

A hybrid pressure–density-based algorithm for the Euler equations at all Mach number regimes

C. M. Xisto^{1,*}, J. C. Páscoa¹, P. J. Oliveira¹ and D. A. Nicolini²

¹*Dep. de Eng. Electromecânica, Universidade da Beira Interior, R. Mq.s D Ávila e Bolama, 6201-001 Covilhã, Portugal*

²*Directorate of Science and Robotic Exploration, European Space Agency, ESTEC, Keplerlaan 1, 2201AZ Noordwijk, The Netherlands*

SUMMARY

In the present work, we propose a reformulation of the fluxes and interpolation calculations in the PISO method, a well-known pressure-correction solver. This new reformulation introduces the $AUSM^+ - up$ flux definition as a replacement for the standard Rhie and Chow method of obtaining fluxes and central interpolation of pressure at the control volume faces. This algorithm tries to compatibilize the good efficiency of a pressure based method for low Mach number applications with the advantages of $AUSM^+ - up$ at high Mach number flows. The algorithm is carefully validated using exact solutions. Results for subsonic, transonic and supersonic axisymmetric flows in a nozzle are presented and compared with exact analytical solutions. Further, we also present and discuss subsonic, transonic and supersonic results for the well known bump test-case. The code is also benchmarked against a very tough test-case for the supersonic and hypersonic flow over a cylinder. Copyright © 2011 John Wiley & Sons, Ltd.

Received 29 July 2011; Revised 8 October 2011; Accepted 16 October 2011

KEY WORDS: finite volume; Euler equations; AUSM; PISO; compressible flow; arbitrary Mach number

1. INTRODUCTION

In the field of aeronautical and aerospace engineering, there is often the need for solving complex flow problems that involve a wide range of Mach numbers. In these problems, a near incompressible regime may coexist within a larger supersonic flow domain. Thus, it is important that the algorithms developed are capable of solving flow problems encompassing the various Mach number regimes. To that purpose, both pressure-based and density-based algorithms have been derived.

During the development of pressure-based methods, great research efforts have been made to devise algorithms capable of solving problems at all Mach number regimes [1–5]. These methods were initially developed for incompressible flow predictions (e.g., Patankar [6]) and have been subsequently extended to high Mach number flows. Moukalled and Darwish [7] developed a pressure-based Mach uniform algorithm that uses a normalized variable and space formulation (NVSF) methodology to bound the convective fluxes and a high resolution scheme for the calculation of the interface density to improve shock wave resolution in high speed flows. Issa and Javareshkian [1] extended the semi-implicit method for pressure-linked equations to high Mach number flows using total variation diminishing and flux difference schemes. To improve the convergence rate on fine grids, Bressloff [3] developed a parallel implementation of the PISO method for both compressible and incompressible flows. Many authors, such as Politis and Giannakoglou [8],

*Correspondence to: C. M. Xisto, Dep. de Eng. Electromecânica, Universidade da Beira Interior, R. Mq.s D Ávila e Bolama, 6201-001 Covilhã, Portugal.

†E-mail: xisto@ubi.pt

Thakur and Wright [9] have applied their compressible pressure methods to complex geometries such as compressors and gas turbines.

At first sight, the extension from the incompressible to the compressible regime may seem trivial, because one only needs to modify the Navier–Stokes equations so that a variable density is accounted for. However, one must ensure that pressure acts both on velocity and density [7]. Hence, density needs also to be corrected leading to a more complex pressure correction equation. It is crucial for the development of such an algorithm that one needs to understand the role of pressure in compressible fluid flow. In the low Mach number limit, where density becomes constant, the role of pressure is to act on velocity through continuity so that conservation of mass is satisfied [7]. In this case, the relation of pressure and density is very weak; consequently, the continuity equation cannot be used to obtain density. The continuity equation is then used to build an equation for pressure or pressure correction. On the other hand, in the hypersonic limit, the changes in pressure significantly affect density. In this case, the pressure and density are related through an equation of state, and the continuity equation can therefore be used to obtain density.

Density-based methods were initially developed for high Mach number flows, and one example is given by the advection upstream splitting method (AUSM) of Liou and Steffen [10]. Density-based methods were also extended to arbitrary Mach number through the development of low Mach number formulations. This was carried out mainly through the use of local precondition techniques, modifying the characteristics of the governing equations [11]. Edwards and Liou [12, 13] also extended the AUSM⁺ method [14] to low Mach number flows, through the use of preconditioned speed of sound and Mach numbers, acting on the weight equations and with the introduction of a diffusion term in the mass flux equation.

The diffusion term introduced by Edwards and Liou was needed to ensure the pressure–velocity coupling in low speed flows. This term has the same purpose of the pressure smoothing term that was introduced by the Rhie and Chow interpolation technique [15]. The Rhie and Chow interpolation technique is usually applied if a collocated arrangement of variables is adopted. This arrangement is very useful mainly because it allows for a simple extension of the code to unstructured grids. However, Nerinckx *et al.* [16] demonstrated that the Rhie–Chow procedure presents bad results in the presence of large pressure gradients. This situation is quite common in supersonic flows and appears in the form of shock waves.

In our earlier work [17], we noticed that this problem causes a decoupling between pressure and velocity at high Mach number flows, which result in unrealistic wiggles. So, we decided to abandon the Rhie–Chow interpolation technique and find an alternative approach, similar to that proposed by Nerinckx *et al.*. Albeit these, authors only use the AUSM⁺ flux definitions in conjunction with the semi-implicit method for pressure-linked equations, to calculate transonic and subsonic flows [16]. The proposed algorithm is based on the PISO method [18], but we apply the AUSM⁺ – *up* [19] flux definition as a replacement for the standard Rhie and Chow method, to obtain the fluxes and central interpolations for pressure at the control volume faces. This algorithm tries to compatibilize the good efficiency of a pressure-based method for low Mach number applications, with the advantages of AUSM⁺ – *up* at high Mach number flows.

In the following sections, we will present the governing equations, and then, we will demonstrate how the AUSM flux definitions can be incorporated in the PISO algorithm. In terms of validation, detailed results for subsonic, transonic and supersonic axisymmetric flows in a nozzle are presented and compared with exact solutions. We also present and discuss subsonic, transonic and supersonic results for the well-known bump case. Finally, we end with the calculation of a 2D supersonic flow over a cylinder; this is known for being a very tough case even for density-based solvers [14].

2. GOVERNING EQUATIONS

Our long term goal is to develop a method capable of solving the compressible magnetohydrodynamic (MHD) equations for arbitrary Mach number flows. In an earlier paper, we reported on the implementation of a 3D compressible MHD solver for transonic MHD flows using the Rhie and Chow interpolation method [17]. However, in that work, we were faced with some problems of unphysical oscillations when trying to simulate supersonic flow. Those problems were not related

to the magnetic part of the algorithm, instead, they were triggered by the gas dynamic component. Herein, we will consider our fluid as inviscid and adiabatic. Hence, we concentrate here on the Euler equations to tackle the problems associated with the convective component because we know already that this term is the source of the main numerical problems we have faced before. The Euler system of equations is given by the continuity, momentum and total energy equations and, for closing the system, density is related with pressure and temperature by an equation of state.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0 \quad (1)$$

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot (\rho \mathbf{U} \mathbf{U}) = -\nabla p \quad (2)$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho e \mathbf{U}) + \nabla \cdot (\mathbf{U} p) = 0 \quad (3)$$

$$\rho = \psi p = \frac{1}{RT} p, \quad (4)$$

where ψ is the compressibility coefficient, and R is the gas constant.

