THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
345 E. 47th St., New York, N.Y. 10017

97-GT-416



# A NAVIER-STOKES INVERSE METHOD BASED ON A MOVING BLADE WALL STRATEGY



**Olivier Léonard and Alain Demeulenaere**
Université de Liège - Institut de Mécanique (C3)
Rue Ernest Solvay 21 - 4000 Liège - Belgium

## ABSTRACT

A two-dimensional viscous inverse method for the design of compressor and turbine blades is presented. An initial geometry is modified iteratively to reach a target pressure distribution imposed on the blade surfaces. The Navier-Stokes equations are solved in a numerical domain of which some boundaries (the blade walls) move during the transient part of the computation. The blade modifications are based on the transpiration principle. The transpiration flux is computed from the difference between the actual and the prescribed pressure distributions.

A high-resolution Navier-Stokes solver has been developed for this purpose, based on a Finite-Volume formulation. Multi-block structured grids allow for a selective concentration of discretization points in the zones of higher gradients. Both explicit (Runge-Kutta) and implicit (Newton-Krylov) time integration schemes have been implemented.

Applications to turbine and compressor blade design illustrate the accuracy of the flow computation, and the efficiency of the inverse method.

## NOMENCLATURE

| | |
|---|---|
| $a$ | speed of sound |
| $M$ | Mach number |
| $p$ | static pressure |
| $V$ | volume |
| $E$ | total energy |
| $t$ | time |
| $\rho$ | density |
| $U$ | vector of conservative variables |
| $\vec{V}$ | velocity vector |
| $\vec{x}$ | position vector |
| $\vec{V}_g$ | grid-point velocity vector |

### Subscripts

| | |
|---|---|
| $n$ | normal component |
| $x$ | axial component |
| $y$ | circumferential component |
| $l$ | left state |
| $r$ | right state |

### Superscripts

| | |
|---|---|
| $req$ | required value |
| $act$ | actual value |

## INTRODUCTION

Improvement of the aerodynamic performance of a turbomachinery element requires a correct tailoring of the velocity distribution along its walls. The velocity distribution determines the aerodynamic load, the behaviour of the viscous layers in which originate the losses, and the heat flux distribution which may be strongly affected by the relative importance of the laminar, transitional and turbulent zones.

Most designers still adopt a 'direct' approach for tailoring: they evaluate the performance of the actual geometry using numerical and/or experimental tools, and they modify it according to empirical rules or to their own experience. This approach can be very time consuming, and can turn out to be uaefficient in some cases. It is clear now that more powerful design strategies can only be obtained by using 'design' methods, which allow for the generation of geometries achieving imposed performance.

Optimization techniques use numerical analysis tools as black boxes and replace the designer by a systematic calculation of the sensitivity of the performance to geometrical parameters. In this approach, a set of geometrical parameters is iteratively modified until the best combination is

Presented at the International Gas Turbine & Aeroengine Congress & Exhibition
Orlando, Florida — June 2–June 5, 1997

found, which minimizes a certain objective function. The cost of such a process is still so high that the flow solvers are based on 'cheap' models, or that the number of parameters used to define the geometry is drastically restricted. The major advantages of optimization methods rely on the capability to find a solution even if the required performance is partly unrealistic - although most of these methods can not guarantee that the true minimum is reached (Jameson, 1995, Eyi et al, 1995).

Inverse techniques produce geometry modifications directly determined from the difference between an existing performance level and a required one, through a physical link. In most of the inverse codes, a required velocity or pressure distribution is imposed as a boundary condition along a part of or the whole blade wall. The major drawback of inverse methods is related to the existence of a solution, non-physical target pressure distributions usually leading to divergence of the solver. Their main advantage rely on the flexibility that they provide for the tailoring of the velocity distribution: there are as many geometrical parameters as points on the blade walls - typically 100...300... for 2D problems. Local modifications can be undertaken, such as leading edge acceleration or the suppression of a shock - boundary layer interaction.

Three-dimensional blade design is still performed by iterating on blade-to-blade and throughflow codes, with aerofoil stacking (for a good review of the design procedures, see Jennions (1994) or Vuillez and Petot (1994). Up to now most of the existing design methods are two- or quasi-three-dimensional, and based on inviscid models. Navier-Stokes codes are more likely used during the verification process than during the design phase.

The blade geometry can be directly deduced from the target velocity distribution only for potential flow problems (Lighthill, 1945, Stanitz, 1953, Korn et al, 1972, Schmidt, 1980, Dulikravitch and Sobieczky, 1982, Sanz, 1988), with the help of a mathematical frame. As soon as the Euler flow model is used, the equations are solved in a domain of which some boundaries are the solution of the problem, resulting in an iterative process: an initial geometry provides a first guess of the calculation domain and of the flow field, and is modified until the target velocity distribution is achieved (Meauzé, 1982, Zannetti, 1987, Giles and Drela, 1987, Léonard and Van den Braembussche, 1992).

Inviscid inverse methods provide 'inviscid' geometries, from which the boundary layers displacement thickness must be subtracted to obtain 'metallic' geometries. This subtraction process is critical and may compromise the precision of the whole design procedure. If a smooth 'inviscid' shape can be expected from a smooth target pressure distribution, boundary layer codes can provide boundary layer displacement thickness distributions somewhat irregular, especially in the transition zone. In most cases, the resulting 'metallic' blade shape needs to be smoothed, whith the risk that the tailoring effort can be strongly affected.

There is another well known problem related to inviscid inverse methods. A preliminary boundary layer calculation is performed, based on the target velocity distribution, which provides a minimum displacement thickness that must be accounted for in the design procedure, in addition to a minimum 'metallic' thickness. This results in a rather large trailing edge thickness, especially for compressor blades. If rounded-off, this thick trailing edge induces a non-physical acceleration followed by a strong slow-down of the fluid. The authors have rather introduced blunt-type trailing edges, through which fluid is blown. The fluid is forced to leave the blade in the camber line direction, resulting in a more realistic velocity distribution. Furthermore blunt trailing edges provide an easy control of their thickness between imposed minimum and maximum values. It is not clear however how much fluid should be ejected, what should be its velocity, and what are the consequences of this (local) lack of conservativity.

