NASA Technical Memorandum 101412

Application of a Lower-Upper Implicit Scheme and an Interactive Grid Generation for Turbomachinery Flow Field Simulations

Yung K. Choo
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

Woo-Yung Soh
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio

Seokkwan Yoon
MCAT Institute
NASA Ames Research Center
Moffett Field, California

Prepared for the
34th International Gas Turbine and Aeroengine Congress and Exposition
sponsored by the American Society of Mechanical Engineers
Toronto, Canada, June 4–8, 1989

NASA

(NASA-TM-101412) APPLICATION OF A LOWER-UPPER IMPLICIT SCHEME AND AN INTERACTIVE GRID GENERATION FOR TURBOMACHINERY FLOW FIELD SIMULATIONS (NASA) 17 p CSCL 01A G3/02

N89-15077

Unclas
0179861

APPLICATION OF A LOWER-UPPER IMPLICIT SCHEME AND AN INTERACTIVE GRID GENERATION FOR TURBOMACHINERY FLOW FIELD SIMULATIONS

Yung K. Choo
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

Woo-Yung Soh
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio 44135

and

Seokkwan Yoon
MCAT Institute
NASA Ames Research Center
Moffett Field, California 94035

ABSTRACT

A finite-volume lower-upper (LU) implicit scheme is used to simulate an inviscid flow in a turbine cascade. This approximate factorization scheme requires only the inversion of sparse lower and upper triangular matrices, which can be done efficiently without extensive storage. As an implicit scheme it allows a large time step to reach the steady state. An interactive grid generation program (TURBO), which is being developed, is used to generate grids. This program uses the control point form of algebraic grid generation which uses a sparse collection of control points from which the shape and position of coordinate curves can be adjusted. A distinct advantage of TURBO compared with other grid generation programs is that it allows the easy change of local mesh structure without affecting the grid outside the domain of dependence. Sample grids are generated by TURBO for a compressor rotor blade and a turbine cascade. The turbine cascade flow is simulated by using the LU implicit scheme on the grid generated by TURBO.

NOMENCLATURE

A,B	Jacobian matrices
C_{ij}	a control point
D	difference operator
$E_j(r)$	control point curve
e	total energy/volume
F	flux vector in the x-direction
$F_i(t)$	control point curve
f	flux vector in ξ-direction
G	flux vector in y-direction
G_α	integrated interpolation function
g	flux vector in η-direction
H_β	integrated interpolation function
J	Jacobian of coordinate transformation
M	number of control points in ξ-direction
N	number of control points in η-direction
$P(r,t)$	position vector
p	static pressure
p_t	total pressure
$Q(r,t)$	Boolean sum transformation
R	residual
r	curve parameterization
r_A, r_B	spectral radii
$T(r,t)$	tensor product transformation
T_r	transpose of matrix
t	curve parametrization
U,V	contravariant velocities
$u,v,$	Cartesian velocities in x- and y-directions
W	vector of flow variables
x,y	Cartesian coordinates
α_i	on-off boundary switch
β	numerical parameter between 0 and 1
γ	ratio of specific heat

ξ,η	transformed coordinate system
λ	eigenvalues of Jacobian matrices
ρ	density
τ	time
ψ	interpolation function

INTRODUCTION

An unconditionally stable implicit scheme that has error terms, at most, of the order $(\Delta\tau)^2$ in any number of space dimensions can be derived by lower-upper (LU) factorization (Jameson and Turkel, 1981). Splitting ensures the diagonal dominance of lower and upper factors and makes use of the built-in implicit dissipation. The LU implicit scheme requires only the inversion of sparse triangular matrices, which can be done efficiently without using extensive storage. As an implicit scheme it allows a large time step to reach the steady state. This approximate factorization scheme was demonstrated to be robust and efficient in a broad flow regime; for example, Jameson and Yoon (1987) for transonic flows and Yoon and Jameson (1986) for a high speed inlet flow. These previous efforts focused on improving efficiency of the LU implicit scheme by using multiple grids with simple isolated airfoils and inlet geometries. In the work presented herein the LU scheme is applied to obtain the solution of the inviscid compressible flow in a turbine cascade. This is a first step towards simulation of flows in turbomachinery that have more complex phenomena. One such example could be a flow in a supersonic through-flow fan (Schmidt et al., 1987). Eventually, for viscous calculations the meshes will need to be very fine to resolve the boundary layer. And it is likely that the time step imposed by an explicit stability bound will be much less than that imposed by the accuracy bound of an implicit scheme. Since an obvious way to accelerate convergence to a steady state is to increase the size of the time step, an implicit scheme is expected to have a faster convergence. Although the alternating direction implicit (ADI) scheme has been valuable in two-dimensional problems (Beam and Warming, 1978), its inherent limitations in three dimensions suggest an alternative approach.

Flow passages in turbomachinery are highly constrained geometric regions. Fine meshes are needed to capture flow phenomena where flow properties change rapidly in several subregions. The orthogonality desired in highly cambered blades conflicts with the periodicity constraints that are required by many flow solvers; the need to cluster the grid conflicts with the smoothness desired. Generating grids for turbomachinery, therefore, requires a sensible compromise between geometric constraints and desired grid structure. One of the best approaches would be to generate the grid interactively by using a method that allows easy local control over the grid distribution (Choo et al., 1988). The control point form of algebraic grid generation was formulated by Eiseman (1987). From the control points sparsely distributed over the flow domain, the shape and position of coordinate curves can be adjusted while the grid conforms precisely to all boundaries. As an algebraic method, the control point form provides explicit control of grid structure and requires relatively few computations. The basic elements of this method are the multisurface transformation for control in a given direction (Eiseman, 1982) and the Boolean sum operation for the combination directions. Eiseman (1987) demonstrated how mesh structure can be changed with control points by using simple duct-like geometries. The control point formulation is used herein to generate grids for turbomachinery geometry.

A menu-driven interactive program called TURBO is being developed to generate well-structured grids for turbomachinery flow field simulations. By using the control point form mentioned previously, TURBO can achieve a balance between the constraints of the turbomachinery geometry and the desired grid structure. A distinct advantage of TURBO compared with other grid-generation programs is that it allows the easy change of local mesh structure without affecting the grid outside the domain of dependence. The boundary surface can be either a fixed form (transfinite) or a free form, which can be changed. Once the grid is improved, it can be closely examined by zooming, translating, and rotating. Further improvement to the grid structure can be made with this closer view of the grid. The program, TURBO, is being built around the control net which guides the coordinate curves. By choosing menus, TURBO now normalizes the grid near the vane or blade surface, provides slope continuity across the periodic boundary in C-type grids, stretches grids by stretching the control points in real time with the movement of the workstation mouse, and simultaneously translates multiple control points on a line in a manner similar to points on a rubber band. These are some of the features that provide convenience to users.

The finite-volume LU implicit scheme was used to simulate an inviscid flow in a turbine cascade on a C-grid generated by TURBO. The grid generated for this flow is normal to the vane surface and has slope continuity across the periodic boundary. For this cambered blade, the rubberbanding feature was used to reduce the grid skewness on the suction side of the vane. Computed results are compared with the experimental data obtained by Goldman and Seasholz (1982).

For clarity and completeness of this paper, governing equations, the LU approximate factorization scheme, and the control point formulation are briefly presented in the following sections, even though detailed derivations are available in the references cited above.

