

## A CLOSER LOOK AT LOW-SPEED PRECONDITIONING TECHNIQUES FOR THE EULER EQUATIONS OF GAS DYNAMICS

R. C. Moura<sup>1</sup>, A. P. Antunes<sup>1</sup>, E. Basso<sup>2</sup>, E. D. V. Bigarella<sup>3</sup>, J. L. F. Azevedo<sup>2</sup>

<sup>1</sup> Instituto Tecnológico de Aeronáutica, ITA ([moura@ita.br](mailto:moura@ita.br))

<sup>2</sup> Instituto de Aeronáutica e Espaço, IAE

<sup>3</sup> Empresa Brasileira de Aeronáutica, EMBRAER

**Abstract.** *It is widely known that compressible flow solvers usually behave badly in terms of both accuracy and convergence rate at the incompressible limit. The present paper focuses on presenting the fundamentals of low-speed preconditioning techniques in a detailed approach, with particular emphasis on the Euler equations in two-dimensional generalized coordinates. Issues related to eigenvalue scaling, local time-stepping correction, artificial dissipation adjustment and boundary conditions implementation are discussed. The computational tests are conducted using a finite difference code, which is constructed for structured grids in general curvilinear coordinates. Spatial discretization uses central differences plus added artificial dissipation. The time march can be performed with either explicit or implicit time discretization methods. Although the investigation is performed with a specific code, the conclusions reached in the process of the present research are quite general and, therefore, useful for CFD practitioners regardless of the type of method they work with. Several airfoil simulations are addressed in order to demonstrate the benefits or even the necessity of using low-speed preconditioning techniques in the context of aerospace applications.*

**Keywords:** *CFD, Low-speed flows, Preconditioning techniques.*

### 1. INTRODUCTION

Historically, once academia began to recognize difficulties in simulating low-speed flows with codes based on general compressible formulations, there was an active effort in devising solvers based directly on the incompressible flow equations. In return of such effort, successful techniques were developed for the simulation of incompressible flows, as the well known pressure-based schemes [6] or the pseudo-compressibility methods [3]. On the other hand, low-speed preconditioning techniques were created not only because it is more practical to have a single code capable of dealing with both compressible and incompressible regimes,

instead of separately maintaining different dedicated codes, but mainly because there is a variety of situations in which only a solver capable of simultaneously handling well both regimes can give satisfactory results, such as in Ref. [1]:

- high-speed flows with large embedded regions of low velocity, *e.g.*, subsonic flows upstream of a strongly converging nozzle;
- low-speed flows that are compressible due to density changes induced by heat sources, *e.g.*, surface heat transfer and combustion simulations;
- flows where compressible and incompressible regimes occur side by side and interact freely, *e.g.*, propulsion and high-lift configurations.

The simplest way to understand the need for preconditioning is perhaps by noticing the disparity between convection and acoustic speeds at the incompressible limit. Considering the one-dimensional Euler equations, one would have the set of eigenvalues  $\{u, u+a, u-a\}$ , where  $u$  and  $a$  are the particle and sound speeds, respectively. The so called condition number ( $C_N$ ), defined as the ratio between the highest eigenvalue (in absolute value) and the lowest eigenvalue (in absolute value), may be understood as a measure of time step restriction, or stiffness, when considering the eigenvalues of the Jacobian matrix of a hyperbolic system of equations to be numerically simulated [1]. For the subsonic regime, where  $u-a < 0 < u+a$ , the condition number can be written as a function of the local Mach number ( $M$ ) by

$$C_N = \frac{\max\{|u|, |u+a|, |u-a|\}}{\min\{|u|, |u+a|, |u-a|\}} = \frac{|u|+a}{|u|} = \frac{M+1}{M}. \quad (1)$$

Once  $\lim_{M \rightarrow 0} C_N = +\infty$ , common flow solvers present serious time step restrictions at low-speed regions and, consequently, overall convergence problems. The approach in low-speed preconditioning techniques consists, essentially, in altering the original partial differential equations in order to have more appropriate eigenvalues, therefore avoiding, or at least delaying, the condition number divergence at the incompressible limit. This way, codes based on general compressible formulations become capable of simulating nearly incompressible flows with relative easiness.

At the moment, there is a pacified understanding among researchers that preconditioning methods consist of successful and well established techniques for dealing with simulations involving low-speed flows. Despite of that, one can have great difficulty in finding an introductory-level contribution that discuss the physical essence common to all of the preconditioning approaches as well as the basic implementation issues in a detailed manner. The present study comes in to address such difficulty and basically intends to popularize and make low-speed preconditioning techniques better understood among general CFD practitioners. To that end, the two-dimensional Euler equations of gas dynamics are solved using the classical finite-difference discretization in generalized curvilinear coordinates, although the discussed concepts and principles have much wider applicability.

This paper is organized as follows. Section 2 reviews the two-dimensional Euler equations in Cartesian and curvilinear coordinates, introduces the preconditioned formulation and

discusses how the eigenvalues are modified by the preconditioning. Section 3 addresses important details such as how to alter the CFL-based local time step in order to obtain maximum efficiency from the preconditioning technique, as well as issues related to artificial dissipation adjustment and boundary conditions implementation. In section 4, several airfoil simulations are analyzed in order to demonstrate the benefits or even the necessity of using low-speed preconditioning techniques in the context of aerospace applications.

## 2. PRECONDITIONING THE EULER EQUATIONS

### 2.1. The General Compressible Formulation

In conservative form, the two-dimensional Euler equations in Cartesian coordinates are given by

$$\frac{\partial Q}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = 0, \quad (2)$$

where  $Q = [\rho, \rho u, \rho v, E]^T$  is the vector of conserved variables and  $F_x = [\rho u, \rho u^2 + p, \rho uv, (E + p)u]^T$  and  $F_y = [\rho v, \rho vu, \rho v^2 + p, (E + p)v]^T$  are the inviscid flux vectors. Moreover,  $\rho$  stands for density,  $u$  and  $v$  are the Cartesian components of the velocity vector,  $E = \rho[e_i + (u^2 + v^2)/2]$  is the total energy per unit volume, and  $e_i$  is the specific internal energy. The static pressure  $p$  is obtained using the equation of state for a perfect gas, namely,  $p = \rho(\gamma - 1)e_i$ , where  $\gamma$  is the fluid ratio of specific heats, which assumes the value  $\gamma = 7/5$  for the air.

