

LOW MACH NUMBER PRECONDITIONING FOR 2D AND 3D UPWIND FLOW SOLUTION SCHEMES ON UNSTRUCTURED GRIDS

V.G. Asouti, D.I. Papadimitriou, D.G. Koubogiannis, K.C. Giannakoglou

Lab. of Thermal Turbomachines,
National Technical University of Athens,
P.O. Box 64069, 15710, Athens, Greece,
e-mail: vasouti@mail.ntua.gr

Keywords: Preconditioning

Abstract. *An existing method and the corresponding software for the numerical solution of the Euler and Navier–Stokes equations in high–speed flows is extended to account for low–speed flows as well. This is achieved through the multiplication of the governing equations by a precondition matrix which is defined at each grid node in terms of the local Mach number and ensures adequately clustered eigenvalues and, thus, optimal convergence characteristics at all flow speeds. A second–order upwind scheme is adapted to the preconditioned system of equations through appropriate assumptions, which are clearly presented in this paper. 2D or 3D, inviscid or turbulent flow problems are analyzed, in external aerodynamics and turbomachinery.*

1 INTRODUCTION

It is well known that the numerical solution of the compressible fluid flow equations for the low Mach regime suffers from slow convergence and increased computing cost. It is also known that the main reason for the performance degradation of the relevant software, based on time–marching schemes and the theory of hyperbolic system of equations, is the large disparity between acoustic waves and fluid speeds. To overcome this problem, completely different formulations for the prediction of low–speed or incompressible flows such as pressure correction and pseudo–compressibility methods, have been developed. However, maintaining and extending two different CFD tools by the same research group is, in fact, cumbersome. The only way to use the same time–marching solution method regardless of the flow speed, is through preconditioning.

Conceptually, preconditioning is based on the multiplication of the pseudo–time derivative by an appropriate precondition matrix without affecting the steady state solution. The precondition matrix is defined in terms of the local Mach number [1], [2], [3] and degenerates to the unit matrix at sonic speed. According to the hyperbolic system theory, the Jacobians of the convection terms are multiplied by the inverse of the precondition matrix and this gives rise to much more clustered eigenvalues compared to those of standard Jacobians. The selection of the precondition matrix depends on the vector of solution variables. In the literature, different precondition matrices have been proposed depending on whether the flow equations are solved in terms of $\vec{Q} = [p \ u \ v \ T]^T$ or $\vec{W} = [\rho \ \rho u \ \rho v \ E]^T$ (see [4] or [5] and [6], respectively).

The implementation of preconditioning in a numerical flow solver depends practically on the discretization scheme used. The present method [7], [8] is based on the finite volume technique for unstructured grids with an upwind scheme for the discretization of convection terms. The latter are computed by sweeping the grid edges and employing a 1D Riemann flow solver between the two edge nodes. To maintain the existing formulation, a couple of assumptions concerning the management of the precondition matrix are made. These assumptions concern even the residual of the iteratively solved equations and contribute to the elimination of pressure oscillations that the conventional system of equations produce at low Mach numbers, particularly close to the leading and trailing edges of airfoils.

2 GOVERNING EQUATIONS AND LOW-MACH PRECONDITIONING

Though this paper is concerned with both 2D and 3D all-speed flows, for the sake of simplicity, the analysis of the method will be restricted to 2D flows; any extension to 3D flows is straightforward and, thus, omitted. The 2D Euler equations for compressible flows are written, in conservative form, as follows

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} = 0 \quad (1)$$

where

$$\vec{W} = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{Bmatrix}, \quad \vec{F} = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{Bmatrix}, \quad \vec{G} = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{Bmatrix} \quad (2)$$

Here ρ is the density, u and v the velocity components, E the total energy per unit volume and p the pressure. Note that only steady flow simulations are of interest. Starting from eq. 1, the preconditioned equations are obtained by multiplying the pseudo-time derivative term by the inverse of an appropriate precondition matrix Γ , namely

$$\Gamma^{-1} \frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} = 0 \quad (3)$$

or

$$\frac{\partial \vec{W}}{\partial t} + \Gamma A_x \frac{\partial \vec{W}}{\partial x} + \Gamma A_y \frac{\partial \vec{W}}{\partial y} = 0 \quad (4)$$

