:

$$\Phi_{i+1/2,j}^{UNO3} = \sigma(\lambda_{i+1/2,j}^l) (\beta_{i,j}^l + \beta_{i+1,j}^l) \\ + \begin{cases} \tilde{\sigma}(\lambda_{i+1/2,j}^n) (\tilde{\beta}_{i,j}^l + \tilde{\beta}_{i+1,j}^l) - \psi(\lambda_{i+1/2,j}^l + \gamma_{i+1/2,j}^l + \tilde{\gamma}_{i+1/2,j}^l) \alpha_{i+1/2,j}^l & \text{if } |\alpha_{i-1/2,j}^l| \leq |\alpha_{i+1/2,j}^l| \\ \hat{\sigma}(\lambda_{i+1/2,j}^n) (\hat{\beta}_{i,j}^l + \hat{\beta}_{i+1,j}^l) - \psi(\lambda_{i+1/2,j}^l + \gamma_{i+1/2,j}^l + \hat{\gamma}_{i+1/2,j}^l) \alpha_{i+1/2,j}^l & \text{otherwise} \end{cases} \quad (91)$$

where the σ , $\tilde{\sigma}$ and $\hat{\sigma}$ functions are given by:

$$\sigma = \frac{1}{2} [\psi(z) - \Delta t_{i,j} z^2]; \quad (92)$$

$$\tilde{\sigma} = \frac{1}{6} [2|z| - 3\Delta t_{i,j} |z|^2 + \Delta t_{i,j}^2 |z|^3]; \quad (93)$$

$$\hat{\sigma} = \frac{1}{6} [\Delta t_{i,j}^2 |z|^3 - |z|], \quad (94)$$

and

$$\beta_{i,j}^l = m(\alpha_{i+1/2,j}^l, \alpha_{i-1/2,j}^l); \quad (95)$$

$$\tilde{\beta}_{i,j}^l = \bar{m}(\Delta_- \alpha_{i-1/2,j}^l, \Delta_+ \alpha_{i-1/2,j}^l) \quad \text{if} \quad |\alpha_{i-1/2,j}^l| \leq |\alpha_{i+1/2,j}^l|; \quad (96)$$

$$\hat{\beta}_{i,j}^l = \bar{m}(\Delta_- \alpha_{i+1/2,j}^l, \Delta_+ \alpha_{i+1/2,j}^l) \quad \text{if} \quad |\alpha_{i-1/2,j}^l| > |\alpha_{i+1/2,j}^l|; \quad (97)$$

$$\gamma_{i+1/2,j}^l = \sigma(\lambda_{i+1/2,j}^l) \begin{cases} (\beta_{i+1,j}^l - \beta_{i,j}^l) / \alpha_{i+1/2,j}^l & \text{if } \alpha_{i+1/2,j}^l \neq 0 \\ 0, & \text{otherwise} \end{cases}; \quad (98)$$

$$\tilde{\gamma}_{i+1/2,j}^l = \tilde{\sigma}(\lambda_{i+1/2,j}^l) \begin{cases} (\tilde{\beta}_{i+1,j}^l - \tilde{\beta}_{i,j}^l) / \alpha_{i+1/2,j}^l & \text{if } \alpha_{i+1/2,j}^l \neq 0 \\ 0, & \text{otherwise} \end{cases}; \quad (99)$$

$$\hat{\gamma}_{i+1/2,j}^l = \hat{\sigma}(\lambda_{i+1/2,j}^l) \begin{cases} (\hat{\beta}_{i+1,j}^l - \hat{\beta}_{i,j}^l) / \alpha_{i+1/2,j}^l & \text{if } \alpha_{i+1/2,j}^l \neq 0 \\ 0, & \text{otherwise} \end{cases}. \quad (100)$$

The same expressions can be extended to the η direction in a straightforward way. To the inviscid implicit cases, the LNI form is applied. The RHS for this algorithm is given by:

$$RHS(YH)_{i,j}^n = -\frac{\Delta_{i,j}}{V_{i,j}} (E_{i+1/2,j}^{UNO} - E_{i-1/2,j}^{UNO}) - \frac{\Delta_{i,j}}{V_{i,j}} (F_{i,j+1/2}^{UNO} - F_{i,j-1/2}^{UNO}). \quad (101)$$

To the explicit viscous case, the Strang method is applied:

$$Q_{i,j}^{n+2} = L_\xi(\Delta t) L_\eta(\Delta t) L_\eta(\Delta t) L_\xi(\Delta t) Q_{i,j}^n \quad (102)$$

The L_ξ operator is defined by

$$L_\xi(\Delta t) Q_{i,j}^n = Q_{i,j}^n - \Delta t (E_{i+1/2,j}^{UNO} - E_{i-1/2,j}^{UNO}). \quad (103)$$

7 Implicit Formulation

7.1 Implicit Scheme to the TVD algorithm of [12] and the UNO algorithm of [25]

In the flux difference splitting cases of [12; 25] algorithms, a Linearized Nonconservative Implicit form is applied which, although the resulting schemes lose the conservative property, they preserve their unconditional TVD properties. Moreover, the LNI form is mainly useful to steady state problems where the conservative property is recovery by these schemes in such condition. This LNI form was proposed by [11].

The LNI form is defined by the following two step algorithm:

$$[I - \Delta t_{i,j} J_{i+1/2,j}^- \Delta_{i+1/2,j} + \Delta t_{i,j} J_{i-1/2,j}^+ \Delta_{i-1/2,j}] \Delta Q_{i,j}^* = [RHS]_{i,j}^n, \quad (101)$$

in the ξ direction;

$$[I - \Delta t_{i,j} K_{i,j+1/2}^- \Delta_{i,j+1/2} + \Delta t_{i,j} K_{i,j-1/2}^+ \Delta_{i,j-1/2}] \Delta Q_{i,j}^{n+1} = \Delta Q_{i,j}^*, \quad (102)$$

in the η direction;

$$Q_{i,j}^{n+1} = Q_{i,j}^n + \Delta Q_{i,j}^{n+1}, \quad (103)$$

where RHS is defined by Eq. (41), if the [12] scheme is being solved, and by Eq. (101), if the [25] scheme is being solved. The difference operators are defined as:

$$\Delta_{i+1/2,j}(\cdot) = (\cdot)_{i+1,j} - (\cdot)_{i,j}, \quad \Delta_{i-1/2,j}(\cdot) = (\cdot)_{i,j} - (\cdot)_{i-1,j}; \quad (104)$$

$$\Delta_{i,j+1/2}(\cdot) = (\cdot)_{i,j+1} - (\cdot)_{i,j}, \quad \Delta_{i,j-1/2}(\cdot) = (\cdot)_{i,j} - (\cdot)_{i,j-1}; \quad (105)$$

As aforementioned, this three-diagonal linear system, composed of a 4x4 block matrices, is solved using LU decomposition and the Thomas algorithm, defined by a block matrix system.

The separated matrices J^+ , J^- , K^+ and K^- are defined as follows:

$$J^+ = R_\xi \text{diag}(D_\xi^+) R_\xi^{-1}, \quad J^- = R_\xi \text{diag}(D_\xi^-) R_\xi^{-1} \quad (106)$$

$$K^+ = R_\eta \text{diag}(D_\eta^+) R_\eta^{-1}, \quad K^- = R_\eta \text{diag}(D_\eta^-) R_\eta^{-1} \quad (107)$$

in which the R_ξ and R_η matrices are defined by Eq. (31) applied to the respective coordinate; and R_ξ^{-1} and R_η^{-1} defined by Eq. (48) applied to the respective coordinate direction.