At transonic flow conditions, the pressure distribution along a turbine blade suction side is much influenced by the flow behaviour in the trailing edge zone, which has always been hardly predicted by inviscid computations. The use of blunt trailing edges has allowed for good predictions, as the computed streamlines are close to the 'real' ones in the trailing edge zone.

Solving the Navier-Stokes equations is probably the only way to avoid these modelling problems and the boundary layer subtraction, so that the pressure distribution is prescribed on the 'real' blade shape, and so that the accuracy of the tailoring depends only on the ability of the solver to predict the pressure distribution along the blade walls.

## DESCRIPTION OF THE METHOD

The two-, quasi-three- and three-dimensional inverse codes initially developed by the authors (Léonard and Van den Braembussche, 1992, Demeulenaere and Van den Braembussche, 1996) are based on a robust finite volume Euler solver. One design cycle includes an 'inverse' calculation, wherein the flow simulation is performed until convergence with the target pressure distribution imposed as a boundary condition on **fixed** and **permeable** walls, a transpiration process for the geometry modification, and a flow analysis on the new blade shape. Usually 3 to 10 cycles are needed, depending on the amplitude of the geometry changes.

When applied to viscous flow computations, the well-posedness of permeable wall boundary conditions is questionable, from the physical as well as from the mathematical point of view. Therefore the strategy of modifying the blades at each iteration of the time-marching algorithm has been introduced. The flow computation is performed on a **moving** mesh with **impermeable** blade walls, and follows the transient which occurs while the blade walls move and finally reach a steady shape.

Each iteration of the inverse procedure is separated into two steps:

- the first step determines the change in blade geometry (and mesh), and is based on the transpiration concept.

The transpiration flux is governed by the difference between the actual and prescribed pressure distributions;

- the second step of the procedure updates the flow field, by means of one single time iteration, with impermeable blade walls, and a moving mesh.

In this procedure, the prescribed pressure is not explicitly imposed along the blades. However, the blade walls are modified as long as the pressure distribution is different from the prescribed one. Therefore the flow computation converges to a steady state only when the prescribed pressure distribution is reached on the blade surfaces.

## THE GEOMETRY MODIFICATION STEP

The blade walls are modified by means of a transpiration model, which makes use of a distribution of normal and tangential velocities along the suction and pressure sides, to approximate the position of the two stagnation streamlines. These streamlines form the new blade shape.

As mentioned above, in the authors' previous inviscid method, the normal velocity distribution is obtained from a time-marching computation with permeable wall boundary conditions along the walls. At each iteration, the normal velocities are obtained from the compatibility relation associated to the outgoing pressure wave (Léonard, 1990):

$$V_n = V_n^* + \frac{p^* - p^{req}}{\rho a} \tag{1}$$

where the state $^*$ refers to a first approximation, extrapolated from the inside flow.

The same compatibility relation is used for inviscid 'direct' calculations, to derive the pressure from the imposed zero normal velocity. The direct and inverse solvers are then 'compatible,' meaning that once convergence is reached, both 'inverse' and 'direct' boundary conditions maintain the stable converged solution.

The same approach is still adopted in the inviscid version of the present method, but only one time iteration is performed.

The relation (1) has been modified for viscous problems, so that the 'inverse' boundary condition is also compatible with the 'direct' one, which imposes zero velocity and zero normal pressure gradient:

$$V_n = \frac{p^{act} - p^{req}}{\rho a} \tag{2}$$

The normal velocity distribution is simply deduced from the difference between the actual and prescribed pressure, and no time-marching iteration is needed.

The existence of a solution to Euler and Navier-Stokes inverse problems (a manufacturable closed profile, and inlet/outlet flow conditions meeting the design requirements) can never be guaranteed. In our method, after completion of the transpiration procedure, the resulting suction and pressure sides are slightly rotated around the leading edge, so that a prescribed trailing edge thickness is obtained. This ensures that the blade shape remains closed, even if the solution corresponding to the prescribed pressure distribution is not a closed contour. In this latter case, the solver does not necessarily diverge, hut the convergence history usually stagnates.

Another way to treat the problem of profile closure is the addition of a degree of freedom in the prescribed pressure distribution, which is automatically adjusted during the transient of the computation. This degree of freedom is also used to meet other requirements, such as the outlet flow angle and/or the mass flow. Other authors proposed to impose the pressure along the suction side only, or the pressure difference between the pressure and the suction sides (Meauzé, 1982, Zannetti, 1987). The whole blade shape is deduced from an imposed blade thickness distribution. This approach offers lower control on the blade performance, and has rather been applied to the design of thin compressor blades.

## THE FINITE VOLUME STEP

After the blade shape modification, a new mesh is generated around the new profile. The displacements of all grid points are computed, from which a 'grid-point velocity' field is deduced.

The conservation of mass, momentum and energy is applied to a control volume, whose faces are moving with the velocity $\vec{V}_g$:

$$\frac{d}{dt}\int_V \rho dV = \oint_S \rho(\vec{V} - \vec{V}_g)\vec{n}dS \tag{3}$$

$$\frac{d}{dt}\int_V \rho u_i dV = \oint_S [\rho u_i(\vec{V} - \vec{V}_g)\vec{n} + pn_i]dS \tag{4}$$

$$\frac{d}{dt}\int_V \rho E dV = \oint_S [\rho E(\vec{V} - \vec{V}_g) + p\vec{V}]\vec{n}dS \tag{5}$$

where $\vec{n}$ is the inward pointing normal vector.

### Space Conservation Law (SCL)

In a uniform flow field with zero-velocity, the above equations lead to the so-called Space Conservation Law (SCL):

$$\frac{d}{dt}\int_V dV + \oint_S \vec{V}_g\vec{n}dS = 0 \tag{6}$$

It is clear that if this relation is not respected, artificial mass sources or sinks are introduced, leading to instabilities. The grid-point velocities must therefore be defined in such a way that the integral of $(\vec{V}_g\vec{n}dS)$ and the volume change of the control element cancel each other out.