3. NUMERICAL MODEL

In this section, we will explain in detail the proposed algorithm, which was developed to try to compatibilize the advantageous properties that the pressure-based and density-based solvers are known to possess separately. We will describe how to calculate the mass flux, the velocity and the pressure at the cell face with the $\text{AUSM}^+ - up$, and how the AUSM flux definitions can be used in a PISO type of algorithm. Further, we make use of the openFOAM (OpenCFD, Ltd., UK) package as a developer tool for our new code. The openFOAM uses a finite volume discretization that solves the equations in a segregated approach, meaning that the flow variables remain frozen as one solves for the other variables at a given time step. This package has been widely used by researchers in the development of their own codes [20]. The main advantage of this tool is that, it is an open source, which allows the user to modify the source code and take advantage of contributions from a worldwide community.

3.1. $\text{AUSM}^+ - up$ interpolation

For clarity reasons, we show first how the fluxes are calculated with the $\text{AUSM}^+ - up$ scheme of Liou [19]. For simplicity, it is enough to consider a 1D case, because the extension to multidimensions follows the standard procedure of direction splitting. Let us consider that the subscript f represents a control volume face between cells P and F, see Figure 1. Using the $\text{AUSM}^+ - up$ method, the cell face velocity is determined by a Mach-dependent interpolation:

$$U_f = a_f M_f, \quad (5)$$

where a_f is the common speed of sound and M_f is the Mach number at the face. This concept of common speed of sound is very useful because it allows the exact resolution of discontinuities,

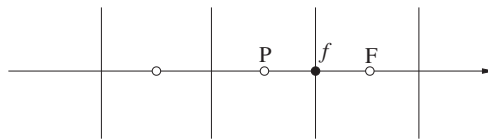


Figure 1. One-dimensional representation of a numerical mesh used with the new method. The left-state (L) and right-state (R) values are obtained using the left-side or right-side face variables information.

such as contacts and shocks. Moreover, it can be readily extended to deal with low speed flows, as proposed by Liou and Edwards [13] in the framework of a density-based solver. Following [19], a_f can be obtained as follows:

$$a_f = \min(\hat{a}_L, \hat{a}_R) \quad (6)$$

where the left-state and right-state speed of sound are given by the following:

$$\hat{a}_L = \frac{(a^*)^2}{\max(a^*, U_L)}, \quad \hat{a}_R = \frac{(a^*)^2}{\max(a^*, -U_R)}. \quad (7)$$

Where the variables U_L and U_R are the left-side and right-side face contributions of velocity, respectively. In multidimensional flows, the velocity magnitudes are calculated as $U = \mathbf{U} \cdot \mathbf{S}_f$, with \mathbf{S}_f being the unit normal vector of the cell face under consideration. The critical speed of sound, a^* , for a perfect gas can be calculated as follows:

$$a^* = \sqrt{\frac{2(\gamma - 1)}{\gamma + 1} H_t}, \quad (8)$$

where $H_t = e + p/\rho$ is the local total enthalpy and γ is the adiabatic index. The left and right Mach numbers are given by the following:

$$M_L = \frac{U_L}{a_f}, \quad M_R = \frac{U_R}{a_f}. \quad (9)$$

To properly scale the numerical dissipation provided by the AUSM scheme with the flow speed, there is the need to introduce a scaling factor, f_a . It has been demonstrated by Edwards and Liou [12, 13] that this scaling factor can be used to calculate preconditioned Mach numbers and a preconditioned speed of sound. However, we intend to follow a simpler and more robust approach, and for that, we are guided by the method of Liou [19]. So, the scaling factor is not applied to scale the Mach number and the speed of sound, but rather to scale the dissipative terms that are introduced in Equations (16) and (19). The scaling factor is a function of the reference Mach number (M_o):

$$f_a(M_o) = M_o(2 - M_o) \geq 0 \quad f_a \approx 2M_o, \text{ if } M_o \rightarrow 0 \quad (10)$$

with M_o given by:

$$M_o^2 = \min(1, \max(\bar{M}^2, M_\infty^2)), \quad (11)$$

where $\bar{M}^2 = \frac{1}{2}(M_L^2 + M_R^2)$ is the mean local Mach number and M_∞ is the freestream Mach number.

Having M_L , M_R and f_a , the interface Mach number is calculated from the following:

$$M_f = \mathcal{M}_{(4)}^+(M_L) + \mathcal{M}_{(4)}^-(M_R) + M_p, \quad (12)$$

where the split Mach numbers $\mathcal{M}_{(4)}(M)$ are polynomial functions of fourth order [14] given as follows:

$$\mathcal{M}_{(1)}^\pm(M) = \frac{1}{2}(M \pm |M|), \quad (13)$$

$$\mathcal{M}_{(2)}^\pm(M) = \pm \frac{1}{4}(M \pm 1)^2. \quad (14)$$

and

$$\mathcal{M}_{(4)}^\pm(M) = \begin{cases} \mathcal{M}_{(2)}^\pm(M) \left(1 \mp 16\beta \mathcal{M}_{(2)}^\mp(M)\right) & |M| < 1 \\ \mathcal{M}_{(1)}^\pm(M), & |M| \geq 1 \end{cases} \quad (15)$$

The pressure diffusion term M_p , is included in the calculation of the interface Mach number (Equation (12)) to promote coupling between velocity and pressure at low speed:

$$M_p = -\frac{K_p}{f_a} \max(1 - \bar{M}^2, 0) \frac{p_R - p_L}{\rho_f a_f^2}, \quad (16)$$

where $0 \leq K_p \leq 1$ is a dissipation coefficient, and the factor $\max(1 - \bar{M}^2, 0)$ acts as a switch, which monotonically increases from zero to one and becomes activated in regions of $\bar{M}^2 \leq 1$. The mean local density is obtained by averaging the left and right values, $\rho_f = (\rho_L + \rho_R)/2$. In low speed flows, the pressure diffusion term without f_a may not provide sufficient dissipation because $\Delta p = p_R - p_L$ is then very small. However, as discussed earlier, Liou [19] in the context of a density-based solver, derived an efficient way of scaling the pressure diffusion term with the flow speed using the scaling factor f_a defined by Equation (10).