The diagonal matrices of the [12; 25] schemes are determined by:

$$\text{diag}(D_\xi^+) = \begin{bmatrix} D_1^{\xi,+} & & & & \\ & D_2^{\xi,+} & & & \\ & & D_3^{\xi,+} & & \\ & & & D_4^{\xi,+} & \\ & & & & D_5^{\xi,+} \end{bmatrix} \quad \text{and} \quad \text{diag}(D_\xi^-) = \begin{bmatrix} D_1^{\xi,-} & & & & \\ & D_2^{\xi,-} & & & \\ & & D_3^{\xi,-} & & \\ & & & D_4^{\xi,-} & \\ & & & & D_5^{\xi,-} \end{bmatrix} \quad (108)$$

with the D terms expressed as

$$D_{\xi}^{\pm} = 0.5[\Psi(\lambda_{\xi}^l + \gamma_{\xi}^l) \pm (\lambda_{\xi}^l + \gamma_{\xi}^l)]$$

$$D_{\eta}^{\pm} = 0.5[\Psi(\lambda_{\eta}^l + \gamma_{\eta}^l) \pm (\lambda_{\eta}^l + \gamma_{\eta}^l)], \quad (109)$$

where:

Ψ defined by Eq. (34);

λ_{ξ}^l and λ_{η}^l are the eigenvalues of the Euler equations, determined by Eqs. (21-22), in each coordinate direction;

$$(\gamma_{\xi}^l)_{i+1/2,j} = \begin{cases} [(g_{\xi}^{\cdot})_{i+1,j} - (g_{\xi}^{\cdot})_{i,j}] / (\alpha_{\xi}^l)_{i+1/2,j}, & \text{if } (\alpha_{\xi}^l)_{i+1/2,j} \neq 0.0; \\ 0.0, & \text{if } (\alpha_{\xi}^l)_{i+1/2,j} = 0.0 \end{cases}; \quad (110)$$

$$(\gamma_{\eta}^l)_{i,j+1/2} = \begin{cases} [(g_{\eta}^{\cdot})_{i,j+1} - (g_{\eta}^{\cdot})_{i,j}] / (\alpha_{\eta}^l)_{i,j+1/2}, & \text{if } (\alpha_{\eta}^l)_{i,j+1/2} \neq 0.0; \\ 0.0, & \text{if } (\alpha_{\eta}^l)_{i,j+1/2} = 0.0 \end{cases}; \quad (111)$$

$$(g_{\xi}^{\cdot})_{i,j}^l = \text{signal}_{\xi}^l \text{MAX} \left[0.0, \text{MIN} \left(\sigma_{i+1/2,j}^l \left| (\alpha_{\xi}^l)_{i+1/2,j} \right|, \text{signal}_{\xi}^l \sigma_{i-1/2,j}^l \left| (\alpha_{\xi}^l)_{i-1/2,j} \right| \right) \right]; \quad (112)$$

$$(g_{\eta}^{\cdot})_{i,j}^l = \text{signal}_{\eta}^l \text{MAX} \left[0.0, \text{MIN} \left(\sigma_{i,j+1/2}^l \left| (\alpha_{\eta}^l)_{i,j+1/2} \right|, \text{signal}_{\eta}^l \sigma_{i,j-1/2}^l \left| (\alpha_{\eta}^l)_{i,j-1/2} \right| \right) \right]; \quad (113)$$

$$\sigma^l = 1/2 \Psi^1(\lambda^l) \text{ to steady state simulations.} \quad (114)$$

Finally, $\text{signal}_{\xi}^l = 1.0$ if $(\alpha_{\xi}^l)_{i+1/2,j} \geq 0.0$ and -1.0 otherwise; $\text{signal}_{\eta}^l = 1.0$ if $(\alpha_{\eta}^l)_{i,j+1/2} \geq 0.0$ and -1.0 otherwise.

This implicit formulation to the LHS of the TVD or UNO schemes of [12] and [25], respectively, is second order accurate in space and first order accurate in time due to the presence of the characteristic numerical speed γ associated with the numerical flux function g^{\cdot} . In this case, the algorithms accuracy is definitely second order in space because both LHS and RHS are second order accurate.

7.2 Implicit Scheme to the TVD/ENO algorithms of [21; 24]

For these algorithms, a backward Euler method in time and approximate factorization ADI form can be employed. The factorization in each coordinate direction is presented below:

$$[I + \Delta t_{i,j} \hat{A}_{i-1/2,j}^+ \Delta_- A_{\xi} + \Delta t_{i,j} \hat{A}_{i+1/2,j}^- \Delta_+ A_{\xi}] \Delta Q_{i,j}^* = \text{RHS}_{i,j}^n; \quad (115)$$

$$[I + \Delta t_{i,j} \hat{B}_{i,j-1/2}^+ \Delta_- B_{\eta} + \Delta t_{i,j} \hat{B}_{i,j+1/2}^- \Delta_+ B_{\eta}] \Delta Q_{i,j}^n = \Delta Q_{i,j}^*; \quad (116)$$

$$Q_{i,j}^{n+1} = Q_{i,j}^n + \Delta Q_{i,j}^n. \quad (117)$$

Equations (115-117) lead to standard block three-diagonal inversion procedure. The Thomas algorithm is employed to solve this system.

The matrices above are all defined along this manuscript, being unnecessary repeat them herein.

It is noted that each added high-order term of the right-hand-side operator [Eqs. (66; 81)] is a function of the time step $\Delta t_{i,j}$, and consequently the steady state solutions will depend on the time step.

It is important to emphasize that the RHS of the flux difference splitting implicit schemes present steady state solutions which depend of the time step. With this behavior, the use of large time steps can affect the stationary solutions, as mentioned in [38]. This is an initial study with implicit schemes and improvements in the numerical implementation of these algorithms with steady state solutions independent of the time step is a goal to be reached in future work of both authors.

8 Spatially Variable Time Step

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

In this work were used two types of time step: one to convective flow (Euler equations) and the other to convective plus diffusive flow (Navier-Stokes equations). They are defined as follows:

8.1 Convective Time Step

According to the definition of the CFL number, it is possible to write:

$$\Delta t_{i,j} = \text{CFL}(\Delta s)_{i,j} / c_{i,j}, \quad (118)$$

where CFL is the ‘‘Courant-Friedrichs-Lewy’’ number to provide numerical stability to the scheme; $c_{i,j} = \left[(u^2 + v^2)^{0.5} + a \right]_{i,j}$ is the maximum characteristic speed of information propagation in the calculation domain; and $(\Delta s)_{i,j}$ is a characteristic length of information transport. On a

finite volume context, $(\Delta s)_{i,j}$ is chosen as the minor value found between the minor centroid distance, involving the (i,j) cell and a neighbor, and the minor cell side length.

8.2 Convective + Diffusive Time Step

In this model, the time step is defined according to the [39] model:

$$\Delta t_{i,j} = \left(\frac{\text{CFL}(\Delta t_c \Delta t_v)}{\Delta t_c + \Delta t_v} \right)_{i,j}, \quad (119)$$

with Δt_c being the convective time step and Δt_v being the viscous time step. These quantities are defined as:

$$(\Delta t_c)_{i,j} = \frac{V_{i,j}}{(\lambda_c)_{i,j}}; \quad (120)$$

$$(\lambda_c)_{i,j} = \text{MAX}(\lambda_{i,j-1/2}^{\text{max}}, \lambda_{i+1/2,j}^{\text{max}}, \lambda_{i,j+1/2}^{\text{max}}, \lambda_{i-1/2,j}^{\text{max}}); \quad (121)$$

$$(\lambda^{\text{max}})_{\text{int}} = \left(|u_{\text{int}} n_x + v_{\text{int}} n_y| + a_{\text{int}} \right) S_{\text{int}}; \quad (122)$$

$$(\Delta t_v)_{i,j} = K_v \frac{V_{i,j}}{(\lambda_v)_{i,j}}; \quad (123)$$

$$(p1)_{i,j} = \frac{\gamma^{3/2} M_\infty}{(\text{Re Pr d}_L) V_{i,j}}; \quad (124)$$

$$(p2)_{i,j} = \frac{\mu_{M_{i,j-1/2}}}{\rho_{i,j-1/2}} S_{i,j-1/2}^2 + \frac{\mu_{M_{i+1/2,j}}}{\rho_{i+1/2,j}} S_{i+1/2,j}^2 + \frac{\mu_{M_{i,j+1/2}}}{\rho_{i,j+1/2}} S_{i,j+1/2}^2 + \frac{\mu_{M_{i-1/2,j}}}{\rho_{i-1/2,j}} S_{i-1/2,j}^2; \quad (125)$$

$$(\lambda_v)_{i,j} = (p1 \times p2)_{i,j}, \quad (126)$$

where the interface properties are calculated by arithmetical average, M_∞ is the freestream Mach number and K_v is equal to 0.25, according to [39].

