Geometry Modeling and Multi-Block Grid Generation For Turbomachinery Configurations+

Ming H. Shih *

and

Bharat K. Soni **

National Science Foundation Engineering Research Center
For
Computational Field Simulation, Mississippi State University

ABSTRACT

An interactive three-dimensional grid generation code, TIGER (*T*urbomachinery *I*nteractive *G*rid gen***ER***ation) has been developed for general turbomachinery configurations. TIGER features the automatic generation of multi-block structured grids around multiple blade rows for either internal, external or internal-external turbomachinery flow fields. Utilization of the Bezier's curves achieves a smooth grid and better orthogonality. A graphic environment utilizing FORMS Library serves as the interface on the Silicon Graphics Inc. (SGI) IRIS 4D platforms. Based on the geometry information with its built-in pseudo-AI algorithm, TIGER generates the algebraic grid automatically. However, due to the large variation of turbomachinery configurations, this initial grid may not always be as good as desired. TIGER therefore provides graphical user interactions during the process which allow the user to design, modify, as well as manipulate the grid, including the capability of elliptic surface grid generation.

The computational mapping, geometry modeling, and the user-interactions are the the main procedures of TIGER. This presentation will cover these issues associated with general turbomachinery geometries. Various examples have been exercised to demonstrate the success of the developed algorithm.

+ This research is supported by a grant from NASA Lewis Research Center.

* Graduate Research Assistant, Ph.D. Student
Student member AIAA
Member Sigma Gamma Tau

** Associate Professor
Member AIAA

C-6

INTRODUCTION

During the last decade, computational fluid dynamics (CFD) has evolved as an essential technique for solving engineering problems associated with flow fields. CFD allows greater accuracy and improves the time-cost efficiency. Applications of CFD are now pratical on the flow field predictions for very complex geometries such as full scale aircrafts, counter-rotating propfans, and coastlines. CFD developments also help the numerical prediction on magnetic field and bio-chemistry. Great advances can be found in the flow field prediction associated with turbomachinery configurations. Acoustic prediction with CFD methods is currently an important issue in turbomachinery applications since environmental problems become the focus of the public. However, due to the complexity of the geometry, numerical grid generation associated with the flow field about turbomachinery systems is difficult and time-consuming. Even with the advances in the general purpose, interactive grid generation codes like GENIE[1,2,3,4] and EAGLEView[5,6,7], it is still a labor-intensive task to generate a grid for such applications which have large variations in design. As a consequence, there arises a strong demand for a grid generation tool to generate quality grids for a large variety of designs of the turbomachinery configurations in a time-efficient manner.

TIGER[7,8,9,10], an interactive grid generation code customized for turbomachinery applications, has been developed to meet this demand. The overall objective of TIGER was to develop an efficient and robust computational grid generation system tailored for complex turbomachinery applications that would be timely enough for engineering designs using computational fluid dynamics. It is written in both Fortran-77 and C languages, with Fortran routines doing most of the mathematical calculations and C routines driving the graphic interfaces. The graphic interface is an application of the FORMS Library[11] which is a graphical user interface toolkit applicable to the SGI IRIS 4D platforms. Figure 1-a is the main panel of the user interface with graphic window. Figure 1-b is an option panel for blade information inputs. TIGER is currently capable of generating structured grids for internal, external, and internal-external flow fields about turbomachinery systems. With a simple switch, the user is able to choose the preferred grid topology. TIGER automatically maps the physical configuration into the computational domain. It reads in various industrial geometry definitions for the solid entities, such as the blades, the hub/spinner and the duct/shroud from the data files in the form of discreted data points. Built-in geometry manipulators in TIGER will revolve, intersect, spline and generate the grid for these surfaces in sequence. Graphical user interactions allow the user to design and manipulate the grid mesh with mouse buttons and the graphic entities on the screen. After the boundary surfaces have all been generated, algebraically or with user's manipulation, the grid generation for each sub-block will take place with the default transfinite interpolation (TFI) or with the elliptic solver system.

