GRID3O – Computer Program for Fast Generation of Multilevel, Three-Dimensional Boundary-Conforming O-Type Computational Grids

Djordje S. Dulikravich

SEPTEMBER 1981
GRID3O – Computer Program for Fast Generation of Multilevel, Three-Dimensional Boundary-Conforming O-Type Computational Grids

Djordje S. Dulikravich
Lewis Research Center
Cleveland, Ohio

NASA
National Aeronautics and Space Administration
Scientific and Technical Information Branch
1981
Summary

A fast algorithm has been developed for accurately generating boundary-conforming, three-dimensional, consecutively refined computational grids applicable to arbitrary wing-body and axial turbomachinery geometries. The method is based on using an analytic function to generate two-dimensional grids on a number of coaxial axisymmetric surfaces positioned between the centerbody and the outer radial boundary. These grids are of the O-type and are characterized by quasi-orthogonality, geometric periodicity, and an adequate resolution throughout the flow field. Because the built-in nonorthogonal coordinate stretching and shearing cause the grid lines leaving the blade or wing trailing edge to end at downstream infinity, the numerical treatment of the three-dimensional trailing vortex sheets is simplified.

Introduction

When solving partial differential equations governing fluid flow, exact boundary conditions should be applied on all boundaries. This requirement is easily met with a grid generated by the computer program GRID3O, which creates a three-dimensional grid conforming to an irregularly shaped blade (or wing), hub (or fuselage), and duct (or outer radial boundary or wind tunnel wall). Such a boundary-conforming grid is transformed to a parallelepipedal $X, Y, Z$ computational space (fig. 1) in which mesh cells become adjacent cubes and a finite difference discretization of the governing fluid dynamic equations can be performed.

Three-dimensional boundary-conforming grids for cascades of blades (as well as for the single wing-body combination) can be generated by a variety of methods (ref. 1). The method used in the present work has the distinct advantage of being exceptionally fast and easy to understand. It is based on a single conformal mapping function (refs. 2 to 4) and on several additional simple analytical relations that perform coordinate stretchings and shearings. GRID3O also achieves a very high accuracy and a remarkable reliability in the discretized representation of the solid boundaries.

Applicability of the Computer Program

Computer program GRID3O is capable of generating $x, y, z$ or $r, \theta, z$ coordinates of a maximum of three consecutively refined boundary-conforming...
grids for any of the following three-dimensional configurations:

(1) Arbitrarily shaped wing midmounted on an axisymmetric fuselage: This wing-body combination can be in free air or in a wind tunnel having a doubly infinite axisymmetric wall.

(2) Propeller-type (usually referred to as "horizontal axis") windmill rotor having an arbitrary number of arbitrarily shaped blades mounted on an axisymmetric hub.

(3) Helicopter rotor.

(4) Propeller (prop-fan) for aircraft propulsion: The propeller can be in free air or in an axisymmetric wind tunnel.

(5) Axial turbomachinery stator.

(6) Axial turbomachinery rotor.

(7) Ducted fan with rotor diameter smaller than the duct diameter.

(8) Propeller for marine propulsion: The propeller can be either free or ducted.

It is important to note that both the hub (or centerbody) and the duct (or wind tunnel wall) can have arbitrary axisymmetric shapes. The number of blades is theoretically arbitrary, although GRID30 has certain limitations, which are explained in the following section. Each blade can be defined by as many as 51 different local airfoil sections. The general blade (or wing) can incorporate an arbitrary spanwise distribution of chord length, thickness, twist angle, sweep angle, and dihedral angle. GRID30 is equally applicable to blades (or wings) with blunt (or rounded), wedged, and cusped trailing and/or leading edges.

Limitations of the Computer Program GRID30

Computer program GRID30 has several minor limitations. They are the following:

(1) All geometric parameters defining the blade shape must vary smoothly in the spanwise direction.

(2) The hub and shroud are doubly infinite. If the actual hub has a finite length, it must be specified as having thin "stings" extending from the end points toward the axial infinities (fig. 2).

(3) The hub and shroud radii must vary smoothly in the direction of the axis of rotation.

(4) The blade tip chord length must have a nonzero value.

