NASA Technical Memorandum 86753

Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations

Joseph L. Steger and Yehia M. Rizk

June 1985

NASA
National Aeronautics and Space Administration

NASA Technical Memorandum 86753

Generation of Three-Dimensional Body-Fitted Coordinates Using Hyperbolic Partial Differential Equations

Joseph L. Steger, Ames Research Center, Moffett Field, California
Yehia M. Rizk, Informatics General Corporation, Palo Alto, California

June 1985

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field California 94035

GENERATION OF THREE DIMENSIONAL BODY FITTED COORDINATES USING HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS

JOSEPH L STEGER AND Y. M. RIZK

SUMMARY

An efficient numerical mesh generation scheme capable of creating orthogonal or nearly orthogonal grids about moderately complex three dimensional configurations is described The mesh is obtained by marching outward from a user specified grid on the the body surface. Using a spherical grid topology, grids have been generated about full span rectangular wings and a simplified space shuttle orbiter.

INTRODUCTION

Body conforming curvilinear grids are often used in finite difference flow field simulations One reason for this is that the application of boundary conditions can be simplified in finite difference calculations if grid lines coincide with boundary lines. This is especially important in high Reynolds number viscous flow simulation in which high flow gradients near the body surface must be resolved

The task of generating a satisfactory body conforming coordinate system is not easy. The grids must not be too distorted, they should have smooth variation. and they should be clustered to flow field action regions – typically near boundary surfaces. Moreover, the grids should be generated in an automatic manner that requires a minimum of user input.

One approach for generating body conforming grids with minimum user input has been to solve a set of partial differential equations. In this technique level lines of $\xi(x,y,z)$, $\eta(x,y,z)$, and $\varsigma(x,y,z)$ that have monotone variation are sought as a solution of a set of partial differential equations. Generally, values of ξ, η, and ς are user-specified on the boundary surface, and constraints expressed as differential equations are used to develop the grid away from the boundaries. The most popular such approach requires the solution of a set of elliptic equations that satisfy the maximum principle [1-5]; however, hyperbolic [6,7] and parabolic [8] governing equations have been used as well, at least in two dimensional applications.

In this paper one way of extending the hyperbolic grid generation method of Steger and Chaussee [6] to three dimensions is developed. In two dimensions this grid generation method requires the solution of two partial differential constraints

$$\xi_x \eta_x + \xi_y \eta_y = 0 \tag{1a}$$

$$\xi_x \eta_y - \xi_y \eta_x = (\Delta V)^{-1} \tag{1b}$$

where the subscripts x and y denote partial derivatives These equations are transformed to ξ, η computational space as

$$x_\xi x_\eta + y_\xi y_\eta = 0 \tag{2a}$$

$$x_\xi y_\eta - x_\eta y_\xi = \Delta V \tag{2b}$$

and are solved by marching in η from an initial data plane $\eta(x, y) = const$ The first equation is a constraint of orthogonality. The second equation controls the mesh spacing with the user specifying the mesh control volume ΔV (actually area in two dimensions). A linearized version of Eqs. (2) is readily shown to be hyperbolic and suitable for marching in η. Equations (2) are solved in computational space to give the x, y location of the specified $\xi = const$ and $\eta = const$ grid lines

The two partial differential equations, expressed as either Eqs. (1) or (2), have been referred to as a cell-volume procedure for grid generation For two dimensional external flow simulation in which the outer flow boundary need not be precisely located, this cell-volume hyperbolic partial differential grid generation procedure has been found to be an efficient way of generating orthogonal or nearly orthogonal grids In the next section, a three dimensional extension of this procedure is developed

THREE DIMENSIONAL GRID GENERATION EQUATIONS

A body fitted exterior grid about an arbitrary closed boundary surface is desired. Only a simple topology such as that illustrated in Fig. (1) will be considered here. The body surface is chosen to coincide with $\varsigma(x, y, z) = 0$ and the surface grid-line distributions of $\xi = \text{const}$ and $\eta = \text{const}$ are user-specified. The outer boundary $\varsigma(x, y, z) = \varsigma_{max}$ is not specified; it is only required to be sufficiently far removed from the inner boundary. Using ς as the marching direction, partial differential equations are sought that produce surfaces of constant ξ, η, and ς to form a nonsingular mesh system

An extension of the cell-volume procedure to three dimensions is proposed. In three dimensions, however, there are three orthogonality relations and one cell-volume constraint. At any point four equations are available to predict the three unknowns x, y, and z, so one equation must be discarded. Because ς is the marching direction, it is natural to use only the two orthogonality relations that involve ς, this leads to the governing equations

$$x_\xi x_\varsigma + y_\xi y_\varsigma + z_\xi z_\varsigma = 0 \tag{3a}$$

$$x_\eta x_\varsigma + y_\eta y_\varsigma + z_\eta z_\varsigma = 0 \tag{3b}$$

$$x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi = \Delta V \tag{3c}$$

or, with $\vec{r}$ defined as $(x, y, z)^t$

$$\vec{r}_\xi \cdot \vec{r}_\zeta = 0 \tag{4a}$$

$$\vec{r}_\eta \cdot \vec{r}_\zeta = 0 \tag{4b}$$

$$\left| \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right| = J^{-1} = \Delta V \tag{4c}$$

The first two equations represent orthogonality relations between ξ and ζ and between η and ζ, and the last equation is the volume or finite Jacobian constraint.

Equations (3) comprise a system of nonlinear partial differential equations in which x, y, and z are specified as initial data at $\zeta = 0$. As developed below, linearization and analysis of Eqs. (3) about a nearby known state reveals that the system is hyperbolic with ζ taken as the marching direction.

Let x^0, y^0, z^0 represent a nearby known state so that

$$x = x^0 - (x - x^0) = x^0 - \tilde{x}$$
$$y = y^0 - \tilde{y}$$
$$z = z^0 + \tilde{z}$$

where $\tilde{x}$, $\tilde{y}$, and $\tilde{z}$ are small. Substitution of these expressions into Eqs. (3) and elimination of products of tilde terms results in the locally linearized system

$$A_0(\vec{r} - \vec{r}_0)_\xi + B_0(\vec{r} - \vec{r}_0)_\eta + C_0(\vec{r} - \vec{r}_0)_\zeta = \vec{f} \tag{5}$$

with

$$A = \begin{pmatrix} x_\zeta & y_\zeta & z_\zeta \\ 0 & 0 & 0 \\ (y_\eta z_\zeta - y_\zeta z_\eta) & (x_\zeta z_\eta - x_\eta z_\zeta) & (x_\eta y_\zeta - x_\zeta y_\eta) \end{pmatrix} \tag{6a}$$

$$B = \begin{pmatrix} 0 & 0 & 0 \\ x_\zeta & y_\zeta & z_\zeta \\ (y_\zeta z_\xi - y_\xi z_\zeta) & (x_\xi z_\zeta - x_\zeta z_\xi) & (x_\zeta y_\xi - x_\xi y_\zeta) \end{pmatrix} \tag{6b}$$

$$C = \begin{pmatrix} x_\xi & y_\xi & z_\xi \\ x_\eta & y_\eta & z_\eta \\ (y_\xi z_\eta - y_\eta z_\xi) & (x_\eta z_\xi - x_\xi z_\eta) & (x_\xi y_\eta - x_\eta y_\xi) \end{pmatrix} \tag{6c}$$

and

$$\vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \vec{f} = \begin{pmatrix} -\left(\frac{\partial \vec{r}}{\partial \xi} \cdot \frac{\partial \vec{r}}{\partial \zeta}\right)_0 \\ -\left(\frac{\partial \vec{r}}{\partial \eta} \cdot \frac{\partial \vec{r}}{\partial \zeta}\right)_0 \\ \Delta V - \Delta V_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \Delta V - \Delta V_0 \end{pmatrix} \tag{6d}$$

Equation (5) is rewritten as

$$A_0\bar{r}_\xi - B_0\bar{r}_\eta - C_0\bar{r}_\varsigma = \bar{e} \tag{7}$$

with $\bar{e} = (0.0, \Delta V - 2\Delta V_0)^t$ Now C_0^{-1} exists unless $(\Delta V_0) \to 0$ r. which we will not impose. so (7) can be rewritten as

$$C_0^{-1}A_0\bar{r}_\xi - C_0^{-1}B_0\bar{r}_\eta - \bar{r}_\varsigma = C_0^{-1}\bar{e} \tag{8}$$

Although the algebraic verification is nontrivial $C_0^{-1}A_0$ and $C_0^{-1}B_0$ are symmetric matrices and are given in the Appendix (The matrices and the verification of symmetry were carried out using MACSYMA by Dennis Jesperson of the Ames Research Center.) Because $C_0^{-1}A_0$ and $C_0^{-1}B_0$ are symmetric, the linearized system Eq. (8) is hyperbolic and can be marched with ς serving as the "time-like" direction.

It can be pointed out that an analysis was attempted for the three orthogonality relations alone. These equations, however, are found to be improperly posed for marching with initial data in ς Indeed, as best as we can discern, these three relations do not lend themselves to unique solutions regardless of the type of boundary conditions specified

SOLUTION PROCEDURE

The nonlinear system of grid generation equations given by Eqs. (3) are solved with a noniterative implicit finite difference scheme. An unconditionally stable implicit scheme is chosen so the marching step size in ς can be arbitrarily selected based only on considerations of accurately generating the grid. Iterative solution of the nonlinear grid generation equations is avoided by expanding the equations about the previous marching step. As a consequence Eq. (7) is solved with the nearby known state 0 taken from the previous ς step.

The following subsections describe the numerical method for marching in ς and describe the procedures for chosing the mesh-cell increments. Special differencing must also be incorporated at axis singularities and at ξ and η boundaries. This detail is also described.