For codes based on structured grids, the Euler equations are re-written in generalized curvilinear coordinates as

$$\frac{\partial \bar{Q}}{\partial t} + \frac{\partial \bar{F}_\xi}{\partial \xi} + \frac{\partial \bar{F}_\eta}{\partial \eta} = 0, \quad (3)$$

where  $\bar{Q} = J^{-1}Q$ ,  $\bar{F}_\xi = J^{-1}(\xi_x F_x + \xi_y F_y)$ ,  $\bar{F}_\eta = J^{-1}(\eta_x F_x + \eta_y F_y)$ , and  $J^{-1} = x_\xi y_\eta - x_\eta y_\xi$  is the reciprocal value of the transformation Jacobian. The metric terms composing the flux vectors can be calculated through the identities [4]

$$\begin{bmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{bmatrix} = \begin{bmatrix} \partial \xi / \partial x & \partial \xi / \partial y \\ \partial \eta / \partial x & \partial \eta / \partial y \end{bmatrix} = J \begin{bmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{bmatrix} = J \begin{bmatrix} \partial y / \partial \eta & -\partial x / \partial \eta \\ -\partial y / \partial \xi & \partial x / \partial \xi \end{bmatrix}. \quad (4)$$

As will be seen in the following sections, it is convenient, for preconditioning purposes, to work with the Euler equations written in terms of the vector of primitive variables  $Z = [p, u, v, e_i]^T$ , such that

$$\frac{\partial \bar{Q}}{\partial t} = \frac{\partial \bar{Q}}{\partial \bar{Z}} \frac{\partial \bar{Z}}{\partial t} = \frac{\partial Q}{\partial Z} \frac{\partial \bar{Z}}{\partial t}, \quad (5)$$

$$\frac{\partial \bar{F}_\xi}{\partial \xi} = \left( \xi_x \frac{\partial F_x}{\partial Z} + \xi_y \frac{\partial F_y}{\partial Z} \right) \frac{\partial \bar{Z}}{\partial \xi} = A \frac{\partial \bar{Z}}{\partial \xi}, \quad (6)$$

$$\frac{\partial \bar{F}_\eta}{\partial \eta} = \left( \eta_x \frac{\partial F_x}{\partial Z} + \eta_y \frac{\partial F_y}{\partial Z} \right) \frac{\partial \bar{Z}}{\partial \eta} = B \frac{\partial \bar{Z}}{\partial \eta}. \quad (7)$$

Naturally,  $\bar{Z} = J^{-1}Z$ ,  $A = (\partial F_x/\partial Z)\xi_x + (\partial F_y/\partial Z)\xi_y$  and  $B = (\partial F_x/\partial Z)\eta_x + (\partial F_y/\partial Z)\eta_y$ . Furthermore, the Jacobian matrices  $\partial Q/\partial Z$ ,  $\partial F_x/\partial Z$  and  $\partial F_y/\partial Z$  are given explicitly by

$$\frac{\partial Q}{\partial Z} = \begin{bmatrix} 1/[(\gamma-1)e_i] & 0 & 0 & -\rho/e_i \\ u/[(\gamma-1)e_i] & \rho & 0 & -\rho u/e_i \\ v/[(\gamma-1)e_i] & 0 & \rho & -\rho v/e_i \\ \frac{1}{(\gamma-1)} \left(1 + \frac{u^2+v^2}{2e_i}\right) & \rho u & \rho v & -\frac{\rho(u^2+v^2)}{2e_i} \end{bmatrix}, \quad (8)$$

$$\frac{\partial F_x}{\partial Z} = \begin{bmatrix} u/[(\gamma-1)e_i] & \rho & 0 & -\rho u/e_i \\ 1 + u^2/[(\gamma-1)e_i] & 2\rho u & 0 & -\rho u^2/e_i \\ 1 + uv/[(\gamma-1)e_i] & \rho v & \rho u & -\rho uv/e_i \\ \frac{u}{(\gamma-1)} \left(\gamma + \frac{u^2+v^2}{2e_i}\right) & \rho \left(\gamma e_i + \frac{3u^2+v^2}{2}\right) & \rho uv & -\frac{\rho u(u^2+v^2)}{2e_i} \end{bmatrix}, \quad (9)$$

$$\frac{\partial F_y}{\partial Z} = \begin{bmatrix} v/[(\gamma-1)e_i] & 0 & \rho & -\rho v/e_i \\ 1 + vu/[(\gamma-1)e_i] & \rho v & \rho u & -\rho vu/e_i \\ 1 + v^2/[(\gamma-1)e_i] & 0 & 2\rho v & -\rho v^2/e_i \\ \frac{v}{(\gamma-1)} \left(\gamma + \frac{u^2+v^2}{2e_i}\right) & \rho vu & \rho \left(\gamma e_i + \frac{u^2+3v^2}{2}\right) & -\frac{\rho v(u^2+v^2)}{2e_i} \end{bmatrix}. \quad (10)$$

Therefore, the Euler equations in general curvilinear coordinates can be rewritten as

$$\frac{\partial Q}{\partial Z} \frac{\partial \bar{Z}}{\partial t} + A \frac{\partial \bar{Z}}{\partial \xi} + B \frac{\partial \bar{Z}}{\partial \eta} = 0, \quad (11)$$

where it should be noted that absolutely no preconditioning was applied up to this point. In order to obtain the eigenvalues associated with Eq. (11), it should be pre-multiplied by  $(\partial Q/\partial Z)^{-1}$ , which gives

$$\frac{\partial \bar{Z}}{\partial t} + M_\xi \frac{\partial \bar{Z}}{\partial \xi} + M_\eta \frac{\partial \bar{Z}}{\partial \eta} = 0, \quad (12)$$

where

$$M_\xi = \left( \frac{\partial Q}{\partial Z} \right)^{-1} A, \quad (13)$$

$$M_\eta = \left( \frac{\partial Q}{\partial Z} \right)^{-1} B, \quad (14)$$

whose sets of eigenvalues are the usual (curvilinear) ones, namely,

$$\lambda(M_\xi) = \left\{ U, U, U \pm a (\xi_x^2 + \xi_y^2)^{1/2} \right\}, \quad (15)$$

$$\lambda(M_\eta) = \left\{ V, V, V \pm a (\eta_x^2 + \eta_y^2)^{1/2} \right\}, \quad (16)$$

where  $a = (\gamma p/\rho)^{1/2}$  is the speed of sound and  $U = u\xi_x + v\xi_y$  and  $V = u\eta_x + v\eta_y$  are the well-known contravariant velocity components [4].

## 2.2. Introducing the Preconditioning Matrix

Different preconditioning methods are obtained depending on how the eigenvalues are modified. The present work follows the technique introduced in Ref. [9], which was found to perform very well for a diversity of flow cases. The main idea is to multiply the first column of the matrix  $\partial Q/\partial Z$  by a  $1/\epsilon$  factor, which can be incorporated into the  $\Gamma$  matrix, given by

$$\Gamma = \begin{bmatrix} 1/[\epsilon(\gamma-1)e_i] & 0 & 0 & -\rho/e_i \\ u/[\epsilon(\gamma-1)e_i] & \rho & 0 & -\rho u/e_i \\ v/[\epsilon(\gamma-1)e_i] & 0 & \rho & -\rho v/e_i \\ \frac{1}{\epsilon(\gamma-1)} \left(1 + \frac{u^2+v^2}{2e_i}\right) & \rho u & \rho v & -\frac{\rho(u^2+v^2)}{2e_i} \end{bmatrix}. \quad (17)$$

Thus, Eq. (11), after preconditioning, becomes

$$\Gamma \frac{\partial \bar{Z}}{\partial t} + A \frac{\partial \bar{Z}}{\partial \xi} + B \frac{\partial \bar{Z}}{\partial \eta} = 0. \quad (18)$$

In order to obtain the eigenvalues associated with Eq. (18), it should be premultiplied by  $\Gamma^{-1}$ , which gives

$$\frac{\partial \bar{Z}}{\partial t} + M_\xi^{pc} \frac{\partial \bar{Z}}{\partial \xi} + M_\eta^{pc} \frac{\partial \bar{Z}}{\partial \eta} = 0. \quad (19)$$

The new matrices are defined as

$$M_\xi^{pc} = \Gamma^{-1} A, \quad (20)$$

$$M_\eta^{pc} = \Gamma^{-1} B, \quad (21)$$

whose sets of preconditioned eigenvalues are

$$\lambda(M_\xi^{pc}) = \left\{ U, U, \frac{1}{2} \left[ U(1+\epsilon) \pm [U^2(1-\epsilon)^2 + 4\epsilon a^2(\xi_x^2 + \xi_y^2)]^{1/2} \right] \right\}, \quad (22)$$

$$\lambda(M_\eta^{pc}) = \left\{ V, V, \frac{1}{2} \left[ V(1+\epsilon) \pm [V^2(1-\epsilon)^2 + 4\epsilon a^2(\eta_x^2 + \eta_y^2)]^{1/2} \right] \right\}. \quad (23)$$

The preconditioned eigenvalues return the original ones as  $\epsilon \rightarrow 1$ .

