

NASA Technical Memorandum 103748

1N-02  
1695  
P13

## Implementation of Control Point Form of Algebraic Grid-Generation Technique

(NASA-TM-103748) IMPLEMENTATION OF CONTROL  
POINT FORM OF ALGEBRAIC GRID-GENERATION  
TECHNIQUE (NASA) 13 p CACL 01A

N91-19054

Unclas

G3/02 0001695

Yung K. Choo and David P. Miller  
*Lewis Research Center*  
*Cleveland, Ohio*

Charles J. Reno  
*Cleveland State University*  
*Cleveland, Ohio*

Prepared for the  
Third International Conference on Numerical Grid Generation in  
Computational Fluid Dynamics and Related Fields  
cosponsored by the UPC, University of Catalonia and European Space Agency  
Barcelona, Spain, June 3-7, 1991

**NASA**



## IMPLEMENTATION OF CONTROL POINT FORM OF ALGEBRAIC GRID-GENERATION TECHNIQUE

Yung K. Choo and David P. Miller  
National Aeronautics and Space Administration  
Lewis Research Center  
Cleveland, Ohio 44135 U.S.A.

Charles J. Reno  
Cleveland State University  
Cleveland, Ohio 44115 U.S.A.

### Abstract

The control point form (CPF) provides explicit control of physical grid shape and grid spacing through the movement of the control points. The control point array, called a control net, is a sparse grid-type arrangement of locations in physical space with an index for each direction. As an algebraic method CPF is efficient and works well with interactive computer graphics. A family of menu-driven, interactive grid-generation computer codes (TURBO) is being developed by using CPF. Key and unique features of TurboI (a TURBO member) are discussed and typical results are presented. TurboI runs on any IRIS 4D series workstation.

### 1. INTRODUCTION

The numerical solution of fluid flow problems is dependent on the discrete representation of the flow domain. In order to accurately simulate complex flows, it is important to use a grid that is boundary fitted, aligned with the flow, concentrated where flow properties change rapidly, and as nearly orthogonal as possible. But geometric constraints often conflict with these desired grid qualities. One of the best approaches to a sensible compromise between these conflicting demands would be to generate the grid interactively by using a method that allows easy and precise local control over the grid distribution.

The control point form (CPF) provides precise control of the physical grid shape and grid spacing through the movement of the control points. The control point array is a sparse grid-type arrangement of locations in physical space with an index for each direction. Each control point has a domain of influence on the locally surrounding grid points. This local control capability of CPF is invaluable when a compromise is needed in a conflict between the desired grid structure and geometric constraints. Two- and three-dimensional forms of CPF and earlier demonstrations are presented in detail in references 1 to 4, but for completeness and clarity the two-dimensional form of CPF is included in this paper.

A family of menu-driven, interactive grid-generation programs, called TURBO, is being developed by using the CPF technique to generate grids for numerical flow simulations. This paper focuses on TurboI, a member of the TURBO family. TurboI is the very basic grid-generation code of TURBO.

Other TURBO family members under development are TurboAZ, which will have automatic zoning capability; TurboAD, which will adapt the grid to solutions, grid quality measure, and geometry; and TurboT, which will be specialized for turbomachinery. With TurboI a user can interactively change the structure of the control net and subsequently change the grid structure. Key features of TurboI are discussed with illustrations in subsequent sections. TurboI constructs an initial control net (from either a given boundary grid or a given field grid that requires modification), has various levels of local and global grid control, redefines a block to operate on, and performs interactive curve discretization for a boundary grid. Boundary curve discretization, if desired, can be performed interactively by choosing a curve segment and assigning a desired number of points on the chosen segment. A damped bicubic spline (ref. 5) is available for discretization of curve segments with sharp corners. TURBO can be used as a stand-alone code as well as a supportive and complementary operating tool.

## 2. CONTROL POINT FORM OF ALGEBRAIC GRID-GENERATION TECHNIQUE

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 between two opposing boundaries (fig. 2). These boundaries are the fixed endpoints of the curves. The remaining control points are in the interior of the sequence and are used to control the shape of the curve.

From any one control point the two neighboring points on each side form local segments within the entire piecewise linear curve. The local segments about each control point define a change in direction as shown in figures 1 and 2. The rate of change is determined by the two linear connections attached to the given control point as the curve assumes the respective directions between each pair of control points.

