HOMAR: A Computer Code for Generating Homotopic Grids Using Algebraic Relations

Users' Manual

Anutosh Moitra
High Technology Corporation
Hampton, Virginia

Prepared for
Langley Research Center
under Contract NAS1-18240
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>II.</td>
<td>Specification Level 0</td>
<td>4</td>
</tr>
<tr>
<td>III.</td>
<td>Specification Level 1</td>
<td>6</td>
</tr>
<tr>
<td>IV.</td>
<td>Specification Level 2</td>
<td>15</td>
</tr>
<tr>
<td>V.</td>
<td>Computed Body/Grid Examples</td>
<td>27</td>
</tr>
<tr>
<td>VI.</td>
<td>References</td>
<td>39</td>
</tr>
<tr>
<td>VII.</td>
<td>Figures</td>
<td>40</td>
</tr>
<tr>
<td>VIII.</td>
<td>Appendix A. Data Dictionary</td>
<td>55</td>
</tr>
<tr>
<td>IX.</td>
<td>Listing of Program HOMAR</td>
<td>66</td>
</tr>
</tbody>
</table>
List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.a</td>
<td>Data context diagram for HOMAR</td>
<td>40</td>
</tr>
<tr>
<td>Figure 1.b</td>
<td>Control context diagram for HOMAR</td>
<td>41</td>
</tr>
<tr>
<td>Figure 2.</td>
<td>Process diagram for HOMAR</td>
<td>42</td>
</tr>
<tr>
<td>Figure 3.a</td>
<td>O-Grid index notations</td>
<td>43</td>
</tr>
<tr>
<td>Figure 3.b</td>
<td>C-Grid index notations</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4.a</td>
<td>Schematic of body geometry showing intermediate cross-section</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.b</td>
<td>Diagram of normalized initial body cross-section at $x = x_n$</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.c</td>
<td>Diagram of normalized terminal cross-section at $x = x_f$</td>
<td>45</td>
</tr>
<tr>
<td>Figure 4.d</td>
<td>Diagram of normalized grid outer boundary cross-section</td>
<td>46</td>
</tr>
<tr>
<td>Figure 4.e</td>
<td>Diagram of normalized intermediate body cross-section</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5.</td>
<td>Definition of base cross-section</td>
<td>47</td>
</tr>
<tr>
<td>Figure 6.</td>
<td>Alignment of outer boundary and body</td>
<td>47</td>
</tr>
<tr>
<td>Figure 7.</td>
<td>Notation for imposing orthogonality</td>
<td>47</td>
</tr>
<tr>
<td>Figure 8.</td>
<td>Definition of body geometry</td>
<td>48</td>
</tr>
<tr>
<td>Figure 9.</td>
<td>Sample case A: Blended wing-body geometry</td>
<td>49</td>
</tr>
<tr>
<td>Figure 10.</td>
<td>Sample case B: Blended wing-body geometry</td>
<td>50</td>
</tr>
<tr>
<td>Figure 11.</td>
<td>Sample case C.1: Grid for blended wing-body</td>
<td>51</td>
</tr>
<tr>
<td>Figure 12.</td>
<td>Sample case C.2: Grid for blended wing-body</td>
<td>51</td>
</tr>
<tr>
<td>Figure 13.</td>
<td>Sample Case D: Grid for discretely defined wing cross-section</td>
<td>52</td>
</tr>
<tr>
<td>Figure 14.</td>
<td>Sample case E: Grid for discretely defined wing section with leading wedge</td>
<td>53</td>
</tr>
<tr>
<td>Figure 15.</td>
<td>Sample case E: Enlarged view</td>
<td>53</td>
</tr>
<tr>
<td>Figure 16.</td>
<td>Sample case F: Grid for wing-body-wake combination (discretely defined)</td>
<td>54</td>
</tr>
<tr>
<td>Figure 17.</td>
<td>Sample case F: Enlarged view</td>
<td>54</td>
</tr>
</tbody>
</table>
Summary

A computer code for fast automatic generation of quasi-three-dimensional grid systems for aerospace configurations is described. The code employs a homotopic method to algebraically generate two-dimensional grids in cross-sectional planes, which are stacked to produce a three-dimensional grid system. Implementation of the algebraic equivalents of the homotopic relations for generating body geometries and grids are explained. Procedures for controlling grid orthogonality and distortion are described. Test cases with description and specification of inputs are presented in detail. The Fortran computer program and notes on implementation and use are included.
Introduction

Numerical solution of partial differential equations for problems in practically all disciplines of engineering and physics depend to a very large degree on the capability for generating a system of spatial coordinates by numerical methods. The current efforts at grid-generation generally fall in two easily discernible groups, (1) methods employing partial differential equations, and (2) algebraic methods. The algebraic methods generate grid points in space by means of interpolations and blending functions based on the given boundary data. The interpolations and blending relations are specified algebraically and consequently eliminate the need for solving systems of partial differential equations. Interesting work along this approach has been reported by Eiseman [1], Smith, et al. [2], and Erickson [3], among others. An algebraic method based on homotopic relations was proposed by Barger, et al. [4].

Algebraic procedures, by nature, provide fast means for generating grid systems for arbitrary domains. Homotopic mappings between boundaries involve relationships that lend themselves to algebraization and therefore are easily adaptable to algebraic grid-generation techniques. A detailed description of the theory underlying algebraic homotopy techniques for grid-generation has been given by the author in a companion paper [5]. The present manual will describe the structural organization and instructions for use of a computer code (HOMAR) for generating algebraic grids by a homotopic procedure. The code generates grids in two-dimensional planes for each cross-section of the body geometry. The planar grids are then connected in the third direction to produce a quasi-three-dimensional grid system for the solution domain. Such quasi-three-
dimensional grids have been demonstrated to be effective for computation of supersonic flows about complex configurations [6], [7]. The body geometry must be available as a series of cross-sectional shapes, the planes of cross-sections being normal to the body axis for wing-body combinations and chord-wise planes for isolated wings. The code provides for automatic generation of blended wing-body class of body geometries, while geometries of other classes may be input as a discrete set of point coordinates. Wing geometries are strictly to be input as a set of points defining chord-wise cross-sections. Grids for complex configurations can be generated efficiently with regard to computer resources and man hours. This makes it possible to quickly generate, view and modify grids until desirable grids are obtained, e.g., in an interactive environment. Despite the speed afforded by algebraic techniques of grid generation, one approaches them with some trepidation because of their innate inability to limit propagation of boundary discontinuities and the consequent intersection of grid lines of the same family and distortion of cells. In the present code this problem has been alleviated by providing a mechanism for redistributing the homotopy parameter based on boundary shape data. The code produces nearly orthogonal non-intersecting grid lines while preserving smoothness.

A detailed documentation of the computer program is presented in the remaining chapters of this report. Firstly, specifications for the software are presented at a series of three levels. The broad purpose of the program and explanations for the contexts of the various data and control parameters are presented in level 0. An overall view of the structural organization and interactions of the various modules appears in the level 1 specification. Detailed description of individual routines is covered in level
2. A chapter containing sample test cases and their corresponding input values is presented next. A data dictionary is provided as an appendix to assist in quick look-ups of the meaning and context of any data element. Finally, the Fortran computer code is included in its entirety.
LEVEL 0. SPECIFICATION
The program HOMAR (the name is derived from "Homotopic Algebraic Relations") is a general purpose software package for generating grids between arbitrary shaped boundaries. It is fast and simple with respect to logic and input data structure. The input geometry may be supplied as a table of point coordinates defining successive cross-sections of the body or generated within the program by analytic means. Once the input geometry is specified, outer boundaries are defined for each cross-section and the code proceeds to bridge the gap between the inner and the outer boundaries by a family of transition lines produced by a homotopic mapping between the surfaces.

Values for a set of data elements and control parameters must be supplied to the code prior to execution in order to generate grids of desired sizes and characteristics. A data context diagram for HOMAR is presented in figure 1.a. The required input data are the grid size, the input geometry, locations of the nose and the tail of blended wing/body configurations and outer boundary radii. The output is the grid system produced by the code. The control parameters fall in two classes: (1) decision parameters, and (2) control values. The decision parameters activate or deactivate certain processes that determine some aspects of the final grid. In the control context diagram in figure 1.b, the decision parameters are denoted by "?" marks. The control value parameters furnish various coefficients used in the program to control orthogonality, etc., and index values defining the portion of the grid to be generated.
LEVEL 1. SPECIFICATION
At this level of description, a very broad view of the modularity and logical structure of HOMAR will be given. The program HOMAR consists of a main program and 12 subroutines, and is written in FORTRAN. An overview of the logical structure is illustrated by the following pseudocode for HOMAR.