where $A_x = \frac{\partial \vec{F}}{\partial \vec{W}}$ and $A_y = \frac{\partial \vec{G}}{\partial \vec{W}}$ are the Jacobian matrices for the conservative variables. Eq. 4 can also be written in terms of the primitive variable array $\vec{V} = [\rho \ u \ v \ p]^T$ as follows

$$\frac{\partial \vec{V}}{\partial t} + \bar{\Gamma} \bar{A}_x \frac{\partial \vec{V}}{\partial x} + \bar{\Gamma} \bar{A}_y \frac{\partial \vec{V}}{\partial y} = 0 \quad (5)$$

where \bar{A}_x and \bar{A}_y are the corresponding Jacobian matrices, $\bar{A}_x = M^{-1} A_x M$, $\bar{A}_y = M^{-1} A_y M$, $\bar{\Gamma} = M^{-1} \Gamma M$ and $M = \frac{\partial \vec{W}}{\partial \vec{V}}$.

As already mentioned in the introduction, the role of the precondition matrix Γ (and, subsequently, the role of $\bar{\Gamma}$) is to alleviate the disparity between acoustic waves and the fluid speeds which characterize the non-preconditioned (regular) system of flow equations and is the main reason for the slow convergence of any numerical solution method applied to low Mach number flows. This paper adopts the precondition matrix originally proposed by [6], namely

$$\bar{\Gamma} = \begin{bmatrix} 1 & 0 & 0 & -\frac{1-a}{c^2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & a \end{bmatrix} \quad (6)$$

where $a = \min[1, M^2]$ and M is the local Mach number; Γ can be obtained from $\bar{\Gamma}$, since $\Gamma = M \bar{\Gamma} M^{-1}$.

Through the application of the Gauss' divergence theorem, the integration of eqs. 5 over any finite volume cell leads to the integral of numerical fluxes crossing its boundary. Let $\vec{n} = (n_x, n_y)$ denote the normal vector to the boundary; the application of any upwind scheme to numerically compute these fluxes as well as the convergence characteristics of the resulting solution method is determined by the eigenvalues of the preconditioned directional Jacobian matrix

$$\bar{A}_\Gamma = \bar{\Gamma} \bar{A} = \bar{\Gamma} (\bar{A}_x n_x + \bar{A}_y n_y) \quad (7)$$

which are

$$\begin{aligned} \lambda_1 &= \lambda_2 = \vec{v} \cdot \vec{n} \\ \lambda_{3,4} &= \frac{1}{2} \{ (1+a) \vec{v} \cdot \vec{n} \pm \sqrt{[(1-a) \vec{v} \cdot \vec{n}]^2 + 4ac^2 |\vec{n}|^2} \} \end{aligned} \quad (8)$$

It can be shown that, for low Mach number flows, the eigenvalues, (eq. 8) are much more clustered than those of the non-preconditioned system $\lambda_1^* = \lambda_2^* = \vec{v} \cdot \vec{n}$, $\lambda_3^* = \lambda_4^* = \vec{v} \cdot \vec{n} + c|\vec{n}|$, giving thus rise to better convergence properties. Fig. 1 compares the ratio λ_3/λ_1 of the preconditioned and the non-preconditioned system λ_3^*/λ_1^* for two different ranges of the Mach number. It can clearly be observed that, through preconditioning, eigenvalues become much more clustered as Mach number approaches zero.

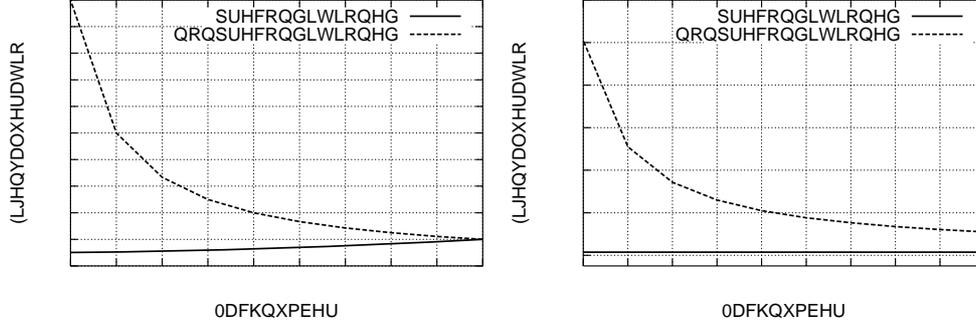


Figure 1: Comparison of the λ_3/λ_1 ratio for the preconditioned and the non-preconditioned system for $0.1 \leq M \leq 1$ (left) and $0.01 \leq M \leq 0.1$ (right).

