PROJET CASTOR

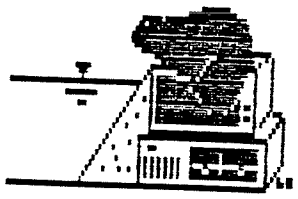
GRID GENERATION SYSTEMS FOR TURBOMACHINERY APPLICATIONS

R. CAMARERO
B. OZELL
M. REGGIO
A. GARON

EPM/RT-86/32

ECOLE POLYTECHNIQUE DE MONTREAL

Case Postale 6079 Succursale "A"
Montréal, Québec H3C 3A7
Canada



Août 1986

TABLE OF CONTENTS

1. INTRODUCTION

The complexity of the geometries encountered in practical fluid flow applications is such that a well defined need has been identified for the treatment of complex shapes. This is particularly the case of the numerical simulation of three-dimensionnal flows in turbomachinery components. Historically one can trace such approaches to the conformal mapping techniques or more recently to algebraic transformation techniques. However a truly modern methodology for the handling of arbritray three-dimensional geometries in a generalized approach, dates from the early seventies with the work of Professor Joe F. Thompson on body-fitted grid generation. Although originally intended for finite differences methods, these grid generators apply equally well to finite volumes and finite element methods.

Such coordinate systems are called body-fitted or body-conforming curvilinear grids. Their essential feature is that yield coordinate curves which are aligned with the domain boundaries. There are two major advantages. The first is that the numerical scheme for the governing equations are carried out on a rectangular mesh. There results a simpler and more accurate algorithm since boundaries coincide with coordinate grids, and no interpolation is required. The second advantage is that the geometric complexity, through the transformation, is imbedded into the coefficients of the governing equations. This allows the possibility of writting generalized codes applicable to a variety of different geometries. This results in a great saving in the code development effort.

2. BASIC CONCEPTS

2.1 WHAT IS A GRID

Presented simply, a grid is a method to organize a domain for calculation purposes. This is achieved by the mathematical concept of parametrization. That is the mapping of the physical domain unto a parameter space which is rectangular by construction. In this manner there is a unique association of points, curves or surfaces in the parameter space with their images in the physical space.

This results in a series of curves which span the entire domain in an orderly fashion. Each curve is identified by one value of a parameter (or more graphically each corresponds to one line in parameter space). So that a given point in space can be identified or more imaginatively, can be reached by displacements along these curves. It is intuitively obvious that the number of these, depends on the dimension of the domain.

The parametrization allows also to identify the neighbors of points or lines as required for computational purposes.

It is clear that a given domain can be organized in many different ways, the simplest being the cartesian system. Other possibilities come to mind such as the classical polar, cylindrical, or spherical coordinate systems. All of these provide a means of associating (or mapping) a point in space to a set of parameters.

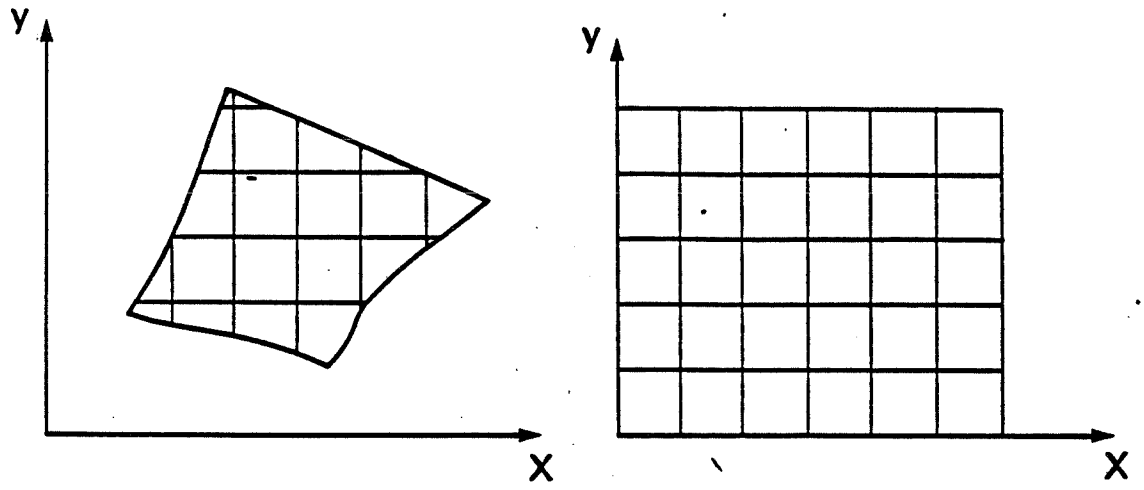


Fig. 1a Cartesian coordinate grid

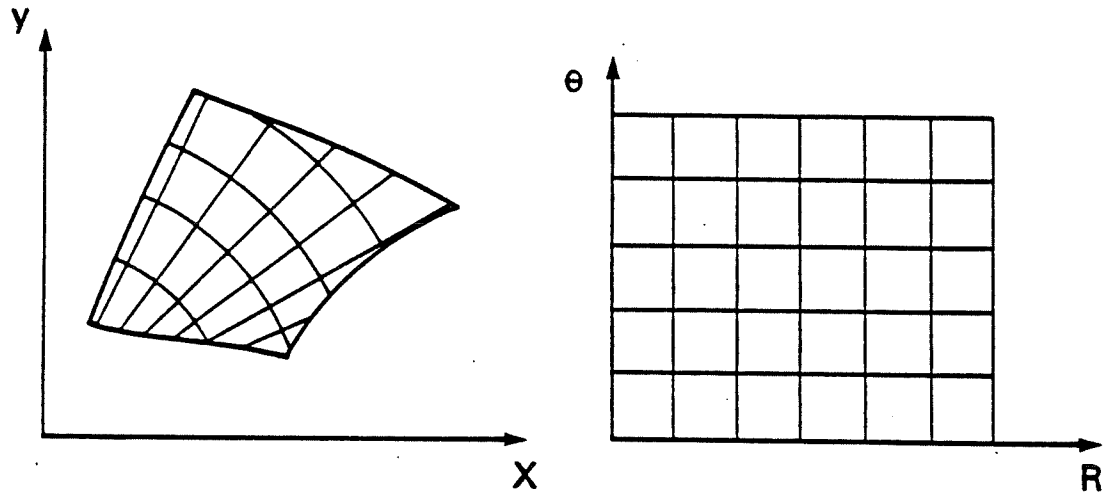


Fig. 1b Polar coordinate grid

From experience we know that the choice of a coordinate system is related to the geometry of the problem. So that there is a need to have a set of relations between the physical space and the chosen parametrization. For some very simple geometries (cylindrical, toroidal etc) there exist such relations in analytical forms. However, for most applications, this is not possible and the curvilinear system must be generated numerically. In that case, one no longer has a system of coordinate curves which span the physical domain, but rather a set of points or nodes which approximate these. These nodes are ordered according to a corresponding lattice in a computational space.

One can think of a discrete grid as the set of nodes resulting from the intersection of curvilinear curves of different families.

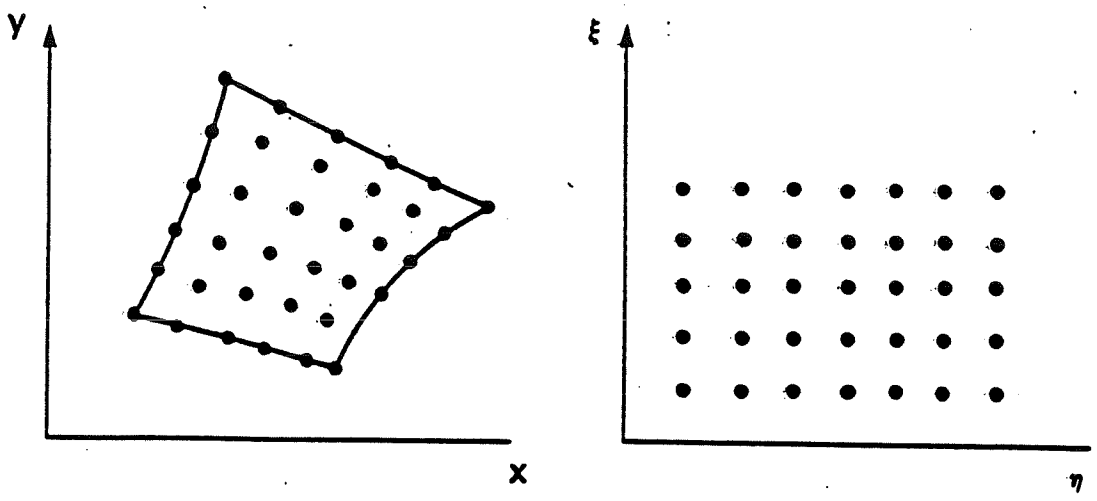


Fig. 2 Discrete grid

It is seen that the discretization of the physical domain can be as fine as desired by increasing the number of nodes. The organization of the nodes is achieved by their numbering

according to the parametrization so that the neighbors of a given point are immediately known.

2.2 CURVILINEAR GRIDS

The basic characteristic of a curvilinear coordinate system is to have the coordinate curves (or surfaces) conform to (or align with) the boundaries. The degree of conformity varies from exact at the boundaries where these must coincide, to some global alignment in the interior of the region. To fix ideas more precisely, the annular domain between two concentric circles is used.

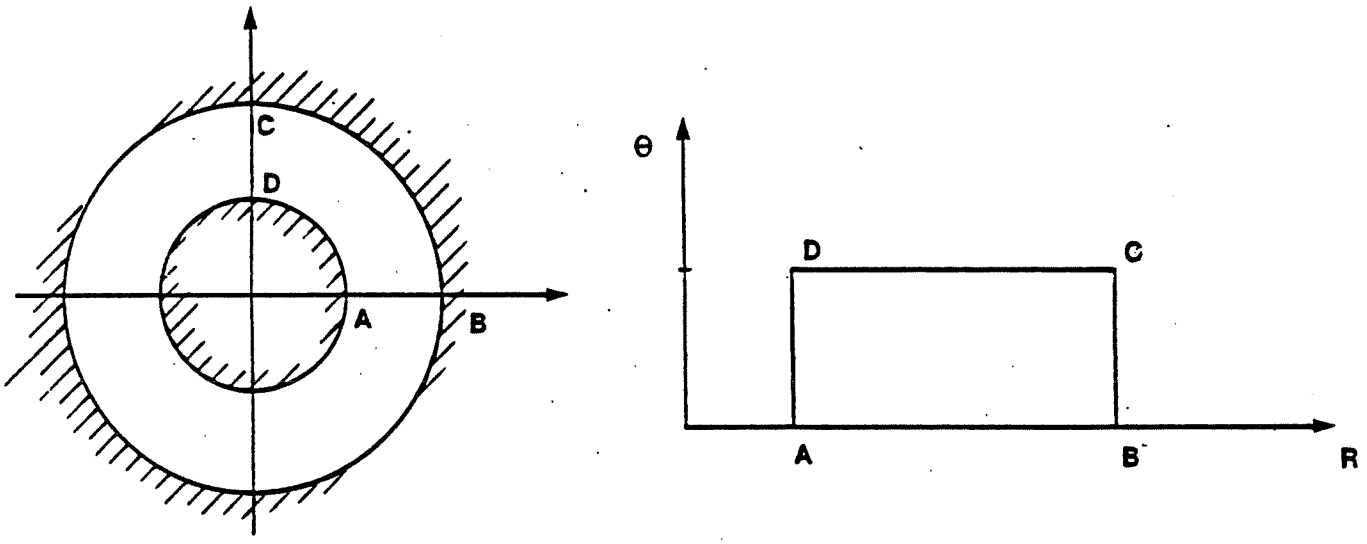


Fig. 3 Polar curvilinear grid

An appropriate coordinate system for this problem is the familiar polar coordinate system. The mapping between the physical coordinates (x,y) and the curvilinear coordinates (r,θ) can be expressed by analytical expressions

$$x = r \cos \theta$$
$$y = r \sin \theta$$

By varying one of the two parameters while keeping the other constant generates two sets of coordinate curves. These are respectively concentric circles and radial lines which conform to the boundaries. Furthermore, for the value of the parameter $r = r_1$ the line AB maps into the radial line A'B'. Similarly, the other lines BC, CD and DA yield coordinate curves B'C', C'D' and D'A' which coincide exactly with the boundaries of the physical domain.

This idea can be extended to general configurations in 2-D and 3-D where complex shapes in physical space can be mapped into rectangles in the parameter space. This latter is usually called the computational space.

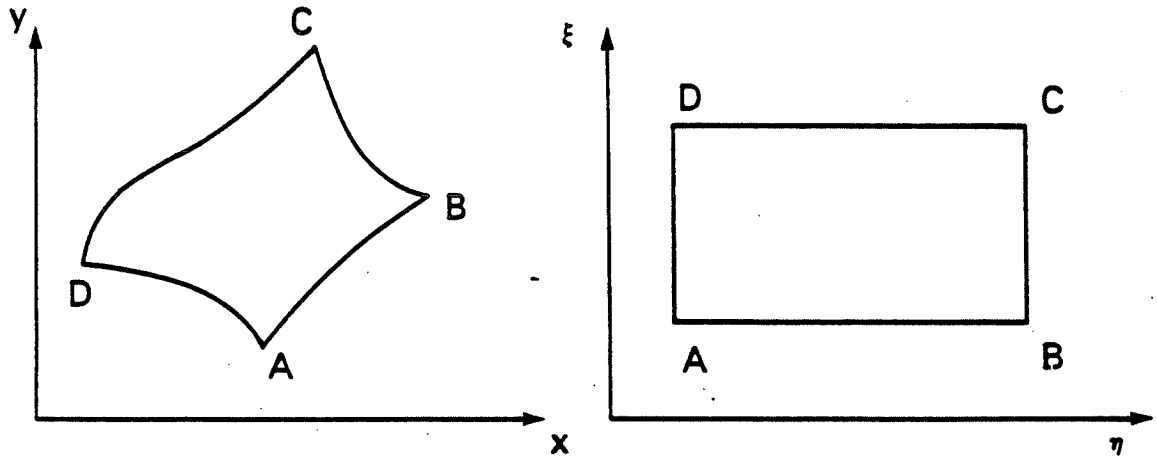


Fig. 4 Mapping from physical to computational space

For example, to generate a curvilinear coordinate system for the domain in Fig. 4, one seeks a functional relationship between the physical space (x,y) and the parameter space (ξ,η)

$$x = f(\xi,\eta)$$
$$y = g(\xi,\eta)$$

The difference with the previous example and other familiar systems is that for complex boundaries the functions f and g cannot be expressed analytically. It is required that f and g map lines of constant ξ into curves which conform to the boundaries AB and CD and similarly, that lines of constant η map into curves which conform to the other set of boundaries B'C' and A'D'. Furthermore, it is required that f and g map AB ($\eta=\eta_1$) exactly into A'B' and similarly for the other boundaries.

It is finally required that the correspondence between a point (x,y) and its image (ξ,η) and vice-versa be unique. The result is a set of curves in physical space which for the first family verify

$$\xi(x,y) = \xi_i \quad \text{where } \xi_1 \leq \xi_i \leq \xi_m$$

and for the second family

$$\eta(x,y) = \eta_j \quad \text{where } \eta_1 \leq \eta_j \leq \eta_n$$

The values (ξ_1,ξ_m) and (η_1,η_n) define the range of the parameters ξ_i and η_j respectively and correspond to the boundary curves. The values of ξ_i and η_j vary monotonically within this range.
The actual values of these depend on the chosen parametrization and are in a sense arbitrary. One reasonnable choice is to normalize the parameters giving

$$0 \leq \xi_i \leq 1$$
$$0 \leq \eta_j \leq 1$$

Another possibility and perhaps better suited to programming considerations is to use a parametrization based on the integer values of the nodes, giving

$$1 \leq \xi_i \leq m$$
$$\text{and} \quad 1 \leq \eta_j \leq n$$

Essentially, the concept of curvilinear grids can be cast in the form of a boundary value problem. An analogy is presented to illustrate this. One can imagine a membrane (for 2D problems) of rectangular form with a cartesian grid laid upon it. This membrane is placed upon the physical domain and stretched so that the boundaries match. The membrane is then pegged at the boundary nodes. The original grid on the membrane is deformed and yields a curvilinear grid.

This analogy is easily extended to 3D by replacing the membrane by block of "foam" and the remainder of the procedure is identical.

2.3 GRID CONFIGURATIONS (TOPOLOGY)

In the transformed space the region is bounded by pairs of opposite boundaries. In two dimensions these are two pairs of line segments, whereas in three dimensions these are three pairs of planar segments. The coordinate curves join corresponding nodes in a given pair of boundaries.

In the physical space, the domain is also bounded by boundaries but their grouping in pairs is not unique and in some degree is arbitrary. Since the sense of the grid is dictated by this grouping, the manner in which the correspondence between boundaries in physical and parameter space is realized, determines the grid topology.

These concepts will be presented in an orderly fashion proceeding from simple to more complex configurations. These are simply connected regions, multiply-connected regions and composite grids.

Simply connected Regions

The first step in the grid design is to identify the four boundary curves that map into the corresponding sides of the rectangular domain. For the various examples used so far this is obvious. If however, a discontinuity exists along one of these, then there exist several ways to logically connect the boundaries. To illustrate one may consider an L-shaped domain, such as a rear-facing step.

In the fist configuration shown in Fig. 5a, the grouping is as follows. Boundary ABCD is grouped with FE and boundary AF with DE. This results in grid lines with run from AF to DE for the first family and from ABCD to FE for the second family.

Although there are two discontinuities, points B and C, these do not appear in the grid. This can be appreciated from the analogy presented in the previous section and will be more formally established through the basic smoothing properties of the grid equations in latter sections.

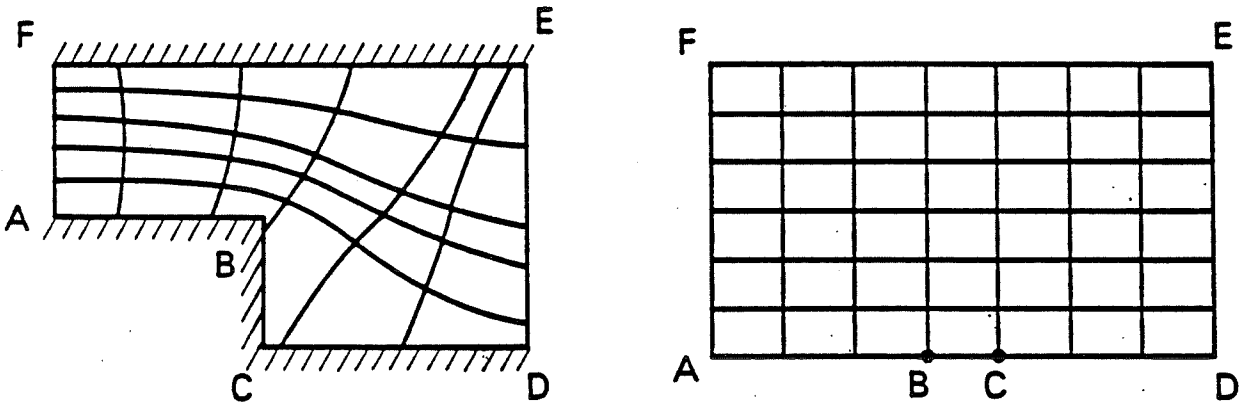


Fig. 5a Grid topology I

A second possibility is to pair ABC with FED and AF with CD. A similar discussion applies.

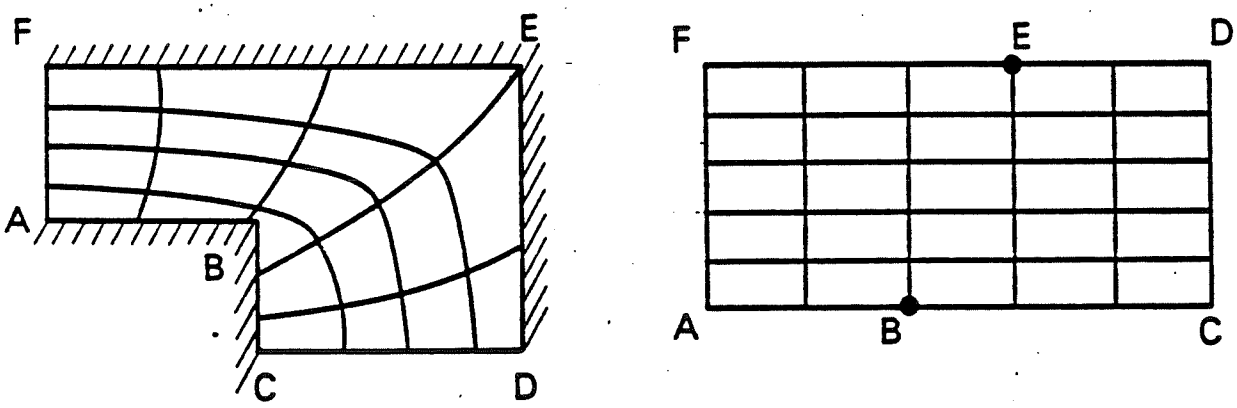


Fig. 5b Grid topology II

Composite grids

Yet another possibility is to have a composite grid generated by dividing the domain into two (or more) simple subdomains. Then within each of these a distinct grid is generated. In the present example the subdomains could be ABGF and GCDE or ABHEF and BCDH if two subdomains are used. The following three subdomains could have also been used ABGF, BHEG and BCDH. Such composite grids must then be matched along some common boundaries.

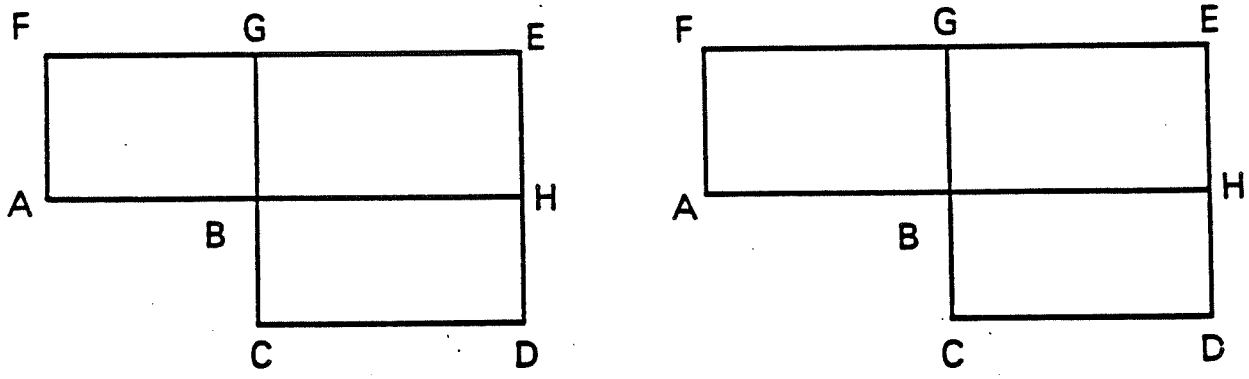


Fig. 5c Grid topology III

A number of properties can be required. First, it is required that the grid lines be continuous, that is the connecting boundaries are defined with the same nodes. This is to avoid the situation illustrated in Fig. 6.

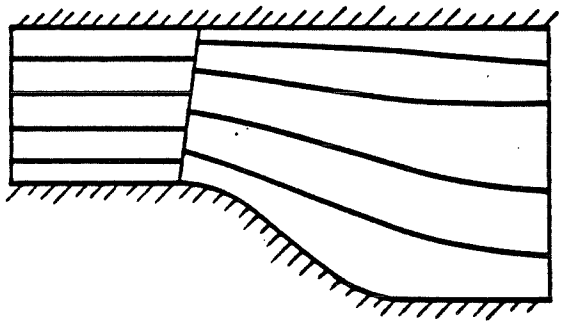


Fig. 6 Discontinuous coordinate curves

Then it can be required that the grid curves have continuous slopes. This is to avoid the case illustrated in Fig. 7.

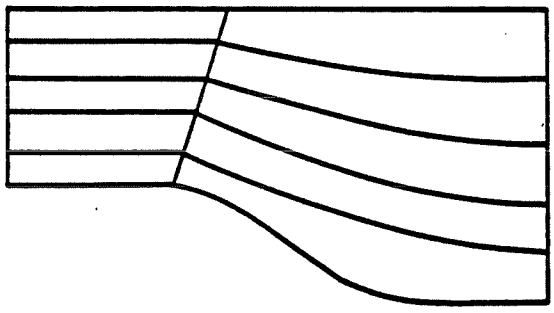


Fig. 7 Slope discontinuous grid

These few examples illustrate the need to relate the grid design to the problem at hand. Clearly the grid in Fig. 5a should be used for a problem of the flow over a step, whereas the grid in Fig. 5b is best for the flow around a square

corner.

The necessity for grid continuity is also dependent on the problem but more so on the method of solution used to solve the problem. For example, in finite element techniques one could use discontinuous grids. Finite volume methods will work with discontinuous first derivative whereas most finite difference scheme require C^1 continuity for the grid.

Another interesting configuration arises when the physical boundary is closed and continuous such as a duct with a circular cross section. There are no natural points at which to break this boundary so as to group opposite side into pairs. One then must break it at arbitrary points, equally spaced (unless there is some other prevailling approach) A, B, C and D.

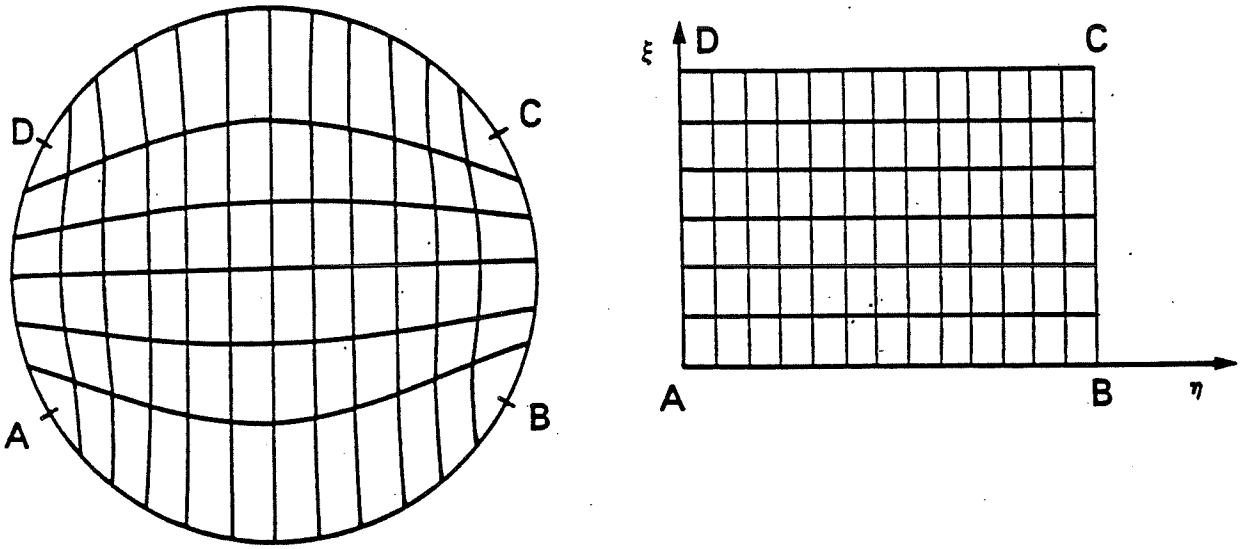


Fig. 8 Grid for a domain with continuous boundary

The grid curves will join sides AB to DC for the first family and AD to BC for the second family. This creates a special node or control volume with two sides having the same slope at these break points. The treatment of this node requires special care since the Jacobian of the transformation vanishes.

