IN-61
68716
P-27

NASA Contractor Report 189104

IGB Grid: User's Manual (A Turbomachinery Grid Generation Code)

T.A. Beach
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

and

G. Hoffman
Applied Research Lab
State College, Pennsylvania

January 1992

Prepared for
Lewis Research Center
Under Contract NAS3-25266

National Aeronautics and
Space Administration

IGBGrid: User' Manual
(A Turbomachinery Grid Generation Code)

T.A. Beach
Sverdrup Technology, Inc.
Lewis Research Center Group
Brook Park, Ohio

and

G. Hoffman
Applied Research Lab
State College, Pennsylvania

I. Introduction

The IGB code is designed for use with the NASA Lewis KSTAGE flow code, which computes three-dimensional, viscous, turbulent flow through a multistage turbomachine. The grid is tailored for use with KSTAGE and the users manual is written with this application in mind. Nevertheless, the code may be used for other turbomachinery flow codes, the applications to which should be evident.

Before going into the workings of the grid code, a brief description of KSTAGE will be given to help those unfamiliar with KSTAGE understand certain aspects of IGB which may not be directly relevant to other flow codes. KSTAGE solves the average passage equations, as defined by Adamczyk . The earlier VSTAGE code, developed by Adamczyk and Celestina, was extended by Kirtley to allow the use of body orthogonal grids instead of sheared grids as in VSTAGE. This and other changes led to the KSTAGE code.

The approach used in KSTAGE is to model the flow through a multi blade row turbomachine using a separate three dimensional grid for each blade row overlaid on portions of a single, common axisymmetric grid. Figure 1 shows blade-to-blade hub surface grid for one blade of a four blade machine together with the common axisymmetric grid. Source terms in the flow equations are calculated based on the geomtrically correct region of each grid and applied to the other grids in order to account for blade row coupling. A necessary condition for convergence is that all three-dimensional grids produce the same axisymmetric flowfield. With regard to the grid, the main point is that KSTAGE requires a common axisymmetric grid as well as detailed three-dimensional grids for each blade row. Each of the detailed grids extends through the entire machine, but only one blade per grid is modelled geometrically correctly. This configuration is what IGB was written to produce.

II. Description of the IGB Code

IGB creates a 3-D grid in a turbomachine by stacking a series of gridless surfaces between the hub and the tip. These surfaces are formed by rotating a meridian curve about the axis of symmetry of the turbomachine and are thus, axisymmetric. This is the common axisymmetric grid. When creating what is called "detailed 3-D grid" about a particular blade, the user selects a certain number of these surfaces on which to create interactively 2-D grids of the H-type. On each surface a grid is created only in a single blade passage. The user interactively partitions the blade passage into quadrilateral sub-domains. On command, within each sub-domain, the code then creates a grid by transfinite interpolation which is continuous over the entire blade passage. Smoothing of these initial grids may be effected by interactively invoking an elliptic smoother. After all the control-surface grids have been created interactively, the program computes the grids on the remaining surfaces by interpolation. IGB is organized into seven stages or steps, as follows:

1. First, the common axisymetric grid is created in the program called 'axi'. This program collects the geometrical blade, hub and casing data and combines these data with information on desired grid size to form the common axisymmetric grid. The final three-dimensional grid will lie on surfaces of revolution created by rotating the meridian plane curves about the shaft axis. This program also prepares input files for the next step.

2. In the second step, the surface definitions for the detailed three-dimensional blade row grids are determined. This step is performed separately

for each blade row in a program called 'step1'. As already mentioned, the required input files for this step were created by 'axi', but can be modified at the user's discretion. The program 'step1' is presently a batch program which generates the inflow, outflow, and periodic boundaries, as well surface grids for the blades. In other words, 'step1' produces four sides of the computational box excluding the hub and casing which are as yet undefined.

3. At this point, an optional program called 'step12' can be used interactively to redistribute the grid points on the blade, within certain limits. By default, points are clustered at the leading and trailing edges only. Occasionally, there is a requirement to cluster points at some point on the blade as well, due either to geometric or flow gradients.

4. The next step is an interactive program called 'step2'. As mentioned previously, this program uses a combination of algebraic and elliptic techniques to create two-dimensional blade-to-blade grids at selected radial stations, including the hub and tip.

5. A batch program called 'step3' takes the two-dimensional surface grids and fills in the full three-dimensional grid by interpolation.

6. At this point a program called 'to_ascii2' is used to prepare the grid for transporting to a mainframe. The final ascii file may be readily transferred, although such transfers are quite slow. This program may be easily modified to produce a binary file, if a binary transfer is available to the mainframe.

7. The final step is called 'step5'. This program runs on the mainframe and reads the ascii files created by 'to_ascii2' and produces grid files ready for KSTAGE. Modification of 'step5' will undoubtedly be necessary for other applications.

First, a word about using the IGB code on the IRIS. The code is set up to use UNIX in a simple but specific way. In general, each application will have its own directory. In each application directory, separate sub-directories are assumed to exist for each blade row as well as the common axisymmetric grid. Shell files are included with the sample cases, which accompany the IGB tape, to facilitate moving and linking files to the proper directories. Likewise, each program is kept in a separate directory and compiled with the unix 'make' command.

III. Input File Instructions

The three input files required by IGB are:

1. "axi.input"
2. "data.input"
3. "step2.input"

The first two files are required by 'axi' and the last file by 'step2'.

The program 'axi' creates the following data files which are read by the succeeding program 'step1':

"step1.input"
"step1.input2"
"step1.input3"
"step1.input4"

'step1' in turn creates files for 'step2' which in turn creates files for 'step3'.

The two input data files required by 'axi' are as follows:

1. "axi.input"

This file contains general grid parameters. Odd numbered lines contain the names of parameters which appear as numbers on the following line. All data for this and subsequent files are in free format.

Input 1 - 2 integers.

NTMAX - Number of points axially.
NRMAX - Number of points in radially.

Input 2 - Fixed point numbers.

DR1 - Radial grid spacing of meridian surfaces at hub.
DR2 - Radial grid spacing of meridian surfaces at casing.

Input 3 - Fixed point numbers.

DZ1 - Axial grid spacing at inflow plane.
DZ2 - Axial grid spacing at outflow plane.

Input 4 - 1 integer.

NBLADE - Number of blade rows to read.

Input 5 - 2 integers.

NLE - Axial index of leading edge of blade number 1.
NBLADE - Number of points in axial direction on meridian plane projection of blade, including leading and trailing edge.

Input 6 - 2 fixed point numbers.

DZ1 - Axial grid spacing on meridian surface at leading edge of blade. DZ2 - Axial grid spacing on meridian surface at trailing edge of blade.

Input 7 - 1 integer.

NBREAK1 - i index of beginning of detailed 3-D H-grid around blade no. 1.

Input 8-9 - Two lines of Hollerith field.

data.input (or local filename)
NULL, or CAD file name.

Typing "NULL" indicates to the code that the hub, casing and blade data are in the file "data.input" (described below), whereas, typing the CAD file name indicates that "data.input" contains ONLY the hub and casing data, and that the CAD file contains the blade data in a user defined form.

Please note that inputs 6 through 10 are repeated NBLADE times corresponding to the number blade rows. This means that there will be separate "data.input" and CAD files for EACH blade row. In the axi directory, these files must have unique names. They can be linked to the bladerow directory with the name "data.input".

Input 10 - 1 integer.

NBREAK2 - i index of end of detailed 3-D H-grid around last blade.

Input 11 and succeeding lines. These lines give the (x,r) coordinates of the inflow and outflow lines.

First group is inflow line.

NPTIN - Number of points describing inflow line.
ZIN(I) - Inflow axial coordinates
(i=1,2,...,NPT)
RIN(I) - Inflow radial coordinate
(i=1,2,..,NPT)

Second group is outflow line.

NPTOUT - Number of points describing outflow line.
ZOUT(I) - Outflow axial coordinates
(i=1,2,...,NPT)
ROUT(I) - Outflow radial coordinates
(i=1,2,..,NPT)

2. "data.input"

As mentioned above, the file "data.input" contains either (a) the definition of the meridian section of the hub and casing, as well as the blade coordinates (when "NULL" appears in line 10 of "axi.dat"), or (b) the definition of the meridian section of the hub and casing ONLY (when the CAD file name appears in line 10 of "axi.dat"). Case (a) is described below. The CAD file is user defined and requires some modifications to the code. Users desiring to exercise the CAD option should contact the author.

The modifications necessary to "data.input" to run a 2-D case are given in Appendix II.

Input 1 - A80

TITLE - Title of problem being run.

Input 2 - free format, 5 integers.

NBLADE - Number of blades.