In a similar fashion, the interface pressure is calculated from

$$p_f = \mathcal{P}_{(5)}^+(M_L)p_L + \mathcal{P}_{(5)}^-(M_R)p_R + p_u, \quad (17)$$

where $\mathcal{P}_{(5)}^+(M)$ is a polynomial function of fifth order given by the following:

$$\mathcal{P}_{(5)}^\pm(M) = \begin{cases} \mathcal{M}_{(2)}^\pm(M) \left[(\pm 2 - M) \mp 16\alpha M \mathcal{M}_{(2)}^\mp(M) \right] & |M| < 1 \\ \frac{1}{M} \mathcal{M}_{(1)}^\pm(M) & |M| \geq 1 \end{cases} \quad (18)$$

The p_u term is a velocity diffusion term having the same purpose as the pressure diffusion term of Equation (12); it is obtained from the following:

$$p_u = -K_u \mathcal{P}_{(5)}^+(M_L) \mathcal{P}_{(5)}^-(M_R) (\rho_L + \rho_R) f_a a_f (U_R - U_L). \quad (19)$$

Here, $0 \leq K_u \leq 1$ is, again, a dissipation coefficient, and the scaling factor f_a is introduced to achieve a proper scaling between velocity and the diffusive contribution of the AUSM scheme.

Parameters β and α in the polynomial functions are defined as $\beta = 1/8$ and:

$$\alpha = \frac{3}{16} (-4 + 5f_a^2) \in \left[-\frac{3}{4}, \frac{3}{16} \right]. \quad (20)$$

following Liou [19] for arbitrary Mach number flows. In all calculations, we set $K_p = 0.25$ and $K_u = 0.75$. Because density is a transported quantity, it can be upwinded, and the mass flux is given by:

$$\dot{m}_f = \frac{1}{2} [U_f (\rho_L + \rho_R) - |U_f| (\rho_R - \rho_L)]. \quad (21)$$

For the pressure equation, there is the need to calculate fluxes of the compressibility coefficient, which are denoted sonic flux F_s and are calculated as:

$$F_s^* = \frac{1}{2} \left[\hat{U}_f^* (\psi_L^n + \psi_R^n) - |\hat{U}_f^*| (\psi_R^n - \psi_L^n) \right]. \quad (22)$$

3.2. Numerical algorithm

We will now present the implementation of the PISO method. If a segregated approach is to be used, it is necessary to couple velocity, density and pressure through an appropriate algorithm. The segregated approach is iterative by nature and involves predictor and corrector steps. In the predictor step, the velocity field is calculated by solving an implicit version of the momentum equations, which are based on pressure values from the previous time step. In the corrector step, a pressure equation is assembled and solved implicitly, and then the velocity and density fields are corrected by explicit expressions.

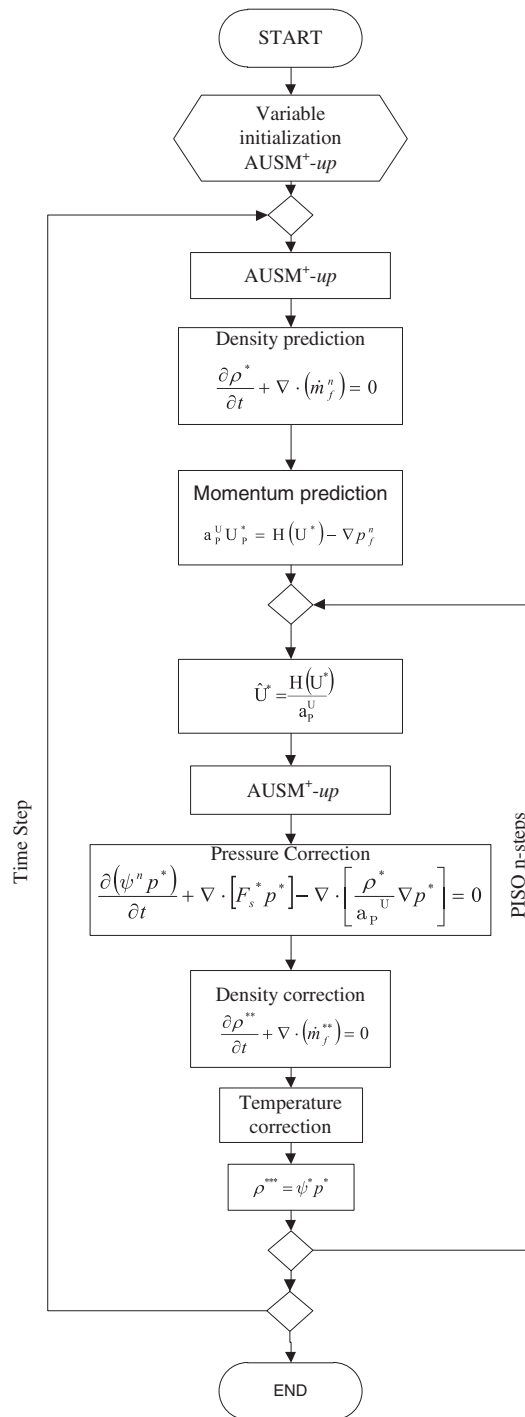


Figure 2. Flowchart corresponding to the new solver with $AUSM^+ - up$ flux definition as a replacement for the Rhie and Chow interpolation technique.

The values obtained in the previous time step are denoted with n , and the consecutive predictions and corrections of the PISO algorithm are denoted with * , ** , *** . The flowchart of the coded algorithm is presented in Figure 2.

3.2.1. Prediction step. All nodal values are assumed to be known at the previous time step n . At the beginning of each time step, the flux definition $AUSM^+ - up$ is used to obtain values for the

mass flux (\dot{m}_f^n), interface velocity (U_f^n) and interface pressure (p_f^n). To improve stability, an explicit equation for continuity is solved at this step,

$$\frac{\partial \rho^*}{\partial t} + \nabla \cdot (\dot{m}_f^n) = 0, \quad (23)$$

where the mass fluxes have just been obtained with the AUSM⁺ – up . This step allows prediction of density. The predicted values at the present time step for the velocity field, \mathbf{U}^* , are obtained by solving an implicit version of the momentum equation. The pressure gradient is treated in an explicit way, using the interface pressure previously obtained with the AUSM⁺ – up . This is an initial step before we start the PISO cycle. The discretized momentum equation, to be solved for \mathbf{U}^* , is

$$a_P^U \mathbf{U}_P^* = \mathbf{H}(\mathbf{U}^*) - \nabla p_f^n, \quad (24)$$

where a_P^U is the central velocity coefficient, and the operator $\mathbf{H}(\mathbf{U})$ is built using the convective terms of the neighbor cells to cell P.

3.2.2. Correction step. Operator $\mathbf{H}(\mathbf{U}^*)$ is constructed by using predicted values of velocity. This term will be used for the solution of the pressure equation in the next step. Using this operator, we can calculate the velocity field without the effects of pressure:

$$\hat{\mathbf{U}}^* = \frac{\mathbf{H}(\mathbf{U}^*)}{a_P^U}. \quad (25)$$

The AUSM⁺ – up fluxes are, again, calculated inside the PISO. The mass flux, the velocity and the pressure are evaluated in terms of the predicted velocity field. This results in the following updated fields: \dot{m}_f^* ; F_s^* ; U_f^* ; p_f^* . Obviously, the mass flux \dot{m}^* does not, at present, satisfy the continuity equation.