Multiply connected regions

In some applications, obstacles are encountered in the domain. This leads to several solutions which depend on the physics of the problem. If there are identifiable points (corners) on the obstacle, a composite grid approach can be useful. For example, with four such points the configuration below is appropriate.

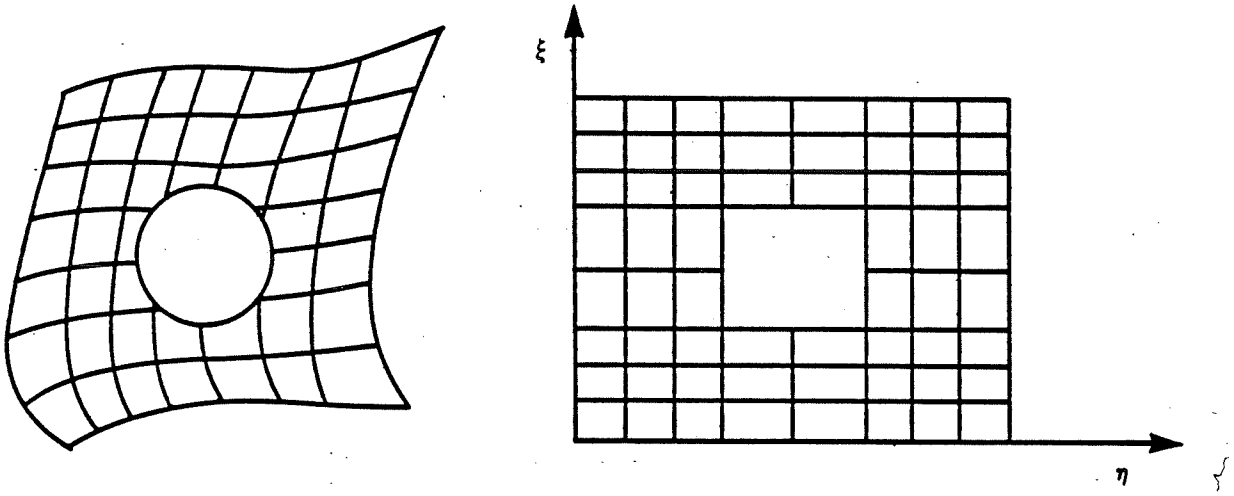


Fig. 9 Multiply connected domain with several "corners"

If there are two identifiable "corners" one might decide on the following two subdomains

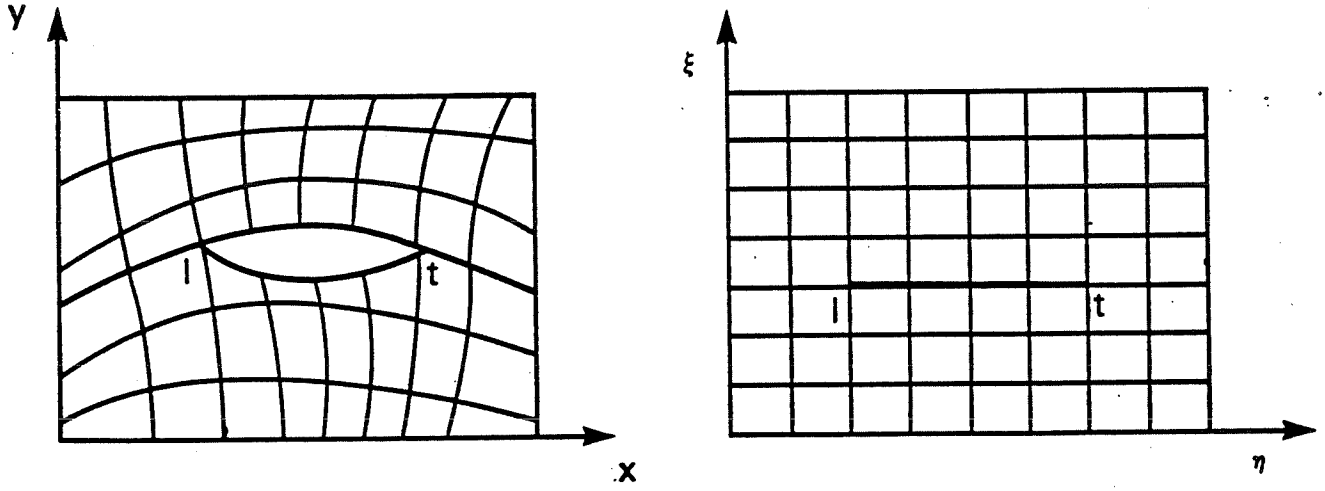


Fig. 10 Multiply connected domain with two "corners"

Of course the notion of corner points may be extended. These need not necessarily coincide with slope discontinuities but some special feature of the physical problem. Examples that come to mind are stagnation or flow separation points.

For these composite grids, the remarks of the previous section on grid continuity apply.

Finally, if only one corner point is identified the problem can be made simply-connected by the introduction of a fictitious boundary or branch cut as illustrated below

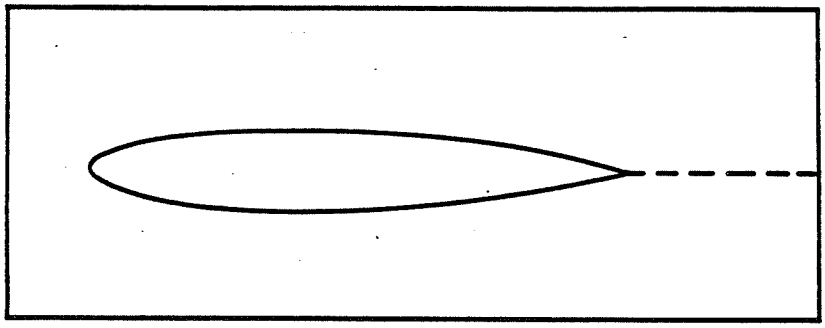


Fig. 11 Branch cut

2.4 APPLICATIONS TO TURBOMACHINERY

The presentation of the subject of grid generation has been carried out so far in rather general terms. At this point these concepts and ideas will be discussed in the particular context of turbomachines. Perhaps the most appropriate application of curvilinear grid is the transformation of the blade-to-blade- channel. The design of a good grid is a rather challenging task which can lead to a multitude of solutions according to the problem and the particular numerical scheme intended.

To concentrate on the aspects of grid generation, only the two-dimensional geometries are discussed at this point. Extensions to three dimensions are relatively straight forward.

Several types of grids are presented and these are related to the previous discussion on grid configuration. The physical domain in question is the following: a region with an obstacle, a blade profile.

This domain can be thought of as a surface of revolution, a cylinder say, which intersects the solid blades in a number of profiles. When this surface is developed one obtains the classical blade row which is a periodic repetition of one single blade-to-blade channel. For the subsequent flow simulation, it is necessary to place the inlet and outlet boundaries upstream and downstream, of the leading and trailing edges of the profile, respectively. This leads to the requirement of permeable, fictitious boundaries which are periodic. It is the manner by which these are defined that determines the grid configuration.

H - grid

In this type of grid two "corner" points are identified on the obstacle, i.e. the profile. These are the leading and trailing edges. From both of these two periodic (permeable) boundaries are extended in the upstream direction and downstream directions to the inlet and outlet sections respectively.

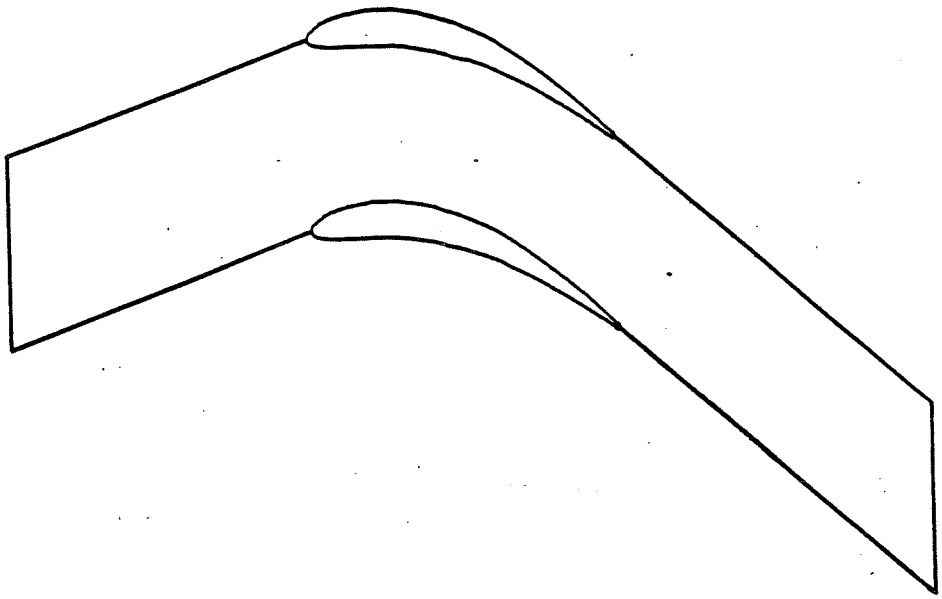


Fig. 12 Domain topology for an H-grid

These are usually straight lines whose angles are arbitrary but should preferably be related to some flow characteristic such as the inlet or outlet flow directions.

Similarly, their lengths should be related to some characteristic dimension of the domain such as the chord of the profile to allow for adequate imposition of boundary conditions at inlet/outlet sections.

So we have a simply connected region bounded by four sides, two of which possess discontinuities. Following the procedure outlined in the previous section, these are paired as follows: the inlet with the outlet and the suction side with the pressure side. The resulting grid yields one family of coordinate curves which are roughly aligned with the flow streamlines, and the other are in a roughly normal direction. It should be noted that the upstream and downstream periodic boundaries are identical and are matched. Consequently the inlet and outlet sections must be vertical for this type of grid. As it will be seen later this can, in some highly cambered profiles, lead to poor grids. This can be remedied by other types of grids.

I - grid

This type of grid is quite similar to the H - variety except for the position of the inlet/outlet, which can now be placed in any orientation. The consequence of this is that the upper and lower periodic boundaries no longer match identically.

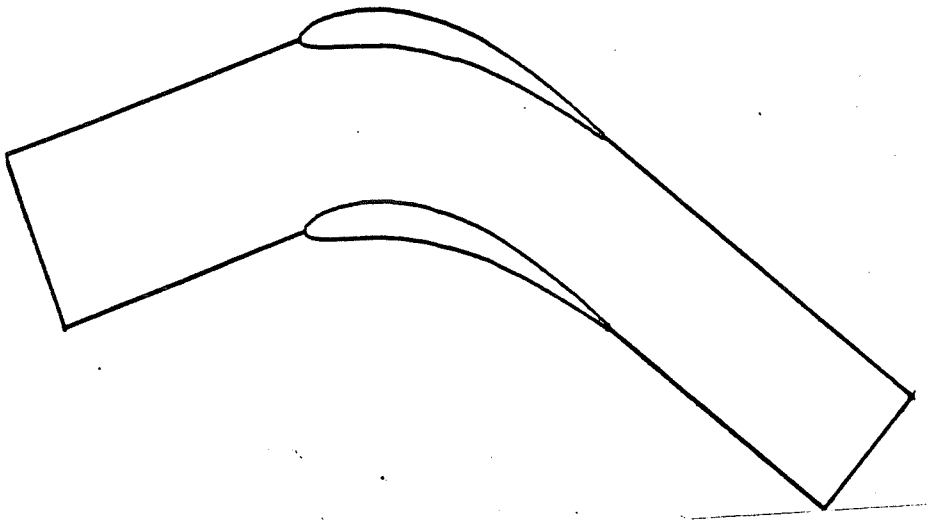


Fig. 13 Domain topology for an I-grid

The resulting grid has a family of coordinate curves running from the inlet to the outlet in roughly the stream direction. The main difference with the H - grid is that a coordinate curve may run from a periodic boundary to a solid boundary as shown in Fig. 14.

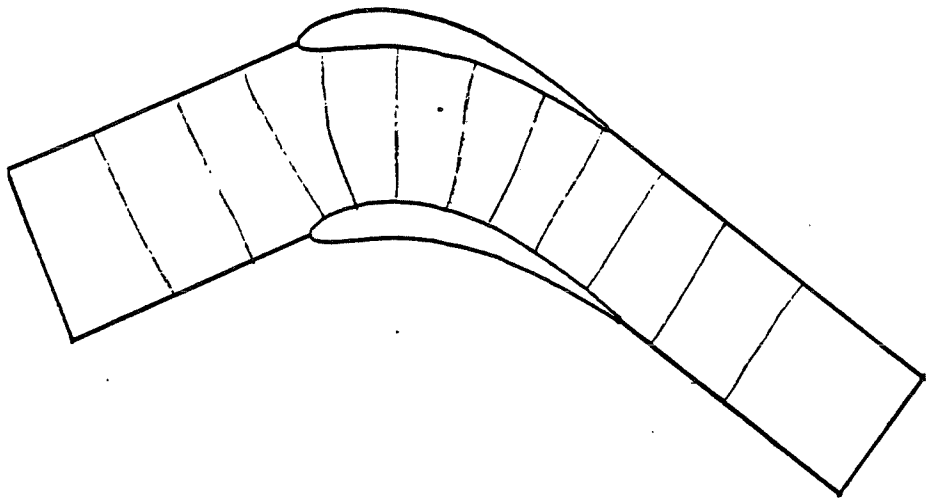


Fig. 14 Region periodicity

The grid boundaries in the H - configuration are line periodic whereas now it is an entire region which is periodic. This does not pose any particular problem at the grid generation level but is important when treating boundary conditions in the subsequent flow simulation.

Essentially one may think of an I - grid as an H - grid where the nodes are allowed to slide freely along the boundaries from the solid to the periodic parts and vice-versa. With the result that for highly cambered profiles, the stream-wise coordinate curves tend to fit the body more closely, particularly in the leading edge region where H - grid tend to be poor.

C - grid

One common characteristic of both previous grids is that in the vicinity of a concave discontinuity such as the leading edge the grid lines along this boundary tend to be distributed away from that boundary. This is not desirable since one would like to concentrate coordinate curves towards the wall in these regions. A new type of grid, called the C - grid, is presented. The domain is constructed with only one corner point which is the trailing edge. The boundaries are made up of the profile itself and other imaginary permeable boundaries. The first of these is a line or curve extending downstream of the leading edge and making a cut into the domain.

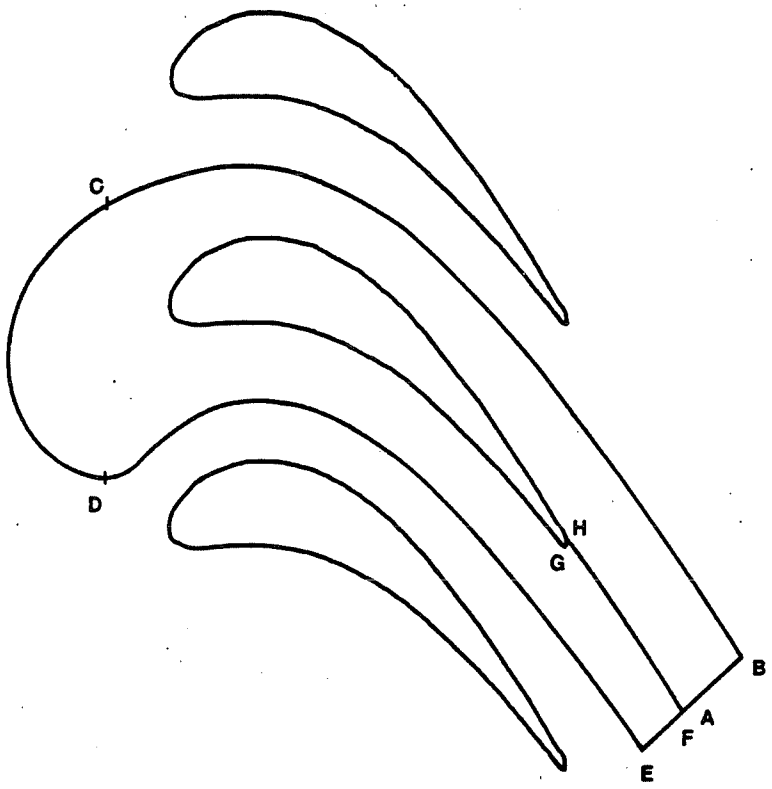


Fig. 15 Domain construction for a C-grid

The other boundary is in the form of the letter C and wraps around the profile or obstacle. This curve starts from the exit of the domain in the upstream direction, around the

leading edge into the inlet of the domain and finally downstream to the exit but on the opposite side of the cut. This curve may be quite arbitrary from the point of view of grid design. However, for the solution of the flow equations, it is necessary to exercise some care so that the boundary conditions may be imposed in some meaningfull way. This leads to the division of the curve into three parts. Two periodic curves BC and ED which run midway through the blade-to-blade channel

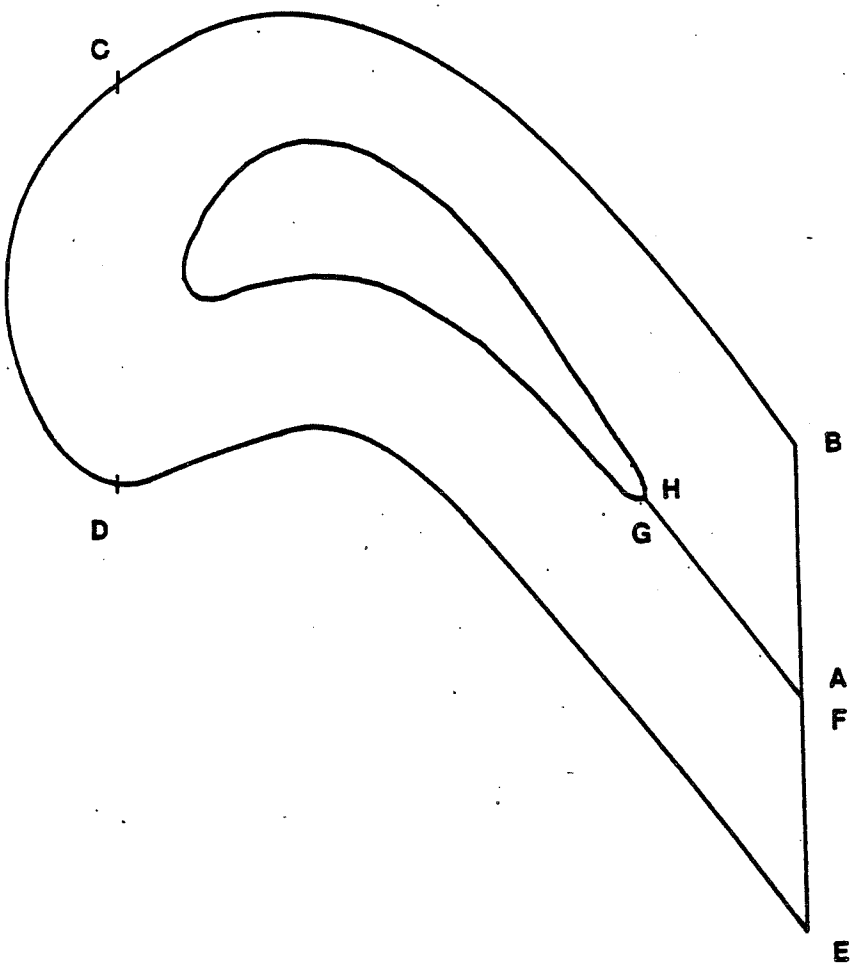


Fig. 16 C-grid with line periodicity

These then join an inlet curve CD with matching slopes at points C and D. This last curve is arbitrary but must be such that inflow conditions can be applied along it. Along the cut there are in fact two curves AH and FG. This may be straight or curved and possibly could match the periodic curves BC or ED. Finally it is noted that these are not necessarily of equal lengths.

For this configuration the pairing of boundaries is as follows: AB with FE, and BCDE with AHGF. This results in two families of coordinate curves; one running C-like from outlet around the leading edge to outlet, the other in a quasi-radial fashion from the profile and cut, to the periodic boundary or

outermost "C". The mapping of this domain in the transformed space is shown in Fig. 17.

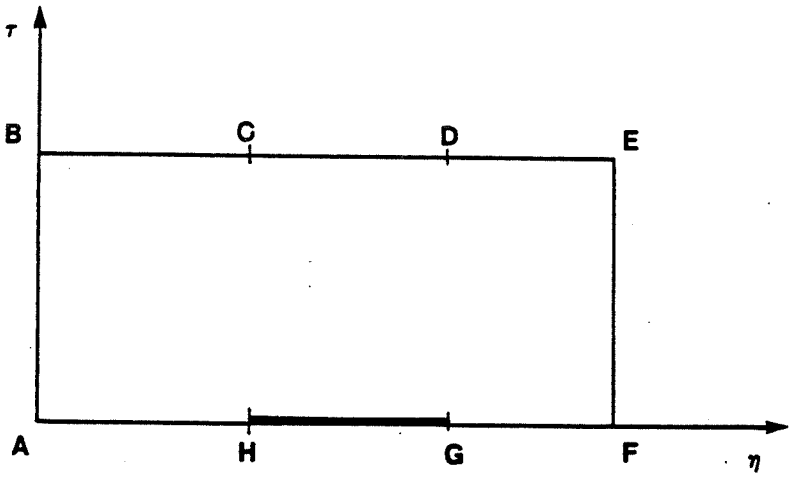


Fig. 17 Application of a C-grid to computational space

A variation on this configuration arises when the outlet boundaries AB and EF are not along the same vertical line.

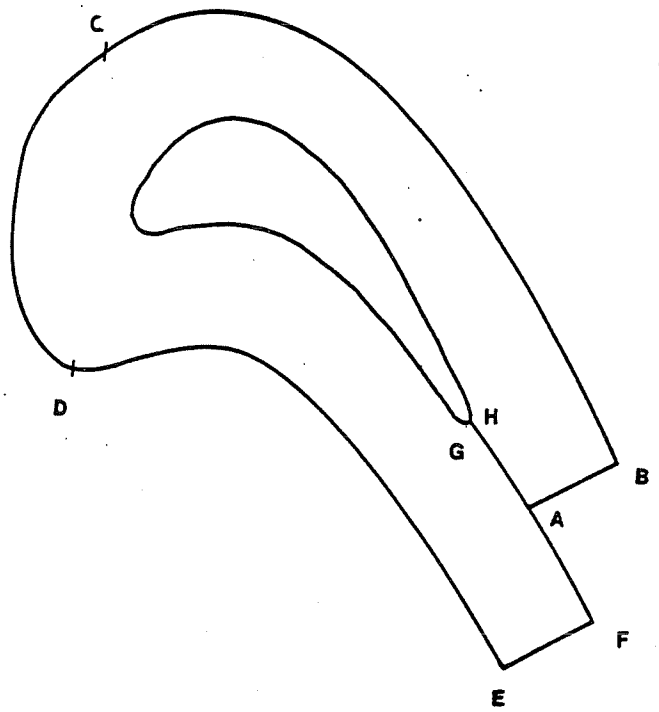


Fig. 18 C-grid with region periodicity

The periodic boundaries are no longer line periodic but rather region periodic in a similar fashion to the passage from H - to I - grids.

O - grid

The final type of grid is obtained when no corner point is identified. The domain may be thought of as a band around the profile. The domain is bounded by a closed curve made up of four parts; an inlet and outlet and two periodic curves. In this instance again this requirement is for the imposition of boundary conditions for the flow equations.

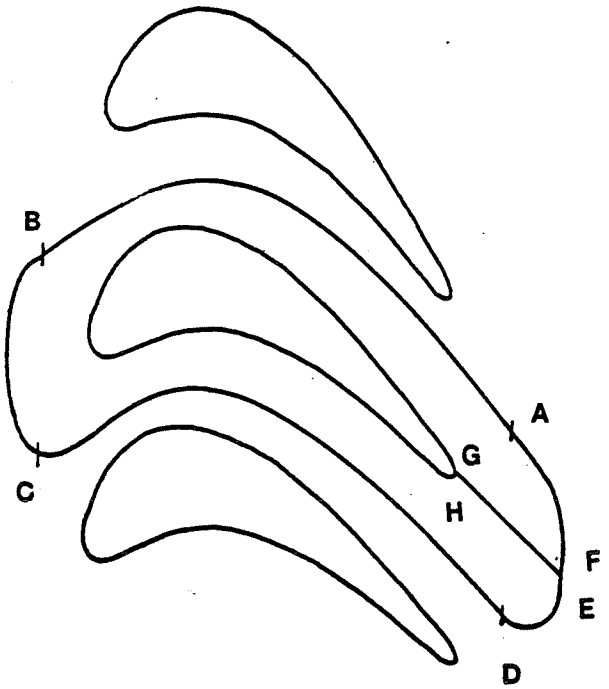


Fig. 19 Domain construction for an O-grid

One also requires an additional fictitious boundary which cuts the domain from the leading edge. The solid profile boundary FG is paired with the closed boundary ABCDE to yield the first family of coordinate curves. These run from the profile in a quasi-radial fashion toward to the O - curve. The fictitious boundary along the cut is multivalued and is matched with itself, GH with FE. This results in the second family of grid curves which form closed curves. These vary monotonically from the profile to the outermost O - curve which they match identically. In most cases special care will be exercised at the cut EF/GH to insure continuity and slope continuity of the coordinate curves. The segments AB and DC

are curve - periodic. The mapping in the computational space is shown in Fig. 20.

