Interactive Grid Generation for Turbomachinery Flow Field Simulations

Yung K. Choo
Lewis Research Center
Cleveland, Ohio

Peter R. Eiseman
Columbia University
New York, New York

and Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

Charles Reno
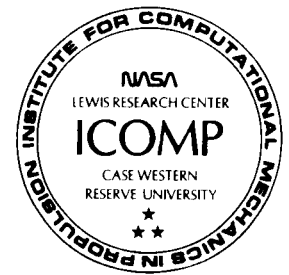*Lewis Research Center*
Cleveland, Ohio

(NASA-TM-101301) INTERACTIVE GRID
GENERATION FOR TURBOMACHINERY FLOW FIELD
SIMULATIONS (NASA) 10 p   CSCL 01A
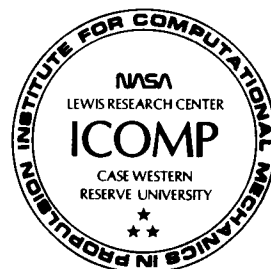
N89-11717

Unclas

G3/02   0174592

NASA

Interactive Grid Generation for Turbomachinery Flow Field Simulations

Yung K. Choo
Lewis Research Center
Cleveland, Ohio

Peter R. Eiseman
Columbia University
New York, New York

and Institute for Computational Mechanics in Propulsion
Lewis Research Center
Cleveland, Ohio

Charles Reno
Lewis Research Center
Cleveland, Ohio

Summary

The control point form of algebraic grid generation presented here provides the means that are needed to generate well-structured grids for turbomachinery flow simulations. It uses a sparse collection of control points distributed over the flow domain. The shape and position of coordinate curves can be adjusted from these control points while the grid conforms precisely to all boundaries. An interactive program called TURBO, which uses the control point form, is being developed. Basic features of the code are discussed and sample grids are presented. A finite-volume LU implicit scheme is used to simulate flow in a turbine cascade on the grid generated by the program.

Introduction

An interactive grid generation program called TURBO is being developed to generate well-structured grids for flow field simulations in turbomachinery. This program, which uses the control point form of algebraic grid generation [1], provides the means with which a balance can be achieved between the constraints from the geometry and desired grid structure.

Stator vanes and rotor blades of turbomachinery typically have a large camber, substantial thickness, and high solidity. These vanes and blades define highly constrained geometric regions for the flows in turbomachinery. Flow properties change rapidly in several subregions in which grid clustering is needed to resolve flow characteristics. The orthogonality desired in highly cambered blades conflicts with the periodicity constraints that are required by many flow solvers, and similarly the need to cluster the grid conflicts with the smoothness desired [2]. One of the best approaches to a sensible compromise between these conflicting demands would be to generate the grid interactively using a method that allows easy local control over the grid distribution.

The control point form of algebraic grid generation establishes local control points to control the grid structure. It uses a sparse collection of control points from which the shape and position of coordinate curves can be adjusted while the grid conforms precisely to all boundaries. As an algebraic method, it provides explicit control of the physical grid shape and grid spacing, and requires relatively few computations. The basic elements of this method are the multisurface transformation for control in a given direction [3] and the

Boolean sum operation for the combination directions.

The program TURBO generates single-block grids for stator vanes and rotor blades. It reads in discrete data that define the geometry, generates a surface grid, establishes a control net, computes a grid, and displays the initial results on the screen of an IRIS workstation. Selection and translation of a control point is easily accomplished by clicking the keys on the mouse and translating it. The grid structure can be changed by changing the locations of the control points. The boundary surface can be either a fixed form (transfinite) or a free form which can be changed in the same manner as the grid lines in the interior. Once the grid is improved, it can be closely examined by zooming, translating, and rotating. And further improvement of the grid structure can be made with this closer view of the grid. TURBO is an interactive algebraic grid generation code that uses the mouse to select options from the menu. Additional features that give users convenience include the direct insertion and deletion of control lines and the movement of groups of control points. A user also can attach a control net to an existing grid in order to improve its structure.

A finite-volume lower-upper implicit scheme [4,5] is used to simulate a flow in an annular turbine cascade [6] on the grid generated by the interactive program.

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 (figure 2). 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. This determines a smooth first derivative of the entire curve. The desired curve is then obtained by a parametric integration.