It is now important to consider the effect of preconditioning in the steady-state solution of the modified equation. For this sake, Eq. (18) can be written as

$$\frac{\partial \bar{Z}}{\partial t} = \Gamma^{-1} R, \quad (24)$$

where  $R$  stands for the residue of the Euler equations which, recalling Eqs. (6) and (7), is given by

$$R = - \left( A \frac{\partial \bar{Z}}{\partial \xi} + B \frac{\partial \bar{Z}}{\partial \eta} \right) = - \left( \frac{\partial \bar{F}_\xi}{\partial \xi} + \frac{\partial \bar{F}_\eta}{\partial \eta} \right). \quad (25)$$

This term is, naturally, equal to zero when the steady-state solution is reached by means of the preconditioned formulation, as indicated in Eq. (24). Therefore, in both formulations, *i.e.*, original or preconditioned, the same result is obtained at the steady state.

Finally, Eq. (24) can also be solved for the vector of the conserved variables through Eq. (5), which gives

$$\frac{\partial \bar{Q}}{\partial t} = M^{pc} R, \quad (26)$$

where  $M^{pc} = (\partial Q / \partial Z) \Gamma^{-1}$  is called the preconditioning matrix in the present study. This matrix is given explicitly by

$$M^{pc} = (\epsilon - 1) \begin{bmatrix} \frac{1}{\epsilon-1} + \frac{u^2+v^2}{2e_i} & -u/e_i & -v/e_i & 1/e_i \\ u(u^2+v^2)/(2e_i) & \frac{1}{\epsilon-1} - \frac{u^2}{e_i} & -vu/e_i & u/e_i \\ v(u^2+v^2)/(2e_i) & -uv/e_i & \frac{1}{\epsilon-1} - \frac{v^2}{e_i} & v/e_i \\ \frac{(2e_i+u^2+v^2)(u^2+v^2)}{4e_i} & -\frac{u(2e_i+u^2+v^2)}{2e_i} & -\frac{v(2e_i+u^2+v^2)}{2e_i} & \frac{\epsilon}{\epsilon-1} + \frac{u^2+v^2}{2e_i} \end{bmatrix}. \quad (27)$$

It is worth noting that Eq. (26) makes very clear that, even though the steady-state solution of the preconditioned equation is precisely the same of the original one, the intermediate states are not, since the preconditioning matrix does alter the coherence of the temporal evolution. Hence, preconditioning techniques alone are of no use when there is interest in transient states or when simulating unsteady flows. Fortunately, preconditioning techniques are easily extendable to unsteady flow applications in conjunction with dual time stepping algorithms.

### 2.3. Analyzing the Eigenvalue Scaling

Up to this point, no definition was given for the factor  $\epsilon$ . In order to better visualize how this parameter affect the eigenvalues, it is useful to consider once again (as in section 1) the one-dimensional Euler equations in Cartesian components. The eigenvalues obtained by preconditioning such equations would be

$$\lambda(M^{pc}) = \left\{ u, \frac{1}{2} \left[ u(1+\epsilon) \pm [u^2(1-\epsilon)^2 + 4\epsilon a^2]^{1/2} \right] \right\}. \quad (28)$$

The main strategy is to define  $\epsilon$  as a function of the Mach number in order to maintain the condition number "well behaved" in low-speed flow regions. This work follows the simple suggestion of Ref. [8], where

$$\epsilon = \min \{ 1, \max \{ M^2, M_\infty^2 \} \}, \quad (29)$$

being  $M$  the local Mach number (since the preconditioning matrix has to be evaluated at each grid node when performing a numerical simulation) and  $M_\infty$  is a reference free-stream Mach number value (for external airfoil flows of interest here,  $M_\infty$  is the prescribed Mach number at infinity).

Figures 1 to 4 illustrate how the set of eigenvalues in Eq. (28) behaves as function of the local Mach number  $M$  for different values of the free-stream Mach number  $M_\infty$  when approaching the incompressible limit. In such figures, each color is associated with one eigenvalue ( $u+a$  in blue,  $u$  in green and  $u-a$  in red). The condition number  $C_N$  is given in black. Also, curves shown with "empty dots" refers to the original eigenvalues and curves with "full dots" refers to the preconditioned ones.

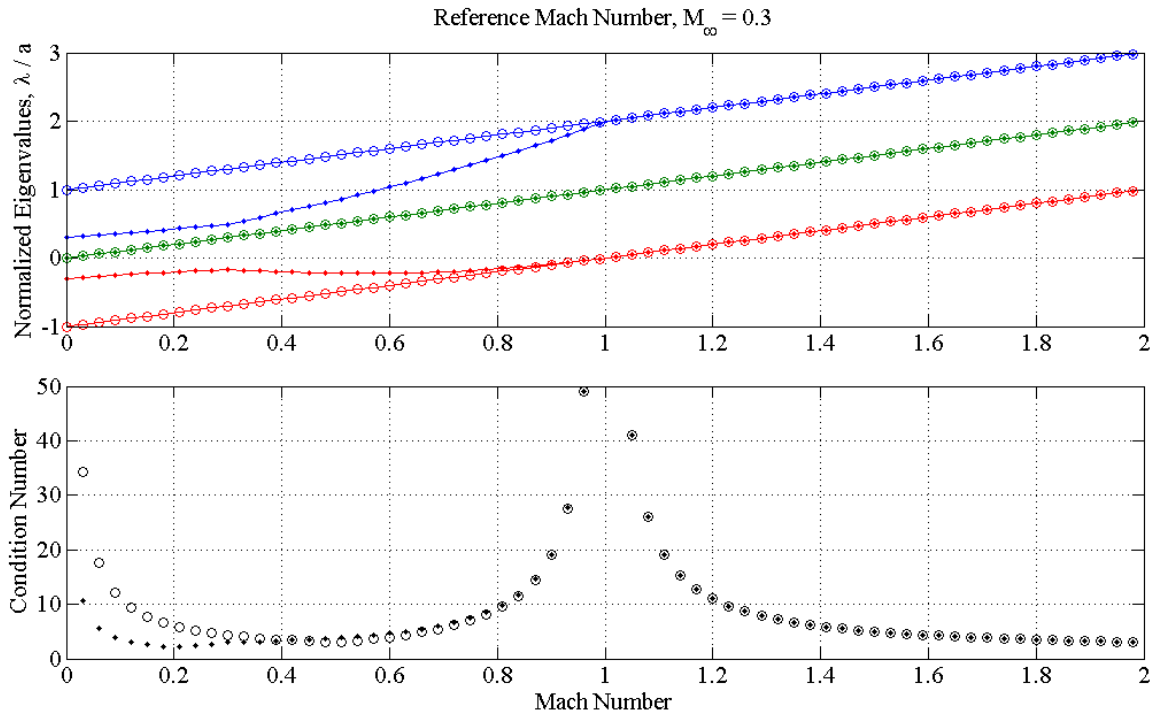


Figure 1. Eigenvalues and condition number analysis for  $M_\infty = 0.3$ .