CONSTRUCTING PROCEDURES

There are three major procedures in TIGER during the construction of the grid. They are (1) Computational Mapping, which controls the mapping from the physical domain into the computational domain; (2) Geometry Modeling, which contains several geometry manipulation routines, such as body of revolution, intersection of two surfaces, and curve/surface spline; and (3) User Interactions, which allow the user to design, modify, or manipulate the grid mesh interactively.

Computational Mapping

The establishment of a mapping from the physical domain to the computational domain is the first step toward the generation of the numerical grid. This can be accomplished by dividing a complex 3D flow field into a collection of contiguous, simply connected blocks filled with discreted points. Each of the solid surfaces forms the full or partial sides in a computational block. Figure 2–a is a typical two-block propfan application, and figure 2–b is the computational mapping for such a system.

Various types of grid topology are available in TIGER. If we use the designation "CH" to indicate C–type grid for the overall configuration, and H–type grid for the grid around the blades, there are two types of grid topology available to date. They are HH and CH types. HC and CC types of grid are currently under development. With a comprehensive input from the graphical interface or through the journal file, the user can choose the preferred topology. Based on this input, TIGER automatically maps the physical domain into the computational domain, and requests the user to provide appropriate information through the interface or the journal file.

Geometry Modeling

Geometry modeling is considered as the most delicate part of the whole process. We must pay close attention to preserve the definitions of the geometry surfaces, such as the blades and the hub, during the manipulations of these surfaces. Cubic spline is the most common technique used to spline the curves or surfaces to the desired number of points and point distribution. Cubic spline usually gives a very satisfactory result for a smooth curve or surface. However, it suffers the defects of introducing wiggles for a curve with large curvature change. If the free–end boundary condition is applied to the cubic spline, it is difficult for the curve to keep its slope at the ends.

Inversed B–Spline[12,13] algorithms, developed by Yoon and Thompson and later modified by Yu and Soni, are used for splining curves and are going to be used for splining surfaces in TIGER to avoid the problems caused by the cubic spline. This spline technique calculates the control vertices of an existing curve/surface (inverse calculation), and allows the user to input a new number of points and a new distribution; hence it allows the user the ability to re–spline the curve/surface to the desired mesh without loosing the definition. It also keeps the 3rd order smoothness of the curve/surface, viz., it has continuous second derivatives.

Due to the geometrical characteristic of the turbomachinery being axisymmetric, it provides the advantage to simplify the transformation between the 3D revolved surfaces into 2D surfaces. A simple transformation law may be expressed as the following.

$$\boldsymbol{m} = \sum_{i=2}^{i=NI} \alpha_i \tag{1}$$

$$\text{where } \alpha_i = \sqrt{(z_i - z_{i-1})^2 + (r_i - r_{i-1})^2}$$

$$\sigma = \theta \boldsymbol{r}_m \tag{2}$$

where $\boldsymbol{r}_m$ is the reference radius

The inverse transformation from (m,σ) into (Z,R,θ) is done with the aid of the B-spline algorithm.

User Interactions

It is the goal of TIGER to generate the smooth algebraic grid automatically with very few user inputs. However, it is not practical to expect a grid generation code that is smart enough to generate the grid automatically without any need for modification for any design of the turbomachinery configurations, because that there are too many variations in terms of configuration design. Therefore, certain user interactions must be provided for the user to modify minor portions of the grid to achieve the favorable grid. With this in mind, TIGER features three interactive procedures, which give the user the freedom to design their favorable grids. These interactions are to (1) design the "ruler lines"; (2) design the "segment curves"; and (3) manipulate the surfaces. They will be discussed in detail later.

Bezier curve/surface formulation plays a very important role in these interactions. A bicubic Bezier's surface can be expressed as

$$\boldsymbol{r}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} \boldsymbol{p}_{i,j}\, \boldsymbol{B}_i^m(u)\, \boldsymbol{B}_j^n(v), \tag{3}$$

where $\boldsymbol{B}_i^m(u)$ is Bernstein Polynomial of degree m,

$$B_i^m(u) = \binom{m}{i} (1 - u)^{m-i} u^i \tag{4}$$

and m, n are the numbers of Bezier points in u,v direction , respectively

$0 \le u \le 1 \quad ,0 \le v \le 1$

Set n = 0, then Eq. (3) becomes

$$r(u) = \sum_{i=0}^{m} p_i B_i^m(u) \tag{5}$$

The Bezier's algorithm in TIGER was programmed according to Eq. 3, viz., it is generalized for any degree of Bezier's surfaces. However, TIGER only applies Eq. 5 in its algorithm.