In the simple case of a two-dimensional cartesian grid (fig. 1), one can easily observe that if both sets of grid lines move, the grid-point velocities can not be simply computed from the displacements of the central points of the faces. For the element considered here, the error introduced is shown as a

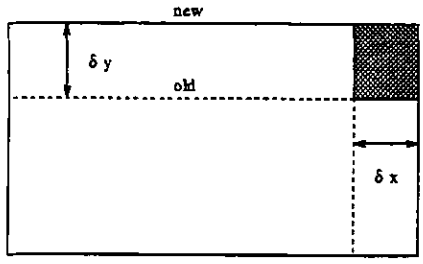


**Fig. 1: Change of control volume in $\Delta t$**

shaded area in figure 1, and is proportional to the square of the time step ($\delta x \delta y \Delta t^2$). Not-satisfying of the SCL therefore imposes an additional constraint on the time step size, which in many cases may be more severe than the temporal discretization stability constraint (Demirdzic and Peric, 1988).

The volume of a mesh cell is given by:

$$V = \frac{1}{2}\oint_S \vec{x} d\vec{S} = \frac{1}{2}\oint_S (x n_x + y n_y) dS \qquad (7)$$

The expression of the volume change from one time iteration to the other is then:

$$\delta V = \frac{1}{2}\oint \vec{x}^{new} d\vec{S}^{new} - \frac{1}{2}\oint \vec{x}^{old} d\vec{S}^{old}$$

This expression can be rewritten as follows:

$$\begin{aligned} \delta V &= \frac{1}{2}\oint (\vec{x}^{new} - \vec{x}^{old})(d\vec{S}^{new} + d\vec{S}^{old}) \\ &- \frac{1}{2}\oint (\vec{x}^{new} d\vec{S}^{old} - \vec{x}^{old} d\vec{S}^{new}) \end{aligned}$$

Rewriting this equation with the expressions of the normal vectors, one can show that the second integral is zero. The two components of the scalar product give rise to expressions of the grid-point velocity components which guarantee the SCL to be respected:

$$V_{gx} = \frac{\delta x}{\Delta t}\frac{S_x^{old} + S_x^{new}}{2S_x^{new}} \qquad (8)$$

$$V_{gy} = \frac{\delta y}{\Delta t}\frac{S_y^{old} + S_y^{new}}{2S_y^{new}} \qquad (9)$$

The velocity components are divided respectively by $S_x^{new}$ and $S_y^{new}$, which means that the time iteration is performed on the 'new' mesh.

## Additional driving terms

If the time derivative is split into two terms, the discretized form of the mass conservation equation (Eq.3) becomes:

$$\Delta\rho = \left[\sum_{faces} \rho \vec{V}\vec{n}\Delta S - \left(\sum_{faces} \rho \vec{V}_g \vec{n}\Delta S + \rho\frac{\Delta V}{\Delta t}\right)\right]\frac{\Delta t}{V}$$

The first right-hand-side term is the residual for a fixed mesh, while the two additional terms are due to the mesh movement. Their physical meaning is easily explained. The advective fluxes must be measured by the 'relative' velocity, obtained by subtracting the mesh velocity from the flow velocity, resulting in the first additional term. The advective fluxes entering the control volume during a time step cause a variation of mass in the control element, which is the product density × volume. In case of moving grids, both can change, which explains the presence of the second additional term.

In the frame of the inverse method, the two additional terms resulting from the mesh movement converge to zero at steady state. Moreover it is clear that the two terms should compensate each other in smooth flows, provided that the SCL is satisfied. It could be concluded from this that the presence of those terms is not necessary for steady computations, as proposed by Demirdzic and Peric (1990) for incompressible flows. However, experience has shown that these additional terms could in some cases be essential for a stable and monotonic convergence, especially at transonic flow conditions with shocks (Demeulenaere et al, 1997).

The effect of the additional terms is illustrated by the inviscid example of the redesign of the subcritical blade proposed by Sanz (AGARD-AR-275), which has been redesigned for a higher incidence (inlet flow angle of 42 deg instead of 36 deg). The initial isentropic Mach number distribution on the blades is compared to the target in figure 2. The target distribution presents a smoother acceleration along the whole suction side, especially in the leading edge region. The inverse method has been tested with and without the additional terms. The convergence histories are plotted in figure 3. The density error curves show that even at subsonic flow conditions, the additional driving terms al-

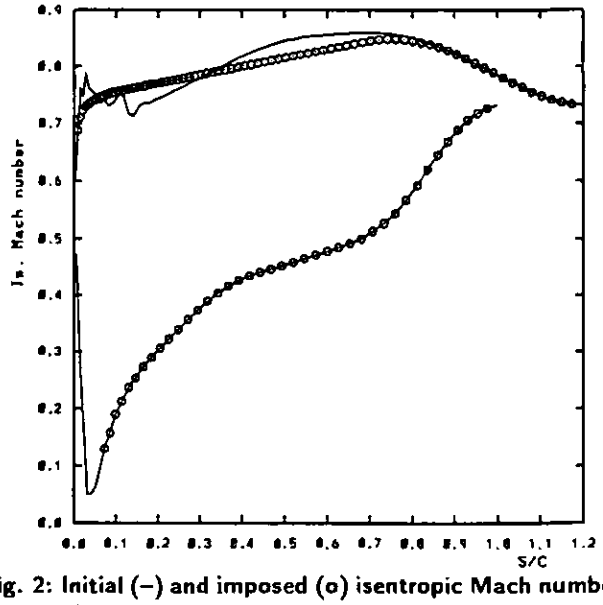


**Fig. 2: Initial (–) and imposed (o) isentropic Mach number distributions on the blades**

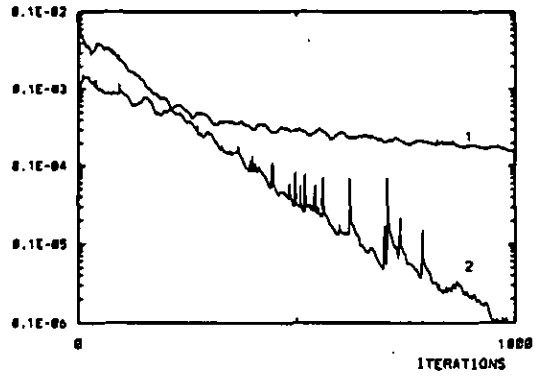


**Fig. 3: Convergence history (density error) - With (2) and without (1) the additional terms**

low for a significant increase of the convergence rate. One can also observe that the computations including the driving terms are more sensitive to the constraints applied on the geometry (constant pitch-to-chord ratio, control of the trailing edge gap), resulting in an irregular convergence history. This is explained by the fact that an active constraint modifies the grid-point velocity field, which is directly introduced in the equations.