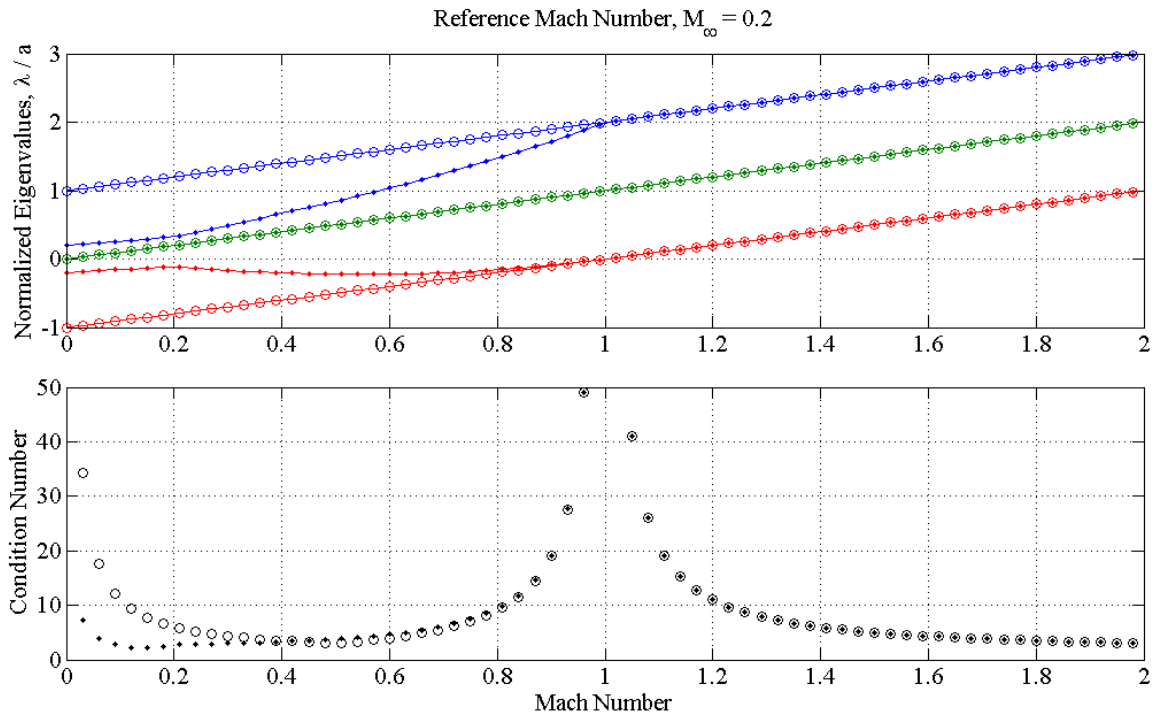


Figure 2. Eigenvalues and condition number analysis for  $M_\infty = 0.2$ .

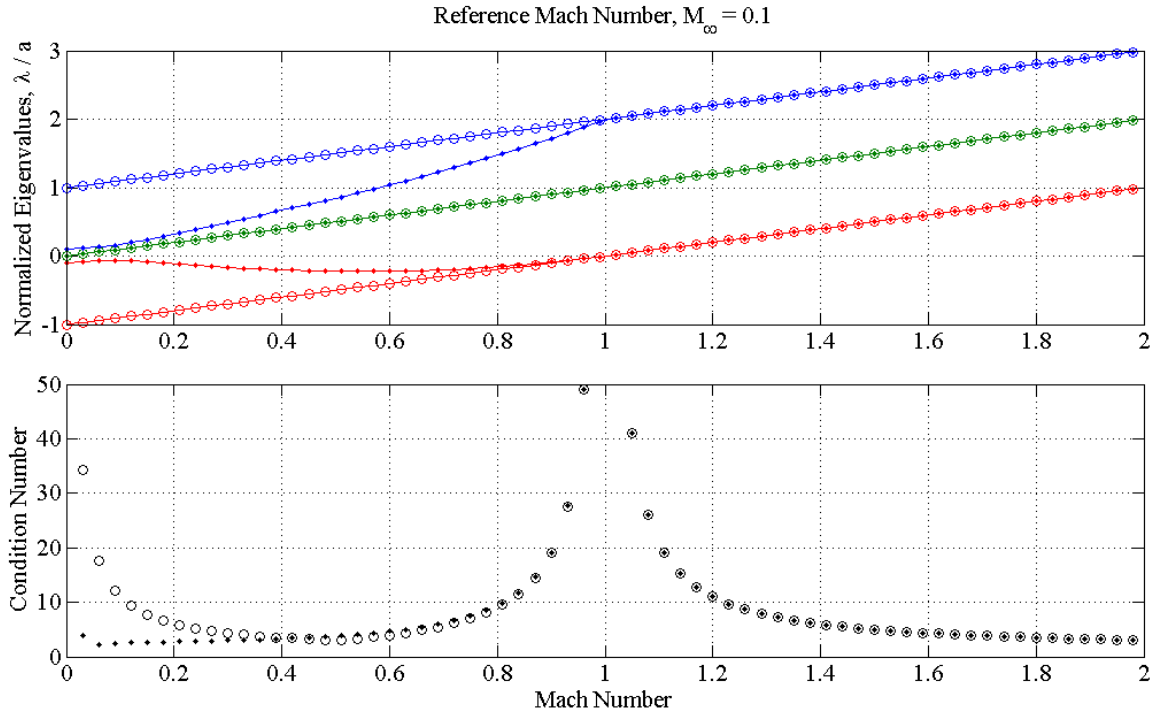


Figure 3. Eigenvalues and condition number analysis for  $M_\infty = 0.1$ .