A continuous direction field is obtained in a smooth manner by interpolation. The independent variable for the interpolation is the curve parameterization. The interpolated result defines the field of vectors that are tangent to the desired curve. Simply stated it is an interpolation of first parametric derivatives. This determines a smooth first derivative of the

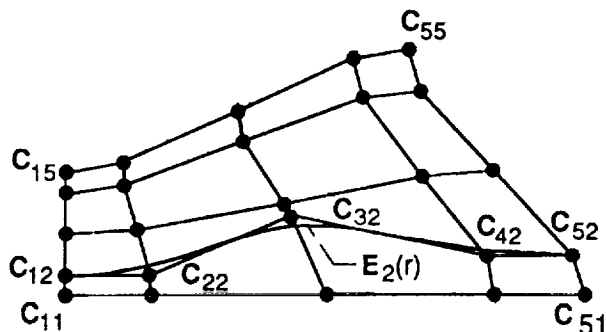


Figure 1.—Control net and a curve.

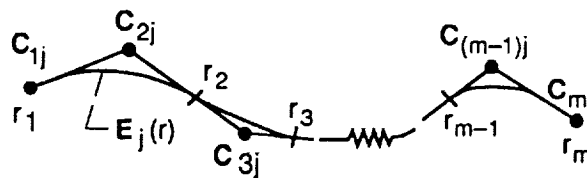


Figure 2.—Construction of a curve.

entire curve. The desired curve is then obtained by a parametric integration. The integration here is taken so that the curve connects the specified endpoints.

To state the result mathematically, let  $C_{1j}, C_{2j}, \dots, C_{Mj}$  be the given sequence of  $M$  points in space as shown in figure 2 (e.g.,  $C_{12}, C_{22}, \dots, C_{52}$  for  $j=2$  in fig. 1); let  $r$  be the curve parameterization; let  $E_j(r)$  be the position at  $r$  along the desired curve; let  $r_1, \dots, r_2, \dots, r_{M-1}$  be the successive parametric locations used to interpolate the directions of  $(C_{2j} - C_{1j}), (C_{3j} - C_{2j}), \dots, (C_{Mj} - C_{(M-1)j})$ ; and let  $\psi_1, \psi_2, \dots, \psi_{M-1}$  be the corresponding interpolation functions that separate each successive pair of segments by assuming a nonzero value at the associated location while vanishing at the remaining locations for interpolation. In two dimensions  $C_{aj} = (x_{aj}, y_{aj})$  and  $E_j(r) = [x_j(r), y_j(r)]$ . With the notation the desired curve is given by

$$E_j(r) = C_{1j} + \sum_{\alpha=1}^{M-1} G_\alpha(r) [C_{(\alpha+1)j} - C_{\alpha j}] \quad \text{for } j = 1, 2, \dots, M \quad (1)$$

where

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

Local interpolation functions are used to obtain local grid controls. With local functions the movement 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. The simplest local interpolants are the piecewise linear functions that do not vanish over at most two intervals defined by  $r_1 < r_2 < \dots < r_{M-1}$ . With the uniform partition  $r_k = k$  for simplicity, an explicit form of the normalized interpolants, which yields  $G_k(r_{N-1}) = 1$  for all  $k$ , can be given by

$$\psi_1(r) = \begin{cases} 2(2-r) & \text{for } 1 \leq r < 2 \\ 0 & \text{for } 2 \leq r \leq (N-1) \end{cases} \quad (3)$$

$$\psi_k(r) = \begin{cases} 0 & \text{for } 1 \leq r < (k-1) \\ (r-k)+1 & \text{for } (k-1) \leq r < k \\ (k-r)+1 & \text{for } k \leq r < (k+1) \\ 0 & \text{for } (k+1) \leq r \leq (N-1) \end{cases} \quad (4)$$

$$\psi_{(N-1)}(r) = \begin{cases} 0 & \text{for } 1 \leq r < (N-2) \\ 2(r-N+2) & \text{for } (N-2) \leq r \leq (N-1) \end{cases} \quad (5)$$

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

$$F_i(t) = C_{1i} + \sum_{\beta=1}^{M-1} H_{\beta}(t) (C_{i,\beta+1} - C_{i\beta}) \quad \text{for } i = 1, 2, \dots, N \quad (6)$$

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

$$T(r, t) = E_1(r) + \sum_{\beta=1}^{M-1} H_{\beta}(t) [E_{\beta+1}(r) - E_{\beta}(r)] \quad (7)$$

or alternatively by

$$T(r, t) = F_1(t) + \sum_{\alpha=1}^{N-1} G_{\alpha}(t) [F_{\alpha+1}(r) - F_{\alpha}(r)] \quad (8)$$

These two expressions are equivalent. The tensor product matches  $E_j$  or  $F_i$  at the extremities of  $i$  and  $j$  (e.g., at corner points in fig. 1).