GOVERNING EQUATIONS

The conservative form of the Euler equations in Cartesian coordinates for two space dimensions is

$$\frac{\partial \mathbf{W}}{\partial \tau} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{1}$$

where $\mathbf{W}$ is the vector of dependent variables and $\mathbf{F}$ and $\mathbf{G}$ are convective flux vectors as follows:

$$\mathbf{W} = (\rho,\ \rho u,\ \rho v,\ e)^{T_r} \tag{2a}$$

$$\mathbf{F} = \left[\rho u,\ \rho u^2 + p,\ \rho uv,\ u(e + p)\right]^{T_r} \tag{2b}$$

$$\mathbf{G} = \left[\rho v,\ \rho uv,\ \rho v^2 + p,\ v(e + p)\right]^{T_r} \tag{2c}$$

where ρ, u, v, e, and p are density, Cartesian velocity components in the x- and y-directions, total energy, and pressure respectively. The total energy e is the sum of the internal energy and the kinetic energy per

unit volume of fluid. The system of Eqs. (1) and (2) is solved with time τ for the primary unknowns, (ρ, ρu, ρv, and e), and the pressure is obtained subsequently by the equation of state as,

$$p = (\gamma - 1)\left[e - \rho \frac{(u^2 + v^2)}{2}\right] \tag{3}$$

where γ is the ratio of specific heats. Because of the constrained flow passage geometry of turbomachinery, a coordinate transformation, is performed in such a way that the transformed coordinate coincides with the flow boundary. With the coordinate transformation, $\xi = \xi(x,y)$ and $\eta = \eta(x,y)$, the vector form of the Cartesian components of Euler equations is

$$\frac{\partial W}{\partial \tau} + J\left(\frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta}\right) = 0 \tag{4}$$

where $f = (\xi_x F + \xi_y G)/J$, $g = (\eta_x F + \eta_y G)/J$, and $J = \xi_x \eta_y - \xi_y \eta_x$. With the relations $\xi_x = J y_\eta$, $\eta_x = -J y_\xi$, $\xi_y = -J x_\eta$, $\eta_y = J x_\xi$ and the use of contravariant velocities, $U = y_\eta u - x_\eta v$ and $V = -y_\xi u + x_\xi v$, vectors f and g in Eq. (4) are written as

$$f = \begin{pmatrix} \rho U \\ \rho u U + y_\eta p \\ \rho v U - x_\eta p \\ (e + p)U \end{pmatrix} \qquad g = \begin{pmatrix} \rho V \\ \rho u V - y_\xi p \\ \rho v V + x_\xi p \\ (e + p)V \end{pmatrix} \tag{5}$$

LOWER-UPPER IMPLICIT SCHEME

To advance the Euler equations with time we discretize Eq. (4) in time as

$$\frac{W^{n+1} - W^n}{\Delta \tau} + \beta J\left(\frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta}\right)^{n+1} + (1 - \beta) J\left(\frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta}\right)^{n} = 0 \tag{6}$$

where β varies between 0 and 1. For $\beta = 0$, Eq. (6) is a purely explicit scheme, and for $\beta = 1/2$, a second order accuracy holds in time. Nonlinear terms at time level $n + 1$ are linearized by the following approximations:

$$\left(\frac{\partial f}{\partial \xi}\right)^{n+1} = \left(\frac{\partial f}{\partial \xi}\right)^{n} + A^n \delta W, \qquad \left(\frac{\partial g}{\partial \eta}\right)^{n+1} = \left(\frac{\partial g}{\partial \eta}\right)^{n} + B^n \delta W$$

where $\delta W = W^{n+1} - W^n$, and A and B are Jacobian matrices defined to be $J^{-1}\,\partial f/\partial W$, and $J^{-1}\,\partial g/\partial W$, respectively. The truncation error of the above expression is of $O(\Delta \tau^2)$. The linearized implicit scheme for Eq. (6) can be formulated as

$$\left[I + \beta J\,\Delta\tau\left(D_\xi A + D_\eta B\right)\right]\delta W + \Delta \tau\, R = 0 \tag{7}$$

where I is the identity matrix and R is the steady state part of Eq. (4) given as

$$R = J\left(\frac{\partial f}{\partial \xi} + \frac{\partial g}{\partial \eta}\right)^{n} = D_\xi f^n + D_\eta g^n$$

Here, D_ξ and D_η are central difference operators that approximate $\partial/\partial\xi$ and $\partial/\partial\eta$. An LU factorization scheme is introduced to approximate Eq. (7) as

$$\left[I + \beta J\,\Delta\tau\left(D_\xi^- A^+ + D_\eta^- B^+\right)\right]\left[I + \beta J\,\Delta\tau\left(D_\xi^+ A^- + D_\eta^+ B^-\right)\right]\delta W + \Delta\tau\, R = 0 \tag{8}$$

where D_ξ^- and D_η^- are backward difference operators and D_ξ^+ and D_η^+ are forward difference operators. Here, A^+, A^-, B^+, and B^- are constructed so that the eigenvalues of "+" matrices are nonnegative and those of "-" matrices are nonpositive, that is

$$A^+ = \frac{1}{2}\left(A + r_A I\right), \qquad A^- = \frac{1}{2}\left(A - r_A I\right)$$

$$B^+ = \frac{1}{2}\left(B + r_B I\right), \qquad B^- = \frac{1}{2}\left(B - r_B I\right)$$

where

$$r_A \geq \max\left(|\lambda_A|\right), \qquad r_B \geq \max\left(|\lambda_B|\right)$$

Here λ_A and λ_B represent eigenvalues of Jacobian matrices. Equation (8) is inverted in two consecutive steps. The LU implicit scheme requires the inversion of sparse triangular matrices, which can be done efficiently without using extensive storage. This scheme has only two factors in three space dimensions.

To make the central difference scheme (Eq. (8)) stable, an artificial dissipation term in the fourth order difference form is added explicitly to R in Eq. (8) for the entire flow region. For transonic or supersonic calculation, additional dissipation needs to be added implicitly in the neighborhood of shocks (Jameson, 1987). In the present subsonic case only the fourth order explicit dissipation is employed.

The cascade flow measured by Goldman and Seasholz (1982) is fully subsonic. To simulate this flow, the far-field boundary conditions used for an isolated airfoil by Jameson and Yoon (1987) were replaced by in-and-out flow boundary conditions and periodic boundary on the sides. At the inlet boundary, three out of four flow variables were specified and the remaining one was obtained as a solution of the flow. At the exit boundary, only one flow variable was specified. In the present work, the density ρ was allowed to change while other variables were fixed at the inlet boundary. At the exit, only the static pressure was specified. Periodic boundary condition is enforced on the sides because of the cascade configuration of the present problem.

CONTROL POINT FORMULATION

The control point array is a sparse grid-type arrangement of locations in physical space with an index for each direction. In two dimensions it will be denoted by (C_{ij}). As an algebraic method, the control point form provides explicit control of the physical grid shape and spacing through the dynamic movement of the control points. Figure 1 shows an example of a two-dimensional control point array.