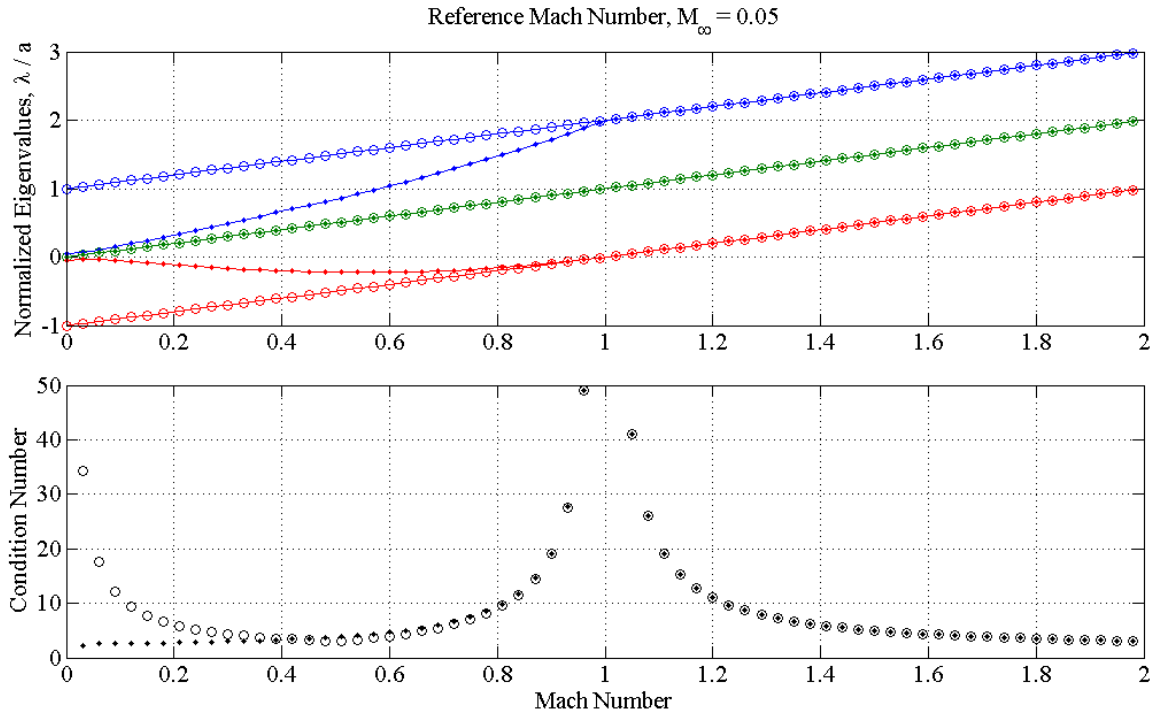


Figure 4. Eigenvalues and condition number analysis for  $M_\infty = 0.05$ .



As can be seen in figures 1 to 4, the original and preconditioned eigenvalues coincide if  $M > 1$ . For  $M = 1$ , a well-known  $C_N$  divergence is observed since the lowest eigenvalue (in absolute value) approaches zero. Such divergence is not critical for most flow cases and is not addressed here. The reader is referred to Ref. [2] and references therein for further discussions.

For subsonic cases, figures 1 to 4 show effective difference between the original and preconditioned eigenvalues, but such difference becomes effective for the condition number only when  $M < M_\infty$ , precisely where a discontinuous change of inclination can be noted in the curves associated with the preconditioned eigenvalues. Clearly, the preconditioned condition number shows better behavior in terms of divergence when compared with the original condition number, effect caused by the clustering of eigenvalues around the “central” one, which is not modified by the preconditioning and goes to zero at the incompressible limit, resulting in a bounded ratio of eigenvalues.

It is also important to point out that the divergence of the preconditioned condition number is delayed appropriately with decreasing  $M_\infty$ , assuring that the stiffness problem is properly bypassed, as expected of a good low-speed preconditioning method.

### 3. HANDLING IMPORTANT DETAILS

#### 3.1. Adjusting the CFL-Based Time Step Restriction

Numerical experiments show that if the same constant, or fixed, time step is used to simulate a given flow case, the preconditioned formulation will be less efficient in terms of convergence rate than the original formulation, and this difficulty becomes worse with decreasing Mach number. Such assertion sounds contradictory since the preconditioning main purpose is obviously the gain in terms of convergence rate, specifically when dealing with low-speed flows.

In order to understand such problem, it is useful to consider the classical shock-tube problem [7]. In such problem, the temporal evolution happens through the traffic of information carried by shock and expansion waves (and also by the contact discontinuity) which are associated with the standard set of eigenvalues  $\{u, u + a, u - a\}$ . Essentially, the information speed is directly related with the magnitude of the problem’s eigenvalues. The fact is that such concepts can be extended naturally for general numerical flow simulations since, beginning from a given initial condition, information waves transmit modifications on the physical properties during convergence until a steady-state is achieved.

As seen in figures 1 to 4, the magnitude of the preconditioned eigenvalues is very little (especially at the incompressible limit) when compared with the original ones, and that is the reason why, when using a same constant/fixed time step, one gets slower convergence for the preconditioned formulation. Such disadvantage, however, comes from the same root that makes low-speed preconditioning techniques so efficient for low-speed flows, namely, the clustering of eigenvalues. The explanation is that the preconditioned system of equations admits a much larger time step (at the limit of numerical divergence) than the original system of equations, since the original eigenvalues easily extrapolates the boundaries of the stability envelope

(having in mind constant/fixed time-stepping algorithms) because of their magnitude disparity. Therefore, the effectiveness of preconditioning consists essentially in allowing larger numerical time steps. The simplest way to explore such capacity is by using a CFL-based preconditioned local time step.

The alteration in the CFL-based time step is straightforward and basically consists in using the preconditioned eigenvalues instead of the original ones. For the one-dimensional Euler equations with no preconditioning, a typical expression for the time step based on the CFL number would be

$$\Delta t = \frac{\text{CFL}}{|u| + a} \Delta x . \quad (30)$$

For the preconditioned equations, the eigenvalues in (28) would be used instead, giving

$$\Delta t = \frac{\text{CFL}}{\frac{1}{2} \left[ |u| (1 + \epsilon) + [u^2(1 - \epsilon)^2 + 4\epsilon a^2]^{1/2} \right]} \Delta x . \quad (31)$$

Now, for the bi-dimensional formulation (in curvilinear coordinates) with no preconditioning, a typical CFL-based time step would be

$$\Delta t = \frac{\text{CFL}}{r_\xi + r_\eta} , \quad (32)$$

where  $r_\xi$  and  $r_\eta$  are obtained from (15) and (16), being given by

$$r_\xi = |U| + a(\xi_x^2 + \xi_y^2)^{1/2} , \quad (33)$$

$$r_\eta = |V| + a(\eta_x^2 + \eta_y^2)^{1/2} . \quad (34)$$

Analogously, for the preconditioned equations, the preconditioned eigenvalues should be used, giving

$$\Delta t = \frac{\text{CFL}}{r_\xi^{pc} + r_\eta^{pc}} , \quad (35)$$

where  $r_\xi^{pc}$  and  $r_\eta^{pc}$  are now obtained from (22) and (23), being given by

$$r_\xi^{pc} = \frac{1}{2} \left[ |U| (1 + \epsilon) + [U^2(1 - \epsilon)^2 + 4\epsilon a^2(\xi_x^2 + \xi_y^2)]^{1/2} \right] , \quad (36)$$

$$r_\eta^{pc} = \frac{1}{2} \left[ |V| (1 + \epsilon) + [V^2(1 - \epsilon)^2 + 4\epsilon a^2(\eta_x^2 + \eta_y^2)]^{1/2} \right] . \quad (37)$$

### 3.2. Dealing with Numerical Dissipation and Boundary Conditions

In this work, where artificial dissipation is added explicitly to the formulation, it was found convenient to leave the dissipation term unaltered (in its original form) and to accommodate it with the residue  $R$  given in (25), being the stabilized version of equation (26) given by

$$\frac{d\bar{Q}}{dt} = M^{pc} R_{stab} , \quad (38)$$

where  $R_{stab} = R + D$  would be the stabilized residue, being  $D$  the original dissipation term. By doing this, the steady state solution obtained is precisely the same given by the original formulation, namely, the one corresponding to  $R_{stab} = 0$ . Such implementation is obviously independent of the specific numerical viscosity model being used and is, therefore, quite general.

It has been shown that numerical dissipation generally scales badly as the Mach number is reduced [5], but it was found in the present study that such problem becomes significant only for extremely low free-stream Mach numbers, being of no practical importance for a wide variety of low-speed test flow cases simulated. Additionally, it is also known that better performance can in fact be extracted of the preconditioned formulation if the dissipation term is modified according to the preconditioned eigenvalues. In order to avoid the overhead associated with the evaluation of preconditioning matrices for the artificial viscosity in each grid node, the present work did not followed such modification ideas further, since the performance gain was found already very satisfactory. The reader interested in how to modify the dissipation term may consult Ref. [8].