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 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  $P(r, M-1)$ . In order to include the boundaries, the multisurface transformation is performed again as before, 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 and 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

$$\begin{aligned} Q(r, t) = & T(r, t) + \alpha_1 [1 - G_1(r)] [P(1, t) - F_1(t)] \\ & + \alpha_2 G_{N-1}(r) [P(N-1, t) - F_N(t)] \\ & + \alpha_3 [1 - H_1(t)] [P(r, 1) - E_1(r)] \\ & + \alpha_4 H_{M-1}(t) [P(r, M-1) - E_M(r)] \end{aligned} \quad (9)$$

where each of the four terms following the tensor product  $T(r, t)$  represents a transfinite conformity to a boundary when each boundary on-off switch is 1. When any  $\alpha_i$  is set 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  $t = M - 1$ .

### 3. IMPLEMENTATION OF CPF (PROGRAM TurboI)

#### 3.1 Construction of Initial Control Net

TurboI first constructs an initial control net from either given geometry data, a given boundary grid, or any given field grid that one wants to modify. Once a control net is constructed, the grid control features of TurboI are ready to operate. When geometry data are given, they are usually not sufficient to be used as a boundary grid, as for example in figure 3(a). In such a case TurboI interactively chooses a boundary curve segment, assigns the desired number of grid points, and stretches them as needed, as shown in figure 3(b). An initial control net is then constructed by an interpolation (fig. 4). Then from the boundary grid and the initial control net, the initial grid is calculated as shown in figure 5. If the field grid in figure 5 is given instead of the geometry data or the boundary grid, an initial control net similar to the one in figure 4 is constructed by "attachment" (ref. 3) to the given grid. Then TurboI produces a field grid, which is similar to the given one, from the boundary grid of the given data and the control net.

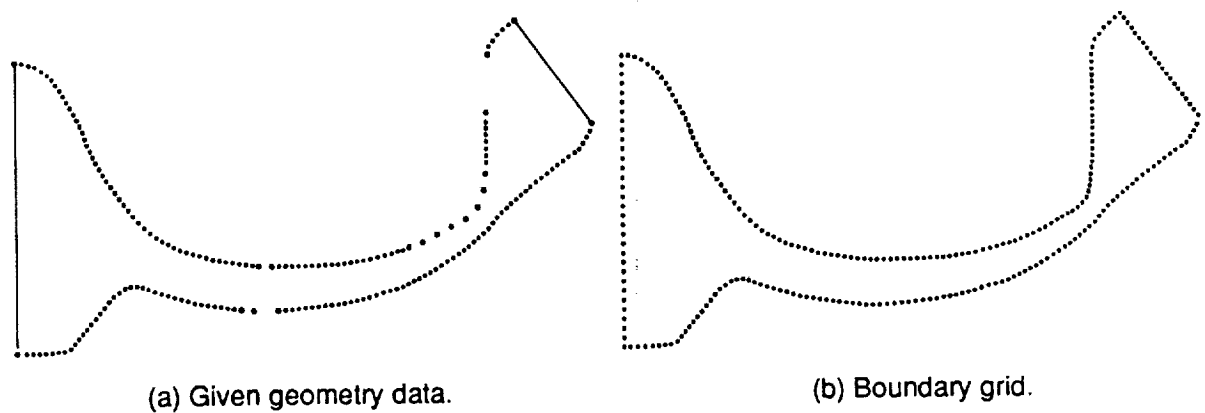


Figure 3.—Boundary grid generation.

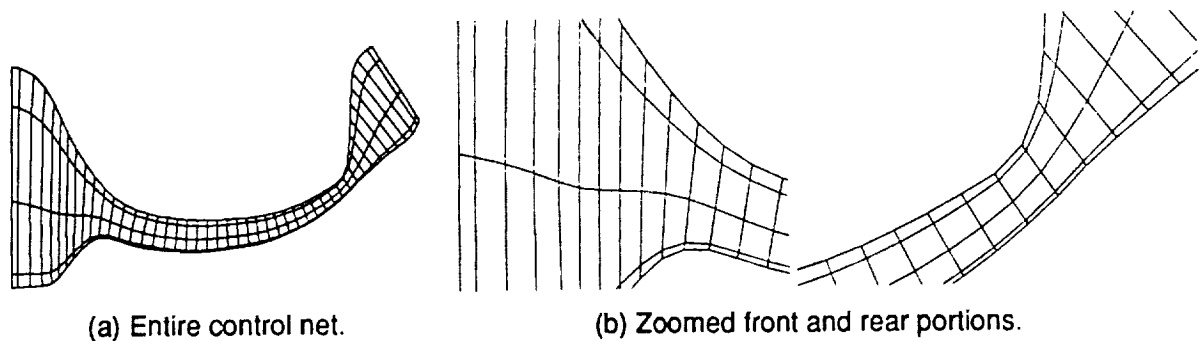


Figure 4.—Construction of initial control net by interpolation.