A fundamental part of the control point formulation is the construction of curves. This construction represents algebraic coordinate generation in a single direction wherein two opposing boundaries are connected by the newly created curves (e.g., the curve $E_2(r)$ in Fig. 1). With the restriction to only a single curve, the opposing boundaries are each represented by a point. The first and last points lie on opposing boundaries and are the fixed end points of the curve. The remaining points are in the interior of the sequence and are used to control the shape of the curve. To enforce the successive assumptions of the desired direction in a smooth manner, a continuous direction field is obtained by interpolation. The independent variable for the interpolation is simply the curve parameterization. Altogether, the interpolated result defines the field of vectors that are tangent to the desired curve and is simply stated as an interpolation of the first parametric derivatives. Thus a smooth first derivative of the entire curve is determined. The desired curve is then obtained by a parametric integration.

Let $\mathbf{C}_{1j}, \mathbf{C}_{2j}, \ldots, \mathbf{C}_{Mj}$, be the given sequence of M points in space (e.g., $\mathbf{C}_{12}, \mathbf{C}_{22}, \ldots, \mathbf{C}_{52}$ for $j = 2$ in Fig. 1); let r be the curve parameterization; let $\mathbf{E}_j(r)$ be the position at r along the desired curve; let $r_1, r_2, \ldots, r_{M-1}$ be the successive parametric locations to interpolate the directions of $(\mathbf{C}_{2j} - \mathbf{C}_{1j})$, $(\mathbf{C}_{3j} - \mathbf{C}_{2j}), \ldots, (\mathbf{C}_{Mj} - \mathbf{C}_{M-1,j})$; and let $\Psi_1, \Psi_2, \ldots, \Psi_{M-1}$ be the corresponding interpolation functions which successively separate each direction by assuming a nonzero value at the associated location while vanishing at the remaining locations for interpolation. With this notation the desired curve is given by

$$\mathbf{E}_j(r) = \mathbf{C}_{1j} + \sum_{\alpha=1}^{N-1} G_\alpha(r)\left[\mathbf{C}_{\alpha+1,j} - \mathbf{C}_{\alpha j}\right] \qquad j = 1, 2, \ldots, M \qquad (9)$$

where

$$G_\alpha(r) = \int_{r_1}^{r} \Psi_\alpha(\mu)\,d\mu \qquad (10)$$

To apply the method, the interpolation functions must be chosen. With local functions, the alteration of a control point results in an alteration of the constructed curve that is restricted to a local region about the point. The remaining regions are unaltered. Therefore local sections can be manipulated in an independent manner. The simplest local interpolants are the piecewise linear functions that do not vanish over, at most, two intervals defined by $r_1 < r_2 < \ldots < r_{M-1}$. To further simplify the form of the multisurface transformation, the height of each interpolant shall be adjusted so that each interpolant integrates to unity. Then, the evaluation of the transformation at r_{M-1} will reduce to $\mathbf{C}_{Mj}$ by means of a telescopic collapse of terms. An explicit form of the normalized interpolation functions (Eiseman and Smith, 1980) is given for a case with the uniform partition ($r_k = k$, $k = 1,2, \ldots, N-1$)

$$\Psi_1(r) = \begin{cases} 2(2 - r) & \text{for } 1 \le r < 2 \\ 0 & \text{for } 2 \le r \le N-1 \end{cases} \qquad (11)$$

$$\Psi_k(r) = \begin{cases} 0 & \text{for } 1 \le r < k-1 \\ (r - k) + 1 & \text{for } k-1 \le r < k \\ (k - r) + 1 & \text{for } k \le r < k+1 \\ 0 & \text{for } k+1 \le r \le N-1 \end{cases} \qquad (12)$$

$$\Psi_{N-1}(r) = \begin{cases} 0 & \text{for } 1 \le r < N-2 \\ 2(r - N + 2) & \text{for } N-2 \le r \le N-1 \end{cases} \qquad (13)$$

Similarly, for the index value of i, a curve can be constructed

$$\mathbf{F}_i(t) = \mathbf{C}_{i1} + \sum_{\beta=1}^{M-1} H_\beta(t)\left[\mathbf{C}_{i,\beta+1} - \mathbf{C}_{i\beta}\right] \qquad i = 1, 2, \ldots, N \qquad (14)$$

where t is the curve parameterization and $H_\beta(t)$ is the integration of the multisurface interpolants. The tensor product form depends only upon $\mathbf{C}_{ij}$ and is given by

$$\mathbf{T}(r,t) = \mathbf{E}_1(r) + \sum_{\beta=1}^{M-1} H_\beta(t)\left[\mathbf{E}_{\beta+1}(r) - \mathbf{E}_\beta(r)\right] \qquad (15)$$

or alternatively by

$$\mathbf{T}(r,t) = \mathbf{F}_1(t) + \sum_{\alpha=1}^{N-1} G_\alpha(t)\left[\mathbf{F}_{\alpha+1}(t) - \mathbf{F}_\alpha(t)\right] \qquad (16)$$

These two expressions are equivalent. The tensor product matches $\mathbf{E}_j$ or $\mathbf{F}_i$ at the extremities of i and j.

When boundaries are to be specified, the corresponding data appear at the extremities of the values for r and t. Since the coordinate transformations are generally expressed in the form of a vector $\mathbf{P}(r,t)$ for the desired positions of all points in physical space, it is convenient to express the boundary specifications in terms of the position vector. Thus, the boundaries are denoted by $\mathbf{P}(1,t)$, $\mathbf{P}(N-1,t)$, $\mathbf{P}(r,1)$, and $\mathbf{P}(r,M-1)$. To include the boundaries, the multisurface transformation is performed again as above, but now with the actual boundaries inserted. This results in a modification of $\mathbf{T}$ for both the r and t directions. In each such directional construction, the actual boundaries appear as end conditions for the corresponding variable whereas the remaining boundaries are solely generated by the control points. Thus, by subtracting $\mathbf{T}$ from the sum of both directional constructions, the actual boundaries become end conditions for each variable. This process follows a Boolean sum format and upon simplification becomes

$$\begin{aligned} \mathbf{Q}(r,t) = \mathbf{T}(r,t) &+ \alpha_1\left[1 - G_1(r)\right]\left[\mathbf{P}(1,t) - \mathbf{F}_1(t)\right] \\ &+ \alpha_2\, G_{N-1}(r)\left[\mathbf{P}(N-1,t) - \mathbf{F}_N(t)\right] \\ &+ \alpha_3\left[1 - H_1(t)\right]\left[\mathbf{P}(r,1) - \mathbf{E}_1(r)\right] \\ &+ \alpha_4\, H_{M-1}(t)\left[\mathbf{P}(r,M-1) - \mathbf{E}_M(r)\right] \end{aligned} \qquad (17)$$

where each of the four terms following the tensor product $\mathbf{T}(r,t)$ represents a transfinite conformity to a boundary when each α_i is 1. By setting any α_i to 0, the corresponding boundary becomes available for free-form modeling by means of the control points. In the order listed, the boundaries are for $r = 1$, $r = N - 1$, $t = 1$, and $t = M - 1$. Further details are discussed by Eiseman (1987).

RESULTS

The basic interactive process of TURBO is illustrated with a simple geometry in Fig. 2. It starts with construction of a simple control net and surface grid. An initial grid is generated and examined. If desired, the grid structure can be improved through the dynamic movement of the control points. In this example a control point was moved to obtain dense meshes in the middle of the flow region.