9 Initial and Boundary Conditions

9.1 Initial Condition

To the physical problems studied in this work, freestream flow values are adopted for all properties as initial condition, in the whole calculation domain ([40-41]). Therefore, the vector of conserved variables is defined as:

$$Q_{i,j} = \left\{ 1 \quad M_\infty \cos \alpha \quad M_\infty \sin \alpha \quad \frac{1}{\gamma(\gamma-1)} + 0.5M_\infty^2 \right\}^T, \quad (127)$$

being α the flow attack angle.

9.2 Boundary Conditions

The boundary conditions are basically of three types: solid wall, entrance and exit. The far field condition is a case of entrance and exit frontiers. These conditions are implemented in special cells named ghost cells.

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

$$n_x = \Delta y / \sqrt{\Delta x^2 + \Delta y^2}; \quad (128)$$

$$n_y = -\Delta x / \sqrt{\Delta x^2 + \Delta y^2}; \quad (129)$$

where, for the $(i+1/2,j)$ interface:

$$\Delta x = x_{i+1,j+1} - x_{i+1,j}; \quad (130)$$

$$\Delta y = y_{i+1,j+1} - y_{i+1,j}. \quad (131)$$

Hence, the ghost cell velocity components are written as:

$$u_g = (n_y^2 - n_x^2)u_r - (2n_x n_y)v_r; \quad (132)$$

$$v_g = -(2n_x n_y)u_r + (n_y^2 - n_x^2)v_r, \quad (133)$$

with “g” related with ghost cell and “r” related with real cell. To the viscous case, the boundary condition imposes that the ghost cell velocity components be equal to the real cell velocity components, with the negative signal:

$$u_g = -u_r; \quad (134)$$

$$v_g = -v_r, \quad (135)$$

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

(b) Entrance condition:

(b.1) Subsonic flow: Three properties are specified and one is extrapolated, based on analysis of information propagation along characteristic directions in the calculation domain ([41]). In other words, three characteristic directions of information propagation point inward the computational domain and should be specified. Only the characteristic direction associated to the “ (q_n-a) ” velocity cannot be specified and should be determined by interior information of the calculation domain. The pressure was the extrapolated variable from the real neighbor volume, to the studied problems. Density and velocity components had their values determined by the freestream flow properties. The total energy per unit fluid volume is determined by the state equation of a perfect gas.

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

(c) Exit condition:

(c.1) Subsonic flow: Three characteristic directions of information propagation point outward the computational domain and should be extrapolated from interior information ([41]). The characteristic direction associated to the “ (q_n-a) ” velocity should be specified because it penetrates the calculation domain. In this case, the ghost volume’s pressure is specified by its freestream value. Density and velocity components are extrapolated and the total energy is obtained by the state equation of a perfect gas.

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

10 Results

Tests were performed in a personal computer (notebook) with Pentium dual core processor of 2.20GHz of clock and 2.0Gbytes of RAM memory. Converged results occurred to 3 orders of reduction in the value of the maximum residual. The maximum residual is defined as the maximum value obtained from the discretized conservation equations. The value used to γ was 1.4. To all problems, the attack or entrance angle was adopted equal to 0.0° .

The physical problems to be studied are the shock impinging a wall, causing a reflection effect; and the supersonic flow along a compression corner, in both inviscid and viscous cases.

10.1 Shock Reflection Problem - Inviscid

The first problem to be studied is the shock reflection problem. It was suggested by [38] and is described by an oblique shock wave impinging a wall and reflecting in direction to the far field. Figure 2 exhibits the computational domain.

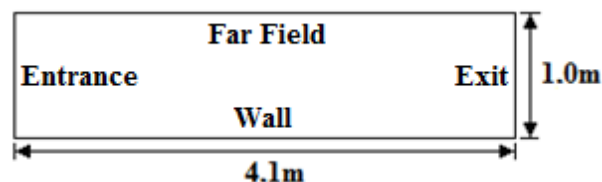


Figure 2. Computational domain to the reflection shock problem.

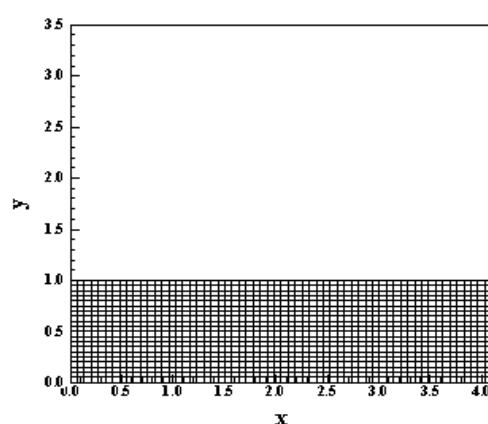


Figure 3. Mesh configuration (61x21).

Figure 3 shows the mesh configuration to this problem. It is composed of 1,200 cells and 1,281 nodes or in a finite difference representation 61x21 points.

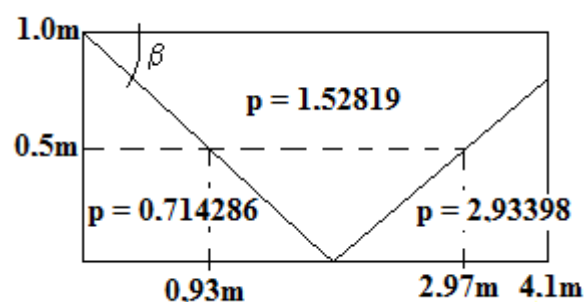


Figure 4. Sketch of the shock reflection solution.

The physical problem presents an oblique shock wave generated at the far field, with shock angle β of 29° , impinging a wall. After the interaction with the wall, the reflected shock is directed to the far field. The freestream Mach number to this simulation is 2.9, a moderate supersonic flow. The analytical solution, in terms of pressure contour lines as well in terms of pressure coefficient, is

presented and serves as numerical comparison. The sketch of the pressure contour lines and the pressure coefficient distribution at $y = 0.5\text{m}$ are presented in Figs. 4 and 5, respectively.

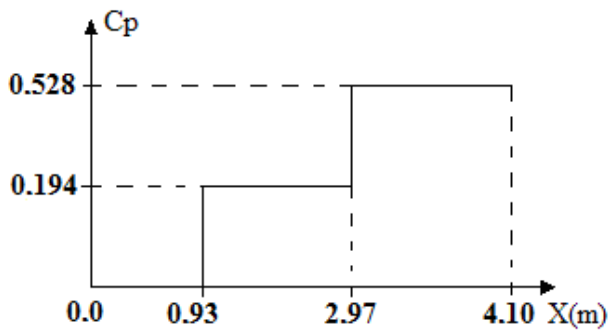


Figure 5. Pressure coefficient distribution at $y = 0.5\text{m}$.

In the figures below is employed the following nomenclature: TVD2 (Total Variation Diminishing, second order accurate in space), ENO2 (Essentially Nonoscillatory, second order accurate in space), TVD3 (Total Variation Diminishing, third order accurate in space), ENO3 (Essentially Nonoscillatory, third order accurate in space) and UNO3 (Uniformly Nonoscillatory, third order accurate in space).

Figure 6 shows the pressure contours obtained by the [12] TVD algorithm. It is possible to note that the region of the incident shock and the region of the reflected shock agree with the analytical results. Only the intermediate region between the two shock waves is in disaccord with the analytical solution. The error in this prediction is about 1.84%, the same error found in the first part of this study [42].