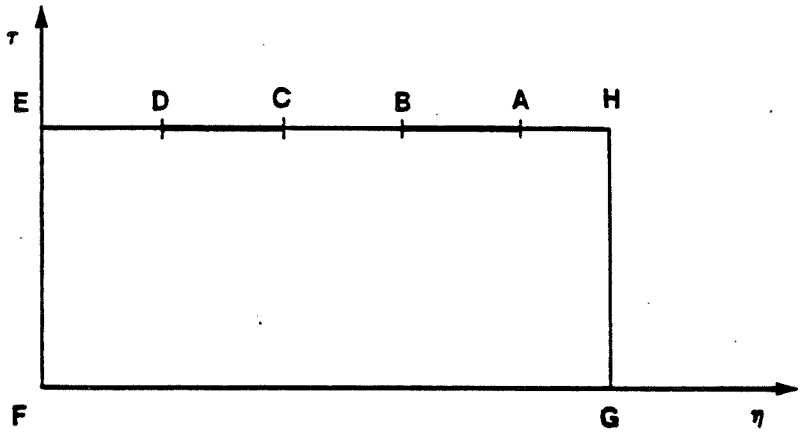


Fig. 20 Application of O-grid to computational space

3. GRID EQUATIONS

3.1 ELLIPTIC SYSTEMS

The basic ideas and concepts of curvilinear body-conforming grids have been presented in a rather qualitative fashion. The specific manner to actually obtains such grids is now presented.

The generation of a curvilinear coordinate system consists in evaluating, in the interior of a given physical domain, the location of the curvilinear coordinates. For a two-dimensional problem these are two families of curves along which one coordinate is constant while the other varies monotonically. Similarly, for a three-dimensional problem, the curvilinear coordinates are three families of surfaces along which one coordinate is constant while the other two vary. Upon each of these constant coordinate surfaces there lie two families of curves which represent a two-variable problem

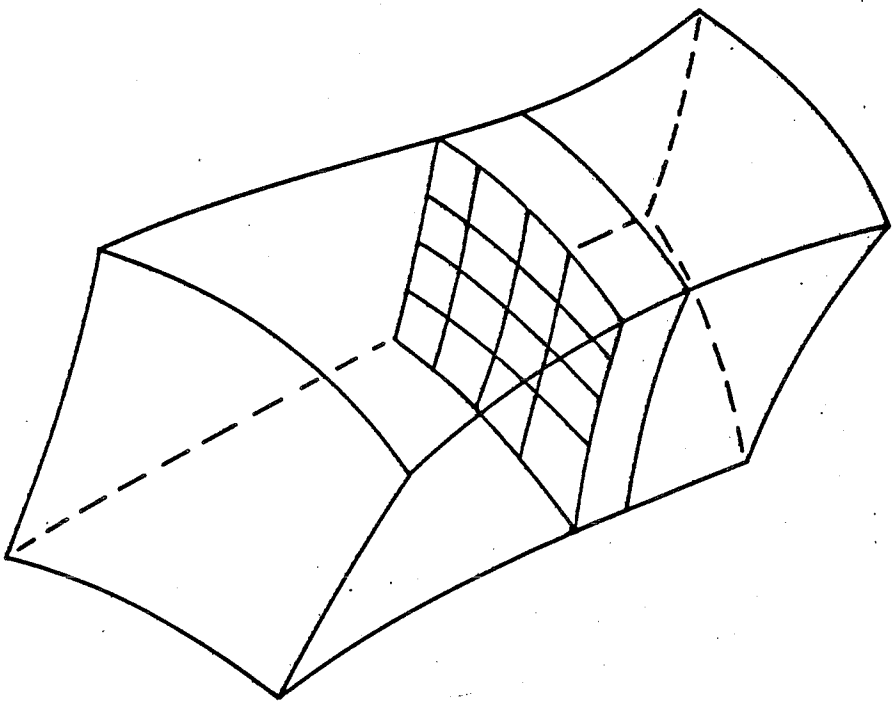


Fig. 21 Grid generation as a boundary value problem

When one member of each family is made to coincide with an appropriate side of the physical domain, the resulting system is called boundary - fitted or boundary - conforming. For a given four-sided physical domain, there are therefore four known coordinate curves: the first and last of each family.

Likewise in a three-dimensional problem, there are six known coordinates surfaces.

Several approaches can be applied to the problem of generating the internal coordinate lines (or surfaces). One very simple method is to interpolate between the boundaries. This consists in interpolating the coordinate values of corresponding nodes on a pair of opposite boundaries. This however, can lead to difficulties with certain geometries

i) coordinate lines leave the domain

ii) coordinate lines overlap

iii) discontinuities in the boundaries can propagate within the domain

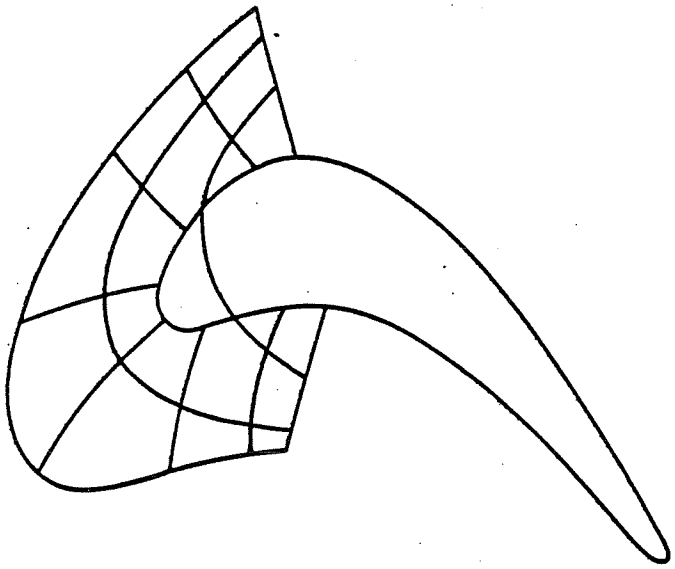


Fig. 22 Example of non-elliptic grid

These can be avoided by exploiting the well known extremum principles which characterize elliptic systems. More specifically, such systems will guarantee that the values of each of the coordinates families will vary monotonically from a minimum on one face to a maximum on the opposite face since extremum can only be attained on the boundaries. Also such systems exhibit smoothing properties that prevent the propagation and/or presence of discontinuities in the grid.

These essential features for coordinate grids, however, require the solution of a system of partial differential equation. To introduce these elliptic system, the following analogy is proposed; given a physical domain bounded by four sides.

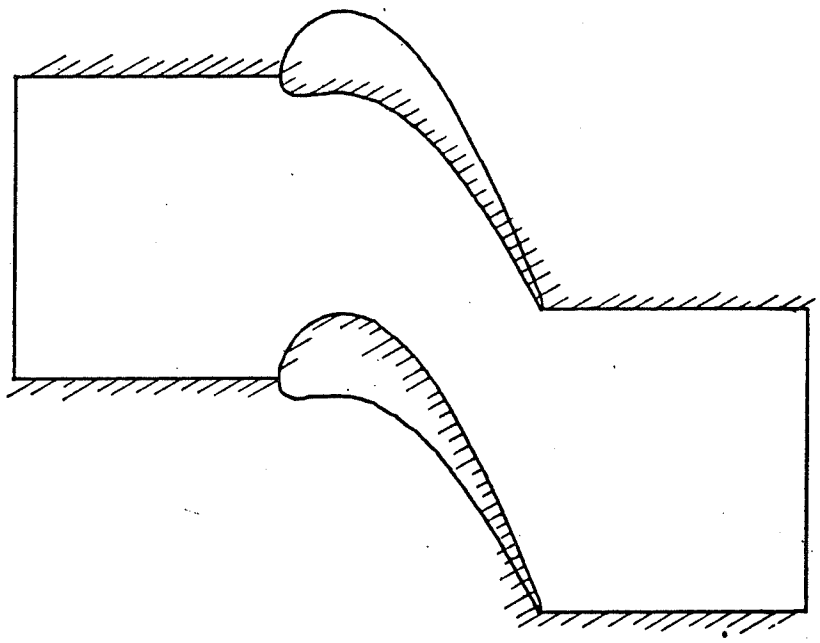


Fig. 23 Boundary conditions for the first family of grid lines

One poses that a family of body-fitted coordinate system could be generated by solving the temperature field with two opposite sides thermally insulated and the other pair at a given temperature differential. The solution of the resulting temperature field yields a family of constant temperature lines which are body-fitted coordinates. Similarly by interchanging the pair of sides which are insulated and those at a temperature differential, one can generate the second family of coordinate lines.

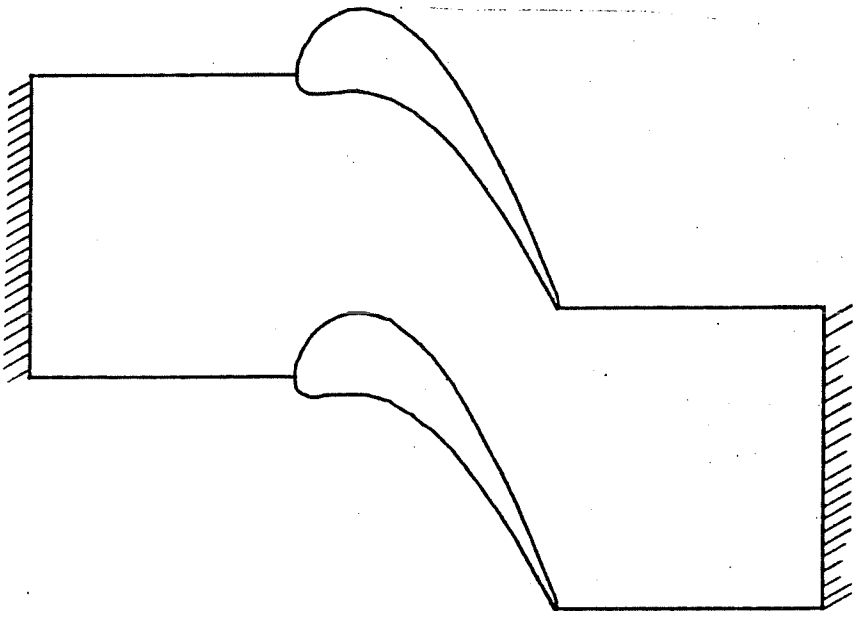


Fig. 24 Boundary conditions for the second family of grid lines

Let us denote the first family of coordinate curves by τ and the second by η. By the above analogy, the τ coordinates are lines of constant temperature in the problem corresponding to the first set of boundary condition Fig. 23. These can be obtained by solving the following problem characterised by the heat conduction of Laplace's Equation.

$$\nabla^2 \tau = 0$$

Similarly the set of η coordinate is obtained by solving the above problem with the second set of boundary conditions Fig. 24.

$$\nabla^2 \eta = 0$$

Thus we have now cast or formulated the generation of body-conforming coordinates as the solution of a system of two Laplace's equations. Being elliptic these will yield solutions (i.e. coordinate curves) that verify the requirements of

i) unicity (i.e. no overlapping)

ii) smoothness

More formally, the curvilinear coordinates ξ^i

with $\xi^1 = \tau$ and $\xi^2 = \eta$

are generated by the system

$$\nabla^2 \xi^i = 0 \qquad i = 1,2 \qquad (1)$$

subject to the boundary conditions:

1) for the first pair of boundaries

$$\xi^1 = \xi_1^1(x,y)$$

$$\xi^2 = \xi_1^2 = \text{cst}$$

$$\xi^1 = \xi_m^1(x,y)$$

$$\xi^2 = \xi_m^2 = \text{cst}$$

2) for the second pair of boundaries

$$\xi^1 = \xi_1^1 = \text{cst}$$

$$\xi^2 = \xi_n^2(x,y)$$

$$\xi^1 = \xi^1_n = cst$$

$$\xi^2 = \xi^2_n(x,y)$$

The functions: $\xi^i(x,y)$ represent the shape of the boundaries, and the constants ξ^i_1, ξ^i_2, ... the values of the parameter along these boundaries.

The extension to three dimensions is immediate by replacing the coordinate curves by coordinate surfaces and adding a third family.

3.2 TRANSFORMATION RELATIONS

The solution of Eq. (1) can only be carried out numerically for general arbitrary shapes. It is not pratical to solve the problem as formulated since the Laplacian would have to be discretized in the Cartesian domain. This, ironically, would require a grid which is precisely the objective of the problem. Thus one last step in the formulation of the grid generation problem is to invert the dependent and independent variables. So that instead of seeking the coordinate curves as functions of the cartesian variables

$$\xi^i (x_j)$$

the solution is carried out in the curvilinear space and the physical variables are cast as functions of the curvilinear coordinates or transformed space

$$x_i (\xi^j)$$

This change of variable is carried out and the Laplace equations are rewritten in the transformed domain. Using the chain rule of calculus

$$\frac{\partial}{\partial x_i} = \frac{\partial}{\partial \xi^j} \frac{\partial \xi^j}{\partial x_i}$$

The system of Eq. (1) becomes, for two variables

$$\sum_{j=1}^{2} \sum_{i=1}^{2} g^{ij} \frac{\partial^2 x_\ell}{\partial \xi^i \partial \xi^j} = 0 \qquad \ell = 1,2 \qquad (2)$$

where $x_1 = x \quad x_2 = y$

$$g_{11} = \left(\frac{\partial x}{\partial \xi^1}\right)^2 + \left(\frac{\partial y}{\partial \xi^1}\right)^2$$

$$g_{22} = \left(\frac{\partial x}{\partial \xi^2}\right)^2 + \left(\frac{\partial y}{\partial \xi^2}\right)^2 \tag{3}$$

$$g_{12} = \frac{\partial x}{\partial \xi^1}\frac{\partial x}{\partial \xi^2} + \frac{\partial y}{\partial \xi^1}\frac{\partial y}{\partial \xi^2}$$

and their inverses are obtained from

$$g^{11} = g_{22}/g$$

$$g^{22} = g_{11}/g \tag{4}$$

$$g^{12} = -\, g_{12}/g$$

and

$$g = \det\,(g_{ij})$$

Similarly, for three dimensional space, the system of Eq. (1) becomes

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\frac{\partial^2 x_\ell}{\partial \xi^i \partial \xi^j} = 0 \qquad \ell = 1,3 \tag{5}$$

where $x_1 = x$, $x_2 = y$ and $x_3 = z$

$$g_{11} = \left(\frac{\partial x}{\partial \xi^1}\right)^2 + \left(\frac{\partial y}{\partial \xi^1}\right)^2 + \left(\frac{\partial z}{\partial \xi^1}\right)^2$$

$$g_{22} = \left(\frac{\partial x}{\partial \xi^2}\right)^2 + \left(\frac{\partial y}{\partial \xi^2}\right)^2 + \left(\frac{\partial z}{\partial \xi^2}\right)^2 \tag{6}$$

$$g_{33} = \left(\frac{\partial x}{\partial \xi^3}\right)^2 + \left(\frac{\partial y}{\partial \xi^3}\right)^2 + \left(\frac{\partial z}{\partial \xi^3}\right)^2$$

and
$$g_{12} = \frac{\partial x}{\partial \xi^1}\frac{\partial x}{\partial \xi^2} + \frac{\partial y}{\partial \xi^1}\frac{\partial y}{\partial \xi^2} + \frac{\partial z}{\partial \xi^1}\frac{\partial z}{\partial \xi^2}$$

$$g_{13} = \frac{\partial x}{\partial \xi^1}\frac{\partial x}{\partial \xi^3} + \frac{\partial y}{\partial \xi^1}\frac{\partial y}{\partial \xi^3} + \frac{\partial z}{\partial \xi^1}\frac{\partial z}{\partial \xi^3} \qquad (7)$$

$$g_{23} = \frac{\partial x}{\partial \xi^2}\frac{\partial x}{\partial \xi^3} + \frac{\partial y}{\partial \xi^2}\frac{\partial y}{\partial \xi^3} + \frac{\partial z}{\partial \xi^2}\frac{\partial z}{\partial \xi^3}$$

The inverse of the g^{ij} are obtained from

$$G_1 = g_{22}\, g_{33} - (g_{23})^2$$

$$G_2 = g_{11}\, g_{33} - (g_{13})^2$$

$$G_3 = g_{11}\, g_{22} - (g_{12})^2 \qquad (8)$$

$$G_4 = g_{13}\, g_{23} - g_{12}\, g_{33}$$

$$G_5 = g_{12}\, g_{23} - g_{13}\, g_{22}$$

$$G_6 = g_{12}\, g_{13} - g_{11}\, g_{23}$$

giving

$$g^{11} = G_1/g$$

$$g^{22} = G_2/g$$

$$g^{33} = G_3/g \qquad (9)$$

$$g^{12} = G_4/g$$

$$g^{13} = G_5/g$$

$$g^{23} = G_6/g$$

where $g = \det(g_{ij})$.

3.3 APPLICATIONS TO TURBOMACHINERY

The grid equations developed in the previous section are completely general and applicable to any two or three dimensional problem. These can be rewritten in a more explicit and more widely accepted notation of the field of turbomachinery. The variables ξ^i in the transformed space become η, τ and ξ for $i = 1, 2$ and 3 respectively. In the physical space the variables x, y and z are replaced by Z, ϕ and R.

In two dimensions, the transformation between the physical space (ϕ, Z) of a blade-to-blade region into the computational or transformed space (τ, η) is accomplished by the elliptic system of Eq.(2) rewritten with the appropriate notation. This becomes

$$\alpha\phi_{\eta\eta} + \gamma\phi_{\tau\tau} - 2\beta\phi_{\eta\tau} = 0$$

$$\alpha Z_{\eta\eta} + \gamma Z_{\tau\tau} - 2\beta Z_{\eta\tau} = 0 \qquad (10)$$

where

$$\alpha = \phi_{\tau}^2 + Z_{\tau}^2$$

$$\gamma = \phi_{\tau}^2 + Z_{\eta}^2$$

$$\beta = \phi_{\eta}\phi_{\tau} + Z_{\eta}Z_{\tau}$$

$$J = \phi_{\eta}Z_{\tau} - \phi_{\tau}Z_{\eta}$$

This is completed with a set of conditions imposed on the domain boundary Γ. This is made up of four sides denoted by Γ_1, Γ_2, Γ_3 and Γ_4 and coincide with $\eta = \eta_1$ for Γ_1, $\eta = \eta_2$ for Γ_2, $\tau = \tau_1$ for Γ_3 and $\tau = \tau_2$ for Γ_4.

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}f_1(\eta_1,\tau) \\ f_2(\eta_1,\tau)\end{matrix} \qquad \text{along } \Gamma_1$$

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}g_1(\eta_2,\tau) \\ g_2(\eta_2,\tau)\end{matrix} \qquad \text{along } \Gamma_2$$

$$\begin{matrix} \phi \\ Z \end{matrix} = \begin{matrix} h_1(\eta,\tau_1) \\ h_2(\eta,\tau_1) \end{matrix} \quad \text{along } \Gamma_3 \qquad (11)$$

$$\begin{matrix} \phi \\ Z \end{matrix} = \begin{matrix} q_1(\eta,\tau_2) \\ q_2(\eta,\tau_2) \end{matrix} \quad \text{along } \Gamma_4$$

Similarly, for three dimensional applications, the system of Eq. (5) can be rewritten explicitly in the cylindrical coordinates notation, as

$$a_1 r_{\xi\xi} + a_2 r_{\eta\eta} + a_3 r_{\tau\tau} + 2a_4 r_{\xi\eta} + 2a_5 r_{\eta\tau} + 2a_6 r_{\tau\xi} - \frac{1}{r} = 0$$

$$a_1 \phi_{\xi\xi} + a_2 \phi_{\eta\eta} + a_3 \phi_{\tau\tau} + 2a_4 \phi_{\xi\eta} + 2a_5 \phi_{\eta\tau} + 2a_6 \phi_{\tau\xi} = 0 \qquad (12)$$

$$a_1 Z_{\xi\xi} + a_2 Z_{\eta\eta} + a_3 Z_{\tau\tau} + 2a_4 Z_{\xi\eta} + 2a_5 Z_{\eta\tau} + 2a_6 Z_{\tau\xi} = 0$$

where

$$a_1 = \xi_r^2 + \frac{1}{r^2}\xi_\phi^2 + \xi_z^2$$

$$a_2 = \eta_r^2 + \frac{1}{r^2}\eta_\phi^2 + \eta_z^2$$

$$a_3 = \tau_r^2 + \frac{1}{r^2}\tau_\phi^2 + \tau_z^2$$

$$a_4 = \xi_r\eta_r + \frac{1}{r^2}\xi_\phi\eta_\phi + \xi_z\eta_z \qquad (13)$$

$$a_5 = \eta_r\tau_r + \frac{1}{r^2}\eta_\phi\tau_\phi + \eta_z\tau_z$$

$$a_6 = \tau_r\xi_r + \frac{1}{r^2}\tau_\phi\xi_\phi + \tau_z\xi_z$$

4. NUMERICAL SOLUTIONS

4.1 DISCRETIZATION

The solution to the system of Eqs (2) or (5) can be carried out for general boundary conditions only by numerical methods. The equations are coupled and highly non-linear and there exist a number of very efficient and reliable numerical schemes for such systems. These can be quite complex but for the purposes of this presentation only the simplest methods need to be discussed to illustrate the basic principles.

The overall approach consist of two steps i) the differential equations are discretized and ii) the resulting set of algebraic equations are solved by numerical techniques. These are essentially iterative methods based on the relaxation procedures. The first step is to discretize the transformed domain into a mesh. Since the actual values ξ^i are irrelevant to the resulting grid, one can without loss of generality, assume any range of values for each coordinate family. This could be a variation from 0 to 1 or more appropriately for numerical applications, from 1 to the number of nodes. Typically, these are different for each coordinate family and we will denote these by m, n and l for the ξ, η and τ directions. For a function f, we will denote by the indices i and j (in 2-D) and i, j and k (in 3-D) the values at the grid nodes in the computational or parameter space. Thus the discretized values of the dependent variables τ or η will be represented by

$$\tau_i = (i - 1)\Delta\tau \qquad 1 \leq i \leq m$$

or (14)

$$\eta_j = (j - 1)\Delta\eta \qquad 1 \leq j \leq n$$

where the increments $\Delta\tau$, $\Delta\eta$, etc are the range of the corresponding variables discretized into equal intervals, i.e.

$$\Delta\tau = (\tau_M - \tau_1)/(m - 1)$$

The value of a function at a node (i,j) is

$$f_{i,j} = f(\tau_i, \eta_j) \tag{15}$$

The extension of this notation to three dimensions is immediate.

The next step is to discretize the system of elliptic differential equations. This is carried out by replacing the

derivatives appearing in these equations by the appropriate finite difference approximations. Since both first and second derivatives appear, it is natural to use centered differences which yield second order accuray for both.

$$\frac{\partial f}{\partial \tau} \simeq \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta\tau}$$

$$\frac{\partial f}{\partial \eta} \simeq \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta\eta}$$

$$\frac{\partial^2 f}{\partial \tau^2} \simeq \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta\tau^2} \qquad (16)$$

$$\frac{\partial^2 f}{\partial \eta^2} \simeq \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta\eta^2}$$

$$\frac{\partial^2 f}{\partial \tau \partial \eta} \simeq \frac{f_{i+1,j+1} - f_{i+1,j-1} - f_{i-i,j+1} + f_{i-1,j-1}}{4\Delta\tau\Delta\eta}$$

Substituting the expressions of Eq. 16 into the elliptic system of Eq. (2) and analogous expressions for three dimensions into Eq. (5) yields an equivalent algebraic equation for each node of the computational space. When this is done for all nodes of the domain an algebraic system is obtained. Like its differential counterpart this is highly nonlinear because the coefficients g^{ij} depend on the solution.

There exist many techniques to solve such systems numerically. Essentially these are iterative in the sense that the solution is approached asymptotically by improving a given initial solution in a step by step procedure. Without going into a formal and lengthy comparison of the numerous methods available, the successive overrelaxation scheme is proposed. This is based on an informal evaluation as well as what appears to be a consensus of the users in this field. It is felt that such schemes are accurate, robust and easy to program and that they give very efficient overall codes.

4.2 NUMERICAL SCHEMES

The discussion of the numerical scheme for the solution of the grid equations will be carried out with the explicit form of the these in the (ϕ,Z) coordinate referential. This is to keep the presentation at a specific level knowing that generalisations and extensions are simple when the basic approach is well understood. So, using the expressions of Eqs. (16) and replacing the derivatives in the elliptic system of Eqs. (10), one obtains equivalent algebraic equations for every node (i,j) of the computational space. One of these is for the tangential coordinate $\phi_{i,j}$:

$$\alpha'[\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}] + \gamma'[\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}]$$

$$- 2\beta'[\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1}] = 0 \qquad (17)$$

and the second is for the axial coordinate $Z_{i,j}$:

$$\alpha'[Z_{i+1,j} - 2Z_{i,j} + Z_{i-1,j}] + \gamma'[Z_{i,j+1} - 2Z_{i,j} + Z_{i,j-1}]$$

$$- 2\beta'[Z_{i+1,j+1} - Z_{i-1,j+1} - Z_{i+1,j-1} + Z_{i-1,j-1}] = 0 \qquad (18)$$

where

$$\alpha' = ((\phi_{i,j+1} - \phi_{i,j-1})^2 + (Z_{i,j+1} - Z_{i,j-1})^2)/((2\Delta\tau)^2 * (\Delta\eta)^2)$$

$$\gamma' = ((\phi_{i+1,j} - \phi_{i-1,j})^2 + (Z_{i+1,j} - Z_{i-1,j})^2/((2\Delta\eta)^2 * (\Delta\tau)^2)$$

$$\beta' = ((\phi_{i+1,j} - \phi_{i-1,j})(\phi_{i,j+1} - \phi_{i,j-1}) + (Z_{i+1,j} - Z_{i-1,j})$$

$$(Z_{i,j+1} - Z_{i,j-1}))/((2\Delta\eta)^2 * (2\Delta\tau)^2).$$

When applied to every node, this yields a system of non-linear coupled equations which must be solved iteratively. At each step of this procedure the coefficients of Eqs. (17) and (18) are frozen and updated after new values of the unknowns are obtained. Essentially two approaches have been used for this class of problems

i) Successive-overrelaxation

ii) The alternating-direction-implicit schemes

In the latter approach the elliptic problem is transformed into a parabolic problem by the addition of a transient term. This can be thought of as an artificial time and each time step may be associated to an iteration of the SOR method. So both methods are similar and the only criteria should be the rate of convergence and ease of programming. The choice was made on an intuitive basis and experience gathered by the present authors on prior applications of both of these methods but more importantly it is the method that lends itself best to improvements. It is felt that a scheme based on relaxation will yield a more efficient overall scheme. These can be classified as:

i) Point SOR

ii) Line SOR

Point SOR

The simplest scheme is the point SOR. The defined correction approach was used where provisional values of the variables are computed by sweeping the computational domain in a lexicographic order. In the calculation of these provisional values one uses corrected values and old values as these appear in the computational molecule illustrated in Fig. 25.