(5) The maximum number of blades is limited by the fact that the minimum local value of the gap-to-chord ratio at any spanwise station must be greater than approximately \((h/c)_{\text{min}} \geq 0.65\), where \(h\) is the spacing between the blades and \(c\) is the local blade chord length. This limiting ratio can be higher if the blade sections are highly staggered. Other factors that can increase \((h/c)_{\text{min}}\) are higher airfoil camber and relative thickness. The maximum number of blades \(NB\) can be \textit{a priori} estimated as

\[
NB \leq 2\pi \left( \frac{r}{c} \right) \left( \frac{1}{(h/c)_{\text{min}}} \right)
\]

where \(r\) is radial distance from the axis of rotation.

(6) The maximum absolute value of the local stagger angle (angle between chord line and meridional plane) for typical axial compressor blades is limited in GRID30 to \(|\beta|_{\text{max}} \leq 65^\circ\). The value of \(|\beta|_{\text{max}}\) can be considerably larger \((|\beta|_{\text{max}} \approx 90^\circ)\) in the outer portions of wind turbine blades or propellers. In an axial gas turbine blade row, \(|\beta|_{\text{max}}\) will be lower because of the low values of \((h/c)_{\text{min}}\) and the high camber and relative thickness (ref. 5).

(7) GRID30 does not allow for the existence of part-span dampers between blades.

(8) Besides blades, hub, and shroud, no other solid boundaries may be present in the flow field.

Grid-Generating Procedure

When analyzing any of the configurations mentioned in the section on applicability (except the wing-body combination), it is sufficient to consider a
Figure 3. - An axisymmetric cutting surface.

Figure 3. - An axisymmetric cutting surface.

Figure 4. - Two-dimensional axisymmetric surface $\bar{w}$.

Figure 4. - Two-dimensional axisymmetric surface $\bar{w}$.

single rotationally periodic segment of the flow field (fig. 3). This segment is a doubly infinite periodic volume stretching in the direction of the axis of rotation. The volume has a constant angular width of $360^\circ / NB$, where NB is the total number of blades. The blades have arbitrary spanwise distributions of taper, sweep, dihedral, and twist angles. The local airfoil shapes can vary in an arbitrary fashion along the blade span. The rotor hub and the duct (or shroud) can have different arbitrary axisymmetric shapes. Such an arbitrary three-dimensional physical domain is first discretized in the spanwise direction by a number of coaxial axisymmetric surfaces that are, in general, irregularly spaced between the hub and the shroud. Each of these surfaces is then discretized.

The major problem in generating and discretizing the axisymmetric surfaces is an accurate determination of the intersection contours between the irregular blade surface and the coaxial axisymmetric surfaces cutting the blade. The coordinates of the points on these contours are defined by fitting cubic splines along the blade and interpolating at the radial stations corresponding to each axisymmetric surface, $r = \text{constant}$ (fig. 2).

Once the shape of the intersection contour on a particular cutting axisymmetric surface is known, the problem becomes one of discretizing a doubly connected two-dimensional domain, $\bar{w} = x' + i r \theta'$ (fig. 4).

Each two-dimensional grid should have the following features:

(1) Grid cells should conform to the local contour shape and the shape of the periodic boundaries $\bar{a} \bar{b}$ and $\bar{c} \bar{d}$.

(2) Grid should be geometrically periodic in the $\theta''$ direction, meaning that the grid points along the periodic boundary $\bar{a} \bar{b}$ must have the same respective $x''$ coordinates (fig. 4) as the grid points along the periodic boundary $\bar{c} \bar{d}$.

(3) Grid lines should not be excessively nonorthogonal near solid boundaries.

(4) A grid line emanating from the trailing edge should end at downstream infinity.

(5) Grid cells should be concentrated in the regions of high surface curvature.