Let C_{1j}, C_{2j}, ..., C_{Mj}, be the given sequence of M points in space as shown in figure 2 (e.g., C_{12}, C_{22}, ..., C_{52} for $j = 2$ in figure 1); let r be the curve parameterization; let $E_j(r)$ be the position at r along the desired curve; let r_1, r_2, ..., r_{M-1} be the successive parametric locations to interpolate the directions of $(C_{2j} - C_{1j})$, $(C_{3j} - C_{2j})$, ..., $(C_{Mj} - C_{M-1,j})$; and let ψ_1, ψ_2, ..., ψ_{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 the notation, the desired curve is given by

$$E_j(r) = C_{1j} + \sum_{\alpha=1}^{N-1} G_\alpha(r) \left[C_{\alpha+1,j} - C_{\alpha j} \right] \quad j = 1, 2, ..., M \tag{1}$$

where

$$G_\alpha(r) = \int_{r_1}^{r} \psi_\alpha(\mu) d\mu \tag{2}$$

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 $r_1 < r_2 < ... < r_{M-1}$. 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 C_{Mj} by means of a telescopic collapse of terms. The development and application of such local controls is given in reference 3.

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

$$F_i(t) = C_{i1} + \sum_{\beta=1}^{M-1} H_\beta(t) \left[C_{i,\beta+1} - C_{i\beta} \right] \quad i = 1, 2, ..., N \tag{3}$$

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

$$T(r,t) = E_1(r) + \sum_{\beta=1}^{M-1} H_\beta(t) \left[E_{\beta+1}(r) - E_\beta(r) \right] \tag{4}$$

or alternatively by

$$T(r,t) = F_1(t) + \sum_{\alpha=1}^{N-1} G_\alpha(t) \left[F_{\alpha+1}(t) - F_\alpha(t) \right] \tag{5}$$

These two expressions are equivalent. The tensor product matches E_j or 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 $P(r,t)$ 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 $P(1,t)$, $P(N-1,t)$, $P(r,1)$, and $P(r,M-1)$. 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 T for the r and t directions, respectively. In each such directional construction, the actual boundaries appear as end conditions for the corresponding variable while the remaining boundaries are solely generated by the control points. Thus, by subtracting 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

$$\begin{aligned} Q(r,t) = T(r,t) &+ \alpha_1 \left[1 - G_1(r) \right] \left[P(1,t) - F_1(t) \right] \\ &+ \alpha_2 G_{N-1}(r) \left[P(N-1,t) - F_N(t) \right] \\ &+ \alpha_3 \left[1 - H_1(t) \right] \left[P(r,1) - E_1(r) \right] \\ &+ \alpha_4 H_{M-1}(t) \left[P(r,M-1) - E_M(r) \right] \end{aligned} \tag{6}$$

where each of the four terms following the tensor product $T(r,t)$ 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 of appearance, the boundaries are for $r = 1$, $r = N-1$, $t = 1$, and $t = M-1$. Further details are given in reference 1.

Finite-Volume LU Implicit Scheme

An unconditionally stable implicit scheme that has error terms at most of order $(\Delta t)^2$ in any number of space dimensions can be derived by lower-upper (LU) factorization [4,5]. Splitting ensures the diagonal dominance of lower and upper factors and makes use of the built-in implicit dissipation.

The LU implicit scheme only requires the inversion of sparse triangular matrices, which can be done efficiently without using large storage.

Grid Generation Using TURBO

The basic interactive process of TURBO is illustrated with a rather simple geometry in figure 3. When fully developed, TURBO will be a menu-driven code that operates by use of the mouse only without using any keys on the keyboard.

The first graphic display in a TURBO session is a view which shows an initial control net overlaid on an initially computed grid. If needed, with a stroke of a key, the user can call upon the graphical help system of TURBO at any time. The display tells users where they are in the interactive process and what options are currently available. The help system at this stage is shown in figure 4. The double circles in the middle (one red circle and one blue circle on the screen) tell the user that the process is at the "BASIC VIEW." Various activities one can engage in from the "BASIC VIEW" are displayed in smaller circles. The arrows on the lines indicate possible directions to go from the "BASIC VIEW" to one of the interactive features, and letters in the squares and rectangles indicate actions needed. Currently available options are listed on the left-hand side of the screen. At the "BASIC VIEW," for example, three options are listed: (1) Q, to quit from the TURBO session, (2) N, to scroll the graphic display from the control net and grid combination to the grid only, and then to the control net only, and (3) W, to read in previously created control net data.