The pressure equation is built and solved using the compressibility of the previous time step, ψ^n and the first estimation of density, ρ^* . It is derived by combining Equations (24) and (25) and forcing satisfaction of the continuity Equation (1):

$$\frac{\partial(\psi^n p^*)}{\partial t} + \nabla \cdot [F_s^* p^*] - \nabla \cdot \left[\frac{\rho^*}{a_P^U} \nabla p^* \right] = 0. \quad (26)$$

This equation enables us to obtain the first pressure prediction, p^* . Several non-orthogonality steps are then performed for non-orthogonal mesh correction. For the treatment of non-orthogonality (see Figure 3) the product $\nabla \cdot \nabla p^*$ must be split in two parts: $\mathbf{S}_f \cdot (\nabla p^*)_f = |\Delta| (p_N^* - p_P^*) / |\mathbf{d}| + \mathbf{k} \cdot (\nabla p^*)_f$, where Δ is the vector parallel to \mathbf{d} , representing the orthogonal contribution, whereas the vector \mathbf{k} brings in the non-orthogonal contribution. These two vectors are evaluated so that the equation $\mathbf{S}_f = \Delta + \mathbf{k}$ is satisfied in a vector sense; Reference [21] gives more details on this issue.

By using the predicted pressure, the flux can now be corrected:

$$\dot{m}_f^{**} = \mathbf{S}_f \cdot \left[F_s^*(p)_f^* - \left(\frac{\rho^*}{a_P^U} \nabla p^* \right)_f \right]. \quad (27)$$

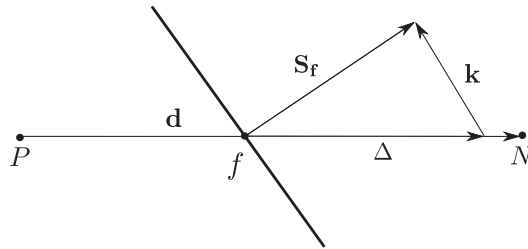


Figure 3. Treatment of non-orthogonality. [21].

The correction for velocity \mathbf{U}^{**} is performed in an explicit way using the new pressure gradient ∇p^* and the first predicted velocity:

$$\mathbf{U}^{**} = \frac{\mathbf{H}(\mathbf{U}^*)}{a_p^U} - \frac{\nabla p^*}{a_p^U} = \hat{\mathbf{U}}^* - \frac{\nabla p^*}{a_p^U}. \quad (28)$$

To correct the density ρ^{**} , we solve a continuity equation based on the new flux, similar to Equation (23).

We can now solve for the total energy equation, and afterwards, the temperature T is updated using the equation of state ($e = c_v T + \frac{1}{2} U^2$);

$$T^* = \frac{1}{\rho^{**} c_v} \left[\rho^{**} e^* - \frac{1}{2} \rho^{**} (U^{**})^2 \right]. \quad (29)$$

By using the updated temperature, we can calculate the following: the first prediction for the compressibility $\psi^* = 1/RT^*$; and a new correction for density $\rho^{***} = \psi^* p^*$.

When all the previously defined steps for PISO are performed, we can make a last correction for density using the final pressure. These steps should be enough for the continuity equation to be satisfied. Therefore, the continuity equation is effectively introduced inside the PISO cycle, and it is possible to check the mass conservation error. For that purpose, we use the density obtained from continuity equation, and we compare it with the value obtained from state equation $\rho = \psi p$.

4. RESULTS

This section presents and discusses the results obtained with the method just described, which has been embodied into what we call the new code. In addition, we report on comparisons between results from literature and results using this new code and a code we used before, which was based on the Rhie–Chow interpolation [17]. Three test cases are considered: first, we tackle the problem of an axisymmetric flow in a nozzle under subsonic, transonic and supersonic conditions, and we compare the numerical results with exact solutions; then, we analyze a subsonic, transonic and supersonic case flow past a bump; finally, we calculate a supersonic flow impinging on a cylinder. All the calculations were stopped when the maximum residual over the domain fell below 10^{-6} , except for the blunt body case calculation with the Rhie–Chow interpolation technique, where that value could not be attained.

4.1. Axisymmetric flow in a nozzle

The first test case under consideration has been used by several researchers for validation purposes [7, 22]. It is basically a 1D flow in a smooth converging/diverging axisymmetric nozzle. This case will allow validation of the new code for arbitrary Mach number flows because exact analytical solutions can be obtained and compared with the numerical results. The cross-sectional area of the nozzle varies as follows:

$$S_{(x)} = S_{th} + (S_i - S_{th}) \left(1 - \frac{x}{5} \right)^2, \quad (30)$$

where $S_i = 2.035$ and $S_{th} = 1$ are the inlet and throat areas, and the length of the nozzle is equal to 10 units. An axisymmetric 2D structured grid with 100×15 nodes was used in all calculations. A wide range of inlet Mach numbers were considered, resulting in the following: fully subsonic near-incompressible results ($Ma = 0.1$); transonic results ($Ma = 0.3$); and some fully supersonic results ($2.5 \leq Ma \leq 5$). For the subsonic test case, the following boundary conditions were considered: at the inlet, we prescribed total pressure and temperature and the angle of the flow; at the outlet, only the pressure was fixed and we extrapolated all other variables. For the supersonic cases, all variables are prescribed at the inlet and are extrapolated at the outlet.

The Mach number area-averaged numerical results are compared with the exact analytical solution in Figure 4, where six cases are shown. As we can see, the 2D predictions are in good agreement with the analytical solution, and no unphysical behavior is seen to occur. These results are also quite satisfactory when compared with 1D results from the literature of [7, 22].

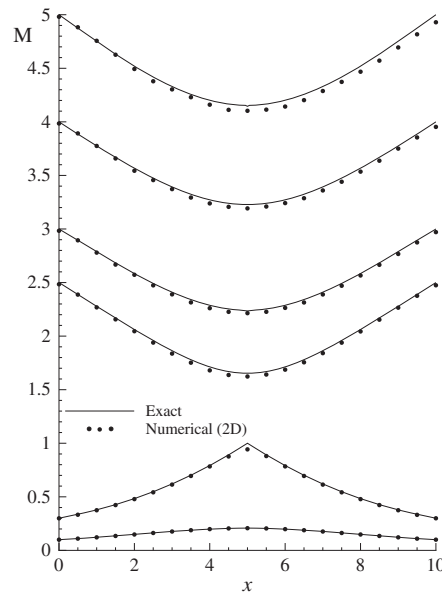


Figure 4. Comparison between the present numerical results and exact analytical solutions for the nozzle flow.

4.2. Flow past a circular bump in a channel

We present now the results obtained for the subsonic, transonic and supersonic flows past a bump. There is no analytical exact solution for these cases, but references solutions can be obtained from fine mesh simulations [23]. Here, we will not perform a grid independence study because we use the same meshes as in [17], where that issue was discussed. Regarding the geometry, it consists of a channel of height L and length $3L$, and along the bottom wall and centrally placed, a circular arc of length L and thickness $0.1L$. For the supersonic case, the thickness of the bump is equal to $0.04L$.

A 2D structured mesh is used in all cases. The mesh that was used for the subsonic cases has a total of 128×35 nodes. For the transonic case, to compare our results with others in literature, we used a different mesh with a total of 63×16 nodes as shown in Figure 5. The mesh for the supersonic flow has 158×78 nodes (Figure 6).