Concerning the implementation of eigenvalue-based boundary conditions such as the usual non-reflecting far-field ones (very common in the context of external flow simulations), a precise approach should consider the modified characteristic variables (or Riemann invariants) of the preconditioned system. In the present study, however, it was found that no significant differences become apparent by using the original (unmodified) boundary treatment, at least for steady-state problems. Heuristically (yet having in mind the shock-tube problem), in order to minimize errors from such source, it is important that the symmetric form of the eigenvalues  $\{u - f(a), u, u + f(a)\}$  is maintained by the preconditioning method being used. The reader interested in how to modify the far-field boundary conditions may also consult Ref. [8].

## 4. NUMERICAL RESULTS

In this section, some airfoil low-speed simulations are addressed in order to investigate the gain in convergence rate and accuracy of the preconditioned formulation over the original one. All simulations were performed using the classical 2nd order-accurate Runge-Kutta (explicit) time-stepping scheme. Also, in all the pairs of compared convergence curves, the same CFL-based time step was used, as explicitly informed in each figure. The well-known NACA0012 and RAE2822 airfoils were chosen as examples of typical profiles used in the aerospace context. Their related computational meshes can be seen in figures 7 and 10, respectively. It should be said that both meshes are of circular type and have 81x40 (NACA) and 129x200 (RAE) grid points along surface and radial coordinates, in that order. For each case, the far-field was “safely” fixed at a considerable distance (typically tenths of chords) away from the airfoils.

Figure 5 shows how convergence speed is loosed with decreasing free-stream Mach number. This effect is true for both preconditioned and original formulations, but it is far more critical without preconditioning. All charts of figure 5 refer to simulations for the NACA0012 profile with zero angle of attack (AOA). It is important to note that better convergence speed is obtained with the preconditioned formulation when the such rate is based on iterations. If based

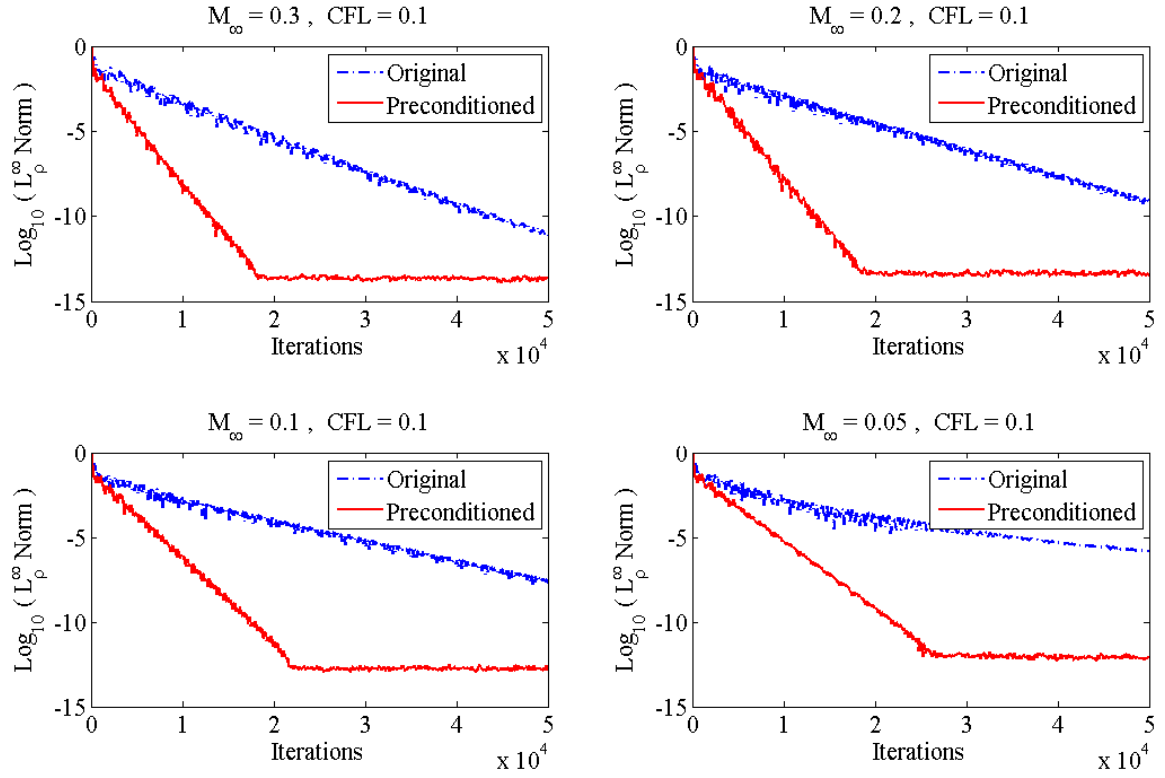


Figure 5. Convergence rate analysis (NACA0012, AOA = 0°)

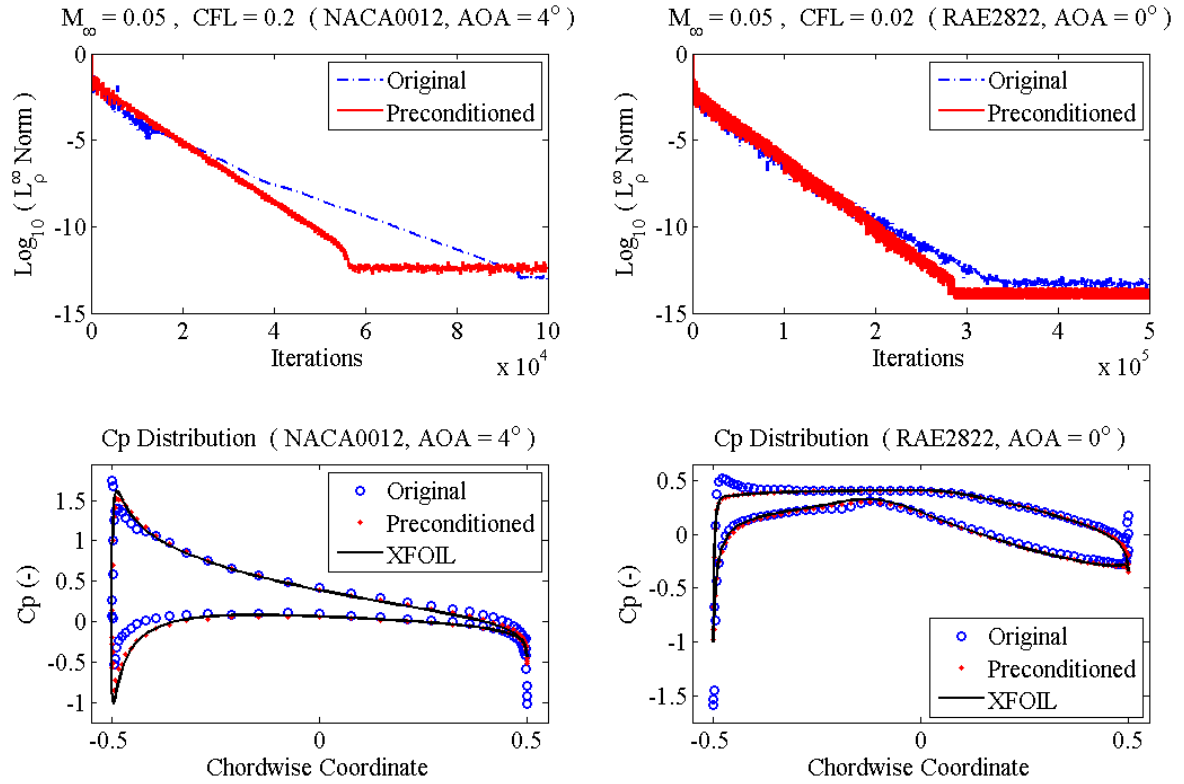


Figure 6. Cp distribution comparison (NACA0012, AOA = 4°) and (RAE2822, AOA = 0°)