Figure 3 shows an H-grid generated for a cambered turbine stator vane. Only those basic interactive features illustrated in Fig. 2 were used in this example. Figure 3(a) shows an initial control net; Fig. 3(b) shows an initial grid. By scrolling graphic images for one view (combination view of the control net and grid) to another (control net only or grid only) and by expanding the graphic image around a center of expansion, a user can closely examine the grid and determine what modification of the control net is needed to improve the grid structure. The modified control net and grid are shown in Fig. 3 parts (c) and (d), respectively. The modified grid has a densely clustered mesh around the leading edge and a more nearly orthogonal grid near the vane surfaces.

Many grid generation programs do not allow local mesh control as TURBO does. For the TURBO program the initial grid may be generated by any program familiar to a user; then an initial control net may be obtained from the grid by attachment to essentially reproduce the grid structure. The initial control net shown in Fig. 4(a) was created by attachment. Figure 4(b) shows an initial grid of a compressor rotor blade. Once the control net is created, the interactive process to be followed is the same as described previously; the initial control net is modified to the one shown in Fig. 4(c), and a new grid (Fig. 4(d)) is generated from the modified control. A closer view of the initial and improved grid is presented in Fig. 5. The improved grid is more othogonal and has slope continuity across the periodic boundary. The shape of the control net can easily be changed by using an interactive process illustrated in Fig. 6. In part (a) of the figure, a user first chooses a control line to be changed and then picks a hinge point. In part (b), control point 1 is moved to point 2 by moving the mouse of the workstation. Point 1' automatically moves to point 2' while maintaining a pitch with the points 1 and 2, respectively. In part (c), the desired shape of the control line is obtained by choosing a menu called "rubberbanding."

The finite-volume LU implicit scheme was used to simulate an inviscid flow in a turbine cascade on the grid shown in Fig. 7. Part (a) of the figure shows the control net from which the grid shown in part (b) was generated. Part (c) is a closer view of the grid. This grid is nearly orthogonal in the vicinity of the solid vane surface and has slope continuity across the periodic boundary. For this cambered vane, the skewness of this single block grid was reduced to some degree by using the "rubberbanding" feature and free-form periodic boundary. Detailed stator vane geometry, cascade flow conditions, laser anemometer survey measurements, and vane surface static pressure measurements are presented by Goldman and Seasholz (1982). To operate the cascade facility, ambient air from the test cell was drawn through the cascade and exhausted into the laboratory altitude exhaust system. The test conditions in the cascade were set by controlling the pressure ratio across the vane row with two throttle valves located in the exhaust system. A hub static tap located downstream from the test section, where the flow was assumed to be nearly circumferentially uniform, was used to set this pressure ratio. The hub-static- to inlet-total-pressure ratio was maintained at a value of 0.65. This corresponds to a static- to inlet-total-pressure ratio of 0.687 at a 50-percent span where a two-dimensional inviscid computation was made.

Core turbine stator vane geometry at the mean section is shown in Fig. 8(a). Also shown are three axial locations where measured data are compared with computed results. In Fig. 8(b), laser measurements are compared with computation at 10.2, 50, and 90.1 percent of the axial chord at the 50-percent span. The computed critical velocity ratios (V/V_{cr}) agree well with the laser measurements. Figure 9 shows a good agreement between the blade surface measurement and computation. For this comparison the vane surface static pressure measurements are used to obtain the critical velocity ratios by using the following relation:

$$\frac{V}{V_{cr}} = \sqrt{\frac{\gamma + 1}{\gamma - 1}\left[1 - \left(\frac{p}{p_t}\right)^{\gamma-1/\gamma}\right]} = \sqrt{6\left[1 - \left(\frac{p}{p_t}\right)^{2/7}\right]}$$

where p is static pressure and p_t is total pressure. A static pressure contour plot is presented in Fig. 10 without comparison. The pressure was normalized by using the inlet static pressure.

CONCLUDING REMARKS

The LU approximate factorization scheme was demonstrated to be robust and efficient in the transonic-flow and very high Mach-number-flow regimes. In these previous works, the main interest was in improving the efficiency of the scheme and therefore simple isolated airfoil and inlet geometries were used. In the work presented, the finite-volume LU implicit scheme was applied to obtain the solution of the inviscid flow in a turbine cascade on a grid generated by TURBO. This is a first step towards simulation of more complex flows in turbomachinery.

TURBO is a menu-driven interactive algebraic grid generation program. It is being built around the control net which guides the coordinate curves. From the control points sparsely distributed over the flow domain, TURBO can easily modify the grid structure of a local region (a domain of dependence) without affecting the grid outside of it. For the complex flows in turbomachinery, the control point formulation is a very useful method because of this precise local mesh control capability. With the advent of faster graphics workstations with more memory and better linking to mainframe supercomputers, interactive grid generation by means of the control point formulation will become a more effective and useful approach.

REFERENCES

Beam, R.M. and Warming, R.I., 1978, "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," AIAA Journal, Vol. 16, No. 4, pp. 393-402.

Choo, Y., Eiseman, P., Reno, C., 1988, "Interactive Grid Generation for Turbomachinery Flow Field Simulations," Presented at the Second International Conference on Numerical Grid Generation in Computational Fluid Dynamics, Miami Beach, FL. (NASA TM-101301.)

Eiseman, P.R. and Smith, R.E., 1980, "Mesh Generation Using Algebraic Techniques", Numerical Grid Generation Techniques, NASA CP-2166, pp. 73-120.

Eiseman, P.R., 1982, "Coordinate Generation with Precise Controls Over Mesh Properties," Journal of Computational Physics, Vol. 47, pp. 331-351.

Eiseman, P.R., 1987, "A Control Point Form of Algebraic Grid Generation," Numerical Methods in Laminar and Turbulent Flow, W.G. Habashi, M.M. Hafez, and C. Taylor, eds., Pineridge Press, pp. 1083-1091.

Goldman, L.J. and Seasholz, R.G., 1982, "Laser Anemometer Measurements in an Annular Cascade of Core Turbine Vanes and Comparison with Theory," NASA TP-2018.

Jameson, A. and Turkel, E., 1981, "Implicit Schemes and LU Decompositions," Mathematics of Computation, Vol. 37, No. 156, pp. 385-397.

Jameson, A., 1987, "Successes and Challenges in Computational Aerodynamics," AIAA Paper 87-1184.

Jameson, A. and Yoon, S., 1987, "Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations," AIAA Journal, Vol. 25, No. 7, pp. 929-935.

Schmidt, J.F., Moore, R.D., Wood, J.R., and Steinke, R.I., 1987, "Supersonic Through-Flow Fan Design," AIAA Paper 87-1746. (NASA TM-88908.)

Yoon, S. and Jameson, A., 1986, "An LU Implicit Scheme for High Speed Inlet Analysis," AIAA Paper 86-1520.

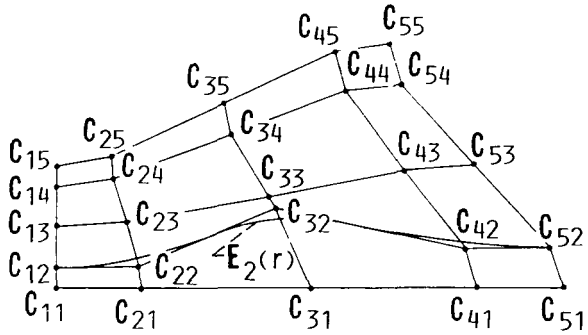


FIG. 1. - CONTROL NET - AN EXAMPLE IN TWO DIMENSIONS.