4.2.1. Near incompressible flow. We begin with the subsonic case, with the purpose of validating our algorithm for low Mach numbers. This flow is characterized by being totally subsonic, and in this case, we expect a symmetric distribution of the Mach number on the lower wall. The boundary conditions were the same for all cases of subsonic flow: at the inlet, total pressure and temperature were given, as well as the angle of the flow; at the outlet, static pressure was fixed, and all the other variables were extrapolated from inside the domain. These boundary conditions were imposed so that the inlet Mach number was equal to $Ma = 0.1$. To correctly calculate the left and right values of the variables at the cell face, we applied the quadratic upstream interpolation for convective kinematics scheme.

Figure 7 shows the results obtained for this subsonic flow case, in terms of contours of the Mach number field and its distribution along the lower and upper walls. These results are quite satisfactory when compared with similar ones from the literature (see Figure 8), and as expected, a symmetric flow is seen to occur.

To verify that the new code achieve second-order convergence in space, we performed a grid convergence study for the subsonic test case with an inlet Mach number equal to 0.5. For that purpose, we simulated the flow in four mesh densities: 20×5 ; 40×10 ; 80×20 ; 160×40 . The order of accuracy of the method was estimated by examining the behavior of the relative error for the pressure

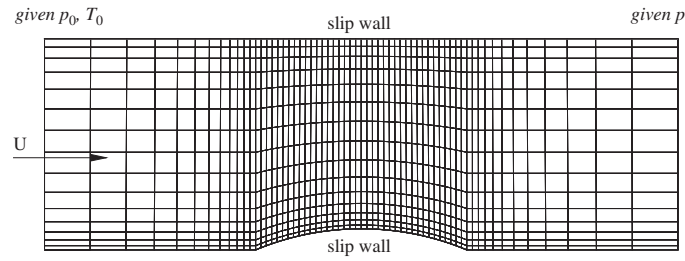


Figure 5. 2D coarse mesh used in the transonic bump flow simulations (63×16) and the corresponding boundary conditions.

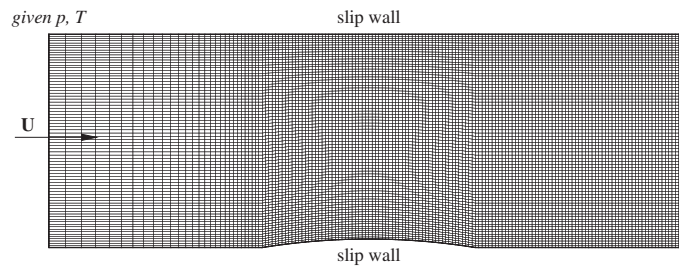


Figure 6. Two-dimensional mesh used in the supersonic bump flow simulation (158×78) and the corresponding boundary conditions.

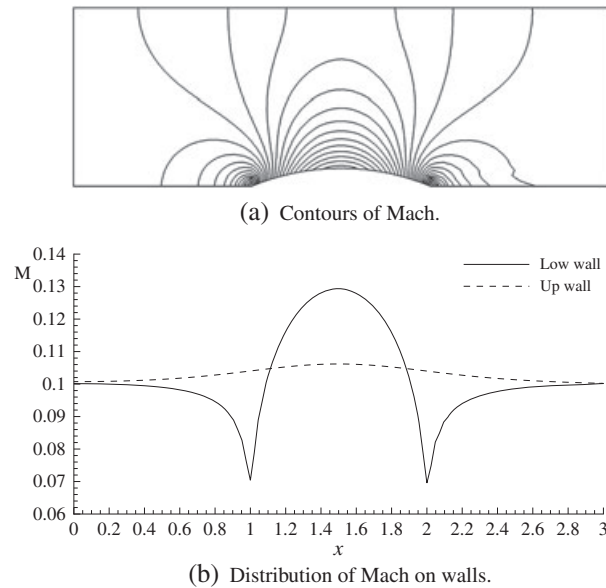


Figure 7. Results obtained for the subsonic bump flow case, $Ma = 0.1$; (a) contours of Mach and (b) distribution of Mach on walls.

coefficient in the lower wall calculated with the L_1 norm:

$$\|e_h\|_{L1} = \frac{|c_{p(\text{estimated})} - c_{p(\text{ref})}|}{c_{p(\text{ref})}}, \quad (31)$$

where $c_{p(\text{ref})}$ is the pressure coefficient calculated in the lower wall for a reference grid, in our case 320×80 .

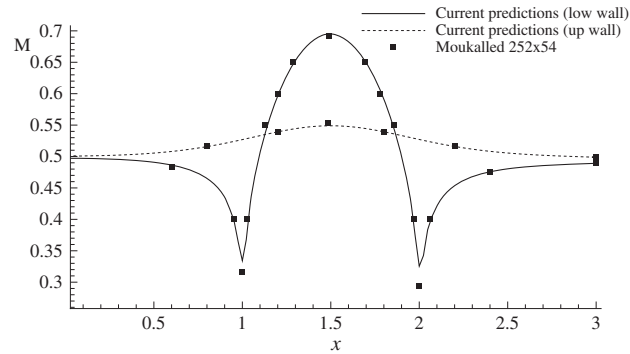


Figure 8. Comparison of current flow predictions (128×35) and with the results obtained by Moukalled [7], with a denser grid (252×54) for the subsonic bump flow test-case, $Ma = 0.5$.

In Figure 9, we plotted the L_1 norm result on a log-log scale, where the cell spacing is given in meters. If we compare the calculated norm with the first and second order grid convergence slopes, we can observe that for the new solver (AUSM), it is shown to drop in a second-order manner as the grid is refined. As expected, the previous code (Rhie-Chow) presents a similar behavior. However, for coarser grids (40×10), the grid convergence of the new code is higher than the previous code.

4.2.2. Transonic flow. The next case under consideration refers to a transonic flow at $Ma = 0.675$, which is characterized for remaining subsonic at both the inlet and the outlet. However, the increased inlet velocity, compared with the previous case, will create a shock wave in a region downstream the top of the bump. The boundary conditions for this case were identical to the subsonic case. Because of the presence of a discontinuity, we used the convergent and universally bounded interpolation scheme for the treatment of advection (CUBISTA) [24], a high resolution interpolation scheme for the convective terms. This same scheme was applied to obtain the right and left values of the variables at the cell faces.

In Figure 10, we present the Mach number field and a comparison between the results (M on lower wall) obtained with the method described in Section 3 and those obtained by Moukalled [7], that developed a code that uses the Rhie-Chow interpolation technique. As we can see, the representation of the shock wave is much sharper when we use the AUSM flux definitions. A similar beneficial conclusion was previously reported by Nerinckx *et al.* [16], who noticed that the fourth-order pressure diffusion term introduced by the Rhie-Chow interpolation scheme becomes extremely large when high-pressure gradients occur, thus introducing extreme smearing. Our new

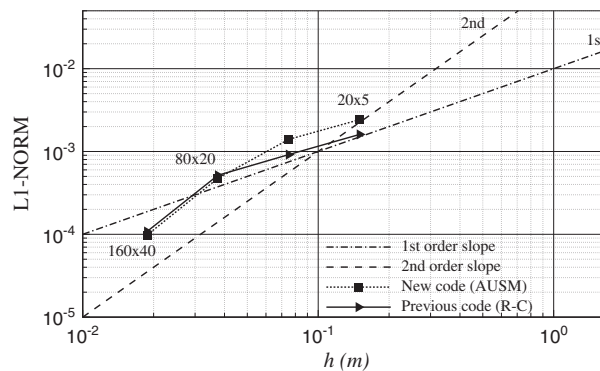


Figure 9. Spatial error L_1 norm for the pressure coefficient in the lower wall, for the subsonic bump flow test-case, $Ma = 0.5$.