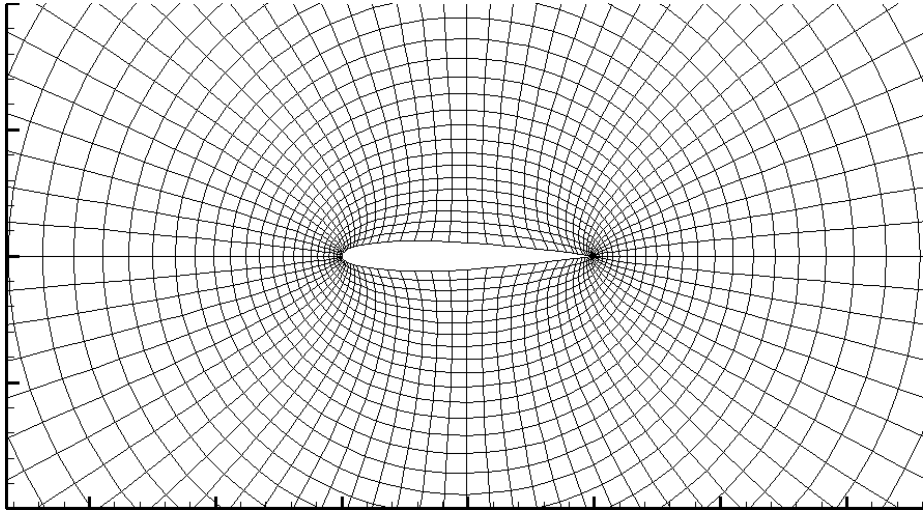


Figure 7. NACA0012 computational mesh with 81x40 grid points

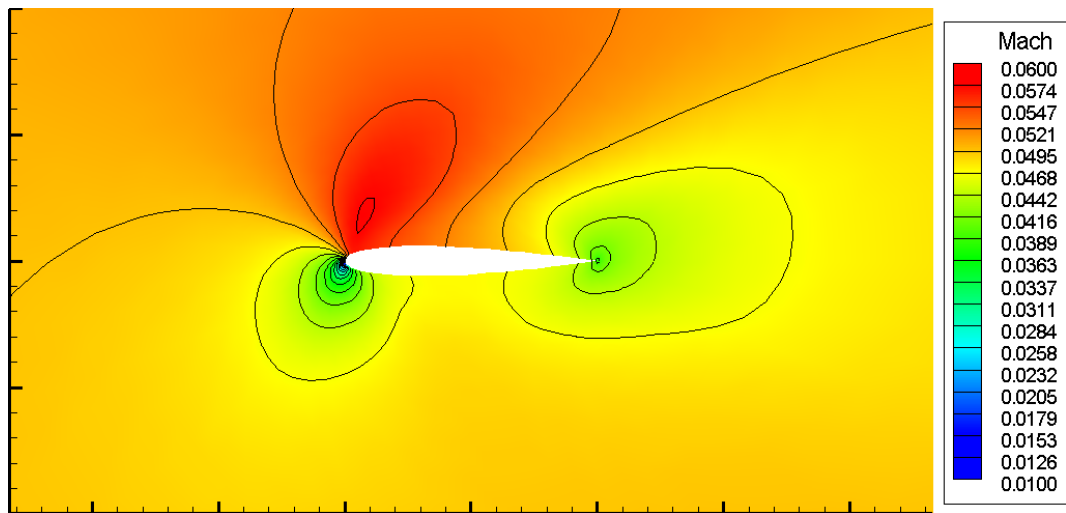


Figure 8. Mach number contours without preconditioning (NACA0012, AOA = 4°,  $M_\infty = 0.05$ )

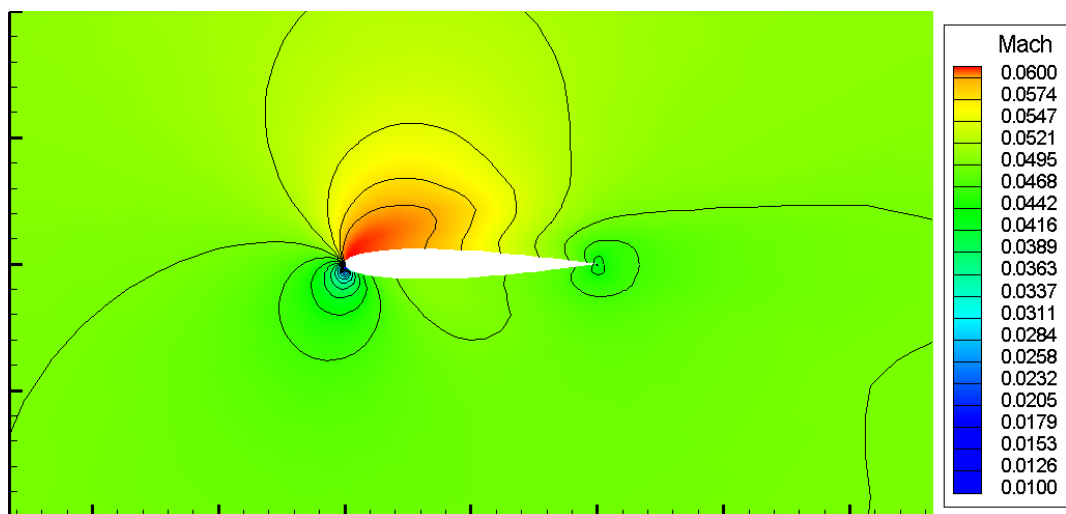


Figure 9. Mach number contours with preconditioning (NACA0012, AOA = 4°,  $M_\infty = 0.05$ )