A grid with these properties can be most easily generated by using a single analytic function. One such function is (ref. 4)

$$\bar{w} = e^{i \beta} \ln \left( \frac{m - \bar{z}}{m + \bar{z}} \right)$$

$$+ e^{-i \beta} \ln \left( \frac{1 - m \bar{z}}{1 + m \bar{z}} \right)$$

where $\bar{w} = x' + i r \theta'$ and $\bar{z} = \xi + i \eta$. This complex function maps conformally a unit circle with a straight slit in the middle (fig. 5) whose end points are situated at $\bar{z} = \pm m$ onto the cascade of straight slits in the $\bar{w}$ plane. Each slit has a length $l_s$, where

$$l_s = 4 \left( \cos \beta \sinh^{-1} \frac{2m \cos \beta}{1 - m^2} + \sin \beta \sin^{-1} \frac{2m \sin \beta}{1 + m^2} \right)$$
The slits are spaced $2\pi \cos \beta$ distance apart, where $\beta$ is the stagger angle of the cascade of slits (fig. 4). The unit circle is “unwrapped” by using elliptic polar coordinates (ref. 6), resulting in a deformed rhomboidal shape that is then sheared in the horizontal and vertical directions to give a rectangular $X,Y$ computational domain.

The transformation of an actual cascade of airfoils will result in a cascade of unit circles that are deformed (fig. 5). Consequently, more nonorthogonality will be introduced in the transformation by additional shearing of coordinates (fig. 6). A uniform grid in the $X,Y$ plane that is symmetrically spaced with respect to the $Y$ axis remaps back into the physical $x^*, r\theta^*$ plane as an $O$-type boundary-conforming grid (fig. 7). The actual radial coordinates are obtained by fitting cubic splines along the elliptic grid lines and interpolating at a number of axial stations at which the radius of the corresponding axisymmetric surface is known.

A disadvantage of the present method is that it is not applicable for very thick, highly twisted blades that are very closely spaced (fig. 8). This problem can be resolved by using a different form of the mapping function, for example, one that maps a cascade of circles into a cascade of circular arcs instead of a cascade of straight slits.

**Computer Program**

**General Description**

The computer program GRID3O consists of a main program (fig. 9) and eight subroutines. Input to GRID3O is accomplished by a separate deck of cards, which will be discussed in detail in the following section. Parts of the output from GRID3O appear in standard computer printout form; the major portion of the output (representing
coordinates of each grid point) is automatically written on tapes or disks (units 11, 12, and 13).

The main program reads all input data and determines coordinates of the intersection contours created by the axisymmetric computational surfaces and the blade surface (fig. 3). This is performed by fitting cubic splines (subroutine SPLIF) along the blade span and interpolating (subroutine INTPL) at desired radial stations. The intersection contour on each axisymmetric surface is expressed in \( x'' , r \phi'' \) coordinates and iteratively mapped by using subroutine CONMAP onto a deformed unit circle (fig. 5). Auxiliary coordinate stretchings and an iterative determination of the slit length in the circle plane are performed by subroutine CENTRL.

CONMAP "unwraps" the circle with the slit in the middle by using an elliptic polar coordinate transformation. The resulting deformed rhomboidal shape is then transformed into a parallelepipedal shape by using nonorthogonal coordinate stretching and shearing in both coordinate directions. Subroutine REMAP maps points (which must be equidistantly spaced with respect to \( X = 0 \)) back into the \( x'' , r \phi'' \) space.

Figure 9. - Flow chart of GRID30 computer program.

Figure 10. - Example of a two-dimensional grid.
Coordinates of the resulting body-conforming grid (fig. 10) on each \( x'' \), \( r'' \) strip are then interpolated on their respective axisymmetric surface by using subroutine SURF.

For easier numerical application of the boundary conditions at axial infinities the grid has to be suitably shaped at the axial infinity cutoffs. This task is performed by the subroutine XYZINF.

Finally, subroutine PISMO writes \( x, y, z \) or \( x, \theta, r \) coordinates on disks or tapes.

<table>
<thead>
<tr>
<th>PENN STATE UNIVERSITY - SUBSONIC DUCTED ROTOR GEOMETRY</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>( y )</td>
<td>( z )</td>
<td>( x )</td>
<td>( y )</td>
<td>( z )</td>
</tr>
<tr>
<td>0.900000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.800000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.700000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.600000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.500000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.400000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.300000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.200000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.100000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
<tr>
<td>0.000000</td>
<td>0.350000</td>
<td>0.700000</td>
<td>0.000000</td>
<td>0.000000</td>
<td>0.000000</td>
</tr>
</tbody>
</table>

Input

The entire input to GRID30, in card form, is shown in figure 11. The first card is the title card on which the user can write as many as 80 alphanumerical characters (fig. 12) specifying the name of the input deck. This text will also appear on the output listing from GRID30.