3 DISCRETIZATION AND NUMERICAL SOLUTION

The preconditioned equations are integrated over vertex-centered finite volumes \mathcal{U} defined in fig. 2 for 2D problems. To carry out the integration, the assumption that Γ stays out of the integral, which facilitates considerably the subsequent development of equations, is made. The meaning of this assumption is that, in any vertex-centered cell, Γ remains fixed and equal to that defined at the enclosed node P . So

$$\iint_{\mathcal{U}} \frac{\partial \vec{W}}{\partial t} d\mathcal{U} + \Gamma \int_{\partial \mathcal{U}} (\vec{F} n_x + \vec{G} n_y) d\partial \mathcal{U} = 0 \quad (9)$$

which, through further analysis of terms, yields

$$\frac{\mathcal{U}_P}{\Delta t_P} \delta \vec{W}_P + \Gamma_P \sum_{Q \in nei(P)} \vec{\Phi}_{PQ} = 0 \quad (10)$$

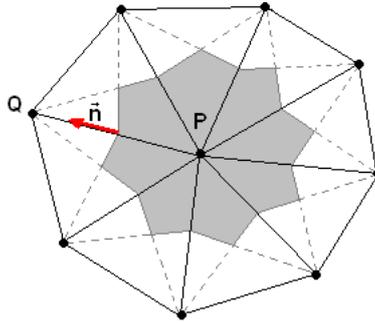


Figure 2: Control volume surrounding a node.

or, equivalently,

$$\frac{\mathcal{U}_P}{\Delta t_P} \delta \vec{W}_P + \Gamma_P \sum_{Q \in nei(P)} (\Gamma^{-1} \Gamma \vec{\Phi}_{PQ}) = 0 \quad (11)$$

where $nei(P)$ is the set of grid nodes that are linked with P through grid segments and $\vec{\Phi}_{PQ}$ is the numerical flux crossing the interface of the finite volumes defined around adjacent nodes P and Q . Between P and Q , the 1D Roe approximate Riemann solver [9] is employed, according to which

$$\vec{\Phi}_{PQ} = \frac{1}{2}[H(\vec{W}_P, \vec{n}_{PQ}) + H(\vec{W}_Q, \vec{n}_{PQ})] - \frac{1}{2}|\tilde{A}_{PQ}|\Delta\vec{W}_{PQ} \quad (12)$$

where \tilde{A}_{PQ} is the Roe-averaged Jacobian at the midnode between P and Q and $\Delta\vec{W}_{PQ} = \vec{W}_P - \vec{W}_Q$. For second order accuracy, $\vec{W}_L = \vec{W}_P + \frac{1}{2}\vec{PQ}\nabla\vec{W}_P$ and $\vec{W}_R = \vec{W}_Q + \frac{1}{2}\vec{PQ}\nabla\vec{W}_Q$ should substitute for \vec{W}_P and \vec{W}_Q , respectively. The last term in eq. 12 is rewritten as follows:

$$\begin{aligned} |\tilde{A}_{PQ}|\Delta\vec{W}_{PQ} &= |\tilde{\Gamma}_{PQ}^{-1}\tilde{\Gamma}_{PQ}\tilde{A}_{PQ}|\Delta\vec{W}_{PQ} \\ &\simeq \tilde{\Gamma}_{PQ}^{-1}|\tilde{\Gamma}_{PQ}\tilde{A}_{PQ}|\Delta\vec{W}_{PQ} \\ &= \tilde{\Gamma}_{PQ}^{-1}|\tilde{A}_{\Gamma_{PQ}}|\Delta\vec{W}_{PQ} \end{aligned} \quad (13)$$