After the initial grid is obtained, a user may want to have a closer look at the grid. Pressing the V-key gets the user to the "MODIFY VIEW" feature shown in figure 4. The help system now has two circles around the "MODIFY VIEW," and provides options to expand, to translate, and to rotate the graphic object on the screen. With these options the user can expand and examine the grid closely to identify the area where the grid structure needs improvement. Pressing the R-key returns the process back to the "BASIC VIEW" with the expanded graphic object.

Selection and translation of control points are accomplished by clicking the keys on the mouse and translating it. After the control net is modified, a new grid is computed by simply pressing a key (C-key for compute). Then the program computes a new grid based on the modified control net and returns the process back to the "BASIC VIEW."

Other features of TURBO can be accessed and used in a similar fashion. They include (1) insertion and deletion of control lines ("MODIFY CONTROL NET" feature in figure 4), (2) rubber banding (or rubber sheeting) which allows the movement of several control points (or a group of control points) simultaneously, and (3) a dynamic change of the total number of control points and grid points.

Figure 5 shows an H-grid generated for a cambered turbine stator vane. In this example only those interactive features illustrated in figure 3 were used. Figure 5(a) shows an initial control net; figure 5(b) shows an initial grid. By scrolling graphic images from 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 can determine what modification of the control net is needed to improve the grid structure. The modified control net and grid are shown in figure 5, parts (c) and (d), respectively. The modified grid has a densely clustered mesh around the leading edge and has a more nearly orthogonal grid near the vane surfaces.

Many grid generation programs do not allow local mesh control as TURBO does. The initial grid may be generated by any program familiar to a user, and then an initial control net may be obtained from the grid (by "attachment") to essentially reproduce the grid structure. Figure 6(a) shows a grid generated by GRAPE [7] for a compressor rotor blade. Figure 6(b) shows a control net created by attachment. Once the control net is created, the interactive process to be followed is the same as described previously. The initial control net is then modified to the one shown in figure 6(c) by using the rubber banding feature of the program. The two red control lines in figure 6(c) are about to be changed by rubber banding. In this exercise, the control lines are hinged at the rotor blade surface. As one control point on the outer periodic boundary is moved, all control points on the two red lines move as in a rubber band while maintaining the periodic condition. A new grid generated from the modified control net is shown in figure 6(d).

Flow Simulation

As a means of demonstrating the usefulness of this grid generation technique, the finite-volume LU implicit scheme [5] was used to simulate a flow in a turbine cascade on the grid generated by TURBO. The computed loading diagram is compared with measured data [6] in figure 7. It shows good agreement. Static pressure contours in the flow field of the stator vane are presented without comparison in figure 8 because of insufficient experimental data. The overall pressure distribution is reasonable.

References

1. EISEMAN, P.R.—A Control Point Form of Algebraic Grid Generation. *Numerical Methods in Laminar and Turbulent Flow*, Vol. 5, Part 2, C. Taylor, W.G. Habashi, and M.M. Hafez, eds., Pineridge Press, Swansea, U.K., 1987, pp. 1083–1091.
2. CHOO, Y.K., YOON, S., and CIVINSKAS, K.C.—Composite Grid and Finite-Volume LU Implicit Scheme for Turbine Flow Analysis. *AIAA 8th Computational Fluid Dynamics Conference*, AIAA, New York, 1987, pp. 313–321. (AIAA Paper 87-1129, NASA TM-89828).

3. EISEMAN, P.R.—Coordinate Generation with Precise Controls over Mesh Properties. *J. Comput. Phys.*, Vol. 47, pp. 331–351, Sept. 1982.

4. JAMESON, A. and TURKEL, E.—Implicit Schemes and LU Decompositions. *Math. Comput.*, Vol. 37, No. 156, pp. 385–397, Oct. 1981.

5. JAMESON, A. and YOON, S.—Lower-Upper Implicit Schemes with Multiple Grids for the Euler Equations. *AIAA J.*, Vol. 25, No. 7, pp. 929–935, July 1987.