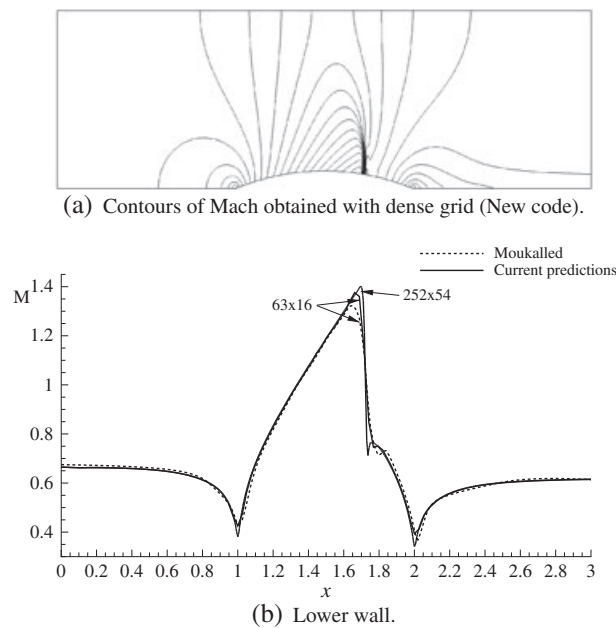


Figure 10. Comparison between (a) contours of Mach obtained with dense grid (new code) and with (b) current predictions and the results obtained by Moukalled [7] for the transonic bump flow case, $Ma = 0.675$.

approach has the ability to switch off this pressure dissipation term in areas of high Mach number, and thus significantly improve the accuracy.

4.2.3. Supersonic flow. For the supersonic case, different boundary conditions need to be applied. At the inlet, we imposed static pressure, temperature and velocity, and at the outlet, all variables were extrapolated from the interior of the solution domain. These boundary conditions were applied to obtain an inlet Mach number equal to $Ma = 1.4$. In this flow case, the CUBISTA scheme was applied to all the interpolations and convective terms.

This test case is characterized for being fully supersonic; it produces two shock waves that interact before leaving the solution domain. The shock wave that is formed at the upstream region of the bump is still reflected by the lower wall, after intersecting the other shock wave, and before leaving the domain.

In Figure 11(a), we show the results obtained with the new code: the intersection and reflection are very well resolved without unrealistic oscillations. On the other hand, the results obtained with the Rhie–Chow interpolation, see Figure 11(b), give signs of numerical dissipation and some unrealistic flow patterns. A comparison between the two approaches, in terms of the Mach number predicted over the lower wall, is shown in Figure 11(c), supporting the earlier mentioned comments, especially regarding the resolution of the second reflection of the first shock.

In Figure 11(c), we compare our results with others taken from the literature of [7]; those are based on codes that uses the Rhie–Chow interpolation. This comparison suggests that the new proposed algorithm resolves this flow with improved accuracy. This result was expected because dissipation diminishes as the mesh is refined. To magnify the difference between the two algorithms, we need to obtain results on a coarse mesh; this was carried out in detail for the transonic case.

4.3. Supersonic and hypersonic flow over a blunt body

The final test case here discussed is the most difficult to compute. Unlike the previous case, the flow now is not entirely supersonic, presenting a subsonic zone with a region of low velocities eventually leading to a stagnation point. This kind of flow occurs when a supersonic stream hits a circular obstacle, causing a bow shock to be formed upstream. Downstream of this shock wave, the flow

is subsonic with a stagnation point. Thus, our code must be able to resolve the shock wave and, at the same time, calculate the subsonic flow that develops in the region where the velocities must decrease to zero.

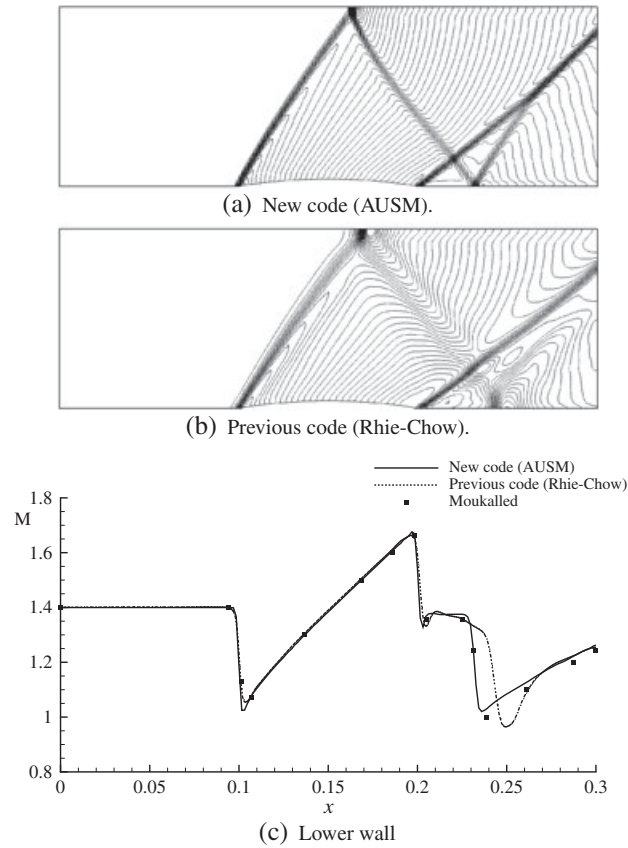


Figure 11. Results obtained for the supersonic bump flow case, $Ma = 1.4$; (a) new code (AUSM), (b) previous code (Rhie-Chow) and (c) lower wall.

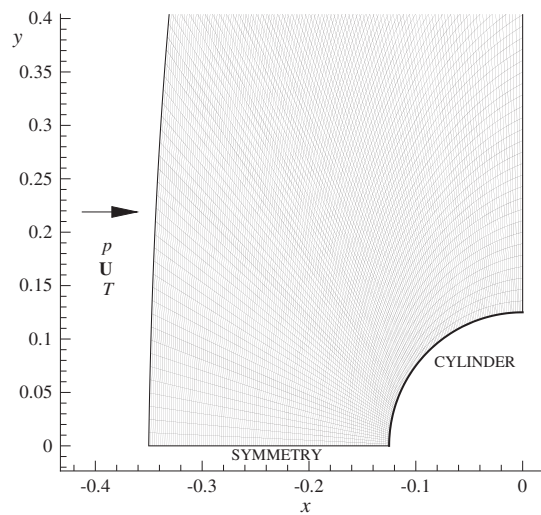


Figure 12. Two-dimensional Mesh used to compute the supersonic flow over a cylinder (120×120).