NPH - Number of points describing hub. Maximum allowed is 233.

NPA - Number of points describing tip (casing). Maximum allowed is 233.

NBS - Number of blade sections in input data. Maximum allowed is 40.

NPPP - Number of points describing each blade. Maximum allowed is 55.

Input 3 - free format, 1 integer.

LBLAD - Flag describing type of blade data.

= 1 THETA1 defines the mid-chord line, THETA2 defines the blade thickness at constant axial location.

= 2 THETA1 defines upper side, THETA2 defines lower side.

Input 4 - 8F10.5

HUBZ - Hub axial coordinates, in order from 1 to NPH.

Input 5 - 8F10.5

HUBR - Hub radial coordinates, in order from 1 to NPH.

Input 6 - 8F10.5

ANNZ - Casing axial coordinates, in order from 1 to NPA.

Input 7 - 8F10.5

ANNR - Casing radial coordinates, in order from 1 to NPA.

Input 8 - 4F10.5

ZHIN - Axial coordinate of first hub point.
ZTIN - Axial coordinate of first casing point.
RHIN - Radial coordinate of first hub point.
RTIN - Radial coordinate of first casing point.

The values of these inputs are not used if the 'axi' porgaram was used to create an axisymmetric grid. They are included for backward compatability only.

Input 9 - 4F10.5

ZHOUT - Axial coordinate of last hub point.
ZTOUT - Axial coordinate of last casing point.
RHOUT - Radial coordinate of last hub point.
RTOUT - Radial coordinate of last casing point.

The values of these inputs are not used if the 'axi' porgaram was used to create an axisymmetric grid. They are included for backward compatability only.

Input 10 - 8F10.5

ZBL - Blade axial coordinates, in order from 1 to NPPP. These data are input for each blade section, in order from 1 to NBS.

Input 11 - 8F10.5

RBL - Blade radial coordinates, in order from 1 to NPPP. These data are input for each blade section, in order from 1 to NBS.

Input 12 - 8F10.5

THETA1 - Blade angular coordinate in radians, in order from 1 to NPPP. These data are input for each blade section, in order from 1 to NBS.

Input 13 - 8F10.5

THETA2 - Blade angular coordinate in radians, in order from 1 to NPPP. These data are input for each blade section, in order from 1 to NBS.

The blade data are input beginning at the leading edge, the index of which is taken as unity. The trailing edge is then NPPP.

Note that blade input data in standard NASA/Lewis form do not necessarily close at the leading and trailing edges. Closure is performed by the computer program by putting in circular arcs tangent to the first and last input point pairs to form a leading and trailing edge. If the leading edge points are equal on the pressure and suction side of the blade, this modelling will not be done.

As mentioned previously, the program 'axi' creates four files which are read by 'step1'. The contents of these files will now be presented as a reference should the user desire to make modifications to them.

1. "step1.input"

This file contains quantities that define the extent of the grid and the geometry file names. Odd numbered lines indicate the inputs which follow, only even numbered lines contain input.

Line 2 - seven integers.

NRMAX - Number of radial grid points. Maximum allowed is 45.

NTMAX - Number of axial grid points (in the main flow direction). Maximum allowed is 233.

NBMAX - Number of grid points desired on the blade.

NFRONT - Number of grid points desired in front of the blade - between the inflow surface and blade leading edge.

NWAKE - Number of grid points desired behind the blade - between the trailing edge and outflow boundary.

Line 4-6 - hollerith field.

FILNAME - Input file names. These are: data.input (data.input in form as given above) or CAD file name (user supplied).

GRIDTYPE - A C-grid option is planned but not currently supplied. Use: HGRID

2. "step1.input2"

This file contains a block of data for each "radial cut," i.e., the axial-running grid line that intersects the blade leading edge and leaves the blade trailing edge (this is called the periodic line). Figure 2 shows a schematic which should help clarify these inputs.

Line 1 - two integers.

IDTE - Number of indices to move cut point at trailing edge. The default is "0". The convention is that a postive index moves the point clockwise and a minus index, counterclockwise. See Figure 2.

IDLE - Number of indices to move cut point at leading edge. Same convention as for IDTE.

Line 2 - step sizes (arc lengths) along blade pressure surface (low index). Two fixed point numbers.

DS1 - Step size at leading edge cut point.

DS2 - Step size at trailing edge cut point.

Line 3 - step sizes (arc lengths) along blade suction surface (high index). Two fixed point numbers.

DS1 - Step size at leading edge cut point.

DS2 - Step size at trailing edge cut point.

3. "step1.input3"

This file contains grid step size information for meridian surfaces and blade passages on axisymmetric surfaces, as well as parameters to locate the periodic boundaries from the inflow boundary to the blade leading edge and from the blade trailing edge to the outflow boundary. Figures 3-6 are schematics which explain the various data in this file.

The data for "step1.input3" are in two blocks of identical information.

First block: Describes portion of blade passage between inflow boundary and blade leading edge.

Input 1 - one fixed point number.

EXTA - Vertical location of periodic line at inflow boundary, fraction of unity. See Figure 3 for schematic. The forward periodic line is taken to be a cubic curve extending from the leading edge "cut point" to the inflow line and has zero slope at the inflow line.

Input 2 - contains NRMAX (number of radial surfaces) lines of data giving step sizes and blade angles at the leading edge of each radial surface. Lines are input in order from 1 (at hub) to NRMAX (at casing). Each line contains the following four data:

INNBS - Surface counter.

ANG - Angle of inclination of periodic line at airfoil leading edge with respect to horizontal. See Figure 3 for schematic. The angle "ANG" is measured in the first quadrant and in the schematic is positive. If ANG $\geq$ 360 degrees, an extension of the mean camber line is used.

DS1 - Normal step size from top surface of lower blade to first grid line off surface, at intersection of periodic line with leading edge. See Figure 4 for schematic.

DS2 - Normal step size from bottom surface of lower blade to first grid line off surface, at intersection of periodic line with leading edge. See Figure 4 for schematic. If DS2 is not equal to DS1, then linear interpolation is used to set values along the blade.

Second block: Describes portion of blade passage between blade trailing edge and outflow boundary.

Input 1 - one fixed point number.

EXTB - Vertical location of periodic line at outflow boundary, fraction of unity. See Figure 5 for schematic. The aft periodic line is taken to be a cubic curve extending from the trailing edge "cut point" to the outflow line and has zero slope at the outflow line.

Input 2 - contains NRMAX (number of radial surfaces) lines of data giving step sizes and blade angles at the trailing edge of each radial surface. Lines are input in order from 1 (at hub) to NRMAX (at casing). Each line contains the following four parameters:

INNBS - Surface counter.

ANG - Angle of inclination of aft periodic line at blade trailing edge with respect to horizontal. See Figure 5 for schematic. The angle "ANG" in this case is also measured in the first quadrant. In the schematic it is negative. As in the forward periodic line, if ANG $\geq$ 360 degrees, an extension of the mean camber line is used.

DS1 - Normal step size from top surface of lower blade to first grid line off surface, at intersection of periodic line with trailing edge. See Figure 6 for schematic.

DS2 - Normal step size from bottom surface of lower blade to first grid line off surface, at intersection of periodic line with trailing edge. See Figure 6 for schematic.

4. "step1.input4"

This file contains grid step size information in the main flow direction for each meridian surface. The data for "step1.input4" consist of two columns of numbers, as follows:

First block: Describes portion of meridian surfaces between inflow boundary and blade leading edge. There will be NRMAX lines of input per block.

DS1 - Step size on stream surface at inflow boundary (low index).

DS2 - Step size on stream surface immediately in front of blade leading edge (high index).

Second block: Describes portion of meridian surfaces between blade trailing edge and outflow boundary. Note that this section of grid input is appended directly with no blank spaces to the first block above.

DS1 - Step size on stream surface immediately aft of trailing edge (low index).

DS2 - Step size on stream surface immediately in front of outflow boundary (high index).

'step2' reads only one input file, as follows:

1. step2.input

This file tells 'step2' how many interactive grid surfaces there there will be and where these surfaces are located among the gridless axisymmetric surfaces created by 'step1'. The hub and casing are always the first and last of these surfaces to be grided. All inputs to this file are integers in free form.

Input 1.

NPLAN - Number of surfaces to be grided. Maximum allowed is 15.

Input 2.

NN(I) - Vertical index location of i'th grid surface. There are NPLAN of these lines of data, one for each surface to be grided, beginning at the hub (NN = 1), in succession to the casing (NN = NRMAX, from "step1.input"). Maximum allowed is 45.