6. GOLDMAN, L.J. and SEASHOLTZ, R.G.—Laser Anemometer Measurements in an Annular Cascade of Core Turbine Vanes and Comparison with Theory. NASA TP-2018, 1982.

7. SORENSON, R.L.—A Computer Program to Generate Two-Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation. NASA TM-81198, 1980.
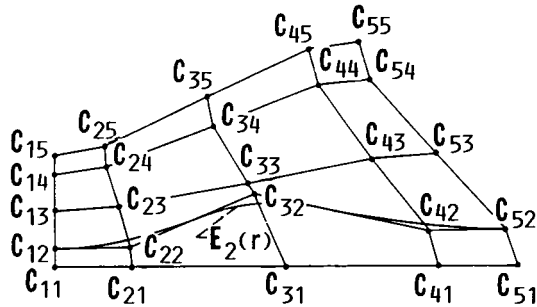
Figure 1.—Control net—an example in two dimensions.
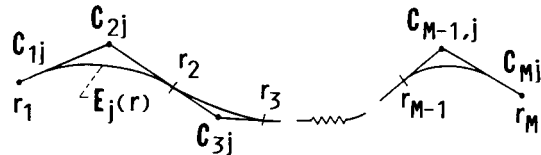
Figure 2.—Construction of a curve.

CONSTRUCT SIMPLE CONTROL NET; GENERATE SURFACE GRID

COMPUTE AND EXAMINE INITIAL GRID

TRANSLATE CONTROL POINT (E.G., TO GET FINE MESHES IN THE MIDDLE)

SELECT CONTROL POINT TO MODIFY GRID

RE-COMPUTE

EXAMINE NEW GRID - ACCEPT OR REPEAT PROCESS
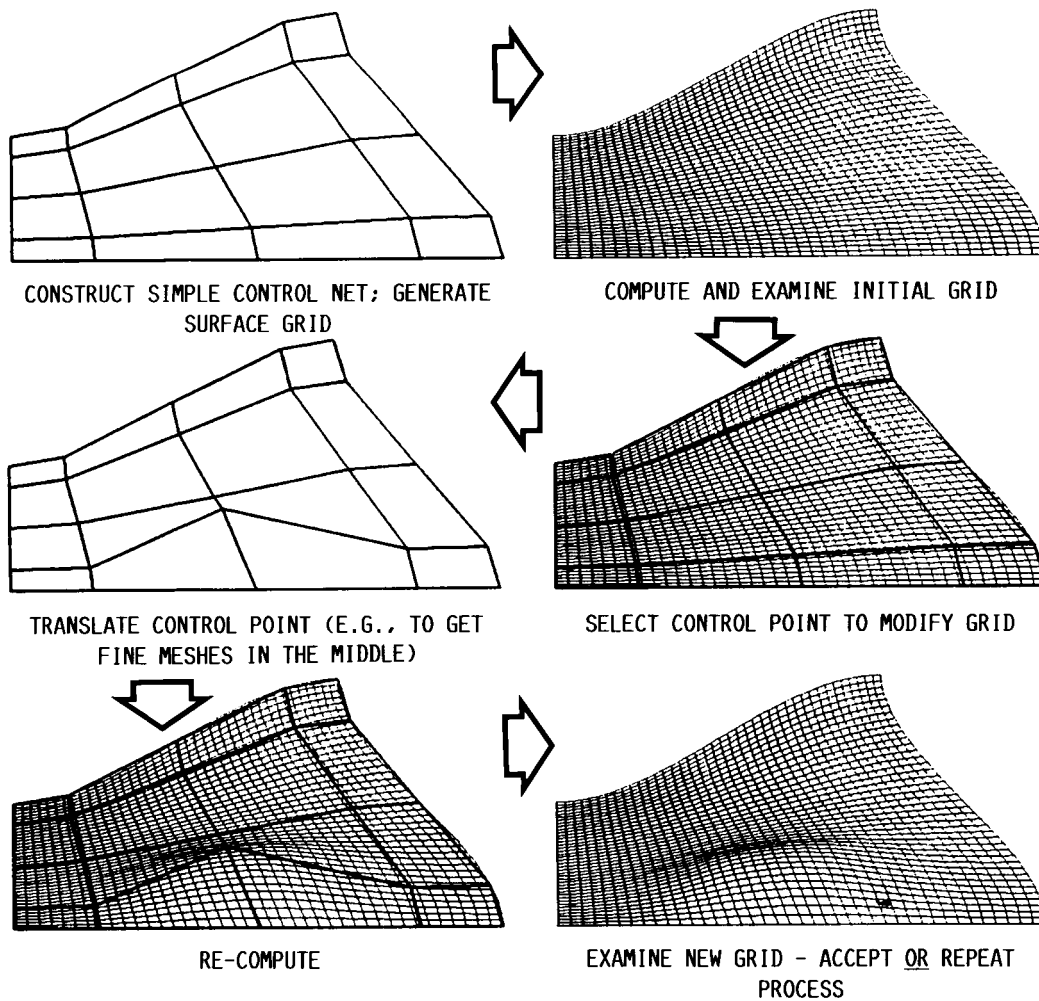
Figure 3.—Basic interactive process.