The second card contains the following 11 input parameters:
XK number of grid cells around the blade, that is, the number of grid cells on the first (coarse) grid in the direction of the computational X axis. Suggested value is XK = 24. Users of GRID30 should remember that XK must be an even number.

KY number of grid cells in the computational Y direction, that is, the number of elliptic (wraparound) layers of grid cells around the blade on the first (coarse) grid. Suggested value is KY = 6. It is advisable to specify the value of KY from the following relation:

\[ KY = \frac{KX}{4} \]

KZ number of grid cells in the spanwise direction (computational Z direction) from the hub to the outer radial boundary (fig. 2) on the first (coarse) grid. The shape and radial distance of the outer boundary surface will be specified later in the input.

KT number of grid cells in the spanwise direction from hub to tip on the first (coarse) grid. The minimum value of KZ must be

\[ KZ \geq KT + 3 \]

KT is equal to KZ only when \( r_t \) is equal to the radius of the outer boundary, where \( r_t \) is the rotor radius. The blade tip will be in the middle of the (KT + 1)th grid cell except in the case where KT = KZ. Then the blade tip will be touching the shroud (duct) wall.

NM number of grid refinements. If NM = 1, only the coarse three-dimensional grid consisting of \((KX \times KY \times KZ)\) grid cells will be generated and its \(x,y,z\) (or \(x,\theta,r\)) coordinates permanently stored on unit 11. If NM = 2, this coarse grid will be refined by doubling the number of grid cells in each of three computational directions. Coordinates of this refined (second) grid will then be automatically written on unit 12, but the coordinates of the coarse (first) grid will remain stored on unit 11. If NM = 3 on the second input card, the second grid will also be refined in the same manner as was the first grid. The coordinates of this fine (third) grid will then be written on unit 13. Users of GRID30 should be aware of the computer storage requirements resulting from the successive grid refinement procedure (fig. 13).

The present version of GRID30 is capable of automatically generating coordinates for as many as three (NM = 3) refined grids, the third grid having a maximum of \(192 \times 48 \times 48\) grid cells. These values can be changed by changing COMMON and DIMENSION statements.
NM is given a negative sign, output from GRID30 program will be \(x, \theta, r\) coordinates.

**NH** number of input points defining hub and sting surface. The maximum value for NH is 51.

**ND** number of input points defining duct (or outer radial boundary) surface. The maximum value for ND is 51. ND must be equal to or less than NH.

**NB** number of blades. The minimum value for NB is 2.

**SA** setting (pitch) angle of the blade, deg. SA represents the constant angle that will be automatically added to the local stagger (twist) angle at each spanwise input station. When generating a sequence of grids for a variable-pitch rotor (or stator), it is necessary each time to change the value of SA only; the rest of the input should stay the same.

**DZ** length scaling coefficient. DZ multiplies all output length parameters generated by GRID30 program.

<table>
<thead>
<tr>
<th>Input length units</th>
<th>Output length units</th>
<th>DZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>meters</td>
<td>meters</td>
<td>1.000</td>
</tr>
<tr>
<td>feet</td>
<td>feet</td>
<td>1.000</td>
</tr>
<tr>
<td>feet</td>
<td>meters</td>
<td>0.3048</td>
</tr>
<tr>
<td>meters</td>
<td></td>
<td>3.2808</td>
</tr>
</tbody>
</table>

**RT** rotor radius, or wing half span.

A number of the following input cards (card 3 to card \(m\) in fig. 12) specify hub and shroud geometry. Each of these input cards lists the following values:

- \(x_h\) \(x\) coordinate of the input point on the hub or sting surface
- \(r_h\) \(r\) coordinate of the input point on the hub or sting surface
- \(x_s\) \(x\) coordinate of the input point on the outer radial boundary
- \(r_s\) \(r\) coordinate of the input point on the outer radial boundary

Hub and shroud surfaces should be defined at least 10 chord lengths upstream and downstream of the rotor.