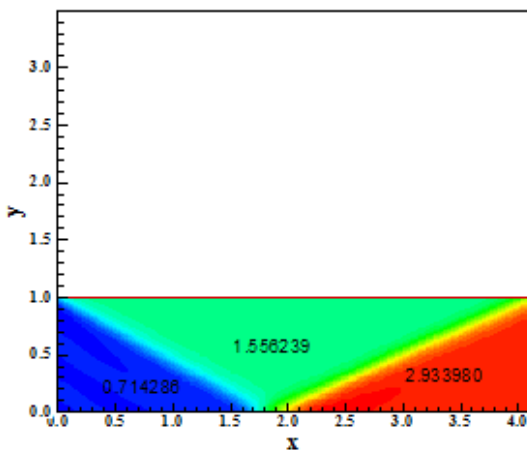


Figure 6. Pressure contours ([12]-TVD2).

In Figures (7-8) are exhibited the pressure contours of both TVD2 and ENO2 solutions due to [21] algorithm. As can be observed the two

numerical solutions present a good comparison in the shock regions, only presenting error at the intermediate region between shock waves. The error found is again 1.84%. The shock waves thicknesses are also thicker in relation to the solution of [12], emphasizing that the [12] algorithm presents the best description of such shock waves.

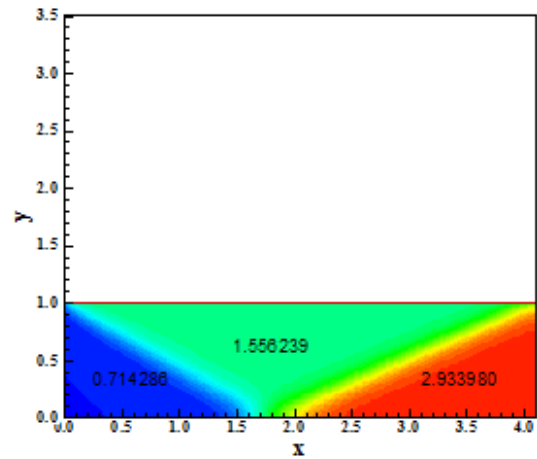


Figure 7. Pressure contours ([21]-TVD2).

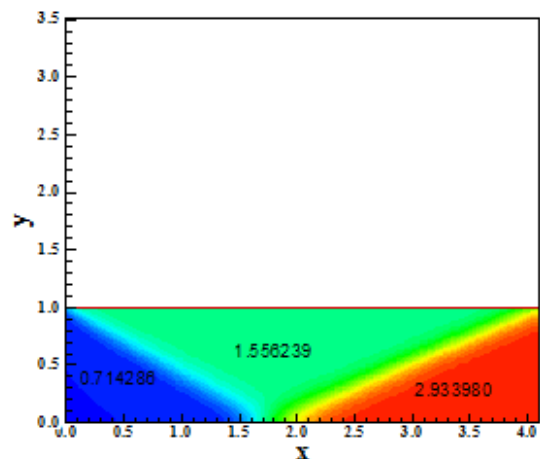


Figure 8. Pressure contours ([21]-ENO2).

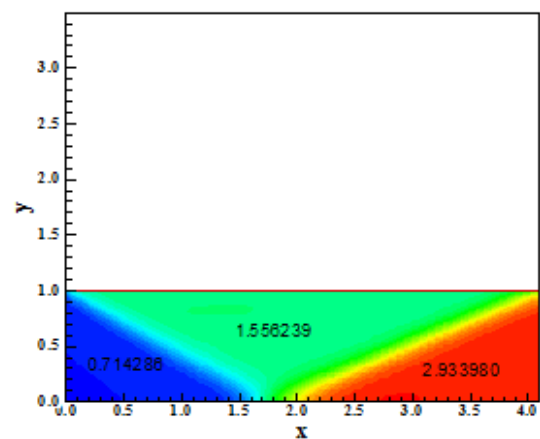


Figure 9. Pressure contours ([24]-TVD3).

Figures (9-10) show the pressure contours obtained by the [24] algorithm in its third-order accuracy version, for both TVD and ENO formulations. As can be seen, the shock waves are well captured, the thicknesses of the shock waves are thicker than the [12] solution, and the intermediate region between shock waves presents an error of 1.84% in relation to the analytical solution.

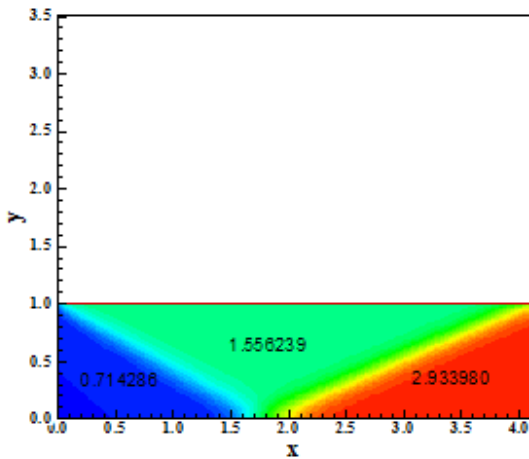


Figure 10. Pressure contours ([24]-ENO3).

Finally, Figure 11 presents the results of the [25] algorithm, in its UNO3 version. The shock waves thicknesses are thicker than the [12] solution, but are well captured. Only the intermediate region between shocks is in disagreement with the analytical solution, presenting an error of 1.84%.

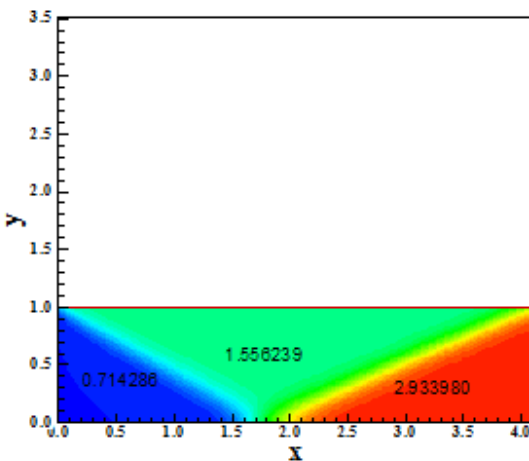


Figure 11. Pressure contours ([25]-UNO3).

As seen in Figure 12, the six (6) Cp distributions relative to the six (6) studied algorithms are plotted and compared. Notably, the solution presented by the [12] scheme is the closest with the analytical solution, demonstrating the excellent properties of

this TVD2 scheme. All solutions are free of oscillations. Hence, the best solution on this problem is due to [12].

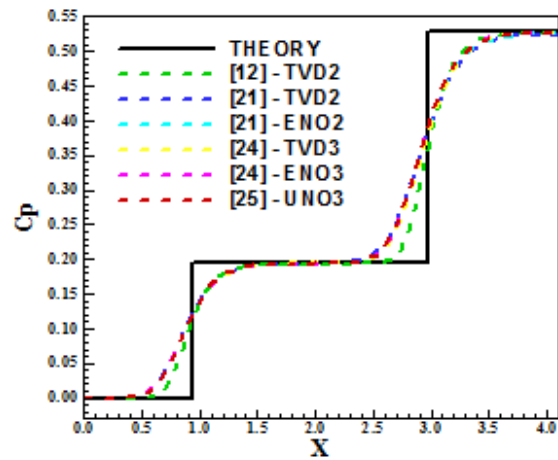


Figure 12. Cp distributions at y = 0.5m

10.2 Compression Corner Problem - Inviscid

The compression corner configuration is described in Fig. 13. The corner inclination angle is 10° . An algebraic mesh of 70×50 points or composed of 3,381 rectangular cells and 3,500 nodes was used and is shown in Fig. 14. The points are equally spaced in both directions.

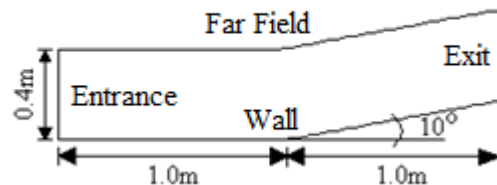


Figure 13. Computational domain to the compression corner problem.