These driving terms are considered by the authors as an acceleration technique for steady computations, which is used simultaneously with local time stepping, enthalpy damping (for inviscid computations only), and residual averaging. These techniques do not influence the steady solution. It should be noticed that as the time step differs from one cell to the next, the time, and therefore the grid-point velocities, do not really have a physical meaning. Moreover as the main goal is to obtain a fast convergence to steady state, it is clear that if the additional driving terms imply the suppression of the local time stepping, it is not the worth while introducing them.

For Euler-type grids, the time step size does not differ significantly from one cell to the next, so that the grid-point velocities remain consistent with each other. For the highly stretched grids used in Navier-Stokes computations, the local time stepping induces large time step variations in the direction normal to the wall. The additional terms become unphysical in the wall region, and cause the divergence of the calculation. For this reason, they are only activated outside of the viscous layers, so that local time stepping and the additional terms can be used simulaneously.

### Advective fluxes computation

The 'relative' advective flux through a face is given by:

$$H_n(U) = \begin{pmatrix} \rho(V_n - V_{gn}) \\ \rho u(V_n - V_{gn}) + p n_x \\ \rho v(V_n - V_{gn}) + p n_y \\ \rho E(V_n - V_{gn}) + p V_n \end{pmatrix} \qquad (10)$$

The advective fluxes are splitted in function of the propagation of the information, according to the so-called Flux Vector Splitting (van Leer, 1982) or to the so-called Flux Difference Splitting (Roe, 1981).

The classical Roe formula is rewritten, with the modified expression for the fluxes $H_n(U_l)$ and $H_n(U_r)$:

$$H(U_l, U_r) = \frac{H_n(U_l) + H_n(U_r)}{2} - \sum_{i=1}^{4} |\lambda_i| \vec{r}_i \delta\vec{W} \qquad (11)$$

In this expression, the eigenvalues $\lambda_i$ are calculated by $V_n - V_{gn}$, $V_n - V_{gn}$, $V_n - V_{gn} + a$ and $V_n - V_{gn} - a$, because the propagation speed of the waves must be computed relatively to the mesh (Trépanier et al, 1991). On the contrary, the right eigenvectors $\vec{r}_i$ and the characteristic variables $\delta W_i$ are not modified by the mesh movement.

In the expressions of the flnxes proposed by van Leer, the normal velocity $V_n$ can simply be replaced by $\overline{V_n} = V_n - V_{gn}$, except for the energy equation (Eq. 5), as the pressure is multiplied by the absolute velocity (Singh et al, 1995):

$$H^{\pm} = F_m^{\pm} \begin{pmatrix} 1 \\ u \pm (2 \mp \overline{M}_n) \frac{a n_x}{\gamma} \\ v \pm (2 \mp \overline{M}_n) \frac{a n_y}{\gamma} \\ \frac{2a^2}{\gamma^2 - 1} \pm \frac{\overline{M}_n a^2 (2 \mp \overline{M}_n)}{\gamma + 1} + \frac{V^2}{2} + V_{gn} \frac{-\overline{V_n} \pm 2a}{\gamma} \end{pmatrix}$$

with $F_m^{\pm} = \pm \frac{\rho a}{4} (1 \pm \overline{M}_n)^2$.

## THE NAVIER-STOKES SOLVER

### Space Discretization

The Navier-Stokes solver (Léonard et al, 1997) is based on a Finite-Volume approach. Structured grids are used to discretize the numerical domain. A multi-block strategy allows the use of hyperbolic O- and C-grids along the blades, providing a perfect orthogonality of the mesh cells near the walls. The control elements are identical to the mesh cells, resulting in a cell-centered approach.

As mentioned above, an upwind-biased evaluation of the advective fluxes is adopted. It allows for a low numerical entropy generation, and a sharp capturing of the grid-aligned shock waves. The advective fluxes are computed from a left and a right state, which are evaluated at the face centers by means of third-order Taylor expansions along the grid directions. This procedure is generally referred to as the MUSCL approach (van Leer, 1979). Non-linear corrections are introduced in the form of limiting functions in order to avnid oscillations in the solution near discontinuities.

The viscous fluxes are determined for each segment from the velocity and temperature gradients. These gradients are calculated from Green's theorem, using the same control elements as for the residuals, i.e. the mesh cells. This technique is particularly well suited for multi-block configurations.

The gradients are stored at the cell centers and are evaluated at the segment centers by interpolation. One-sided derivatives must also be computed along the blade walls for a better precision on the shear stress and on the heat flux. The molecular viscosity is modelled by the Sutherland law. The Baldwin-Lomax algebraic turbulence model is used for turbulent flows.

The conservativity of the discretization is ensured by an appropriate treatment of the irregular block boundaries: the fluxes are evaluated separately for every interface segment defined by the succession of grid points belonging to the two connected blocks, and sent to the corresponding control elements of the two blocks.

## Time Discretization

Our purpose is to compute steady-state solutions as fast as possible. The code has been initially based on a 1st-order in time Runge-Kutta scheme with acceleration techniques such as local time-stepping, enthalpy damping (which cannot be employed for Navier-Stokes problems) or residual averaging. An implicit time-stepping has been also implemented and allows for impressive speed-ups with respect to explicit methods. An Euler-backward time-stepping is used to discretize the time derivatives, resulting in

$$\mathcal{F}(U^{l+1}) = \frac{U^{l+1} - U^l}{\Delta t} - \mathcal{R}(U^{l+1}) = 0 \qquad (12)$$

where $\mathcal{R}(U)$ corresponds to the discretization of the spatial derivatives. The solution $U^{l+1}$ is found through a Newton-Raphson iterative process. For steady calculations, many authors usually limit the number of Newton-Raphson iterations to one.