Bezier's curve is used heavily in TIGER due to two of its important properties, which makes it very useful in grid generation:

(1). $\sum_{i=0}^{m} B_i^m(u) = 1$ which implies that the Bezier curve is invariant under translation and rotation. In other words, the curve is independent of the choice of coordinate system.

(2). $\dot{r}(0) = \phi (p_1 - p_0)$ *and* $\dot{r}(1) = \psi (p_n - p_{n-1})$ where ϕ, ψ are scalars.

This property implies that the Bezier curve expresses the tangents at the end points in terms of difference with the Bezier control points p_1 and p_{n-1}, respectively, multiplied by constants ϕ and ψ.

Design of the "Ruler Lines"

A "ruler line", or "ruler" in short, is nothing but a grid line with constant J–indice. Graphically, TIGER allows the user to use the mouse buttons and graphic entities shown on the screen to "design" the rulers on the meridional surface, i.e. (Z,R) plane, with the Bezier's curve. Once this process is done, the grid generated later in the process will fall into this user–designed trend. TIGER keeps the design process executed by the user into a file in pseudo code format, which allows later reactment. In other words, the user does not need to go through the same process if nothing is changed in this part.

Design of the "Segment Lines"

Similar to the process of designing rulers, this step also allows the user to design the ruler line in segment with the graphical interaction provided by TIGER. It designs segments, however, in (m,σ) plane instead of (Z,R) plane. It therefore assigns the third coordinates θ to the ruler line on the segment basis. User also has to provide the point distribution information during this step for each of the segments.

Surface Manipulation

TIGER generates the surface grid with TFI for each of the surfaces. It may not, however, be the most favorable grid to the user. Therefore, TIGER features the interactive surface manipulation, which allows the user to manipulate the surface with Bezier's curves, elliptic solver, averaging relaxation, and other techniques. A graphic panel with buttons and counters is provided for the user to access those functions easily. To date, such surface manipulation is available for cascade surfaces; i.e. J–constant surfaces. A user may localize the manipulation to an interactively defined zone.

GRID GENERATION

TIGER generates the initial grid by default with the algebraic TFI technique for each surface patch and sub–block. The axiom for dividing the surface patches and volume blocks is decided automatically by an algorithm tracing the critical indices. TIGER converts all of the coordinates into cylindrical coordinates; it matters not if the geometry definition is expressed in Cartesian coordinates or cylindrical coordinates. The reason for keeping the coordinates in a cylindrical system is due to the physical axisymmetricity of the turbomachinery. To generate the surface patches on the hub, for example, if we do the transfinite surface interpolation based on the information from the four boundaries of the surface, we, very likely, are going to loose the geometry definition of the hub, and the surface will turn out to be a surface composed of various "dips" and "bumps". This is definitely not the one we are looking for. It takes additional procedures, such as projection, to bring the transfinite surface back to the axisymmetric hub surface. However, with the transfinite interpolation done in the cylindrical coordinates, the result is very satisfactory. As stated above in the surface manipulation, TIGER also carries the elliptic grid generation algorithm. This is done by transforming the grid from (Z,R,θ) into (m,σ), and, after the elliptic iterations, the grid is transformed inversly back to cylindrical coordinates.

Similar methodology applies to the volume grid generation. The default algebraic grid by TFI is also done in cylindrical coordinates. However, if the grid somehow shows negative Jacobians due to the complexity of the geometry design, 3D Laplace/Poisson elliptic iterations may be performed to smooth out the grid and eliminate the negative Jacobians.

EXAMPLES

To date, TIGER has generated grids for various configurations such as single rotation propfans, counter-rotation propfans, ducted propfans, rotors-stators, and marine propellers. Three configurations are presented in this abstract as examples for internal, external, and internal-external flow fields, respectively.