For reasons of symmetry, we have only simulated the flow in the upper left quadrant, on a 120×120 stretched, elliptic polar-like, structured grid, extending to $x = -0.35$ on the x -axis, and to $y = 1.4$ on the y -axis (see Figure 12 for a detail of the grid near the cylinder). The applied boundary conditions are similar to the previous supersonic test case: all variables at inlet were prescribed and were extrapolated at outlet. Two inlet Mach numbers were addressed, namely $Ma = 1.97$ and $Ma = 6$ with the adiabatic index set to $\gamma = 1.4$. The CUBISTA scheme was used throughout.

In Figures 13(a)–(d), we present a comparison between the results for the density and pressure fields obtained with the code (denoted AUSM) and with the Rhie–Chow interpolation technique for the lower Mach number case ($Ma = 1.97$). The Rhie–Chow technique can actually capture the position and shape of the shock wave; however, as the contour plots show, it tends to produce wiggles, which are similar to those observed in the odd–even decoupling of velocity and pressure. On the

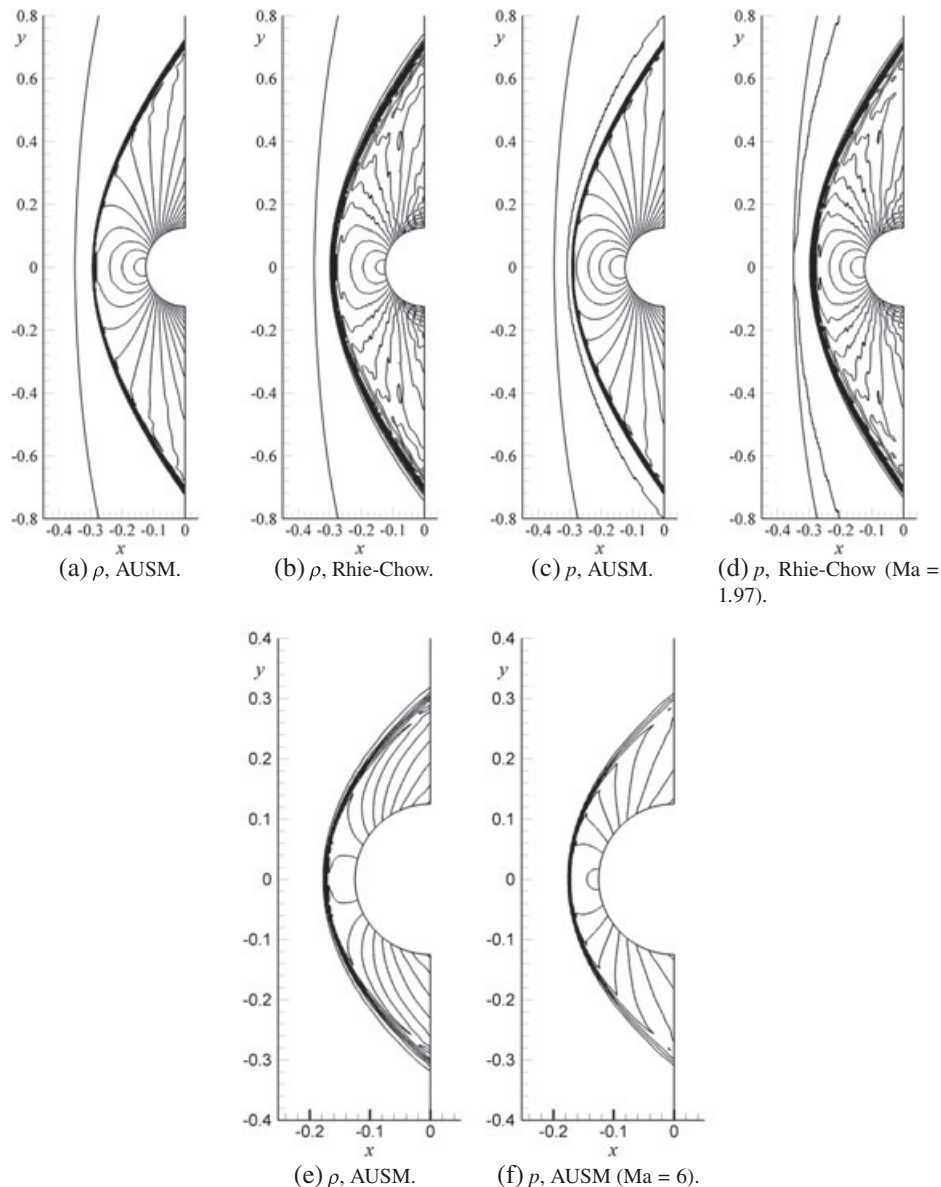


Figure 13. Contour plots of density (a, b, and e) and pressure (c, d, and f), obtained with the AUSM⁺ – up flux definition and the Rhie–Chow interpolation technique. Two cases for the cylinder flow: $Ma = 1.97$ and $Ma = 6$.

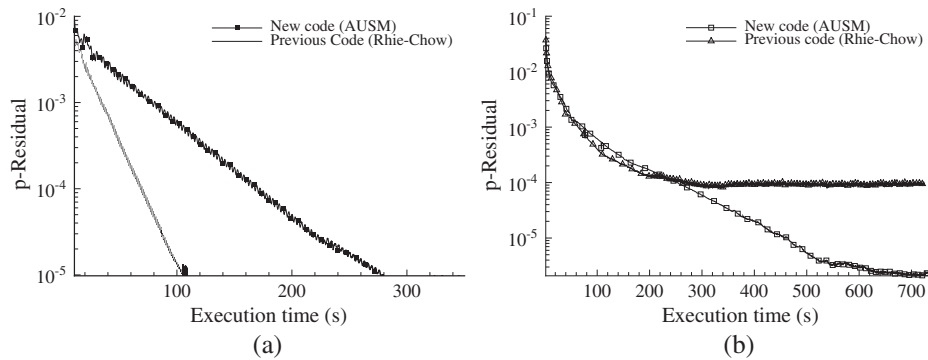


Figure 14. Performance study of iterative convergence for the: (a) subsonic flow in the bump test-case ($Ma = 0.5$); and (b) supersonic flow in the bump test-case ($Ma = 1.4$).

other hand, the new code is free from that kind of problems, even for much higher Mach number flows, as illustrated in Figures 13(e) and (f) for $Ma = 6$. Under these conditions, the code that uses the Rhie–Chow interpolation either diverges or gives completely erroneous answers.

4.4. Performance study of iterative convergence

It is important to analyze how the new method performs regarding its iterative convergence. A performance study was carried out for the subsonic and supersonic flow in the bump test-case, and the results were compared with the ones obtained from the previous code.

For the subsonic flow ($Ma = 0.5$), we used the 80×20 grid of Section 4.2.1, and for the supersonic case ($Ma = 1.4$), we used the (158×78) grid. The Courant number was set in all cases to 0.8.