The main part of the computational effort is devoted to the solution of the linear system $\mathcal{J}(U^l)\,\delta U = \mathcal{R}(U^l)$ where $\mathcal{J}(U) = \frac{1}{\Delta t} - \frac{\partial \mathcal{R}}{\partial U}$ is the Jacobian of $\mathcal{F}$. As an exact solution is in most cases not justified, it is reasonable to solve it approximately with an iterative solver, saving a lot of CPU time with respect to a direct solver. Algorithms such as Jacobi or Gauss-Seidel are attractive because they are easy to implement and to vectorize. However their robustness and convergence rate are only ensured when the matrix is characterized by large diagonal terms, which is unlikely for stiff problems such as those related to Navier-Stokes equations without an important under-relaxation, i.e. small time-steps that reduce the convergence speed. On the contrary, Krylov subspace methods such as the conjugate gradients can be of great interest in this context. Among others, the GMRES is designed to solve non-symmetric (and possibly indefinite) linear systems (Saad, 1994). One of the most interesting features of the GMRES is maybe that it does not require the calculation of the true Jacobian, but only the calculation of a matrix-vector product $\mathcal{J}p$ where $p$ is any solution vector. For non linear equations, this product can be evaluated by a finite-difference quotient of the form

$$\mathcal{J}(U)p = \frac{\mathcal{F}(U + \epsilon p) - \mathcal{F}(U)}{\epsilon} \qquad (13)$$

The convergence of Krylov type solvers can be accelerated by a suitable preconditioning of the system that clusters matrix eigenvalues to each other. In our code an Incomplete LU decomposition is implemented in its most simple form. Preconditioning requires an approximate form of the Jacobian of the system. As suggested by many authors, this approximate Jacobian is built up using a first-order reconstruction and a Van Leer splitting of the advective fluxes only.

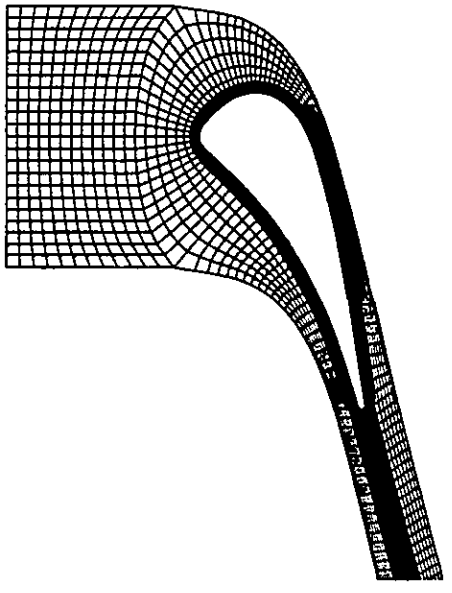

**Fig. 4: General view of the multi-block mesh**

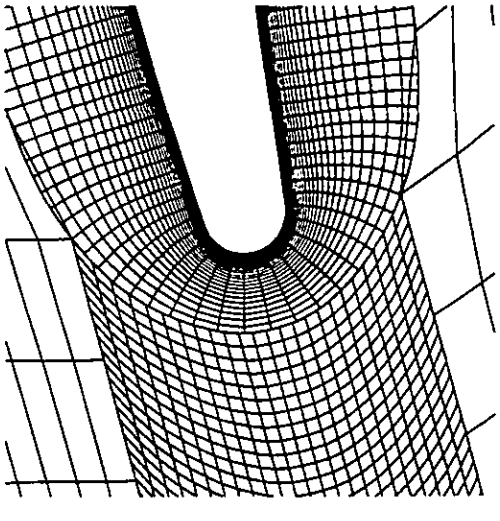

**Fig. 5: Detail of the trailing edge zone**

## REDESIGN OF THE 'VKI-LS82' TURBINE NOZZLE

The first example illustrates the redesign of the VKI-LS82 turbine nozzle (Sieverding, 1982). The turning angle is approximately 79 degrees, with a pitch-to-chord ratio of 0.78. The numerical domain is discretized with 4 blocks (fig. 4,5). A hyperbolic O-grid has first been generated around the profile. An algebraic H-grid extends downstream, to discretize the wake. An elliptic C-grid has been constructed around the O- and H-blocks, with an additional H-grid upstream, in order to increase the upstream distance of the inlet boundary. The total number of nodes is ±16500.

The flow around this blade has been calculated, for an outlet isentropic Mach number of 0.85, a wall temperature of 300 K, and a Reynolds number (based on the total upstream conditions) of 2.025E+06. The measurements have shown

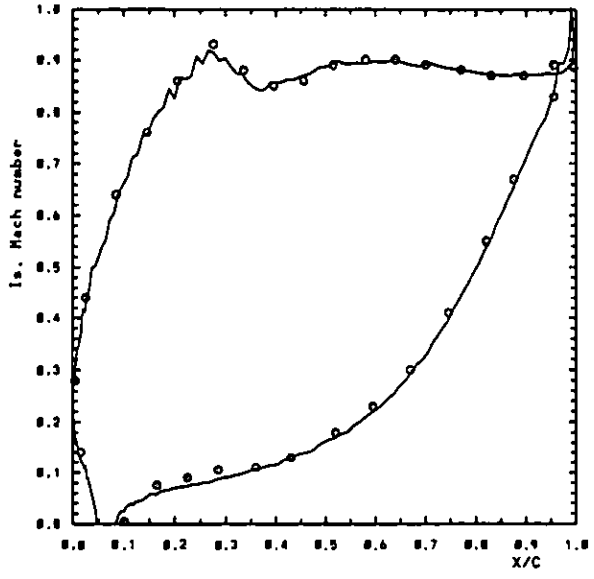


**Fig. 6: Calculated (–) and measured (o) isentropic Mach number distributions on the blades**

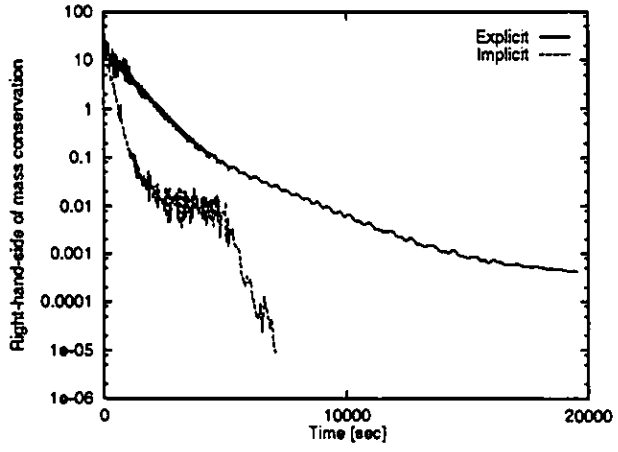


**Fig. 7: Implicit and explicit time integration schemes**

that the transition is expected at the rear part of the blade. No turbulence model was used in our computations. The isentropic Mach number distribution on the blades shows a good agreement with the measurements (fig. 6). The figure 7 compares the convergence histories corresponding to explicit and implicit calculations. Implicit time integration allows for drastic savings of computational time, requiring ±2 hours of CPU time on an 45 MFlops workstation. This gain is even more important for Euler computations, which is explained by the fact that the approximate jacobian of the preconditioner is closer to the true inviscid one.