IV. Running the Code.

Figures 9 and 10 show a flow chart and sample directory tree, respectively. To run IGB, the procedure is as follows:
1. In the directory containing the inputs for the 'axi' program, create the input files "axi.input" (general parameters) and "data.input" (containing the hub, casing and blade definitions for all blade rows, as described in Section III). Note that if a CAD file is being used in place of "data.input", these files reside in the directory containing 'axi'.

2. Run 'axi'. This program is partially interactiver, but can be run without graphics if the inputs in "axi.input" are correct. The first image to appear on the screen is a plot of data points in the merdional plane (i.e. the inflow and outflow boundaries, blade leading edge and trailing edge shapes, and the hub and casing definitions). Note that the images on the screen can be manipulated with the middle and right mouse which function the same as in PLOT3D. To pass on to the next image, press the 'esc' key.

A second image now appears which shows the meridional plane with cut lines delineating the intersections of the detailed blade-to-blade grids with the axisymmetric grids shown in black. Press 'p' to give a plot of the meridional (axisymmetric) grid. Using the left mouse to select portions of the grid (and once again the middle and right mouse buttons to move the grid), you can manipulate the spacings and number of points in the grid by choosing from the options at the bottom of the graphics window and typing the appropriate changes in the text window. Note that these changes only affect the grid generated during this session. They do not affect the input data set, i.e. "axi.input".. Once you are satisfied with the axisymmetric grid, type 'esc' to permit the program to write appropriate output files.

As mentioned in Section III, the following files are created by 'axi':

"step1.input" - general data

"step1.input2" - periodic line data

"step1.input3" - blade to blade data

"step1.input4" - meridional (axisymmetric) data

Typing 'q' at any time will stop the execution of 'axi'. Periodically 'axi' will ask you to press 'tab' to clear the event queue.

3. Execute the UNIX shell 'move' by typing 'sh move '. This will link and/or move the appropriate output files from the 'axi' directory to each of the blade row directories. If CAD files are used, this shell move file must be edited to ensure that the "xxxx-xxx.cad" files in the —/axi directory are linked to their counterparts in the individual blade row directories.

4. Move to the directory of a particular blade row for the generation of the detailed grid associated with that blade row. Note that all remaining steps in the grid generation process for this particular blade row are executed from this directory. These steps consist of running the following programs: 'step1', 'step2', 'step3' and 'to_ascii2', as given below. A program called 'step12' is optional and explained in Appendix III.

5. Check the input files for step1 and make any desired changes. The user may come back to this step several times.

6. Run 'step1'. This program runs without any user intervention.

7. Create "step2.input". This file contains the number and locations of the interactive grid surfaces.

8. Run 'step2'. At the prompt for the input file, type 'step1.bin'. To proceed further, the user should follow the step-by-step instructions of the tutorial in the next section. The tutorial is designed to get you started in the interactive phase of creating a grid with this code and to give you a basic understanding of what the various commands do. Although the sequence of commands and the specific parameters are set up for a specific rotor geometry, they will not vary greatly for any other geometry. Once you have mastered the tutorial, you should be able to handle any other case without much difficulty. Remember, though, that creating an acceptable grid does require a certain amount of practice and some judgment on your part. Don't be afraid to experiment.

9. Run 'step3'. No interaction on your part is required. This program fills in the grid by interpolation between the control grid surfaces created in 'step2'.

Upon completion of 'step3', you will have an unformatted file called "step3.bin" which contains the interactive portion of the grid for the blade row in question. This grid is then mated with the axisymmetric grid in the following two steps.

10. Run 'to_ascii2'. This program takes the file "step3.bin" and writes it out in ascii format to send to the mainframe. The resulting file is called "grid.asc". This file, along with "axi.fmt" from the 'axi' code will be sent to the mainframe, where 'step5' is run.

11. Run 'step5'. This is a mainframe program designed to collect the ascii files being transferred from the IRIS to the mainframe and write them out in a single unformatted file suitable for KSTAGE. Specifically, using the files "grid.asc" and "axi.fmt" (the latter file is from the 'axi' directory and contains the axisymmetric grid), 'step5' merges the two files into one complete grid and writes the result in the file "fort.99", which is an unformatted FORTRAN file. 'step5' also writes the unformatted file "axi.d" which contains the axisymmetric grid.

Note that IGB creates the grid in the same dimensions as the input. At this point in 'step5' non-dimensionalization takes place. The reference length used by the code is called 'rnond' and is hardwired into 'step5'. When using IGB with the flow code KSTAGE, the pre-processor 'axiave' works best when the reference length is the maximum diameter of the machine.

Another point to note when recompiling 'step5' is that the array dimensions must EXACTLY match the size of the grid in each index.

12. Rename the file "fort.99" to a file to be used as input for your particular flow code. For example, if the renamed grid file is 'stage1.d', then the UNIX command is 'mv fort.99 stage1.d'. This is done

to avoid overwriting "fort.99" with the grid from succeeding blade rows.

13. The entire process of running 'step1' to 'step5' is repeated for each blade row. Suppose you have two blade rows for which you require grids. You would perform the sequence twice with the result that you would have three unformatted grid files: the two complete grid files 'stage1.d' and stage2.d' plus the common axisymmetric grid file 'axi.d'.

14. At this point, move the grid files (all detailed grid files and the axisymmetric grid) to the directory containing your flow solver. If you are using the flow solver KSTAGE, then in the flow solver directory, create a file called "axiave.inp". For the foregoing example with two full grids the file would read as follows:

axi.d (axisymmetric grid file)
stage1.d (full grid 1)
0 4 39 (shroud flag and cut indices for this blade row)
stage2.d (full grid 2)
0 39 77 (shroud flag and cut indices for this blade row)

The shroud flag = 0 for shrouded; 1 for unshrouded. Note that there are NO blank lines between the above lines of data. For other flow codes, you will need to write a pre-processor that will overlay the detailed grids on the axisymmetric grid at the proper axial locations.

V. Interactive Grid Tutorial

The geometry of the case chosen to illustrate how a grid is created in the interactive phase is shown in Figure 7. This geometry is a rotor with a rather large turning angle. Let us get started.

After typing 'step2' , the program will ask for the name of the file generated by 'step1'.

Type: 'step1.bin' .

The screen will then divide into two windows with the upper window containing the outline of the first control blade passage (the hub). The lower window is used to type grid generation parameters to the computer. Be sure that the red arrow is in the appropriate window when typing commands.

HUB GRID

You are now ready to PARTITION THE GRID on the first radial surface - the hub. The vertical green line that appears in the upper window next to the inlet plane is used to create partition lines. Certain keys on the keyboard control its movement, angle of intersection with the boundaries, and total arc length, as well as whether it is retained as a permanent partition line. In this tutorial, the keystrokes required will be explained as we proceed. A summary of all keystroke commands may be found in Appendix I.

To move the green line to the right, press the TAB key followed by the R key. These two keystrokes will move the line one step to the right. Notice that in the lower left corner of this screen the current value of the i index is given. If you overshoot the desired value of i, you can move the green line back to the left (one step) by pressing TAB followed by L. By repeatedly pressing TAB followed by R (or L), the green line will continue to move to the right (or left).

At some point in executing the keyboard strokes, you may find that nothing seems to happen. Press the TAB key when you get into this situation and then repeat the desired commands. In general it is a good idea to preface every keyboard stroke by pressing the TAB key. The TAB key had been setup to clear out the previous keyboard commands that were typed into the computer so that new commands will be recognized.

Let us proceed with the partitioning. The values of the index i that should be chosen in this example are as follows:

To select each of these values of i as a partition boundary, move the green line to the appropriate value of i. Then press TAB followed by either I or K. When you move on to the next position, you will see that a red line has appeared at the position you just selected as a partition boundary. A RED LINE indicates a partition boundary. the partition boundaries are formed from BEZIER CURVES (a cubic curve defined by four control points constituting the corners of a trapezoid).

At some of the above values of i, additional keystrokes will be required to produce a line with the correct inclination at the boundaries and with the proper length (extension). Let us take each value of i in the above list and go through the additional keystrokes required.

Value of i Location

6 - Midway between inflow line and Leading edge.

19 - Leading edge of airfoil.

27 - At end of high curvature region at Leading edge.

37 - Close to maximum blade thickness, where the passage begins to curve down sharply.

43 - About midway between maximum thickness and trailing edge.

58 - At beginning of round trailing edge.

65 - Trailing edge.

77 - About midway between trailing edge and outflow line.

i = 6 Here we are in a rectangular section, so no additional keystokes other than I or K are required. Once the green line is at i = 6, press TAB followed by I or K.

i = 19 Notice that when you reach this position, the green line automatically bends forward. The code has been set up so that at the leading and trailing edges, the Bezier curve bisects the angles at the lower and upper boundaries, a requirement for a good H-grid at these points. As the angles are already correct at i = 19 you don't have to change them.