In a turbomachinery stator, rotor, or ducted fan the outer radial boundary represents the surface of a duct or shroud. In the remaining text this surface is referred to as a shroud. In a free propeller, a helicopter rotor in hover, or a wing-body combination the outer radial boundary represents an arbitrarily axisymmetrically shaped surface positioned off the blade tip.

Blade geometry is defined on NP parallel input planes (fig. 14), where the first input plane (\(N=1\)) must be entirely inside the hub and the last input plane must be entirely at or beyond the shroud surface or outer radial boundary surface.

The following input card (card \(m+1\) in fig. 12) contains geometric parameters specifying the first spanwise input plane (\(N=1\)). These parameters are

- **XLEAD** \(x\) coordinate of the blade leading edge on the \(N^{th}\) input plane (fig. 14). The origin of the \(x, r, \theta\) coordinate system is arbitrarily positioned on the \(x\) axis. Blade stacking axis corresponds to \(z\) axis (or \(r\) axis).
- **ZLEAD** \(z\) coordinate of the blade leading edge on the \(N^{th}\) input plane (fig. 15)
- **YLEAD** \(y\) coordinate of the blade leading edge on the \(N^{th}\) input plane (fig. 14)
- **CHORD** chord length (not the projected chord length) of the blade on the \(N^{th}\) input plane (fig. 15)
- **R01,R02** radii of the blade leading and trailing edges, respectively, on the \(N^{th}\) input plane (fig. 15). R01 and R02 must be nondimensionalized with respect to CHORD.
- **TWIST** angle between the blade chord on the \(N^{th}\) input plane and the \(X\) axis (fig. 15), deg
- **NF** input parameter that indicates if a single airfoil shape was used for the blade design. If NF = 0, the airfoil shape on the
Input coordinate system on each \(N\)th input plane is orthogonal on the \(z\) (stacking) axis. The \((N+1)\)th input plane is the same as it was on the \(N\)th input plane. If \(NF = 1\), the airfoil shape on the \((N+1)\)th input plane is different (fig. 11) from the airfoil shape on the \(N\)th input plane. If \(NF = 2\), this input plane is the last input plane, and the local airfoil shape is the same as on the previous input plane. If \(NF = 3\), the input plane is the last input plane and the local airfoil shape is not the same as on the previous plane.

**MAXP**

Number of input points defining airfoil shape on \(N\)th input plane. The input points are given in the clockwise direction (fig. 15) starting from the trailing-edge point. If the airfoil shape is nonsymmetric, \(MAXP\) must be an odd number (counting the trailing-edge point twice). If the airfoil shape is symmetric, \(MAXP\) must be an even number, and only coordinates of points on the lower surface of the airfoil must be specified. The value of \(MAXP\) can vary from one \(N\)th plane to the next. Maximum number is \(MAXP = 165\).

The following set of input cards (card \(m + 2\) to card \(n\) in fig. 12) specifies the \(x'\) and \(y'\) input coordinates (fig. 15) defining the airfoil shape on the \(N\)th input plane. The origin of the \(x', y'\) coordinate system is arbitrarily positioned with respect to the airfoil leading edge (fig. 15). The input coordinates \(x'\) and \(y'\) do not have to be normalized with respect to the chord length. Scalings and rotation of the input coordinates will be performed automatically by the GRID30 program. The coordinate axis \(x'\) does not have to be parallel with the \(x\) coordinate axis, but it must be in the \(N\)th input plane (fig. 15).

All the \(x'\) coordinates are specified in the input separately, followed by all the \(y'\) coordinates (fig. 11). The number of \(x'\) (or \(y'\)) coordinates that can be specified on each input card is eight (fig. 12). If the total number of input points (\(MAXP\)) is not a multiple of eight, the last input card of a set specifying \(x'\) coordinates (as well as the last card specifying \(y'\) coordinates) should be filled with the values \(x' = 0\) (or \(y' = 0\)).

This concludes all the input data needed for the \(N\)th input plane.