A target distribution has been constructed, represented by the circles in figure 8, with a much smoother acceleration along the whole suction side. The inverse method has permitted to reach a perfect convergence to the target in 2000 explicit time iterations. The initial and final hlade shapes are compared in figure 9. The curvature of the new blade is clearly smoother in the nose region. The trailing edge is a bit thicker, which could eventually affect the trailing edge losses. This design required ±4 hours of CPU time (2 times as much as the time required for an analysis).

During the inverse calculation, the outlet isentropic Mach number has been iteratively and automatically modified (final value: 0.8367), so that the mass flow through the cascade remains constant. All inverse methods have to face the problem of existence of a solution. Lighthill (1945) showed that the target velocity distribution must be compatible with the inlet and outlet conditions. This explains that a solution is often more easily obtained with an imposed constant mass flow than with a constant outlet pressure. In the present case, the calculations with constant outlet pressure converge during some 500 iterations toward a similar pressure distribution, but with a higher average velocity level. The calculation then starts diverging, because the initial velocity level

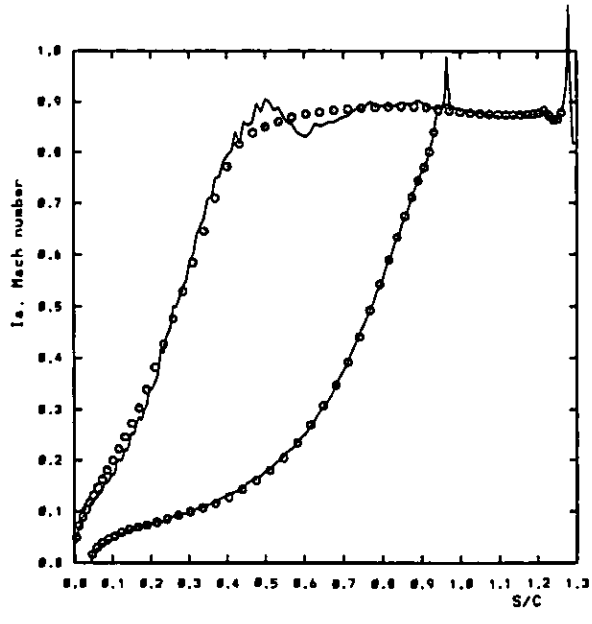


**Fig. 8: Initial (–) and imposed (o) isentropic Mach number distributions on the blades**

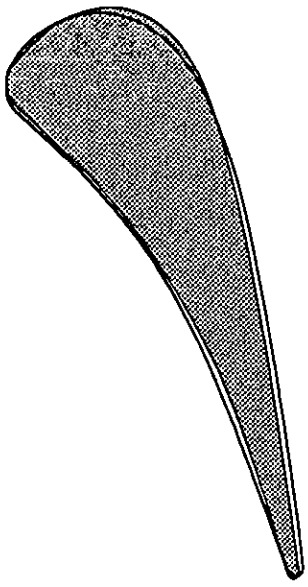

Fig. 9: Initial (grey) and final blade shapes

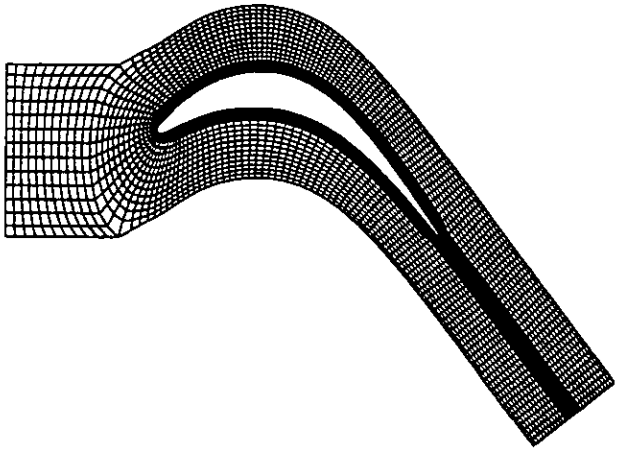

Fig. 10: General view of the multi-block mesh

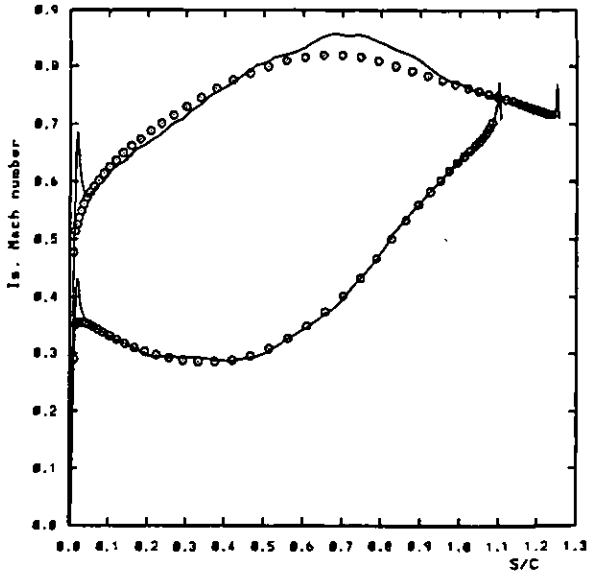


Fig. 11: Initial (–) and imposed (o) isentropic Mach number distributions on the blades

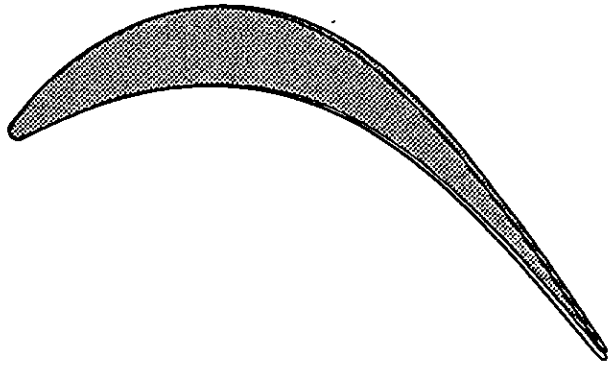

Fig. 12: Initial (grey) and final blade shapes

can not be recovered with the same pressure ratio accross the cascade.