Should a situation occur where you want to change these angles, this can be done by first pressing the TAB key and then one of the following keys:

Z - decreases the angle on the lower blade.

X - increases the angle on the lower blade.

A - decreases the angle on the upper blade.

S - increases the angle on the upper blade.

Try holding down one of the above keys and seeing what happens. Be sure to return the partition line to an angle of about 45 degrees.

You can also change the length of the partition line with the following keys:

C - increases length of line near lower blade.

V - decreases length of line near lower blade.

D - increases length of line near upper blade.

F - decreases length of line near upper blade.

After pressing the TAB key, try pressing one of the above keys to see what happens.

The leading edge partition line needs only a small amount of extension at each boundary. The exact amount is a matter of individual judgment.

i = 27 Notice that when you reach this position, the green line is orthogonal at both boundaries, a feature built in to the code. You will want to make a note to yourself that orthogonality of the grid at the boundaries is achieved in the partitioning process, NOT in the succeeding smoothing operations. The line, as it appears, is almost acceptable. The orthogonality does not extend far enough away from each airfoil for a good boundary layer computation. This defect can be rectified by about two strokes of the C key.

i = 37 This line also needs some extension to extend the orthogonality further away from each airfoil. Three strokes each of the C and D keys should suffice. Remember to hit the TAB key before you strike the C or D key; otherwise nothing will happen.

i = 43 This line is acceptable as it appears. To press the C or D keys for more extension will introduce more skewing of the cells in the mid passage in this region. One of the tradeoffs in a periodic and continuous grid in a blade passage with a lot of turning is that you cannot avoid having skewed grid cells. Thus to maintain orthogonality near the airfoils will introduce even more skewing in the interior grid. In this example, skewing will occur in the portion of the blade passage between approximately mid-chord and downstream of the trailing edge, where the periodic line begins to return to the horizontal.

i = 58 Here you should zoom in on the the blade passage so that you can see more clearly the fine details of the lines near the boundaries. This is done by first pressing the UP ARROW button a few times (to increase the speed of the zoom and translation process), and then using the right mouse button to translate and the middle mouse buttom to zoom. This line is almost acceptable as it is. One application each of the C and D keys should improve it so that it is acceptable.

i = 65 You are now at the trailing edge of the airfoil. As it is this line is much too close to the i = 58 line. To move it further away, try four strokes of the C key and two of the D key. This should give you an acceptable line.

i = 77 Almost acceptable as it stands. One stroke of C and one of D will improve it.

Now move the green line to the right as far as it will go (i = 80) by using the TAB and R keys. At this point you are finished with the partitioning process. Figure 8 shows how the blade passage should look after you have finished the partitioning.

Before going on to the next step, press the W key to write a file that contains all the i-locations and key strokes you made to produce the partition on the hub surface. This file is then read at each successive interactive grid surface to provide continuity in the grid partitioning topology. Without this type of continuity the stacked grid will not be smooth in the radial direction.

For a geometry without much twist in the radial direction, the partition commands will not change much as you proceeds outward radially. You may have to change a few line extensions here and there to get what you want. When there is considerable twist, more use of the C and D keys will be required to achieve an acceptable partition.

The NEXT STEP is to GENERATE AN ALGEBRAIC GRID between the partition boundaries you have just created. This is done by pressing the G key. You will notice that the grid is continuous everywhere, but not smooth (there are some kinks in the lines at partition boundaries). The algebraic grid also has other undesirable features; for example, there is too much clustering of blade-to-blade grid lines near the inflow and outflow boundaries. These undesirable features will be largely eliminated and the quality of the grid improved in the next step - elliptic smoothing.

In the ELLIPTIC SMOOTHING process, the code is solving a Poisson equation for the grid over a user-designated portion of the domain. Although there are 9 options in the elliptic smoother (each treating the right-hand-side of the Poisson equation in a different way), only two of the options are useful - option 6 and, to a lesser extent, option 1. Since there is no on-line help file for this step in the interactive grid generation, this part of the tutorial will explain how to use the elliptic smoother.

Before we get started, a few comments are in order about how the elliptic smoother works. Options 1 and 6 take the partition boundaries as created and smooth the interior. This means that the spacing on the partition boundaries does not change. To facilitate doing the smoothing only in desired regions, you will need to remember the i indices corresponding to the partition lines and the j index for the upper boundary (the lower boundary is j = 1).

The quantities P and Q are the right-hand-sides of the Poisson equation and apply, respectively, to the grid lines that run in the streamwise and blade-to-blade directions. These forcing functions are given weights which determine such things as spacing and orthogonality. A value of 0.5 for both P and Q will produce an interior spacing much the same as on the boundaries. For a value of zero, the Poisson equation reduces to Laplace's equation which will attempt to maintain orthogonality and equal cell area everywhere. A value approaching unity will produce clustering of lines in the coordinate corresponding to P or Q.

To invoke the elliptic smoother, type the letter E, making sure that the red arrow is in the upper window. You will then be asked for the following information:

> Elliptic smoother option (1-9), number of iterations, and weights for P and Q.

Upon typing this information, you will be asked for:

> Range in i and j (first i, last i, first j, last j) over which the smoothing is to be applied.

The above two lines of data are typed in the lower window.

To produce an acceptably smooth grid in the present example takes eight sets of smoother commands. These commands are fairly typical of any grid, although some variations will of course occur.

1. In the first step, we are going to smooth the entire grid outside the boundary layers on both airfoils. We will also ask for more clustering in the axial direction than blade-to-blade. Note that if we were to smooth over all values of j, we would lose the orthogonality of the lines near the airfoils. The two lines of data are:

```
6 2 .2 .4
1 81 8 34
```

Upon typing the above two lines of data, the smoothed grid will appear. Use the mouse to zoom in and examine how the grid has changed.

2. Remove the kink in the blade-to-blade lines near the upper surface of the lower airfoil.

```
6 1 .2 .45
1 81 6 9
```

3. Remove the kink in the blade-to-blade lines near the lower surface of the top airfoil.

```
6 1 .2 .45
1 81 33 36
```

4. Open up the blade-to-blade lines downstream of the trailing edge near the top boundary. Use the Laplace equation to do this.

```
1 1 0 0
65 81 19 41
```

5. Open up the blade-to-blade lines downstream of the trailing edge near the bottom boundary.

```
1 1 0 .15
65 81 1 23
```

In step 5, Q is given a small value to produce some blade-to-blade clustering. This is done to make the step size vary continuously across the periodic boundary. If steps 4 and 5 were combined using only Laplace's equation, the step size would not be continuous across the periodic boundary. This behavior is related to the curvature of the periodic boundary.

6. Open up the blade-to-blade lines top and bottom between the inflow line and airfoil leading edge.

```
1 1 0 0
1 19 1 41
```

Here we are able to do the opening up process all at once because the upstream periodic line is straight.

7. Remove the kink in streamwise lines at the airfoil leading edge.

```
6 1 .2 .49
18 20 1 41
```

Notice that here we selected our interval in i to contain just one step on either side of the kink.

8. Remove the kink in the streamwise lines at the airfoil trailing edge.

```
6 1 .2 .49
64 66 1 41
```

As in step 7, we only smooth one step on either side of the kink. Step 8 completes the sequence for grid smoothing.

We still have to correct the grid spacing along the inflow and outflow lines to make them more compatible with the grid nearby, as produced by steps 4 and 5. This is done using the "1" command with the red arrow in the upper window. Upon typing 1 , you will be asked for the following: last index and range. The "last index" refers to i-index where you want to choose the vertical spacing; the "range" refers to the range of the i index over which this spacing is to be applied. This information is typed in the lower window.

9. Correct the blade-to-blade spacing around the lnflow line. Type "1" followed by:

```
3 1 2
```

Here we are choosing the spacing on line 3 and applying it to lines 1 and 2.

10. Correct the blade-to-blade spacing around the outflow line. type "1" followed by:

```
79 80 81
```

Here we have chosen the spacing at line 79 and are applying it to lines 80 and 81.

You are now finished with the grid generation on the hub and are ready to go on to the next surface. In this example, there are four grid surfaces to be created interactively - the hub, two interior surfaces, and the casing.

In an ordinary interactive grid generation session, before proceeding further, you will need to create another window in which to write a file containing all the commands that you typed to do the elliptic smoothing and grid straightening. This file already exists for the example of the tutorial and is called "elliptic.in." When writing your own you can name the file according to what suits you.

When you are creating a grid of your own choosing, as you proceed with the elliptic smoothing, it is a good idea to jot down on a piece of paper those elliptic smoothing commands that work in sequence (and forget the rest). When finished, these notes will enable you to write your own elliptic smoother file.