Numerical Method

Let $\Delta\xi = \Delta\eta = \Delta\varsigma = 1$ such that $\xi = j - 1$, $\eta = k - 1$, and $\varsigma = l - 1$. Central spatial differencing of Eqs. (5) in ξ and η with two-point backward implicit differencing in ς leads to

$$A_l\delta_\xi(\bar{r}_{l+1} - \bar{r}_l) + B_l\delta_\eta(\bar{r}_{l+1} - \bar{r}_l) + C_l\nabla_\varsigma\bar{r}_{l+1} = \bar{g}_{l+1} \tag{9}$$

where

$$\bar{g}_{l+1} = \begin{pmatrix} 0 \\ 0 \\ \Delta V_{l+1} \end{pmatrix}$$

and

$$\delta_\xi \vec{r}_j = \frac{\vec{r}_{j+1} - \vec{r}_{j-1}}{2}, \quad \delta_\eta \vec{r}_k = \frac{\vec{r}_{k-1} - \vec{r}_{k-1}}{2}$$

$$\nabla_\cdot \vec{r}_{l-1} = \vec{r}_{l-1} - \vec{r}_l$$

Note that $A_l \delta_\xi \vec{r}_l$ and $B_l \delta_\eta \vec{r}_l$ were subtracted from the left-hand side of (7) to put the equations in a 'delta' form. Throughout only those indices that change are indicated, thus $r_{l-1} \Rightarrow r_{j,k\,l-1}$ and $r_{j-1} \Rightarrow r_{j-1\,k\,l}$, etc.

Multiplying through by C_l^{-1} gives

$$C_l^{-1} A_l \delta_\xi (\vec{r}_{l+1} - \vec{r}_l) + C_l^{-1} B_l \delta_\eta (\vec{r}_{l-1} - \vec{r}_l) - I(\vec{r}_{l+1} - \vec{r}_l) = C_l^{-1} \vec{g}_{l+1} \tag{10}$$

where I is the identity matrix. To reduce the inversion cost the difference equations are approximately factored as

$$(I - C_l^{-1} B_l \delta_\eta)(I - C_l^{-1} A_l \delta_\xi)(\vec{r}_{l-1} - \vec{r}_l) = C_l^{-1} \vec{g}_{l+1} \tag{11}$$

so that $\vec{r}_{l+1}$ is obtained by solving sequences of one-dimensional-like block tridiagonal systems

$$(I - C_l^{-1} B_l \delta_\eta) \vec{g}^{\,*}_{l+1} = C_l^{-1} \vec{g}_{l+1} \tag{12a}$$

$$(I - C_l^{-1} A_l \delta_\xi) \nabla_\cdot \vec{r}_{l+1} = \vec{g}^{\,*}_{l+1} \tag{12b}$$

$$\vec{r}_{l-1} = \vec{r}_l - \nabla_\cdot \vec{r}_{l-1} \tag{12c}$$

In practice, numerical dissipation terms are added in the ξ- and η- directions. Typically, we have used a combination of fourth and second differences which are explicitly and implicitly included in the basic algorithm as

$$\begin{aligned} [I + C_l^{-1} B_l \delta_\eta - \epsilon_{i\eta} (\Delta\nabla)_\eta][I + C_l^{-1} A_l \delta_\xi - \epsilon_{i\xi} (\Delta\nabla)_\xi](\vec{r}_{l+1} - \vec{r}_l) = \\ C_l^{-1} \vec{g}_{l+1} - [\epsilon_{e\xi} (\Delta\nabla)_\xi^2 + \epsilon_{e\eta} (\Delta\nabla)_\eta^2] \vec{r}_l \end{aligned} \tag{13}$$

where, for example,

$$(\Delta\nabla)_\eta \vec{r} = \vec{r}_{k+1} - 2\vec{r}_k + \vec{r}_{k-1}$$

and

$$(\Delta\nabla)_\xi^2 \vec{r} = \vec{r}_{j+2} - 4\vec{r}_{j-1} + 6\vec{r}_j - 4\vec{r}_{j-1} + \vec{r}_{j-2}$$

The dissipation coefficients are scaled to the local mesh using $\epsilon_{e\xi} = 0.5\|C^{-1}A\|$, $\epsilon_{e\eta} = 0.5\|C^{-1}B\|$ and ϵ_i is $3 \times \epsilon_e$. For simplicity, the matrix norms $\|C^{-1}A\|$ and $\|C^{-1}B\|$ have been approximated by a pseudonorm consisting of the square root of the sum of the squared elements. As an alternative to fourth differences, numerical dissipation terms such as

$$\epsilon |\Delta\nabla \vec{r}_l| \Delta\nabla \vec{r}_l$$

have been tried, but these are less effective than the fourth difference terms

Following Kinsey and Barth [9], additional smoothing and implicitness are put into the algorithm by differencing $\nabla \cdot \bar{r} = F$ as $r_{l-1} - r_l = (1-\theta)F_{l-1} - \theta F_l$, with $\theta \geq 0$. Ignoring the adjustments this entails in the user specified volume increments, which need not be exact, this differencing is ζ is incorporated Eq (13) as

$$I - (1-\theta)C_l^{-1}B_l\delta_\eta - \epsilon_{i,\eta}(\Delta\nabla)_\eta\, I - (1-\theta)C_l^{-1}A_l\delta_\xi - \epsilon_{i,\xi}(\Delta\nabla)_\xi](\bar{r}_{l-1} - \bar{r}_l) = C_l^{-1}\bar{g}_{l-1} - [\epsilon_{e\xi}(\Delta\nabla)_\xi^2 - \epsilon_{e\eta}(\Delta\nabla)_\eta^2]\bar{r}_l \tag{14}$$

Values of θ of 1 to 5 are effective in preventing numerical breakdown in the case of highly concave body shapes. If some loss of grid orthogonality can be tolerated, θ can be set to 3 for most applications

The coefficient matrices A_l, B_l and C_l contain ξ- and η-derivatives which are formed using central differences. These matrices also contain derivatives for x_ζ, y_ζ, and z_ζ which are obtained from Eqs (3) in terms of ξ- and η-derivatives. That is, Eqs. (3) are linear in the unknowns x_ζ, y_ζ, and z_ζ. They are easily solved as

$$\begin{pmatrix} x_\zeta \\ y_\zeta \\ z_\zeta \end{pmatrix} = \frac{\Delta V}{(DetC)} \begin{pmatrix} y_\xi z_\eta - y_\eta z_\xi \\ x_\eta z_\xi - x_\xi z_\eta \\ x_\xi y_\eta - x_\eta y_\xi \end{pmatrix} = C^{-1}\bar{g}$$

with

$$Det(C) = (y_\xi z_\eta - y_\eta z_\xi)^2 + (x_\eta z_\xi - x_\xi z_\eta)^2 + (x_\xi y_\eta - x_\eta y_\xi)^2$$

Note that $(\Delta V)^2/(x_\zeta^2 + y_\zeta^2 + z_\zeta^2) = \mathrm{Det}(C)$ so that $\mathrm{Det}(C)$ will be zero if and only if the user specified $\Delta V = 0$. Hence, C^{-1} will exist as noted earlier. The $\mathrm{Det}(C)$ is the square of a mesh area along a $\zeta = const$ plane.

Cell Volume Specification

The user has control of the grid by means of the initial surface point distribution and by specification of the cell volumes, $\Delta V_{j,k,l}$. Through the cell volumes the extent and clustering of the grid can be essentially controlled. Because the cell volume at each point must be given, it is clear that the user must devise some simple kind of method for specifying volumes There are many possibilities, and two approaches that have been used are illustrated here

One way of generating mesh cell volumes is to form a grid about a 'similiar' but simple reference body for which the grid can be generated analytically, and to use the cell volumes from this reference grid for the more complex problem. For example, suppose we wish to grid an aircraft fuselage as a warped spherical-like grid following the topology shown in Fig.1 To specify cell volumes we first find a sphere that has the same surface area as our fuselage and analytically build a grid.

For simplicity, choose a spherical grid that has uniform angle spacing and a radial distribution that varies exponentially In this special case, the control cell volumes are analytically known The grid cell volumes of this spherical reference grid are then used to specify the cell volumes of the fuselage grid. However, the fuselage will not have the same kind of surface area distribution as a sphere with equal angle distribution So here we use an adjustment of the form

$$\Delta V_{j,k,l} = \left[(1-\nu) - \frac{(\Delta A_{j,k})_{fuselage}}{(\Delta A_{j,k})_{sphere}}\,\nu\right](\Delta V_{j,k,l})_{sphere} \tag{15}$$

where $\nu \to 1$ for small l and $\nu \to 0$ for large l That is, the volumes are adjusted initially to the local boundary surface increments But as the solution is marched out, the uniform spherical volumes gradually become specified. Such an approach has been used, and, as a result, the far field portion of the grid tends to be uniformly spherical. Some of the results shown later will illustrate this behavior.

Another method of specifying the volumes that has been used depends on the grid being generated and for this reason this second method is sometimes less robust. In this method, the specified volume at each point is set equal to the computed surface area element times a user specified arc length Specifically,

$$\Delta V_{j,k,l+1} = \Delta s_{j,k,l} \Delta S_{j,k,l} \tag{16}$$

where Δs is the user specified arc length and ΔS is the surface area. In all of our applications Δs has been given as an exponential stretching. In this kind of volume control specification, if an initial distribution of points is highly clustered in ξ and η, then these points tend to remain highly clustered in ξ and η even far away from the body. To obtain a more uniform far-field distribution, the volumes specified from Eq. (16) have been averaged in ξ and η with each step taken in ς. For example

$$\Delta V_{j,k,l+1} = (\Delta V_{j+1,k,l+1} + \Delta V_{j-1,k,l+1} + \Delta V_{j,k-1,l+1} + \Delta V_{j,k-1,l+1} + 4\Delta V_{j,k,l+1})/8 \tag{17}$$

has been applied one or more times with each step in ς. Alternatively, in the far-field the volumes given by Eq. (16) can be weighted with the uniform spherical volumes.