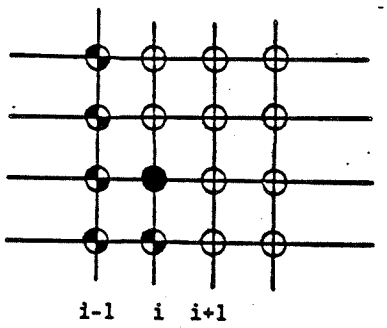


Fig. 25 Point SOR

The relations are derived by expressing Eqs. (17)-(18) in the variables ϕ and Z at (i,j) in discrete form.

$$\alpha'[\phi_{i+1,j} - 2\bar{\phi}_{i,j} + \phi^{+}_{i-1,j}] + \gamma'[\phi_{i,j+1} - 2\bar{\phi}_{i,j} + \phi^{+}_{i,j-1}] - 2\beta'[\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi^{+}_{i+1,j-1} + \phi^{+}_{i-1,j-1}] = 0 \quad (19)$$

and

$$\alpha'[Z_{i+1,j} - 2\bar{Z}_{i,j} + Z^{+}_{i-1,j}] + \gamma'[Z_{i,j+1} - 2\bar{Z}_{i,j} + Z^{+}_{i,j-1}] \quad (20)$$

$$- 2\beta'[Z_{i+1,j+1} - Z_{i-1,j+1} - Z^{+}_{i+1,j-1} + Z^{+}_{i-1,j-1}] = 0$$

Where ϕ and Z are old values, ϕ^{+} and Z^{+} are corrected values, and $\bar{\phi}$ and $\bar{Z}$ are provisional values. An old value is corrected by the provisional value and a relaxation factor, ω, as follows

$$\begin{aligned} \phi^{+} &= \phi + \omega(\bar{\phi}-\phi) \\ Z^{+} &= Z + \omega(\bar{Z}-Z) \end{aligned} \tag{21}$$

Thus one obtains for the provisional values

$$\begin{aligned} \bar{\phi}_{ij} &= \phi_{ij} + CF_{ij}/\omega \\ \bar{Z}_{ij} &= Z_{ij} + CZ_{ij}/\omega \end{aligned} \tag{22}$$

where the corrections are defined as

$$\begin{aligned} CF_{ij} &= \phi^{+}_{ij} - \phi_{ij} \\ CZ_{ij} &= Z^{+}_{ij} - Z_{ij} \end{aligned} \tag{23}$$

Substituting Eq. 22 into Eqs. (19 and (20) one obtains

$$\frac{2(\alpha'+\gamma')}{\omega} CF_{i,j} = RF_{i,j} + \alpha' CF_{i-1,j} - \beta'(CF_{i-1,j-1} - CF_{i+1,j-1}) + \gamma' CF_{i,j-1} \tag{24}$$

$$\frac{2(\alpha'+\gamma')}{\omega} CZ_{i,j} = RZ_{i,j} + \alpha' CZ_{i-1,j} - \beta'(CZ_{i-1,j-1} - CZ_{i+1,j-1}) + \gamma' CZ_{i,j-1} \tag{25}$$

The residuals are

$$\begin{aligned} RF_{i,j} = {} & \alpha'(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}) \\ & + \gamma'(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}) \\ & - 2\beta'(\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1}) \end{aligned} \tag{26}$$

an identical expression for $RZ_{i,j}$ is readily obtained by replacing ϕ by Z in Eq. 26.

With an initial solution, the values of ϕ and Z at each node (i,j) are successively corrected using Eqs. (24) and (25), sweeping the domain as indicated in Fig. 25.

Notes: In the actual program the values of $\Delta\eta$ and $\Delta\tau$ in the expressions in Eqs. (17)-(18) and (24)-(25) are set identically to unity

The coefficients α', β' and γ' involve the values of the variables ϕ and Z. These can be frozen during one domain sweep or updated as they become available in the relaxation process.

Block SOR

In block relaxation, all the nodes in a given linear column are solved implicitly at once. This yields a tridiagonal system of equations which is easily solved. Solving n points implicitly requires about the same computation effort as solving n times one point explicitly. The advantage lies in the fact that boundary conditions which appear as the end points of an implicit block of equations are felt immediately throughout the line. In an explicit scheme it requires about as many sweeps as there are points in the line for the information at the boundary to propagate within the domain. This makes implicit relaxation much more efficient. The correction algorithm is now described for sweeps implicit in either of the two coordinate directions.

SOR Implicit by row

The configuration for a typical row relaxation implicit along a τ = const coordinate line is shown in Fig. 26.

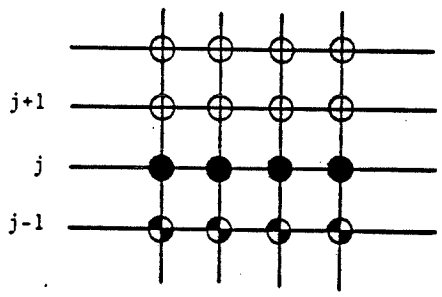


Fig. 26 Block relaxation

The difference equations, Eqs. (17)-(18) are written for the unknown points along the row taking into account, as for point relaxation, the status of the neighboring points, i.e. corrected or old.

$$\alpha'[\bar{\phi}_{i+1,j} - 2\bar{\phi}_{i,j} + \bar{\phi}_{i-1,j}] + \gamma'[\bar{\phi}_{i,j+1} - 2\bar{\phi}_{i,j} + \overset{+}{\phi}_{i,j-1}]$$

$$- 2\beta'[\bar{\phi}_{i+1,j+1} - \bar{\phi}_{i-1,j+1} - \overset{+}{\phi}_{i+1,j-1} + \overset{+}{\phi}_{i-1,j-1}] = 0 \qquad (27)$$

Defining a corrected value in terms of the old and provisional values, Eqs. (21) and (22), one obtains the correction equations for the tangential coordinate:

$$\frac{\alpha'}{\omega} CF_{i-1,j} - \frac{2(\alpha'+\gamma')}{\omega} CF_{i,j} + \frac{\alpha'}{\omega} CF_{i+1,j}$$

$$= -RF_{i,j} - \gamma' CF_{i,j-1} + 2\beta'[CF_{i+1,j-1} - CF_{i-1,j-1}] \qquad (28)$$

and for the axial coordinate:

$$\frac{\alpha'}{\omega} CZ_{i-1,j} - \frac{2(\alpha'+\gamma')}{\omega} CZ_{i,j} + \frac{\alpha'}{\omega} CZ_{i+1,j}$$

$$= -RZ_{i,j} - \gamma' CZ_{i,j-1} + 2\beta'(CZ_{i+1,j-1} - CZ_{i-1,j-1}) \qquad (29)$$

where the residuals $RF_{i,j}$ and $RZ_{i,j}$ are defined by Eq. 26. Equations (28) and (29) differ from their counterpart, Eqs. (24) and (25), in point relaxation in that they are implicit and each equation involves three unknowns. The following tridiagonal systems are obtained:

$$\begin{bmatrix} B_2 & C_2 & & & & \\ A_3 & B_3 & C_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & A_i & B_i & C_i & \\ & & & \ddots & \ddots & \ddots \\ & & & & A_{n-1} & B_{n-1} & C_{n-1} \end{bmatrix} \begin{bmatrix} CF_2 \\ CF_3 \\ \vdots \\ CF_i \\ \vdots \\ CF_{n-1} \end{bmatrix} = \begin{bmatrix} D_2 \\ D_3 \\ \vdots \\ D_i \\ \vdots \\ D_{n-1} \end{bmatrix} \qquad (30)$$

$$\begin{bmatrix} B_2 & C_2 & & & & \\ A_3 & B_3 & C_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & A_i & B_i & C_i & \\ & & & \ddots & \ddots & \ddots \\ & & & A_{n-1} & B_{n-1} & C_{n-1} \end{bmatrix} \begin{bmatrix} CZ_2 \\ CZ_3 \\ \vdots \\ CZ_i \\ \vdots \\ CZ_{n-1} \end{bmatrix} = \begin{bmatrix} DD \\ DD \\ \vdots \\ DD_i \\ \vdots \\ DD_{n-1} \end{bmatrix} \qquad (31)$$

and

$$A_i = \alpha'/\omega$$

$$B_i = -2(\alpha' - Y')/\omega \qquad (32)$$

$$C_i = \alpha'/\omega$$

$$D_i = -RF_{i,j} - Y'CF_{i,j-1} - 2\beta'[CF_{i+1,j-1} - CF_{i-1,j-1}]$$

$$DD_i = -RZ_{i,j} - Y'CZ_{i,j-1} - 2\beta'[CZ_{i+1,j-1} - CZ_{i-1,j-1}]$$

and the boundary conditions are:

$$CF_1 = CF_n = 0$$

$$CZ_1 = CZ_n = 0$$

Solution of the systems of Eqs. (30) and (31) yields corrections for an entire row which are used to modify the corresponding column of ϕ's and Z's.

SOR Implicit by column

Figure 27 illustrates the configuration for a relaxation sweep implicit along the η = constant coordinate direction.

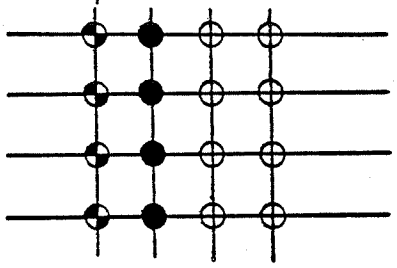


Fig. 27 Block relaxation

The difference equations are written for every node along a given column and this yields,

$$\alpha'(\phi_{i+1,j} - 2\bar{\phi}_{i,j} + \phi^{+}_{i-1,j}) + \gamma'(\bar{\phi}_{i,j+1} - 2\bar{\phi}_{i,j} + \bar{\phi}_{i,j-1}) - 2\beta'(\phi_{i+1,j+1} - \phi^{+}_{i-1,j+1} - \phi_{i+1,j-1} + \phi^{+}_{i-1,j-1}) = 0 \tag{33}$$

and

$$\alpha'[Z_{i+1,j} - 2\bar{Z}_{i,j} + Z^{+}_{i-1,j}] + \gamma'[\bar{Z}_{i,j+1} - 2\bar{Z}_{i,j} + \bar{Z}_{i,j-1}] - 2\beta'[Z_{i+1,j+1} - Z^{+}_{i-1,j+1} - Z_{i+1,j-1} + Z^{+}_{i-1,j-1}] = 0 \tag{34}$$

From which as previously one obtains the correction equations

$$\frac{\gamma'}{\omega} CF_{i,j-1} - \frac{(\alpha'+\gamma')}{\omega} CF_{i,j} + \frac{\gamma'}{\omega} CF_{i,j+1} = -RF_{i,j}$$

$$- \alpha' CF_{i-1,j} + 2\beta'(CF_{i-1,j+1} - CF_{i-1,j-1}) \tag{35}$$

and

$$\frac{\gamma'}{\omega} CZ_{i,j-1} - \frac{2(\alpha'+\gamma')}{\omega} CZ_{i,j} + \frac{\gamma'}{\omega} CZ_{i,j+1} = -RZ_{i,j}$$

$$- \alpha' CZ_{i-1,j} + 2\beta'(CZ_{i-1,j+1} - CZ_{i-1,j-1}) \tag{36}$$

where the residuals are defined in Eq. 26.

Tridiagonal systems equivalent to those of Eqs. (30) and (31) for column relaxation are obtained and one proceeds in similar fashion for both types of procedures.

Direction of relaxation sweep

For a given relaxation scheme the marching direction affects the rate of convergence. If the coefficients of the derivatives in the coordinates η and τ are of the same order of magnitude, then the equation will exhibit no preferred marching direction and it is best to use symmetric relaxation. In general, the marching direction should be chosen to coincide with the direction of propagation of physical information. In this respect the boundary conditions and their types should be taken into account.

If such directions are not readily apparent from either the physics of the problem or when the weighting of the nodes (i.e. the coefficients) changes in the domain then sweeps should be carried out in alternating directions. This will speed up the convergence when the marching direction coincides locally with that of information propagation, and will have a neutral effect otherwise.

This is a two-step procedure. First with a column (y) implicit scheme the entire field is relaxed marching in the positive x-direction. This is followed by a row (x) implicit relaxation marching in the positive y-direction.

4.3 GRID CONTROL

The shape and the distribution of the curvilinear coordinates which result from the solution of the Laplace system, Eq. (1), depend essentially on the shape of the boundaries. Before proceeding with the methods to control these, a discussion of the qualitative behaviour of the Laplace system is presented.

At a boundary or in its vicinity, the coincident coordinate curve or curves, say the η - family for example may be either concave or convex. In the first instance, then we have

$$\eta_{xx} \geq 0$$

so that η_{yy} must be negative if the Laplacian is to be verified. Since η_y is the spacing of η - lines, then this condition is interpreted as the concentration of this family of coordinates.

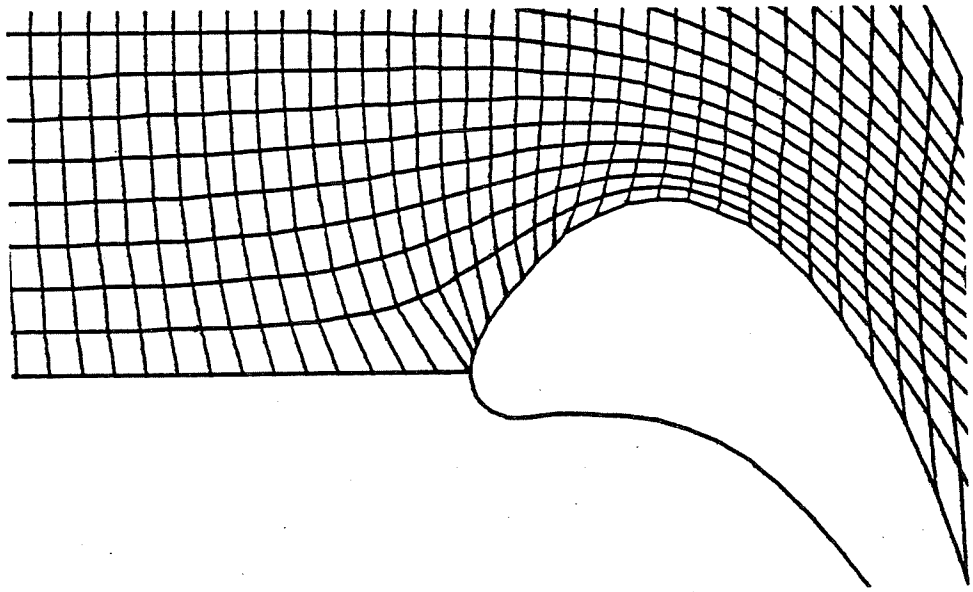


Fig. 28 Concentration of grid lines in the vicinity of concave boundary

So in this example of a concave boundary, the spacing of η - lines decreases as one moves away from the boundary. The same reasoning yields that for a convex boundary, the spacing increases, that is the grid lines tend to concentrate towards the boundary.

Another illustration of effect of the boundaries on the grid lines is the following.

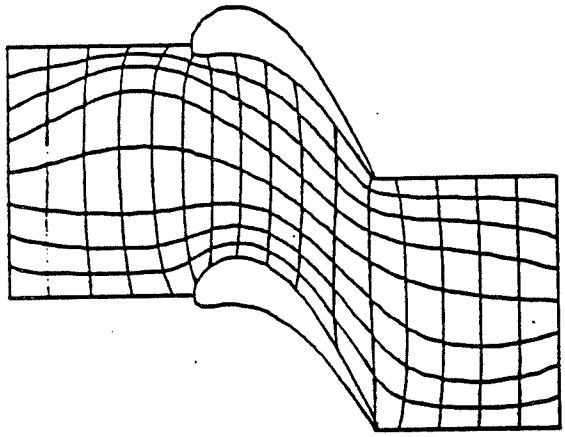


Fig. 29 Concentration of grid lines towards domain boundaries

The nodes on two opposite boundaries are non-uniformally distributed. The need for doing this is obvious for problems where gradients of properties are present. Unfortunately, the resulting grid is not what is expected.

This behaviour can also be deduced from physical analogies such as those presented in Section 3.1. And it is seen that as such, the Laplace system possesses generally good characteristics but some drawbacks such as described above. Namely, that in areas of concave or convex boundaries, the distribution of grid lines usually goes the opposite way that one would like and that the grid spacing cannot be controlled simply by moving boundary nodes.

So there is a need to devise in the present grid generation method, a mechanism to control the distribution of grid lines. This is necessary to correct the above situations, but more generally to have a simple way to influence the grid. This can be achieved by adding a forcing term to the Laplace equation, thus generalizing the elliptic system to Poisson equations.

$$\nabla^2 \xi^i = P_i \qquad i = 1, 2, 3 \qquad (37)$$

Using such forcing terms, P_i, one can achieve any desired result. However it is not possible a priori to predict the nature and the level of the effect of a given forcing function. To acquire such expertise requires a certain experience which comes with trial and error. One can however arrive at certain guide line to enable one to predict trends qualitatively.

Line concentration

There are many ways that one may want to control a grid. These naturally depend on the application. The one that comes immediately to mind is the concentration of grid lines towards a boundary. For purposes of presentation and to introduce the matter in a most natural way, this will be done in a rather simple minded approach first. It will be formalized subsequently once the concept has been understood.

It is a two step procedure whereby the transformed domain ξ^i is mapped into an intermediate domain

$$\bar{\xi}^i = f(\xi^i)$$

In this mapping each variable is mapped separately according to the desired concentration . As an example in two dimensions, the physical domain is first transformed by Laplace' body-fitted system of equation.

$$(Z, \phi) \longrightarrow (\Psi, \xi)$$

Then, this intermediate domain is transformed into the computational domain by means of the following relations

$$\Psi = \Psi(\eta)$$

and (38)

$$\xi = \xi(\tau)$$

The sequence

$$(Z, \phi) \longrightarrow (\Psi, \xi) \longrightarrow (\eta, \tau)$$

is illustrated in Fig. 30.

This has the advantage of clearly separating the effects of concentration on each coordinate direction.

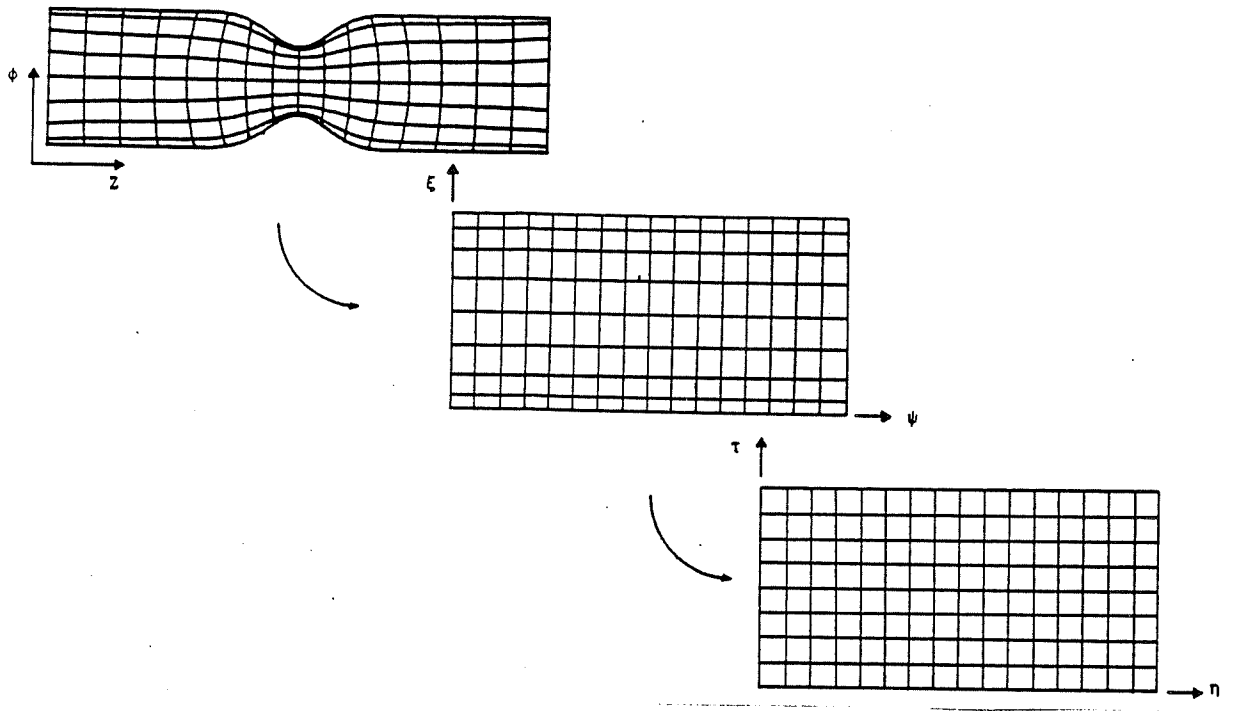


Fig. 30 Successive coordinate transformation

Chaining these transformation yields for the system of Eq. 1 the following Poisson equation for the η-coordinate

$$\nabla^2 \eta = \frac{\Psi_{\eta\eta}}{J^2_{\psi_\eta}} \alpha \tag{39}$$

and for the τ-coordinate

$$\nabla^2 \tau = \frac{\xi_{\tau\tau}}{J^2_{\xi_\tau}} \gamma \tag{40}$$

where Ψ, ξ and their derivatives represent the respective concentration of the coordinate curves. When comparing Eqs. (39) and (40) with the proposed Poisson system, Eq. (37) setting $\xi^1 = \eta$ and $\xi^2 = \tau$, the form of the forcing terms P_1 and P_2 become the right hand sides of Eqs (39) and (40).

So that it is possible to interpret this procedure as equivalent to adding a forcing function to the original Laplace's equation. This can be thought as a force applied to each coordinate curve and the result is a displacement in the appropriate direction. Having established that such compound transformations can be reduced to the addition of a forcing term, one then has an infinity of possibilities corresponding to different choices of the function $\Psi(\eta)$ and the respective functions for the other directions.

5. APPLICATION TO A CASCADE ROW

The approach presented in the previous chapter for a general grid problem will now be applied to a two-dimensional blade-to-blade grid generation.

The first step is to invert the system of Eqs. (39) and (40) to bring the Poisson system to the form of Eqs. (2) or Eqs (10). The equations will now be written explicitly using

$$\xi^1 = \eta$$

$$\xi^2 = \tau$$

and the forcing terms

$$P_1 = Q$$

$$P_2 = R$$

The result is

$$\alpha\phi_{\eta\eta} + \gamma\phi_{\tau\tau} - 2\beta\phi_{\eta\tau} + \left(\frac{J^2 Q}{\psi_\eta} - \alpha\frac{\psi_{\eta\eta}}{\psi_\eta}\right)\phi_\eta + \left(\frac{J^2 R}{\xi_\tau} - \gamma\frac{\xi_{\tau\tau}}{\xi_\tau}\right)\phi_\tau = 0 \tag{41}$$

$$\alpha Z_{\eta\eta} + \gamma Z_{\tau\tau} - 2\beta Z_{\eta\tau} + \left(\frac{J^2 Q}{\psi_\eta} - \alpha\frac{\psi_{\eta\eta}}{\psi_\eta}\right) Z_\eta + \left(\frac{J^2 R}{\xi_\tau} - \gamma\frac{\xi_{\tau\tau}}{\xi_\tau}\right) Z_\tau = 0 \tag{42}$$

The boundary conditions are:

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}f_1(\eta_1, \xi(\tau)) \\ f_2(\eta_1, \xi(\tau))\end{matrix} \quad \text{along } \Gamma_1$$

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}g_1(\eta_2, \xi(\tau)) \\ g_2(\eta_2, \xi(\tau))\end{matrix} \quad \text{along } \Gamma_2 \tag{43}$$

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}h_1(\psi(\eta), \tau_1) \\ h_2(\psi(\eta), \tau_1)\end{matrix} \quad \text{along } \Gamma_3$$

$$\begin{matrix}\phi \\ Z\end{matrix} = \begin{matrix}q_1(\psi(\eta),\tau_2) \\ q_2(\psi(\eta),\tau_2)\end{matrix} \qquad \text{along } \Gamma_4$$

$$\alpha = \phi_\tau^2 + Z_\tau^2 \qquad \gamma = \phi_\eta^2 + Z_\eta^2$$

$$\beta = \phi_\eta \phi_\tau + Z_\eta Z_\tau \qquad J = \phi_\eta Z_\tau - \phi_\tau Z_\eta$$

The functions f_1 ... etc represent the shape of the boundaries Γ_1 except that now the nodes are distributed along these according to the concentration relationships Ψ and ξ of Eqs. (38).

It is noted that Eqs. (41) and (42) differ from their Laplace counterpart, Eqs. (10) by the presence of two terms which represent the effect of coordinate line concentration. It is also noted that each of these terms is made up of two parts. The forcing term $\alpha\Psi_{\eta\eta}/\Psi_\eta$ due to line concentration, and an additional general forcing term J^2Q/Ψ_η. This latter is quite arbitrary and can be set to zero if no other manipulation of the grid is desired. It will be kept in the following developments for generality's sake.