Figure 4.—Graphical help system.

(a) Initial control net.
(b) Initial grid.
(c) Modified control net.
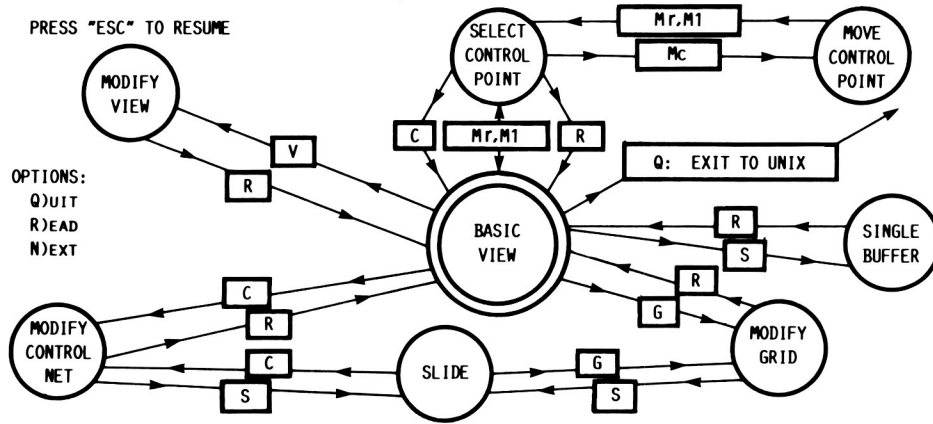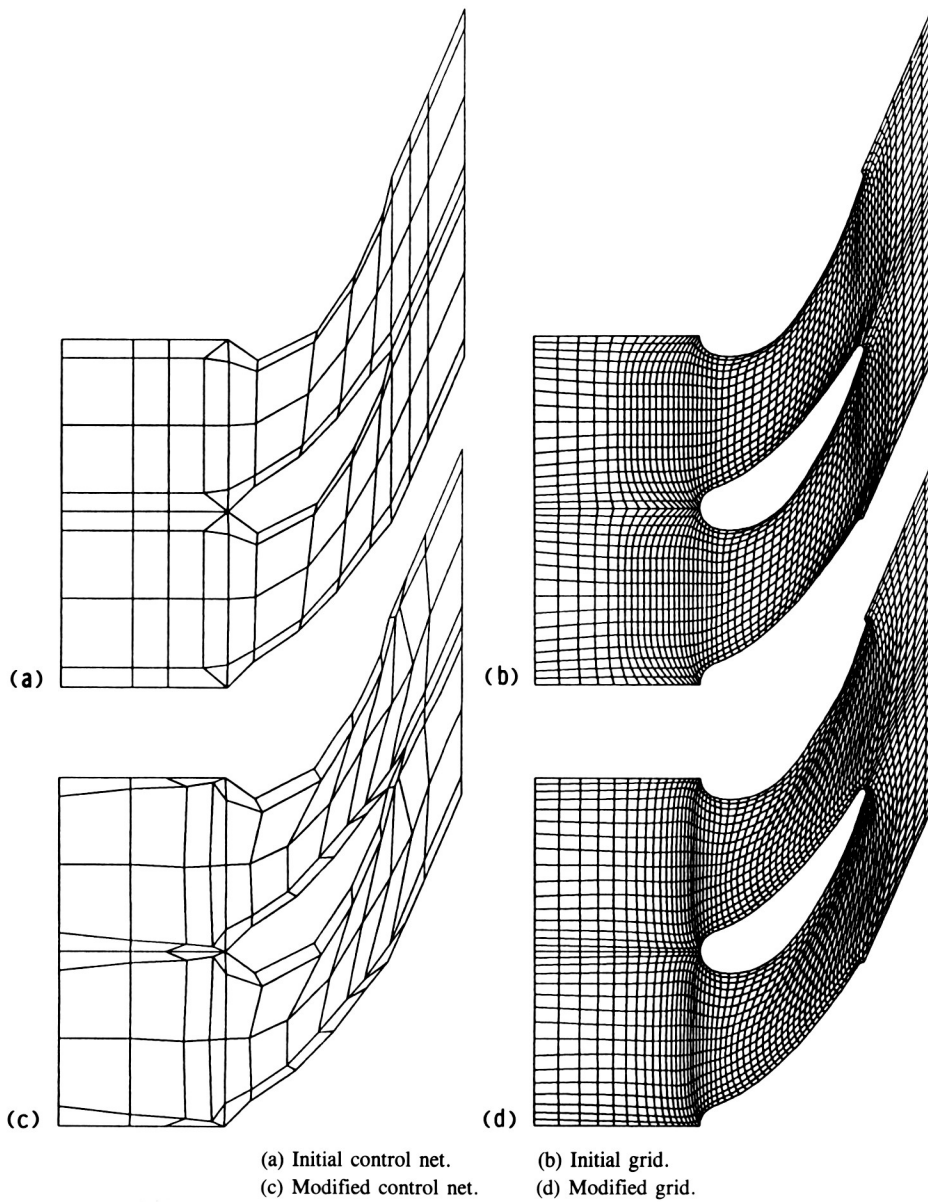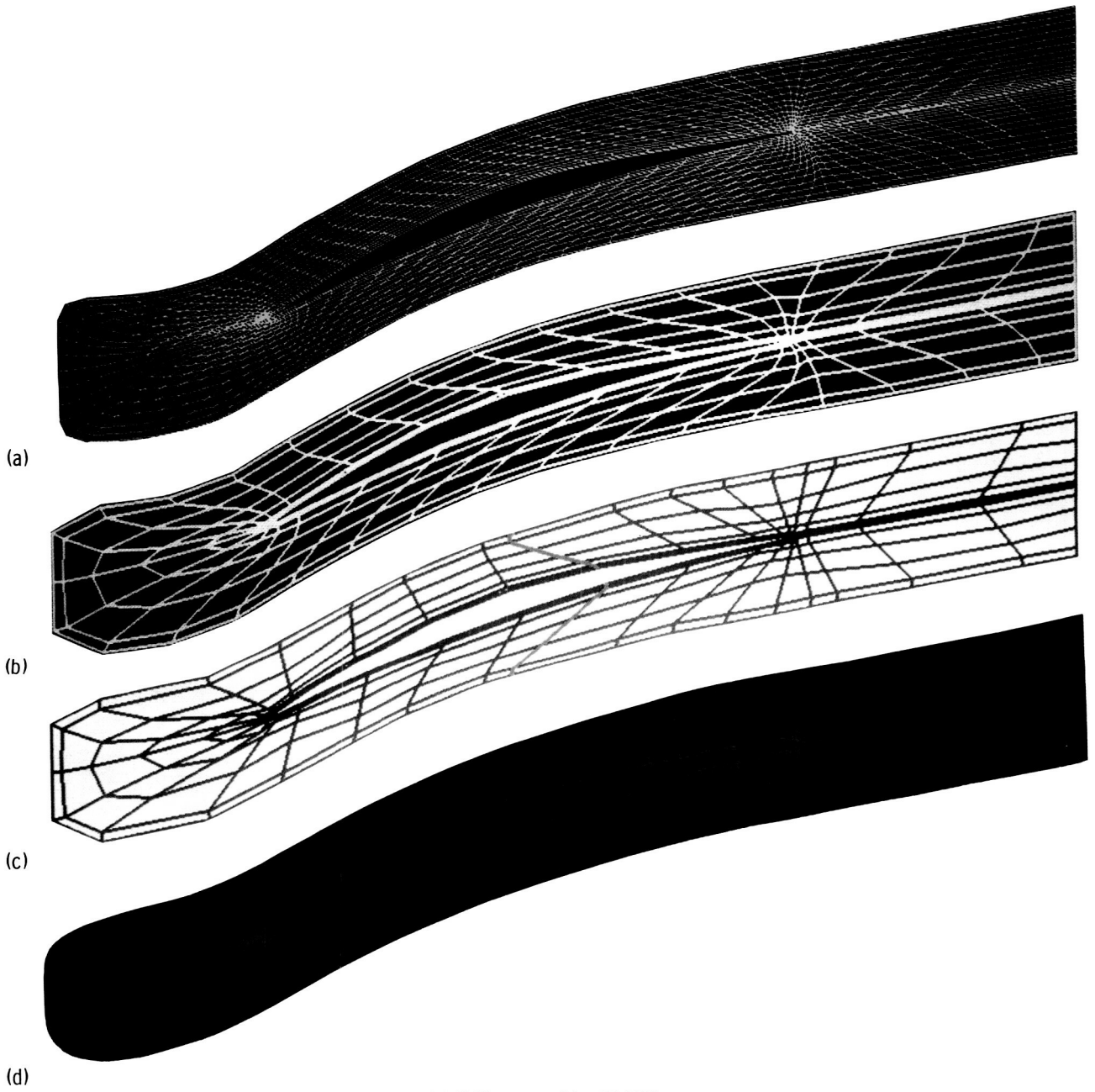(d) Modified grid.