Axis Treatment

A coordinate singularity will be encountered whenever a whole grid face is mapped onto a closed body Here we will generate warped spherical grids, so Eqs. (3) become singular at the axis. With ς the marching direction and using ξ and η as sketched in Fig.1 (i.e. ξ from pole to pole and η equatorial), then the axis at $\xi = 0$ and the axis at $\xi = \xi_{max}$ represent singularities. In particular, η-derivatives approach zero at the pole and Eqs. (3b) and (3c) are lost. There

are $kmax$ points in the η-direction, but, at a given ζ-station, all x,y,z values are the same. Consequently, the difference equation corresponding to Eq. (3a) can be imposed $kmax$ times to solve for only three unknowns, namely x, y, and z on the axis. Thus, the axis points are overdetermined and can be solved for via a generalized inverse scheme However, such a constraint is difficult to implement implicitly into the solution algorithm (12)

In the approach actually used at the axis, the body is aligned so that y changes along the axis more rapidly than x or z We assume that the direction of the axis leaving the body surface $\zeta = 0$ can be predicted from the surface distribution of points near the axis. For example, for the $\xi = \xi_{max}$ axis as sketched in Fig.2, averaging points in η at $\xi = \xi_{max} - \Delta\xi$ and/or $\xi_{max} - 2\Delta\xi$ allows us to find an axis point inside the body from which the direction of the axis can be determined. Knowing the direction of the axis, changes of x and z along the axis can be expressed in terms of changes in y as

$$x_\zeta = \frac{(\nabla x)}{(\nabla y)} y_\zeta \tag{18a}$$

$$z_\zeta = \frac{(\nabla z)}{(\nabla y)} y_\zeta \tag{18b}$$

where $\frac{(\nabla x)}{(\nabla y)}$ and $\frac{(\nabla z)}{(\nabla y)}$ are the known axis slopes and ∇ implies backward differencing in the ζ-direction. These relations along with the orthogonality relation provide enough information to back out the variables on the axis. Eliminating x_ζ and z_ζ from the orthogonality relation Eq. (3a) by using Eq. (18) gives

$$\left[\frac{(\nabla x)}{(\nabla y)}\delta_\xi x + \delta_\xi y + \frac{(\nabla z)}{(\nabla y)}\delta_\xi z\right]\delta_\zeta y = 0$$

or since $\delta_\zeta y \neq 0$

$$\frac{(\nabla x)}{(\nabla y)}\delta_\xi x + \delta_\xi y + \frac{(\nabla z)}{(\nabla y)}\delta_\xi z = 0 \tag{19}$$

Equations (18) and (19) can be readily built into the implicit solution process by modifying the end points of the ξ-block tridiagonal matrix formed by Eq. (12b). Let the axis points be denoted as $j = 1$ and $j = jmax$ and define $M_l = (C_l^{-1}A_l)/2$. Then Eq. (12b) can be represented for a given index k from $j = 1$ to $j = jmax$ as

$$\begin{bmatrix} F_1 & G_1 & 0 & 0 & & 0 \\ -M_2 & I & M_2 & 0 & & 0 \\ 0 & -M_3 & I & M_3 & & 0 \\ & & & -M_{jmax-1} & I & M_{jmax-1} \\ 0 & & & 0 & E_{jmax} & F_{jmax} \end{bmatrix} \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \\ \\ \vec{g}_{jmax} \end{pmatrix} = \begin{pmatrix} \vec{g}_1 \\ \vec{g}_2 \\ \\ \\ \vec{g}_{jmax} \end{pmatrix} \tag{20}$$

where the contribution from the dissipation terms is not being shown The endpoint blocks F_1. G_1. E_{jmax} and F_{jmax} are supplied from the axis conditon, Eqs. (18) and (19), put into delta form as

$$(x_{j,k,l+1} - x_{j,k,l}) - \frac{\nabla x}{\nabla y}(y_{j,k\,l-1} - y_{j,k\,l}) = 0$$

$$\frac{\nabla x}{\nabla y}\delta_\xi(x_{j\,k\,l+1} - x_{j\,k,l}) - \delta_\xi(y_{j,k\,l-1} - y_{j\,k\,l}) - \frac{\nabla z}{\nabla y}\delta_\xi(z_{j,k,l-1} - z_{j,k,l}) = 0$$

$$-\frac{\nabla z}{\nabla y}(y_{j\,k,l-1} - y_{j,k,l}) + (z_{j\,k\,l-1} - z_{j,k,l}) = 0$$

Approximating δ_ξ with Δ_ξ at $j = 1$. the blocks F_1 and G_1 become

$$F_1 = \begin{bmatrix} 1 & -\frac{\nabla x}{\nabla y} & 0 \\ -\frac{\nabla x}{\nabla y} & -1 & -\frac{\nabla z}{\nabla y} \\ 0 & -\frac{\nabla z}{\nabla y} & 1 \end{bmatrix} \tag{21a}$$

$$G_1 = \begin{bmatrix} 0 & 0 & 0 \\ \frac{\nabla x}{\nabla y} & 1 & \frac{\nabla z}{\nabla y} \\ 0 & 0 & 0 \end{bmatrix} \tag{21b}$$

with $\vec{g}_1 = \vec{0}$. At a given $\zeta = const$ step, the inversion of a ξ-block tridiagonal results in slightly different predicted values of x, y, and z around the axis in η Because these values should all be identical, their averaged value is used as the axis point. New axis slopes $\frac{(\nabla x)}{(\nabla y)}$ and $\frac{(\nabla z)}{(\nabla y)}$ are formed as each $\zeta = const$ grid surface is generated, and in this way the axis is allowed to curve.

As a modification to what is described, second-order three-point differencing in ξ is used (slightly weighted with the first-order differencing indicated above). Because three-point differencing is used at the ends, a modified block-tridiagonal

system that allows for additional blocks at locations (1,3) and $(jmax, jmax-2)$ is solved

Boundary Surfaces

If the grid is not being generated about a completely closed body surface as shown in Fig 1, then boundary conditions other than periodicity or axis treatment must be imposed For example as sketched in Fig 3 it may be necessary to generate a grid about a half-span wing attached to fuselage or a plane of symmetry at, say, $\xi = 0$ In this situation, the grid at $\xi = 0$ (see Fig 3) cannot be prescribed as this would be incompatible with the grid developing in the interior when marched in ζ Consequently, the grid distribution on this boundary surface must evolve as part of the overall solution process There are multiple ways of generating the boundary grid along with the interior, one of which is to use one-sided differencing tied to the eigensystem of the $C^{-1}A$ and $C^{-1}B$ coefficient matrices.

The eigenvalues of $C^{-1}A$ and $C^{-1}B$ have not been computed analytically, but, because of the special form of each coefficient matrix, one eigenvalue must be zero and the remaining two eigenvalues are real and of equal and opposite sign (see the Appendix). As a consequence, at least two proper combinations of the grid generation equations can be differenced to one side on a ξ or η boundary surface. Specification of the surface provides the third necessary condition to determine the grid points. At a $\xi = 0$ boundary, for example, the characteristic combination of equations corresponding to the zero and the negative eigenvalues of $C^{-1}A$ can be one- sided differenced. However, at this point we do not have analytical formula for the eigenvalues and eigenvectors of $C^{-1}A$ and $C^{-1}B$, and have, therefore, implemented another less rigorous approach.

In lieu of working with the eigensystem, we have specified the surface and have simply discarded a governing equation. For example, the $\xi = 0$ surface shown in Fig. 3 can be represented as

$$f(x, y, z) = 0 \tag{22}$$

As long as this surface is close to a $y = const$ plane, Eq. (22) can be used in place of the ξ-ζ orthogonality condition. The remaining two conditions, restricted to a $y = const$ plane, are essentially the two-dimensional governing equations, Eq. (2), so this approach would seem to a good one provided we can ensure compatibility with the interior mesh. To ensure this compatibility, we have built Eq. (22) into the solution algorithm, Eq. (11), as follows. Equation (22) is first differentiated with respect to ζ as

$$f_\zeta = f_x x_\zeta - f_y y_\zeta + f_z z_\zeta = 0$$

and then approximating x_ζ as $x_{l+1} - x_l$, etc gives

$$f_x(x_{l+1} - x_l) + f_y(y_{l+1} - y_l) + f_z(z_{l+1} - z_l) = 0 \tag{23}$$

Use of this relation in place of Eq (3a) leaves the coefficient matrix B_l and the vector $\bar{g}_l$ of Eq. (9) unaltered, the first row of A_l is replaced with zero elements, and the first row of C_l is replaced by the elements (f_x, f_y, f_z) In the special case of a planar boundary, say $y = 0$, it can be verified that $C_l^{-1} A_l$ has only zero eigenvalues. Therefore, δ_ξ can be backward-differenced at $\xi = 0$ without instability in this special case Backward differencing of δ_ξ at $\xi = 0$ for a more general surface $f(x, y, z) = 0$ with moderate curvature has also been stable

RESULTS

To demonstrate the capabilities of the current hyperbolic grid solver, grids were generated about two wings and about a simple wing-body configuration In all cases a spherical mesh topology was employed so an axis singularity exists.

As a first illustration, a spherical-like grid was generated about a rectangular wing with a NACA 0012 airfoil section. The planform, the thickness distribution, and the airfoil section are shown in Fig. 4a where a nearly rectangular wing planform is defined from the superellipse

$$(x/x_{max})^8 + (y/y_{max})^8 = 1$$

and the thickness distribution is elliptical

$$(y/y_{max})^2 + (z/z_{max})^2 = 1$$

Also shown is the user specified surface grid-point distribution which was arranged so that an axis extends from the back of the wing tips The surface grid has 79 points distributed in the ξ-direction from pole to pole and 120 points in the periodic η-direction. The $\xi = const$ surface lines were generated by computing the intersection of planes originating from a point located on the symmetry plane with the wing surface. The $\eta = const$ surface lines were clustered as a cosine distribution.

The grid was marched out in ζ using 40 steps to sweep out a distance of approximately 8 chords with a uniform grid spacing of 1/2 percent maximum chord specified as the first grid spacing off the body An entire warped spherical grid was generated without a plane of symmetry being used, but Figs.4b to 4e show only various segments of the generated grid for ξ and $\eta = const$ planes. Portions of two different $\eta = const$ planes in the $z = 0$ plane and in the vicinity of the axis are shown in Fig. 4b to indicate that the grid near the axis is relatively well behaved. The grid at the wing midspan ($y = 0$) is shown in Fig. 4c. This is a $\xi = const$ plane. As seen, the grid is very smooth and uniform off the body and nearly orthogonal everywhere The tendency for the grids lines in the far-field of the grid to be circular reflects the fact that spherical mesh incremental volumes have been specified Finally, Figs. 4d and 4e show close-up views of this same grid

plane and illustrate that the grid generated near the profile nose and trailing edge regions is quite satisfactory

A similar grid was computed as illustrated by Figs 5a to 5d in which the grid spacing off the body is much finer and corresponds to the type of mesh spacing used in viscous flow simulation The wing in this case has a cambered airfoil section with the profile defined as

$$z = 4z_{max}(\bar{x} - x)$$

and a parabolic camber given by

$$z = z + 4cx(1 - x)$$

is sheared into the profile. Figures 5a to 5c show the plane of symmetry grid plane, while Fig. 5d shows a $\xi = const$ plane in the vicinity of the the left wing tip.