Pseudocode:

Begin {HOMAR}

read inputs;

if (analytic geometry case) Then
    Begin {Then}
        determine distribution of cross-sectional planes;
        determine radial distribution of points;
        determine circumferential distribution of points;

        For each cross-section do
            Begin {each cross-section}
                For each circumferential point do
                    Begin {each circumferential point}
                        generate end-shapes;
                        generate outer boundary;
                        determine shape variation parameter;
                        generate surface coordinates by blending end-shapes;

                        For each radial point do
                            Begin {each radial point}
                                determine grid variation parameter;
                                compute coefficients for orthogonal trajectories;
                                determine grid-point coordinates;
                                Write grid-point coordinates;
                            end; {each radial point}

                        end; {each circumferential point}

                    end; {each radial point}

                end; {each circumferential point}

            end; {each cross-section}

        end; {each cross-section}

    end; {Then}

end; {HOMAR}
end' (Then)

Else (discrete geometry case)

Begin (Else)
For each cross-section do
  Begin (each cross-section)
  determine radial point distribution;

  For each circumferential point do
    Begin (each circumferential point)
    read body surface coordinates;
    determine outer boundary coordinates;

  For each radial point do
    Begin (each radial point)
    determine grid variation parameter;
    compute coefficients for orthogonal trajectories;
    determine grid-point coordinates;
    write grid-point coordinates;

    end; (each radial point)

  end; (each circumferential point)

end; (each cross-section)

end; (Else)

end. (HOMAR)

The main program communicates with the subroutines through transfer of data parameters. Data flow in process HOMAR is indicated in figure 2. Subroutine names appear inside rectangles.

A more formal and detailed description of process HOMAR is now presented. A list of inputs and their descriptions will be followed by an explanation of the process.
Process HOMAR

Purpose: HOMAR generates grids between boundaries of arbitrary shapes by an algebraic homotopy procedure.

\[
\begin{align*}
\text{INDGM} & \quad \text{NX, NY, NZ} \\
\text{ISKPEX, ISKPUP, ISEC, NSEC} & \quad \text{PEX, QEX} \\
\text{INPUT} & \quad \text{XN, XF} \\
\text{IREAD} & \quad \text{IYST, IYND} \\
\text{RAD1, RAD2} & \quad \text{X, Y, Z, Table} \\
\text{OUTPUT} & \quad \text{Computed Grid Coordinates}
\end{align*}
\]

Description of Inputs:

**INDGM:** Control parameter

- \( \text{INDGM} = 0 \) Generates body alone by analytical definition.
- \( \text{INDGM} = 1 \) Generates body and grid.

**NX, NY, NZ:** Grid Size

For 0-Grids (see figure 3.a)

- \( \text{NX} \): Number of stations along body axis.
- \( \text{NY} \): Number of points in radial direction.
- \( \text{NZ} \): Number of points along circumference.

For C-Grids (see figure 3.b)

- \( \text{NX} \): Number of points along circumference.
- \( \text{NY} \): Number of points in radial direction.
- \( \text{NZ} \): Number of stations along span.

**ISKPEX:** Control parameter

- \( \text{ISKPEX} = 0 \) Skip wake/exhaust grid.
- \( \text{ISKPEX} = 1 \) Generate wake exhaust grid.
ISKPUP: Control parameter

ISKPUP = 0  Skip grids upstream of nose.
ISKPUP = 1  Generate grids upstream of nose.

ISEC: Control parameter

ISEC = 0  Generate grid for one section only.
ISEC = 1  Generate grids for all sections.

NSEC: Control value. (Ignored if ISEC = 1)

NSEC = Section number for which grid is generated.

PEX, QEX: Control values

Coefficients used in controlling orthogonality and preventing grid intersections

XN, XF: Locations of initial and terminal sections for analytic blended wing-body geometries.

IREAD: Control parameter

IREAD = 0  Analytic geometry case.
IREAD = 1  Discrete input of geometry (read geometry from table).

IYST, IYND: Control values. (Ignored if IREAD = 0)

Starting and final radial indices for region of grid to be computed.

RAD1, RAD2: Outer boundary radii for initial and terminal sections for discrete input case. Ignored if IREAD = 0.

X, Y, Z table: Table of input geometry coordinates. Each line of table contains x, y and z coordinates of a point. The points must be arranged as shown below.
Process description:

HOMAR generates the grid in planar sections by blending between the body surface curves and specified outer boundary curves by means of a homotopic procedure. In the case of body geometries input as discrete point sets, the inner boundary coordinates are read in and the outer boundary is specified by a computed set of points. In the case of analytically defined body geometries, the body surface is generated by blending between specified normalized end-shapes at the nose and the base of the configuration denoted by \([F_1(t), Y_1(t)]\) and \([F_2(t), Y_2(t)]\) respectively, where \(t\) is a parameter that varies between -1 and +1 from one wing-tip to another (see figures 4.a, 4.b and 4.c). The outer boundary is also specified as a normalized shape as shown in figure 4.d. The functions \(F_1\), \(F_2\) and \(F_3\) are defined in subroutine FNDC. A normalized shape at an intermediate
cross-section at station $X$ is computed by combining the $F_1$ and $F_2$ shapes by the following equations in subroutine SURF.

$$
\begin{align*}
RBI &= C(x)F_1 + [1 - C(x)]F_2 \\
YBI &= C(x)Y_1 + [1 - C(x)]Y_2
\end{align*} \tag{1}
$$

where $C(x)$ is a blending function that equals 1 at $X_n$ and 0 at $X_f$. $C(x)$ is denoted by $CC$ in the program and is calculated in subroutine SHAPE. The scaled cross-section at station $x$ is then determined by multiplying the coordinates ($YBI$, $RBI$) at $x$ by a scale function $SC(x)$, which is normally zero at the nose at $X_n$, and given a specified size at the base at $X_f$. $SC(x)$ is calculated in subroutine SHAPE.

At each station $X$, a set of grid lines is calculated based on a distribution of the parameter $E$. The distribution of $E$-values is computed in subroutine EGIN (EGINR in the discrete case), and the distribution of grid lines is determined by the function $SE(E)$ defined in subroutine SHAPG. $SE(E)$ must be 1 for $E = 0$, and 0 for $E = 1$. The actual scaled size of the outer grid line is defined by the factor $SCL3$. Thus the set of transition grid lines from the surface shape ($SC.YBI$, $SC.RBI$) to the outer grid shape ($SCL3.YI3$, $SCL3.F3$) is given by the coordinates

$$
\begin{align*}
Y &= SE.SC.YBI + (1 - SE).SCL3.YI3 \\
RI &= SE.SC.RBI + (1 - SE).SCL3.F3
\end{align*} \tag{2}
$$

computed in subroutine SURF (SURFR in the discrete case).

For discrete geometries, the inner boundary is defined by the geometry data read in from an input file and the outer boundary coordinates are calculated in subroutine OUTBOUN. For orthogonality, the distribution of $SE$ can be modified in subroutine COEFFS (COEFFSR for discrete geometries) prior to executing SURF or SURFR.
Thus the 3 grid parameters are X, t and E. Their distributions can be varied in the subroutines listed below:

- X in subroutine XGIN
- t in subroutine YINC
- E in subroutine EGIN/EGINR

The shape of the body can be changed by varying

- $F_1(t), YI_1(t)$ in FNDC
- $F_2(t), YI_2(t)$

and

- $CC(x)$ in SHAPE
- $SC(x)$

The shape of the grid may be changed by varying $F_3(t), YI_3(t)$ in FNDC.

For the analytic geometry case, the body geometry and the grid lines are generated by the same set of equations [equation (2)] in SURF. Setting $E = 0$ generates only the body geometry, and the grid lines can be generated by allowing $E$ to vary over all values in its range.
LEVEL 2. SPECIFICATION
XGIN - Distribution of Cross-Section Stations

PURPOSE: The XGIN module determines the distribution of X-values for intermediate cross-sectional planes lying between $X_n$ and $X_f$ for analytically generated bodies.

**INPUT**
- NP: number of x-stations
- XN: location of nose
- XF: location of base

**OUTPUT**
- Array X: axial distribution of stations

**PROCESS:** The array X is computed according to a specified distribution (linear, cosine, etc.) function. A simple linear distribution can be generated as follows:

\[
\Delta X = \frac{(X_N - X_F)}{(N - 1)}
\]

\[
X(I) = (I - 1)\Delta X + X_N
\]

where I is the index along the axial direction.
EGIN, EGINT - Distribution of Homotopy Parameter E.

PURPOSE: The EGIN module determines the distribution of E based on the total number of transition grid lines NE, in each cross-section. E has a value of 0 on the inner boundary and 1 on the outer boundary. EGINT serves the same purpose for discretely defined body geometries.

INPUT
NE: total number of grid lines in radial direction

OUTPUT
Array E: distribution of E-values

PROCESS: The array E is computed according to a specified distribution (linear, exponential, etc.) function. A simple linear distribution is given by

\[ E(K) = (K - 1)(NE - 1) \]  

where K is the index in the radial grid direction.
YINC - Distribution of T-values.

PURPOSE: The YINC module defines the distribution of the circumferential parameter T based on the total number N, of circumferential points for analytically generated body geometries.