In Figure 14(a), we present the results for the subsonic case and, as we can see, the previous code achieves faster convergence, exhibiting a shorter time of execution. This was expected because the Rhie–Chow is a more suitable method for solving low speed flows. However, this shortcoming is compensated because, for the supersonic flow, we can see in Figure 14(b) that the new method performs better because it achieves a better iterative convergence rate than the previous code.

5. CONCLUSIONS

The main objective of the present work was the implementation and testing of a reformulated compressible flow solver based on the PISO algorithm, and being adequate for arbitrary Mach number flows. This reformulated methodology uses the $AUSM^+ - up$ flux definitions in substitution of the Rhie and Chow interpolation technique, and allowed us to overcome two important practical problems. The first problem was related to the excess of numerical dissipation that the fourth order pressure diffusion term introduced when large pressure gradients occur. The new formulation has the ability of switching off this pressure diffusion term in high speed areas, and thus improving accuracy. The second problem that we identify was related to the odd–even decoupling that occurs in the blunt body test-case and that was triggered by the large pressure gradients that the Rhie and Chow interpolation technique could not handle.

Evaluation of the improvements introduced by the new algorithm was carried out through numerical computations of three distinct flow configurations under different flow conditions, encompassing wholly subsonic, transonic, wholly supersonic, and mixed subsonic (almost incompressible) and supersonic regions.

The results obtained with the new code, which has implemented the $AUSM^+ - up$ flux definitions, were compared against results obtained with the previous version of the code, which had the Rhie and Chow interpolation technique. We were able to show that the new version of the code works better under supersonic flow conditions, providing sharper shock resolution without the occurrence of

wiggles. In addition, we have also shown that it can produce accurate and physically realistic results in low Mach number flows, especially when there is coexistence of supersonic and subsonic flow regions, as in high-speed streams impinging on blunt bodies, in which case, the standard PISO with the Rhie–Chow interpolation yields unrealistic results with unphysical oscillations. These issues are especially relevant at high Mach numbers (say $Ma \geq 6$).

ACKNOWLEDGEMENTS

This work was supported by FCT grant SFRH/BD/60285/2009 and by FCT project PTDC/CTE-SPA/114163/2009 ‘SpaceProp - MHD Numerical Modeling in nozzles of MPD Thrusters for Space Propulsion’. Additional financial support was provided by CAST - Center for Aerospace Science and Technology at UBI.

REFERENCES

1. Issa RI, Javareshkian MH. Pressure-based compressible calculation method utilizing total variation diminishing schemes. *American Institute of Aeronautics and Astronautics Journal* 1998; **36**:1652–1657. DOI: 10.2514/2.567.
2. Lien F-S. A pressure-based unstructured grid method for all-speed flows. *International Journal for Numerical Methods in Fluids* 2000; **33**(3):355–374. DOI: 10.1002/1097-0363(20000615)33:3<355::AID-FLD12>3.0.CO;2-X.
3. Bressloff NW. A parallel pressure implicit splitting of operators algorithm applied to flows at all speeds. *International Journal for Numerical Methods in Fluids* 2001; **36**(5):497–518. DOI: 10.1002/flid.140.
4. Djavareshkian M, Abdollahi JM. Shock-capturing method using characteristic-based dissipation filters in pressure-based algorithm. *Acta Mechanica* 2010; **209**(1):99–113. DOI: 10.1007/s00707-009-0144-9.
5. Kobayashi MH, Pereira JCF. Characteristic-based pressure correction at all speeds. *American Institute of Aeronautics and Astronautics Journal* 1996; **34**(2):272–280. DOI: 10.2514/3.13061.
6. Patankar SV. *Numerical heat transfer and fluid flow*. Hemisphere Publishing Corporation: New York, 1980.
7. Moukalled F, Darwish M. A high-resolution pressure-based algorithm for fluid flow at all speeds. *Journal of Computational Physics* 2001; **168**(1):101–133. DOI: 10.1006/jcph.2000.6683.
8. Politis ES, Giannakoglou KC. A pressure-based algorithm for high speed turbomachinery flows. *International Journal for Numerical Methods in Fluids* 1997; **25**(1):63–80. DOI: 10.1002/(SICI)1097-0363(19970715)25:1<63::AID-FLD539>3.0.CO;2-A.
9. Thakur S, Wright J. A multiblock operator-splitting algorithm for unsteady flows at all speeds in complex geometries. *International Journal for Numerical Methods in Fluids* 2004; **46**(4):383–413. DOI: 10.1002/flid.763.
10. Liou MS, Steffen CJ. A new flux splitting scheme. *Journal of Computational Physics* Jul 1993; **107**(1):23–39. DOI: 10.1006/jcph.1993.1122.
11. Turkel E. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics* 1987; **72**(2):277–298. DOI: 10.1016/0021-9991(87)90084-2.
12. Edwards JR, Liou MS. Low-diffusion flux-splitting methods for flows at all speeds. *American Institute of Aeronautics and Astronautics Journal* 1998; **36**(9):1610–1617. DOI: 10.2514/2.587.
13. Liou M, Edwards J. Numerical speed of sound and its applications to schemes for all speeds. *Technical Report 1999-209286*, NASA, 1999.
14. Liou MS. A sequel to AUSM: AUSM(+). *Journal of Computational Physics* 1996; **129**(2):364–382. DOI: 10.1006/jcph.1996.0256.
15. Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *American Institute of Aeronautics and Astronautics Journal* 1983; **21**:1525–1532. DOI: 10.2514/3.8284.
16. Nerinckx K, Vierendeels J, Dick E. A Mach-uniform pressure-correction algorithm with AUSM + flux definitions. *International Journal of Numerical Methods For Heat & Fluid Flow* 2006; **16**(6):718–739. DOI: 10.1108/09615530610679075.
17. Xisto CM, Páscoa JC, Oliveira PJ, Nicolini D. Implementation of a 3D compressible MHD solver able to model transonic flows. *Proc. ECCOMAS CFD 2010 - V European Conference on Computational Fluid Dynamics*, 2010.
18. Issa RI. Solution of the implicitly discretized fluid flow equations by operator-splitting. *Journal of Computational Physics* 1986; **62**:40–65. DOI: 10.1016/0021-9991(86)90099-9.
19. Liou MS. A sequel to AUSM, part II: AUSM(+)-up for all speeds. *Journal of Computational Physics* 2006; **214**(1):137–170. DOI: 10.1016/j.jcp.2005.09.020.
20. Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to CFD using object orientated techniques. *Computers in Physics* 1998; **12**–6:620–631. DOI: 10.1063/1.168744.
21. Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows. *PhD Thesis*, University of London, Imperial College, 1996.
22. Demirdzic I, Lilek Z, Peric M. A collocated finite volume method for predicting flows at all speeds. *International Journal for Numerical Methods in Fluids* 1993; **16**:1029–1050. DOI: 10.1002/flid.1650161202.
23. Hirsch C. *Numerical Computation of Internal and External Flows - The Fundamentals of Computational Fluid Dynamics*. Elsevier: Oxford, 2007.
24. Alves MA, Oliveira PJ, Pinho FT. A convergent and universally bounded interpolation scheme for the treatment of advection. *International Journal for Numerical Methods in Fluids* 2003; **41**(1):47–75. DOI: 10.1002/flid.428.