Figure 5.—Grid generation for a cambered stator vane.

(a) Grid generated by GRAPE.
(b) Attached control net.
(c) Modified control net.
(d) Modified grid.

Figure 6.—Grid generation for a compressor rotor blade.

6
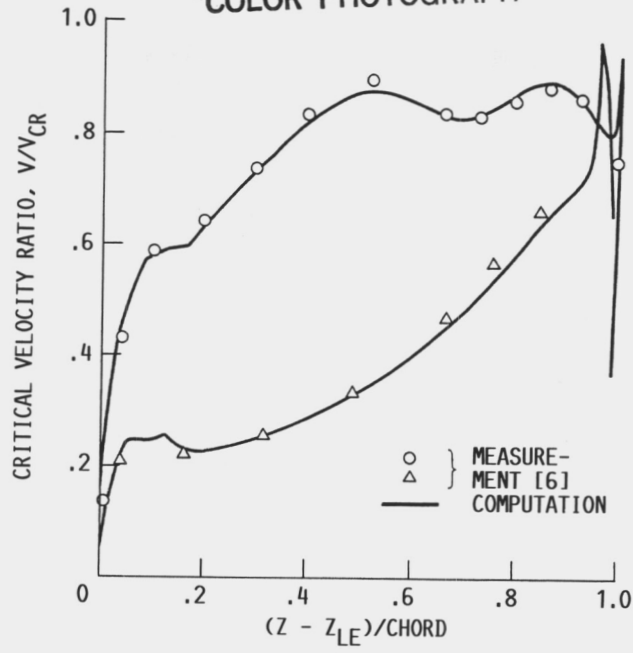
Figure 7.—Loading diagram.
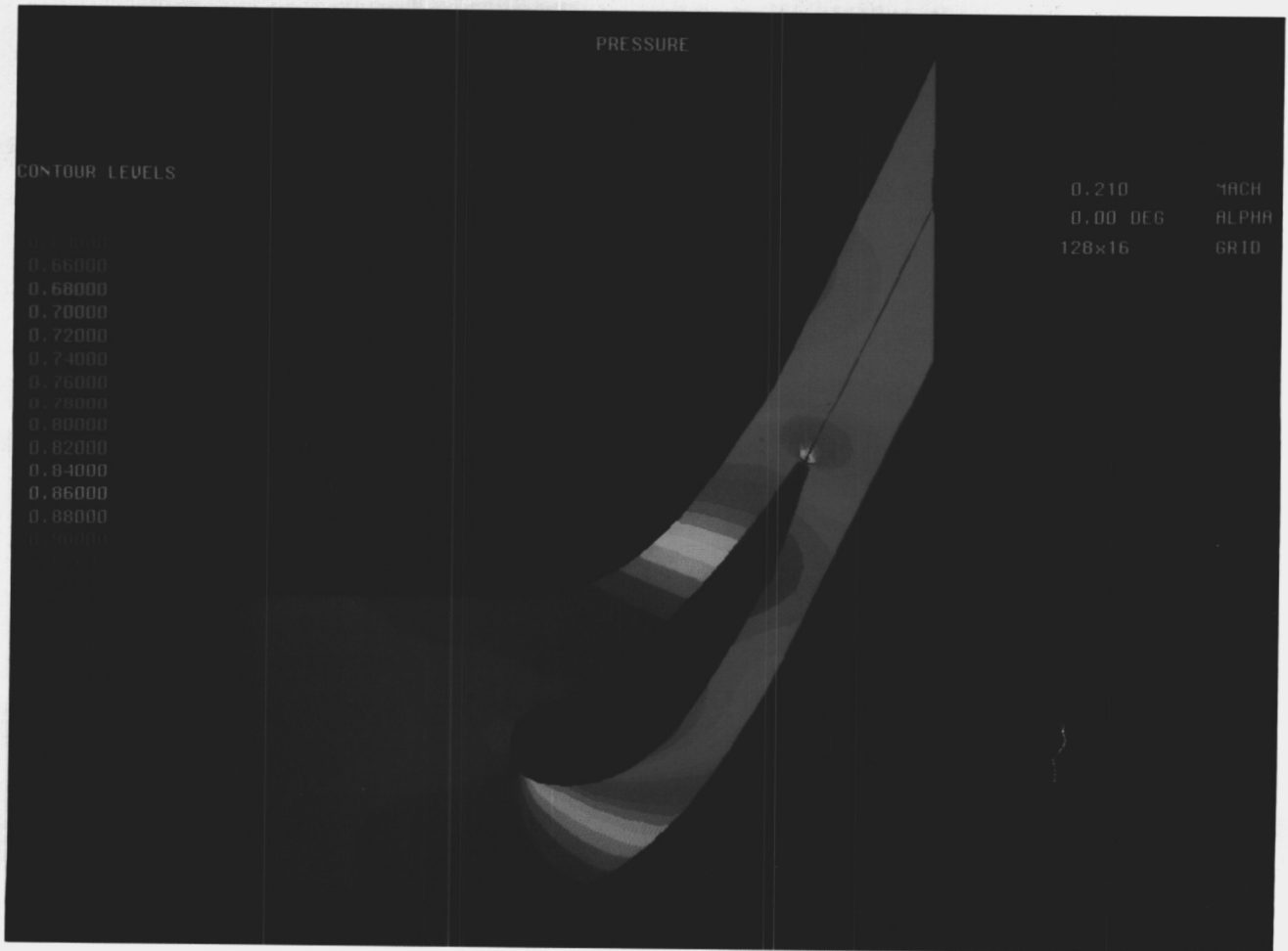
Figure 8.—Static pressure contour.

Report Documentation Page

1. Report No. NASA TM-101301 ICOMP-88-15	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Interactive Grid Generation for Turbomachinery Flow Field Simulations		5. Report Date
		6. Performing Organization Code
7. Author(s) Yung K. Choo, Peter R. Eiseman, and Charles Reno		8. Performing Organization Report No. E-4282
		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

16. Abstract

The control point form of algebraic grid generation presented here provides the means that are needed to generate well-structured grids for turbomachinery flow simulations. It uses a sparse collection of control points distributed over the flow domain. The shape and position of coordinate curves can be adjusted from these control points while the grid conforms precisely to all boundaries. An interactive program called TURBO, which uses the control point form, is being developed. Basic features of the code are discussed and sample grids are presented. A finite-volume LU implicit scheme is used to simulate flow in a turbine cascade on the grid generated by the program.

17. Key Words (Suggested by Author(s)) Grid generation Turbomachinery	18. Distribution Statement Unclassified – Unlimited Subject Category 02		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No of pages 9	22. Price* A02

NASA FORM 1626 OCT 86   *For sale by the National Technical Information Service, Springfield, Virginia 22161