The first example is the NASA Lewis single-stage transonic axial flow ROTOR-67 configuration with 22 rotor blades and 33 stator blades. This is an example of an internal flow field. The flow field is decomposed into two blocks, with a 49x21x25 grid for the rotor block, and a 45x21x17 grid for the stator block. HH grid type is used due to the physical domain. Figure 3-a is the solid image for this geometry. Figure 3-b is the grid mesh behavior on K=1 surface. Figure 3-c is the J= Jmax surface, i.e. the grid mesh on the outer shroud. Note that there is discontinuity between the rotor blocks and the stator blocks. This is due to the fact that these blocks will be rotated against each other to simulate the physical rotation; it is not necessary to link the grid between these two blocks since the Euler flow solver TURBO[14] developed by Mississippi State University will link the grid lines at each time step.

The second example is an external flow field case, a GE counter-rotation propfan with F4-A4 blade design with 8x8 blade count. The physical domain is decomposed into two blocks with 61x36x16 grid points for the front blade block and 61x36x16 grid points for the rear blade block. Figure 4-a is the solid image for this configuration. Figure 4-b is the grid mesh behavior for K=1 surface. Figure 4-c is the surface grid on the hub, namely, J=1 surface.

The third example is the NASA Lewis 1.15 Pressure Ratio Fan with 12 rotor blades and 32 stator blades, which is an internal-external flow field. This geometry is decomposed into four blocks with an approximate grid size of 200,000 grid points for a passage. A narrow gap between the rotor and the lower surface of the duct is simulated with 4 grid cells in between. Figure 5-a is its solid image. Figure 5-b is the grid mesh for K=1 surface, with a closeup image in figure 5-c. Note that the axial grid lines between the rotor blade tip and the lower duct surface remain near the duct surface and spray out from the leading edge and trailing edge of the duct. The reason for keeping grid lines from spraying off the duct surface before they leave the duct is that it will be easy to generate a viscous grid without too much user interaction.

REFERENCES

1. Soni, B.K., *"GENIE: GENeration of Computational Geometry-Grids for Internal-External Flow Configurations"*, Proceedings of "Numerical Grid Generation in Computational Fluid Machanics '88", Pineridge Press, pp. 915-924, 1988.

2. Soni, B.K., and Dorrell, E.W., *"INGRID Interactive Geometry-Grid Generation for Two Dimensional Applications"*, AEDC-TR-86-49.

3. Dorrell, E.W., and McClure, M.D., "*3D INGRID: Interactive Three-Dimensional Grid Generation*", AEDC-TR-87-40.

4. Soni, B.K., McClure, M.D., and Mastin, C.W., "*Geometry and Generation in N+1 Easy Steps*", "*The First International Conference on Numerical Grid Generation in Computational Fluid Dynamics*", Landshut, W. Germany, July 1986.

5. Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W., *Numerical Grid Generation: Foundations and Applications*, North Holland, 1985.

6. Thompson, J.F., "*Program EAGLE-Numerical Grid Generation System User's Manual*", Vols. II, III, AFATL-TR-87-15, March 1987.

7. Soni, B.K., Thompson, J.F., Stokes, M., and Shih, M.H., "GENIE^{++}, EAGLEView and TIGER: General and Special Purpose Graphically Interacitve Grid Systems", 30th Aerospace Sciences Meeting & Exhibit, AIAA-92-0071, Reno, Nevada, January 1992.

8. Soni, B.K., and Shih, M.H., "TIGER: Turbomachinery Interactive Grid GenERation", Proceedings of the Third International Conference of Numerical Grid Generation in CFD, Barcelona, Spain, June 1991.

9. Soni, B.K., and Shih, M.H., "*TURBOGRID: Turbomachinery Applications of Grid Generation*", 26th AIAA/SAE/ASME Joint Propulsion Conference, AIAA-90-2242, Orlando, Florida, July 1990.

10. Shih, M.H., "*TIGER: Turbomachinery Interactive Grid genERation*", Master's Thesis, Mississippi State University, December 1989.