It must be remarked that the above wing calculation contained an airfoil with a sharp trailing edge. but unless the the airfoil trailing edge radius is rounded, the method can frequently break down. Provided the trailing edge radius is not zero. breakdown can be avoided by clustering a sufficient number of grid points in this region and adjusting the specified volumes to be compatible.

Grids have also been generated for low aspect ratio wing-body combinations. and Fig. 6 shows portions of a mesh generated for a simplified space shuttle configuration which is missing the OMS pods. engine bells. and vertical fin. Figure 6a shows segments of the grid in the windward and leeward planes of symmetry ($\eta = const$ planes). A blowup of the nose region, showing the axis and canopy region. is provided by Fig. 6b. (The back of the shuttle was terminated in this case at a $x = const$ plane.) Three cross sectional views (projected onto $x = const$ planes) are provided by Figs. 6c-6d. To prevent grid line crossing in the corner of the upper wing body juncture region, it was necessary to run the grid solver with a value of $\theta = 3$ in Eq. (14). Flow calculations using these grids were recently carried out by Rizk and Ben-Shmuel [10].

CONCLUDING REMARKS

A procedure has been developed for generating body fitted coordinates in three dimensions using hyperbolic partial differential equations. In this paper the scheme has been used to generate warped spherical-like grids about simple wing and wing-fuslage configurations. The hyperbolic grid generator can fail whenever the body surface is discontinuous or when the user specified surface grid distribution is too irregular. For simple continous body shapes, however, the hyperbolic grid generation scheme can be very fast and reliable. It requires a minimum of user interaction, and it can be used to generate grids suitable for either inviscid or viscous flow simulation.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the analysis and helpful suggestions of Dennis Jespersen of the Ames Research Center Mr Jack Strigberger of Stanford University contributed to the formulation of the axis logic. and Dr W. Hankey at AFWAL encouraged and initially supported this effort

APPENDIX

Properties of the matrices $C^{-1}A$ and $C^{-1}B$ are needed to determine the hyperbolicity of the grid generation equations. These matrices can be written as

$$\frac{C^{-1}A}{(JDet|C|)} =$$

$$\begin{bmatrix} (y_\eta \zeta_z - z_\eta \zeta_y)x_\xi + \zeta_x \xi_x J^{-1} & (y_\eta \zeta_z - z_\eta \zeta_y)y_\zeta - \zeta_x \xi_y J^{-1} & (y_\eta \zeta_z - z_\eta \zeta_y)z_\zeta + \zeta_x \xi_z J^{-1} \\ (z_\eta \zeta_x - x_\eta \zeta_z)x_\xi + \zeta_y \xi_x J^{-1} & (z_\eta \zeta_x - x_\eta \zeta_z)y_\zeta - \zeta_y \xi_y J^{-1} & (z_\eta \zeta_x - x_\eta \zeta_z)z_\zeta + \zeta_y \xi_z J^{-1} \\ (x_\eta \zeta_y - y_\eta \zeta_x)x_\xi - \zeta_z \xi_x J^{-1} & (x_\eta \zeta_y - y_\eta \zeta_x)y_\zeta + \zeta_z \xi_y J^{-1} & (x_\eta \zeta_y - y_\eta \zeta_x)z_\zeta - \zeta_z \xi_z J^{-1} \end{bmatrix}$$

and

$$\frac{C^{-1}B}{(JDet|C|)} =$$

$$\begin{bmatrix} (z_\xi \zeta_y - y_\xi \zeta_z)x_\zeta + \eta_x \zeta_x J^{-1} & (z_\xi \zeta_y - y_\xi \zeta_z)y_\zeta + \eta_y \zeta_x J^{-1} & (z_\xi \zeta_y - y_\xi \zeta_z)z_\zeta + \eta_z \zeta_x J^{-1} \\ (x_\xi \zeta_z - z_\xi \zeta_x)x_\zeta + \eta_x \zeta_y J^{-1} & (x_\xi \zeta_z - z_\xi \zeta_x)y_\zeta + \eta_y \zeta_y J^{-1} & (x_\xi \zeta_z - z_\xi \zeta_x)z_\zeta + \eta_z \zeta_y J^{-1} \\ (y_\xi \zeta_x - x_\xi \zeta_y)x_\zeta + \eta_x \zeta_z J^{-1} & (y_\xi \zeta_x - x_\xi \zeta_y)y_\zeta + \eta_y \zeta_z J^{-1} & (y_\xi \zeta_x - x_\xi \zeta_y)z_\zeta + \eta_z \zeta_z J^{-1} \end{bmatrix}$$

where the Jacobian $J = (\Delta V)^{-1}$ is

$$J^{-1} = x_\xi y_\eta z_\zeta + x_\zeta y_\xi z_\eta + x_\eta y_\zeta z_\xi - x_\xi y_\zeta z_\eta - x_\eta y_\xi z_\zeta - x_\zeta y_\eta z_\xi$$

and the transformation metrics which appear in the elements of A, B, and C are

$$\xi_x = J(y_\eta z_\zeta - y_\zeta z_\eta)$$

$$\xi_y = J(x_\zeta z_\eta - x_\eta z_\zeta)$$

$$\xi_z = J(x_\eta y_\zeta - x_\zeta y_\eta)$$

$$\eta_x = J(y_\zeta z_\xi - y_\xi z_\zeta)$$

$$\eta_y = J(x_\xi z_\zeta - x_\zeta z_\xi)$$

$$\eta_z = J(x_\zeta y_\xi - x_\xi y_\zeta)$$

$$\zeta_x = J(y_\xi z_\eta - y_\eta z_\xi)$$
$$\zeta_y = J(x_\eta z_\xi - x_\xi z_\eta)$$
$$\zeta_z = J(x_\xi y_\eta - x_\eta y_\xi)$$

Using the orthogonality relations and the definition the metrics $\xi_x, \xi_y, \ldots$ in terms of $x_\xi, x_\eta, \ldots$ it can be shown that the matrices $C^{-1}A$ and $C^{-1}B$ are symmetric For example, the (3,1) and (1,3) elements of $C^{-1}A$ are found to be equivalent as shown by subtracting (3,1) from (1,3) to obtain

$$(y_\eta \zeta_z - z_\eta \zeta_y)x_\zeta - \zeta_x \xi_z J^{-1} - (x_\eta \zeta_z - y_\eta \zeta_x)x_\zeta - \zeta_z \xi_x J^{-1} = 0?$$

Using the definitions of $J^{-1}\xi_x$ and $J^{-1}\xi_z$ and eliminating terms which cancel leaves

$$-(z_\eta z_\zeta + x_\eta x_\zeta)\zeta_y - x_\eta y_\zeta \zeta_z - y_\zeta z_\eta \zeta_z = 0?$$

Now using the metric definitions for ζ_x etc

$$-(z_\eta z_\zeta + x_\eta x_\zeta)(x_\eta z_\xi - x_\xi z_\eta) + x_\eta y_\zeta (y_\xi z_\eta - y_\eta z_\xi) + y_\zeta z_\eta (x_\xi y_\eta - x_\eta y_\xi) = 0?$$

Collecting the $y_\eta y_\zeta$ terms to complete the orthogonality relation $\vec{r}_\eta \cdot \vec{r}_\zeta = 0$ leaves two terms containing y_ξ that sum to zero, that is

$$-(z_\eta z_\zeta + y_\eta y_\zeta - x_\eta x_\zeta)(x_\eta z_\xi - x_\xi z_\eta) + x_\eta y_\zeta y_\xi z_\eta - y_\zeta z_\eta x_\eta y_\xi = 0$$

The remaining elements have also been checked for symmetry. The reader can also quickly verify that the degenerate case $\xi = x$, $\eta = y$, and $\zeta = z$ gives the symmetric matrices

$$C^{-1}A = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and

$$C^{-1}B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

The matrices $C^{-1}A$ and $C^{-1}B$ have other interesting symmetries. In particular, the determinant and the trace of of both matrices are zero. For example, the $DetA$ is clearly zero from inspection of Eq. (6a) and $Det(C^{-1}A) = DetADetC^{-1}$. The trace of $C^{-1}A$ is given as

$$trace(C^{-1}A) = (y_\eta \zeta_z - z_\eta \zeta_y)x_\zeta + \zeta_x \xi_x J^{-1} + (z_\eta \zeta_x - x_\eta \zeta_z)y_\zeta + \zeta_y \xi_x J^{-1} + (x_\eta \zeta_y - y_\eta \zeta_x)z_\zeta + \zeta_z \xi_z J^{-1}$$

or regrouping

$$trace(C^{-1}A) = (y_\eta\zeta_z - z_\eta\zeta_y)x_\xi - (z_\eta\zeta_x - x_\eta\zeta_z)y_\xi + (x_\eta\zeta_y - y_\eta\zeta_x)z_\xi + (\zeta_x\xi_x + \zeta_y\xi_x + \zeta_z\xi_z)J^{-1}$$

However, $(z_\eta y_\xi - y_\eta z_\xi)\zeta_x$ is $-J^{-1}\xi_x\zeta_x$, etc . so the trace is zero

Because 3 × 3 matrix $C^{-1}A$ is symmetric and has zero trace and determinant, one eigenvalue of $C^{-1}A$ must be zero and the other two real distinct eigenvalues must sum to zero Therefore, these nonzero eigenvalues must be of equal and opposite sign The matrix $C^{-1}B$ has similar properties

REFERENCES

1) Chu. W. H. Development of a general finite difference approximation for a general domain J. Comp. Phys Vol 8. (1971). 392-408

2) Godunov. S K and G P Prokopov. The use of moving meshes in gas-dynamical computations USSR Comput Math Phys.. 12. Vol 2 (1972). 182-195.

3) Thompson. J. F F C. Thames and C M. Mastin. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. J Comp. Phys.. Vol 15. (1974). 299-319

4) Thompson. J F . Elliptic grid generation. in 'Numerical Grid Generation'. Joe F. Thompson. ed. North-Holland. New York. (1982).

5) Sorenson, R. L. and J. L. Steger, Grid generation in three dimensions by Poisson equations with control of cell size and skewness at boundary surfaces. in Advances in Grid Generation. ASME FED-Vol. 5,(1983) 181-187.

6) Steger. J. L. and D. S Chaussee. Generation of body-fitted coordinates using hyperbolic partial differential equations, SIAM J. Sci. Stat. Comput.. Vol. 1, (1980). 431-437.