INPUT N: number of circumferential points

OUTPUT Array T: circumferential distribution of T-values

PROCESS: For a given surface, the value of the parameter T varies between -1 and +1 from one wing-tip to another. Points on the surface are distributed according to the distribution of T. The array T is computed according to a specified distribution function defined in YINC. A simple cosine distribution that concentrates points near the wing tips is given by

\[ T(J) = -\cos\left[\frac{(J - 1)}{N - 1} \pi\right] \]  

(6)

where J is the index in the circumferential direction.
SHAPE - Shape and Scale Variation

PURPOSE: The SHAPE module determines the values of the shape-variation and scale-variation parameters at any x-station given by \( x = x_s \).

\[
\begin{align*}
XS: & \quad \text{X-value for current station} \\
XN: & \quad \text{X-value at nose} \\
XF: & \quad \text{X-value at base} \\
EM: & \quad \text{exponent for exponential shape variation} \\
EX: & \quad \text{exponent for exponential scale variation} \\
SCL1: & \quad \text{scale at nose} \\
SCL2: & \quad \text{scale at base} \\
CC: & \quad \text{shape-variation parameter} \\
SC: & \quad \text{size-scale variation parameter}
\end{align*}
\]

PROCESS: The shape variation parameter CC controls the rate at which blending takes place between the nose and base shapes. An exponential variation can be specified as follows:

\[
CC = \left( \frac{XF - XS}{XF - XN} \right)^{EM} \quad (7)
\]

The scale variation parameter SC, analogously determines the rate of change of the size-scale from the scale-value SCL1 at the nose, and that at the base, SCL2. A smooth exponential variation of the scale is generated by the following equation.

\[
\begin{align*}
RA &= \frac{XS - XN}{XF - XN} \\
SC &= (RA)^{EX}SCL2 + [1 - (RA)^{EX}]SCL1
\end{align*} \quad (8)
\]

\[
\begin{align*}
SC &= (RA)^{EX}SCL2 + [1 - (RA)^{EX}]SCL1
\end{align*} \quad (9)
\]
SHAPG - Grid Variation

PURPOSE: The SHAPG module determines the grid variation parameter SE, in order to define a grid-line of the transition family, corresponding to $E = ES$.

INPUT
- ES: current value of the homotopy parameter $E$
- EG: exponent for grid variation

OUTPUT
- SE: grid variation parameter value at $E = ES$

PROCESS: The rate of variation of the grid lines between the body surface and the outer boundary curves is controlled by SE. The exponent $EG$ is used in order to provide an exponential rate of variation and concentration of grid-lines near the body surface. SE must have a value of 1 at the inner boundary and 0 at the outer boundary, and is computed by the following formula.

$$SE = 1 - (ES)^{EG}$$  \hspace{1cm} (10)
PURPOSE: The module FNDC computes the normalized cross-section shapes at the nose and the base of analytically generated body geometries for later use in generating intermediate cross-sections by blending. A normalized outer boundary shape is also defined for gridding.

INPUT
- T: value of circumferential parameter t
- IT: flag denoting upper (IT = 1) or lower (IT = 2) surface
- Arrays F, YI

OUTPUT
- F(1), YI(1): coordinates of normalized nose shape
- F(2), YI(2): coordinates of normalized base shape
- F(3), YI(3): coordinates of normalized outer boundary

PROCESS: The normalized shape of the nose section is usually specified as an ellipse defined in terms of the circumferential parameter t.

The nose section is defined by
\[
F(1) = A \sqrt{1 - T^2} \\
YI(1) = T
\]

Where A is the eccentricity of the ellipse. The base section is specified as a central circular fuselage with the trailing edges of the wings appearing as straight lines on each side as shown in figure 5. The circumferential locations of the fuselage-trailing edge junctions are given by \( T = BK \) and \( T = -BK \) for the upper surface as well as the lower surface. For \( |T| \leq BK \), i.e., for the central body, the base shape is defined by

\[
F(2) = BK \cdot \sin\left(\frac{(T + BK)}{2 \cdot BK} \pi\right)
\]
\[ YI(2) = -BK \cdot \cos \left[ \frac{(T + BK)}{2BK} \pi \right] \] (13)

For \(|T| > BK\),

\[ F(2) = F.(T - BK) \]
\[ YI(2) = T \] (14)

where \(F\) can be specified to produce either dihedral or anhedral trailing edges.

The outer boundary shape is specified to be a circle as follows:

\[ F(3) = \sqrt{1 - T^2} \]
\[ YI(3) = T \] (15)
OUTBOUN - Outer Boundary for Discrete Geometries

PURPOSE: The OUTBOUN module generates the coordinates of the grid outer boundary by first specifying a distribution of the circumferential parameter.

IX: circumferential index for current point
NX: total number of circumferential points
INPUT RAD: radius of outer boundary
SHIFT: coordinate shift required for alignment of body and outer boundary

OUTPUT XO, YO: outer boundary coordinates

PROCESS: A distribution of the circumferential parameter TH is first specified based on the numbers of circumferential points, NX as follows.

\[ TH = \frac{(NX - 1)}{(IX - 1)} \pi \]

(16)

Using the TH-value for the current circumferential point the coordinates of a circular outer boundary with center at \( x = 0 \) are defined by

\[ XS = -\sin(TH) \cdot RAD \]
\[ YO = -\cos(TH) \cdot RAD \]

(17)

The outer boundary is next aligned with the base of the body by adding the quantity SHIFT (see figure 6) to XS, so that

\[ XO = XS + SHIFT \]

(18)
SURF - Grid Coordinates

PURPOSE: The SURF module generates the coordinates of grid lines by blending between the body surface and the outer boundary for analytically generated geometries.

SCL3: scale value at outer boundary
CC: shape variation parameter
SE: scale variation parameter
PP, QQ: grid variation parameter for current grid line

INPUT sc: scale variation parameter

OUTPUT Y, RI: coordinates of grid point

PROCESS: The homotopy parameter SE must be locally modified in order to achieve near-orthogonality of grid lines. This is accomplished through the exponents PP and QQ. The modified parameters used in blending in the F and YI directions are then given by

\[
SDF = (SE)^{QQ} \\
SDY = (SE)^{PP}
\]  

(19)

The normalized coordinates of an intermediate section are generated by blending between the nose and base shapes through the shape variation parameter CC as follows

\[
RBI = F(1).CC + F(2).(1 - CC) \\
YBI = YI(1).CC + YI(2).(1 - CC)
\]  

(20)

Finally the coordinates RI and Y of the grid point are determined by blending between the inner and outer boundary coordinates and scaling for size. RI and Y are given by

\[
RI = SDF.SC.RBI + (1 - SDF).SCL3.F(3) \\
Y = SDY.SC.YBI + (1 - SDY).SCL3.YI(3)
\]  

(21)
SURFR - Grid Coordinates for Discrete Geometries

PURPOSE: The SURFR module generates the coordinates of the grid points by blending between the body surface and the outer boundary for discrete geometries.

\[ X, Y \text{ coordinates of grid point} \]

**INPUT**
- \( \text{XI, YI: coordinates of point on body surface} \)
- \( \text{XO, YO: coordinates of point on outer boundary} \)
- \( \text{ES: homotopy parameter} \)
- \( \text{PP, QQ: coefficients for orthogonality} \)

**OUTPUT**

**PROCESS:** The grid variation parameter \( SE \) is first computed by the following formula

\[ SE = 1 = E \underline{s} \]

where \( E \underline{s} \) is a coefficient specified by the user to control grid concentration. The distribution of \( SE \) is then modified for orthogonality as follows:

\[ \text{SDX} = (SE)^{PP} \]
\[ \text{SDY} = (SE)^{QQ} \]  \hspace{1cm} (22)

Finally the coordinates \( X \) and \( Y \) of the grid point are computed by blending between the inner and outer boundary coordinates. \( X \) and \( Y \) are given by

\[ X = \text{SDX}.XI + (1 - \text{SDX}).XO \]
\[ Y = \text{SDY}.YI + (1 - \text{SDY}).YO \]  \hspace{1cm} (23)
COEFFS, COEFFSR - Coefficients for Imposing Orthogonality

PURPOSE: The COEFFS module determines the coefficients PP and QQ which are used in redistributing the homotopy parameter SE in order to achieve grid orthogonality at the boundary. These coefficients are determined based on the imposition of the orthogonality condition on the boundary data.

\[
\begin{align*}
\text{XIP1, YIP1:} & \quad \text{coordinates of i+1}^{\text{th}} \text{ point on the boundary} \\
\text{XIM1, YIM1:} & \quad \text{coordinates of i-1}^{\text{th}} \text{ point on the boundary} \\
\text{XI, YI:} & \quad \text{coordinates of i}^{\text{th}} \text{ point on the boundary} \\
\text{INPUT} & \quad \text{X0, Y0:} \quad \text{coordinates of corresponding point on the outer boundary} \\
\text{EE:} & \quad \text{value of the homotopy parameter corresponding to first grid line away from the inner boundary.} \\
\text{OUTPUT} & \quad \text{PP, QQ:} \quad \text{coefficients for orthogonality used in SURF/SURFR} \\
\end{align*}
\]

PROCESS: Orthogonality at the inner boundary point (XI, YI) is imposed by equating the dot product of the vectors \( \overline{A} \) and \( \overline{B} \), as shown in figure 7, with zero. The theoretical basis for the procedure has been described in detail by the author in reference [5]. A point (XX, YY) is first located in order to define a line passing through (XI, YI) and parallel to the line joining the i + 1\text{th} and i - 1\text{th} points on the boundary. The coefficients PP and QQ are determined such that the point \((X, Y)\) computed by equations (21) or (23) will lie on a line nearly perpendicular to the vector \( \overline{A} \). The coefficients PP and QQ are calculated differently for inner boundary segments that are nearly horizontal and nearly vertical. The local slope SLP of the inner boundary segment is determined by

\[
\text{SLP} = \frac{(YIM1 - YI)}{(XIM1 - XI)} \quad (24)
\]
and a limiting value of slope $SLIMIT$ is specified.

Of $SLP \leq SLIMIT$, i.e., for nearly horizontal segments, the coefficients $PP$ and $QQ$ are defined as follows.

$$ A = (Y0 - YI).(YY - YI) \quad \text{(25)} $$

$$ B = (X0 - XI).(XX - XI) \quad \text{(26)} $$

$$ FF = 1 + (A/B).(1 - SE) \quad \text{(27)} $$

$$ PP = \frac{\ln(FF)}{\ln(SE)} \quad \text{(28)} $$

$$ QQ = 1 \quad \text{(29)} $$

where $SE$ has the value corresponding to the grid line next to the inner boundary.

If $SLP > SLIMIT$, i.e., for nearly vertical segment, $PP$ and $QQ$ have the following definitions.

$$ A = (XI - X0).(XX - XI) \quad \text{(30)} $$

$$ B = (YI - Y0).(YY - YI) \quad \text{(31)} $$

$$ FF = 1 + (A/B).(1 - SE) \quad \text{(32)} $$

$$ QQ = \frac{\ln(FF)}{\ln(SE)} \quad \text{(33)} $$

$$ PP = 1 \quad \text{(34)} $$

where $SE$ has the value of the homotopy parameter corresponding to the grid line next to the inner boundary.
COMPUTED BODY/GRID EXAMPLES
Blended Wing/Body Geometries

Two sample cases of blended wing/body geometries generated by the present method are presented. The examples illustrate the ability of the method to produce body geometries of varying complexity from fixed end shapes by specifying different shape and size variation functions. As explained in reference [5], the body surface is generated by blending between specified end-shapes. The process is illustrated schematically in figure 8, where the specified sectional shapes are S1 and S2 at the nose and the base of the configuration respectively. The intermediate sections S3, S4 and S5 are generated by blending between S1 and S2. The following two test cases labelled sample case A and B are examples of two vastly different body geometries generated from the end-shapes presented in figure 8. The distribution of the circumferential parameter t and the normalized end-shape coordinates have the following definitions in both cases.

\[ T(J) = \cos\left(\frac{J-1}{N-1}\pi\right) \]  \hspace{1cm} (35)

End-shapes (specified in subroutine FNDC):

\[ S1 - \]
\[ F(1) = 0.33\sqrt{1 - T^2} \]
\[ YI(1) = T \] \hspace{1cm} (36)

\[ S2 - \]
\[ F(2) = BK \cdot \sin\left(\frac{(T + BK)}{2 \cdot BK} \pi\right) \quad \text{if} \quad |T| \leq BK \]
\[ YI(2) = -BK \cdot \cos\left(\frac{(T + BK)}{2 \cdot BK} \pi\right) \] \hspace{1cm} (37)

\[ F(2) = -0.01(T - BK) \quad \text{if} \quad |T| > BK \]
\[ YI(2) = T \] \hspace{1cm} (38)

where \( BK = 0.674 \).
SAMPLE CASE A: Blended Wing-Body Geometry

INPUT

Parameters in input file -

INDGM
  0
  NX  NY  NZ
  27  25  35
SKPX SKPU ISEC NSEC
  0  0  1  25
PEX  QEX
  0.2  0.1
XN  XF
  0.0  4.0
  SCL1  SCL2  SCL3  EG  EM  EX
   0.00001  1.40000  1.00000  1.20000  1.00000  0.80000
IREAD
  0

Specified Functions -

Shape variation function CC (specified in SHAPE)

\[ CC = \left( \frac{(XF - XS)}{(XF - XN)} \right)^{EM} \]  \hspace{1cm} (39)

Size scale variation function SC (specified in SHAPE)

\[ RA = \frac{(XS - XN)}{(XF - XN)} \]
\[ SC = RA^{EX}.SCL2 + (1 - RA^{EX}).SCL1 \] \hspace{1cm} (40)

OUTPUT

The blended wing-body surface generated from the above input is presented in figure 9. Smooth variation of cross-sectional shapes and sizes along the axial direction is apparent.
SAMPLE CASE B: Blended Wing-Body Geometry

INPUT

The parameter values in the input file are identical to those in sample case A. Only the definition of the size variation function is modified as shown below.

\[ QT = \left[ \frac{(SCL2 - SCL1)}{(XF - XN)} \right]^{1.3} \times (XS - XN) \]  
\[ X3 = \frac{(XF - XN)}{3} \]  
\[ SC = QT + 0.15 \sin \left[ \frac{(XS - XN)}{X3} \pi \right] \]

OUTPUT

The modified body geometry with a more complex planform than that of sample case A, is presented in figure 10. The changed size variation function is seen to have resulted in a different outline of the geometry while maintaining the smoothness of variation of shape and size of cross-sections.

Grid examples

A set of computed planar grids for sections of various body geometries are presented next. The body geometries considered belong to both analytically defined and discretely input classes. Relevant input data sets and definitions of blending functions used to generate the example grids are given in each case.
SAMPLE CASE C: Grids for Blended Wing-Body Configurations

DESCRIPTION

Two planar grids for the blended wing-body configuration described in sample case A are presented. The grids are labelled C.1 and C.2 and represent sectional grids at the 25th and 35th stations along the x-axis respectively.

INPUT (Grid C.1)

Parameters in input file -

INDGM
  1
  NX  NY  NZ
  43  25  35
SKPX SKPU ISEC NSEC
  0  0  0  25
PEX  QEX
  0.2  0.3
XN  XF
  0.0  4.0
SCL1  SCL2  SCL3  EG  EM  EX
  .00001  1.40000  1.00000  1.20000  1.00000  0.80000
IREAD
  0
IYST IYND
  1  25
  RAD1  RAD2
  1000.00  500.00

Note: ISEC has been set zero in order to generate an individual sectional grid corresponding to NSEC = 25. The value zero for IREAD signifies that the body geometry is to be generated analytically and the parameters following IREAD are ignored by the code.
Specified Functions -

The shape and size-scale variation functions are given definitions identical with those specified in sample case A for generating the body geometry. In addition to these a grid variation function $SE$ is specified in subroutine SHAPG as given below.

$$SE = 1 - ES^G$$  \hspace{1cm} (44)

OUTPUT (Grid C.1)

The resulting grid in a cross-sectional plane at axial station corresponding to $NSEC = 25$ is presented in figure 11. The cross-section of the body geometry demonstrates the smooth blending of the wings and the fuselage at an intermediate stage in the development of the wings. The smoothness of the transition of the grid lines from the inner to the outer boundary is clearly evident, and the grid trajectories are seen to intersect the inner boundary with near orthogonal at all points.

INPUT (Grid C.2)

All inputs for C.2 are identical to those for grid C.2 except that $NSEC$ is set equal to 35.

OUTPUT (Grid C.2)

The output grid is shown in figure 12. The grid is seen to have retained its smoothness and near orthogonality despite the appreciable concavity of the wing-body function region. Grid intersections, normally resulting from the use of algebraic techniques in concave regions are noted to have been successfully avoided.
SAMPLE CASE D: Grid for Discretely Defined Wing Cross-Section

DESCRIPTION

A planar grid for a cross-section of a supersonic wing is presented. This is an example of a C-type grid in a plane located at a station along the span of the wing.

INPUT

Parameters in input file -

INDGM
1
NX  NY  NZ
115  33  25
SKPX SKPU ISEC NSEC
0  0  0  20
PEX  QEX
0.2  0.1
XN  XF
0.0  4.0
SCL1  SCL2  SCL3  EG  EM  EX
.00001  1.40000  1.00000  1.20000  1.00000  0.80000
IREAD
1
IYST IYND
1  8
    RAD1  RAD2
800.00  400.00

Note: IREAD has been set equal to 1 signifying that the wing geometry is discretely defined and the coordinates are to be read from an input file. IYST and IYND have been set equal to 1 and 8 respectively in order to present the region near the wing surface with clarity. The radii of the outer boundaries at the root chord and the tip are specified by setting RAD1 and RAD2 to 800 and 400 respectively.
Specified Functions -

The shape and size-scale functions are irrelevant for a discretely defined geometry and are ignored. The grid variation function SE is given the same definition as in equation (44).

OUTPUT:

The resulting planar grid in a chord-wise plane is shown in figure 13. This plane is located at station 20 along the span of the wing as seen from the value of NSEC in the input stream. The smoothness of grid lines and near orthogonality at the boundary are comparable to those resulting in the case of analytically defined geometries.
SAMPLE CASE E: Grid for a Discretely Defined Wing Cross-Section with a Leading Wedge

DESCRIPTION

A C-type planar grid for a supersonic wing with a wedge-shaped leading edge is presented. This case demonstrates the ability of the method to handle sharp leading edges.

INPUT

Parameters in input file -

INDGM

1
NX NY NZ
49 21 13
SKPX SKPU ISEC NSEC
0 0 1 2
PEX QEX
0.2 0.1
XN XF
0.0 4.0
SCL1 SCL2 SCL3 EG EM EX
.00001 1.40000 1.00000 1.20000 1.00000 0.80000
IREAD
1
IYST IYND
1 21
RAD1 RAD2
2.50 1.50

Note: IREAD has been given the value 1 for reading the discretely defined geometry. There are 49 circumferential points in the wing cross-section as seen from the value of NX. IYST and IYND have been set equal to 1 and 21 respectively resulting in 21 grid lines between the inner and the outer boundaries including the boundary lines. The wing is tapered and the
outer boundary radii are 2.5 and 1.5 at the root and the tip of the wing respectively.

**Specified Functions -**

No shape or size-scaling functions need be specified for this discrete case. The grid variation function $SE$ is specified in equation (44).

**OUTPUT:**

The planar grid resulting from the given input appears in figure 14. Although the total number of grid lines were 21, only 15 are plotted for clarity. Smooth blending between the boundaries is noted even in the presence of corners and a sharp leading edge. An enlarged view of the region near the inner boundary is presented in figure 15. Near orthogonality is seen to be maintained near the leading edge.
SAMPLE CASE F: Grid for Wing-Body-Wake Combination (Discretely Defined)

DESCRIPTION

A planar grid is presented for a section of an aircraft with a highly swept wing. The section, stationed in the aft part of the aircraft, contains the fuselage, the outboard part of the wing and a wake line connecting the two. The sectional geometries are discretely defined.

INPUT

Parameters in input file -

INDGM

1

NX NY NZ

57 15 38

SKPX SKPU ISEC NSEC

0 0 0 2

PEX QEX

0.2 0.1

XN XF

0.0 4.0

SCL1 SCL2 SCL3 EG EM EX

.00001 1.40000 1.00000 1.20000 1.00000 0.80000

IREAD

1

IYST IYND

1 15

RAD1 RAD2

35.00 50.00

Note: there are 57 circumferential points defining the sectional geometry including the wake line. IREAD is set to 1 for reading the discrete geometry.
Specified Functions -

Only the grid variation function SE needs be specified in the discrete case. SE is defined as in equation (44).

OUTPUT:

The final grid is shown in figure 16. Smoothness of the grid is apparent. An enlarged view of the portion of the grid containing the wake line and the wing section is presented in figure 17. The grid lines are seen to be smoothly continuous across the wake line.
References


Figure 1.a Data context diagram for HOMAR
Figure 1.b  Control context diagram for HOMAR
Input data

Number of cross-sections

XGIN

Number of circumferential points

Number of radial points

EGIN/EGINR

Distribution of homotopy parameters

Distribution of circumferential points

YINC

Distribution of cross-sections

Discrete body geometry

Point distribution data

FND

SHAPE

Generate end-shapes and shape variation

OUTBOUN

Body surface/outer boundary definition data

Outer boundary coordinates

Body surface/outer boundary definition data

COEFFS/COEFFSR

Coeffs for orthogonality

SHAPG

Grid variation parameters

Grid coordinates

Figure 2. Process diagram for HOMAR
Figure 3.a  O-Grid index notations
Figure 3.b  C-Grid index notations
Figure 4.a Schematic of body geometry showing intermediate cross-section

Figure 4.b Diagram of normalized initial body cross-section at $x = x_n$

Figure 4.c Diagram of normalized terminal cross-section at $x = x_f$
Figure 4.d Diagram of normalized grid outer boundary cross-section

Figure 4.e Diagram of normalized intermediate body cross-section
Figure 5. Definition of base cross-section

Alignment of outer boundary and body

Figure 6.

(XS, YO)  (XO = XS + Shift, YO)

(X, Y)

(X, Y)

Figure 7. Notation for imposing orthogonality
Figure 8. Definition of body geometry
Figure 9. Sample case A: Blended wing-body geometry
Figure 10. Sample case B: Blended wing-body geometry
Figure 11. Sample case C.1: Grid for blended wing-body

Figure 12. Sample case C.2: Grid for blended wing-body
Figure 13. Sample Case D: Grid for discretely defined wing cross-section
Figure 14. Sample case E: Grid for discretely defined wing section with leading wedge

Figure 15. Sample case E: Enlarged view
Figure 16. Sample case F: Grid for wing-body-wake combination (discretely defined)

Figure 17. Sample case F: Enlarged view
APPENDIX A

DATA DICTIONARY
DATA ELEMENT DESCRIPTIONS

The following template has been constructed for defining the data elements references in this specification.

NAME:
DESCRIPTION:
USED IN:
RANGE:
DATA TYPE:
ATTRIBUTE:
DATA STORE LOCATION:

The NAME field gives the name of the variable used in the specification.

The DESCRIPTION field gives a brief description of the variable.

The USED IN field provides a reference to which modules used this variable.

The RANGE field specifies the permissible range of data values for the variable.

The DATA TYPE field specifies the data type to be used in declaring the variable.

The ATTRIBUTE field indicates whether or not the variable contains data, control information, or a data condition.

The DATA STORE LOCATION field references the common region in which the variable must be stored.
NAME: A
DESCRIPTION: Eccentricity of ellipses defining end sections
USED IN: FNDC
RANGE: [+0.3, +1.0]
DATA TYPE: Array (1..3) or real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: CC
DESCRIPTION: Parameter controlling variation from nose to base shape
USED IN: SHAPE, SURF
RANGE: Determined in code
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: E
DESCRIPTION: Distribution of grid parameter
USED IN: EGIN, EGINR, SURF, SURFR
RANGE: [0.0, 1.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: EG
DESCRIPTION: Exponent for grid variation
USED IN: SHAPG
RANGE: [0.5, 2.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP
NAME: EM
DESCRIPTION: Exponent for shape variation from nose to base
USED IN: SHAPE
RANGE: [0.5, 1.5]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: EX
DESCRIPTION: Exponent for exponential scale variation from nose to base
USED IN: SHAPE
RANGE: [0.5, 1.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: F
DESCRIPTION: Coordinates of normalized nose, base and outer boundary shapes (see fig. 4)
USED IN: FNDC, SURF
RANGE: [0.0, 1.0]
DATA TYPE: Array [1..3] of real
ATTRIBUTE: Data
DATA STORE LOCATION: DEF

NAME: INDGM
DESCRIPTION: Flag indicating whether body alone or body and grid are generated
USED IN: HOMAR
RANGE: [0, 1]
DATA TYPE: Integer
ATTRIBUTE: Control variable
DATA STORE LOCATION: None
NAME: IREAD
DESCRIPTION: Flag indicating analytic or discrete geometry input
(0: analytic, 1: discrete)
USED IN: HOMAR
RANGE: [0, 1]
DATA TYPE: Integer
ATTRIBUTE: Control variable
DATA STORE LOCATION: None

NAME: ISEC
DESCRIPTION: Flag indicating whether one or all sections are to be
generated (0: one, 1: all)
USED IN: HOMAR
RANGE: [0, 1]
DATA TYPE: Integer
ATTRIBUTE: Control variable
DATA STORE LOCATION: None

NAME: ISKPEX
DESCRIPTION: Flag indicating whether wake/exhaust grid is generated
or skipped (0: skip, 1: generate)
USED IN: HOMAR
RANGE: [0, 1]
DATA TYPE: Integer
ATTRIBUTE: Control variable
DATA STORE LOCATION: None

NAME: ISKPUP
DESCRIPTION: Flag indicating whether grids upstream of nose are
generated or skipped (0: skip, 1: generate)
USED IN: HOMAR
RANGE: [0, 1]
DATA TYPE: Integer
ATTRIBUTE: Control variable
DATA STORE LOCATION: None
NAME: IYND
DESCRIPTION: Final index for circumferential lines to be generated
USED IN: HOMAR
RANGE: [1, 100]
DATA TYPE: Integer
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: IYST
DESCRIPTION: Starting index for circumferential lines to be generated
USED IN: HOMAR
RANGE: [1, 100]
DATA TYPE: Integer
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: NN
DESCRIPTION: Number of planes ahead of nose to be computed
USED IN: HOMAR
RANGE: [1, 100]
DATA TYPE: Integer
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: NSEC
DESCRIPTION: Section number for sectional grid (used with ISEC = 0)
USED IN: HOMAR
RANGE: [1, 100]
DATA TYPE: Integer
ATTRIBUTE: Data
DATA STORE LOCATION: None
NAME: NX, NY, NZ
DESCRIPTION: For 0-grid
- **NX:** Number of points along axis
- **NY:** Number of points in radial direction
- **NZ:** Number of points along circumference

For C-grid
- **NX:** Number of points along circumference
- **NY:** Number of points in radial direction
- **NZ:** Number of stations along span

USED IN: HOMAR
RANGE: [1, 100]
DATA TYPE: Integer
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: PEX
DESCRIPTION: Exponent of PP, used in preventing grid intersection
USED IN: HOMAR
RANGE: [0.05, 0.5]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: PP
DESCRIPTION: Coefficient for orthogonality
USED IN: COEFS, COEFFSR, SURF, SURFR
RANGE: Determined in code
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None
NAME: QEX
DESCRIPTION: Exponent of QQ, used in preventing grid intersection
USED IN: HOMAR
RANGE: [0.05, 0.5]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: QQ
DESCRIPTION: Coefficient for orthogonality
USED IN: COEFFS, COEFFSR, SURF, SURFR
RANGE: Determined in code
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: RAD1
DESCRIPTION: Outer boundary radius at root chord
USED IN: HOMAR
RANGE: As needed
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: RAD2
DESCRIPTION: Outer boundary radius at wing tip
USED IN: HOMAR
RANGE: As needed
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None
NAME: RI
DESCRIPTION: Coordinate of final scaled grid in the direction of F
USED IN: SURF
RANGE: Determined in code
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: SC
DESCRIPTION: Scale variation parameter for body geometry
USED IN: SHAPE, SURF
RANGE: Determined in code
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: SCL1
DESCRIPTION: Size scale at nose for body geometry.
USED IN: SHAPE
RANGE: [0.0, 0.01]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: SCL2
DESCRIPTION: Size scale at base for body geometry
USED IN: SHAPE
RANGE: [1.0, 5.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP
NAME: SCL3
DESCRIPTION: Size scale for outer boundary
USED IN: SURF
RANGE: [1.0, 10.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: SE
DESCRIPTION: Grid variation parameter
USED IN: SHAPG, SURF
RANGE: [0.0, 1.0]
DATA TYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: TH
DESCRIPTION: Circumferential parameter
USED IN: YINC, FNDC
RANGE: [0.0, 1.0]
DATA TYPE: Array [1..100] of real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: X
DESCRIPTION: Axial distribution of stations
USED IN: XGIN
RANGE: [-10.0, 10.0]
DATA TYPE: Array [1..100] of real
ATTRIBUTE: Data
DATA STORE LOCATION: None
NAME: XF
DESCRIPTION: Location of base of body geometry
USED IN: XGIN
RANGE: [1.0, 10.0]
DATATYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: XN
DESCRIPTION: Location of nose of body geometry
USED IN: XGIN
RANGE: [0.0, 1.0]
DATATYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: BOOKP

NAME: Y
DESCRIPTION: Final scaled coordinate of grid point in the direction of YI
USED IN: SURF
RANGE: Determined in code
DATATYPE: Real
ATTRIBUTE: Data
DATA STORE LOCATION: None

NAME: YI
DESCRIPTION: Coordinates of normalized nose, base and outer boundary shapes (see figure 4)
USED IN: FNDC, SURF
RANGE: [0.0, 1.0]
DATATYPE: Array [1..3] of real
ATTRIBUTE: Data
DATA STORE LOCATION: DEF
PROGRAM HOMAR(INPUT, OUTPUT, TAPE5, TAPE10, TAPE12)

**

* THIS PROGRAM GENERATES QUASI-THREE-DIMENSIONAL GRIDS *
* WITH ORTHOGONALITY AT THE INNER BOUNDARY BY AN ALGEBRAIC *
* HOMOTOPY PROCEDURE. *
* PROGRAM DEVELOPED BY ANUTOSH MOITRA *
* DEVELOPMENT SPONSORED BY NASA, LANGLEY RESEARCH CENTER. *
* FOR INQUIRIES CONTACT ANUTOSH MOITRA *
* CAMSTAFF CORP 1987 *
* NASA, LANGLEY RESEARCH CENTER *
* HAMPTON, VIRGINIA 23665 *

**

COMMON/BOOKP/EM, EX, EG, ECL3, SCL1, SCL2, XN, XF, CC, SC, SE
DIMENSION X(101), TH(101), A(4,4), C(4,4), E(101)

**

SURFACE GEOMETRY MAY BE GENERATED ANALYTICALLY
OR READ FROM INPUT FILE (TAPE 10).
RUN PARAMETERS ARE READ FROM INPUT FILE (TAPE 5 = CRDDAT).
GRID OUTPUT IS WRITTEN ON TAPE 12

**

READ(5,500)
READ(5,510) INDGM
C***** INDGM = 0: GENERATES BODY ONLY
C INDGM = 1: GENERATES BODY & GRID
READ(5,500)
READ(5,510) NX, NY, NZ
C***** FOR O-GRIDS
C NX: # OF POINTS ALONG THE AXIS
C NY: # OF POINTS IN RADIAL DIRECTION
C NZ: # OF POINTS ALONG CIRCUMFERENCE
C***** FOR C-GRIDS
C NX: # OF POINTS ALONG CIRCUMFERENCE
C NY: # OF POINTS IN RADIAL DIRECTION
C NZ: # OF STATIONS ALONG SPAN
NP = NX
NE= NY
N= NZ
READ(5,500)
READ(5,510) ISKPEX, ISKPUP, ISEC, NSEC
C***** ISKPEX = 0: SKIP WAKE-EXHAUST GRID
C ISKPEX = 1: COMPUTE WAKE-EXHAUST GRID
C***** ISKPUP = 0: SKIP GRIDS UPSTREAM OF NOSE
C ISKPUP = 1: COMPUTE GRIDS UPSTREAM OF NOSE
C***** ISEC = 0: GENERATE SECTIONAL GRID ONLY
C ISEC = 1: GENERATE GRIDS FOR ALL SECTIONS
C***** NSEC: SECTION NUMBER ON BODY FOR WHICH
C GRID IS GENERATED (USED WITH ISEC = 0)
READ(5,500)
READ(5,520) PEX, QEX
C***** PEX, QEX: EXPONENTS OF P & Q, USED FOR PREVENTING
C GRID INTERSECTIONS
READ(5,500)
READ(5,520) XN, XF
C***** XN, XF: INITIAL AND FINAL X STATION VALUES
C IGNORED IN DISCRETE INPUT CASE
READ(5,500)
READ(5,515) SCL1, SCL2, SCL3, EG, EM, EX
READ(5,500)
READ(5,510) IREAD
C***** IREAD = 0: ANALYTIC INPUT CASE
C IREAD = 1: DISCRETE INPUT CASE (READ GEOMETRY FROM TAPE10)
C IF(IREAD.EQ.1) GO TO 300
C ANAL
C CROSS SECTIONS
CALL XGIN(NP, X, DX)
CALL EGIN(NE,E)
CALL YINC(TH,N)
CALL SHAPE(X(NP))
CALL SHAPG(E(1))
NH=(N+1)/2
110 FORMAT(3110)

C ***** COMPUTE GRIDS UPSTREAM OF THE NOSE

C ***** NN = NO OF GRIDS AHEAD OF THE NOSE
7

IF(ISKPUP.EQ.0) GO TO 71
DO 11 IR=1,NN
RVRS=FLOAT(NN-IR+1)
C
XX=X(I1)-(1.-COS((3.1415926/2.)/FLOAT(NN)*RVRS))**2.
C
XX=X(I1)-((RVRS/FLOAT(NN))**1.5)
C
XX=X(I1)-RVRS*DX
CALL SHAPE(X(I1))
DO 12 J=NH,N
JI=J
IF(IT.EQ.2) JI=N-(J-NH)
CALL FNDC(I,NP,TH(JI),IT)
DO 13 K=1,NE
CALL SHAPG(E(K))
CALL SURF(I,NP,N,XX,TH(JI),E(K),Y,RI,0.08,1.0)
II=0
IF(J.I.EQ.1) II=1
IF(K.I.EQ.1.AND.J.I.EQ.1) II=2
IF(K.I.EQ.1) Y=0.
IF(K.I.EQ.1) RI=0.
PRINT 103,XX,Y,RI,II,II
WRITE(12,105) XX, Y,RI
CONTINUE
11 CONTINUE
CONTINUE
CONTINUE

C * COMPUTE SURFACE COORDINATES *
71

ISTRT=1
IEND=NP
IF(TSEC.EQ.0) ISTRT=NSEC
IF(TSEC.EQ.0) IEND=NSEC
DO 1 I=ISTRT,IEND
IF(INDGM.EQ.0) NE=1
CALL SHAPE(X(I))
NNN=NH+3
DO 3 J=NH,N
JI=J
IF(IT.EQ.2) JI=N-(J-NH)
CALL FNDC(I,NP,TH(JI),IT)
DO 2 K=1,NE
IF(JI.EQ.N) GO TO 200
IF(JI.EQ.N) GO TO 214
JIM1=JI-1
JIP1=JI+1
CALL SHAPG(E(1))
CALL FNDC(I,NP,TH(JIM1),IT)
CALL SURF(I,NP,N,X(I),TH(JIM1),SE,YIM1,RIM1,0.08,1.0)
CALL FNDC(I,NP,TH(JIP1),IT)
CALL SURF(I,NP,N,X(I),TH(JIP1),SE,YIP1,RIP1,0.08,1.0)
CALL FNDC(I,NP,TH(JI),IT)
CALL SURF(I,NP,N,X(I),TH(JI),SE,YIN,RIN,0.08,1.0)
CALL SHAPG(E(NE))
CALL SURF(I,NP,N,X(I),TH(JI),SE,YO,RO,0.08,1.0)
CALL SHAPG(E(2))
IF(INDGM.EQ.0) GO TO 200
CALL COEFS(YIP1,RIP1,YIM1,RIM1,YO,RO,SE,FP,QQ,YIN,RIN)

67
PSAV=PP
QSAV=QQ
GO TO 210

200 PP=0.05
    QQ=0.3
    GO TO 210
214 PP=PSAV
    QQ=QSAV
210 CALL FNDC(I,NP,TH(JI),IT)
    IF(K.EQ.1) GO TO 211
    FP=(1.0-(FLOAT(K-2)/FLOAT(NE-2))**PEX)
    FQ=(1.0-(FLOAT(K-2)/FLOAT(NE-2))**QEX)
    GO TO 212
211 FP=1.0
    FQ=1.0
212 PP=PP*FP
    QQ=QQ**FQ
    CALL SHAPE(E(K))
    CALL SURF(I,NP,N,X(I),TH(JI),SE,Y,RI,PP,QQ)
    II=0
    IF(INSTD.EQ.0) GO TO 85
    IF(K.EQ.1) II=1
    IF(K.EQ.1.AND.J.EQ.NH) II=2
    GO TO 80
85 IF(J.EQ.NH) II=1
    IF(J.EQ.NH.AND.I.EQ.ISTRT) II=2
80 IF(I.EQ.1.AND.K.EQ.1) Y=0.
    IF(I.EQ.1.AND.K.EQ.1) RI=0.
    PRINT 103,X(I),Y,RI,II,12
    WRITE(12,105) X(I),Y,RI
    2 CONTINUE
3 CONTINUE
1 CONTINUE
    IF(ISKPEX.EQ.0) GO TO 73
C ******** COMPUTE GRIDS AFT OF THE TRAILING EDGE
C ******** NM = NO OF GRIDS BEHIND TRAILING EDGE
C ******** NI = NO OF ETA- LINES INSIDE TE CIRCLE
    NI=6
    DR=0.2
    NM=4
    DO 21 IM=1,NM
        XX=X(NP)+.1*FLOAT(IM-1)
        CALL SHAPE(X(NP))
    C GO TO 72
C ******** COMPUTE GRIDS WITHIN TRAILING EDGE CIRCLE
    CALL SHAPE(E(1))
    DO 25 KK=1,NI
        DO 24 JJ=JEQ1,JEQ2
            CALL FNDC(NP,NP,TH(JJ),IT)
            CALL SURF(NP,NP,N,X(NP),TH(JJ),E(KK),Y,RI,0.08,1.0)
            RI=RI*(1.-FLOAT(KK-1)*DR)
            II=0
            IF(JJ.EQ.JEQ1) II=1
            IF(KK.EQ.1.AND.JJ.EQ.JEQ1) II=2
            PRINT 103,XX,Y,RI,II,12
        24 CONTINUE
    25 CONTINUE
72 DO 23 K=1,NE
    CALL SHAPE(E(K))
    DO 22 J=1,N
        JJ=J
        IF(JT.EQ.2) JJ=N-(J-1)
        CALL FNDC(NP,NP,TH(JJ),IT)
        CALL SURF(NP,NP,N,X(NP),TH(JJ),E(K),Y,RI,0.08,1.0)
        II=0
        IF(J.EQ.1) II=1
IF(K.EQ.1.AND.J.EQ.1) I1=2
C IF(K.EQ.1) RI=0.
PRINT 103,XX,Y,RI,I1,I2
C WRITE (12,105) XX,Y,RI
22 CONTINUE
23 CONTINUE
21 CONTINUE
73 IF(IT.EQ.2)GO TO 20
IT=2
GO TO 7
105 FORMAT (3E20.10)
500 FORMAT (1X)
510 FORMAT (515)
515 FORMAT (6F10.6)
520 FORMAT (5F10.6)
C****** COMPUTATION FOR DISCRETE INPUT CASE.
C****** 300 READ (5,500)
READ (5,510) IYST,IYND
C****** IYST,IYND: STARTING AND FINAL INDICES OF CIRCUMFERENTIAL
C LINES TO BE COMPUTED
READ(5,500)
READ(5,520) RAD1,RAD2
C****** RAD1,RAD2: OUTER BOUNDARY RADIi AT WING ROOT AND TIP
NLE=(NX+1)/2
C ISEC= 0----GENERATE SECTIONAL GRID ONLY
C ISEC= 1----GENERATE GRID FOR ALL SECTIONS
C NSEC: SECTION NUMBER FOR WHICH GRID IS GENERATED
I2=1
IF(ISEC.EQ.1) GO TO 333
NSECM1=NSEC-1
DO 310 I2=1,NSECM1
READ (10,331) Z
DO 311 IX=1,NX
READ (10,332) XS,YS
CONTINUE
311 READ(10,332) XS,YS
310 CONTINUE
333 NEND=NZ
IF(ISEC.EQ.0) NEND=1
DO 320 I2=1,NEND
IF(ISEC.EQ.0) N1=NSEC
CALL EGINR(NY,E)
READ(10,331) Z
331 FORMAT (1X,E14.7)
PI=4.0*ATAN(1.0)
CONV=180.0/PI
DLR=(RAD2-RAD1)/(FLOAT(NZ-1))
RAD=RAD1+(FLOAT(I2-1))*DLR
C****** COMPUTE GRID ON WING
READ(10,332) XS,YS
SHIFT=XS
DO 315 IX=1,NX
IF(IX.LT.NX) READ(10,332) XIP1, YIP1
IF(IX.EQ.1) XIM1=XS+(XS-XIP1)
IF(IX.EQ.1) YIM1=YIP1
IF(IX.EQ.NX) XIPl=XS+(XS-XIM1)
IF(IX.EQ.NX) YIP1=YIM1
FORMAT (2 (1X, E14.7))
332 CALL OUTBOUN(IX,XO,YO,RAD,NX,SHIFT,PI)
CALL COEFFSR (EG,XIP1,YIP1,XIM1,YIM1,XO,YO,E (2),XS,YS,IX,NLE,1PP,QQ)
DO 316 IY=IYST,IYND
IF(IY.EQ.1) GO TO 335
FP=(1.0-(FLOAT(IY-2)/FLOAT(NY-2))**PEX)
FQ=(1.0-(FLOAT(IY-2)/FLOAT(NY-2))**QEX)
GO TO 334
FP=1.0
FQ=1.0
IF(PP.EQ.0.) FP=1.0
PP=PP*FP
QQ=QQ*FQ
CALL SURFR(EG,XS,YS,XO,YO,E(IY),XX,Y,PP,QQ,IY,NY,IX,NLE)
I=0
WRITE(12,100) XX,Y,Z
100 FORMAT(3F15.8)
IF(IY.EQ.IYST) I=1
IF(IX.EQ.1.AND.IY.EQ.IYST) I=2
PRINT 103, XX,Y,Z,II,II
CONTINUE
XIM1-XS
YIM1=YS
XS-XIP1
YS-YIP1
CONTINUE
103 FORMAT(3F10.5,2I1)
CONTINUE
STOP
END
SUBROUTINE XGIN(NP,X,DX)
C*
XGIN DEFINES THE NO. OF X-STATIONS, NP, THE
C*
DISTRIBUTION OF X-VALUES, & THE INCREMENT,DX *
COMMON/BOOKP/EM,EX,EG,SCL3,SCL1,SCL2,XN,XF,CC,SC,SE
DIMENSION X(101)
C FOR THE ANALYTICAL INPUT OPTION
C CALCULATE THE CROSS SECTION X VALUES & DX
DX=(XF-XN)/FLOAT(NP))
XCAL=(XF-XN)/FLOAT(NP-1)
DO 50 I=1,NE
C X(I)=(FLOAT(I-1)*XCAL)+XN
X(I)=-(COS(3.1415926*FLOAT(I-1)/FLOAT(NP-1))-1.)*XF*.5
50 CONTINUE
RETURN
END
SUBROUTINE EGIN(NE,E)
C*
EGIN DEFINES THE NO. OF E-STATIONS, NE, THE
C*
DISTRIBUTION OF E-VALUES *
COMMON/BOOKP/EM,EX,EG,SCL3,SCL1,SCL2,XN,XF,CC,SC,SE
DIMENSION E(101)
C FOR THE ANALYTICAL INPUT OPTION
C CALCULATE THE GRID E-VALUES
DO 50 K=1,NE
E(K)=FLOAT(K-1)/FLOAT(NE-1)
50 CONTINUE
RETURN
END
SUBROUTINE YINC(TH,N)
DIMENSION TH(101)
DO 50 J=1,N
QM=(FLOAT(J-1)/FLOAT(N-1))
TH(J)=COS(QM*3.1415926505)
50 RETURN
END
SUBROUTINE SHAPE(XS)
C*
SHAPE DETERMINES THE VALUES OF THE SHAPE-VARIATION &
C SCALE-VARIATION PARAMETERS AT XS
COMMON/BOOKP/EM,EX,EG,SCL3,SCL1,SCL2,XN,XF,CC,SC,SE
C
C DETERMINE THE RATE OF CHANGE (EM) FROM ONE SHAPE TO
C ANOTHER & THE SHAPE SCALE (EX)
EX=EM
1 R=(XF-XS)/(XF-XN)