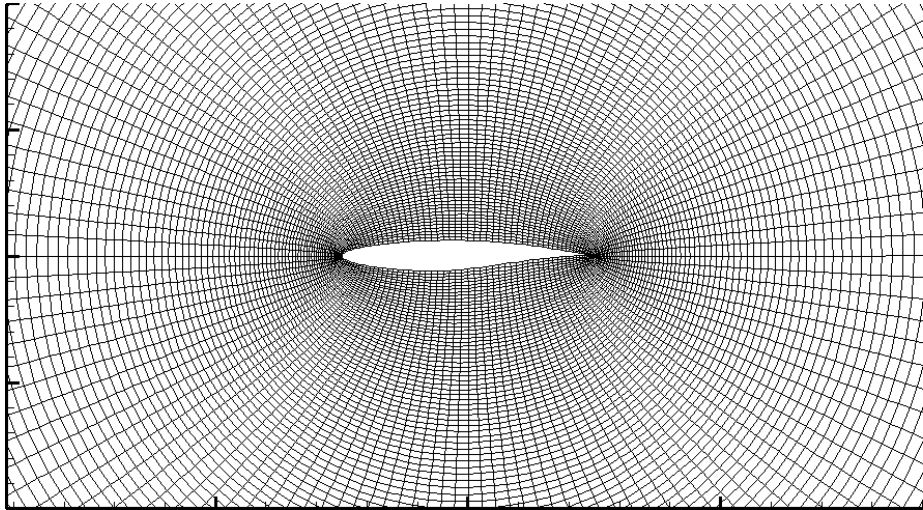


Figure 10. RAE2822 computational mesh with 129x200 grid points

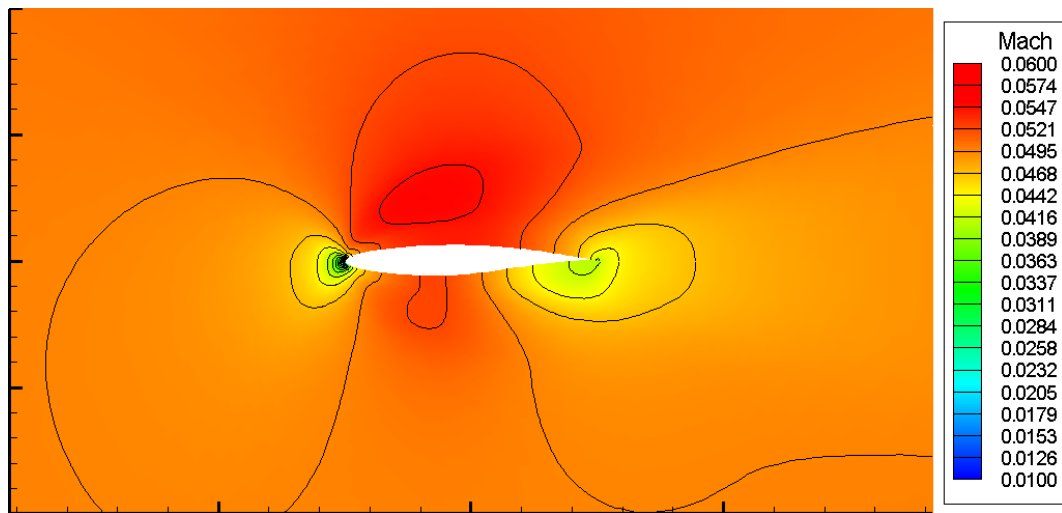


Figure 11. Mach number contours without preconditioning (RAE2822, AOA = 0°,  $M_\infty = 0.05$ )

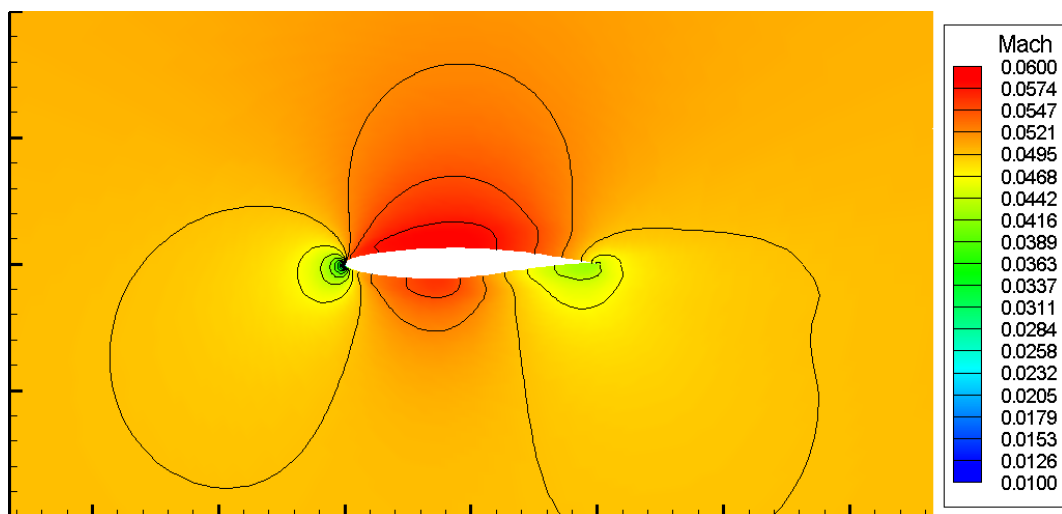


Figure 12. Mach number contours with preconditioning (RAE2822, AOA = 0°,  $M_\infty = 0.05$ )

on computational time, the convergence speed is not always greater because the evaluation of the preconditioning matrix for all grid nodes in each iteration is somewhat costly.

The gain in convergence speed, however, is not the only benefit of using preconditioning techniques. In fact, figure 6 illustrates how steady-state converged flow fields can be wrongly predicted by using the original formulation at low-speeds. The distributions of pressure coefficient ( $C_p$ ) are compared to the “exact” ones, which were obtained with XFOIL<sup>®</sup>. For each airfoil referred in figure 6, the error in the  $C_p$  distribution may lead to significant errors in the force and (specially) moment coefficients, since greater differences appears generally at trailing and leading edges.

To better compare the flow-field differences for the cases referred in figure 6, Mach number contours obtained with and without preconditioning are put side-by-side in figures 8 and 9 for the NACA0012, and in figures 11 and 12 for the RAE2822. In these four figures, the colormaps are adjusted for best contrast “individually”, but all the isolines displayed corresponds to the exact same values in order to highlight the clear flow-field differences between steady-state converged solutions with and without preconditioning.

## 5. CONCLUDING REMARKS

This work presented a simple way to implement a low-speed preconditioning method for the Euler equations of gas dynamics with particular emphasis on the finite difference formulation based on generalized curvilinear coordinates. Nevertheless, concepts and principles common to all of the preconditioning methods were discussed in a detailed and didactically manner. Several airfoil simulations were performed in order to investigate the benefits or even the necessity of using preconditioning algorithms within the context of aerospace applications. It is hoped that the present study may contribute in turning low-speed preconditioning techniques better understood among general CFD practitioners.

## Acknowledgements

The authors gratefully acknowledge the partial support for this research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grants No. 312064/2006-3 and No. 471592/2011-0.

## 6. REFERENCES

- [1] Blazek, J., *Computational fluid dynamics: principles and applications (1st ed.)*, Elsevier Science, Oxford, 2001.
- [2] Buelow, P. E. O., “Convergence Enhancement of Euler and Navier-Stokes Algorithms”, PHD Thesis, Department of Mechanical Engineering, The Pennsylvania State University, 1995.

- [3] Chorin, A. J., “A Numerical Method for Solving Incompressible Viscous Flow Problems”, *Journal of Computational Physics*, Vol. 2, 1967, pp. 12–26.
- [4] Fletcher, C. A. J., *Computational Techniques for Fluid Dynamics (2nd vol.)*, Springer-Verlag, Berlin, 1988.
- [5] Guillard, H., and Viozat, C., “On the behavior of upwind schemes in the low Mach number limit”, *Computers & Fluids*, Vol. 28, 1999, pp. 63–86.
- [6] Patankar, S. V., *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
- [7] Toro, E. F., *Riemann solvers and numerical methods for fluid dynamics (2nd ed.)*, Springer-Verlag, Berlin, 1999.
- [8] Unrau, D., and Zingg, D. W., “Viscous airfoil computations using local preconditioning”, *AIAA Paper*, No. 97–2027, 1997.
- [9] Weiss, J., and Smith, W. A. “Preconditioning applied to variable and constant density flows”. *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2050-2057.