Here $|\tilde{A}_{\Gamma_{PQ}}|$ is defined by

$$|\tilde{A}_{\Gamma_{PQ}}| = \tilde{P}_{\Gamma_{PQ}}|\tilde{\Lambda}_{\Gamma_{PQ}}|\tilde{P}_{\Gamma_{PQ}}^{-1} \quad (14)$$

where $\tilde{\Lambda}_{\Gamma}$ is the diagonal eigenvalue matrix of $\tilde{A}_{\Gamma_{PQ}}$ whereas \tilde{P}_{Γ} and \tilde{P}_{Γ}^{-1} are the diagonalization matrices composed of the right and left eigenvectors, respectively. Subscript Γ denotes that the matrices are derived from the preconditioned system and any quantity marked with $\tilde{}$ is Roe-averaged. Through eqs. 13 and 14, eq. 12, becomes

$$\vec{\Phi}_{PQ} = \frac{1}{2}[H(\vec{W}_P, \vec{n}_{PQ}) + H(\vec{W}_Q, \vec{n}_{PQ})] - \frac{1}{2}\tilde{\Gamma}_{PQ}^{-1}|\tilde{A}_{\Gamma_{PQ}}|\Delta\vec{W}_{PQ} \quad (15)$$

According to the stability criteria applied for the preconditioned system, the local time step is

$$\Delta t = \frac{CFL h_T}{\frac{1}{2}\{(1+a)|\vec{v}| + \sqrt{[(1-a)|\vec{v}|]^2 + 4ac^2|\vec{n}|^2}\}} \quad (16)$$

which is simplified to $\Delta t = \frac{CFL h_T}{|\vec{v}|+c|\vec{n}|}$ for the non preconditioned case ($a = 1$). In the 2D case, eq. 16 is used to compute Δt at each triangle using its minimum height h_T and the CFL number defined by the user. Time steps are then scatter-added to nodes.

4 RESULTS AND DISCUSSION

A number of test problems has been selected to demonstrate the capability of the programmed software to cope with all-speed flows. Our intention is not to demonstrate how accurate the computed results are, but to convince the reader that, through preconditioning, one can exploit software based on the theory of hyperbolic equations even in low Mach number flows where, by nature, time-marching methods are slow. No comparison with experimental or other reference computational results is shown, since the non-preconditioned method was adequately validated in the past at high subsonic and transonic flows. The demonstration that follows is concerned with inviscid and viscous flows, around an isolated airfoil, in a compressor cascade and around a complete aircraft. Any comparison concerning convergence speed is presented as a function of iterations. The CPU cost per iteration of the preconditioned system of equations is slightly higher than that of the non-preconditioned one, due to the excess number of floating point operations it involves. However, the difference in CPU cost is almost negligible and, thus, the comparison in terms of iterations can be interpreted as a comparison in terms of cost.

The first problem is concerned with the computation of the inviscid flow around the isolated NACA12 airfoil. The same unstructured grid (950 nodes, 1800 triangles) was used to predict the flow field at three different infinite Mach numbers, namely $M_\infty = 0.1, 0.01$ and 0.005 , with the same infinite flow angle $\alpha_\infty = 5^\circ$. Fig. 3 compares the residual drop in terms of iterations. Note that no stopping criterion was used, so any comparison between the performance of the non-preconditioned and preconditioned equations can be objectively quantified, depending on the desired maximum allowed residual. In all cases, preconditioning leads to better convergence characteristics; it is clear that the lower the Mach number the lower the computing cost.

In the $M_\infty = 0.1$ case, the gain achieved through preconditioning is not that important, though it does exist. For higher M_∞ values, both systems give very similar convergence characteristics and this can

be explained by the precondition matrix form, eq. 6, which tends to the unit matrix. By comparing the residual curves of the preconditioned equations in all three cases, it can be seen that they remain close to each other whereas the convergence of the non-preconditioned equations drifts much more slowly as the Mach number decreases. In the cases $M_\infty = 0.01$ and 0.005 , assuming a stopping criterion for the residual equal to 10^{-10} , preconditioning leads to convergence of about five to eight times lower CPU cost.

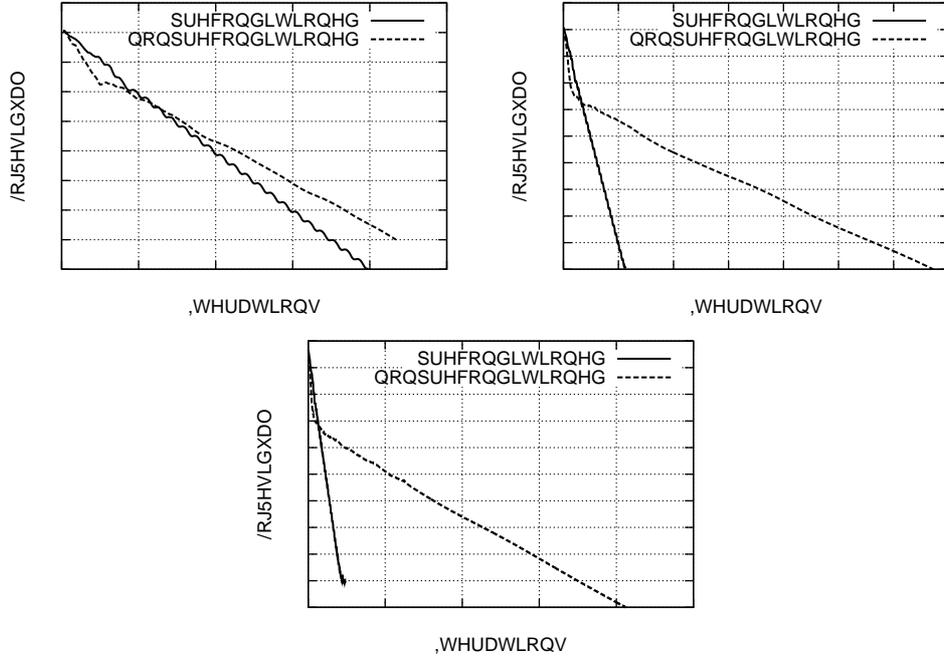


Figure 3: Inviscid flow around the NACA12 airfoil. Convergence diagrams for $M_\infty = 0.1$ (top-left), $M_\infty = 0.01$ (top-right) and $M_\infty = 0.005$ (bottom).

Fig. 4 shows a close up view of the unstructured grid used and the Mach number contours computed through the preconditioned equations around the airfoil, for $M_\infty = 0.005$.

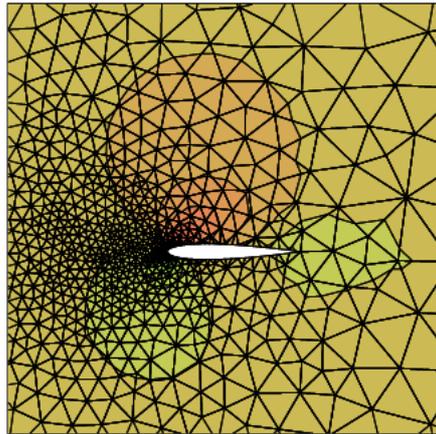


Figure 4: Inviscid flow analysis for NACA12 airfoil. Mach number contours for $M_\infty = 0.005$, and Mach number increment $\Delta M = 0.0002$.

Fig. 5 compares the pressure distribution around the airfoil, produced by the preconditioned and non-preconditioned solver with the same computing cost. Close to the leading and trailing edges, the preconditioned equations eliminate non-physically accepted pressure kinks, thanks to the modified last term in eq. 12 which acts as smoothing term.

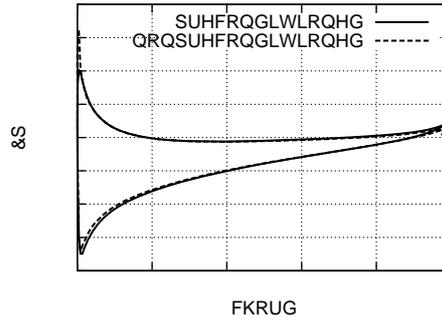


Figure 5: Inviscid flow analysis for NACA12 airfoil. Pressure coefficient distribution ($M_\infty = 0.1$).

The second problem examined is that of the inviscid flow analysis in a 3D compressor cascade. The exit isentropic Mach number equals to 0.1 or 0.3 and $\alpha_1 = 47^\circ$. A 2D unstructured grid (1800 nodes, 3300 triangles) was generated at first which was then stacked in the spanwise direction to create the 3D unstructured grid with 40000 tetrahedra and 9000 nodes. Symmetry conditions were employed over the upper and the lower plane in the spanwise direction. Fig. 6 shows the residual convergence history for $M_{2, is} = 0.3$ and 0.1. In the high Mach case, both solvers, either with or without preconditioning, converge easily; however, the preconditioned equations converge faster. In the low Mach number case, the preconditioned equations solver converges within 400 iterations (the stopping criterion for the residual is the same as in the previous case); on the other hand, without preconditioning, even 4000 iterations do not suffice to get an adequately converged solution.