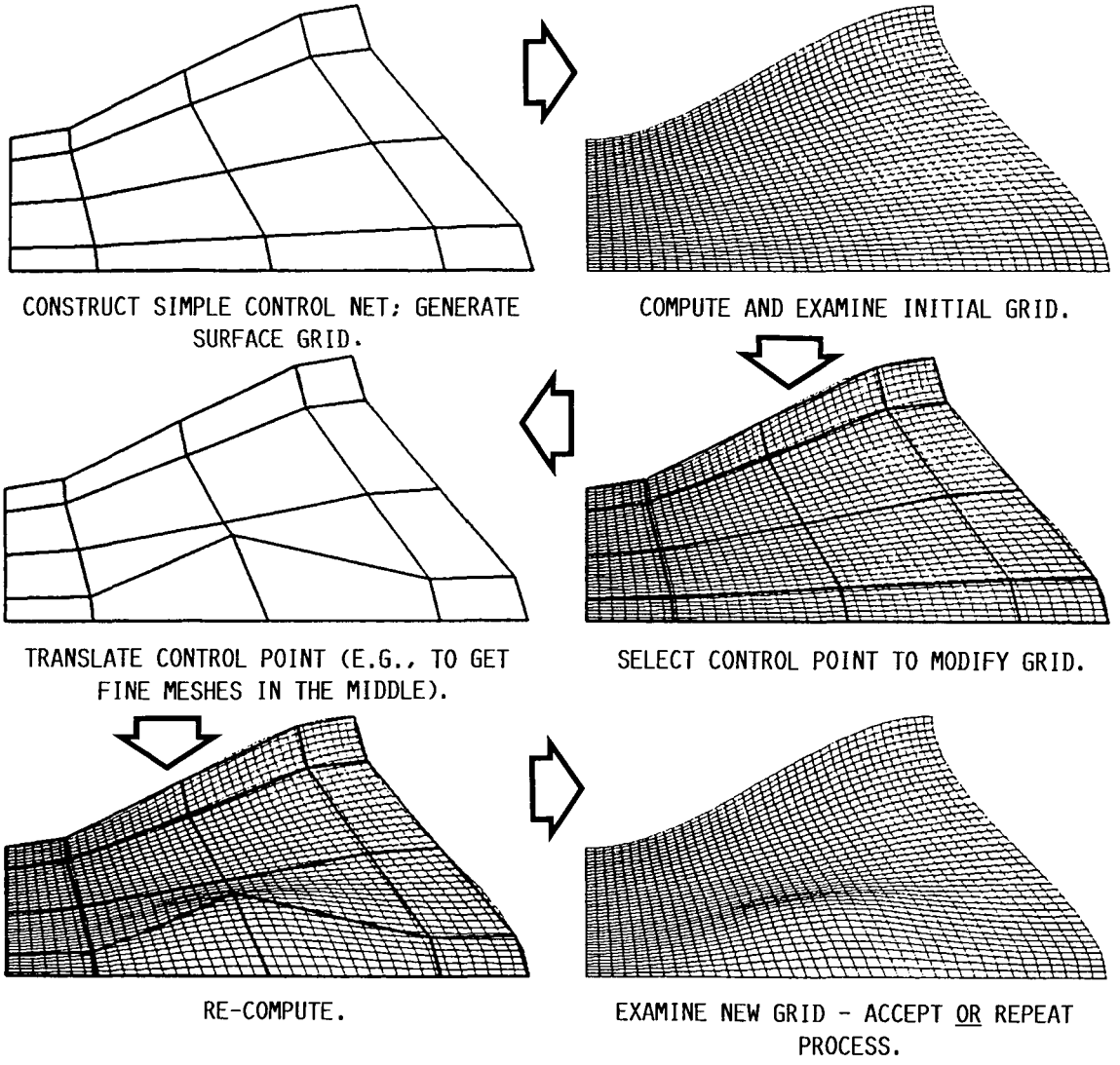
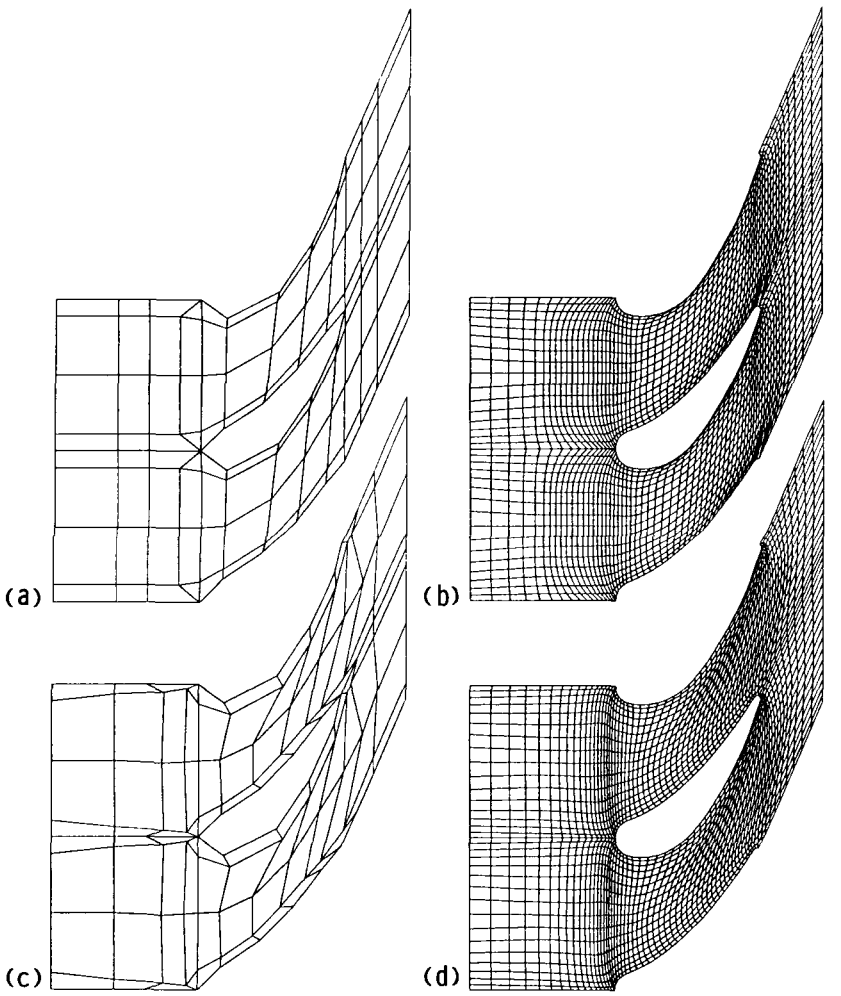
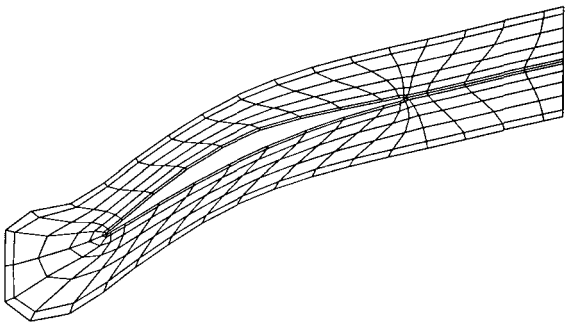


FIG. 2. - BASIC INTERACTIVE PROCESS.