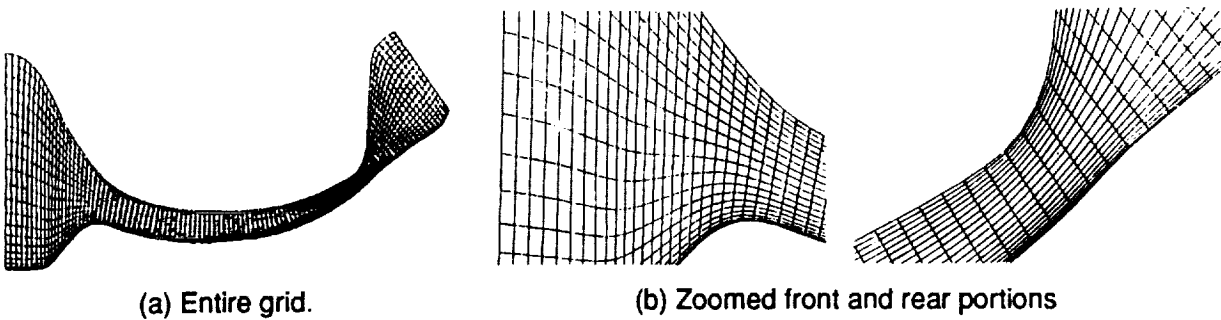


Figure 5.—Calculation of initial grid.

### 3.2 Primary TurboI Features

Using workstation mouse keys, users can select any control point in the control net. It can then be translated to a new position by translating the mouse. Each control point has a domain of influence on the locally surrounding grid points. The change of a control point location does not affect the grid points outside the domain of its influence. The point-by-point control of the control net is the basic control feature of TurboI. This basic control capability of CPF in two and three dimensions is illustrated in references 2 and 3, respectively.

For more precise control one needs to use more control points. Control points can be added or subtracted from a control net interactively by selecting the menu item "Change Number of CPs" and then using the workstation mouse as instructed on the screen. The number of control points between the bottom and top boundaries was increased from five in the initial control net to seven in the control net of figure 6.

The point-by-point control of a control net can be a very tedious process. In order to improve the efficiency of the interactive process and to give convenience to users, global control capabilities have been added to TurboI. One example of this is making the control net orthogonal to the boundaries. One only needs to choose menu items. Then TurboI finds normal vectors to the boundary and makes the control net orthogonal to the boundary. The control net shown in figure 6 was changed to the new one shown in figure 7 by choosing the menu item "Make Control Net Orthognl to Bndry." The new control net is orthogonal to the boundary. This modification in the control net structure changed the grid structure from the initial one in figure 5 to the new grid, which is orthogonal to boundaries as shown in figure 8(a).

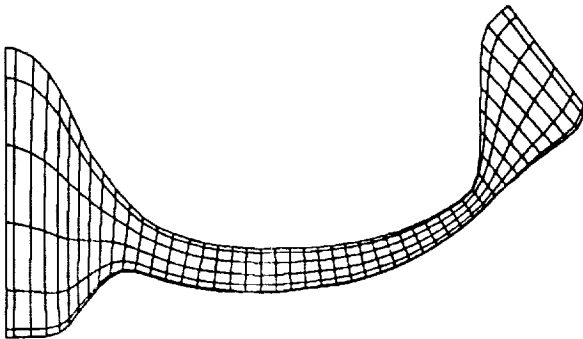


Figure 6.—Control net with increased control points.

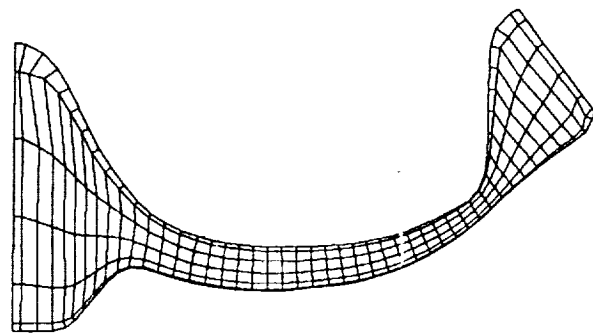


Figure 7.—Control net orthogonal to boundary.