The \((N+1)\)th input plane and the airfoil shape are defined on the following set of input cards, starting with card \(n + 1\) in figure 12. These cards are identically formatted as are the cards from \(m + 1\) to \(n\) (fig. 12). If \(NF = 0\) or \(NF = 2\) on card \(m + 1\), the \((N+1)\)th input plane will be defined by a single input card having the format of card \(m + 1\), as can be seen in figure 16.

### Output

Parts of the output of GRID30 will appear in printed form and require approximately two pages of computer printout (fig. 17). The main results of GRID30 are coordinates of three-dimensional body-conforming grids (or grid, if \(NM = 1\)). The coordinates of the coarse (first) grid will be automatically written on tape (or disk) 11. If \(NM = 2\) in the input data, coordinates of the second (refined) grid will be written on unit 12; and if \(NM = 3\), the coordinates of the third (fine) grid will be written on unit 13. This storage arrangement provides the user with an opportunity to display and analyze each grid separately. All the WRITE statements are unformatted and can be found in subroutine PISMO.

The output listing specifies the data contained on the first \(m\) input cards as well as each \((m+1)\)th card. This serves for easier detecting of errors that may have been made during the preparation of the input.

The output from GRID30 continues with a listing of the following parameters defining airfoil shapes on each of the \(K\) computational axisymmetric surfaces.
Because GRID3O automatically generates a layer of imaginary points inside the blade (J = MAXYP) as well as one layer of points off the periodicity (fig. 10) boundaries (J = 2), the total number of layers of points generated on each Kth axisymmetric surface is

\[ \text{MAXYP} = KY \cdot \text{NM} + 3 \]

Similarly, GRID3O generates double-valued points on the vortex sheet (I = 2 and I = MAXX) trailing from the blade trailing edge (fig. 10). GRID3O also generates two lines (I = 1 and I = MAXXP) of points off the vortex sheet. Hence the total number of lines of points generated on each Kth axisymmetric surface is

\[ \text{MAXXP} = KX \cdot \text{NM} + 3 \]

As a consequence, tape 11 will contain coordinates of \((KX + 3) \cdot (KY + 3) \cdot (KZ + 1)\) grid points, tape 12 will contain coordinates of \((KX \cdot 2 + 3) \cdot (KY \cdot 2 + 3) \cdot (KZ \cdot 2 + 1)\) grid points, and tape 13 will contain coordinates of \((KX \cdot 3 + 3) \cdot (KY \cdot 3 + 3) \cdot (KZ \cdot 3 + 1)\) grid points. They are all written on these tapes in a single-precision unformatted form.

**Lewis Research Center**
National Aeronautics and Space Administration
Cleveland, Ohio, February 12, 1981
### References


---

### Table: Grid Generation Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>XLEAD</td>
<td>0.576000E-01</td>
<td>0.650000E-01</td>
<td>0.644015E-01</td>
<td>0.640559E-01</td>
<td>0.500000E-01</td>
<td>0.500000E-01</td>
<td>0.500000E-01</td>
<td>0.500000E-01</td>
</tr>
<tr>
<td>YLEAD</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
<td>0.477714E-01</td>
</tr>
<tr>
<td>CHORD</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
<td>0.575000E-01</td>
</tr>
<tr>
<td>RO1</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
</tr>
<tr>
<td>RO2</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
</tr>
<tr>
<td>TWIST</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
<td>0.625000E-01</td>
</tr>
</tbody>
</table>

---

Figure 17 - Example of an output listing.
A fast algorithm has been developed for accurately generating boundary-conforming, three-dimensional, consecutively refined computational grids applicable to arbitrary wing-body and axial turbomachinery geometries. The method is based on using an analytic function to generate two-dimensional grids on a number of coaxial axisymmetric surfaces positioned between the centerbody and the outer radial boundary. These grids are of the O-type and are characterized by quasi-orthogonality, geometric periodicity, and an adequate resolution throughout the flow field. Because the built-in nonorthogonal coordinate stretching and shearing cause the grid lines leaving the blade or wing trailing edge to end at downstream infinity, the numerical treatment of the three-dimensional trailing vortex sheets is simplified.
211U, A, 090181 500903DS
DEPT OF THE AIR FORCE
AF WEAPONS LABORATORY
ATTN: TECHNICAL LIBRARY (SUL)
KIRTLAND AFB NM 87117