The numerical solution is obtained by discretizing the computational domain into m and n equal intervals of length $\Delta\eta$ and $\Delta\tau$. As before, these are arbitrary and in the actual computation, they are set identically equal to unity. Replacing the first and second derivatives in Eqs. (41) and (42) by second order accurate differences of Eqs. (16) yields an equivalent algebraic system. This is for the tangential coordinate:

$$\begin{aligned}&\alpha'[\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}] + \gamma'[\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}] \\ &- 2\beta'[\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1}] \\ &+ Q'(\phi_{i+1,j} - \phi_{i-1,j}) + R'[\phi_{i,j+1} - \phi_{i,j-1}] = 0\end{aligned} \tag{44}$$

and for the axial coordinates:

$$\begin{aligned}&\alpha'[Z_{i+1,j} - 2Z_{i,j} + Z_{i-1,j}] + \gamma'[Z_{i,j+1} - 2Z_{i,j} + Z_{i,j-1}] \\ &- 2\beta'[Z_{i+1,j+1} - Z_{i-1,j+1} - Z_{i+1,j-1} + Z_{i-1,j-1}] \\ &+ Q'(Z_{i+1,j} - Z_{i-1,j}) + R'[Z_{i,j+1} - Z_{i,j-1}] = 0\end{aligned} \tag{45}$$

where

$$\alpha' = ((\phi_{i,j+1} - \phi_{i,j-1})^2 + (Z_{i,j+1} - Z_{i,j-1})^2/((2\Delta\tau)^2*(\Delta\eta)^2)$$

$$\gamma' = ((\phi_{i+1,j} - \phi_{i-1,j})^2 + (Z_{i+1,j} - Z_{i-1,j})^2/((2\Delta\eta)^2*(\Delta\tau)^2)$$

$$\beta' = ((\phi_{i+1,j} - \phi_{i-1,j})(\phi_{i,j+1} - \phi_{i,j-1}) + (Z_{i+1,j} - Z_{i-1,j})$$

$$(Z_{i,j+1} - Z_{i,j-1}))/((2\Delta\eta)^2 * (2\Delta\tau)^2). \qquad (46)$$

These equations are similar to the system of Eqs. (17) and (18) except for the presence of the forcing terms Q' and R'. They are also solved by iterative schemes such as point or block SOR. The details are identical to those described in the previous chapter so only the end results for column and row SOR will given.

$$Q' = \left[\frac{J^2 Q}{\psi_\eta} - (\Delta\eta)^2 \alpha' \frac{\psi_{\eta\eta}}{\psi_\eta} \right] / 2\Delta\eta$$

$$R' = \left[\frac{J^2 R}{\xi_\tau} - (\Delta\tau)^2 \gamma' \frac{\xi_{\tau\tau}}{\xi_\tau} \right] / 2\Delta\tau$$

SOR implicit by Column

The tridiagonal system (equivalent to the system of Eqs. (28) and (29)) is for the tangential coordinate:

$$\frac{(\alpha'-Q')}{\omega} CF_{i-1,j} - \frac{2(\alpha'+\gamma')}{\omega} CF_{i,j} + \frac{(\alpha'+Q')}{\omega} CF_{i+1,j} \qquad (47)$$

$$= -RF_{i,j} - \gamma' CF_{i,j-1} - 2\beta' [CF_{i+1,j-1} - CF_{i-1,j-1}] + R' CF_{i,j-1}$$

and for the axial coordinate

$$\frac{(\alpha'-Q')}{\omega} CZ_{i-1,j} - \frac{2(\alpha'+\gamma')}{\omega} CZ_{i,j} + \frac{(\alpha'+Q')}{\omega} CZ_{i+1,j} \qquad (48)$$

$$= -RZ_{i,j} - \gamma' CZ_{i,j-1} - 2\beta' (CZ_{i+1,j-1} - CZ_{i-1,j-1}) + R' CZ_{i,j-1}$$

where

$$RF_{i,j} = \alpha'(\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}) + \gamma'(\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}) - 2\beta'(\phi_{i+1,j+1} - \phi_{i-1,j+1} - \phi_{i+1,j-1} + \phi_{i-1,j-1}) + Q'(\phi_{i+1,j} - \phi_{i-1,j}) + R'(\phi_{i,j+1} - \phi_{i,j-1}) \tag{49}$$

$$RZ_{i,j} = \alpha'(Z_{i+1,j} - 2Z_{i,j} + Z_{i-1,j}) + \gamma'(Z_{i,j+1} - 2Z_{i,j} + Z_{i,j-1}) - 2\beta'(Z_{i+1,j+1} - Z_{i-1,j+1} - Z_{i+1,j-1} - Z_{i-1,j-1}) + Q'(Z_{i+1,j} - Z_{i-1,j}) + R'(Z_{i,j+1} - Z_{i,j-1})$$

These can then be cast in the usual tridiagonal form. For the circumferential coordinate:

$$\begin{bmatrix} B_2 & C_2 & & & \\ A_3 & B_3 & C_3 & & \\ & \ddots & \ddots & \ddots & \\ & & A_i & B_i & C_i \\ & & & \ddots & \ddots \\ & & & A_{n-1} & B_{n-1} & C_{n-1} \end{bmatrix} \begin{bmatrix} CF_2 \\ CF_3 \\ \vdots \\ CF_i \\ \vdots \\ CF_{n-1} \end{bmatrix} = \begin{bmatrix} D_2 \\ D_3 \\ \vdots \\ D_i \\ \vdots \\ D_{n-1} \end{bmatrix} \tag{50}$$

and for the axial coordinate:

$$\begin{bmatrix} B_2 & C_2 & & & \\ A_3 & B_3 & C_3 & & \\ & \ddots & \ddots & \ddots & \\ & & A_i & B_i & C_i \\ & & & \ddots & \ddots \\ & & & A_{n-1} & B_{n-1} & C_{n-1} \end{bmatrix} \begin{bmatrix} CZ_2 \\ CZ_3 \\ \vdots \\ CZ_i \\ \vdots \\ CZ_{n-1} \end{bmatrix} = \begin{bmatrix} DD \\ DD \\ \vdots \\ DD_i \\ \vdots \\ DD_{n-1} \end{bmatrix} \tag{51}$$

where

$$A_i = (\alpha'-Q')/\omega$$

$$B_i = -2(\alpha'- Y')/\omega$$

$$C_i = (\alpha'+Q')/\omega \qquad (52)$$

$$D_i = -RF_{i,j} - Y'CF_{i,j-1} - 2\beta'[CF_{i+1,j-1} - CF_{i-1,j-1}]$$

$$DD_i = -RZ_{i,j} - Y'CZ_{i,j-1} - 2\beta'[CZ_{i+1,j-1} - CZ_{i-1,j-1}]$$

The systems of Eqs. (50) and (51) are subject to the boundary conditions

$$CF_1 = CF_n = 0$$
$$CZ_1 = CZ_n = 0$$

Again it is noted that by setting the forcing terms Q* and R* to zero, the systems of Eqs. (30), (31) and (32) is recovered.

Some practical results will now be shown for a typical cascade. The grid control is illustrated in Fig. 31. The coordinate curves are concentrated towards the leading and trailing edges as well as towards the pressure and suction sides of the profile. The specific distribution function used is a cubic polynomial of the form

$$y = (1-C)\ x^3 + C\ x$$

where y is Ψ or ξ (depending on whether the concentration is in the axial or tangential direction and x is η or τ. This type of function is particularly useful and the grid control is realised through a single parameter C. Fig. 32 shows this for several values of this parameter.

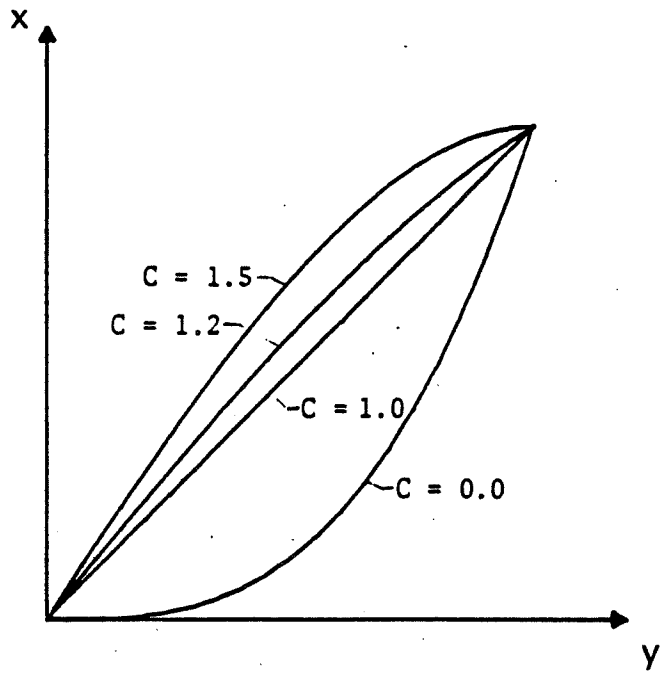


Fig. 32a Concentration function

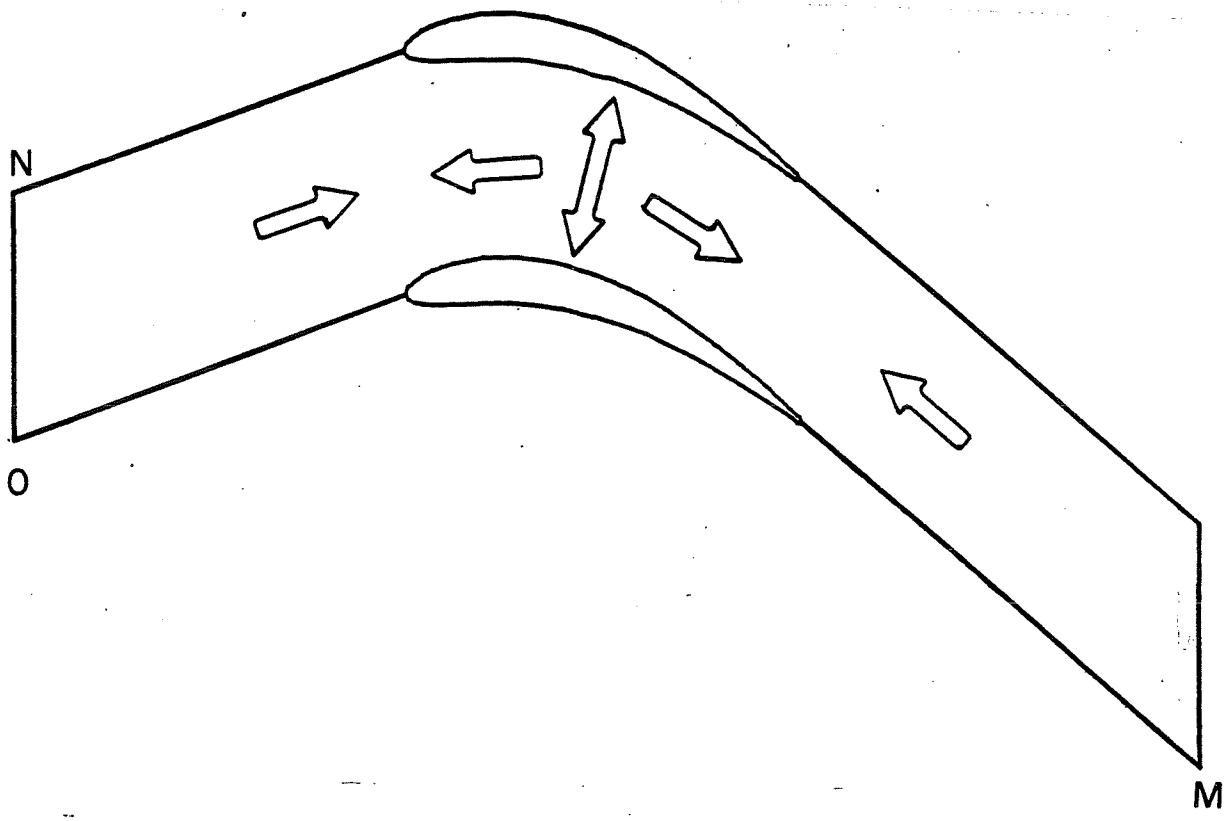


Fig. 32b Concentration pattern

The resulting grids for various values of C in both coordinate directions shown in Figs. 33-37.

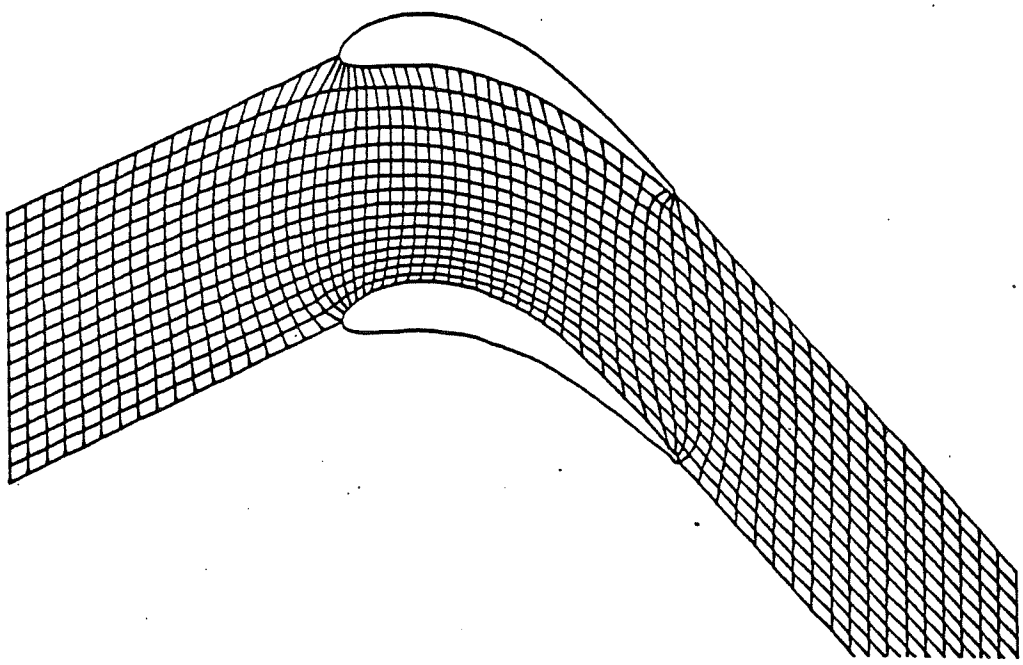


Fig. 33 65x17 grid with no concentration,C1=1, C2=1

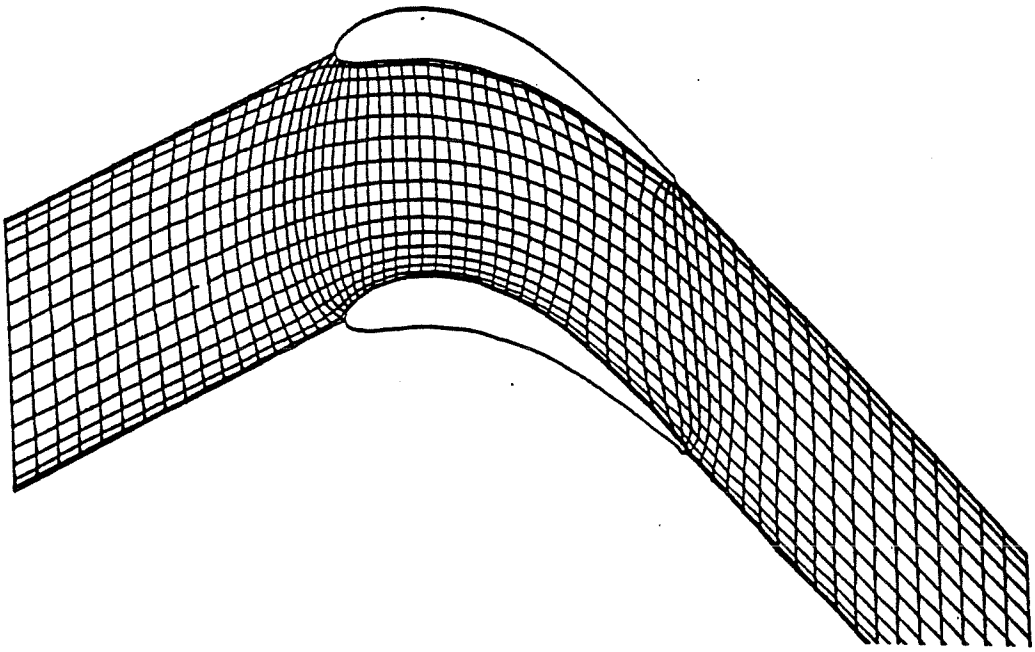


Fig. 34 65x17 grid with maximum concentration in the blade-to-blade direction and minimum concentration in the streamwise direction, C1=1.5, C2=1.2

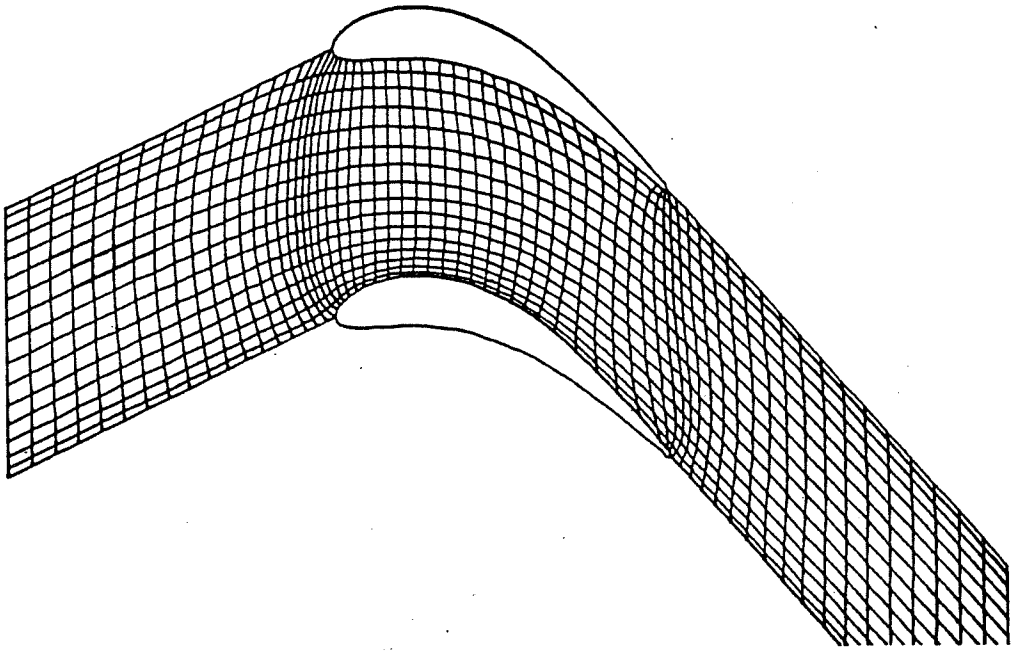


Fig. 35 65x17 grid with medium concentration in both directions, C1=1.5, C2=1.2

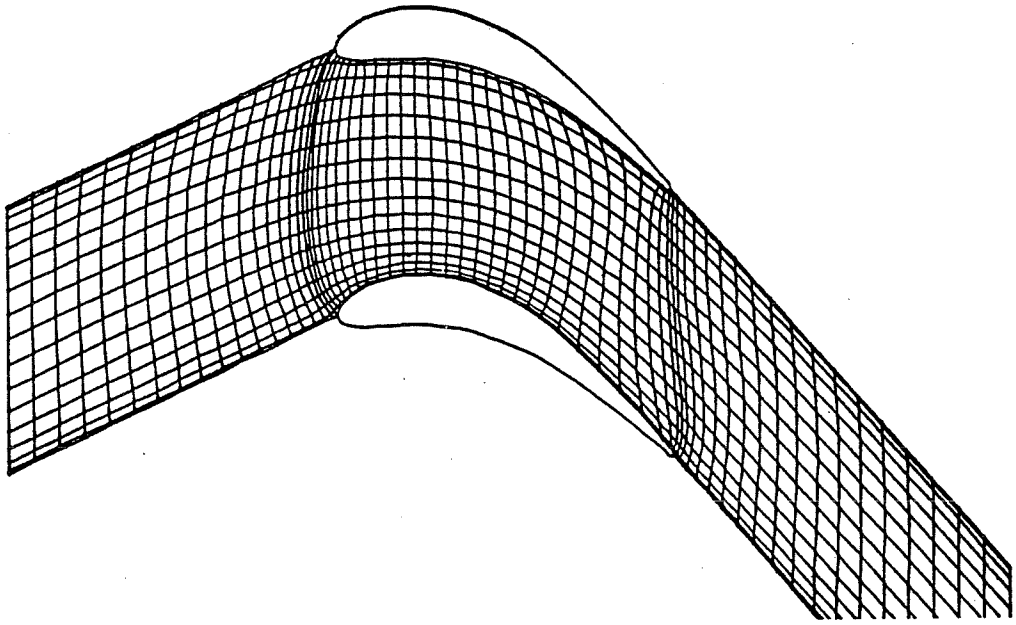


Fig. 36 65x17 grid with C1=1.5, C2=1.4

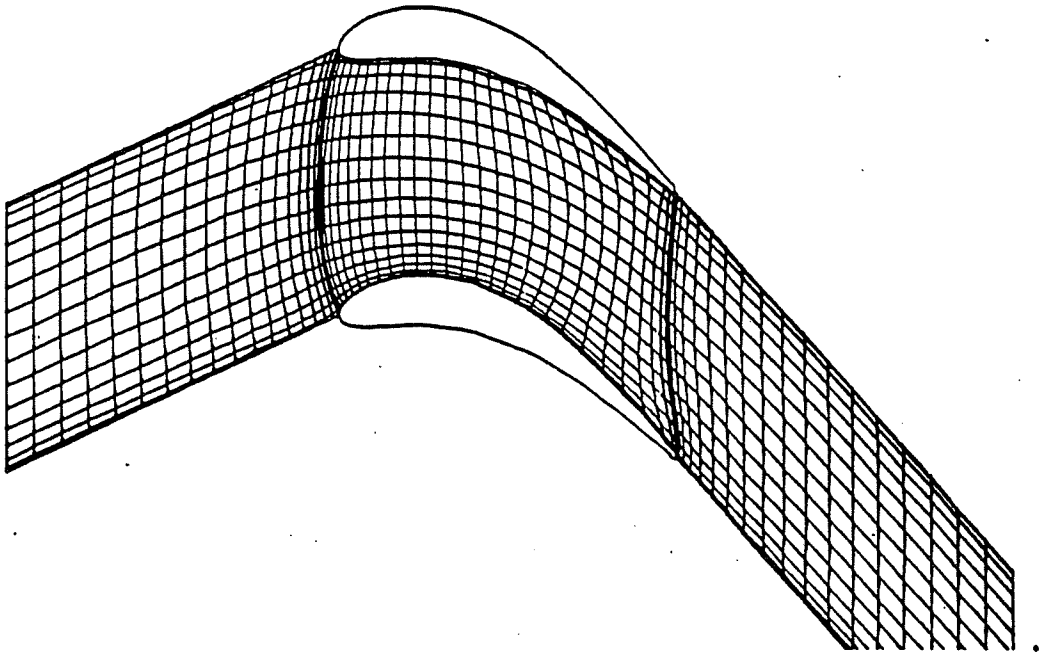


Fig. 37 65x17 grid with C1=1.5, C2=1.4

6. THREE DIMENSIONAL APPLICATIONS

The procedure for generating body-fitted grids for three-dimensional applications is similar to that described for cascades. This elliptic system for the parameter space is written explicitly in cylindrical coordinates since this is the most appropriate for turbomachinery applications. This gives for $\xi^1 = \xi$, $\xi^2 = \eta$ and $\xi^3 = \tau$, and $x^1 = r$, $x^2 = \phi$ and $x^3 = z$

$$\begin{aligned} \xi_{rr} + \frac{1}{r}\xi_r + \frac{1}{r^2}\xi_{\phi\phi} + \xi_{zz} &= P \\ \eta_{rr} + \frac{1}{r}\eta_r + \frac{1}{r^2}\eta_{\phi\phi} + \eta_{zz} &= Q \\ \tau_{rr} + \frac{1}{r}\tau_r + \frac{1}{r^2}\tau_{\phi\phi} + \tau_{zz} &= R \end{aligned} \tag{53}$$

Inverting this system to bring it to the form of Eq. (5) is laborious but yields

(54)

$$a_1 r_{\xi\xi} + a_2 r_{\eta\eta} + a_3 r_{\tau\tau} + 2a_4 r_{\xi\eta} + 2a_5 r_{\eta\tau} + 2a_6 r_{\tau\xi} - \frac{1}{r} + Pr_\xi + Qr_\eta + Rr_\tau = 0$$

$$a_1 \phi_{\xi\xi} + a_2 \phi_{\eta\eta} + a_3 \phi_{\tau\tau} + 2a_4 \phi_{\xi\eta} + 2a_5 \phi_{\eta\tau} + 2a_6 \phi_{\tau\eta} + P\phi_\xi + Q\phi_\eta + R\phi_\tau = 0$$

$$a_1 z_{\xi\xi} + a_2 z_{\eta\eta} + a_3 z_{\tau\tau} + 2a_4 z_{\xi\eta} + 2a_5 z_{\eta\tau} + 2a_6 z_{\tau\xi} + Pz_\xi + Qz_\eta + Rz_\tau = 0$$

where

$$a_1 = \xi_r^2 + \frac{1}{r^2}\xi_\phi^2 + \xi_z^2$$

$$a_2 = \eta_r^2 + \frac{1}{r^2}\eta_\phi^2 + \eta_z^2$$

$$a_3 = \tau_r^2 + \frac{1}{r^2}\tau_\phi^2 + \tau_z^2 \qquad (55)$$

$$a_4 = \xi_r\eta_r + \frac{1}{r^2}\xi_\phi\eta_\phi + \xi_z\eta_z$$

$$a_5 = \eta_r\tau_r + \frac{1}{r^2}\eta_\phi\tau_\phi + \eta_z\tau_z$$

$$a_6 = \tau_r\xi_r + \frac{1}{r^2}\tau_\phi\xi_\phi + \tau_z\xi_z$$

The domain is bounded by the surfaces Γ_1, Γ_2, ... Γ_6 which correspond to coordinate surfaces for which ξ, η and τ are constant as illustrated in Fig (38).

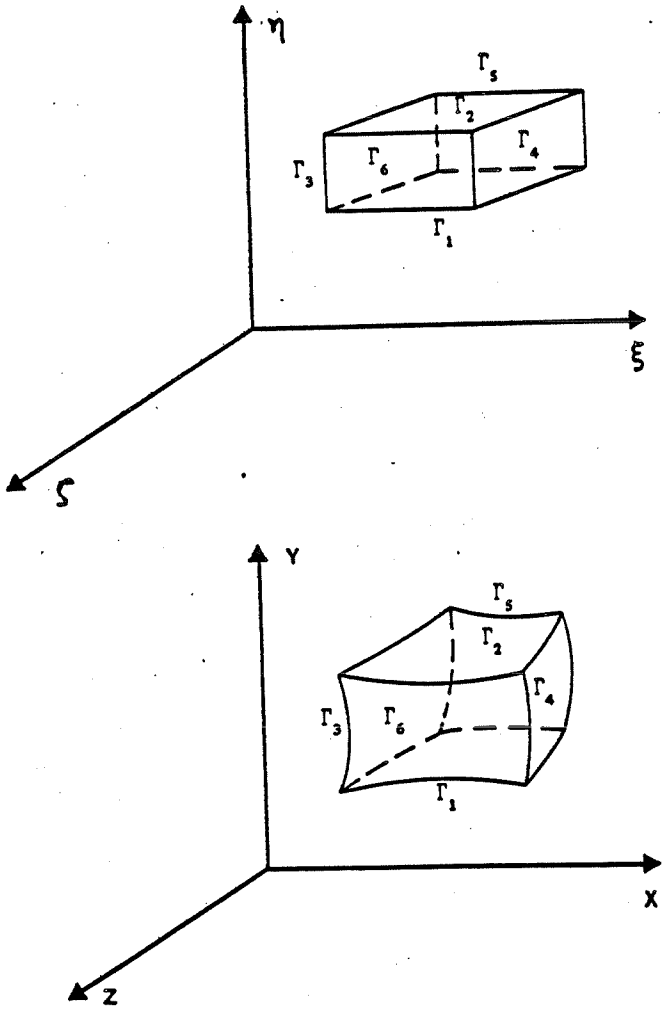


Fig. 38 Mapping of physical domain into logical space in 3-D application

The problem then reduces to solving Eqs. (53) subject to Dirichlet boundary conditions which specify the shape of these boundaries in the following manner.

$$\begin{bmatrix} r \\ \phi \\ z \end{bmatrix} = \begin{bmatrix} f_1(\xi_1,\eta,\tau) \\ f_2(\xi_1,\eta,\tau) \\ f_3(\xi_1,\eta,\tau) \end{bmatrix} \quad \text{along } \Gamma_1$$

$$\begin{bmatrix} r \\ \phi \\ z \end{bmatrix} = \begin{bmatrix} g_1(\xi_2,\eta,\tau) \\ g_2(\xi_2,\eta,\tau) \\ g_3(\xi_2,\eta,\tau) \end{bmatrix} \quad \text{along } \Gamma_2 \tag{56}$$

$$\vdots$$

$$\begin{bmatrix} r \\ \phi \\ z \end{bmatrix} = \begin{bmatrix} h_1(\xi,\eta,\tau_6) \\ h_2(\xi,\eta,\tau_6) \\ h_3(\xi,\eta,\tau_6) \end{bmatrix} \quad \text{along } \Gamma_6$$

These functions represent the bounding surfaces $\Gamma_1 \ldots \Gamma_6$ which for a typical turbine are illustrated in Fig. 39.