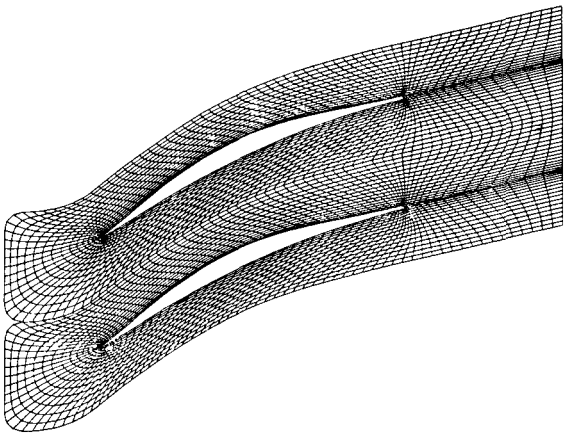


(a) INITIAL CONTROL NET. (b) INITIAL GRID.
(c) MODIFIED CONTROL NET. (d) MODIFIED GRID.

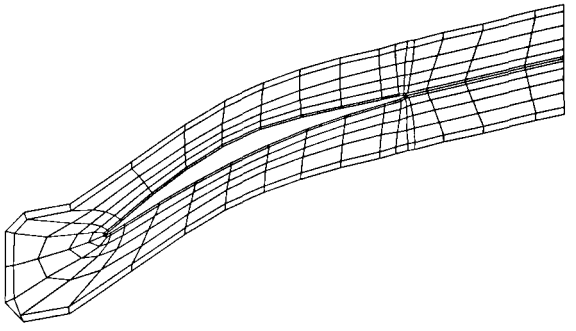
FIG. 3. - GRID GENERATION FOR A CAMBERED STATOR VANE.



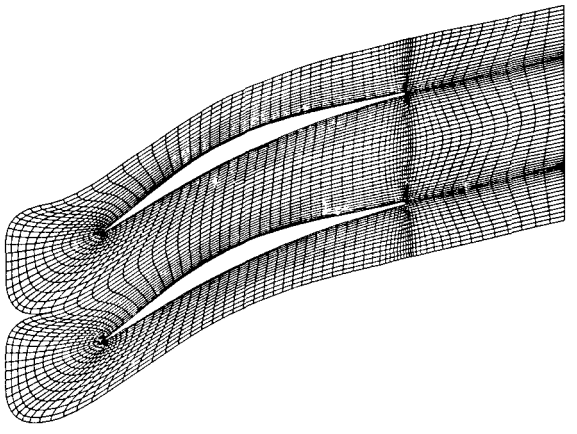
(a) INITIAL CONTROL NET.



(b) INITIAL GRID.

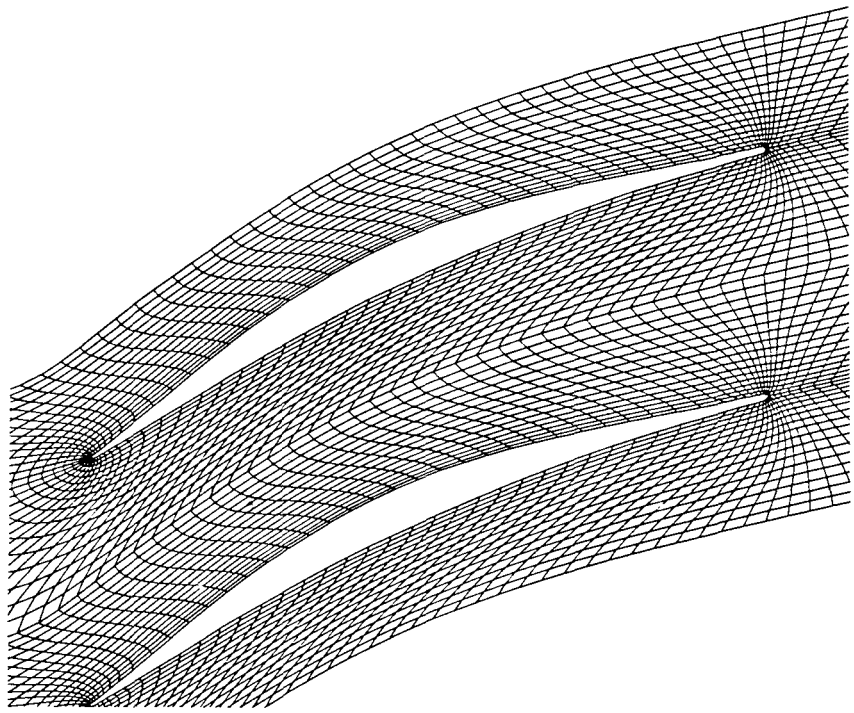


(c) MODIFIED CONTROL NET.

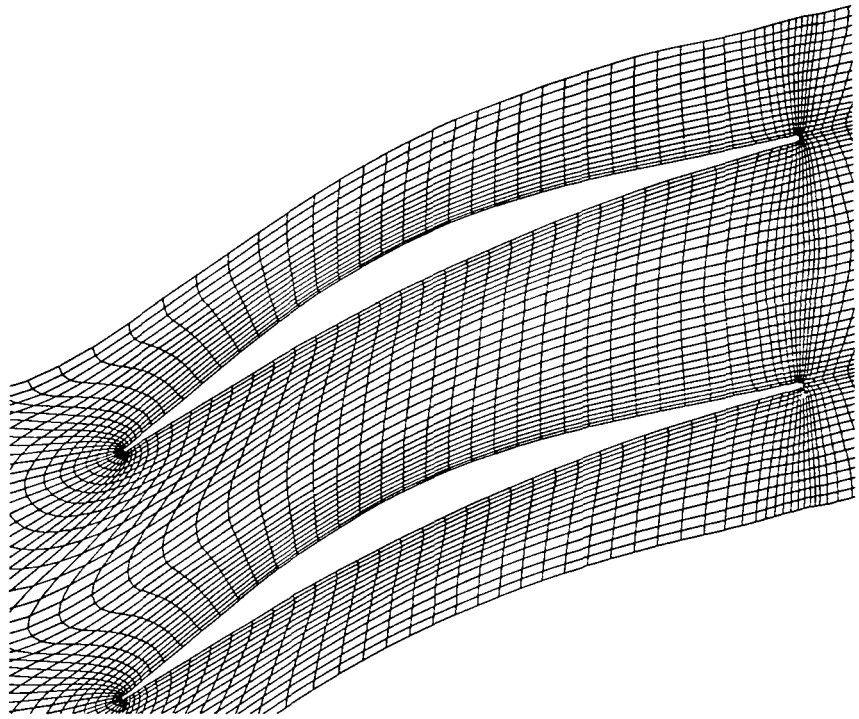


(d) MODIFIED GRID.

FIG. 4. - GRID GENERATION FOR A COMPRESSOR BLADE.

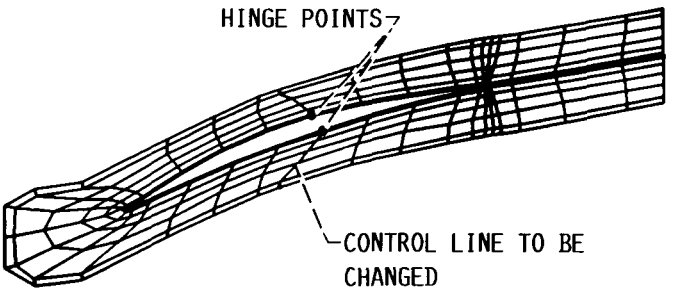


(a) INITIAL GRID.