11. Overmars, M.H., "FORMS: A Graphical User Interface Toolkit for Silicon Graphics Workstations", Department of Computer Science, Utrecht University, Utrecht, Netherlands, 1991.

12. Yoon, Y.H., "Enhancements and Extensions of EAGLE Grid Generation System", PhD Dissertation, Mississippi State University, May 1991.

13. Mortenson, M.E., "Geometric Modeling", John Wiley & Sons, 1989.

14. Janus, J.M., "*Advanced 3-D CFD Algorithm for Turbomachinery*", PhD Dissertation, Mississippi State University, May 1989.

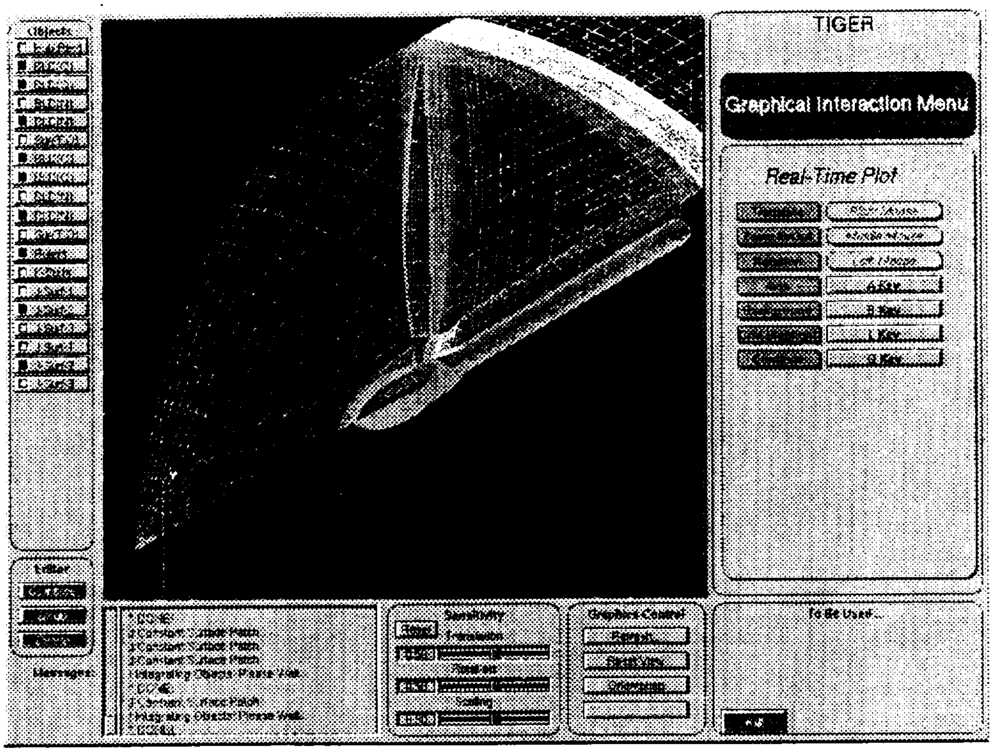

Figure 1–a TIGER's main panel

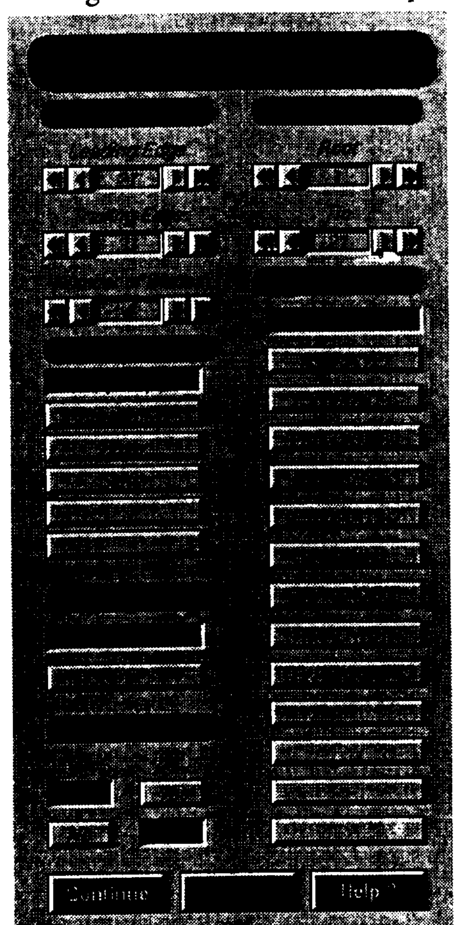

Figure 1–b Option panel for blade information inputs

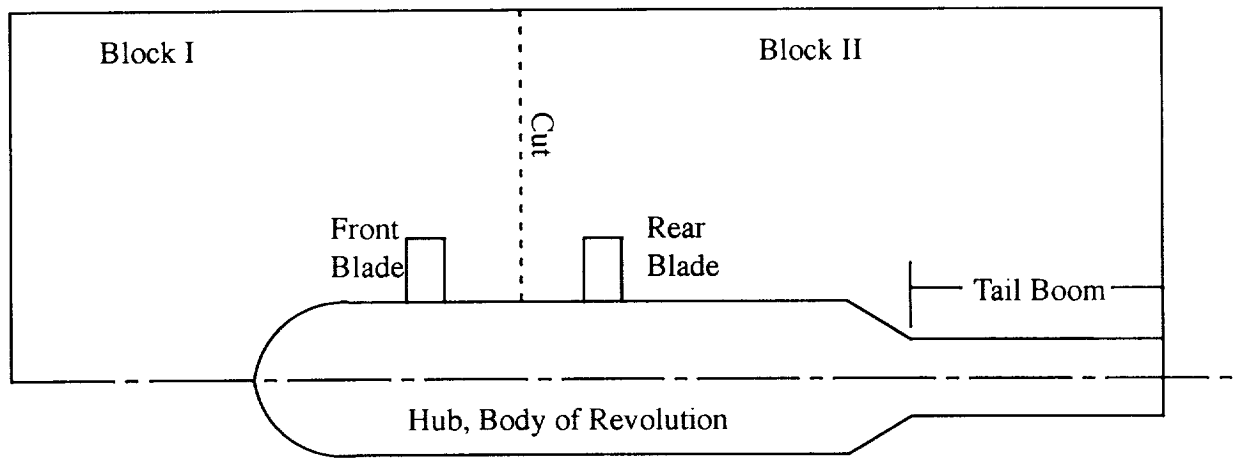

Figure 2-a Two-Block Propfan Application

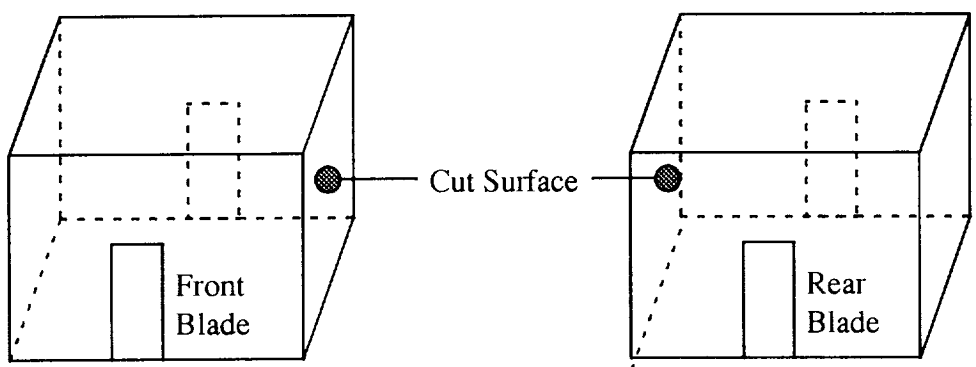