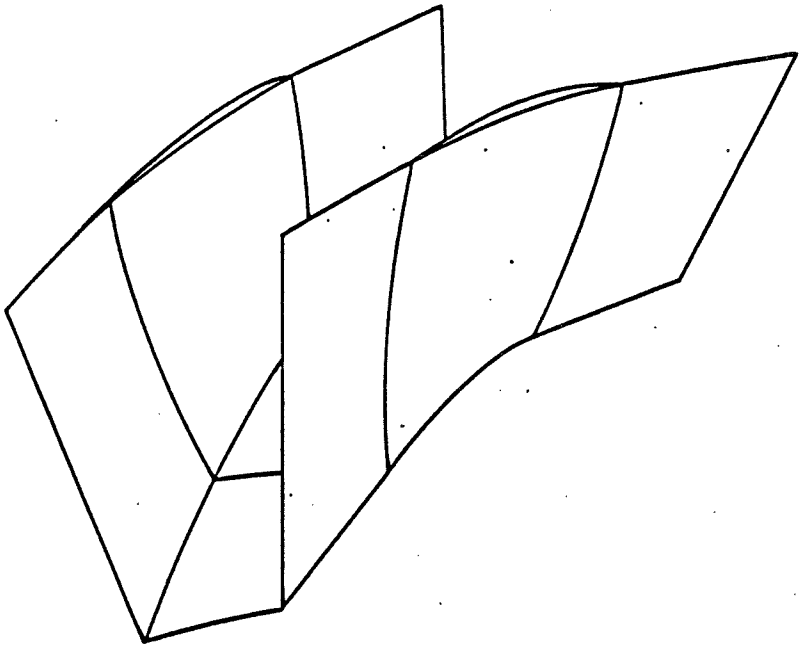


Fig. 39 Illustration of boundary surfaces for a turbomachinery application

This is carried out numerically by discretizing the computational domain into equal intervals in all three directions. Then using central differences, Eq. (54) is replaced by an equivalent set of algebraic equations, one for each coordinate.

Thus at each node (i,j,k) of the computational domain one obtains three such equations. For illustration, the equation for the R coordinate is given in full.

$$
\begin{aligned}
& a'_1 (R_{i+1,j,k} - 2R_{i,j,k} + R_{i-1,j,k}) \\
& + a'_2 (R_{i,j+1,k} - 2R_{i,j,k} + R_{i,j-1,k}) \\
& + a'_3 (R_{i,j,k+1} - 2R_{i,j,k} + R_{i,j,k-1}) \\
& + 2a'_4 (R_{i+1,j+1,k} - R_{i-1,j+1,k} - R_{i+1,j-1,k} + R_{i-1,j-1,k}) \\
& + 2a'_5 (R_{i,j+1,k+1} - R_{i,j-1,k+1} - R_{i,j+1,k-1} + R_{i,j-1,k-1}) \\
& + 2a'_6 (R_{i+1,j,k+1} - R_{i-1,j,k+1} - R_{i+1,j,k-1} + R_{i-1,j,k-1}) \\
& - \frac{1}{R_{i,j,k}} + P'(R_{i+1,j,k} - R_{i-1,j,k}) + Q'(R_{i,j+1,k} - R_{i,j-1,k})
\end{aligned}
\tag{57}
$$

$$+ R'(R_{i,j,k+1} - R_{i,j,k-1}) = 0$$

Similar equations for ϕ, and Z can be obtained by the appropriate substitutions. It is noted that we started this procedure with the Poisson system and these include the forcing terms which are denoted P′, Q′ and R′.

Using a point SOR scheme, expressed in correction form yields (for the R coordinate)

$$\begin{aligned}\frac{2}{\omega}(a'_1 + a'_2 + a'_3)CR_{i,j,k} = {} & CR_{i-1,j,k}(a'_1 - P') + CR_{i,j-1,k}(a'_2 - Q') \\ & + CR_{i,j,k-1}(a'_3 - R') \\ & + 2a'_4(CR_{i-1,j-1,k} - CR_{i+1,j-1,k}) \\ & + 2a'_5(CR_{i,j-1,k-1} - CR_{i,j-1,k+1}) \\ & + 2a'_6(CR_{i-1,j,k-1} - CR_{i+1,j,k-1}) + RR_{i,j,k}\end{aligned} \tag{58}$$

where the residual is

$$\begin{aligned}RR_{i,j,k} = {} & a'_1(R_{i+1,j,k} - 2R_{i,j,k} + R_{i-1,j,k}) \\ & + a'_2(R_{i,j+1,k} - 2R_{i,j,k} + R_{i,j-1,k}) \\ & + a'_3(R_{i,j,k+1} - 2R_{i,j,k} + R_{i,j,k-1}) \\ & + 2a'_4(R_{i+1,j+1,k} - R_{i-1,j+1,k} - R_{i+1,j-1,k} + R_{i-1,j-1,k}) \\ & + 2a'_5(R_{i,j+1,k+1} - R_{i,j-1,k+1} - R_{i,j+1,k-1} + R_{i,j-1,k-1}) \\ & + 2a'_6(R_{i+1,j,k+1} - R_{i-1,j,k+1} - R_{i+1,j,k-1} + R_{i-1,j,k-1}) \\ & - \frac{1}{R_{i,j,k}} + P'(R_{i+1,j,k} - R_{i-1,j,k}) + Q'(R_{i,j+1,k} - R_{i,j-1,k}) \\ & + Q'(R_{i,j,k+1} - R_{i,j,k-1})\end{aligned} \tag{59}$$

Using similar expressions for the axial and circumferential coordinates one can successively correct the coordinate values using

$$\begin{aligned} R^{+}_{i,j,k} &= R_{i,j,k} + CR_{i,j,k} \\ \phi^{+}_{i,j,k} &= \phi_{i,j,k} + C\phi_{i,j,k} \\ Z^{+}_{i,j,k} &= Z_{i,j,k} + CZ_{i,j,k} \end{aligned} \tag{60}$$

These equations were coded and applied to several three-dimensional shapes. The resulting grids are shown in Figs 40 to 45. The convergence of this scheme can be accelerated by using a multigrid technique [1].

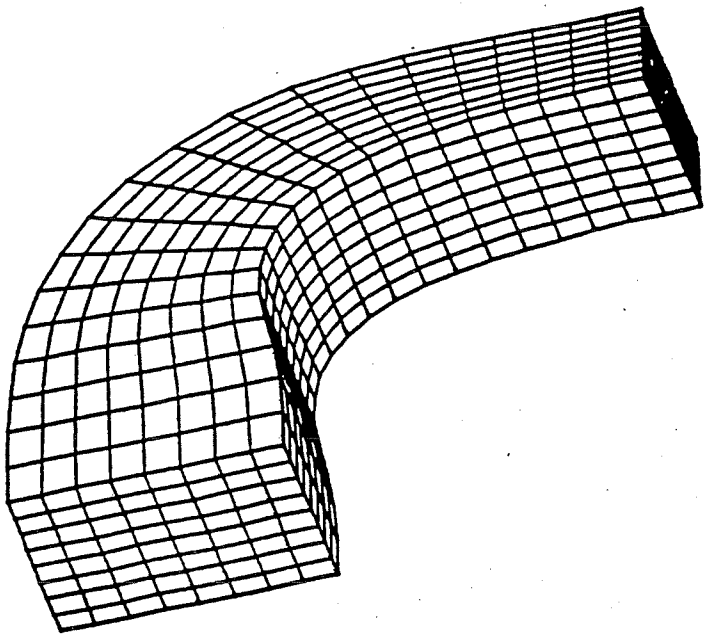


Fig. 40 Axonometric projection of grid for Stanitz' elbow

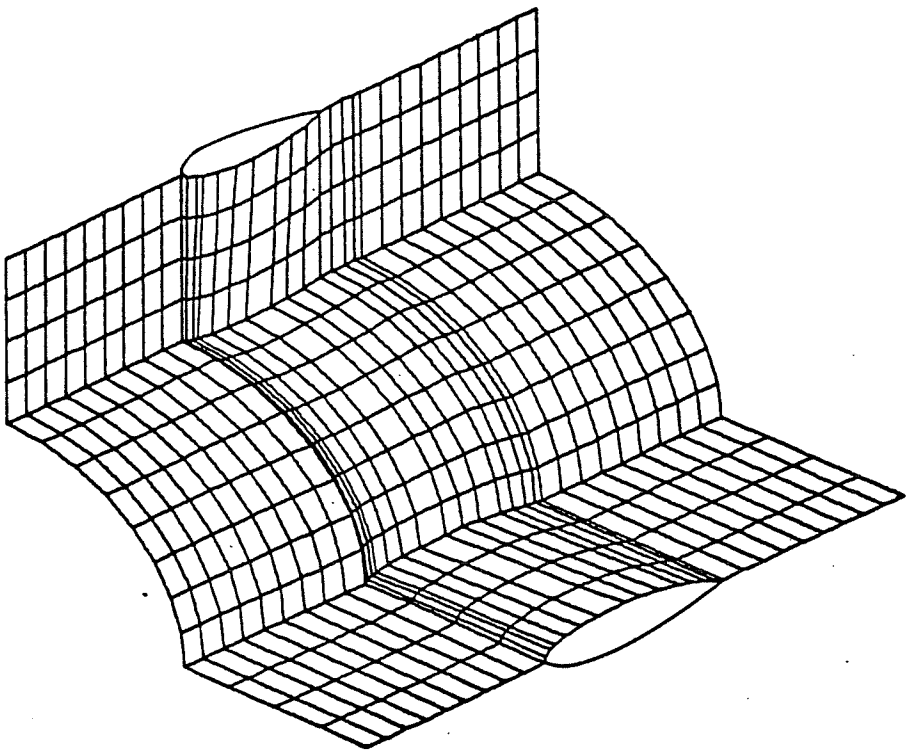


Fig. 41 Grid pattern on blade-to-blade and hub surfaces

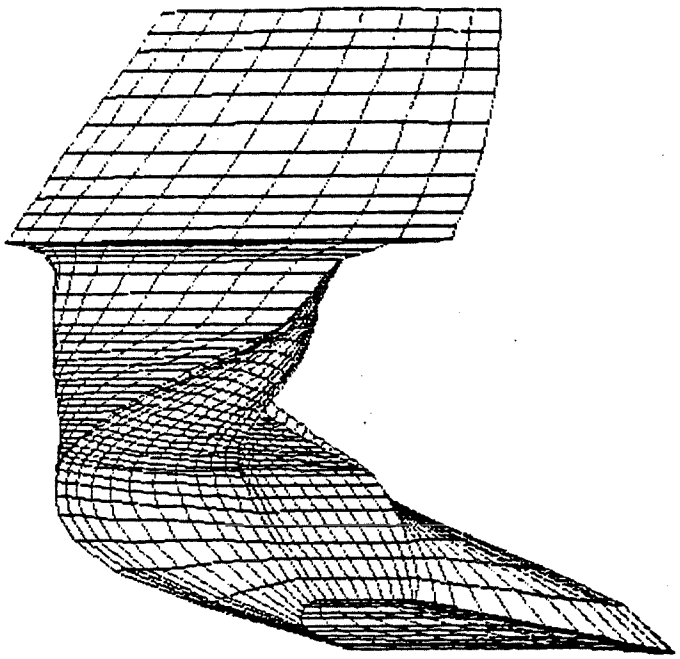
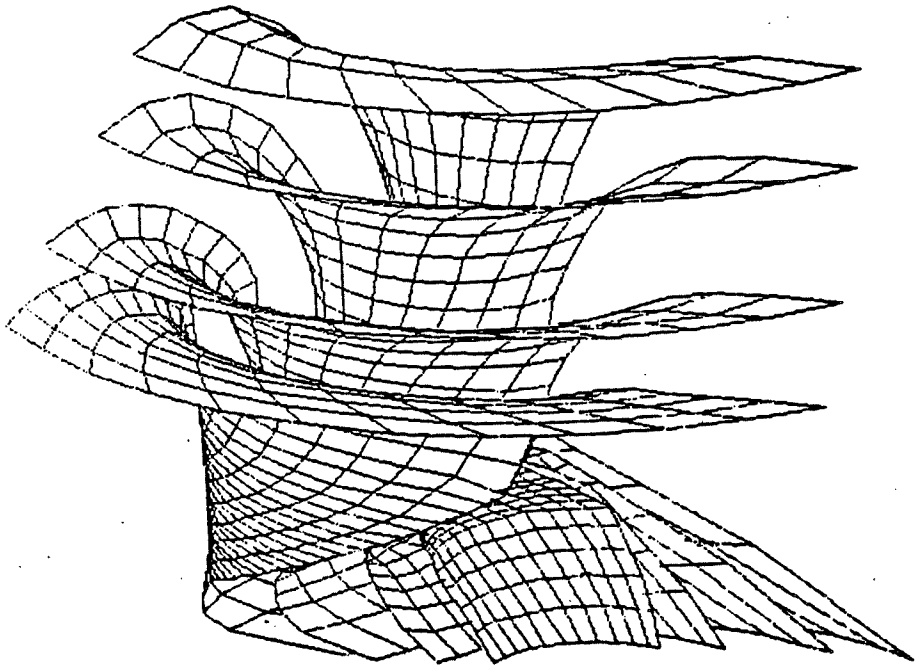


Fig. 42 Grid pattern for 3D blade-to-blade channel of a Francis runner



Fir. 43 Grid surfaces of τ-constant coordinate

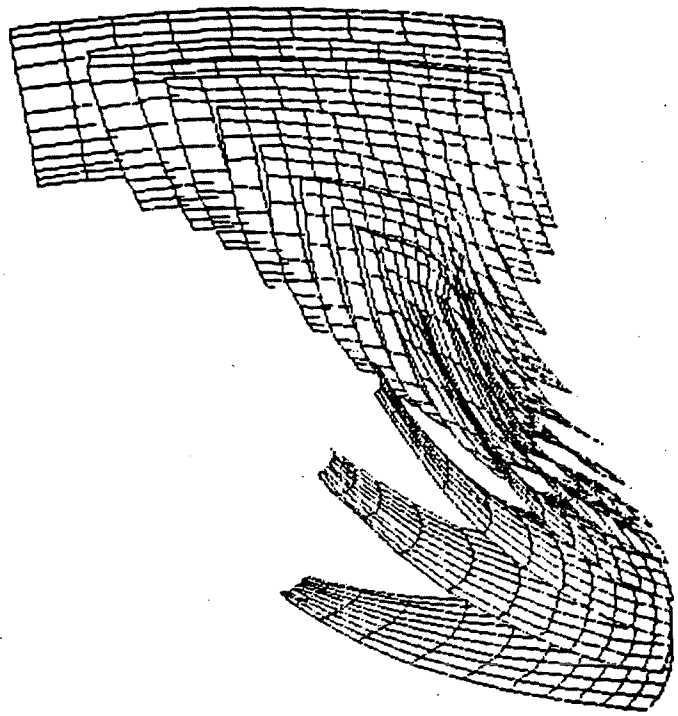
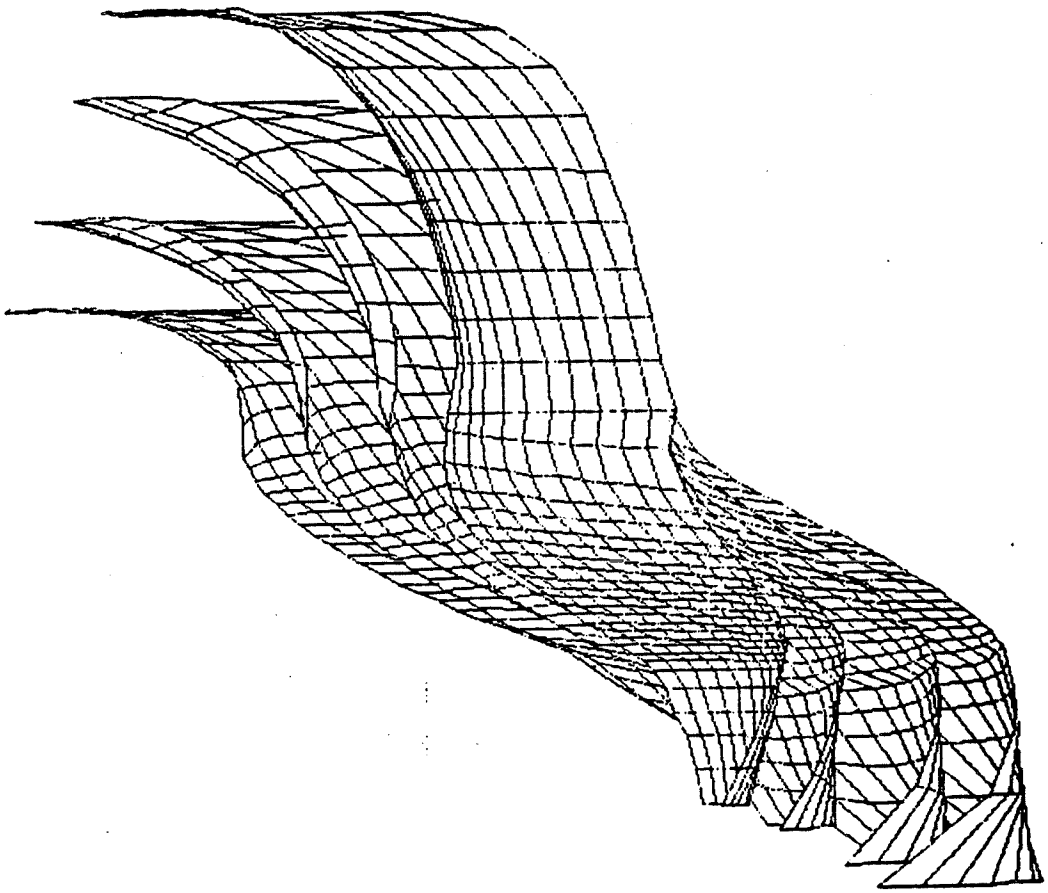


Fig. 44 Grid surfaces of ξ-constant coordinate

Fig. 45 Grid surfaces of η-constant coordinate



7. COMPUTER AIDED GRID DESIGN

7.1 CONTEXT

In this chapter the role of grid generation will be placed in the context of computer integrated manufacturing (CIM) of turbomachinery. The overall process of turbine design comprises three steps which are the hydraulic or aerodynamic design, the mechanical design and finally the manufacturing process. In the present work, the emphasis is on the hydraulic or aerodynamic design. This is further subdivided into four parts:

i) geometric modelling

ii) computational modelling

iii) simulation or computational fluid dynamics

and iv) flow (solution) visualization and analysis

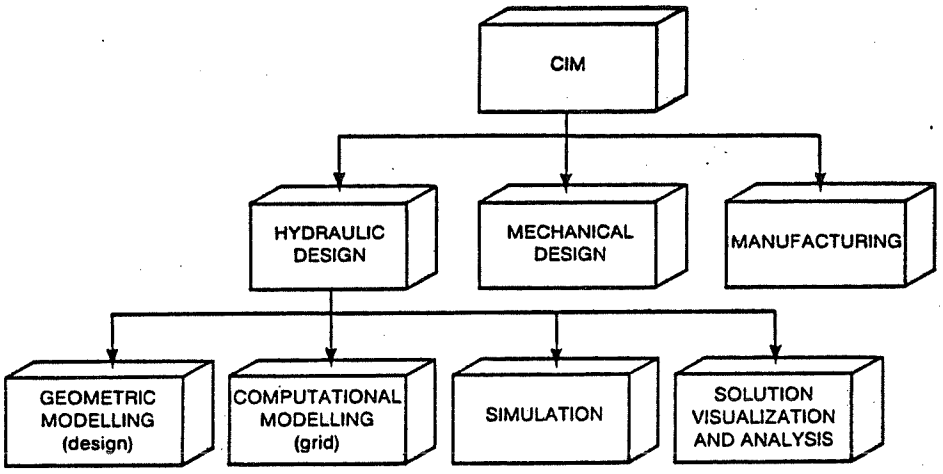


Fig. 46 Context of Computer aided grid design

The subject of grid generation falls in the second part that is computational modelling. It is a somewhat broader topic than simply the grid generation. It includes the aspects of management, manipulation and display of (geometric) information. This reflects the role played by the computer and its peripheral equipment. It is a triviality to say that the methods described in the previous chapters could not be realized without modern computing facilities. And by this, we

do not merely mean the numerical solution of the grid equations, we mean the handling of the enormous quantities of data required and produced by this type of application. This data is geometric and is an ordered collection of points which represent the surfaces and volumes of the computational domains in two or three dimensional space.

It is fair to say that without computer assistance at a sophisticated level none but the simplest examples could be carried out. For grid generation techniques to become a truly functional tool in CFD it must be embedded into interactive software packages which manage the data flow from the component geometry to the simulation codes. Such packages should fulfill the following functions:

1. The creation and the modification of a model; the designer can "edit" the model by means of commands which operate on basic entities such as a point, a profile, or one of the parts making up the turbine (the blade, the hub or the shroud).

2. The refinement of the model by the distribution of points over each profile defining the blade, and then by the interpolation of new intermediate profiles between the hub and the shroud.

3. The construction of surfaces delimiting the blade-to-blade channel boundaries (the calculation domain).

4. The calculation of a body-fitted coordinate system, inside the channel, by the solution of the system of differential equations described in Chapter 3.

Figure 47 illustrates how the information flows among these steps.

7.2 MODELLING

The objective of the modelling process is to create a geometric representation of the various objects under consideration so that they can be manipulated and displayed. In the present application these are the turbine and the computational domain.

A turbine, a compressor or a pump is composed of three main elements: the blades, the hub and the shroud. Since the blades are all identical, the definition of only one is sufficient for the geometric description of a turbine.

The graphic entity corresponding to the body of the machine is, of course, a volume. But, for the hydraulic/aerodynamic design, the interior surfaces (that are in contact with fluid) are of more interest than the body of

the turbine itself. That is why, rather than considering the three elements of the turbines as solids or volumes, we characterize them by the surfaces that they are bounded by. The blade, defined between the hub and the shroud can therefore be represented by a surface folded upon itself in a three-dimensionnal space, while the hub and the shroud are described by surfaces of revolution.

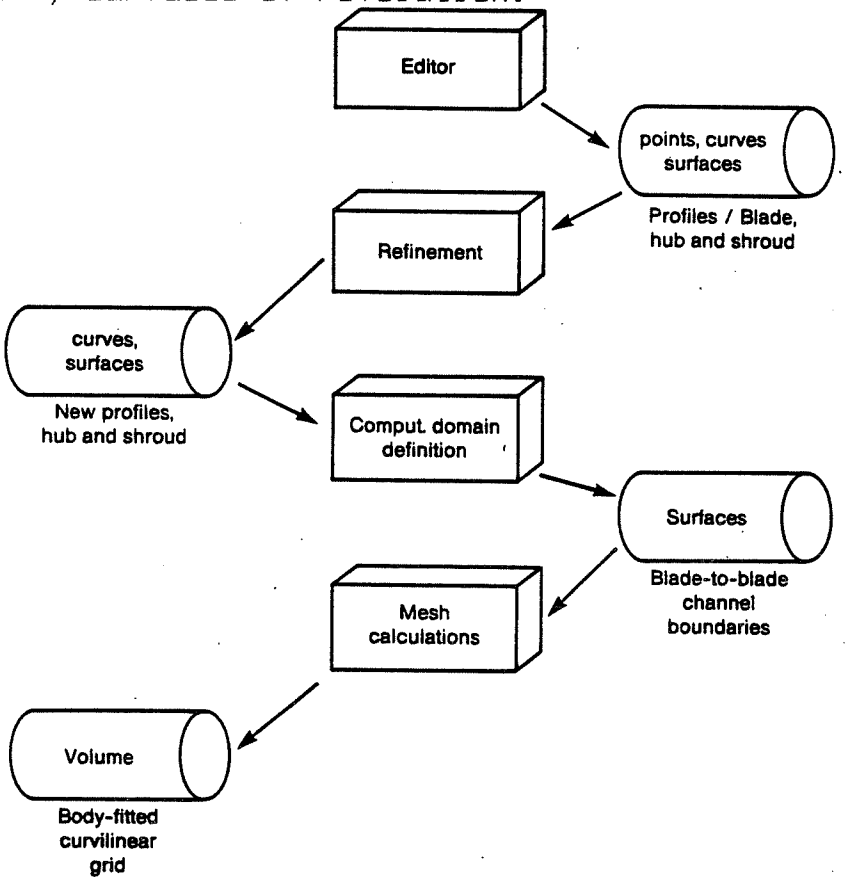


Fig. 47 Geometric and computational modelling

The blade surface is in turn characterized by two sets of lines: the first, traversing from the hub to the shroud, and the second set, going around the blade. These two families of curves form a grid where the intersection of one line of a family with that of another, results in a node. It is these points that serve as a base for defining the blade by means of interpolation, with the use of parametrized cubic splines.

The surfaces of revolution characterizing the hub and the shroud are defined by a curve in a given plane and an axis of revolution. This curve is defined by fitting a 2D parametrized cubic spline through the coordinates of known points.

The second type of object, the computational domain, is not physical in nature but purely geometric.