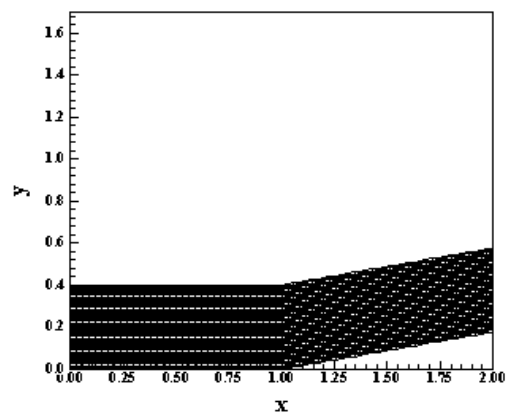


Figure 14. Mesh configuration (70x50).

This problem consists in a moderate supersonic flow impinging a compression corner, where an

oblique shock wave is generated. The freestream Mach number is equal to 3.0. The solutions are compared with the oblique shock wave theory results.

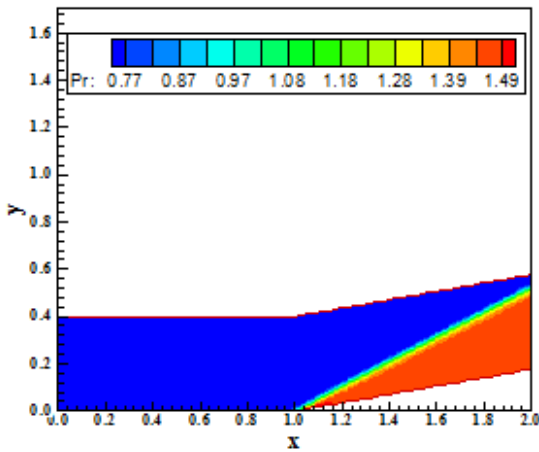


Figure 15. Pressure contours ([12]-TVD2).

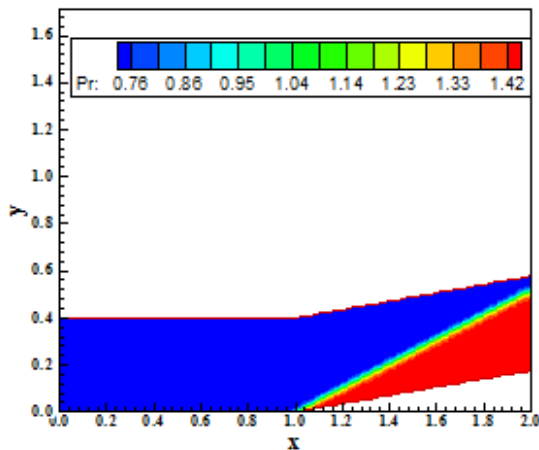


Figure 16. Pressure contours ([21]-TVD2).

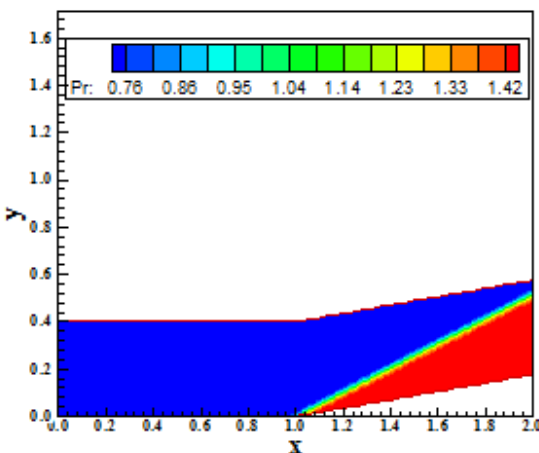


Figure 17. Pressure contours ([21]-ENO2).

Figure 15 exhibits the pressure contours obtained by the [12] TVD2 scheme. As can be seen, a

pressure peak appears at the corner beginning and it is reflected in the value 1.49 of the pressure legend, far from high than the other solutions. Due to this behavior in the pressure contours, the wall pressure distribution of [12] presents this peak.

Figures (16-17) show the pressure contours obtained by the [21] scheme, in its TVD and ENO second order accurate versions. Both solutions are free of oscillations, presenting a good transition between smooth and discontinuity regions.

Figures (18-19) exhibit the pressure contours obtained from [24], in its TVD and ENO versions of third-order accuracy, respectively. The solutions are free of oscillations and present good capture of shock discontinuity. The TVD3 version of the [24] algorithm presents the smallest shock wave thickness.

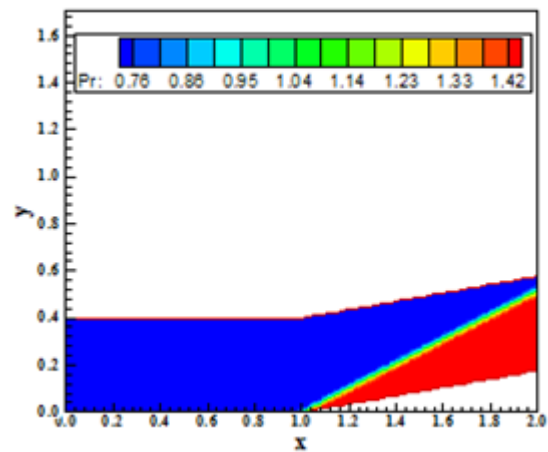


Figure 18. Pressure contours ([24]-TVD3).

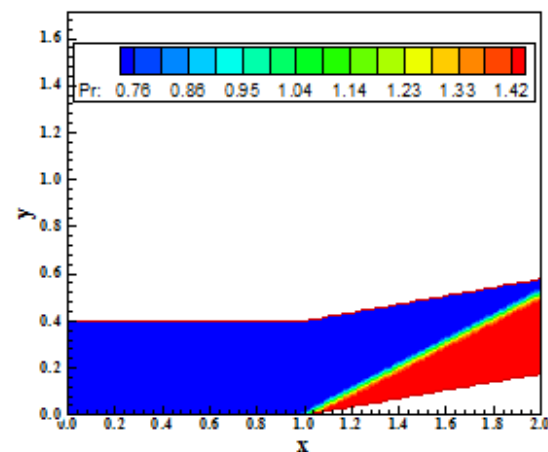


Figure 19. Pressure contours ([24]-ENO3).

Figure 20 exhibits the pressure contours resulting from the [25] UNO3 scheme. As can be seen, no oscillations and good capture of the shock discontinuity are observed.

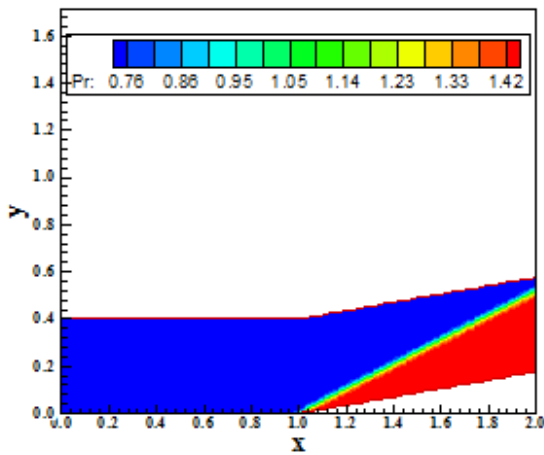


Figure 20. Pressure contours ([25]-UNO3).

Figure 21 shows the wall pressure distributions obtained by all schemes. They are compared with the oblique shock wave theory results. The best solution is that obtained with the [24] TVD3 scheme.

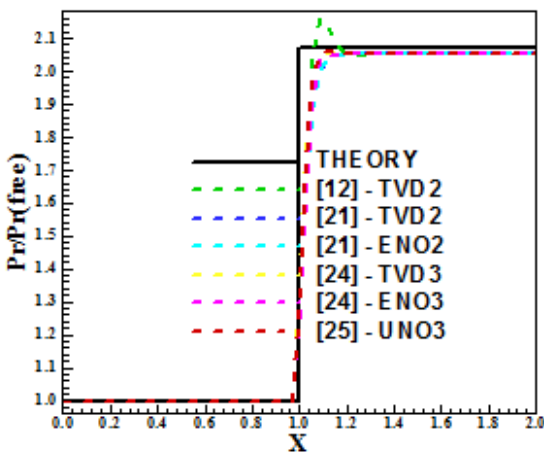


Figure 21. Wall pressure distributions.