Figure 2-b Typical Computational Domain of a Two-Block Propfan Passage

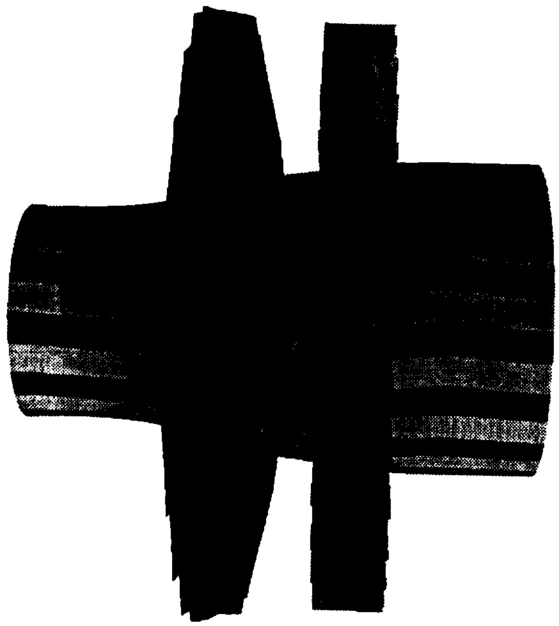

Figure 3-a ROTOR-67 Solid Image

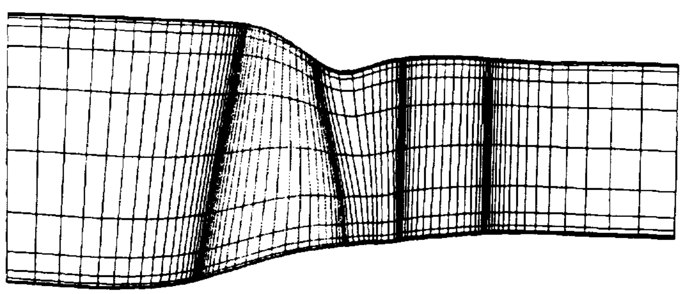

Figure 3-b K=1 Surface for ROTOR-67

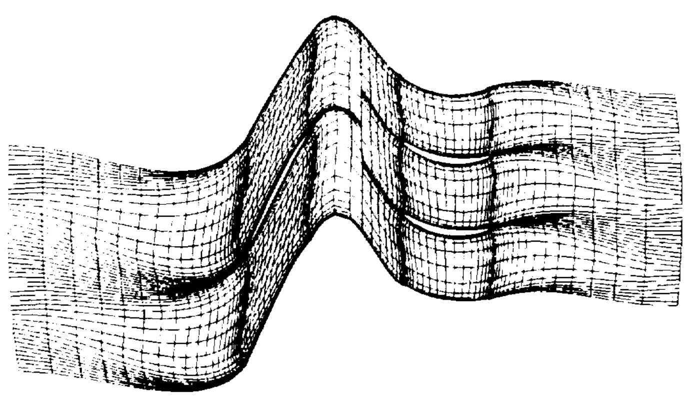

Figure 3-c Mesh on the Shroud

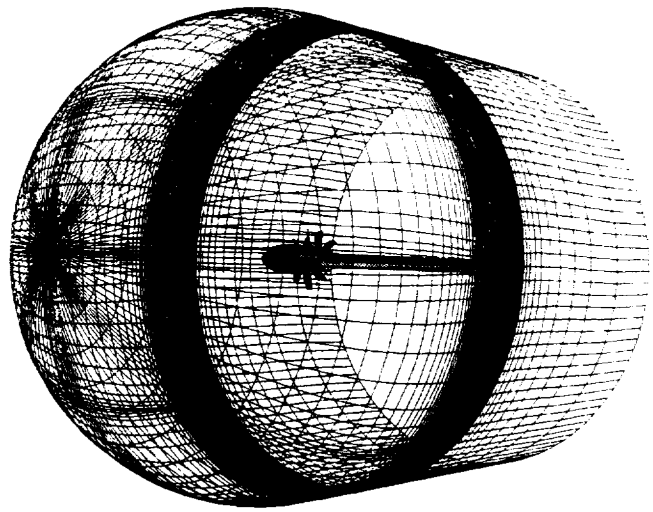

Figure 4-a Counter-Rotation Propfan