For the present application, it is a closed volume called a "blade-to-blade channel", and the set of surfaces that bound this channel will be referred in this work as the "shell". This shell consists of the pressure side and the suction side of two consecutive blades, four ruled surfaces, the inlet and outlet planes, the hub, and the shroud, as shown in Figs. 39 and 48.

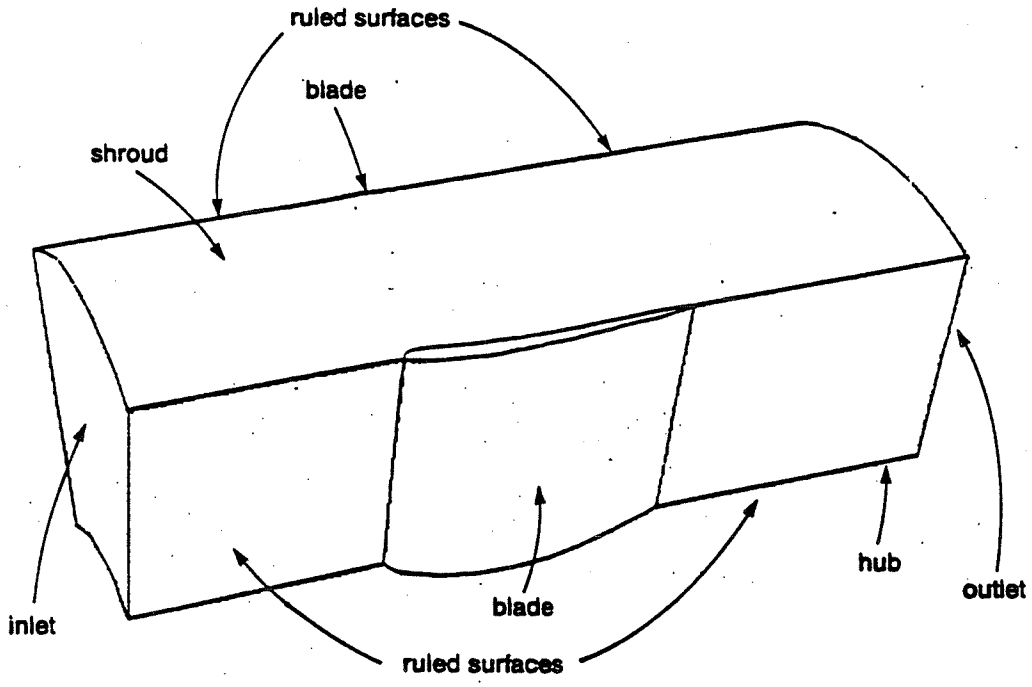


Fig. 48 Definition of computational domain or "shell"

Finally, the last object is the grid itself, which is a collection of ordered points.

The entire modelling process can be carried out in one of three different coordinate systems: cartesian (x,y,z), cylindrical (r,ϕ,z) and toroidal (R,θ,ϕ). A certain similarity in their use is established, if it is recognized that for all systems, the three coordinates indicate respectively the hub to shroud, blade to blade, and inlet to outlet directions.

Taking advantage of this convention all operations are performed in a general manner, on coordinates one, two and three, that is in the computational space. Internal to the modules the actual physical system in not considered. Rather, it is the user who interprets the results appropriately, according to the coordinate system that the choses to use (Fig. 49); a displacement in the direction of one of the coordinate can hence be considered as a rotation or as a translation (depending on whether the coordinate is an angle or not). The application of certain functions may at times seem disconcerting in certain referentials, but do however appear natural enough once their mechanisms are understood.

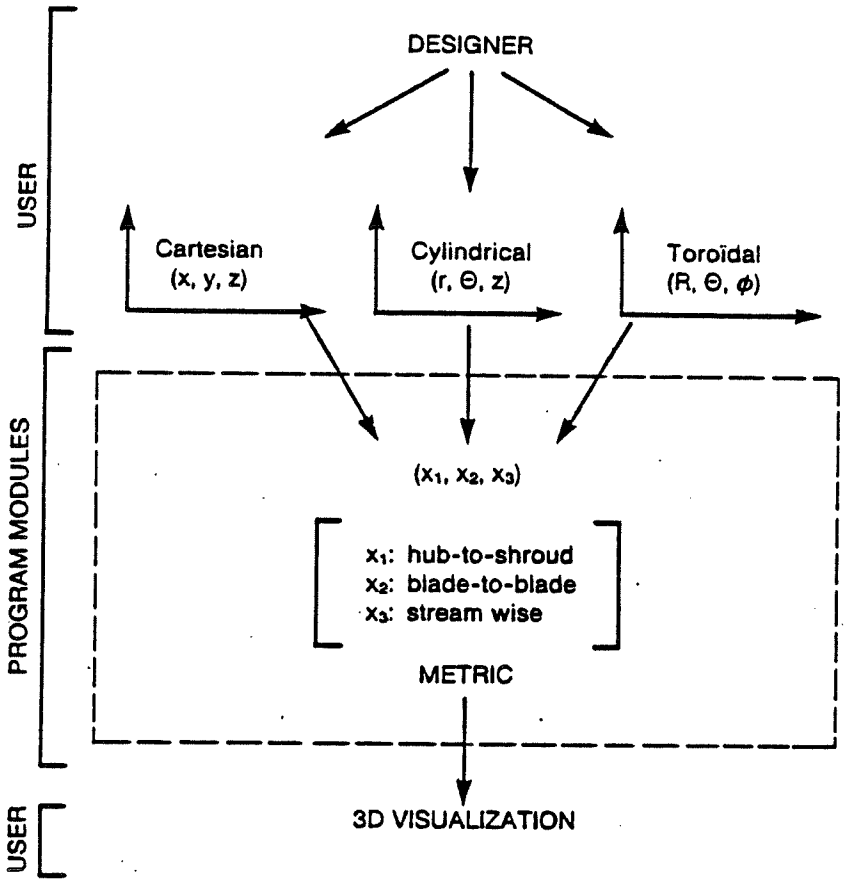


Fig. 49 Coordinate Systems

7.3 MODEL BUILDING

The building of a model for the blade, the hub and shroud is carried out using a graphics editor. The building blocks are basic geometric entities such as points, lines, profiles etc and a number of commands to perform certain operations.

The graphic editor is a menu driven interactive program. The first level of menus permits the choice of the basic graphic entity (point, curve, or surface) to work with, and other general functions such as the display window management. The submenus, at the second level, contain the functions that are related to a particular entity. These functions may vary, depending on the current modelled element and the projection used. If, at any level, parameters or options have to be chosen, these will be requested as a final level of input, while offering a default value. All user's input are entered either directly from the keyboard, or with the use of the cursor (for selecting a particular menu command or option, as well as specifying a position on the screen). Moreover, as a necessary characteristic in an interactive program, all responses from the user are validated and various error messages are provided. Figure 50 shows a typical screen display where the main menu and a submenu are present on the screen, along with a profile being processed.

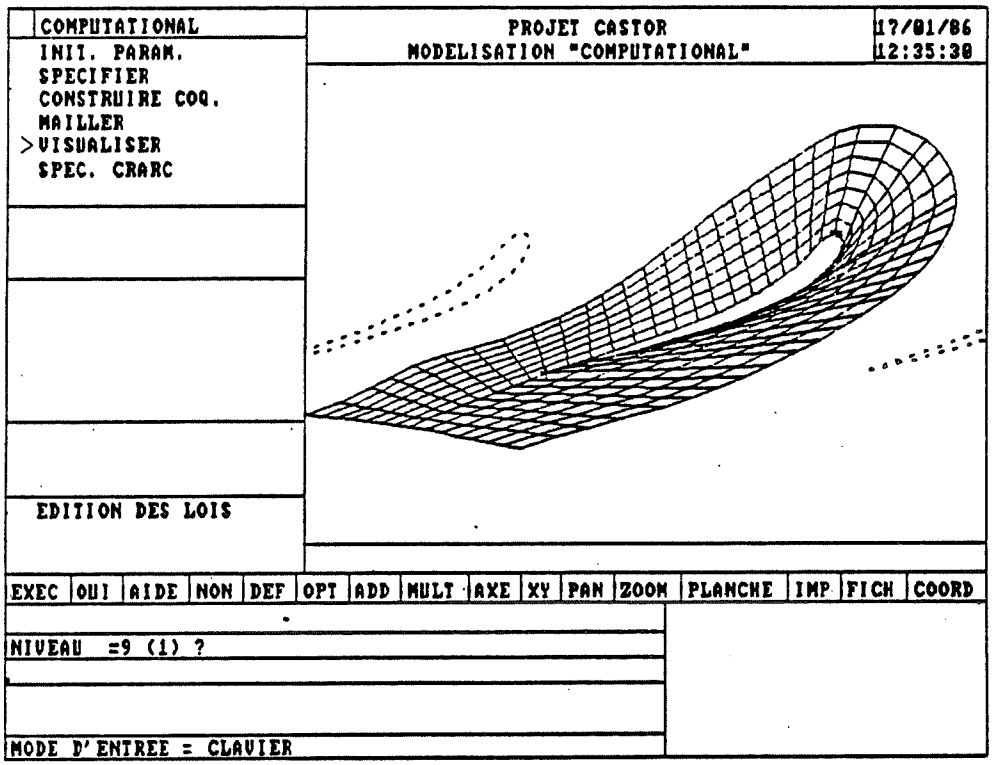


Fig. 50 Typical screen display of the graphic editor

The available functions fall into three categories:

1. Those performed on a surface:

   - recalling either the entire model's geometry, or only part of it from a file (down to a surface level: the hub, the shroud, the entire blade, or only part of it).
   - saving the entire model's geometry in a file.

   - reordering of profiles, that is modifying the blade surface. This can be done automatically by "sorting" the profiles using the first coordinate of their first point, and hence, ordering them according to their relative "height".

2. Those performed on a curve (profile, hub or shroud):

   - translation.

   - rotation (around the origin or a specific point).

   - scaling (centered on the origin or a specific point).

   - addition of profiles by duplicating them.

   - deletion of profiles.

3. Those performed on a point (with the aid of the graphic cursor):

   - creation and insertion between other levels of a new profile; points are entered on the screen on a constant coordinate surface.

   - modification of either profiles or curves representing the hub and the shroud, by adding, deleting, displacing, duplicating, permuting, or renumbering the points that define them.

With the help of this modeler, one can create various type of turbomachines as shown in Fig. 51.

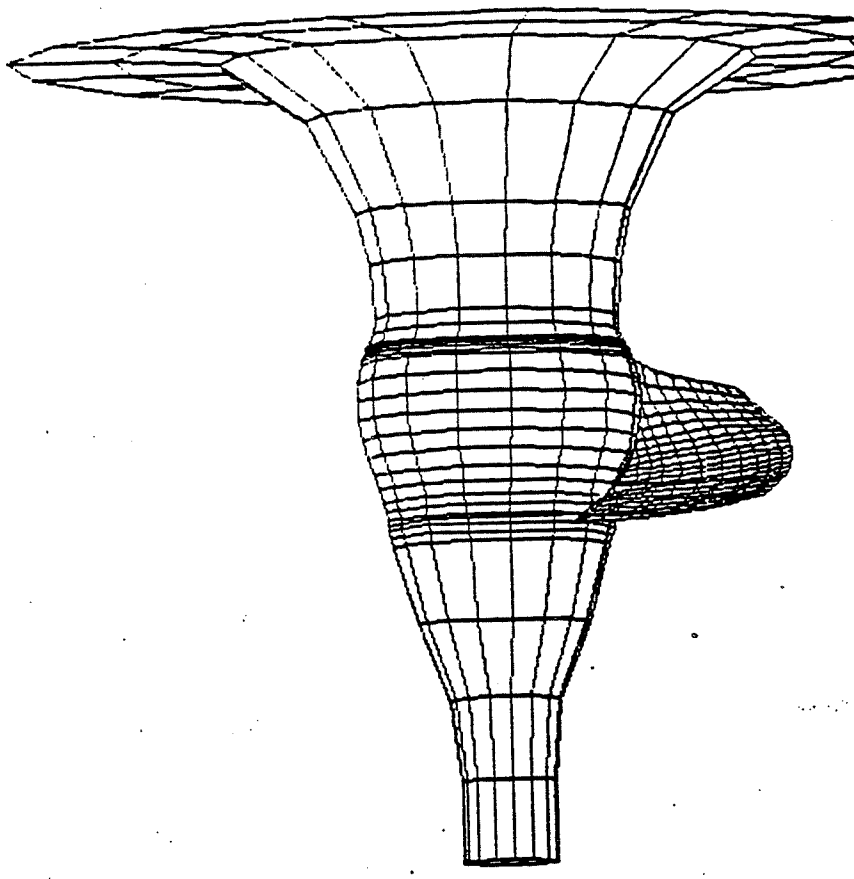
7.4 DOMAIN CONSTRUCTION

The computational domain, or shell, has been defined previously. It uses the geometry data of the blade, hub and shroud plus additional information concerning the inlet and outlet as well as the periodic boundaries.

It is constructed in two steps:

1. the limits of the ruled surfaces are defined by extending lines from the leading (trailing) edge of profiles on the hub and shrouds. These lines lie on the surface of the hub and the shroud, going from the blade to the inlet (exit) planes. The angle and length of these lines are given by the user. Points are then distributed along the lines according to a concentration parameter, also given by the user.

2. by entering the number of blades of the turbine, the user indirectly provides the blade-to-blade channel distance i.e. the width of the shell. The concentration and the number of points in the blade-to-blade direction are given by the user.



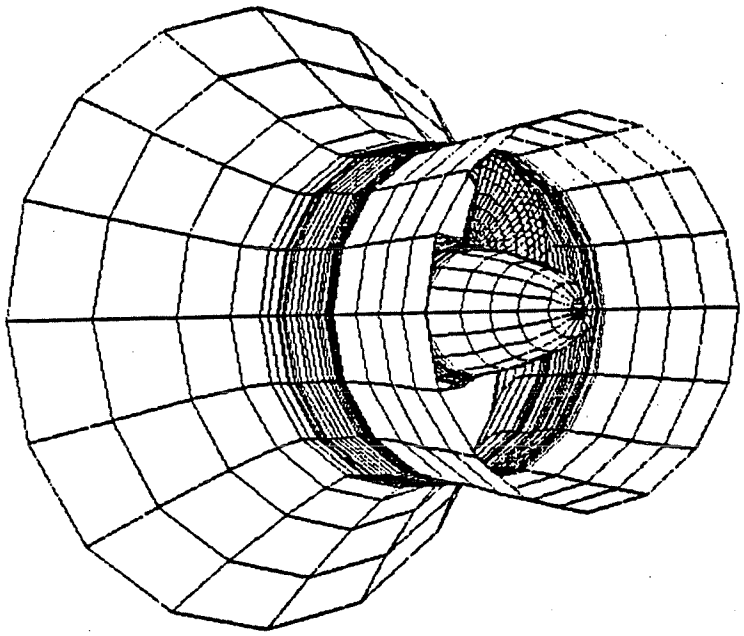
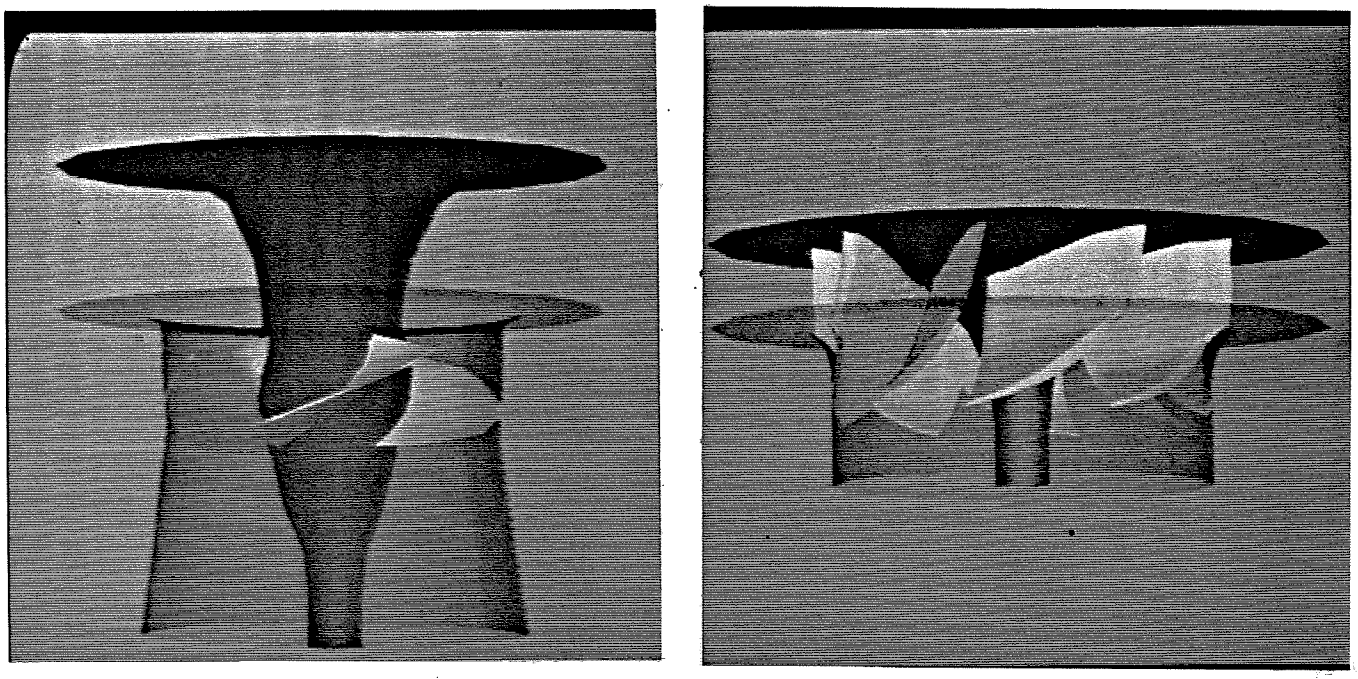


Fig. 51a Wire frame representation of modelled turbines



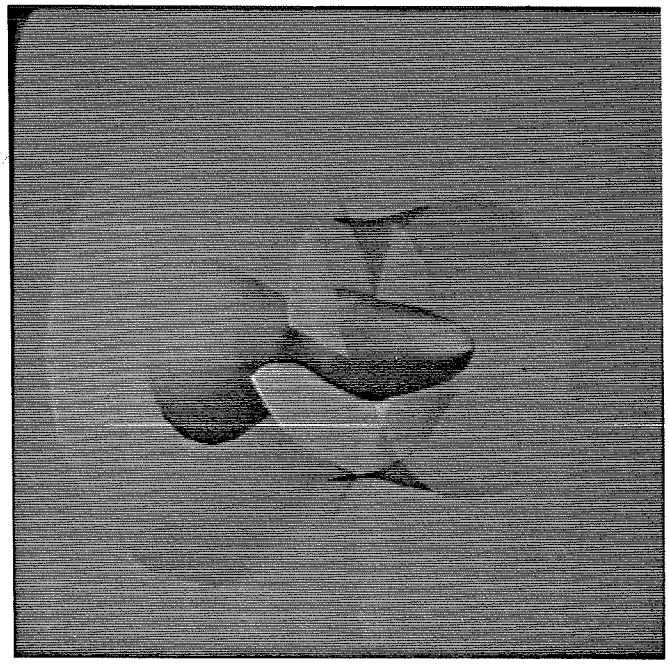


Fig. 51b Solid representation of modelled turbines

Finally, using the shape of the shell as the six surfaces Γ_1, Γ_2, ... Γ_6 described in Fig. 38 the grid equations are solved using the techniques described in Chapter 6 This yields a mapping of the computational space to the physical domain of the blade-to-blade channel as illustrated in Fig. 52.

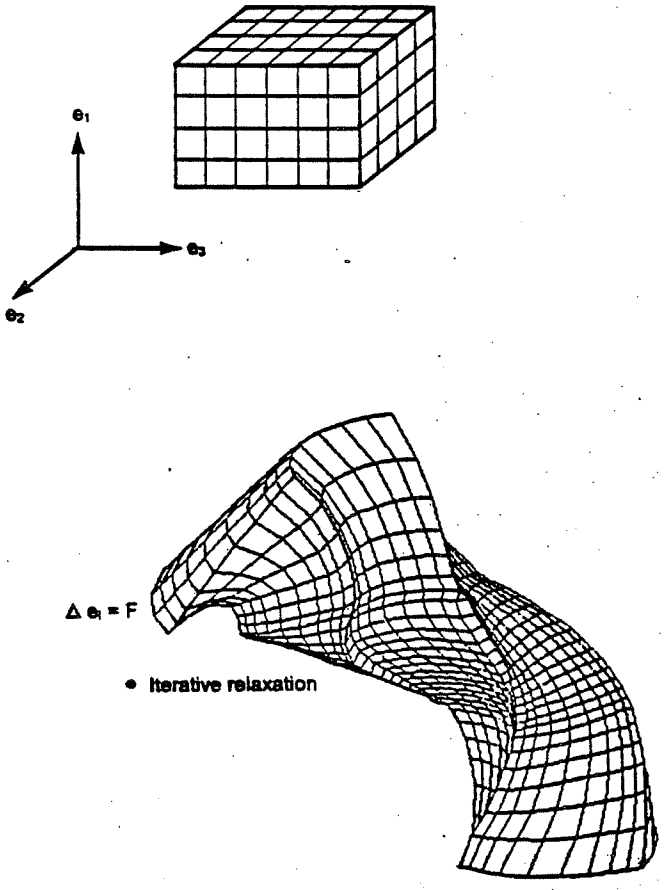


Fig. 52 Automatic mesh generation

The results of such computations can be displayed graphically and manipulated (rotations, zoom, etc) one or several coordinate surfaces at a time using a module whose typical output is shown in Fig. 53.

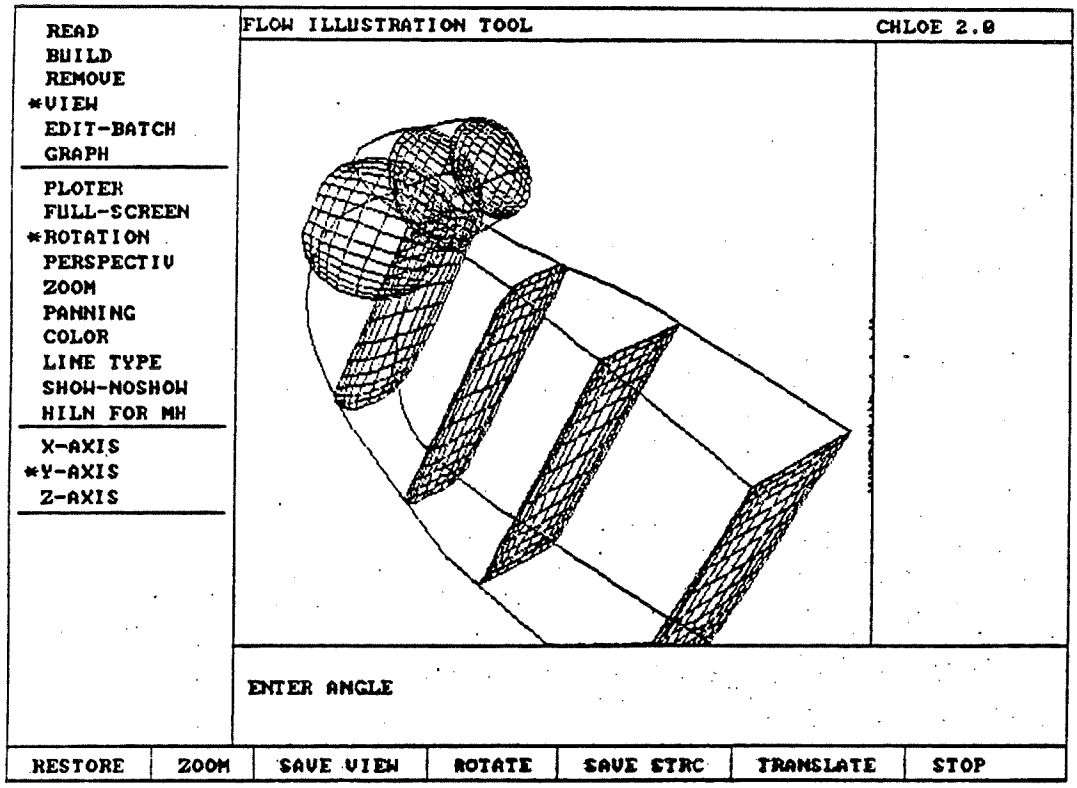


Fig. 53 Visualization module

7.5 DIALOGUE AND DISPLAY

The grid generation methodology described in this work has been formulated numerically and cast as a set of techniques imbedded in software programs. These are interfaced with the user by interactive processors which handle, through a series of dynamic menus, the modelling of the turbine, the computational domain and the grid generation.

The display and modification of three-dimensionnal objects, on a graphic terminal having only two dimensions, undoubtedly entails some limitations. The visualization of 3-D objects by means of various projections is thus necessary and these succeed in rendering rather completely the true geometry characteristics of these objects. The inverse communication, that is from the designer to the program, is more difficult, because the dialogue is restricted by physical devices such as the flat screen or the cursor. These do not allow for the passage of all the geometrical data that one

would desire: only two dimensions can be input to the program at one time, due to the absence of true depth at the screen.

Thus, to develop an acceptable method of communication, one must judiciously combine the use of the available devices, with that of a particular projection.

One can easily project the blade profiles on three different surfaces, each corresponding to a constant value of each coordinate (orthogonal projection). This type of projection, because of the speed and ease with which it can be implemented, was retained. This stems from the fact that one of the coordinates is simply dropped while displaying the other two. The creation and modification of profiles is therefore easily made possible, since the screen's surface corresponds to the developed surface containing two of the profile's dimensions.

A typical screen for the geometric modelling is shown in Fig. 50 which shows how Certain pertinent information is always displayed within the screen's graphic frame:

- profile number
- coordinate dyad indicating the current projection
- file name

The options available during display mode are:

- choice of a projection to display a profile.
- choice of the displayed element (pointer), as well as of the interval of profiles to display along with the chosen element, and the presence or not of the hub and the shroud.
- modification of the display window by translations, scaling (zoom), specifying two new corner points (new frame), or allowing the program to automatically calculate one, slightly larger than the current element.
- display of all profiles at one time in a three-dimensional cartesian referential, for a better grasp of the turbomachine's actual shape (with possibilities of rotating the model with respect to either one of the three cartesian axes). This is done simply by transforming, if necessary, the current referential in a cartesian one.
- choice of the symbol size (i.e. the point's identifier).

- possibility of refreshing the display when too much obsolete data appears on the screen, or simply to visualize the result of a modification to an element.