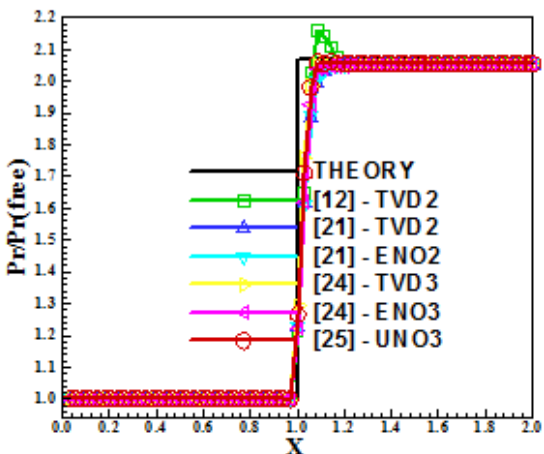


Figure 22. Wall pressure distributions.

In Figure 22, it is exhibited all pressure distributions with symbols to identify the number of cells necessary to capture the shock discontinuity. All schemes capture the discontinuity using three (3) cells, which is a good result for high resolution schemes.

One way to quantitatively verify if the solutions generated by each scheme are satisfactory consists in determining the shock angle of the oblique shock wave, β , measured in relation to the initial direction of the flow field. [43] (pages 352 and 353) presents a diagram with values of the shock angle, β , to oblique shock waves. The value of this angle is determined as function of the freestream Mach number and of the deflection angle of the flow after the shock wave, ϕ . To the compression corner problem, $\phi = 10^\circ$ (ramp inclination angle) and the freestream Mach number is 3.0, resulting from this diagram a value to β equals to 27.5° . Using a transfer in Figures 15 to 20, it is possible to obtain the values of β to each scheme, as well the respective errors, shown in Tab. 2. As can be observed, the [24] algorithm, in its ENO3 version, and the [25] UNO3 algorithm have yielded the best results. Errors less than 2.00% were observed in all solutions.

Table 2. Shock angle and percentage errors.

Algorithm	β ($^\circ$)	Error (%)
[12] – TVD2	27.4	0.36
[21] – TVD2	27.0	1.82
[21] – ENO2	27.4	0.36
[24] – TVD3	27.8	1.09
[24] – ENO3	27.5	0.00
[25] – UNO3	27.5	0.00

10.3 Compression Corner Problem - Viscous

To the viscous case, it was chosen the compression corner problem again. The computational domain and the mesh configuration are described in Figs. 23 and 24, respectively. The mesh is composed of 7,761 rectangular cells and 8,000 nodes on a finite volume context (equivalent to a mesh of 200×40 points in finite differences).

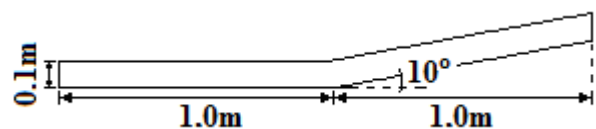


Figure 23. Computational domain to the problem of the compression corner for viscous simulation.

In this viscous problem, the flow is compressed at the corner region and a detached boundary layer is characterized. A circulation bubble is formed at this region. The points of detachment and reattachment are, respectively, 0.90m and 1.10m.

The initial condition to this problem considers a freestream Mach number of 3.0. The Reynolds number was estimated in 1.688×10^4 , according to [34], considering the characteristic length of 0.00305m and an altitude of 20,000m.

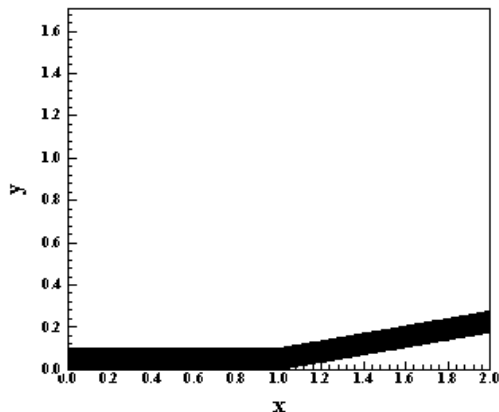


Figure 24. Mesh configuration (200x40).

Figures (25-27) show the boundary layer separation at the corner wall, obtained by the [21] and [24] algorithms. The [12] TVD2, the [24] TVD3 and [25] UNO3 did not present converged solutions. In Figure 25, presenting the solution obtained with the [21] TVD2 scheme, the detachment and reattachment points are positioned, respectively, at $x = 0.96\text{m}$ and $x = 1.09\text{m}$, as seen in Fig. 28. A small circulation bubble is formed at this region, according to the expected behavior. Considering $x = 0.90\text{m}$ and $x = 1.10\text{m}$ as the correct values to detachment and reattachment points, the above results can be admitted as good.

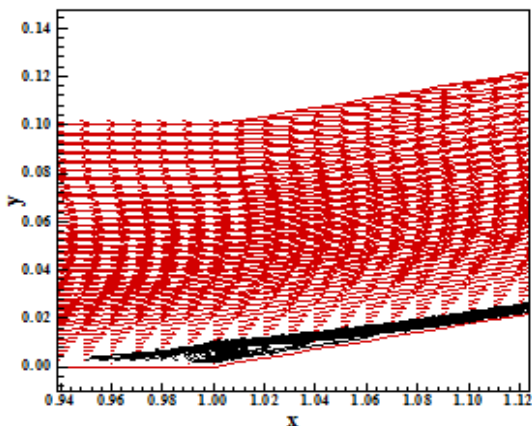


Figure 25. Boundary layer separation and circulation bubble formation ([21]-TVD2).

In Figure 26, presenting the solution obtained with the [21] ENO2 scheme, the detachment is positioned at $x = 0.80\text{m}$ and the reattachment is positioned at $x = 1.60\text{m}$, as seen in Fig. 29. In this case, a great circulation bubble is formed, which disagrees with the expected solution.

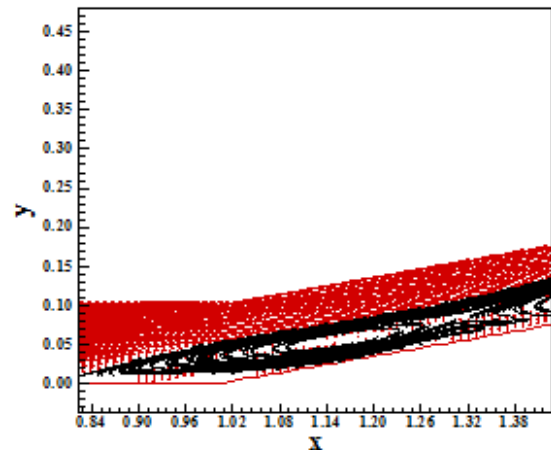


Figure 26. Boundary layer separation and circulation bubble formation ([21]-ENO2).

In Figure 27, the solution obtained with the [24] ENO3 algorithm is shown. The detachment and reattachment points are 0.80m and 1.80m , as seen in Fig. 30. A great circulation bubble is formed, opposed to the expected boundary layer behavior.

It is clear from these figures that the best solution is obtained with the [21] TVD2 scheme, in terms of detachment and reattachment points locations.

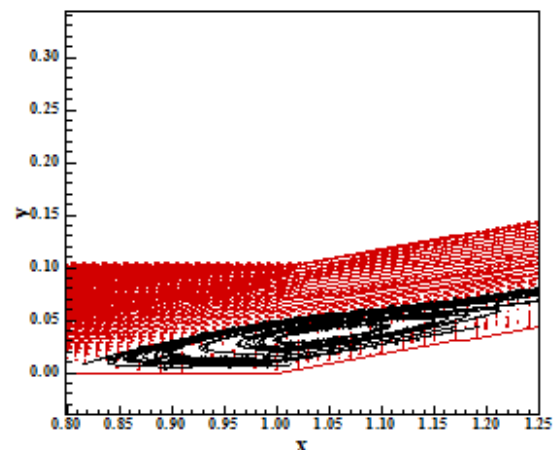


Figure 27. Boundary layer separation and circulation bubble formation ([24]-ENO3).

Figure 28 exhibits the skin friction coefficient distribution, obtained by the [21] TVD2 scheme. As can be observed, the small separation region at the

corner wall is appropriately captured by the numerical algorithm.

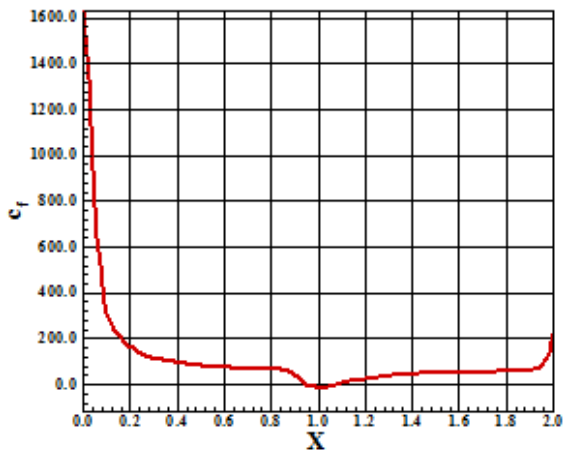


Figure 28. Skin friction coefficient distribution at wall ([21]-TVD2).

Figure 29 shows the skin friction coefficient distribution at the corner wall, obtained by the [21] ENO2 algorithm. It is notable the enormous boundary layer separation formed at the corner wall. This separation develops further downstream from the corner. The points of detachment and reattachment of the boundary layer are clearly pointed out.

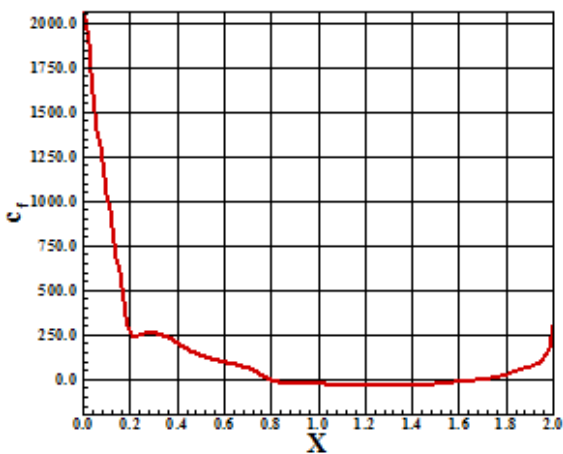


Figure 29. Skin friction coefficient distribution at wall ([21]-ENO2).

In Figure 30, the skin friction coefficient distribution to the [24] ENO3 algorithm is shown. It also captures a circulation bubble greater than that obtained with the [21] TVD2 algorithm. Hence, in terms of circulation bubble formation, an actual physical phenomenon, the best solution is due to the [21] TVD2 scheme.

Figure 31 exhibits the pressure distribution along the corner wall, obtained by the three converged solutions. As can be seen, the best pressure distribution curve is that obtained by the [24] ENO3 scheme, presenting a solution closest to the experimental results of [44] and the numerical results of [45], mainly at the range 1.50m and 1.60m.

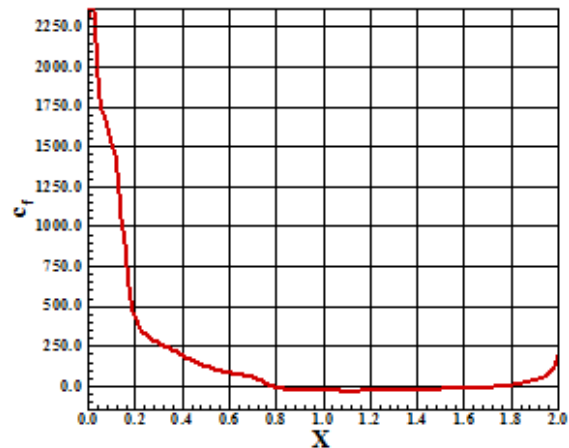


Figure 30. Skin friction coefficient distribution at wall ([24]-ENO3).

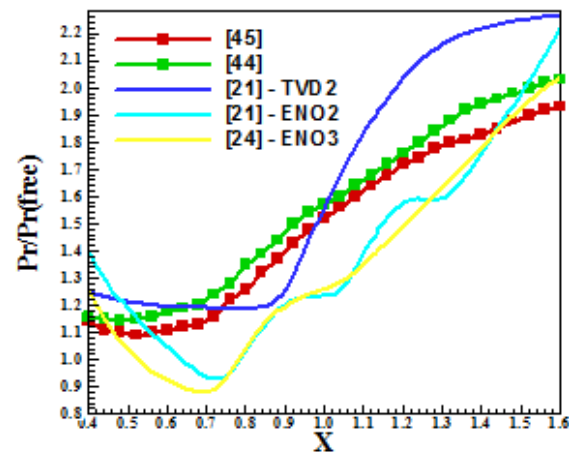


Figure 31. Wall pressure distributions.

10.4 Conclusion of this work

Concluding this analyze, the best algorithm was the [24] ENO3 scheme, presenting good pressure distributions in the shock reflection and compression corner problems, in the inviscid case. Moreover, this scheme also presented the best value to the shock angle of the oblique shock wave, with 0.00% of error. Finally, this algorithm also provides the best wall pressure distribution in the compression corner viscous problem. Hence, this algorithm is the selected one to describe the three

physical problems studied herein, as comparing these six (6) numerical schemes.

10.5 Conclusion of this study

Comparing the results obtained in this work with those obtained in the first part of this study, it is possible to conclude that the [24] ENO3 algorithm is the best as involving all thirteen (13) schemes studied in Part I and the six (6) schemes studied herein. The smooth pressure distributions at the reflection shock problem and at the corner wall, in the inviscid case, rendered the distinction of best algorithm to the ENO3 one. Even not yielding the best pressure distribution, which was better captured by the [46] TVD algorithm, the ENO3 scheme presented a reasonable pressure distribution in the viscous compression corner problem; so, ratifying its choice as the reference scheme to the problems studied herein.

11 Conclusion

In the present work, second part of this study, the [12] TVD, the [21] TVD/ENO, the [24] TVD/ENO, and the [25] UNO schemes are implemented, on a finite volume context and using a structured spatial discretization, to solve the Euler and Navier-Stokes equations in the two-dimensional space. All schemes are high resolution flux difference splitting ones, based on the concept of Harten's modified flux function. The [12] is a TVD second order accurate in space and first order accurate in time algorithm. [21] is a TVD/ENO second order accurate in space and first order accurate in time algorithm. The [24] is a TVD/ENO third order accurate in space and first order accurate in time algorithm. Finally, the [25] is a UNO third order accurate in space and first order accurate in time algorithm. An implicit formulation is employed to solve the Euler equations, whereas a time splitting or Strang methods, explicit methods, are used to solve the Navier-Stokes equations. A Linearized Nonconservative Implicit LNI form or an approximate factorization ADI method is employed by the schemes. The algorithms are accelerated to the steady state solution using a spatially variable time step, which has demonstrated effective gains in terms of convergence rate ([32-33]). All schemes are applied to the solution of physical problems of the supersonic shock reflection at the wall and the supersonic flow along a compression corner, in the inviscid case, whereas in the laminar viscous case, the supersonic flow along a compression corner is solved. The results have demonstrated that the [24]

ENO algorithm, third order accurate in space, has presented the best solutions, in the two works of this study.