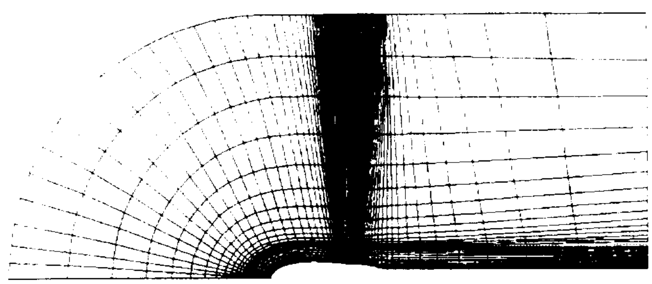

Figure 4-b Grid Surface on K=1 Surface

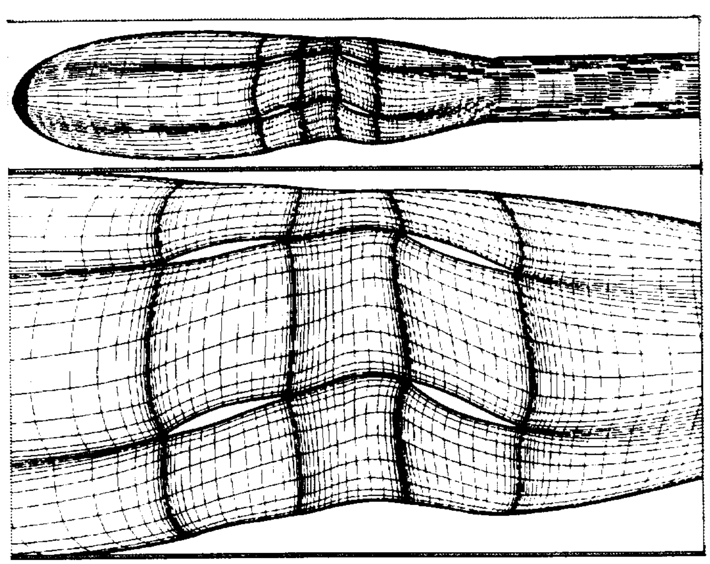

Figure 4-c Surface Grid on the Hub

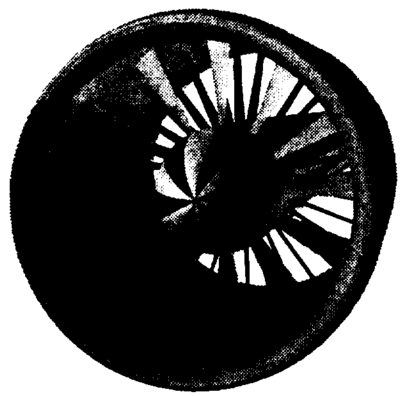

Figure 5-a 1.15 Pressure Ratio Fan

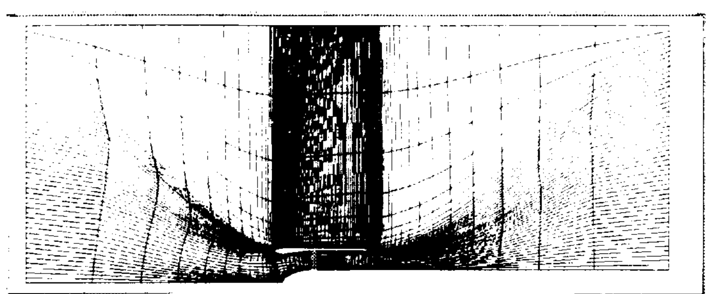

Figure 5-b K=1 Surface Grid

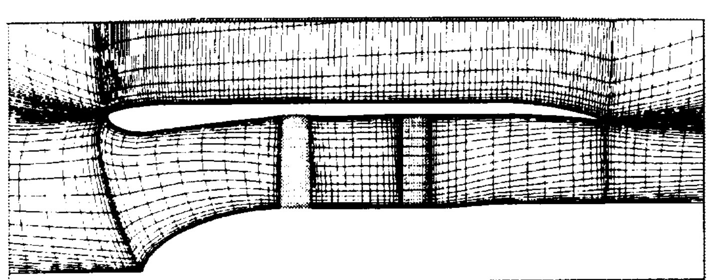

Figure 5-c Closeup Grid for K=1 Surface