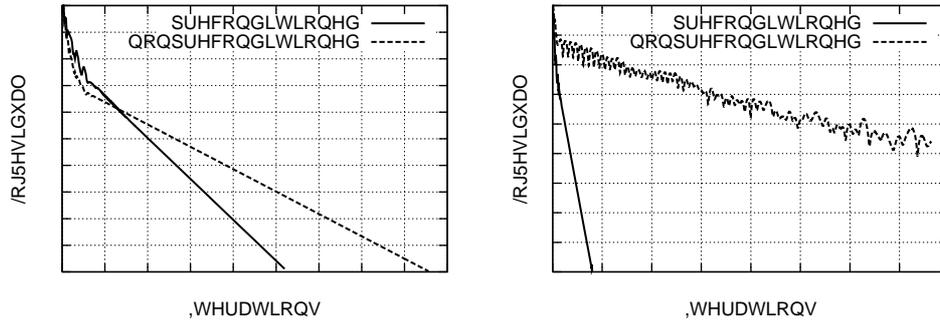


Figure 6: Inviscid flow in a 3D compressor cascade. Convergence diagrams for $M_{2, is} = 0.3$ (left) and $M_{2, is} = 0.1$ (right).

The turbulent flow over the same linear cascade was also analysed. The 2D grid consists of 5000 nodes and 9400 triangles, which results to a 3D grid with 170000 tetrahedra and 34700 nodes. The Spalart–Allmaras one–equation turbulence model [10] was used along with the wall function technique; in particular, a non zero (slip) velocity was allowed to occur over the blade nodes through assuming that the real solid wall is located at distance δ from the boundary node; δ is a user-defined parameter so that boundary nodes be in the logarithmic region of the boundary layer. Fig. 7 shows the calculated Mach number contours and compares the convergence of the preconditioned and non–preconditioned equations for $Re = 100000$ and $M_{2, is} = 0.1$. With the maximum allowed residual value be equal to 10^{-5} , the preconditioned equation converge at half the CPU cost of the conventional solver. The lower the stopping residual threshold, the more important the CPU gain.

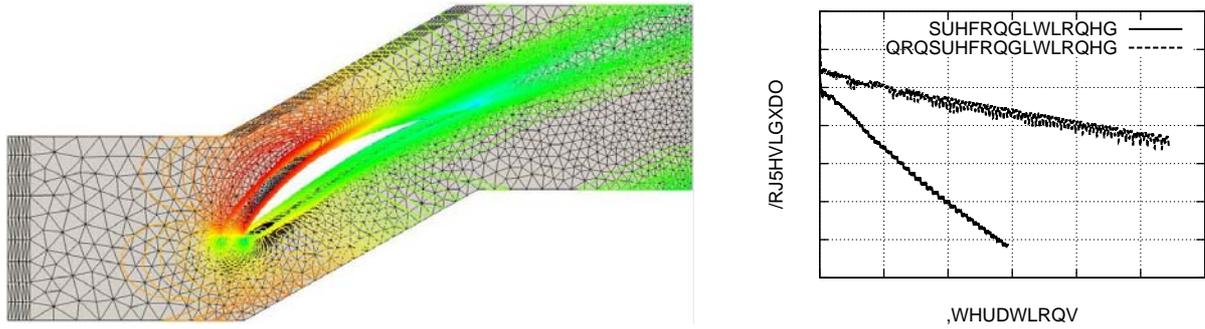


Figure 7: Turbulent flow in a 3D compressor cascade ($Re = 100000$). Mach number contours of the preconditioned system (left) and convergence diagram (right).

Last case is that of the computation of the inviscid flow around a complete aircraft. The computational grid generated around half of the aircraft (due to symmetric flow conditions) consists of 256000 tetrahedra and 45000 nodes.

This case was studied for $M_\infty = 0.1$ and $\alpha_\infty = 0^\circ$. Fig. 8 shows the convergence curves for the preconditioned and non-preconditioned equations and the iso-Mach contours over the aircraft surface. The preconditioned equations converge faster and the gain in CPU is expected to increase at lower Mach numbers.