References:

- [1] A. Harten, A High Resolution Scheme for the Computation of Weak Solutions of Hyperbolic Conservation Laws, *Journal of Computational Physics*, Vol. 49, 1983, pp. 357-393.
- [2] A. Harten, On a Class of High Resolution Total-Variation-Stable Finite-Difference Schemes, *SIAM Journal of Numerical Analysis*, Vol. 21, 1984, pp. 1-23.
- [3] B. Van Leer, Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method, *Journal of Computational Physics*, Vol. 32, 1979, pp. 101-136.
- [4] P. Colella, and P. R. Woodward, The Piecewise-Parabolic Method (PPM) for Gas-Dynamical Simulations, *LBL Rept. 14661*, July, 1982.
- [5] P. L. Roe, Some Contributions to the Modelling of Discontinuous Flows, *Lectures in Applied Mathematics*, Vol. 22, Pt. II, 1985, pp. 163-193.
- [6] S. Osher, Shock Modelling in Transonic and Supersonic Flow, *Recent Advances in Numerical Methods in Fluids*, Vol.4, *Advances in Computational Transonics*, edited by W. G. Habashi, Pineridge Press, Swansea, U. K., 1985.
- [7] H. C. Yee, Generalized Formulation of a Class of Explicit and Implicit TVD Schemes, *NASA TM-86775*, 1985.
- [8] P. L. Roe, Generalized Formulation of TVD Lax-Wendroff Schemes, *ICASE Rept. 84-53*, Oct., 1984.
- [9] S. F. Davis, TVD Finite Difference Schemes and Artificial Viscosity, *ICASE Rept. 84-20*, June, 1984.
- [10] H. C. Yee, On the Implementation of a Class of Upwind Schemes for Systems of Hyperbolic Conservation Laws, *NASA TM 86839*, September, 1985.
- [11] 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.
- [12] H. C. Yee, and A. Harten, Implicit TVD Schemes for Hyperbolic Conservation Laws in Curvilinear Coordinates, *AIAA Journal*, Vol. 25, No. 2, 1987, pp. 266-274.

- [13] A. Harten, and S. Osher, Uniformly High Order Accurate Non-Oscillatory Schemes, I, *SIAM J. Num. Anal.*, Vol. 24, No. 2, 1987, pp. 279-309.
- [14] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, Some results on Uniformly High-Order Accurate Essentially Non-Oscillatory Schemes, *Advances in Numerical Analysis and Applied Mathematics*, edited by J. C. South, Jr. and M. Y. Hussaini, *ICASE Rept. 86-18*, 1986; also *Journal of Applied Num. Mathematics*, Vol. 2, 1986, pp. 347-367.
- [15] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III, *ICASE Rept. 86-22*, 1986.
- [16] S. K. Godunov, A Finite Difference Method for the Numerical Computation of Discontinuous Solutions of the Equations to Fluid Dynamics, *Math. Sb.*, Vol. 47, 1958, pp. 271-290.
- [17] P. Colella, and P. R. Woodward, The Piecewise-Parabolic Method (PPM) for Gas-Dynamical Simulations, *Journal of Computational Physics*, Vol. 54, No. 1, 1984, pp. 174-201.
- [18] A. Harten, High Resolution Schemes for Hyperbolic Conservation Laws, *Journal of Computational Physics*, Vol. 49, No. 3, 1983, pp. 357-393.
- [19] P. K. Sweby, High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws, *SIAM Journal of Numerical Analysis*, Vol. 21, No. 5, 1984, pp. 995-1011.
- [20] E. S. G. Maciel, Explicit and Implicit TVD High Resolution Schemes in 2D, *WSEAS TRANSACTIONS ON APPLIED AND THEORETICAL MECHANICS*, Vol. 7, No. 3, 2012, pp. 182-209.
- [21] J. Y. Yang, Uniformly Second-Order-Accurate Essentially Nonoscillatory Schemes for the Euler Equations, *AIAA Journal*, Vol. 28, No. 12, 1990, pp. 2069-2076.
- [22] J. Y. Yang, C. K. Lombardi, and D. Bershader, Numerical Simulation of Transient Inviscid Shock Tube Flows, *AIAA Journal*, Vol. 25, No. 2, 1987, pp. 245-251.
- [23] J. Y. Yang, Y. Liu, and H. Lomax, Computation of Shock-Wave Reflection by Circular Cylinder, *AIAA Journal*, Vol. 25, No. 5, 1987, pp. 636-689.
- [24] J. Y. Yang, Third-Order Nonoscillatory Schemes for the Euler Equations, *AIAA Journal*, Vol. 29, No. 10, 1991, pp. 1611-1618.
- [25] J. Y. Yang, and C. A. Hsu, High Resolution, Nonoscillatory Schemes for Unsteady Compressible Flows, *AIAA Journal*, Vol. 30, No. 6, 1992, pp. 1570-1575.
- [26] P. L. Roe, Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes, *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357-372.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] J. Douglas, and J. E. Gunn, A General Formulation of Alternating Direction Methods, *Numerische Mathematik*, Vol. 6, 1964, pp. 428-453.
- [31] N. N. Yanenko, *The Method of Fractional Steps*, Springer Verlag, N.Y., EUA, 1971.
- [32] E. S. G. Maciel, Analysis of Convergence Acceleration Techniques Used in Unstructured Algorithms in the Solution of Aeronautical Problems – Part I, *Proceedings of the XVIII International Congress of Mechanical Engineering (XVIII COBEM)*, Ouro Preto, MG, Brazil, 2005. [CD-ROM]
- [33] E. S. G. Maciel, Analysis of Convergence Acceleration Techniques Used in Unstructured Algorithms in the Solution of Aerospace Problems – Part II, *Proceedings of the XII Brazilian Congress of Thermal Engineering and Sciences (XII ENCIT)*, Belo Horizonte, MG, Brazil., 2008 [CD-ROM]
- [34] R. W. Fox, and A. T. McDonald, *Introdução à Mecânica dos Fluidos*, Ed. Guanabara Koogan, Rio de Janeiro, RJ, Brazil, 632p, 1988.
- [35] G. Strang, On the Construction and Comparison of Difference Schemes, *SINUM*, Vol. 5, 1968, pp. 506-517.
- [36] J. Y. Yang, Second- and Third-Order Upwind Flux Difference Splitting Schemes for the Euler Equations, *NASA TM-85959*, July, 1984.
- [37] B. Van Leer, Towards the Ultimate Conservative Difference Scheme II, Monotonicity and Conservation Combined in a Second Order Scheme, *Journal of Computational Physics*, Vol. 14, No. 2, 1974, pp. 361-370.

- [38] H. C. Yee, R. F. Warming, and A. Harten, A High-Resolution Numerical Technique for Inviscid Gas-Dynamic Problems with Weak Solutions, *Proceedings of the 8th International Conference on Numerical Methods in Fluid Dynamics*, E. Krause, Editor, *Lecture Notes in Physics*, Springer-Verlag, Berlin, Germany, Vol. 170, 1982, pp. 546-552.
- [39] D. J. Mavriplis, and A. Jameson, Multigrid Solution of the Navier-Stokes Equations on Triangular Meshes, *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1415-1425.
- [40] A. Jameson, and D. Mavriplis, Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh, *AIAA Journal*, Vol. 24, 1986, pp. 611-618.
- [41] E. S. G. Maciel, Simulação Numérica de Escoamentos Supersônicos e Hipersônicos Utilizando Técnicas de Dinâmica dos Fluidos Computacional, *Doctoral Thesis*, ITA, CTA, São José dos Campos, SP, Brazil, 258p, 2002.
- [42] E. S. G. Maciel, and E. M. Ferreira, TVD and ENO Applications to Supersonic Flows in 2D – Part I, Submitted to *WSEAS TRANSACTIONS ON THEORETICAL AND APPLIED MECHANICS*, 2012 (under review).
- [43] J. D. Anderson Jr., *Fundamentals of Aerodynamics*, McGraw-Hill, Inc., EUA, 563p, 1984.
- [44] J. E. Carter, J., Numerical Solutions of the Supersonic, Laminar Flow over a Two-Dimensional Compression Corner, *Lecture Notes in Physics*, Springer-Verlag, Vol. 19, 1973, pp. 69-78.
- [45] Y. Kergaravat, and D. Knight, A Fully Implicit Navier-Stokes Algorithm for Unstructured Grids Incorporating a Two-Equation Turbulence Model, *AIAA Paper 96-0414*, 1996.
- [46] P. D. Lax, and B. Wendroff, Difference Schemes for Hyperbolic Equations with High Order of Accuracy, *Communications on Pure and Applied Mathematics*, Vol. XVII, 1964, pp. 381-398.