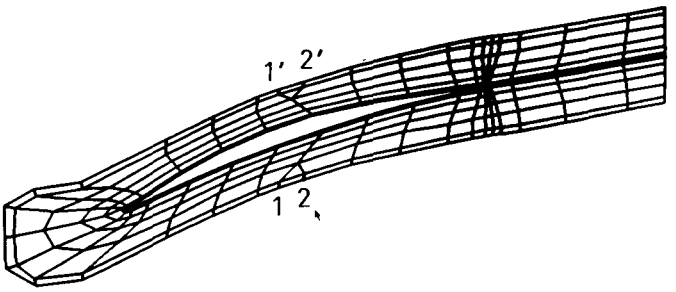


(b) MODIFIED GRID.

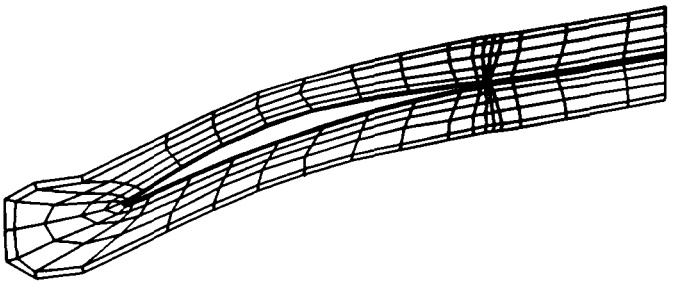
FIG. 5. - CLOSER VIEW OF THE INITIAL AND MODIFIED GRIDS.



(a) SELECT CONTROL LINES AND HINGE POINTS.

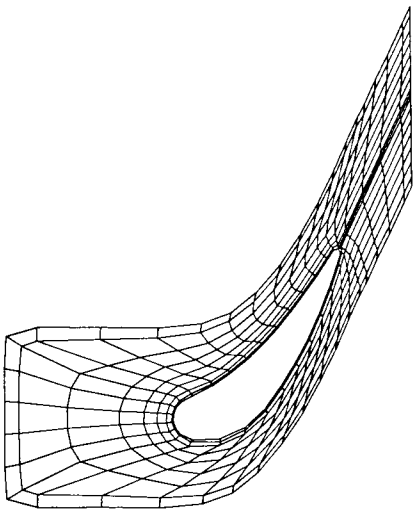


(b) TRANSLATE A CONTROL POINT FROM 1 TO 2.

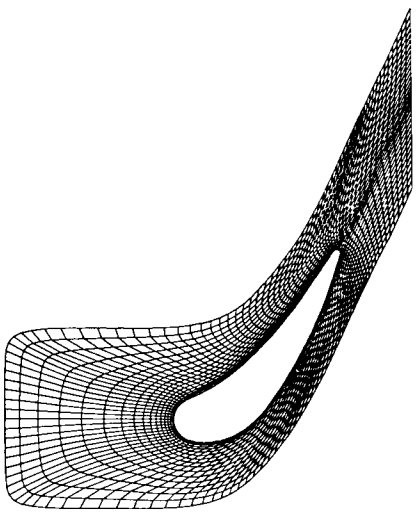


(c) USE "RUBBERBANDING" FEATURE.

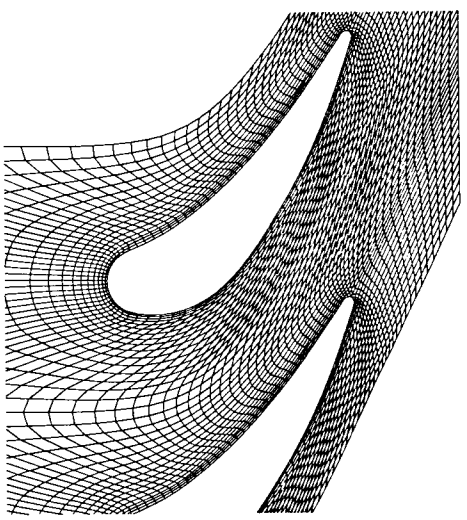
FIG. 6. - CONTROL LINE MODIFICATION BY MEANS OF AN INTERACTIVE FEATURE.



(a) CONTROL NET.

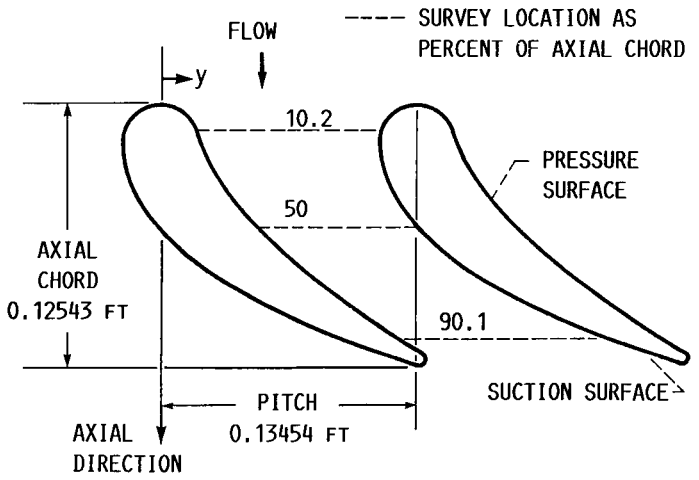


(b) GRID.

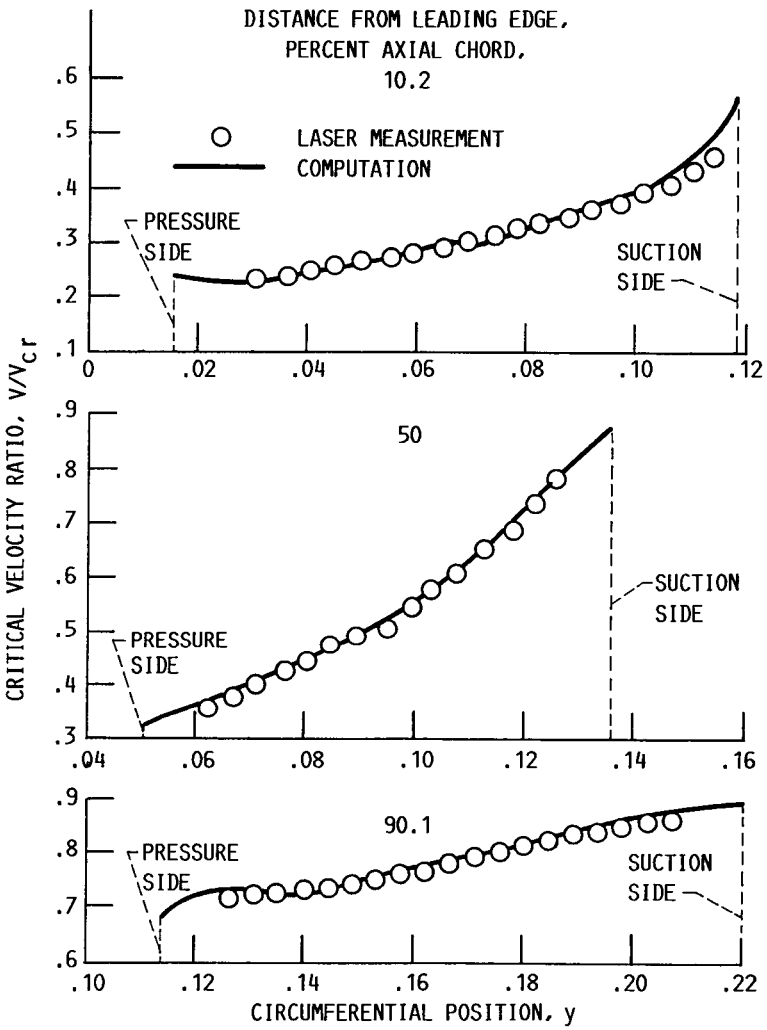


(c) CLOSER VIEW.