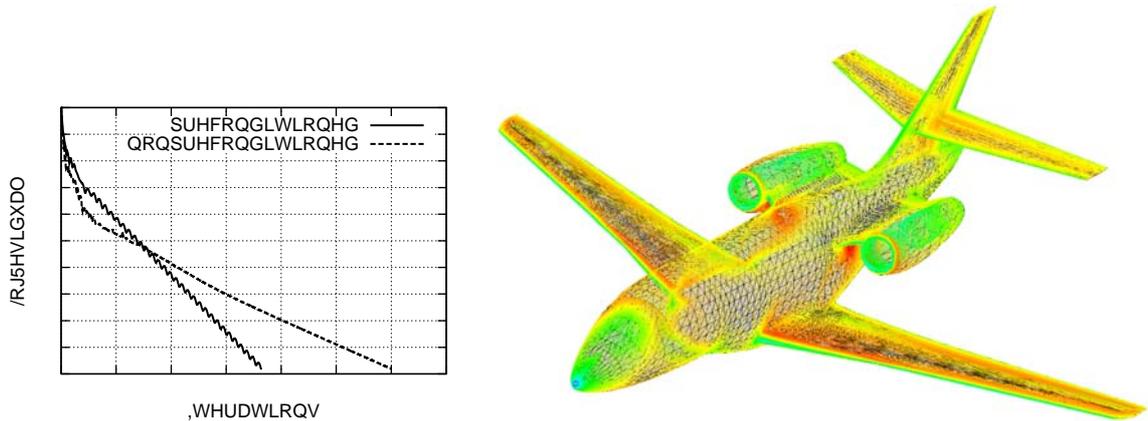


Figure 8: Inviscid flow around an aircraft. Convergence diagram (left) and iso-Mach contours for $M_\infty = 0.1$ (right).

5 CONCLUSIONS

The implementation of low-Mach preconditioning in a time-marching, primitive variable flow solver can increase its robustness, yielding equally satisfactory convergence at all flow speeds. A couple of assumptions is made during this implementation in the context of a second-order upwind scheme, as demonstrated in this paper. These assumptions often affect positively the accuracy of the predictions by improving the quality of the solution in areas close to leading and trailing edge of an airfoil. The proposed method leads to a considerable economy in CPU cost which becomes more important as the Mach number decreases.

AKNOWLEDGEMENTS

The first two authors were supported by grants from the Research Committee of the National Technical University of Athens and the Beneficial Foundation Alexandros S. Onasis.

References

- [1] Turkel E., Radespiel R., Kroll N. "Assesment of Preconditioned Methods for Multidimensional Aerodynamics", *Computers & Fluids*, Vol. 26, No.6, pp. 613-634, 1997.
- [2] van Leer B., Lee W.T., and Roe P., "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations", *AIAA Paper 91-1552*, 1991.
- [3] Choi Y.H., and Merkle C.L., "The Application of Preconditioning in Viscous Flows", *Journal of Computation Physics*, Vol. 105, No. 2, pp. 207-233, 1993.
- [4] Turkel E., "Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations", *Journal of Computation Physics*, Vol. 72, No.2, pp. 277-298, 1987.
- [5] Weiss J.M., Smith W.A., "Preconditioned Applied to Variable and Constant Density Flows", *AIAA*, Vol. 33, No.11, Paper 94-2209, 1995.
- [6] Eriksson Lars-Erik, "A preconditioned Navier-Stokes Solver for Low Mach Number Flows", *Computational Fluid Dynamics*, 1996.
- [7] Giannakoglou K.C., Koubogiannis D.G., Giotis A.P. and Rovas D.V., "A 3-D Solver Using Unstructured Grids with Tetrahedral Elements", *Proc. of the 5th National Congress on Mechanics*, Ioannina, pp. 973-980, August 1998.
- [8] Koubogiannis D.G., Athanassiadis A.N. and Giannakoglou K.C., "One- and Two-Equation Turbulence Models for the Prediction of Complex Cascade Flows Using Unstructured Grids", *Computers & Fluids*, 32, pp. 403-430, 2003.
- [9] Roe P. L. "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", *Journal of Computation Physics*, Vol. 43, pp. 357-372, 1981.
- [10] Spalart P. R. and Allmaras S. R., "A One-Equation Turbulence Model For Aerodynamic Flows", *AIAA*, Paper 92-0439, 1992.