70
CC=RE*EXP
C
SC=(-2.5*RA+3.5)*RA*SCL2+.001
RA=(XS-XN)/(XF-XN)+.001
SC=RA**EX*SCL2+(1.-RA**EX)*SCL1
GO TO 2
QT=((SCL2-SCL1)/(XF-XN))*(XS-XN)
QT=QT**1.3
X3=(XF-XN)/3.
SC=QT+0.15*SIN((XS-XN)/X3*3.1415926)
END
SUBROUTINE SHAPG(ES)
C
SHAPG DETERMINES THE VALUES OF THE GRID-VARIATION &
GRID-VARIATION PARAMETERS FOR E=ES
C
DETERMINE THE RATE OF CHANGE (EM) FROM ONE SHAPE TO
ANOTHER & THE SHAPE SCALE (EX)
SE=1.-ES**EG
2 RETURN
END
SUBROUTINE FNDC(I,NP,T,IT)
C
FNDC COMPUTES THE END SHAPE COORDINATES AS FCNS. OF T,
& STORES THEM IN COMMON/DEF/
COMMON/DEF/F (3), YI (3)
DIMENSION A(3), B(3)
PI=3.14159265
A(1)=.33
A(3)=0.6
B(1)=1.
QUANT=-T**2/B(1)**2
IF (QUANT.LT.0.) QUANT=0.
F(1)=A(1)*SQRT(QUANT)
F(3)=A(3)*SQRT(QUANT)
IF (IT.EQ.2) F(1)=-F(1)
IF (IT.EQ.2) F(3)=-F(3)
XLO=2.521826
AK=1.7/XLO
BK=0.65/XLO
CK=5.2/XLO
DK=SQRT(1.19)/XLO
EK=AK-BK
TT=ABS(T)
IF (TT.GT.BK) GO TO 5
YI(2)=-BK*COS((T+BK)/(2.*BK)*PI)*.7
F(2)=.8*BK*SIN((T+BK)/(2.*BK)*PI)
IF (IT.EQ.2) F(2)=-F(2)
GO TO 7
5 YI(2)=T
F(2)=-0.01*(TT-BK)
7 YI(1)=T
YI(3)=T
RETURN
END
SUBROUTINE COEFFS(XIP1, YIP1, XI, YI, XIM1, YIM1, XOUT, YOUT, EE, PP, QQ, XI, YI)
COMMON/BOOKP/EM, EX, EG, SCL3, SCL1, SCL2, XN, XF, CC, SC, SE
C ANALYTICAL
PI=3.14159265
A(1)=.33
A(3)=0.6
B(1)=1.
QUANT=-T**2/B(1)**2
IF (QUANT.LT.0.) QUANT=0.
F(1)=A(1)*SQRT(QUANT)
F(3)=A(3)*SQRT(QUANT)
IF (IT.EQ.2) F(1)=-F(1)
IF (IT.EQ.2) F(3)=-F(3)
XLO=2.521826
AK=1.7/XLO
BK=0.65/XLO
CK=5.2/XLO
DK=SQRT(1.19)/XLO
EK=AK-BK
TT=ABS(T)
IF (TT.GT.BK) GO TO 5
YI(2)=-BK*COS((T+BK)/(2.*BK)*PI)*.7
F(2)=.8*BK*SIN((T+BK)/(2.*BK)*PI)
IF (IT.EQ.2) F(2)=-F(2)
GO TO 7
5 YI(2)=T
F(2)=-0.01*(TT-BK)
7 YI(1)=T
YI(3)=T
RETURN
END
SUBROUTINE COEFFS(XIP1, YIP1, XI, YI, XIM1, YIM1, XOUT, YOUT, EE, PP, QQ, XI, YI)
COMMON/BOOKP/EM, EX, EG, SCL3, SCL1, SCL2, XN, XF, CC, SC, SE
C***
COEFFS(XIP1, YIP1, XI, YI, XIM1, YIM1, XOUT, YOUT, EE, PP, QQ, XI, YI)
C
COMPUTE VALUE OF EXPONENT OF SE FOR ORTHOGONALITY
XDIFF=XIP1-XIM1
AM=(YIP1-YIM1)/XDIFF
C=YE-AM*XI
XX=XIM1
YY=AM*XX+C
SLP=(YIM1-YI)/(XIM1-XI)
SLP=ABS(SLP)
IF (SLP.GT.0.6) GO TO 30
A = (YOUT - YI) * (YY - YI)  
B = (XOUT - XI) * (XX - XI)  

IF (ABS(B) .LT. 0.0001) GO TO 10  
FF = 1.0 + (1.0 - SE) * (A/B)  
PP = ALOG(FF) / ALOG(SE)  
QQ = 0.8  
GO TO 20  

30 YY = YIM1  
   XX = (YY - C) / AM  
   A = (XI - XOUT) * (XX - XI)  
   B = (YI - YOUT) * (YY - YI)  
   IF (ABS(B) .LT. 0.0001) GO TO 10  
   FF = 1.0 + (1.0 - SE) * (A/B)  
   QQ = ALOG(FF) / ALOG(SE)  
   PP = 1.0  
   GO TO 20  

10 PP = 0.08  
   QQ = 1.0  
20 RETURN  
END

SUBROUTINE SURF (I, NP, N, X, T, E, Y, RI, PP, QQ)  
C * SURF TAKES X,E & T, & RETURNS Y & RI IN THE PARAMETER LIST,  
C * ANY CALL TO SURF MUST BE PRECEDED BY CALLS TO FNDC & SHAPE  
C POINT (X,Y,RI)  
C NORMAL (XI,YI,21)  
COMMON/BOOKP/EM,EX,EG,SCL3,SCL1,SCL2,XN, XF, CC, SC, SE  
COMMON/DEF/F(3), YI(3)  
SDY=SE**PP  
SDF=SE**QQ  
3 RBI=F(1) *CC+F(2) *OMC  
   YBI=YI(1) *CC+YI(2) *OMC  
   RI=SDF*SC*RBI+(1.0-SDF)*SKAL*F(3)  
   Y=SDY*SC*YBI+(1.0-SDY)*SKAL*YI(3)  
RETURN  
END  
SUBROUTINE EGINR(NE,E)  
DIMENSION E(101)  
DO 50 K=1,NE  
RETURN  
END  
SUBROUTINE OUTBOUN (IX, XO, YO, RAD, NX, SHIFT, PI)  
TH=(PI/FLOAT (NX-1)) *FLOAT (IX-1)  
XS=-SIN (TH) *RAD  
XO-XS+SHIFT  
YO=-COS (TH) *RAD  
FAC=(RAD-ABS (XS)) *(RAD-ABS (YO))  
FAC=1.1*FAC/ (RAD*RAD)  
YO=YO* (1.0-FAC)  
RETURN  
END  
SUBROUTINE COEFFSR (EG, XIP1, YIP1, XIM1, YIM1, XO, YO, EE, XI, YI, IX, NLE, C****** EGIN DEFINES THE DISTRIBUTION OF E - VALUES  
DIMENSION E(101)  
DO 50 K=1,NE  
50 E(K) = FLOAT(K-1)/FLOAT(NE-1)  
RETURN  
END  
SUBROUTINE OUTBOUN (IX, XO, YO, RAD, NX, SHIFT, PI)  
TH=(PI/FLOAT (NX-1)) *FLOAT (IX-1)  
XS=-SIN (TH) *RAD  
XO-XS+SHIFT  
YO=-COS (TH) *RAD  
FAC=(RAD-ABS (XS)) *(RAD-ABS (YO))  
FAC=1.1*FAC/ (RAD*RAD)  
YO=YO* (1.0-FAC)  
RETURN  
END  
SUBROUTINE COEFFSR (EG, XIP1, YIP1, XIM1, YIM1, XO, YO, EE, XI, YI, IX, NLE, C****** COMPUTES VALUE OF EXPONENT OF SE FOR ORTHOGONALITY  
IF (IX.EQ.NLE) GO TO 10  
XDIF=(XIP1-XIM1)  
AM=(YIP1-YIM1)/XDIF  
CC=AM*AM  
XX=AM  
IF (IX.GT.NLE) XX=XIP1  
YY=AM*X X+CC
SLP = (YIM1-YI) / (XIM1-XI)
SLP = ABS(SLP)
IF (SLP .GT. 20.0) GO TO 30
E2 = 1.0 - EE**EG
A = (YO-YI) * (YY-YI)
B = (XO-XI) * (XX-XI)
IF (B .LT. 0.00001) GO TO 10
FF = 1.0 + (1.0 - E2) * (A/B)
PP = ALOG(FF) / ALOG(E2)
PP = ABS(PP)
QQ = 1.0
GO TO 20
30
YY = YIM1
XX = (YY-CC) / AM
A = (XI-XO) * (XX-XI)
SE = 1.0 - EE**EG
IF (ABS(B) .LT. 0.0001) GO TO 10
FF = 1.0 + (1.0 - SE) * (A/B)
QQ = ALOG(FF) / ALOG(SE)
PP = 1.0
GO TO 20
10
PP = 0.06
QQ = 1.0
20
RETURN
END

SUBROUTINE SURFR(EG, XI, YI, XO, YO, ES, X, Y, PP, QQ, IY, NY, IX, NLE)
C****** SURF TAKES INNER BOUNDARY AND OUTER BOUNDARY X,Y AND
C****** RETURNS X,Y OF GRID LINES
SE = 1.0 - ES**EG
SDX = SE**PP
SDY = SE**QQ
X = SDX*XI + (1.0-SDX)*XO
Y = SDY*YI + (1.0-SDY)*YO
RETURN
END
A computer code for fast automatic generation of quasi-three-dimensional grid systems for aerospace configurations is described. The code employs a homotopic method to algebraically generate two-dimensional grids in cross-sectional planes, which are stacked to produce a three-dimensional grid system. Implementation of the algebraic equivalents of the homotopic relations for generating body geometries and grids are explained. Procedures for controlling grid orthogonality and distortion are described. Test cases with description and specification of inputs are presented in detail. The Fortran computer program and notes on implementation and use are included.