## REDESIGN OF THE 'LA' SUBSONIC TURBINE BLADE

The second example concerns the redesign of the LA subsonic turbine blade, which has been tested at the Whittle Laboratory (AGARD-AR-275). The pitch-to-chord ratio is 0.56, the inlet flow angle is 38.8 degrees, and the turning angle is more or less 90 degrees. The multi-block grid is composed of 4 blocks, of the same type as the ones used for the LS82 blade (fig. 10).

The flow around this blade has been calculated, for an outlet isentropic Mach number of 0.71, and a Reynolds number (based on total upstream conditions) of $0.39E+06$. The experiments showed a separation bubble near 83% of the axial chord on the suction side, and a separation at about 15% of the pressure surface, with turbulent reattachment. In order to simulate this flow configuration, the Baldwin-Lomax model has been activated, respectively at 70% of the axial chord on the suction side and at 10% on the pressure side.

The isentropic Mach number distribution on the blades is compared to the target in figure 11. The main goals of this prescribed distribution are a reduction of the suction side maximum Mach number, and the suppression of the two peaks observed in the leading edge region, due to the discontinuity of curvature usually obtained when rounded leading edges are used.

The inverse method has permitted to reach a perfect

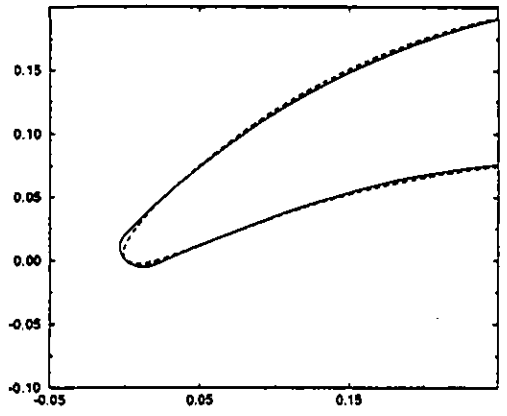


Fig. 13: Zoom of the leading edge - Initial (—) and final (- -) blade shapes

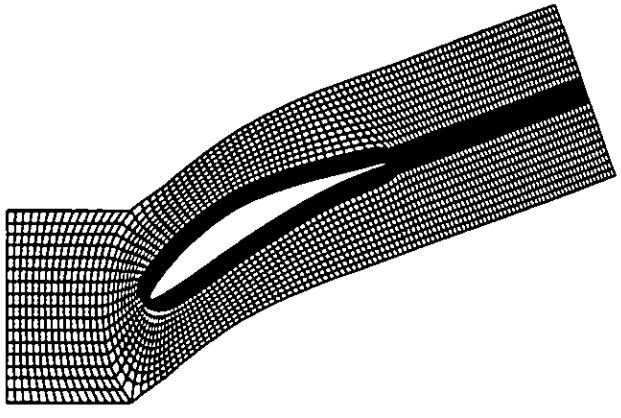

Fig. 14: General view of the multi-block mesh

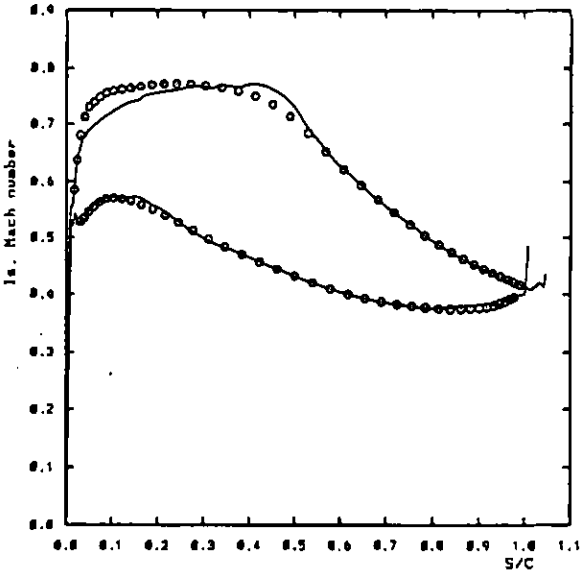


Fig. 15: Initial (–) and imposed (o) isentropic Mach number distributions on the blades

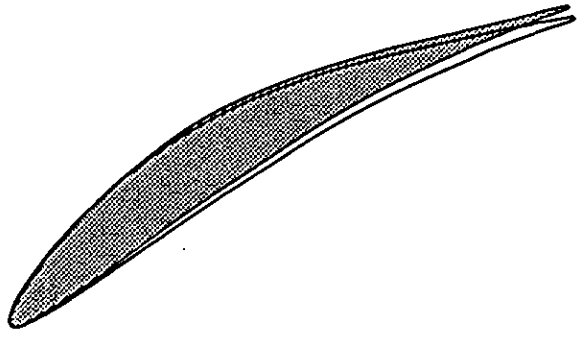

Fig. 16: Initial (grey) and final blade shapes

agreement between the final pressure distribution and the target. The initial and final geometries are compared in figure 12. The pressure distribution is perfectly smooth around the nose of the final blade. A detailed view of the leading edge of the two blades shnws that the new leading edge is rather thinner and closer to an ellipse than to a circle (fig. 13).

## REDESIGN OF THE 'DLR' COMPRESSOR BLADE

The last example concerns the redesign of a controlled diffusion compressor hlade proposed hy the DLR. The pitch-to-chord ratio is 0.68, the inlet flow angle is 47 degrees, and the turning angle is more or less 26 degrees. The multi-block grid is composed of 4 hlocks (fig. 14), of the same type as in the previous examples.

The flow around this blade has been calculated, for an outlet isentropic Mach number of 0.4, and a Reynolds number (based on the inlet total conditions) of $1.0E+06$. The Baldwin-Lomax turbulence model has been used for this calculation. The isentropic Mach number distribution on the blades is compared to the target in figure 15. The goal of this target is to increase the suction side acceleration, and hence the loading in the front part of the blade, without increasing the total amount of diffusion along the suction side.