In the present example, the file "elliptic.in" is as follows:

```
6 2 .2 .4
1 81 8 34
6 1 .2 .45
1 81 6 9
6 1 .2 .45
1 81 33 36
1 1 0 0
65 81 19 41
1 1 0 0
65 81 1 23
1 1 0 0
1 19 1 41
6 1 .2 .49
18 20 1 41
6 1 .2 .49
64 66 1 41
10 3 1 2
0 0 0 0
10 79 80 81
0 0 0 0
```

Notice that the first eight pairs of lines are the data exactly as you typed them to the screen in steps 1 - 8 to smooth the hub grid. The last two pairs of lines correspond to steps 9 and 10 - correcting the blade-to-blade spacing at the inflow and outflow lines. Since all these data are read in the same two-line format, four numbers to a line, a set of meaningless zeroes is added to the second line. Also, the "1" (which is the command to correct the line spacing) is replaced by "10".

In an ordinary grid generation session, once the ellptic command file has been written, kill the extra window, and return to the grid window.

SUCCEEDING GRID SURFACES

After creating the grid on the hub, only a few key strokes are usually required to create the grid on succeeding surfaces. These steps are the same for each surface and are as follows:

step1.bin - contains the input geometry read-in in 'step1'.

step2.bin - contains the grid surfaces created interactively in 'step2'.

1. Upon striking the ESCAPE key, the existing grid will disappear, then you will see numbers scroll on the lower window and suddenly in the upper window will appear the outline of the next radial surface with partition lines also appearing. The shapes of these lines may not correspond exactly to the lines from the preceding surface.

2. To obtain the correct partition lines, STRIKE THE B KEY to eliminate the unwanted partition lines. An unwanted grid can also be erased this way.

3. Now strike the N key to read the file from the surface where you last struck the W key (which contains Bezier curve information). In the tutorial you will have struck this key for the hub grid and will not need to strike it again. The W key will resurrect the desired partition lines.

If one or more of the partition lines does not look quite right to you, use the TAB and R key to bring the green line to the offending partition line. To change the line, strike the TAB key and then the N key. You can now go through whatever keystrokes are necessary to modify the line. Once satisfied with it, strike the I key to replace the old line.

4. Next, strike the Y key to read your elliptic smoother file. A message will appear in the upper window asking you for the name of this file. Bring the red arrow to the lower window and type "elliptic.in" (or whatever the name of the file happens to be). You will see the algebraic grid appear and evolve through the sequence of elliptic smoothing and line straightening steps, as written in this file.

Once the last interactive grid surface has been created (the casing), upon pressing ESCAPE, you will see integers scrolling on the lower window. When the left-most integer reaches the number of desired radial grid surfaces, STEP2 is finished, and a prompt will appear. This completes the TUTORIAL for STEP2. Go back to Section IV of the manual and proceed to STEP3.

VI. Plotting the Output

The plot files created by IGB for PLOT3D are all in cylindrical coordinates and end in the suffix ".bin". These files are as follows:

step3.bin - contains the complete 3-D grid for a single blade passage created by 'step3'.

In order to plot these files using PLOT3D, the user must run the utility program "ref". This module takes any ".bin" file, transforms the coordinates to Cartesian, and then writes a multiblock PLOT3D file. Upon executing "ref" , the program will ask you for the following information:

What is the filename?

How many passages?

"How many passages" refers to the number of blade passages that you would like to view using PLOT3D. This number is the also the number of blocks in the PLOT3D multiblock grid file. Upon typing the above information to the screen, the program prints out the number of radial grid surfaces (tubular surfaces) that you requested in the input. This portion of the program takes a long time to run (about 1 hour per blade passage). When finished, the code will tell you that the unformatted file "p3d.dat" has been written for PLOT3D. In PLOT3D, the read command is:

```
read/unformatted/mgr/x=p3d
```

VII. Utility Programs

The IGB code has a number of useful utility programs, all of which are located in the utility sub-directory. These files are listed here and a brief description is given of their use.

pblade - reads the input file "data.input" which contains the raw data points for the blades. It then writes an unformatted output file called "nublo.dat" for PLOT3D. This file is useful for viewing your raw input data to determine its quality.

cat_p3d - takes any of the PLOT3D files created by "ref" and puts them together (concatenates) into a single unformatted file called "catp3d.dat". This program is useful in making sure that the grid and the input data coincide and that the blade data are correct.

to_ascii2 - puts out an "ascii" file to transfer to the CRAY computer.

neg_jac - reads "p3d.dat" file and calculates the cell volumes to determine any negative jacobians.

The programs 'ref' and 'to_ascii2' also reside in the utility sub-directory.

Appendix I. Interactive Keyboard Control Commands for STEP2.

KEY	FUNCTION
TAB	Clear the event queue.
R	Move the current grid line (green) to the right. Must be preceded by TAB.
L	Move the current grid line (green) to the left. Must be preceded by TAB.
Z	Decrease angle of grid line at lower boundary in blade passage (low index).
X	Increase angle of grid line at lower boundary in blade passage (low index).
A	Decrease angle of grid line at upper boundary in blade passage (high index).
S	Increase angle of grid line at upper boundary in blade passage (high index).
C	Increase arc length (extension) of grid line near lower boundary in blade passage (low index).
V	Decrease arc length (extension) of grid line near lower boundary in blade passage (low index).
D	Increase arc length (extension) of grid line near upper boundary in blade passage (high index).
F	Decrease arc length (extension) of grid line near upper boundary in blade passage (high index).
K	Keep current line (green) as a partition boundary (red).
I	Insert the current line (green) as a new boundary between two existing lines (red).
M	Modify an existing partition boundary.
U	Delete an existing partition boundary.
G	Create an algebraic grid in the blade passage between chosen partition boundaries.
E	Perform elliptic smoothing. The program will type on the grid screen: "what type of smoothing 1-9, number of

iterations, and weights of P and Q." The user types the requested information in the lower window. The program then asks: "type in range i1, i2, j1, j2,." Again, type the requested information in the lower window.

1 Allows the user to correct the grid spacing near the inflow and outflow boundaries. The program will type on the grid screen: "type in last index and range." The user types in this information in the lower window. The last index refers to the i-index where you want to choose the vertical spacing. The "range" refers to the range in the i index over which the grid is to be corrected. Only one boundary (inflow or outflow) can be corrected per application of "1".

O Restore the previous grid if the result of the elliptic smoothing is not satisfactory.

B Begin again with no partition boundaries and no grid.

W Write a file containing the Bezier curve boundary information.

N Read the file containing the Bezier curve boundary information.

Y Read a file containing a series of elliptic smoothing instructions. The program will ask the name of this file.

Q Exit STEP2 immediately.

ESC Go to the next radial surface.

UP ARROW Increase sensitivity of mouse.

DOWN ARROW Decrease sensitivity of mouse.

MOUSE

MIDDLE BUTTON Zoom

RIGHT BUTTON Translate

Appendix II. Input File Modifications for 2-D Case.

In order to run IGB in the 2-D mode, the "data.input" file read in 'step1' must be modified, as shown below. Note that dummy inputs must be present, but their value is meaningless.

Input 1 - A80

TITLE Title of problem being run.

Input 2 - free format, 5 integers.

NBLADE Dummy.

NPH Dummy.

NPA Dummy.

NBS 1

NPPP Number of points describing blade.
Maximum allowed is 55.

Input 3 - free format, 1 integer.

LBLAD Flag describing type of blade data.

= 1 THETA1 defines the mid-chord line,
THETA2 defines the blade thickness at
constant axial location.

= 2 THETA1 defines upper side,
THETA2 defines lower side.

Input 4 - F10.5

SOLIDITY Ratio of pitch to chord.

STAGGER ANGLE Complement of leading edge sweep angle
(90 - sweep angle).

Input 5 - 4F10.5

ZHIN Axial coordinate of inflow plane.

ZTIN Axial coordinate of inflow plane.

RHIN Dummy.

RTIN Dummy.

Input 6 - 4F10.5

ZHOUT Axial coordinate of outflow plane.

ZTOUT Axial coordinate of outflow plane.

RHOUT Dummy.

RTOUT Dummy.

Input 7 - 8F10.5

ZBL Blade axial coordinates, in order from 1 to NPPP.
These data are input for each blade section, in order from 1 to NBS.

Input 8 - 8F10.5

RBL Blade radial coordinates, in order from 1 to NPPP.
These data are input for each blade section, in order from 1 to NBS.

Input 9 - 8F10.5

THETA1 Blade angular coordinate in radians, in order from 1 to NPPP.
These data are input for each blade section, in order from 1 to NBS.