7) Starius, G. Constructing orthogonal curvilinear meshes by solving initial value problems, Numerische Mathematik. Vol. 28, (1977). 25-48.

8) Nakamura. S. Marching grid generation using parabolic partial differential equations. in 'Numerical Grid Generation'. Joe F. Thompson. ed. North-Holland, New York, (1982).

9) Kinsey. D. W. and Barth, T. J. Description of a Hyperbolic Grid Generating Procedure for Arbitrary Two-Dimensional Bodies, AFWAL Technical Memorandum No. 84-191 -FIMM(July. 1984)

10) Rizk. Y. M. and Ben-Shmuel, S. Computation of the Viscous Flow Around the Shuttle Orbiter at Low Supersonic Speeds. AIAA Paper No. 85-0168, 1985.

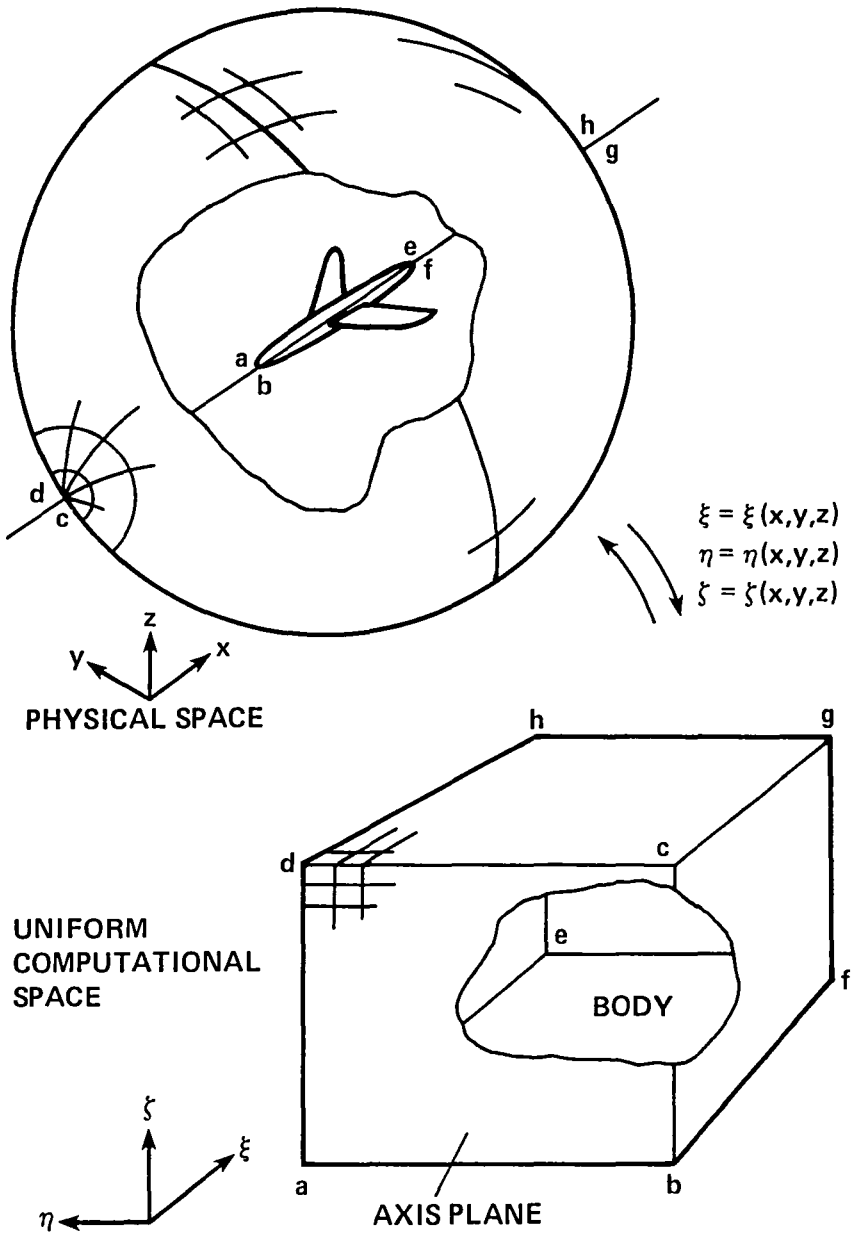


Fig 1 Well-Ordered warped spherical grid mapping

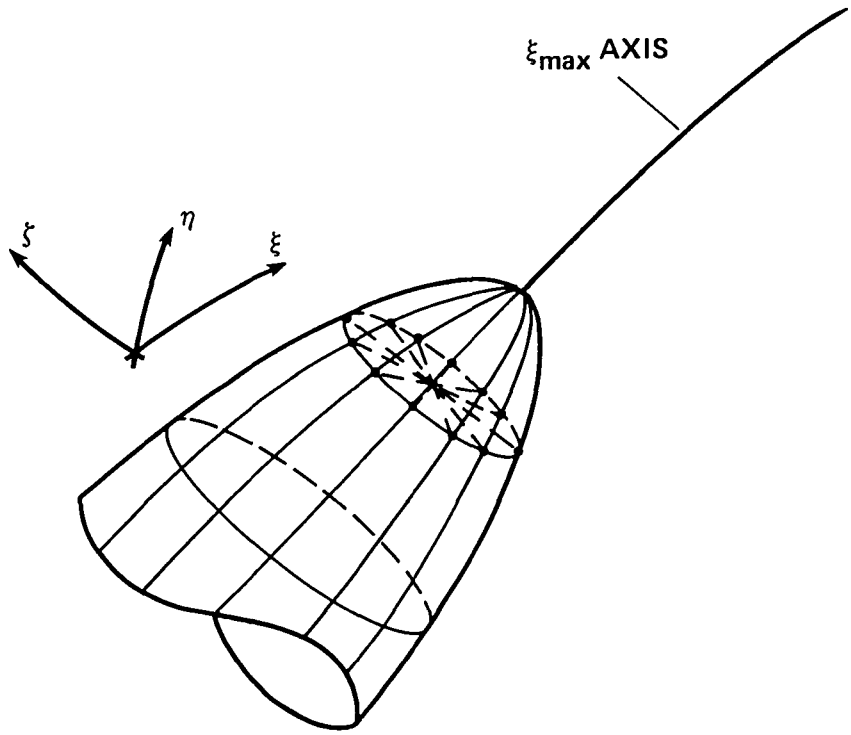


Fig 2 Geometric method for determining the grid axis

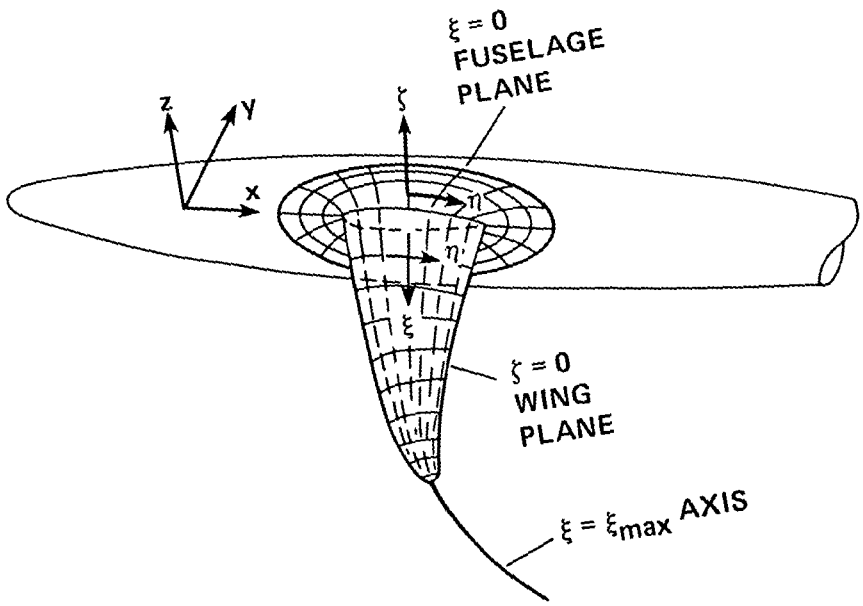
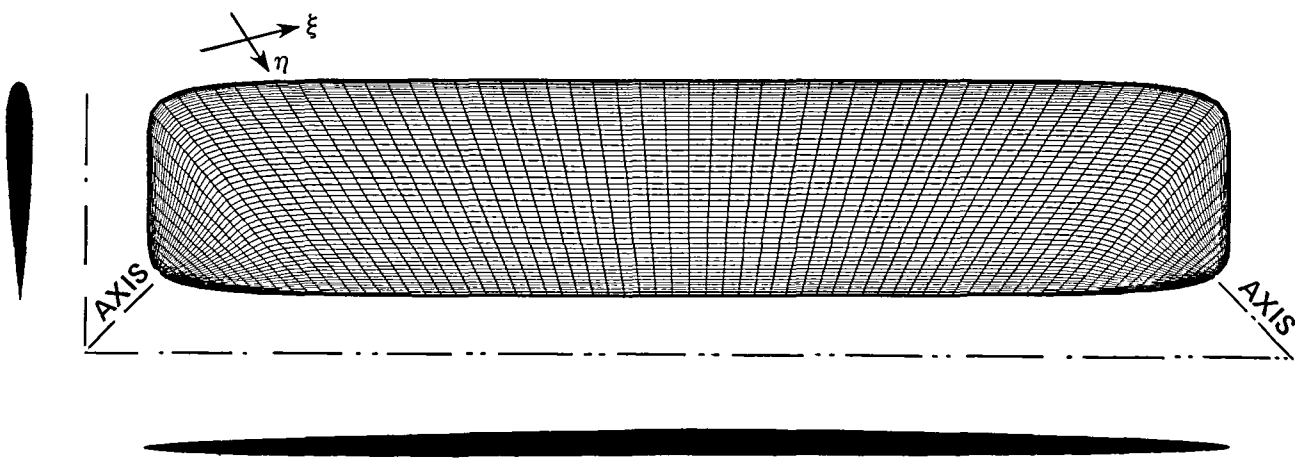
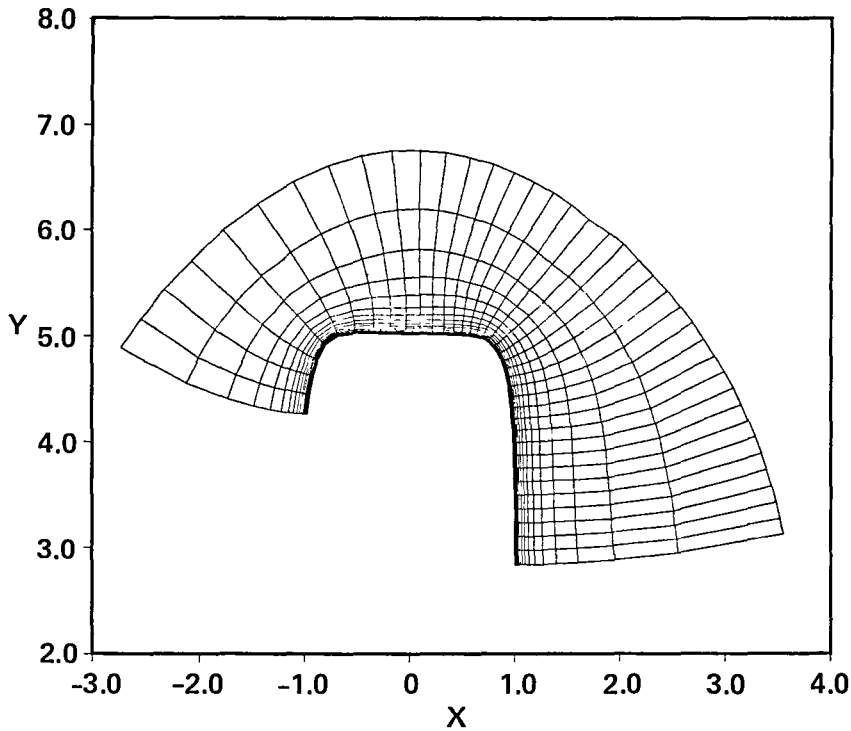


Fig 3 Grid generation along a boundary surface



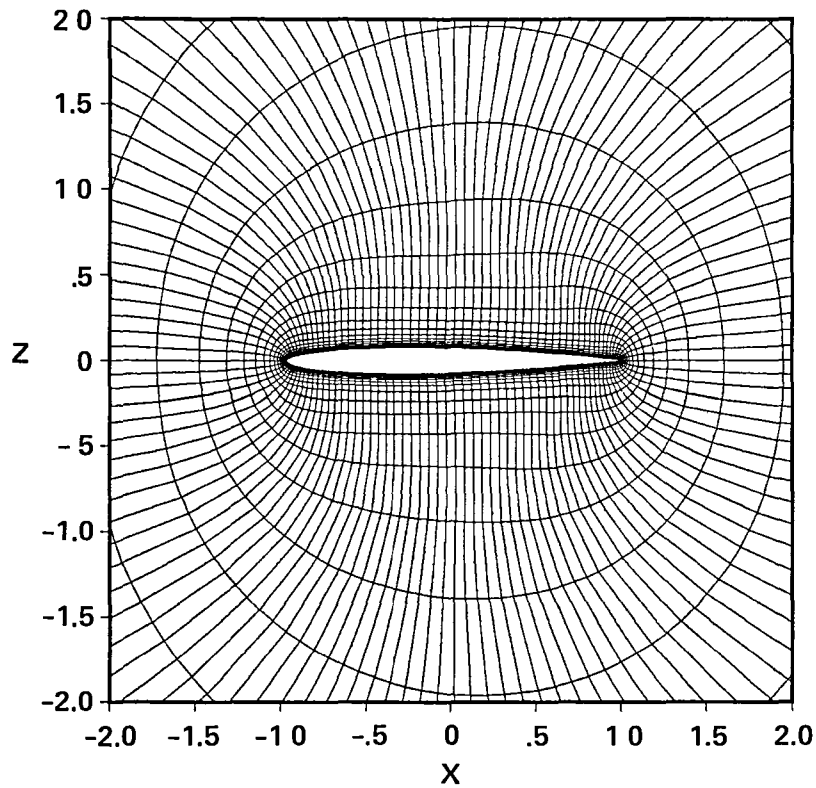
a geometry and surface-point grid distribution on a rectangular wing

Fig 4 Warped spherical grid about a rectangular wing



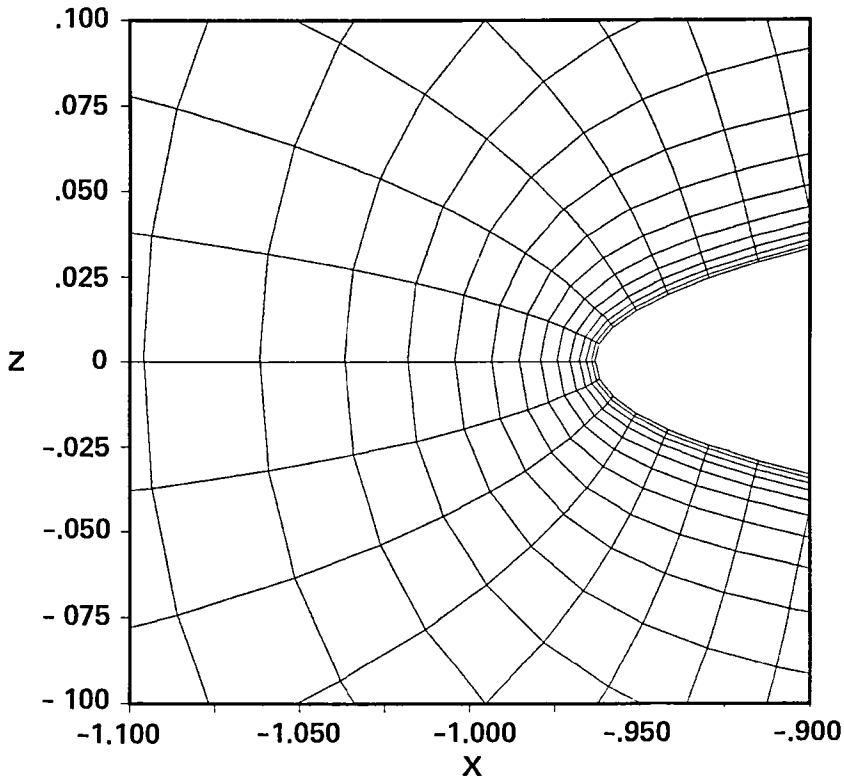
b view of grid segments in the z =0 plane near the wing tip

Continued



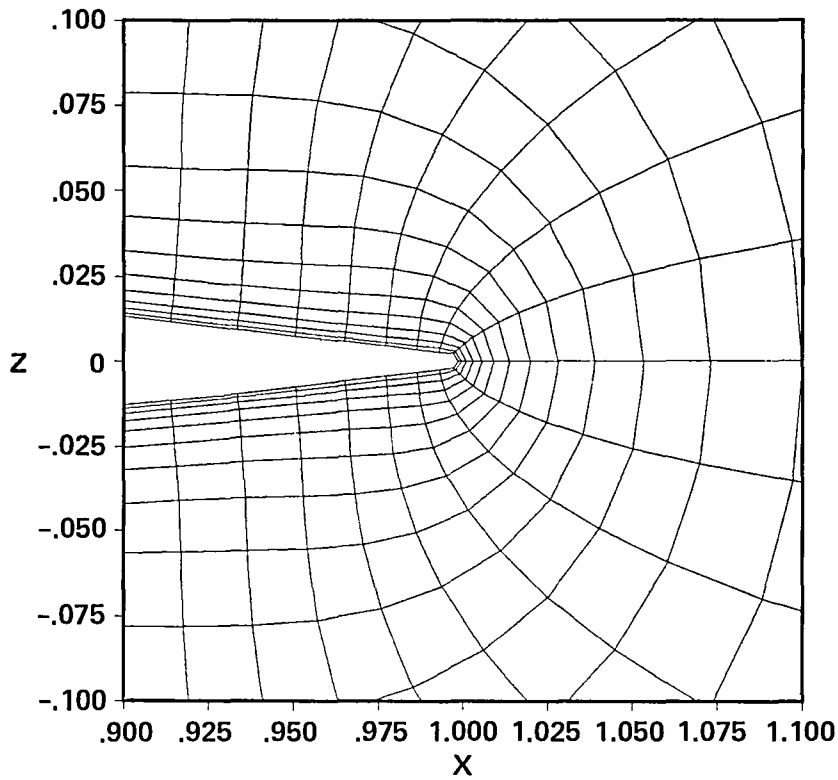
c. view of wing grid at midspan.

Continued



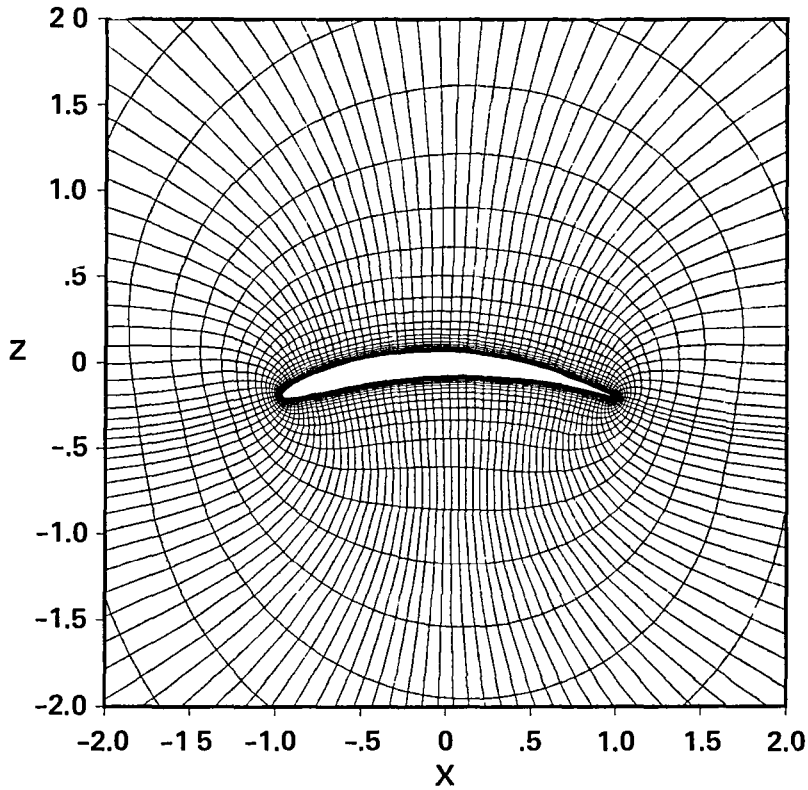
d close-up of the leading edge midspan grid

Continued



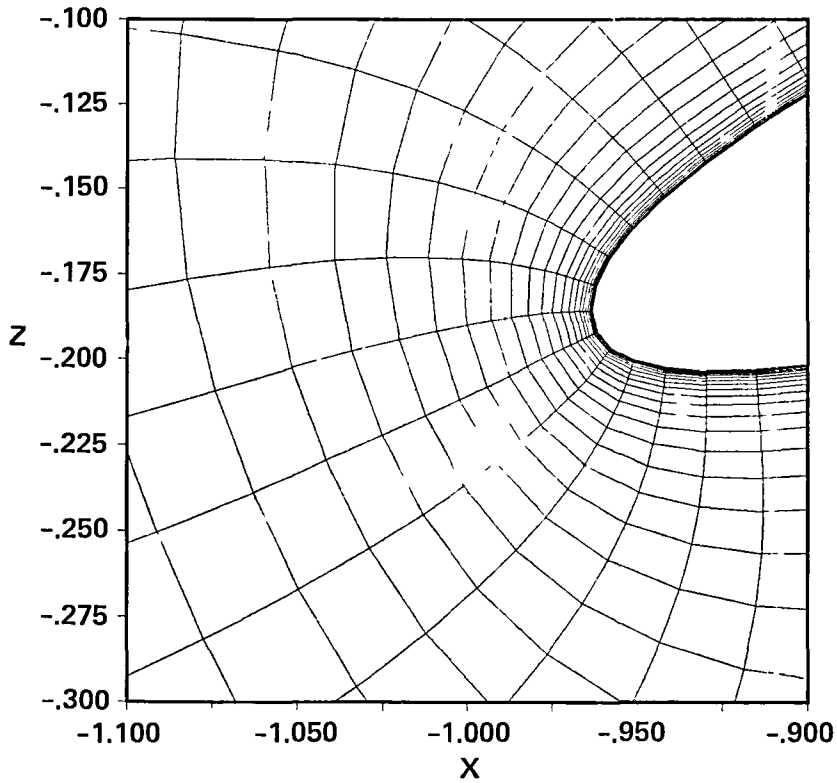
e close-up of the trailing edge midspan grid

Continued



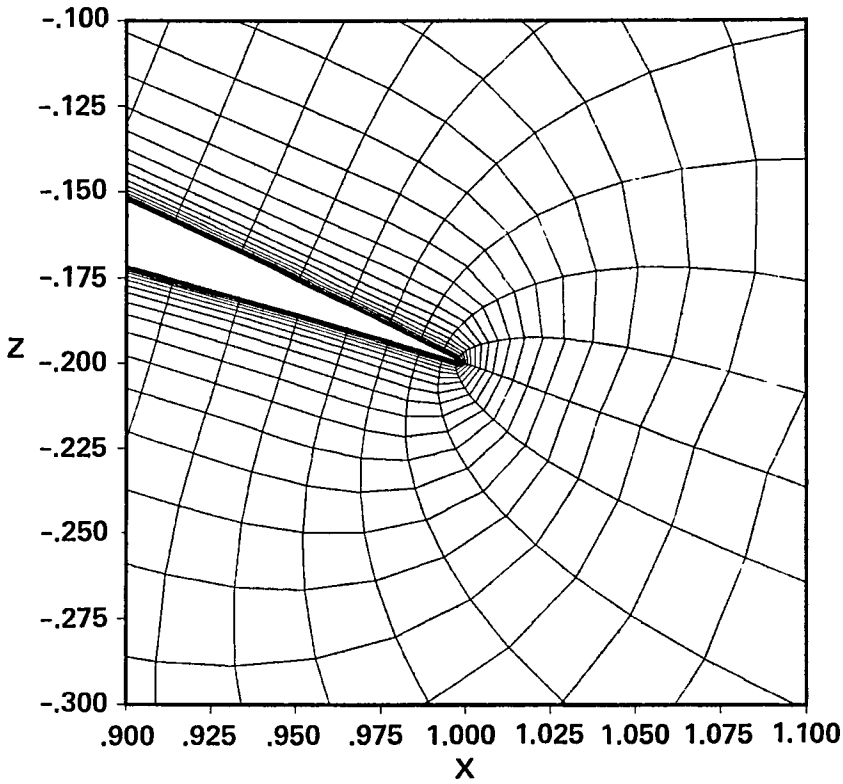
a. view of wing grid at midspan with cambered profile.

Fig. 5 Hyperbolic grid generated about a rectangular wing with camber



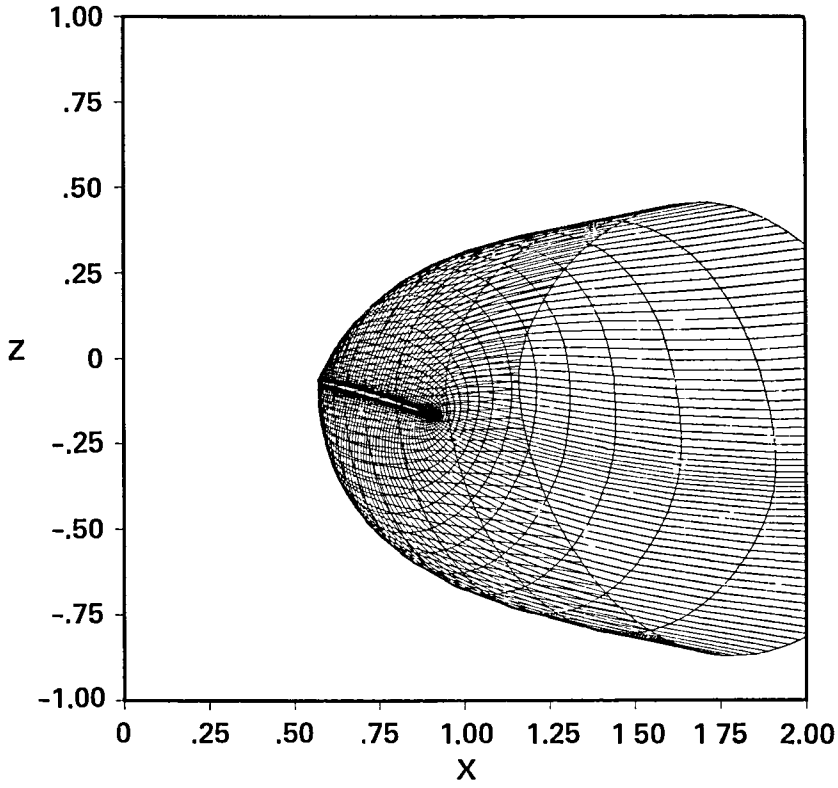
b. close-up of the leading edge midspan grid

Continued



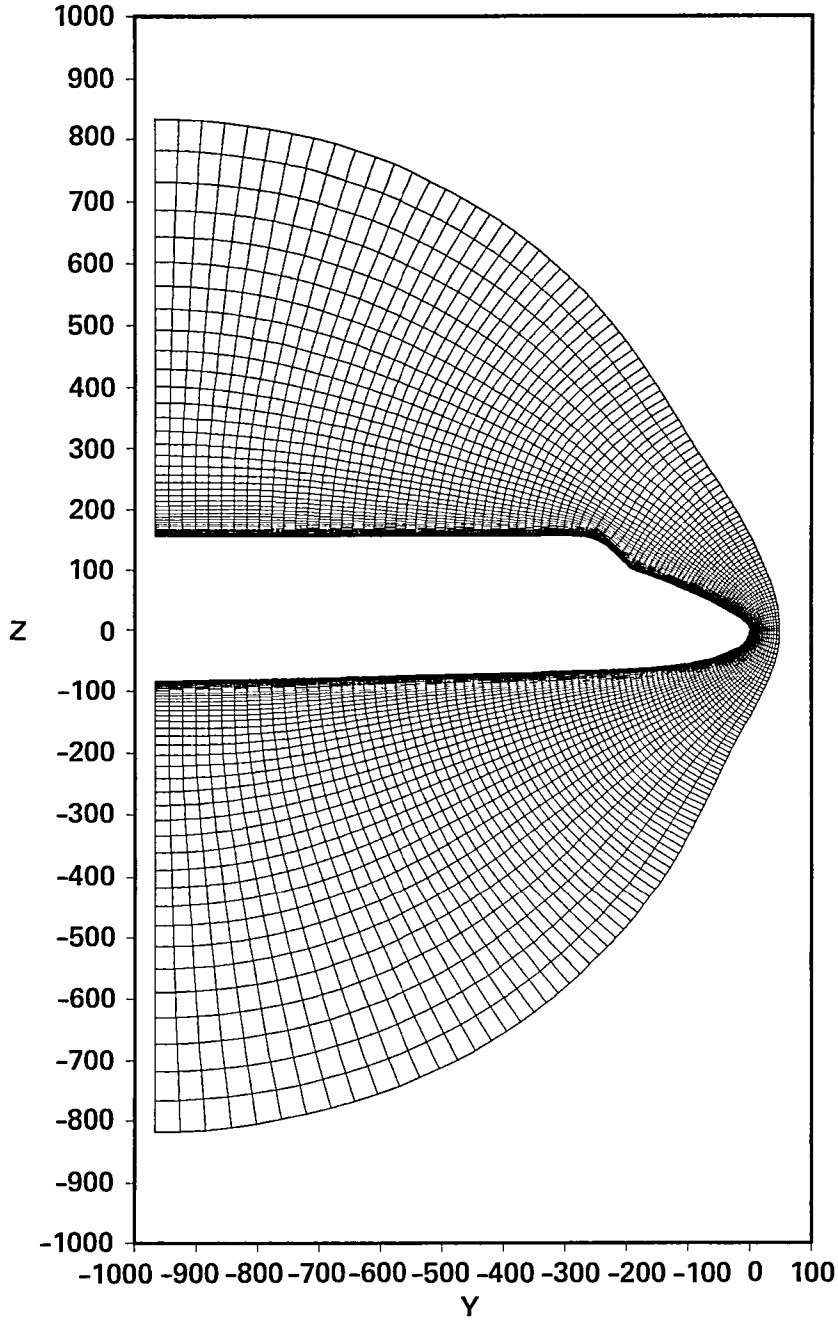
c. close-up of the trailing edge midspan grid.

Continued



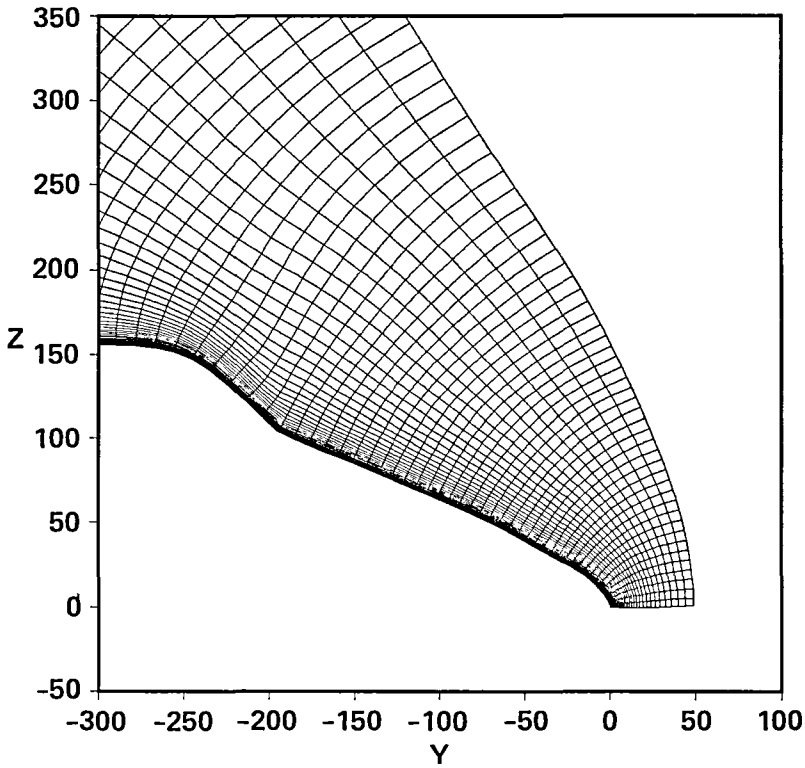
d. $\xi = const$ plane near a wing tip axis.

Continued



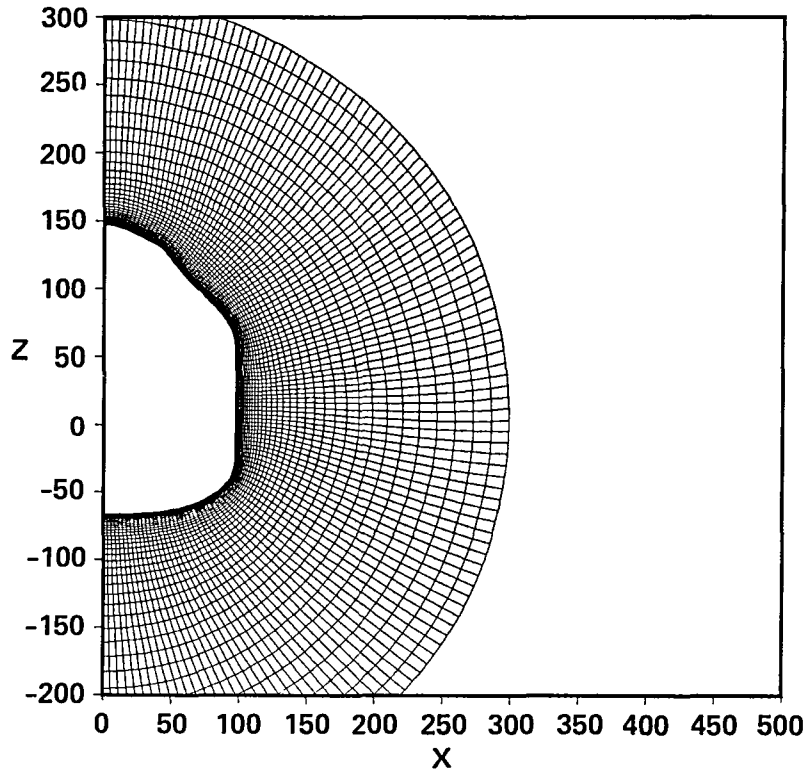
a. grid segments in the windward and leeward plane of the shuttle orbiter.

Fig 6 Spherical-like grid generated for a simplified space shuttle orbiter.



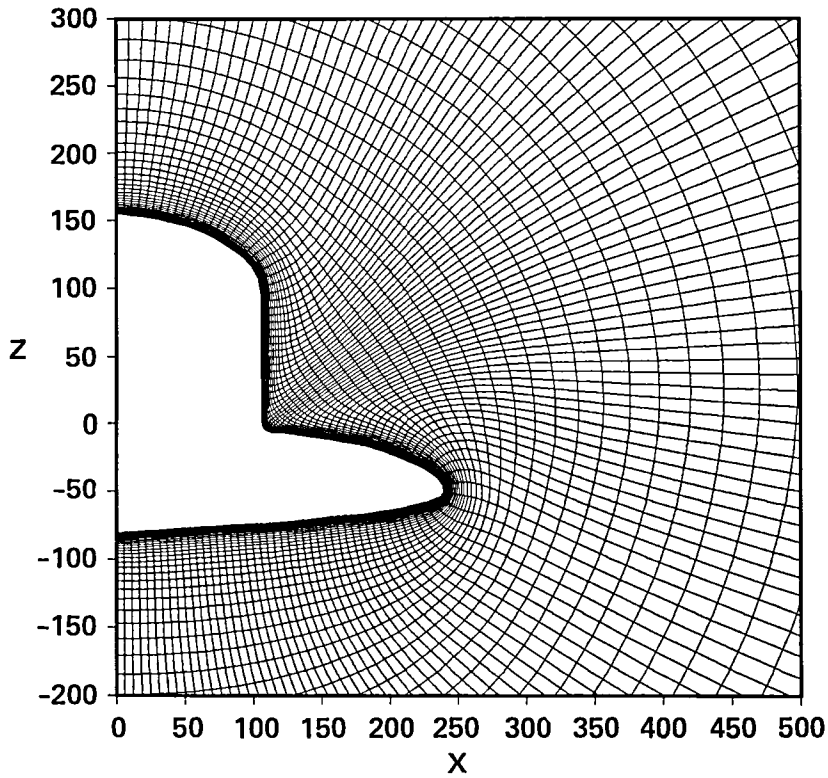
b. Grid in the leeward plane near the nose of the shuttle orbiter.

Continued



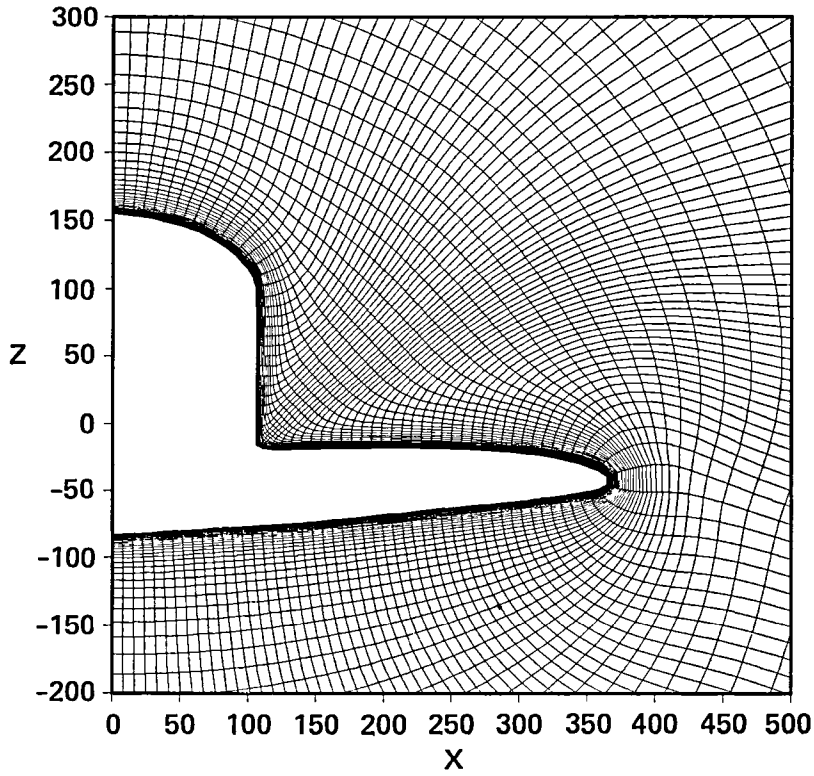
c Cross-sectional grid view of orbiter, near canopy, projected onto an $x = const$ plane.

Continued



d. Cross-sectional grid view of orbiter, strake region, projected onto an $x = const$ plane

Continued



e. Cross-sectional grid view of orbiter wing region, projected onto an $x = const$ plane

Concluded

1 Report No	2 Government Accession No	3 Recipient's Catalog No
NASA TM-86753		

4 Title and Subtitle	5 Report Date
GENERATION OF THREE DIMENSIONAL BODY FITTED COORDINATES USING HYPERBOLIC PARTIAL DIFFERENTIAL EQUATIONS	June 1985
	6 Performing Organization Code
7 Author(s)	8 Performing Organization Report No
Joseph Steger and Yehia Rizk (Informatics General Corp., Palo Alto, CA)	85246
9 Performing Organization Name and Address	10 Work Unit No
Ames Research Center Moffett Field, Calif. 94035	T-6458
	11 Contract or Grant No
	13 Type of Report and Period Covered
12 Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546	14 Sponsoring Agency Code 505-31-01

15 Supplementary Notes

Point of Contanct: Joseph Steger, Ames Research Center, MS 202A-1
Moffett Field, Calif. 94035
415-694-6459 or FTS 448-6459

16 Abstract

An efficient numerical mesh generation scheme capable of creating orthogonal or nearly orthogonal grids about moderately complex three dimensional configurations is described. The mesh is obtained by marching outward from a user specified grid on the body surface. Using spherical grid topology, grids have been generated about full span rectangular wings and a simplified space shuttle orbiter.

17 Key Words (Suggested by Author(s))	18 Distribution Statement
Grid generation Hyperbolic grids Finite Difference methods	Unlimited Subject Category - 64

19 Security Classif (of this report)	20 Security Classif (of this page)	21 No of Pages	22 Price*
Unclassified	Unclassified	36	A03

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document