The inverse method has permitted to reach a perfect agreement between the final pressure distribution and the target. The initial and final geometries are compared in figure 16. The turning angle has been increased from 26 to 27 degrees. A constant mass flow has been imposed during the inverse calculations. The final outlet Mach number is only of 0.5% lower than the initial one: in this case, inverse calculations with constant mass flow and constant outlet pressure give the same results.

## CONCLUSIONS

An inverse method has been developed, based on flow computations with moving boundaries. The main advantage of this method is that it can be applied to both viscous and inviscid problems.

The method is based on an efficient Euler/Navier-Stokes flow solver whose accuracy is demonstrated by the LS82 example.

The Navier-Stokes inverse method modifies the blade shape so that the prescribed pressure distribution is reached on the blade surface. A very precise tailoring of the velocity distribution can be achieved with the method.

The inverse Navier-Stokes solver is of course much more CPU time consuming than the Euler inverse solver. We believe that the two approaches are complementary. The major blade shape and velocity modifications should be undertaken with the inverse Euler method. The inverse Navier-Stokes solver should be used for a final tailoring of the velocity distribution. The viscous method permits to eliminate the uncertainties resulting from an inviscid design, i.e. the boundary layer subtraction, and provides a realistic treatment of the trailing edges during the design phase.

## REFERENCES


Bauer F., Garabedian, P., and Korn, D., 1972, "Supercritical Wing Sections", Vol. I, Springer-Verlag, New York.

Demeulenaere, A., and Van den Braembussche, R., 1996, "Three-Dimensional Inverse Method for Turhomachinery Blading Design", ASME Paper 96-GT-039.

Demeulenaere, A., Léonard, O. and Van den Braembussche, R., 1997, "A Navier-Stokes Inverse Solver for Compressor and Turbine Blade Design", submitted for publication at the 2nd International Conference on Turbomachinery- Fluid Dynamics and Thermodynamics, Antwerp.

Demirdzic, I., and Peric, M., 1988, "Space Conservation Law in Finite Volume Calculations of Fluid Flow", International Journal for Numerical Methods in Fluids, Vol. 8, 1037-1050.

Demirdzic, I., and Peric, M., 1990, "Finite Volume Method for Prediction of Fluid Flow in Arbitrarily Shaped Domains with Moving Boundaries", International Journal for Numerical Methods in Fluids, Vol. 10, 771-790.

Dulikravitch, D.S. and Sobieczky, H., 1982, "A Computational Design Method for Transonic Turbomachinery Cascades", ASME Paper 82-GT-117.

Eyi, S., Lee, K.D., Rogers, S.E., and Kwak, D., 1995, "High-Lift Design Optimization Using the Navier-Stokes Equations", AIAA 95-0477.

Giles, M., and Drela, M., 1987, "Two-Dimensional Transonic Aerodynamic Design Method", AIAA Journal, Vol 25, No 9, 1199-1206.

Jameson, A., 1995, "The Present Status, Challenges, and Future Developments in Computational Fluid Dynamics", AGARD 77th Fluid Dynamics Panel Symposium, Seville.

Jennions,I.K., 1994, "Elements of a Modern Turbomachinery Design System",AGARD-LS-195.

Léonard, O., 1990, "Subsonic and Transonic Cascade Design", AGARD R-780, "Special Course on Inverse Methods for Airfoil Design for Aeronautical and Turbomachinery Applications", pp 7.1-7.18.

Léonard, O. and Van den Braembussche, R., 1992, "Design Method for Subsonic and Transonic Cascade with Prescribed Mach Number Distribution", Transactions of the ASME, Vol. 114, No. 3, pp 553-560.

Léonard, O., Rogiest, P., and Delanaye, M., 1997, "Blade Analysis and Design Using an Implicit Flow Solver", submitted for publication at the 2nd International Conference on Turbomachinery- Fluid Dynamics and Thermodynamics, Antwerp.

Lighthill, J.M., 1945, "A New Method of Two-Dimensional Aerodynamic Design", ARC R&M 2112.

Meauzé, G., 1982, "An Inverse Time Marching Method for the Definition of Cascade Geometry", Journal of Engineering for Power (ASME), Vol. 104, pp 650-656.

Roe, P.L., 1981, "Approximate Riemann Solvers, Parameter Vectors and Difference Schemes", Journal of Computational Physics, 43:357-372.

Sanz, J.M., 1988, "Automated Design of Controlled Diffusion Blades", ASME Paper 88-GT-139.

Saad, Y., 1994, "Krylov Subspace Techniques, Conjugate Gradients, Preconditioning and Sparse Matrix Solvers", VKI Lecture Series 1994-5.

Schmidt, E., 1980, "Computation of Supercritical Compressor and Turbine Cascades with a Design Method for Transonic Flows", Journal of Engineering and Power (ASME), Vol. 102, pp 68-74.

Sieverding, C.H.,1982, "Workshop on Two-Dimensional and Three-Dimensional Flow Calculations in Turbine Bladings", VKI-LS 82-07, Vol.3.

Singh, K.P., Newman, J.C., and Baysal, O., 1995, "Dynamic Unstructured Method for Flows Past Multiple Objects in Relative Motion", AIAA Journal, Vol. 33, No 4, pp 641-649.

Stanitz, J.D., 1953, "Design of Two-Dimensional Channels with Prescribed Velocity Distributions along the Channel Walls", NACA Report 1115.

Trépanier, J.Y., Reggio, M., Zhang, H., and Camarero, R., 1991, "A Finite Volume Method for the Euler Equations on Arbitrary Lagrangian-Eulerian Grids", Computer Fluids, Vol. 20, No 4, pp 399-409.

van Leer, B., 1979, "Towards the Ultimate Conservative Difference Scheme, V: A Second Order Sequel to Godunov's Method", Journal of Computational Physics, 32:101-136.

van Leer, B., 1982, "Flux Vector Splitting for the Euler Equations", ICASE, Report No 82-30.

Vuillez, C., and Petot, B., 1994, "New Methods, New Methodology, Advanced CFD in the SNECMA Turbomachinery Design Process", AGARD-LS-195.

Zannetti, L., 1987, "Time Dependent Computation of the Euler Equations for Designing Fully 3D Turbomachinery Blade Rows", AIAA-87-0007.