Input 10 - 8F10.5

THETA2 Blade angular coordinate in radians, in order from 1 to NPPP.
These data are input for each blade section, in order from 1 to NBS.

For the 2D case, the program 'axi' is not run. THe input files for step1 must be created by the user with NRMAX = 1 .

Appendix III. Notes on the use of "step12"

Sometimes the default method of distributing points on the blade is not sastifactory. This usually occurs on radial machines which have relativly long blades with a bend in the middle. The program "step12" can be used to correct this. This program is run after step1 and is similar to "step2" as far as the user interface is concerned.

The first screen shows an picture of the blade passage at the hub with the current points shown with surface normals. There is also a red line connecting the blade surfaces. The intersection of this line and the blade surface will become a packing point for the new distribution. The red line can be moved back and forth by using the "z" and "x" keys. Upon positioning the packing points in the correct location, pressing the "r" key will start the redistribution. The program will request the grid spacing at the packing point on each side of the blade; after which, the number of points to be placed before this point will be requested. Typing "l" or "t" will allow new leading and trailing edge spacings to be input. This should be done at least once. Pressing the "esc" key will save this information and move on to a picture of the blade at the tip, where the process is repeated. Pressing the "esc" in this screen will cause the program to redistibute the points on all blade sections based on an interpolation between hub and tip. In practice, this has been found to be acceptable. The result of "step12" is a new grid file called 'step12.bin' and is used in "step2" as described in the text.

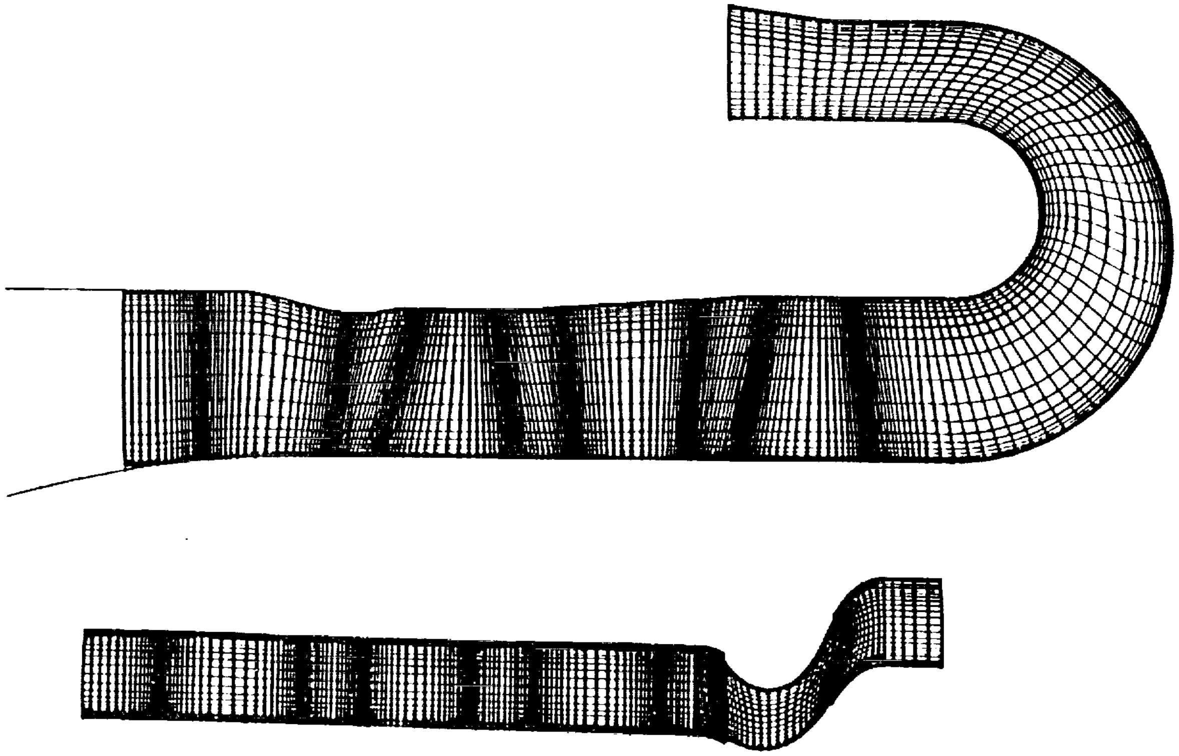

Figure 1. Axisymmetric mesh for a four blade machine with detailed mesh for the last blade row.

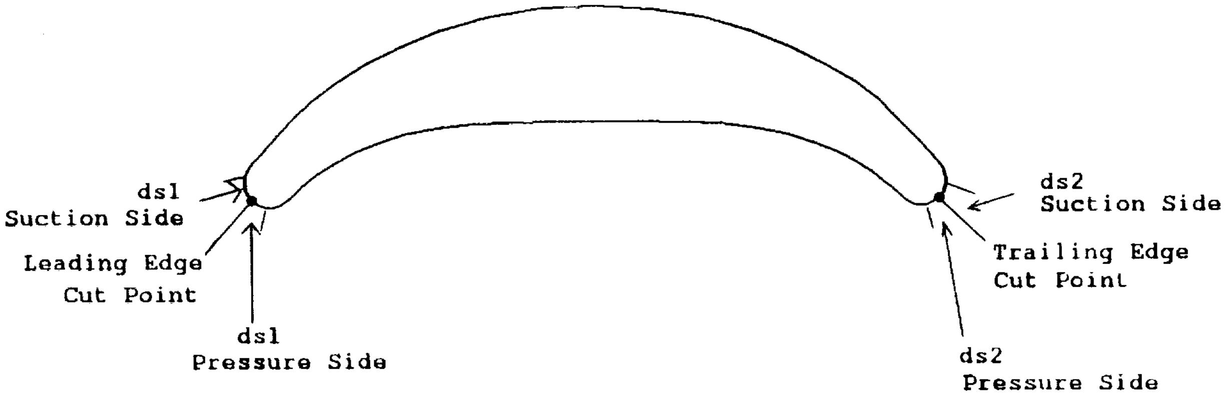

Figure 2. Schematic for "step1.input2" input file.

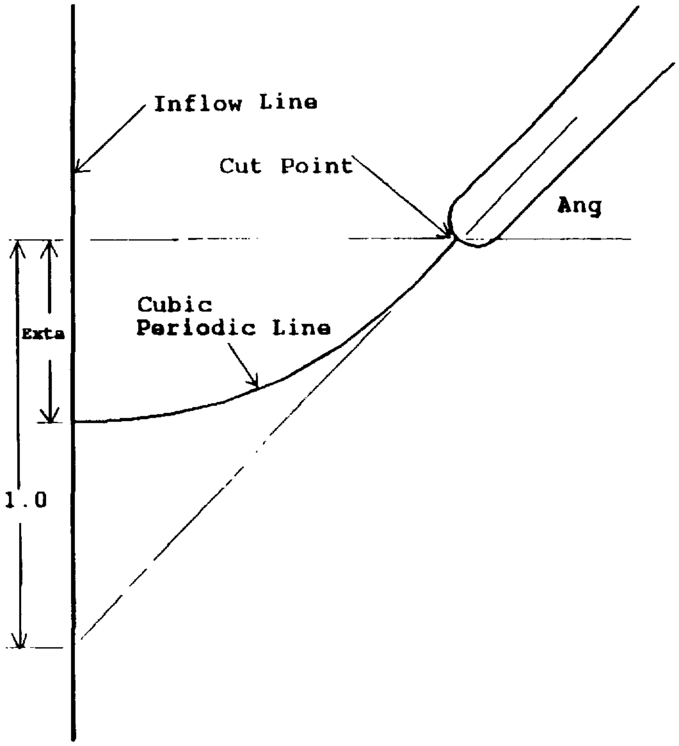

Figure 3. Schematic for inputs Exta and Ang.