FIG. 7. - GRID GENERATION FOR A TURBINE CASCADE SIMULATION.



(a) CORE TURBINE STATOR VANE GEOMETRY, SURVEY LOCATIONS, CIRCUMFERENTIAL POSITION (y).



(b) COMPARISON OF CRITICAL VELOCITY RATIO.

FIG. 8. - COMPARISON OF COMPUTED RESULTS WITH EXPERIMENTAL DATA.

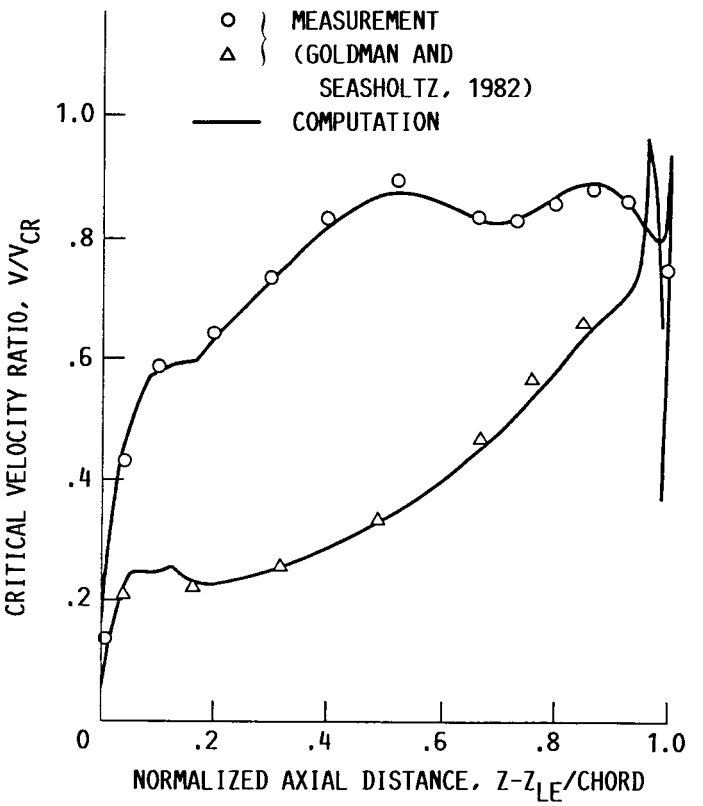
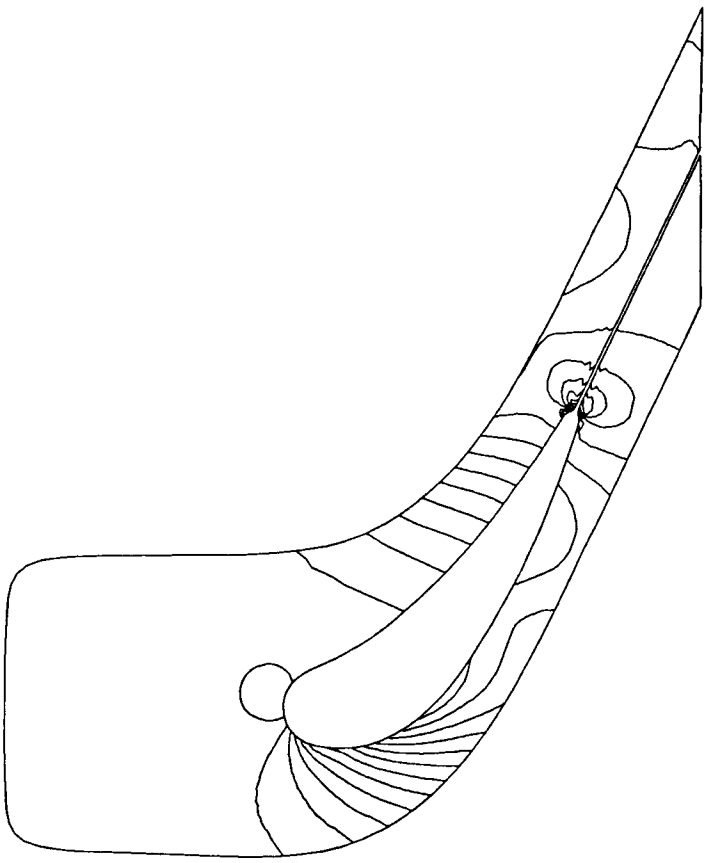


FIG. 9. - LOADING DIAGRAM.



INLET MACH 0.210 ALPHA 0.00
PS/POIN ABS MIN 0.704 MAX 1.169 INC 0.034

FIG. 10. - STATIC PRESSURE CONTOUR.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-101412		

4. Title and Subtitle

Application of a Lower-Upper Implicit Scheme and an Interactive Grid Generation for Turbomachinery Flow Field Simulations

5. Report Date

6. Performing Organization Code

7. Author(s)

Yung K. Choo, Woo-Yung Soh, and Seokkwan Yoon

8. Performing Organization Report No.

E-4374

10. Work Unit No.

505-62-21

9. Performing Organization Name and Address

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Memorandum

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546-0001

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared for the 34th International Gas Turbine and Aeroengine Congress and Exposition, sponsored by the American Society of Mechanical Engineers, Toronto, Canada, June 4-8, 1989. Yung K. Choo, NASA Lewis Research Center; Woo-Yung Soh, Sverdrup Technology, Inc., NASA Lewis Research Center Group, Cleveland, Ohio 44135; Seokkwan Yoon, MCAT Institute, NASA Ames Research Center, Moffett Field, California 94035.

16. Abstract

A finite-volume lower-upper (LU) implicit scheme is used to simulate an inviscid flow in a turbine cascade. This approximate factorization scheme requires only the inversion of sparse lower and upper triangular matrices, which can be done efficiently without extensive storage. As an implicit scheme it allows a large time step to reach the steady state. An interactive grid generation program (TURBO), which is being developed, is used to generate grids. This program uses the control point form of algebraic grid generation which uses a sparse collection of control points from which the shape and position of coordinate curves can be adjusted. A distinct advantage of TURBO compared with other grid generation programs is that it allows the easy change of local mesh structure without affecting the grid outside the domain of dependence. Sample grids are generated by TURBO for a compressor rotor blade and a turbine cascade. The turbine cascade flow is simulated by using the LU implicit scheme on the grid generated by TURBO.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Numerical scheme Grid generation Turbomachinery	Unclassified – Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No of pages	22. Price*
Unclassified	Unclassified	16	A03

NASA FORM 1626 OCT 86

*For sale by the National Technical Information Service, Springfield, Virginia 22161