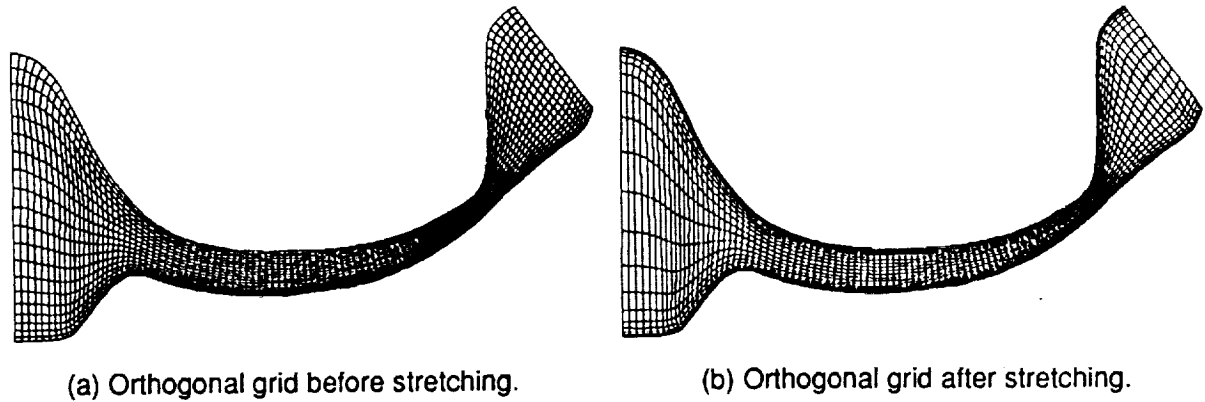


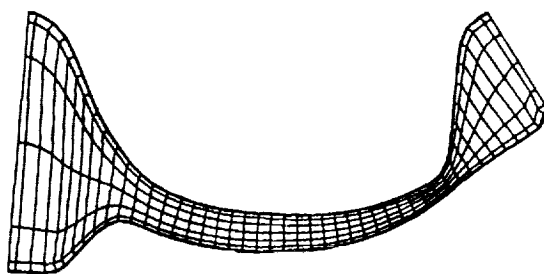
Figure 8.—Grid stretching.

Another global control capability is for grid stretching. Grid stretching can be accomplished with TurboI in two ways: through the stretching of a control net or through direct grid stretching. Direct grid stretching using a stretching function results in more satisfying results, especially for viscous flow simulations. The degree of stretching can be changed interactively with TurboI. When the grid is stretched as shown in figure 8(b), TurboI automatically constructs a new control net by reattachment. Further refinement of the grid structure is possible. As an example, the number of control points was increased further from seven to nine between the bottom and top boundaries. The new control net was then made orthogonal to the boundary. This time the orthogonality was extended to the second control points from the boundary, as shown in figure 9. A new grid was calculated and is shown in figure 10. An entire view, a zoomed front portion, and a zoomed rear portion of the new modified grid are shown in parts (a), (b), and (c) of the figure, respectively.

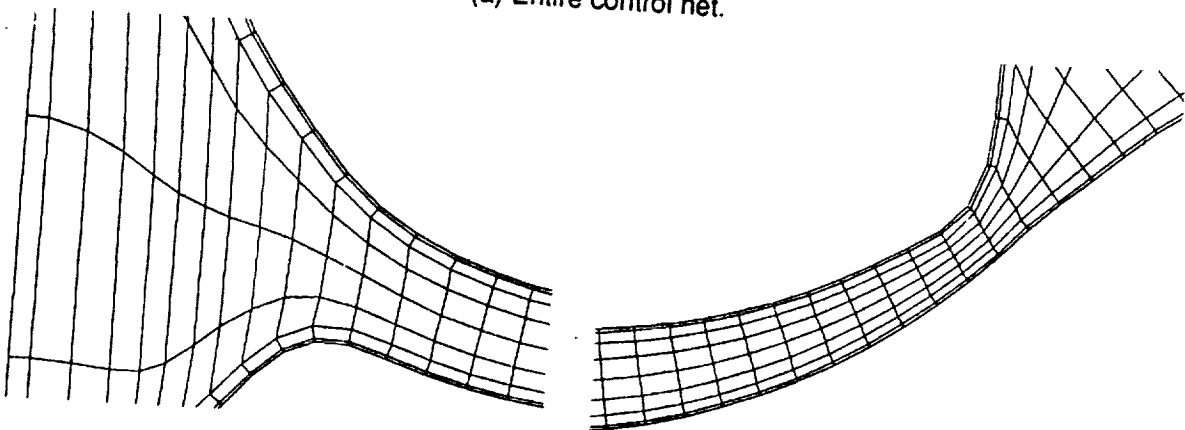
TurboI can choose a control point,  $C_{ij}$ , as a hinge point and choose another control point,  $C_{ij+n}$ , as a current control point (an intersection of two highlighted control net lines on the workstation screen) and can move several control points ( $n - 1$  points) between the two simultaneously as if they are on a stretched rubberband between  $C_{ij}$  and  $C_{ij+n}$ . This feature is particularly useful when this operation is needed for several values of  $i$  as illustrated in reference 2.

### 3.3 Geometry Manipulation Using Free-Form Boundary

The default boundary condition of TurboI is a Dirichlet boundary condition (i.e., the initial boundary grid distribution is fixed). But if desired, the boundary grid can be allowed to move to meet the Neumann boundary condition that is specified by the orthogonal control net, for instance, or to change the geometry itself. To allow the boundary grid points to move, select the menu item "Free Form Boundary," which is equivalent to turning off one of the boundary on-off switches ( $\alpha_i$ ) of equation (9). For illustration, eight control points ( $C_{11,1}, \dots, C_{11,4}$  and  $C_{12,1}, \dots, C_{12,4}$ ) of the control net in figure 9 were moved to the new positions shown in figure 11(a) by using the point-by-point control feature. The menu item "Free Form Boundary" was chosen to set  $\alpha_1 = 0$  and a new grid was calculated. Figure 11(b) shows the change of the bottom boundary due to the use of the free-form boundary and the domain of influence of the eight control points whose positions were changed.



(a) Entire control net.



(b) Zoomed front and rear portions.

Figure 9.—Extension of orthogonality to second control points from boundary.

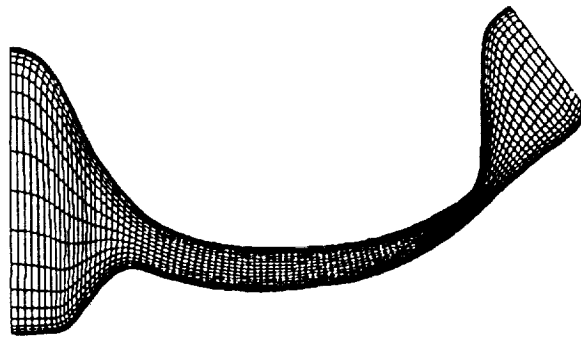
If the free-form boundary is used on a straight-line boundary, the boundary grid points can be redistributed without altering the geometry. For the grids shown in figures 7, 8, and 10, the free-form boundary feature was turned on for the straight-line boundaries on the left and right ends of the duct.

### 3.4 Interactive Block Operation

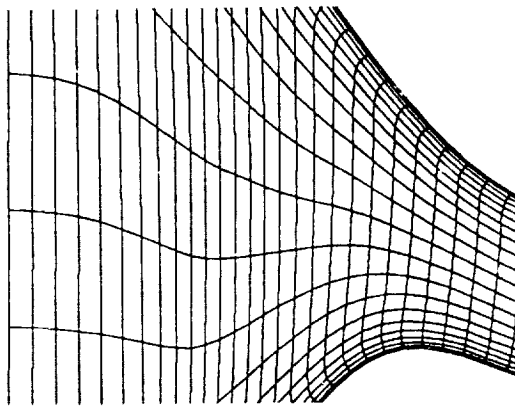
TurboI can define a subblock and operate on it alone. All interactive features of TurboI, including those used in the previous examples, are available to the subblock. Selection of the menu item called "Redefine Block" gets TurboI ready to define a subblock. Actual selection of a subblock is done by moving the cursor to the desired subblock corner points and then clicking workstation mouse keys. Figure 12 illustrates the subblock operation. In the example shown in figure 12(a), which shows the grid before subblock operation, the two corner points  $(i_1, j_1)$  and  $(i_2, j_2)$  define the subblock. Figure 12(b) is the modified control net for the subblock. Figure 12(c) is the modified grid after the subblock operation.

### 3.5 TurboI Menus

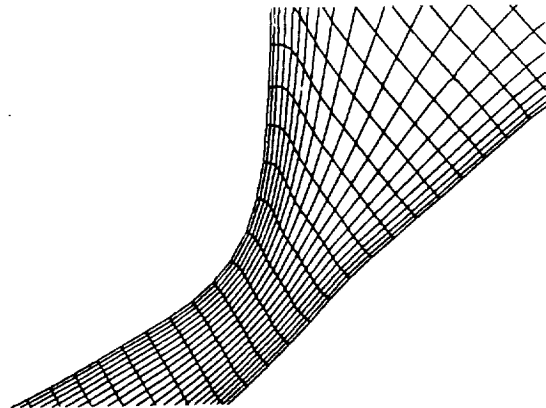
There are three main TurboI menus: (1) HOME, (2) MODIFY CONTROL NET, and (3) MODIFY VIEW. Some entries of these menus are either main menu or submenu names. Entries of the HOME menu are MODIFY CONTROL NET, MODIFY VIEW, I/O for Grid, Redefine Block, and Quit. The MODIFY CONTROL NET menu contains entries that are needed to modify the control net and the grid structures. The MODIFY VIEW menu includes entries that help viewing and visual examination of the grid and the control net (e.g., "Zoom & Move," "Change Colors," "Scroll View"). TurboI menus are listed in a separate file, which can be changed easily.



(a) Entire grid.

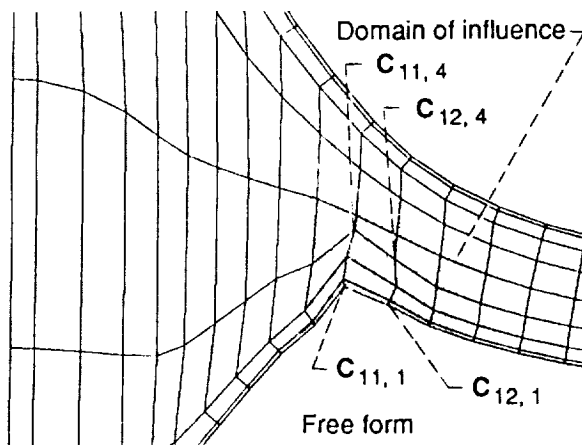


(b) Zoomed front portion.

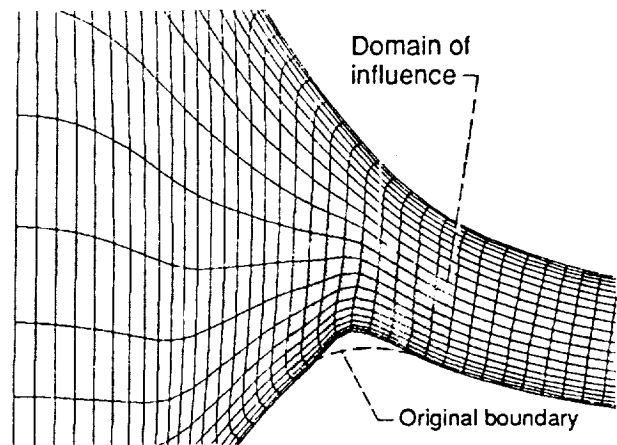


(c) Zoomed rear portion.

Figure 10.—Grid for extended orthogonality.

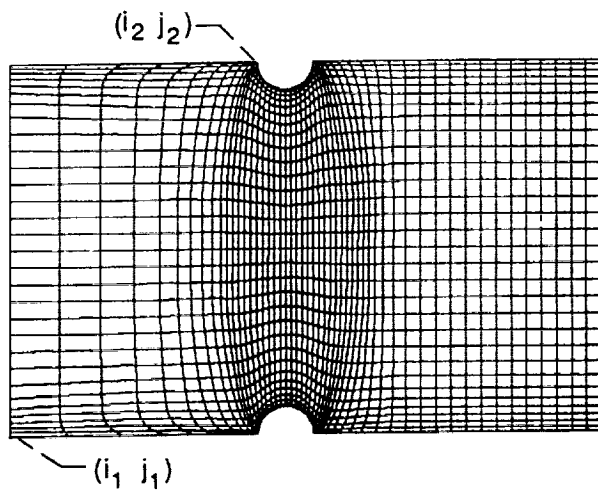


(a) New control point positions.

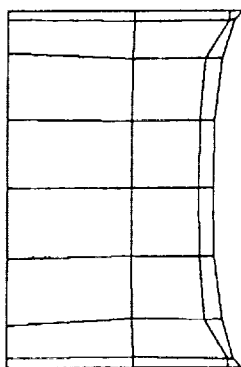


(b) Change in bottom boundary.

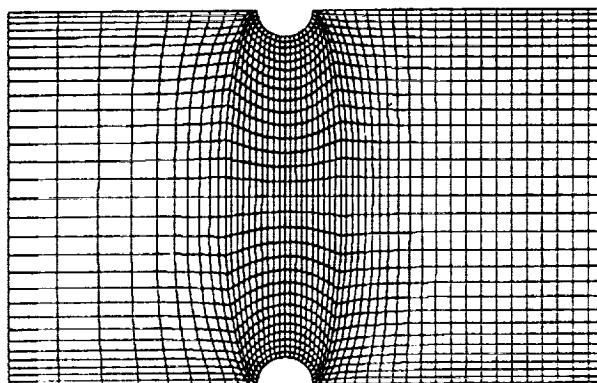
Figure 11.—Manipulation using "Free Form Boundary."



(a) Before subblock operation.



(b) Modified subblock control net.



(c) After subblock operation.

Figure 12.—Subblock operation.

### 3.6 Input and Output

Input data for TurboI can be geometry data, a boundary grid, or a field grid for attachment. TurboI can save grid and control net data in a file at any stage of an interactive session. These saved data can be used for flow simulation or can be read in for a future interactive session.

## 4. CONCLUDING REMARKS

TurboI demonstrates its interactive grid-generation and grid manipulation capability when the control point form is used. TurboI can be used as a stand-alone code as well as a complementary operating tool for explicit grid manipulation.

## 5. REFERENCES

1. Eiseman, P.R.; Snyder, A.; and Choo, Y.K.: Dynamics of Local Grid Manipulations for Internal Flow Problems. Computational Fluid Dynamics Symposium on Aeropropulsion, NASA CP-10045, 1990, pp. 30-1 to 30-26.
2. Choo, Y.K.; Soh, W.Y.; and Yoon, S.: Application of a Lower-Upper Implicit Scheme and an Interactive Grid Generation for Turbomachinery Flow Field Simulations. ASME Paper 89-GT-20, June 1989.
3. Eiseman, P.R.: Interactive Grid Generation with Control Points. Computing Systems in Engineering, vol. 1, nos. 2-4, 1990, pp. 293-304.
4. Eiseman, P.R.: A Control Point Form of Algebraic Grid Generation. Int. J. Numer. Methods Fluids, vol. 8, 1988, pp. 1165-1181.
5. Lee, K.D.; and Loellbach, J.M.: Geometry-Adaptive Surface Grid Generation Using a Parametric Projection. J. Aircraft, vol. 26, no. 2, Feb. 1989, pp. 162-167.



# Report Documentation Page

<b>1. Report No.</b> NASA TM-103748	<b>2. Government Accession No.</b>	<b>3. Recipient's Catalog No.</b>
<b>4. Title and Subtitle</b> Implementation of Control Point Form of Algebraic Grid-Generation Technique	<b>5. Report Date</b>	
	<b>6. Performing Organization Code</b>	
<b>7. Author(s)</b> Yung K. Choo, David P. Miller, and Charles J. Reno	<b>8. Performing Organization Report No.</b> E-5997	
	<b>10. Work Unit No.</b> 505-62-21	
<b>9. Performing Organization Name and Address</b> National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191	<b>11. Contract or Grant No.</b>	
	<b>13. Type of Report and Period Covered</b> Technical Memorandum	
<b>12. Sponsoring Agency Name and Address</b> National Aeronautics and Space Administration Washington, D.C. 20546-0001	<b>14. Sponsoring Agency Code</b>	
	<b>15. Supplementary Notes</b> Prepared for the Third International Conference on Numerical Grid Generation in Computational Fluid Dynamics and Related Fields cosponsored by ETS, d'Enginyers de Camins, Canals i Ports, UPC, University of Catalonia, and European Space Agency, Barcelona, Spain, June 3-7, 1991. Yung K. Choo and David P. Miller, NASA Lewis Research Center. Charles J. Reno, Cleveland State University, Cleveland, Ohio 44115 (work funded by NASA Cooperative Agreement NCC3-153). Responsible person, Yung K. Choo (216) 433-5868.	
<b>16. Abstract</b> <p>The control point form (CPF) provides explicit control of physical grid shape and grid spacing through the movement of the control points. The control point array, called a control net, is a sparse grid type arrangement of locations in physical space with an index for each direction. As an algebraic method CPF is efficient and works well with interactive computer graphics. A family of menu-driven, interactive grid-generation computer codes (TURBO) is being developed by using CPF. Key and unique features of TurboI (a TURBO member) are discussed and typical results are presented. TurboI runs on any IRIS 4D series workstation.</p>		
<b>17. Key Words (Suggested by Author(s))</b> Grid generation Computational fluid dynamics Internal flow	<b>18. Distribution Statement</b> Unclassified - Unlimited Subject Category 02	
<b>19. Security Classif. (of this report)</b> Unclassified	<b>20. Security Classif. (of this page)</b> Unclassified	<b>21. No. of pages</b> 12
		<b>22. Price*</b> A03