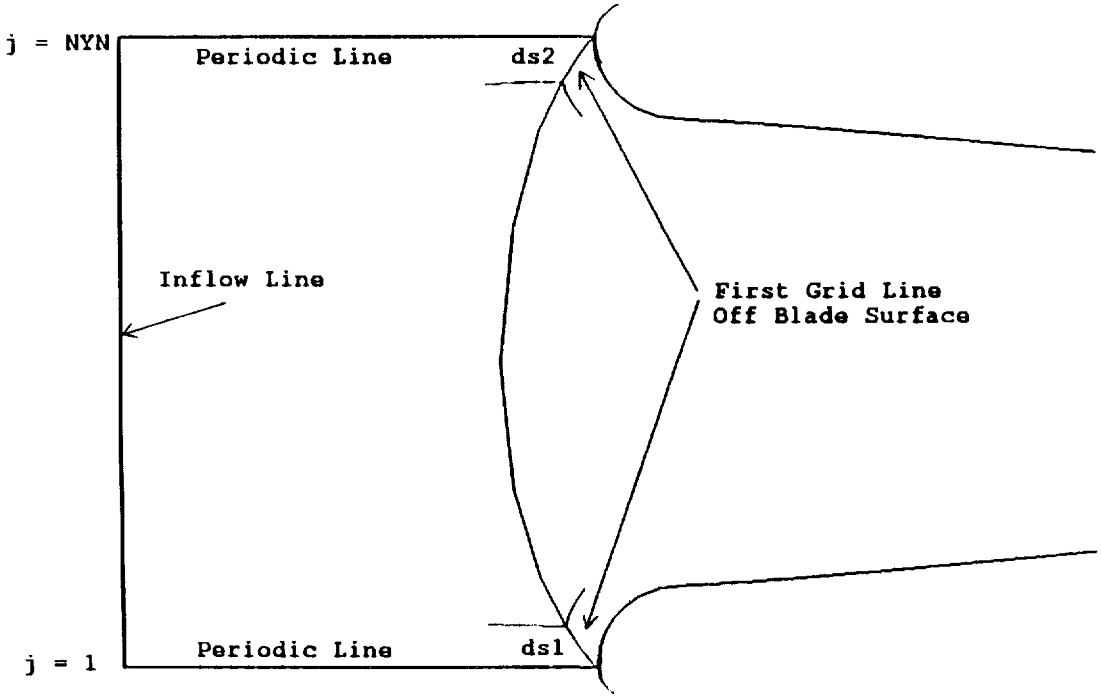

Figure 4. Schematic for inputs ds1 and ds2 in file "step1.input3".

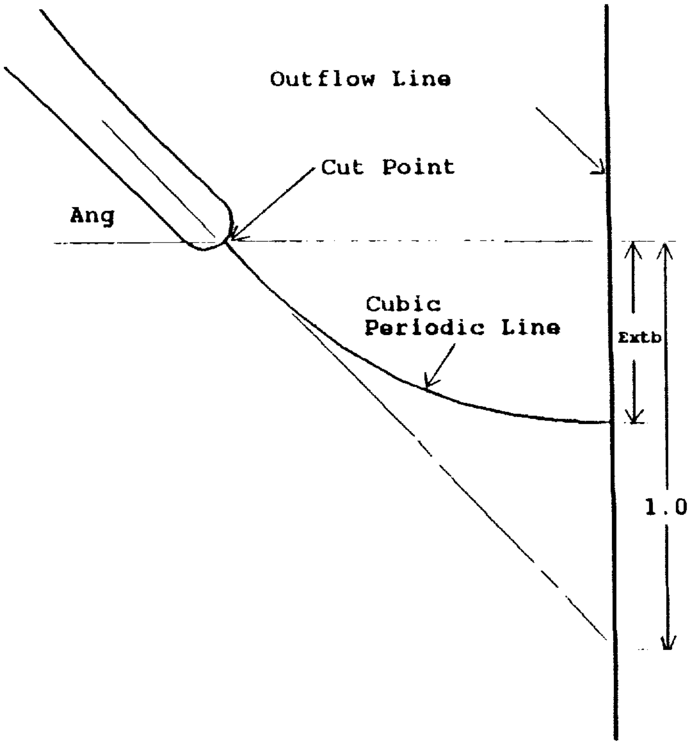

Figure 5. Schematic for inputs Extb and Ang.

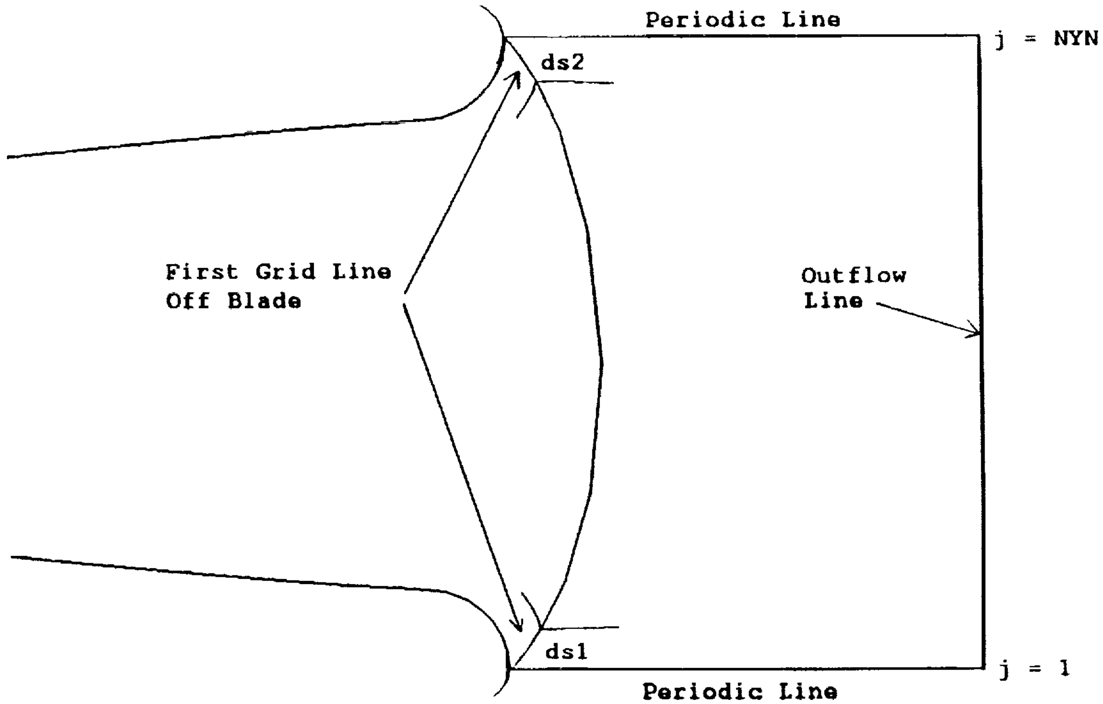

Figure 6. Schematic for inputs ds1 and ds2 in file "step1.input3".

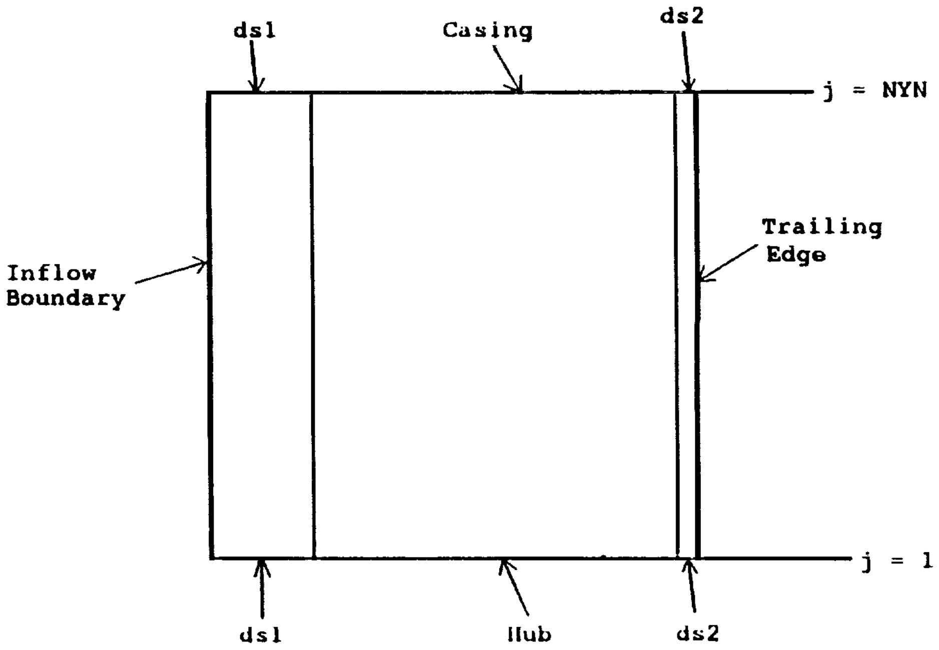

Figure 7. Schematic for inputs dsl and ds2 in file "stepl.input4".