There are two levels of display. The first uses line drawing for quick display of models and grids. A second mode provides a more sophisticated image with hidden line removal, color, shading transparency. This is achieved using a translator to the MOVIE.BYU software:

The overall display structure is shown in Fig. 54.

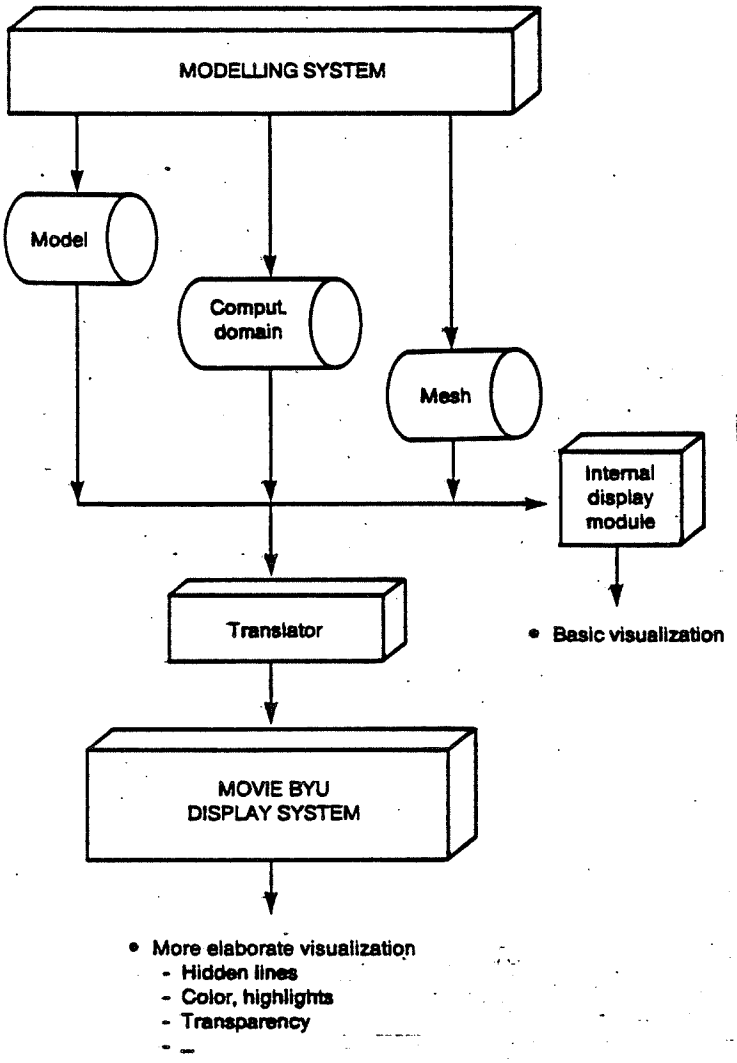
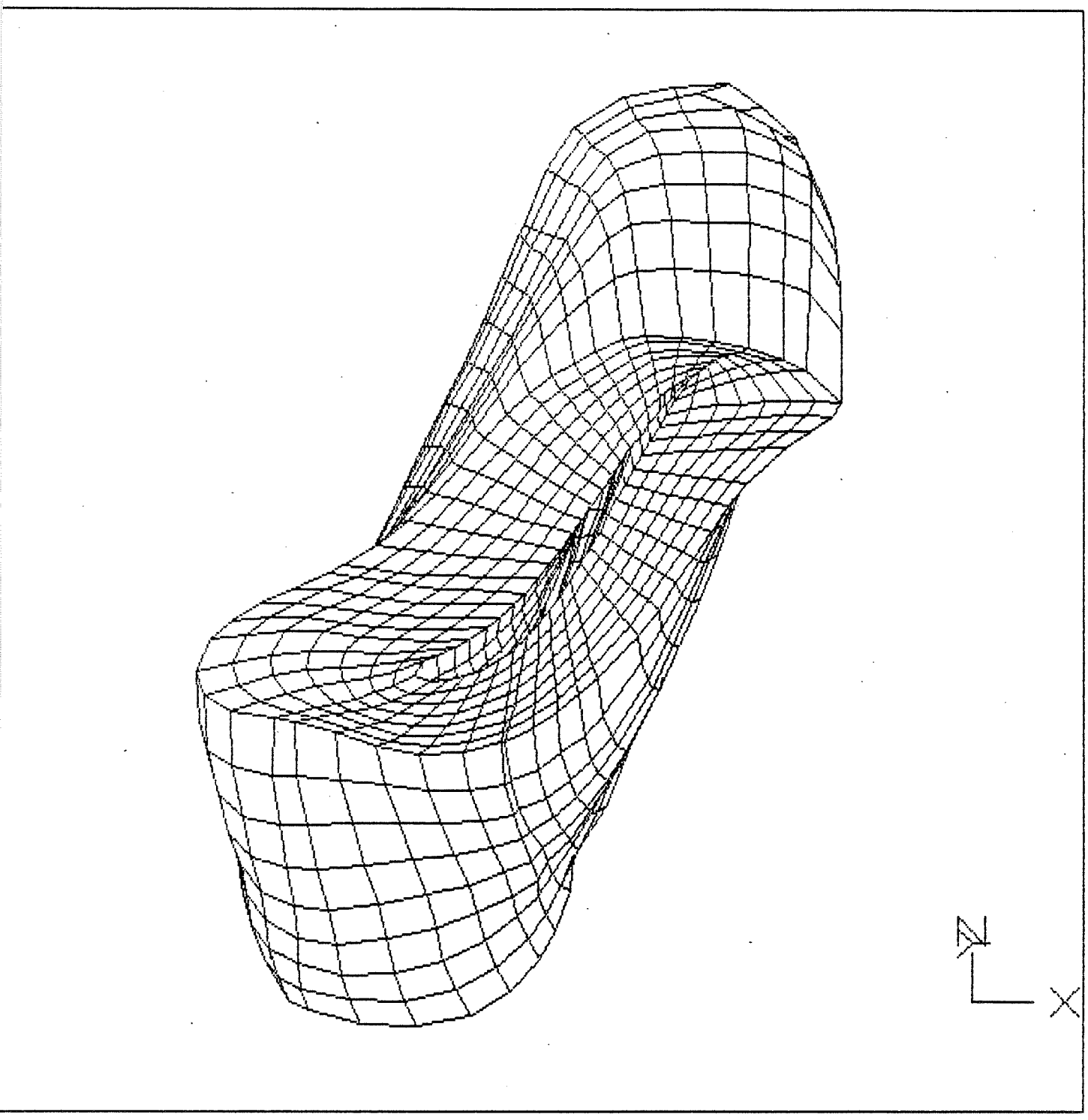
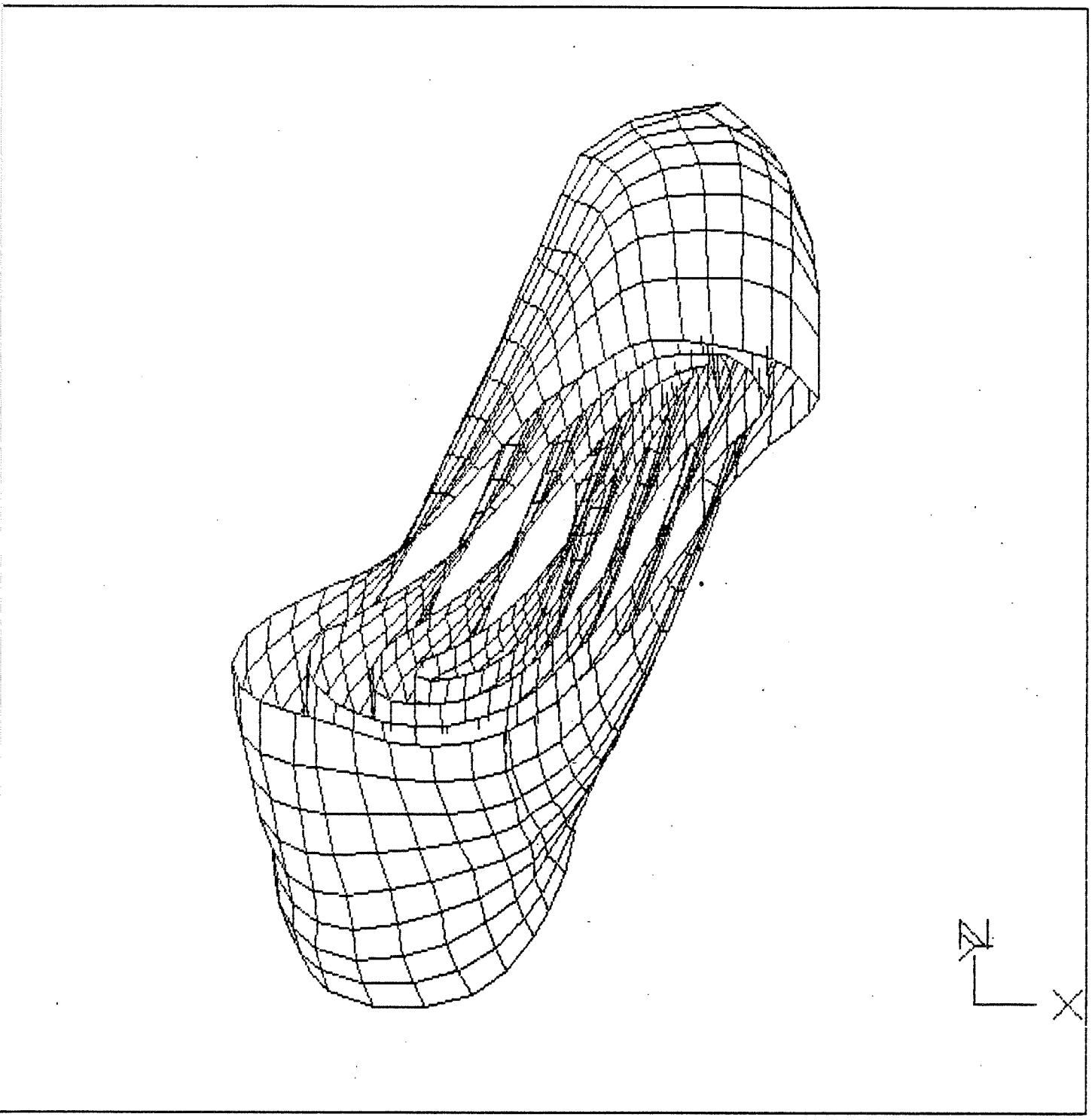
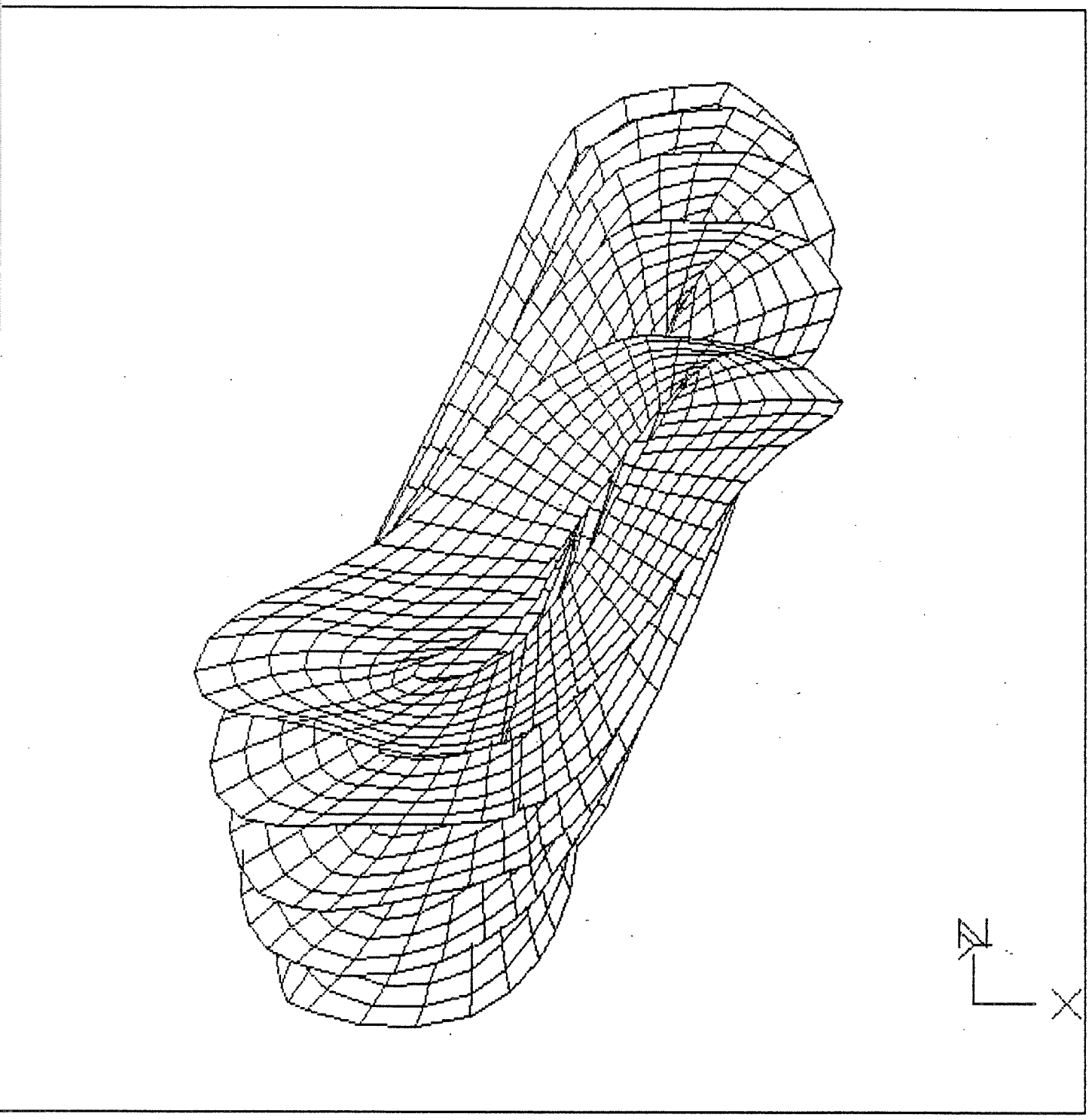
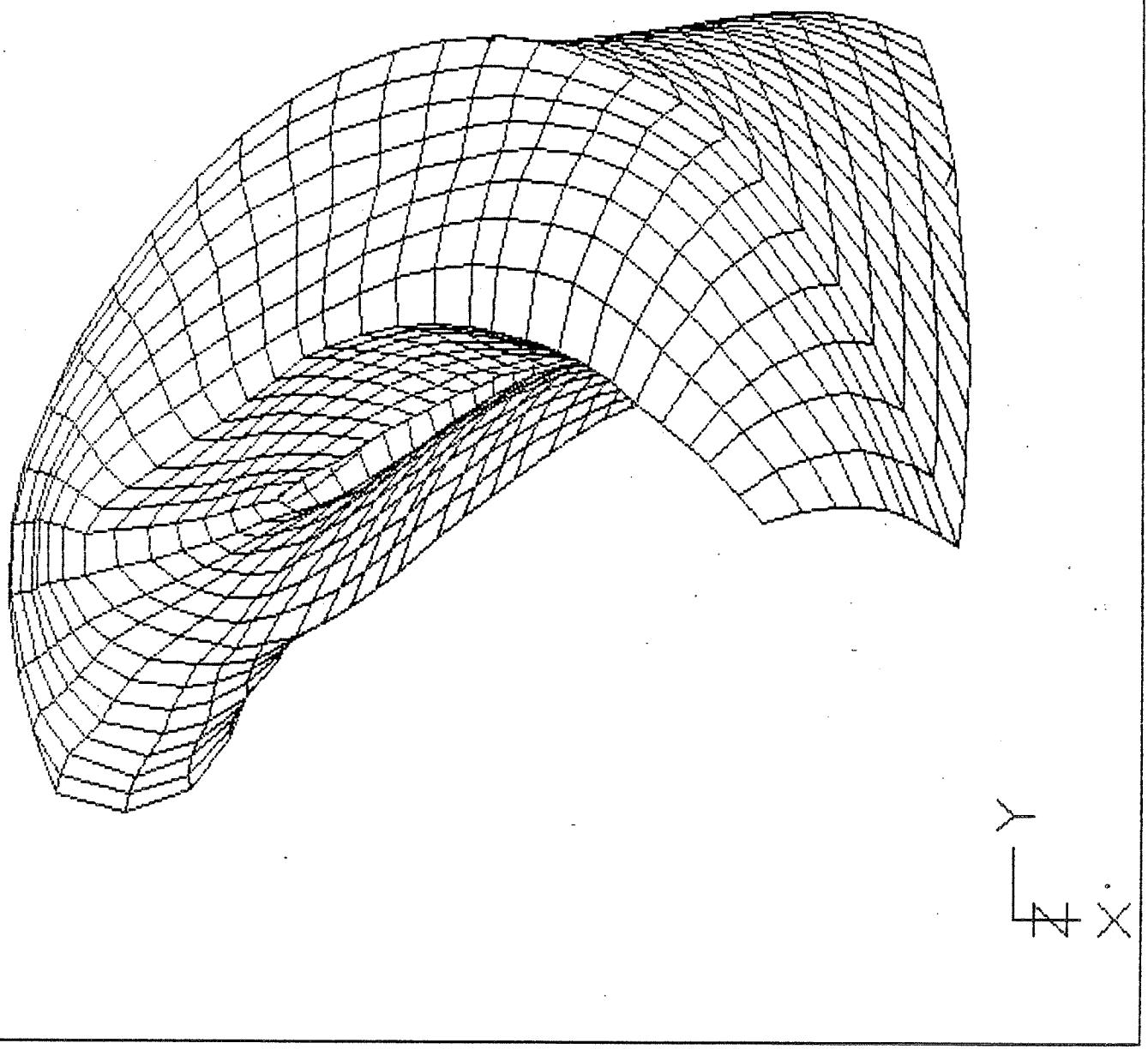


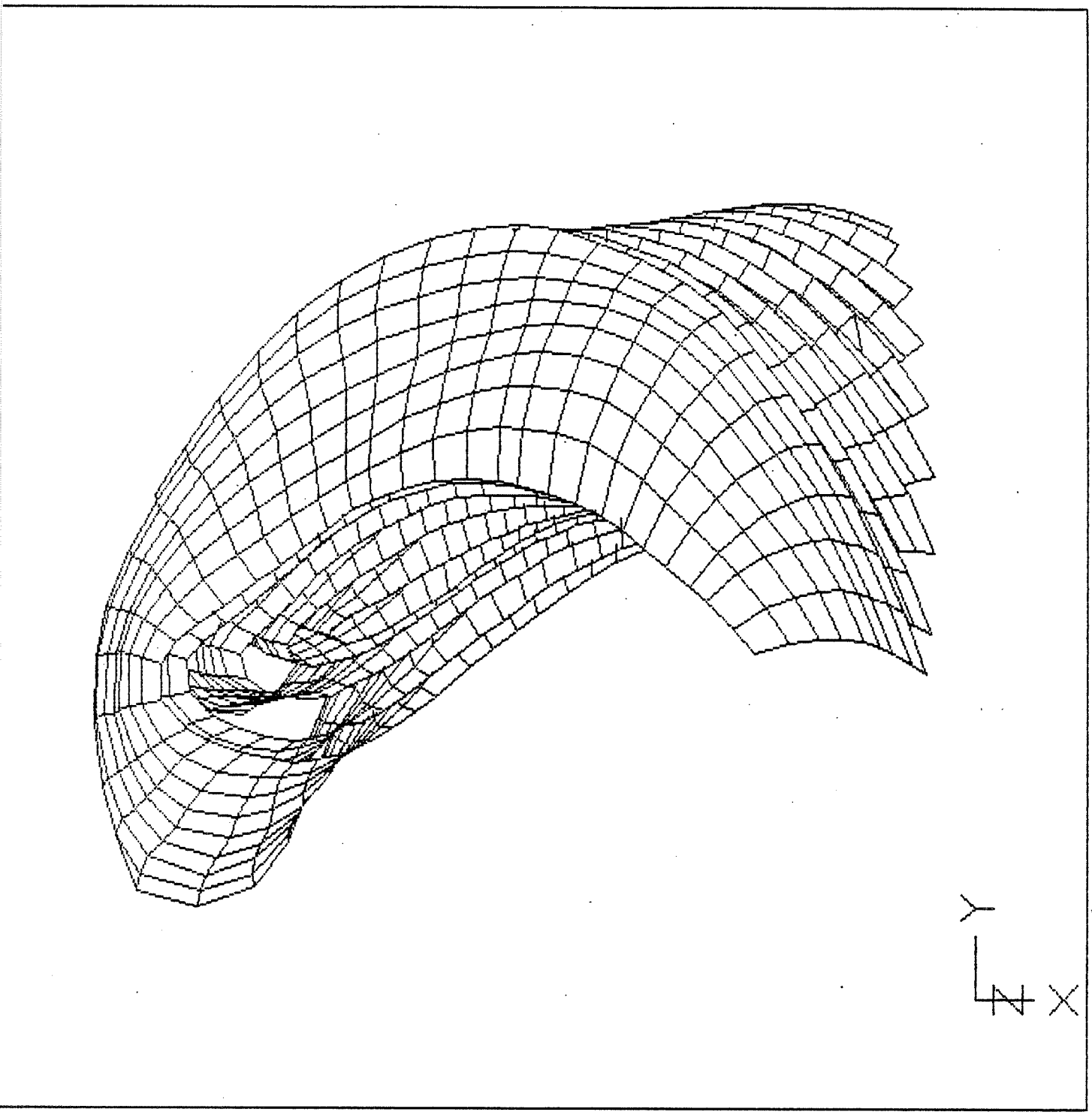
Fig. 54 Data structure for graphic display

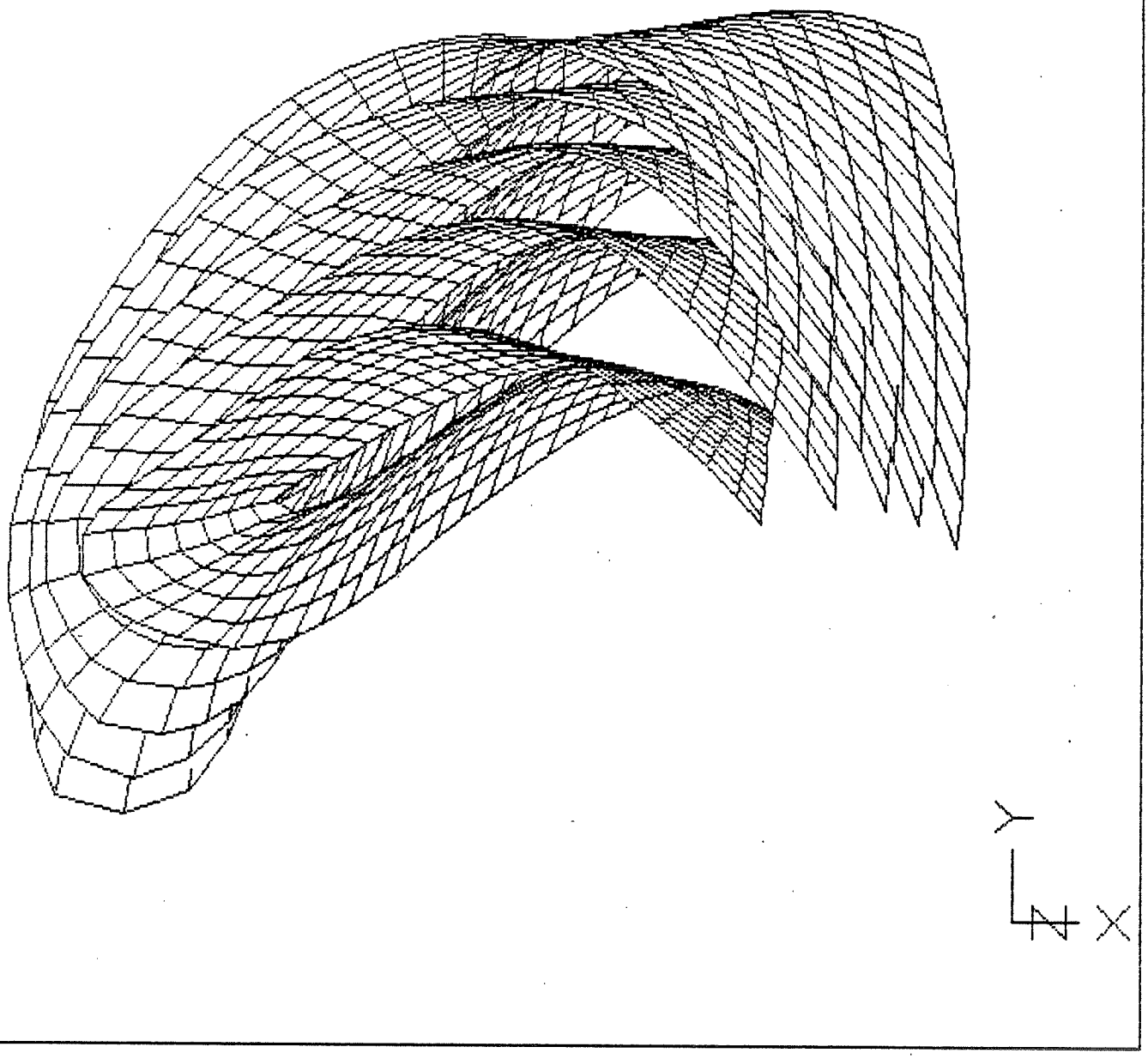

Z
X


Z
X


Z
X


Y
Z
X


Y
X


Y
X

BIBLIOGRAPHY

R. Camaréro, M. Reggio "A multigrid scheme for 3-D body fitted coordinates in turbomachinery applications", ASME J. of Fluids Eng., Vol. 105, March 1983, p. 76.

R. Camaréro, M. Younis "Efficient generation of body-fitted coordinates for cascade using Multigrid.", AIAA J., Vol. 18 No 5, May 1980, p. 487.

Kennon S. R., Dulikravich G. S, "Efficient turbomachinery Grid Generation Using Traupd's Conformal Mapping", Adv. in grid generation, ASME Fluids Engineering Conference, Houston, June 1983.

K. N. Ghia, U. Ghia (Ed.) Advances in grid generation, FED-Vol 5, ASME Applied Mechanics; Bioengineering and fluids Engineering conference, Houston, 1983

K. N. Ghia, U. Ghia, "Numerical generation of a system of Curvilinear Coordinates for turbine cascade Flow analysis", Report No AFL 75-4-17, University of Cincinnati, April 1975.

J. F. Thompson, Z. U. A. Warzi, C. W. Mastin, Numerical grid generation, North-Holland, 1985