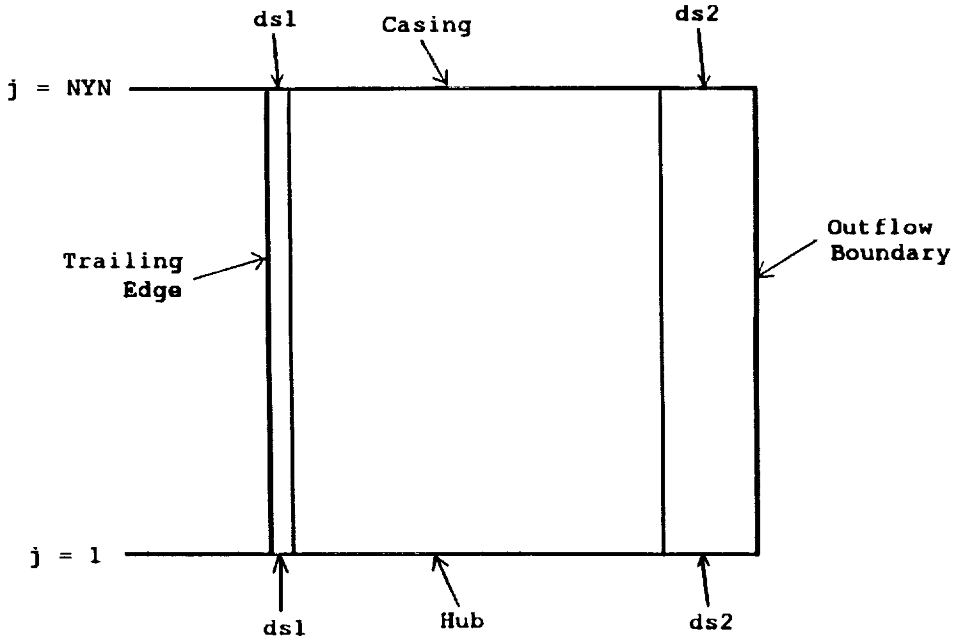

Figure 8. Schematic for inputs dsl and ds2 in file "stepl.input4".

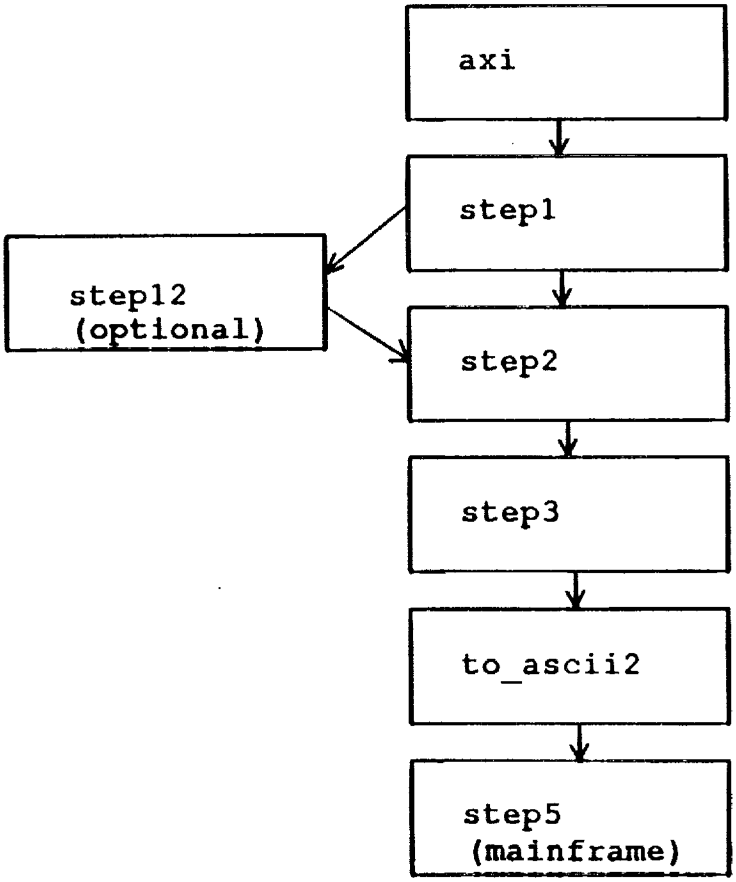

Figure 9. Flowchart for IGBgrid code

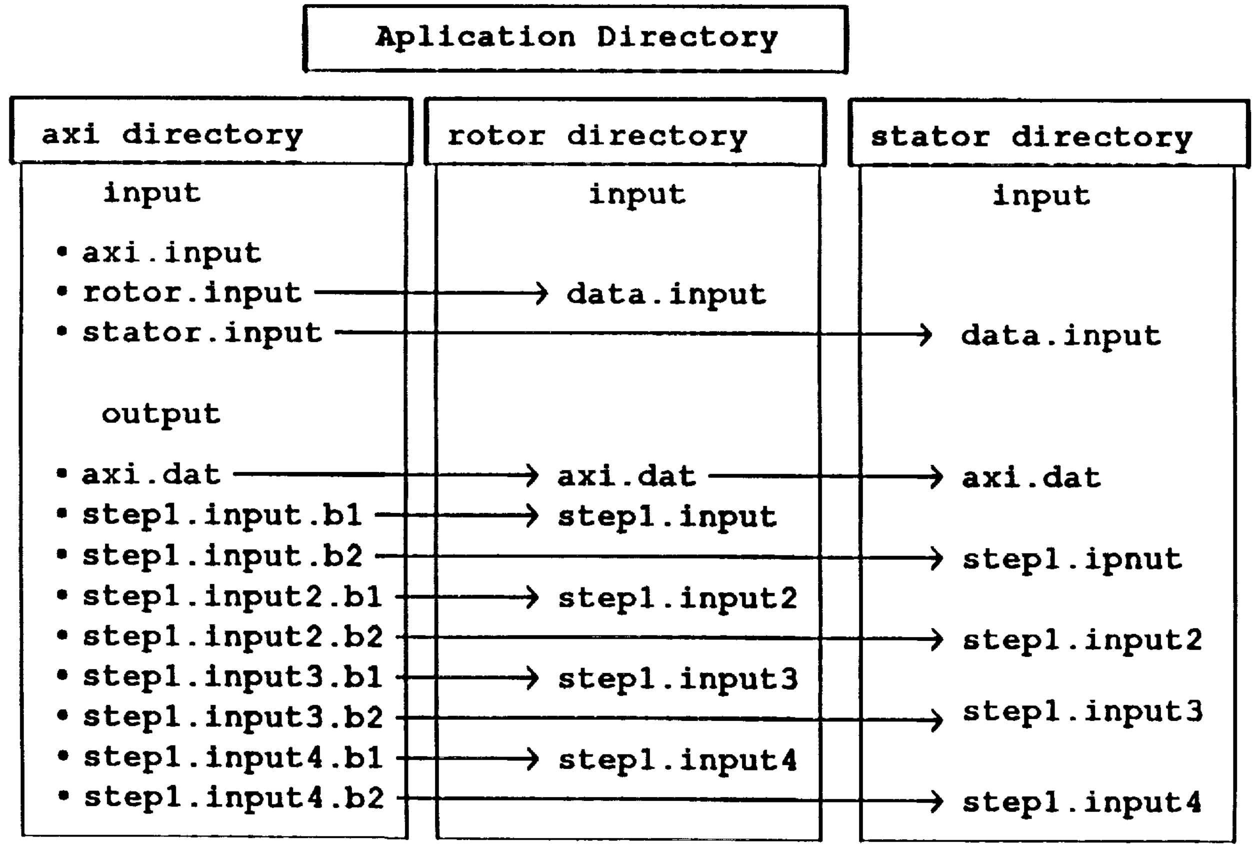

Figure 10. Directory tree showing inputs for IGBgrid code Arrows indicate equivalent files, either copied or linked.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
January 1992

3. REPORT TYPE AND DATES COVERED
Final Contractor Report

4. TITLE AND SUBTITLE
IGB Grid: User's Manual (A Turbomachinery Grid Generation Code)

5. FUNDING NUMBERS
WU-505-62-21
C-NAS3-25266

6. AUTHOR(S)
T.A. Beach and G. Hoffman

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Sverdrup Technology, Inc.
Lewis Research Center Group
2001 Aerospace Parkway
Brook Park, Ohio 44142

8. PERFORMING ORGANIZATION REPORT NUMBER
E-6771

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135-3191

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-189104

11. SUPPLEMENTARY NOTES
Project Manager, John Adamczyk, Lewis Research Academy, NASA Lewis Research Center, (216) 433-5829.
T.A. Beach, Sverdrup Technology, Inc.; G. Hoffman, Applied Research Lab, State College, Pennsylvania.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Categories 02 and 61

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

A grid generation code called IGB is presented for use in computational investigations of turbomachinery flowfields. It contains a combination of algebraic and elliptic techniques coded for use on an interactive graphics workstation. Included are the instructions for use and a test case.

14. SUBJECT TERMS
Turbomachinery blades; Computational grids

15. NUMBER OF PAGES
26

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT