23

# MESH GENERATION USING ALGEBRAIC TECHNIQUES

Peter R. Eiseman
Institute for Computer Applications in
Science and Engineering, USRA

and

Robert E. Smith
NASA Langley Research Center

Coordinate transformations are powerful tools for the solution of the partial differential equations which describe physical phenomena. The use of transformations leads to well ordered discretizations of the physical domain and thereby renders a simplification in a numerical solution process. The discretization is constrained by the underlying physics, the problem geometry and the topology of the region where the solution is to be obtained. The constraints can be stated in geometric terms. In particular they can be categorized as boundary constraints, uniformity constraints, and internal constraints. Boundary constraints include: the basic geometry of solid objects, the transmissive junctures between and around solid objects, the choice of representation for the boundaries, the angles at which transverse coordinate curves intersect boundaries, and the rate of entry for such coordinate curves. Uniformity constraints are applied to either local or global distributions of coordinate curves or points to form a basis from which the curves or points can be redistributed.

This may be based on an a priori specification of a distribution function or on a solution adaptive approach. In either case, the redistribution must not be distorted by the underlying transformation. Internal constraints are applicable when an interior shape or interior mesh structure is to be smoothly embedded within a global mesh to simplify the simulation of physical processes in the given region.

Algebraic mesh generation techniques are highly advantageous for meeting the constraints described above. Algebraic techniques provide exact control of the mesh properties necessary to satisfy the given constraints. Although other methods have been developed which provide some degree of control, the level of control is not in general sufficient to satisfy certain of the constraints. For example, the smooth embedding of a Cartesian mesh within a global mesh structure cannot be readily constructed with the application of differential equation techniques. Also, three dimensional meshes are not in general readily obtained with non-algebraic techniques. On the other hand, algebraic techniques require more complex specification of the data to assemble a mesh. The purpose of this paper is to present an overview of algebraic techniques for mesh generation and set forth the underlying concepts which have been successful. Both two- and three-dimensional domains are considered.

## The Multi-Surface Transformation

When curvilinear coordinates are employed in the numerical solution of a boundary value problem, constraints must often be placed upon the coordinates, in addition to the basic requirement that the bounding surfaces are coordinate surfaces of one or more coordinate systems. The locations of the constraints can occur anywhere in the problem domain. On the boundaries, a particular pointwise distribution may be needed; in regions near boundaries, a particular coordinate form may be advantageous; and away from the boundaries, an internal coordinate specification may also be required. Typically, the constraints will arise either to resolve regions with large solution gradients or to cause some simplification in the problem formulation and solution.

In conjunction with the demand for constraints, the general multi-surface transformation [1] will be examined. The multi-surface transformation is a method for coordinate generation between an inner bounding surface $\vec{P}_1$ and outer bounding surface $\vec{P}_N$. To establish a particular distribution of mesh points on each bounding surface, a common parameterization $\vec{t}$ is chosen for each surface. This is equivalent to a coordinate description of the surfaces which yields the desired surface mesh when the parametric components of $\vec{t}$ are given a <u>uniform</u> discretization. With the parametric description, the inner and outer bounding surfaces are denoted by $\vec{P}_1(\vec{t})$ and $\vec{P}_N(\vec{t})$ respectively. In continuation, parameterized intermediate surfaces $\vec{P}_2(\vec{t}),\ldots,\vec{P}_{N-1}(\vec{t})$ are introduced so that they can be used as controls over the internal form of the coordinates. The intermediate surfaces are not coordinate surfaces but, instead, are surfaces which are used to establish a vector field that is composed of tangent vectors to the coordinate curves spanning the coordinate system to connect bounding surfaces. It is also assumed that the collection of surfaces $\vec{P}_1(\vec{t})$, $\vec{P}_2(\vec{t}),\ldots,\vec{P}_N(\vec{t})$ is ordered from bounding surface to bounding surface. An illustration is given in Fig. 1. For a fixed

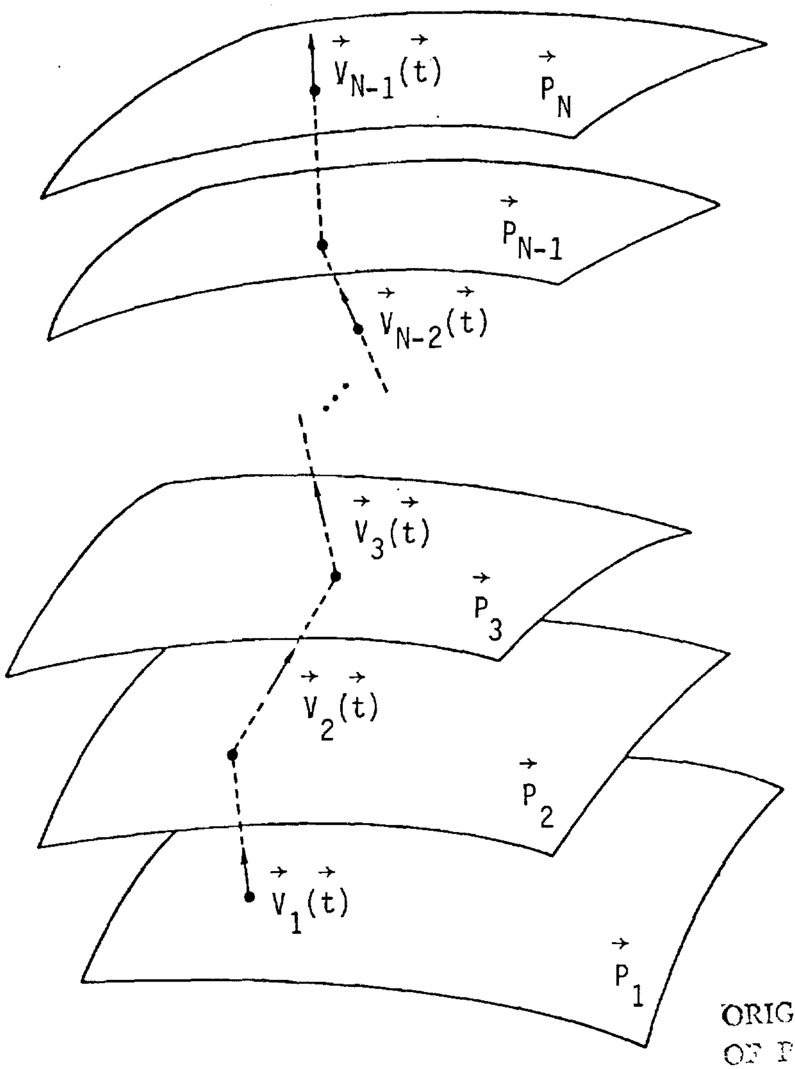

Fig. 1 - A piecewise linear curve and its tangent field.

parameter value $\vec{t}$, there is a corresponding point on each surface. The piecewise linear curve obtained by connecting corresponding points is given by the dashed curve in Fig. 1. From the figure, it can be observed that the tangent directions determined by the piecewise linear curve are piecewise constants. As $\vec{t}$ is varied, the field of tangent directions obtain their smoothness (level of differentiability) only in $\vec{t}$. To obtain smoothness in going from bounding surface to bounding surface, a sufficiently smooth interpolation must be performed. The result is a smooth vector field of undetermined magnitude which gives the desired tangential directions for coordinate curves connecting the bounding surfaces. A unique vector field of tangents is then obtained by correctly choosing magnitudes so that, on integration, the bounding surfaces are fit precisely.

In symbols, a vector field tangent to the piecewise linear curves is given by

$$\vec{V}_k(\vec{t}) = A_k[\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \ , \qquad (1)$$

between the kth and (k + 1)th surfaces where $k$ is taken to vary (if $N > 2$) from the first bounding surface to the final intermediate surface. These vectors are indicated in Fig. 1. The coefficients $A_k$ are scalars which determine the magnitude of the vectors but not the directions. When an independent variable $r$ is assumed for the spanning direction, a partition $r_1 < \ldots < r_{N-1}$ can be specified in correspondence with the tangents of Eq. 1. The partitioned variable can then be used to represent the tangents as the discrete vector valued function which maps $r_k$ into $\vec{V}_k$ for $k = 1 \ldots N-1$. A sufficiently smooth vector field $\vec{V}(r,\vec{t})$ is then obtained by a sufficiently smooth interpolation $\vec{V}(r_k,\vec{t}) = \vec{V}_k(\vec{t})$. With $r$ as a continuous independent variable, the r-derivative of the coordinate transformation $\vec{P}(r,\vec{t})$ is equal to the interpolant and is given by

$$\frac{\partial \vec{P}}{\partial r} = \vec{V} = \sum_{k=1}^{N-1} \psi_k(r)\vec{V}_k(\vec{t}) \;, \tag{2}$$

where $\psi_k(r_j)$ is unity at $k = j$ and vanishes otherwise. When Eq. 2 is integrated with an initial $r_1$ value of $\vec{P}_1(\vec{t})$, the transformation becomes

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + \sum_{k=1}^{N-1} A_k G_k(r)[\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \;, \tag{3}$$

where

$$G_k(r) = \int_{r_1}^{r} \psi_k(x)dx \;, \tag{4}$$

from which we observe that the interpolants $\psi_k$ must be continuously differentiable up to an order which is one less than the level of smoothness desired for the coordinates. The construction of local controls on the coordinates will rely upon the development of suitably smooth interpolation functions. If the magnitudes $A_k$ of Eq. 1 are chosen so that each $A_k G_k(r_{N-1})$ is unity, then the evaluation of the transformation at $r_{N-1}$ will reduce to $\vec{P}_N(t)$ by means of a telescopic collapse of terms in Eq. 3. With this choice, we obtain the general multisurface transformation of Eiseman [1] which is given by

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + \sum_{k=1}^{N-1} \frac{G_k(r)}{G_k(r_{N-1})} [\vec{P}_{k+1}(\vec{t}) - \vec{P}_k(\vec{t})] \;. \tag{5}$$

On examination, each interpolation function $\psi_k$ can be rescaled without changing the transformation; hence, the original vector field interpolation becomes an interpolation only on vector directions.

When the interpolants $\psi_k$ are polynomials in $r$, the coordinate curves which connect the bounding surfaces are globally defined by polynomials in $r$ of one greater degree. The specification of boundary properties for the curves and a global control over their general form are obtained by choices of intermediate surfaces and the associated partitions of $r$. For notational simplicity, let $r_1 = 0$ and $r_{N-1} = 1$. In the simplest case when there are no intermediate control surfaces, there is just one vector field direction $\vec{V}_1(\vec{t})$ which is determined solely by the bounding surfaces. The interpolant $\psi_1$ is then a constant function, $G_1(r) = r\psi_1$, $G_1(r)/G_1(1) = r$, and the polynomial 2-surface transformation becomes

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + r[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})] \ , \qquad (6)$$

which is the case of linear coordinate curves connecting boundaries. The linear case has occurred in many studies including [2], [3], and [4] and is limited to at most one prescribed coordinate property per boundary which can be either a pointwise distribution or a distribution of angles with the linear transverse coordinate curves. To allow for the prescription of an additional coordinate property on one of the boundaries, an intermediate control surface is introduced and the polynomial 3-surface transformation is computed from Eq. 3 with $\psi_1 = 1 - r$ and $\psi_2 = r$ corresponding to directions of $\vec{V}_1(\vec{t})$ and $\vec{V}_2(\vec{t})$ of Eq. 1 and Fig. 1. The integrals from Eq. 4 become

$$G_1(r) = r - \frac{r^2}{2} , \qquad G_2(r) = \frac{r^2}{2} , \tag{7}$$

and since $G_1(1) = G_2(1) = \frac{1}{2}$, the original vector field which is discrete in $r$ is determined by

$$\vec{V}_1(\vec{t}) = 2[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})], \qquad \vec{V}_2(\vec{t}) = 2[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})], \tag{8}$$

because $A_k = 1./G_k(1)$ in Eq. 1. Upon substitution from Eq. 7 the polynomial 3-surface transformation is given by

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + r(2 - r)[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})] + r^2[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})]. \tag{9}$$

In continuation, an additional coordinate property can be prescribed on each boundary when two intermediate control surfaces are used. The polynomial 4-surface transformation is computed from Eq. 3 with interpolants

$$\psi_1(r) = (1 - r)(r_2 - r),$$

$$\psi_2(r) = r(1 - r), \tag{10}$$

$$\psi_3(r) = (r - r_2)r,$$

which respectively correspond to the directions of $\vec{V}_1(\vec{t})$, $\vec{V}_2(\vec{t})$, and $\vec{V}_3(\vec{t})$ which, in turn, are respectively associated with partition points $r_1 = 0$, $r_2$, and $r_3 = 1$ and which are defined to vanish at all partition points except the ones of association for each function. For simplicity, we will set $r_2 = \frac{1}{2}$ so that the partition is uniform. The nonuniform case is a simple but algebraically more complex extension [1]. With $r_2 = \frac{1}{2}$, the integrals from Eq. 4 become

$$G_1(r) = \frac{1}{2} r - \frac{3}{4} r^2 + \frac{1}{3} r^3,$$

$$G_2(r) = \frac{1}{2} r^2 - \frac{1}{3} r^3, \tag{11}$$

$$G_3(r) = \frac{1}{3} r^3 - \frac{1}{4} r^2,$$

and from an evaluation at the endpoint $r = 1$ we have $G_1(1) = \frac{1}{12}$, $G_2(1) = \frac{1}{6}$, and $G_3(1) = \frac{1}{12}$. By substitution, the polynomial 4-surface transformation is given by

$$\vec{P}(r,\vec{t}) = \vec{P}_1(\vec{t}) + r(6 - 9r + 4r^2)[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})]$$

$$+ r^2(3 - 2r)[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})] \tag{12}$$

$$+ r^2(4r - 3)[\vec{P}_4(\vec{t}) - \vec{P}_3(\vec{t})],$$

with r-derivative

$$\frac{\partial\vec{P}}{\partial r} = 6(1 - r)(1 - 2r)[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})]$$

$$+ 6r(1 - r)[\vec{P}_3(\vec{t}) - \vec{P}_2(\vec{t})] \tag{13}$$

$$+ 6r(2r - 1)[\vec{P}_4(\vec{t}) - \vec{P}_3(\vec{t})] .$$

By direct evaluation

$$\vec{P}(0,\vec{t}) = \vec{P}_1(\vec{t}),$$

$$\tag{14}$$

$$\vec{P}(1,\vec{t}) = \vec{P}_4(\vec{t}),$$

and

$$\frac{\partial\vec{P}}{\partial r}(0,\vec{t}) = 6[\vec{P}_2(\vec{t}) - \vec{P}_1(\vec{t})],$$

$$\tag{15}$$

$$\frac{\partial\vec{P}}{\partial r}(1,\vec{t}) = 6[\vec{P}_4(\vec{t}) - \vec{P}_3(\vec{t})],$$

which explicitly shows that in addition to fitting the boundaries (Eq. 14), the intermediate surfaces $\vec{P}_2(t)$ and $\vec{P}_3(\vec{t})$ can be used to control the angles at which the transverse coordinate curves in $r$ intersect the boundaries (Eq. 15). Moreover, the choice of intermediate surfaces can also be used to control the shape of the transverse curves and the distribution of the constant $r$ coordinate surfaces. The general derivation and discussion is given in Eiseman [1]. For our purposes, the discussion on coordinate system controls will be deferred until local methods are presented for our survey of some of the material developed in Eiseman [5], [6].

An alternative form of the polynomial 4-surface transformation (Eq. 12) can be obtained from the evaluations of the transformation (Eq. 14) and its derivatives (Eq. 15). With the evaluations, the intermediate surfaces can be expressed entirely in terms of boundary data, which results in

$$\vec{P}_2(\vec{t}) = \vec{P}(0,\vec{t}) + \frac{1}{6}\frac{\partial\vec{P}}{\partial r}(0,\vec{t}),$$

(16)

and

$$\vec{P}_3(\vec{t}) = \vec{P}(1,\vec{t}) - \frac{1}{6}\frac{\partial\vec{P}}{\partial r}(1,\vec{t}).$$

Upon substitution of Eqs. 14 and 16 into Eq. 12 we obtain

$$\begin{aligned}\vec{P}(r,\vec{t}) &= (1 - 3r^2 + 2r^3)\vec{P}(0,\vec{t}) + r^2(3 - 2r)\vec{P}(1,\vec{t}) \\ &\quad + r(1 - r)^2 \frac{\partial\vec{P}}{\partial r}(0,\vec{t}) - r^2(1 - r)\frac{\partial\vec{P}}{\partial r}(1,\vec{t}),\end{aligned} \tag{17}$$

after grouping terms by boundary type. By examination, the coefficients of the boundary evaluations for the transformation and its r-derivative can easily be identified as just the canonical Hermite cubic interpolants on the unit interval $0 \leq r \leq 1$. When the r-derivatives are specified to be normal to the respective boundaries, we obtain the transformation presented by Smith and Weigel [3].

In continuation, polynomial N-surface transformations can be systematically established from Eqs. 3-4 and the interpolants

$$\psi_k(r) = \prod_{\substack{i=1 \\ i \neq k}}^{N-1} (r - r_i), \tag{18}$$

for $k = 1,2,\ldots,N-1$. In each case, the transverse coordinate curves are polynomials of degree N-1 in $r$ with vector valued coefficients which are functions of the surface coordinates $\vec{t}$. Polynomials, however, are globally defined for all $r$, and as a consequence, local mesh properties cannot be controlled without a global effect. As an example, suppose that we wish to smoothly embed a general rectilinear Cartesian system within a global mesh structure to obtain a system of the form illustrated in Fig. 2 where the Cartesian region within the mesh is bounded by the darkened curves. In the Cartesian part of the mesh, coordinate curves in $r$ would be lines which pass through it at a uniform rate. Since global polynomials in $r$ would be uniquely determined by the Cartesian region, curved boundaries could not be fitted.

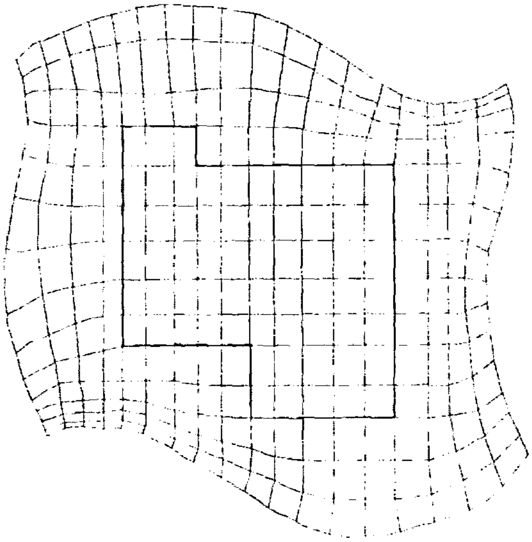

Fig. 2 - A smoothly embedded Cartesian region within a global mesh structure.

To obtain precise local controls which could be successfully applied to generate a mesh as illustrated above in Fig. 2, local forms of the multisurface transformation (Eqs.3-4) were established and analyzed by Eiseman [5], [6], and [7]. Our discussion will follow the development given by Eiseman in [6] and will focus upon two-dimensional applications with a surface coordinate $\vec{t} = t$. When the interpolants $\psi_k$ are nonvanishing on only a local region, the precise local controls over the coordinates that were obtained will be illustrated with the local piecewise linear interpolants that are depicted in Fig. 3. For algebraic simplicity, the analysis is restricted to the case with the uniform partition $r_k = k$ with the clear understanding that nonuniform partitions will follow the same analytic pattern. Since the multi-surface transformation remains unchanged when the interpolants are scaled by any sequence of nonzero

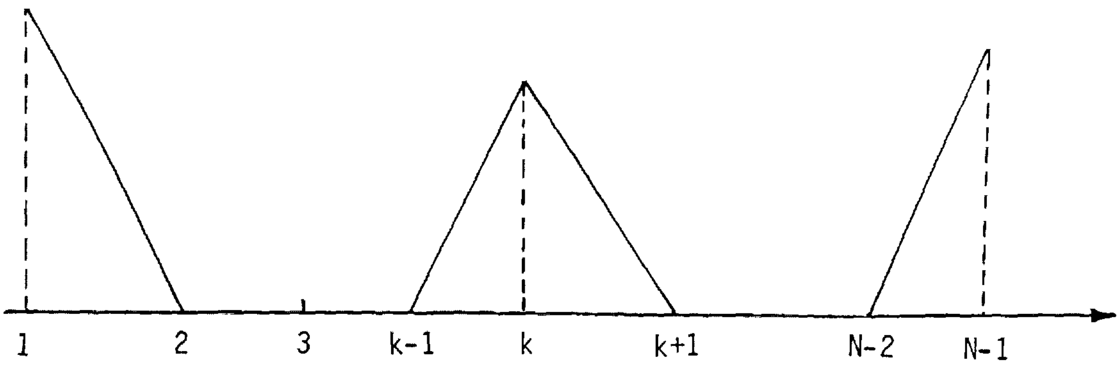

Fig. 3 - Piecewise linear local interpolants with partition points $r_k = k$ for $k = 1,2,\ldots,N-1$.

numbers, the height $\psi_k(r_k)$ of each interpolant can be chosen arbitrarily. In particular, the form of the multi-surface transformation can be simplified when the heights are adjusted so that each interpolant integrates to unity which yields $G_k(r_{N-1}) = 1$ for all $k$. The integrals are obtained from triangular areas, and by direct observation, lead to the height adjustments $\psi_1(r_1) = 2$, $\psi_k(r_k) = 1$, and $\psi_{N-1}(r_{N-1}) = 2$ in correspondence with the successive illustrations in Fig. 3. Also, in correspondence, the explicit form of the normalized interpolants are given by

$$\psi_1(r) = \left\{ \begin{array}{ll} 2(2 - r) & \text{for } 1 \leq r < 2 \\ & \\ 0 & \text{for } 2 \leq r \leq N - 1 \end{array} \right\}; \quad (19)$$

$$\psi_k(r) = \left\{ \begin{array}{ll} 0 & \text{for} \quad 1 \leq r < k - 1 \\ (r - k) + 1 & \text{for} \quad k - 1 \leq r < k \\ (k - r) + 1 & \text{for} \quad k \leq r < k + 1 \\ 0 & \text{for} \quad k + 1 \leq r \leq N - 1 \end{array} \right\} ; \qquad (20)$$

$$\psi_{N-1}(r) = \left\{ \begin{array}{ll} 0 & \text{for} \quad 1 \leq r < N - 2 \\ & \\ 2(r - N + 2) & \text{for} \quad N - 2 \leq r \leq N - 1 \end{array} \right\} ; \qquad (21)$$

and their integrals defined in Eq. 4, by

$$G_1(r) = \left\{ \begin{array}{ll} 1 - (2 - r)^2 & \text{for} \quad 1 \leq r < 2 \\ & \\ 1 & \text{for} \quad 2 \leq r \leq N - 1 \end{array} \right\} ; \qquad (22)$$

$$G_k(r) = \left\{ \begin{array}{ll} 0 & \text{for} \quad 1 \leq r < k - 1 \\ 1/2(r - k)^2 + (r - k) + 1/2 & \text{for} \quad k - 1 \leq r < k \\ -1/2(k - r)^2 - (k - r) + 1/2 & \text{for} \quad k \leq r < k + 1 \\ 1 & \text{for} \quad k + 1 \leq r \leq N - 1 \end{array} \right\} ; \qquad (23)$$

$$G_{N-1}(r) = \left\{ \begin{array}{ll} 0 & \text{for} \quad 1 \leq r < N - 2 \\ & \\ (r - N + 2)^2 & \text{for} \quad N - 2 \leq r \leq N - 1 \end{array} \right\} , \qquad (24)$$

which are depicted in Fig. 4.

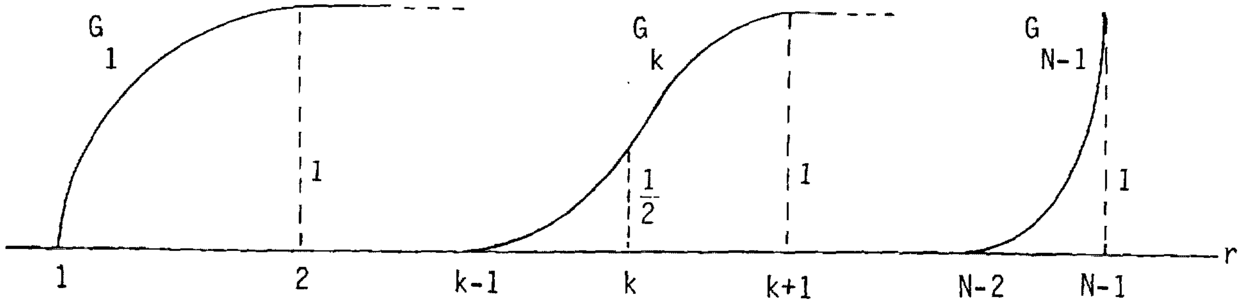

Fig. 4 - Integrals of normalized interpolants for the partition $r_k = k$.

On the interval $k \leq r < k + 1$, the integrals $G_i(r)$ which correspond to interpolants defined over nonintersecting intervals are either unity or vanishing depending upon whether the interval of definition precedes or follows the interval under examination. When $i = 1,2,\ldots,k - 1$ which is nonvacuous for $k > 1$, the integrals $G_i(r)$ have been evaluated over the entire domain for which the respective interpolant $\psi_i$ is nonvanishing; hence, these preceding integrals are unity by the chosen normalization. When $i = k + 2, k + 3,\ldots,N - 1$, the interpolants $\psi_i$ each vanish on $1 \leq r < k + 1$; hence, the integrals $G_i$ also vanish there. As a consequence, $G_i$ for $i = k, k + 1$ yield the only nontrivial contributions for the multi-surface transformation which reduces to

$$\vec{P}(r,t) = \vec{P}_1(t) + \sum_{i=1}^{k-1} [\vec{P}_{i+1}(t) - \vec{P}_i(t)] + G_k(r)[\vec{P}_{k+1}(t) - \vec{P}_k(t)]$$

$$+ G_{k+1}(r)[\vec{P}_{k+2}(t) - \vec{P}_{k+1}(t)] + 0 \qquad (25)$$

$$= \vec{P}_k(t) + G_k(r)[\vec{P}_{k+1}(t) - \vec{P}_k(t)] + G_{k+1}(r)[\vec{P}_{k+2}(t) - \vec{P}_{k+1}(t)],$$

which depends upon only the three control surfaces $\vec{P}_k$, $\vec{P}_{k+1}$, $\vec{P}_{k+2}$ which can be arbitrarily selected to our advantage when they are not bounding surfaces. With substitutions from Eqs. 22-24, we obtain the partition point ($r_k = k$ for $k = 1,2,\ldots,N - 1$) evaluations

$$\begin{aligned}
\vec{P}(1,t) &= \vec{P}_1(t),\\
\vec{P}(2,t) &= \tfrac{1}{2}\,[\vec{P}_2(t) + \vec{P}_3(t)],\\
&\;\;\vdots\\
\vec{P}(k,t) &= \tfrac{1}{2}\,[\vec{P}_k(t) + \vec{P}_{k+1}(t)],\\
\vec{P}(k+1,t) &= \tfrac{1}{2}\,[\vec{P}_{k+1}(t) + \vec{P}_{k+2}(t)],\\
&\;\;\vdots\\
\vec{P}(N-2,t) &= \tfrac{1}{2}\,[\vec{P}_{N-2}(t) + \vec{P}_{N-1}(t)],\\
\vec{P}(N-1,t) &= \vec{P}_N(t),
\end{aligned} \tag{26}$$

which, in addition to boundary fitting at the end points $r = 1$ and $r = N - 1$, also shows that the transformation passes through the midpoints of the lines which connect the intermediate control surfaces for any fixed surface coordinate $t$. Moreover, from the general multi-surface construction (Eqs. 1-5 and Fig. 1), the transverse coordinate curves are tangent to the connecting lines at the partition point evaluations. The tangents at partition points can be explicitly obtained from substitutions of the interpolation functions (Eqs. 19-21) into the r-derivative of the transformation (Eq. 25) and are given by

$$
\begin{aligned}
\frac{\partial \vec{P}}{\partial r}(1,t) &= 2[\vec{P}_2(t) - \vec{P}_1(t)], \\
\frac{\partial \vec{P}}{\partial r}(2,t) &= \vec{P}_3(t) - \vec{P}_2(t), \\
&\vdots \\
\frac{\partial \vec{P}}{\partial r}(k,t) &= \vec{P}_{k+1}(t) - \vec{P}_k(t), \\
\frac{\partial \vec{P}}{\partial r}(k+1,t) &= \vec{P}_{k+2}(t) - \vec{P}_{k+1}(t), \\
&\vdots \\
\frac{\partial \vec{P}}{\partial r}(N-2,t) &= \vec{P}_{N-1}(t) - \vec{P}_{N-2}(t), \\
\frac{\partial \vec{P}}{\partial r}(N-1,t) &= 2[\vec{P}_N(t) - \vec{P}_{N-1}(t)].
\end{aligned}
\tag{27}
$$

In graphical form, this process is depicted in Fig. 5.

C-2

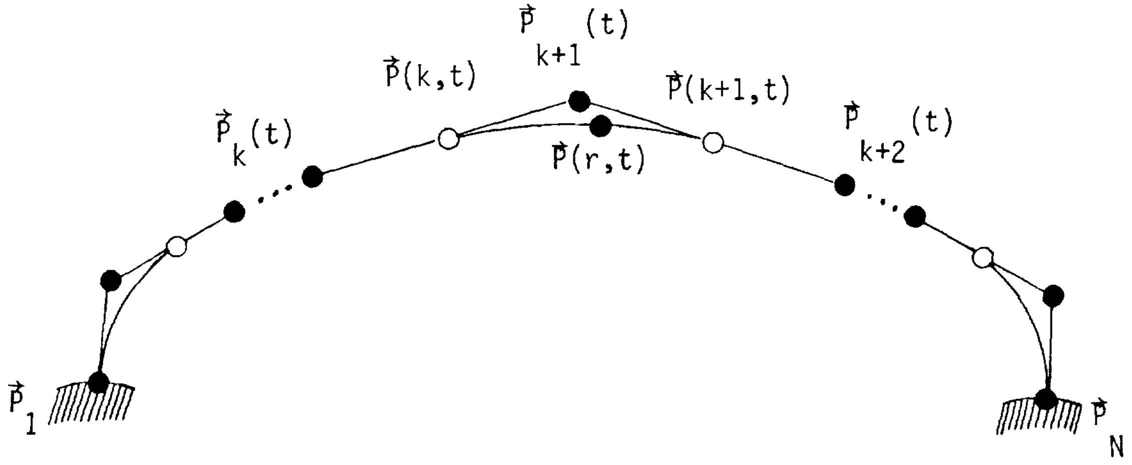

Fig. 5 - Coordinate curve segments for $k \leq r < k+1$.

Between the control surfaces $\vec{P}_i$ and $\vec{P}_j$ for $i > j$, the distribution of constant $r$ coordinate surfaces can be controlled for the general multisurface transformation (Eqs. 4-5) when uniformity can be specified along a direction of measurement

$$\hat{\tau}(\vec{t}) = \frac{\vec{P}_i(\vec{t}) - \vec{P}_j(\vec{t})}{||\vec{P}_i(\vec{t}) - \vec{P}_j(\vec{t})||}, \qquad (28)$$

for then arbitrary distributions can be applied relative to uniform conditions. An illustration is given in Fig. 6.

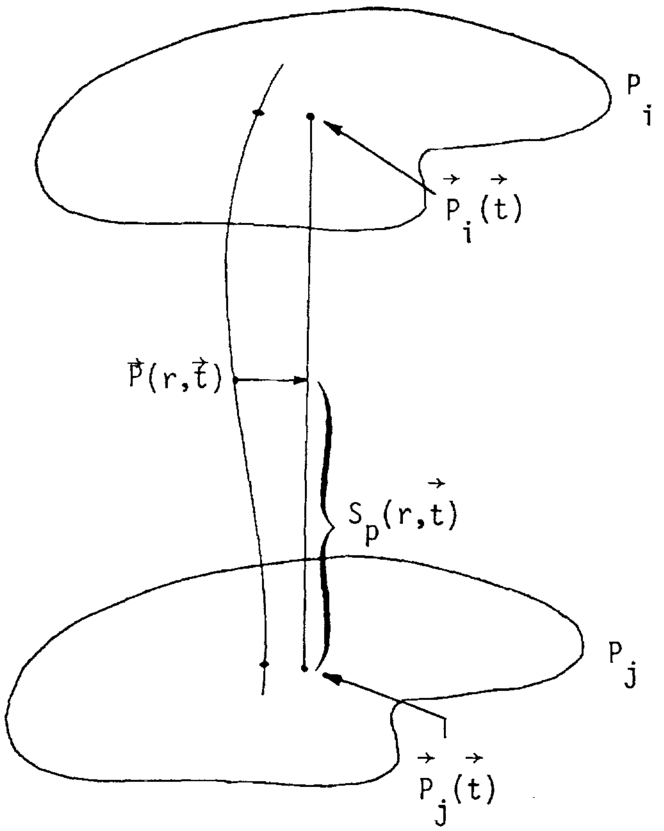

Fig. 6 - Measurement of uniformity.

To obtain uniformity, the projected arc length

$$S_p(r,\vec{t}) = [\vec{P}(r,\vec{t}) - \vec{P}_j(\vec{t})] \cdot \hat{\tau}(\vec{t}), \tag{29}$$

depicted in Fig. 6 must be linear in $r$, or equivalently $\partial S_p/\partial r$ must be independent of $r$. But then from Eq. 25 and with the relative projections $C_m(t) = [\vec{P}_{m+1}(t) - \vec{P}_m(t)] \cdot \hat{\tau}(t)$, we have

$$\frac{\partial S_p}{\partial r} = \psi_k(r)C_k(t) + \psi_{k+1}(r)C_{k+1}(t)$$

$$= \left\{ \begin{array}{ll} -2C_1(t) + C_2(t) & \text{for } k = 1 \\ -C_k(t) + C_{k+1}(t) & \text{for } 1 < k < N - 1 \\ -C_{N-2}(t) + 2C_{N-1}(t) & \text{for } k = N - 1 \end{array} \right\} r + \text{function of } t, \quad (30)$$

where the last equality comes from Eqs. 19-21. Hence, for $k = j, j+1, \ldots, i-1$, uniformity is obtained if $2C_1(t) = C_2(t)$ should $k = 1$, $C_k(t) = C_{k+1}(t)$ should $1 < k < N - 1$, and $C_{N-2}(t) = 2C_{N-1}(t)$ should $k = N - 1$. A more thorough discussion on uniformity is available in Eiseman [1] for the global case, in Eiseman [5], [6] for the local case, and in Eiseman [7] for the general cases.

To explicitly demonstrate the application of the local controls, and at the same time, reveal the basic algorithmic steps, coordinates will be obtained for a simple transition from a purely Cartesian system into a purely Polar system. For $0 \leq t \leq 1$, the Cartesian coordinates will be specified below a line $\vec{Q}(t) = (2t-1,0)$ and the Polar coordinates, beyond a circular arc $\sqrt{2}\,\hat{u}(t)$ where $\hat{u}(t) = (-\cos\theta, \sin\theta)$ for $\theta = (2t + 1)\pi/4$. The line and the arc are depicted in Fig. 7.

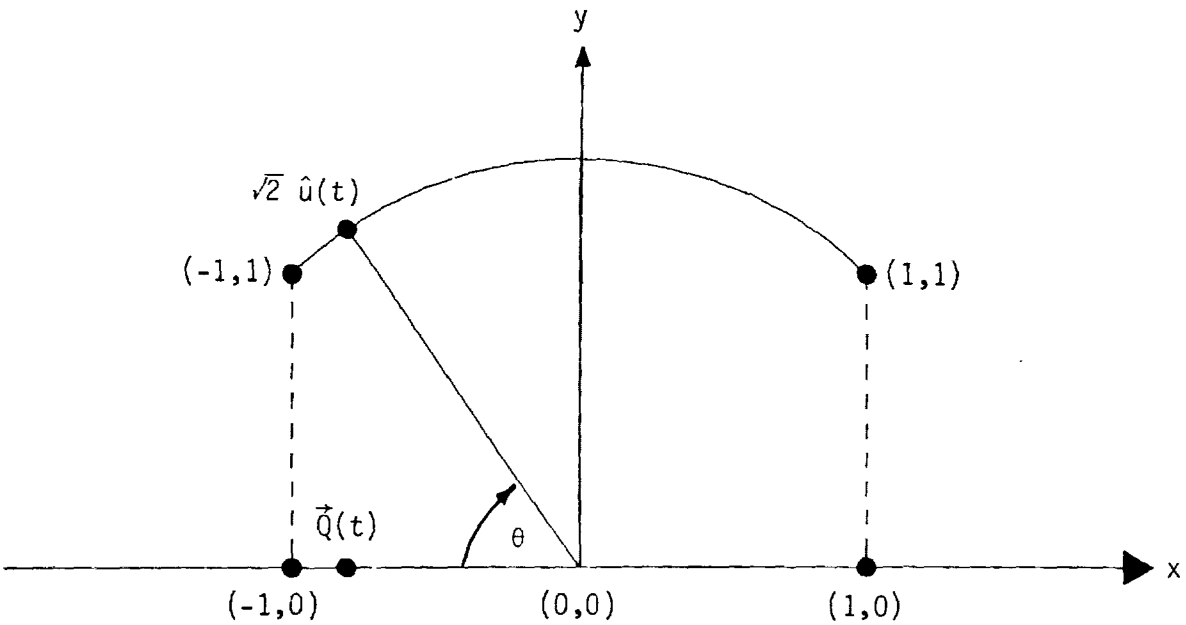

Fig. 7 - Basic curves for the Cartesian to Polar transition example.

To obtain uniformity near the sides (t = 0,1) of the transitional region, the unit vertical distance will be used as a basis for displacements to establish uniformity below the line and beyond the circular arc. For the line, let

$$\begin{aligned}
\vec{P}_1(t) &= \vec{Q}(t) - (0,\tfrac{5}{2}),\\
\vec{P}_2(t) &= \vec{Q}(t) - (0,2),\\
\vec{P}_3(t) &= \vec{Q}(t) - (0,1),\\
\vec{P}_4(t) &= \vec{Q}(t),
\end{aligned} \tag{31}$$

so that for

$$\hat{\tau}(t) = \frac{\vec{P}_4(t) - \vec{P}_1(t)}{||\vec{P}_4(t) - \vec{P}_1(t)||} = (0,1), \tag{32}$$

we have

$$\begin{aligned} C_1(t) &= [\vec{P}_2(t) - \vec{P}_1(t)] \cdot \hat{\tau}(t) = (0,\tfrac{1}{2}) \cdot (0,1) = \tfrac{1}{2}, \\ C_2(t) &= [\vec{P}_3(t) - \vec{P}_2(t)] \cdot \hat{\tau}(t) = (0,1) \cdot (0,1) = 1, \\ C_3(t) &= [\vec{P}_4(t) - \vec{P}_3(t)] \cdot \hat{\tau}(t) = (0,1) \cdot (0,1) = 1, \end{aligned} \tag{33}$$

which satisfies uniformity for $1 \leq r \leq 3$ and yields a Cartesian system from

$$\vec{P}(1,t) = \vec{P}_1(t) = \vec{Q}(t) - (0,\tfrac{5}{2}), \tag{34}$$

up to

$$\vec{P}(3,t) = \tfrac{1}{2}\,[\vec{P}_3(t) + \vec{P}_4(t)] = \vec{Q}(t) - (0,\tfrac{1}{2}). \tag{35}$$

Similarly, for the circular arc, let

$$\begin{aligned} \vec{P}_5(t) &= \sqrt{2}\ \hat{u}(t), \\ \vec{P}_6(t) &= (1 + \sqrt{2})\ \hat{u}(t), \end{aligned} \tag{36}$$

$$\vec{P}_7(t) = (\tfrac{3}{2} + \sqrt{2})\ \hat{u}(t),$$

be the last three surfaces so that for

$$\hat{\tau}(t) = \frac{\vec{P}_7(t) - \vec{P}_5(t)}{||\vec{P}_7(t) - \vec{P}_5(t)||} = \hat{u}(t) \tag{37}$$

we have

$$C_5(t) = [\vec{P}_6(t) - \vec{P}_5(t)] \cdot \hat{\tau}(t) = \hat{u}(t) \cdot \hat{u}(t) = 1,$$

and (38)

$$C_6(t) = [\vec{P}_7(t) - \vec{P}_6(t)] \cdot \hat{\tau}(t) = \frac{1}{2}\ \hat{u}(t) \cdot \hat{u}(t) = \frac{1}{2},$$

which satisfies uniformity for $5 \leq r \leq 6$ and yields a Polar system from the circular arc

$$\vec{P}(5,t) = \frac{1}{2}\ [\vec{P}_5(t) + \vec{P}_6(t)] = (\tfrac{1}{2} + \sqrt{2})\ \hat{u}(t), \tag{39}$$

up to the circular arc

$$\vec{P}(6,t) = \vec{P}_7(t) = (\tfrac{3}{2} + \sqrt{2})\ \hat{u}(t). \tag{40}$$

The entire collection of bounding and intermediate surfaces are depicted in Fig. 8.

Table 1

| MESH INDEX | r | k | $G_k$ | $G_{k+1}$ | MESH INDEX | r | k | $G_k$ | $G_{k+1}$ |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.00 | 1 | .00 | .00 | 11 | 3.50 | 3 | .88 | .13 |
| 2 | 1.25 | 1 | .44 | .03 | 12 | 3.75 | 3 | .97 | .28 |
| 3 | 1.50 | 1 | .75 | .13 | 13 | 4.00 | 4 | .50 | .00 |
| 4 | 1.75 | 1 | .94 | .28 | 14 | 4.25 | 4 | .72 | .03 |
| 5 | 2.00 | 2 | .50 | .00 | 15 | 4.50 | 4 | .88 | .13 |
| 6 | 2.25 | 2 | .72 | .03 | 16 | 4.75 | 4 | .97 | .28 |
| 7 | 2.50 | 2 | .88 | .13 | 17 | 5.00 | 5 | .50 | .00 |
| 8 | 2.75 | 2 | .97 | .28 | 18 | 5.25 | 5 | .72 | .06 |
| 9 | 3.00 | 3 | .50 | .00 | 19 | 5.50 | 5 | .88 | .25 |
| 10 | 3.25 | 3 | .72 | .03 | 20 | 5.75 | 5 | .97 | .56 |
|  |  |  |  |  | 21 | 6.00 | 5 | 1.00 | 1.00 |

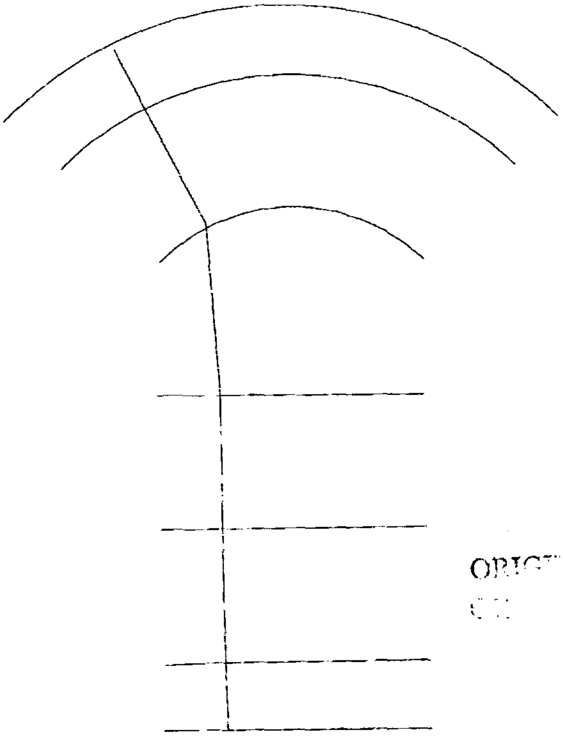

Fig. 8 - Control surfaces for polar-rectangular mesh.

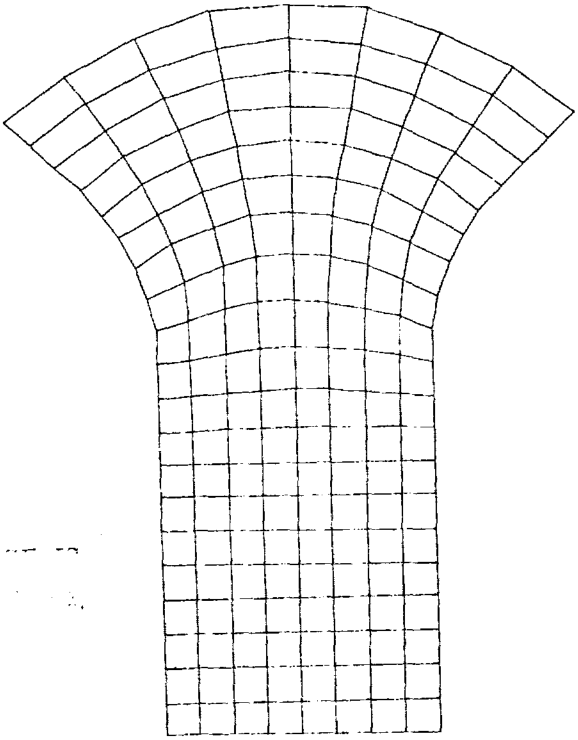

Fig. 9 - Polar-rectangular mesh.

For 21 equally spaced mesh points in r, the evaluation of the r-dependent functions is given with two decimal places of accuracy in the table. For a given mesh point, the interval $k \leq r < k + 1$ containing it determines the index $k$ for $G_k$ and $G_{k+1}$ respectively in Eqs. 22-24. Due to the uniform selection of partition points $r_k$, a repetitive pattern in the $G_k$ evaluations can be observed and is indicative of translated versions of the same function. When 9 uniformly spaced mesh points are chosen for $0 \leq t \leq 1$, and when the multi-surface transformation of Eq. 9 is evaluated for the 21 × 9 mesh, we obtain the coordinate mesh which is displayed in Fig. 9. From uniformity and Table 1, the first 8 and the last 5 mesh points in $r$ are seen to be uniformly distributed, and the mesh is respectively purely Cartesian and purely Polar for those points. To illustrate the computational aspect, we shall explicitly evaluate the transformation at the point with curvilinear variables $r = 4.5$, $t = 0$. At $t = 0$, we have

$$\vec{Q}(0) = (2(0)-1,0) = (-1,0),$$

and (41)

$$\hat{u}(0) = (-\cos\frac{\pi}{4}, \sin\frac{\pi}{4}) = (-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}).$$

For $r = 4.5$, we are at the 15th mesh index in Table 1 where we read across to note that we are in the 4th interval $(4 \leq 4.5 \leq 5)$ with $G_4(4.5) = .88$ and $G_5(4.5) = .13$. By substitution into the transformation (Eq. 25 for $k = 4$) we obtain

$$\vec{P}(4.5,0) = \vec{P}_4(0) + G_4(4.5)[\vec{P}_5(0) - \vec{P}_4(0)] + G_5(4.5)[\vec{P}_6(0) - \vec{P}_5(0)]$$

$$= \vec{Q}(0) + G_4(4.5)[\sqrt{2}\ \hat{u}(0) - \vec{Q}(0)] + G_5(4.5)\ \hat{u}(0)$$

$$= (-1,0) + .88[\sqrt{2}(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}) - (-1,0)] + .13(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$$

$$= (-1,0) + .88(0,1) + (-.09,09) \qquad (42)$$

$$= (-1 + 0 -.09,\ 0 + .88 + .09)$$

$$= (-1.09,\ .97).$$

In continuation with local methods, the case with nonuniform partitions for the piecewise linear functions is given in Eiseman [5]. In addition, local interpolants with a higher level of smoothness (derivative continuity) can be used and are developed in Eiseman [7]. With the local controls over the transverse coordinate curves which connect two bounding surfaces, lateral bounding surfaces can also be approximately fit. A precise fit of the lateral boundaries can be obtained with blending functions which were used by Gordon and Hall [8] to create a global method. Further applications of blending functions will be presented at this workshop by Ericksson [9], by Forsey, Edwards, and Carr [10] and by Anderson and Spradley [11].

## Algebraic Mesh Generation - Three Dimensions

An algebraic approach to mesh generation in three dimensions results in algebraic functions that relate a computational domain to a physical domain. If the computational domain is defined by the three variables $r$, $\xi$, and $\zeta$ on the unit cube

$$0 \leq r \leq 1 \ ,$$

$$0 \leq \xi \leq 1 \ , \tag{43}$$

$$0 \leq \zeta \leq 1,$$

then the physical domain in Cartesian (x,y,z) coordinates is given by the transformation $\vec{P}(r,\xi,\zeta) = (x,y,z)$ where

$$x = x(r,\xi,\zeta) \ ,$$

$$y = y(r,\xi,\zeta) \ , \tag{44}$$

$$z = z(r,\xi,\zeta).$$

When Eq. 44 is nonsingular it has an inverse transformation denoted by

$$r = r(x,y,z) \, ,$$

$$\xi = \xi(x,y,z) \, , \qquad (45)$$

$$\zeta = \zeta(x,y,z) \; .$$

A uniform mesh is defined on the computational domain by constants $\Delta r$, $\Delta\xi$, $\Delta\zeta$ (Fig. 10). This mesh maps using Eq. 44 to a corresponding mesh

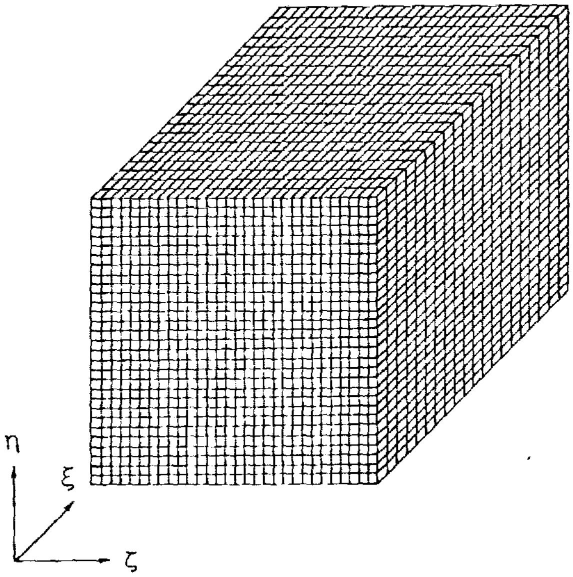

Fig. 10 - Computational domain.

in the physical domain which is not necessarily uniform. A simple example for Eq. 44 is given by

$$x = \left(\frac{e^{k_1\xi} - 1}{e^{k_1} - 1}\right) X_L,$$

$$y = \xi X_L\{\tan \theta_y + Y_L \left(\frac{e^{k_2 r} - 1}{e^{k_2} - 1}\right)\}, \tag{46}$$

$$z = \xi X_L\{\tan \theta_z + Z_L \left(\frac{e^{k_3\zeta} - 1}{e^{k_3} - 1}\right)\},$$

or

$$\xi = \frac{1}{k_1} \ell n\{1 + (e^{k_1} - 1) \frac{x}{X_L}\},$$

$$r = \frac{1}{k_2} \ell n\{1 + (e^{k_2} - 1)(\frac{y}{x} - \tan \theta_y) \frac{1}{Y_L}\}, \tag{47}$$

$$\zeta = \frac{1}{k_3} \ell n\{1 + (e^{k_3} - 1)(\frac{z}{x} - \tan \theta_z) \frac{1}{Z_L}\},$$

where $k_1$, $k_2$, $k_3$, $\theta_y$, $\theta_z$, $X_L$, $Y_L$, and $Z_L$ are constants. For $\xi_0 \leq \xi \leq 1$, $0 \leq r \leq 1$, $0 \leq \zeta \leq 1$ and $Y_L = Z_L$ the uniform computational domain maps into a frustrum of a paramid (Fig. 11) and the mesh is concentrated in the physical domain according to the magnitudes and signs of $k_1$, $k_2$, and $k_3$.

Equation 44 must satisfy the constraints outlined in the introduction and which vary from problem to problem. For many mesh generation problems, the constraints reduce to having the boundaries in the computational domain map to boundaries in the physical

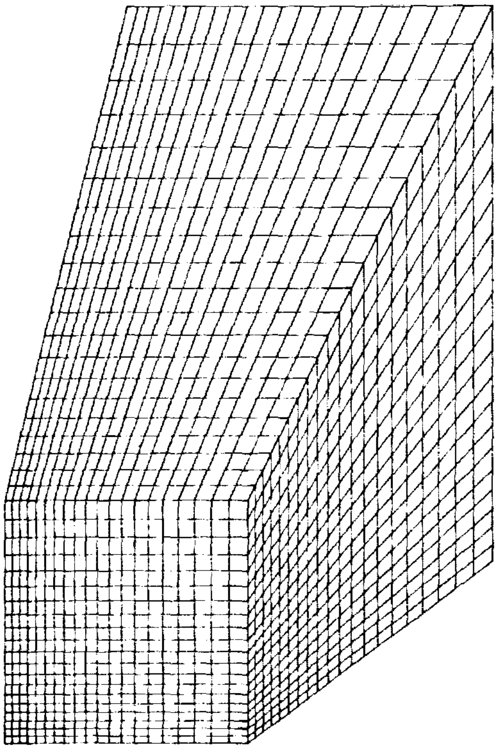

Fig. 11 - Physical domain for Eqs. 46-47.

domain and concentrating the mesh in specified regions of the physical domain. The polynomial N-surface transformations (Eq. 6-18) are global algebraic mesh generation techniques which satisfy the basic boundary constraints and result in polynomial functions of degree $(N - 1)$ with respect to one of the independent variables. For a small $N$ the polynomials are particularly simple. If the surface coordinates are $\vec{t} = (\xi,\zeta)$, the transformation $\vec{P}(r,\vec{t}) = (x(r,\vec{t}),\ y(r,\vec{t}),\ z(r,\vec{t}))$

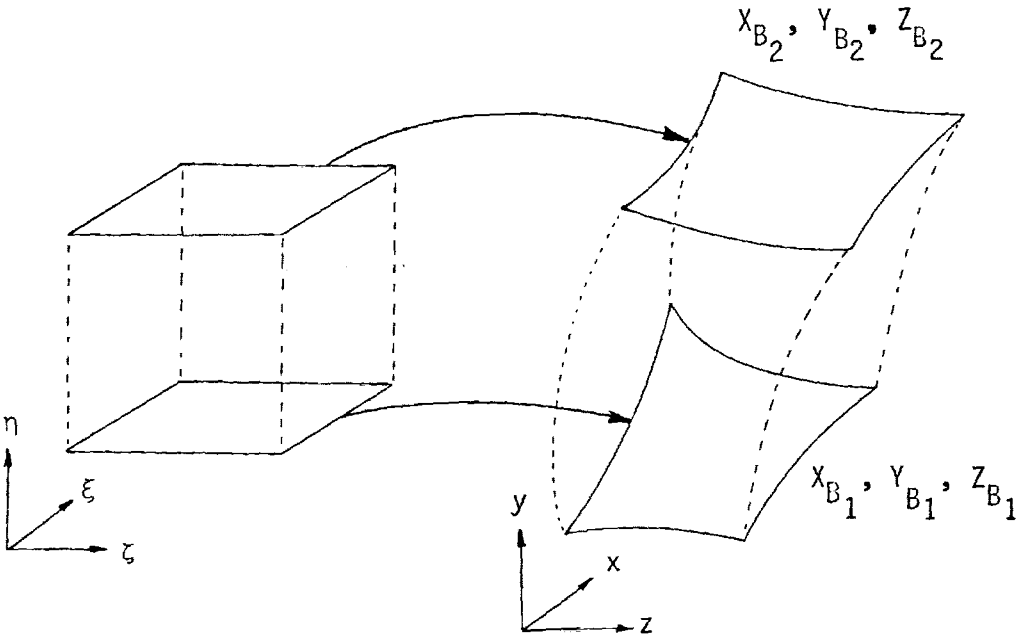
$X_{B_2}$, $Y_{B_2}$, $Z_{B_2}$
$\eta$
$\xi$
$\zeta$
y
x
z
$X_{B_1}$, $Y_{B_1}$, $Z_{B_1}$

Fig. 12 - Boundary mapping.

is defined such that

$$\begin{aligned}
X_{B_1} &= x(0,\xi,\zeta) = X_1(\xi,\zeta), \\
Y_{B_1} &= y(0,\xi,\zeta) = Y_1(\xi,\zeta), \\
Z_{B_1} &= z(0,\xi,\zeta) = Z_1(\xi,\zeta), \\
X_{B_2} &= x(1,\xi,\zeta) = X_2(\xi,\zeta), \\
Y_{B_2} &= y(1,\xi,\zeta) = Y_2(\xi,\zeta), \\
Z_{B_2} &= z(1,\xi,\zeta) = Z_2(\xi,\zeta),
\end{aligned} \tag{48}$$

where $\vec{P}_1(\xi,\zeta) = (X_{B_1}, Y_{B_1}, Z_{B_1})$ is one boundary and $\vec{P}_N(\xi,\zeta) = (X_{B_2}, Y_{B_2}, Z_{B_2})$ is the other boundary in the physical domain (Fig. 12).

The polynomial 4-surface transformation (Eq. 12 and 17) allows a constraint to be placed on the mesh in addition to that of fitting the boundaries. This constraint occurs when the physical mesh is required to be orthogonal at the boundaries. Since the derivatives $\frac{\partial X}{\partial r}(0,\xi,\zeta)$, $\frac{\partial X}{\partial r}(1,\xi,\zeta)$, etc. can be computed from the cross product of the tangential derivatives $\frac{dX_1}{d\xi}(\xi,\zeta)$, $\frac{dX_1}{d\zeta}(\xi,\zeta)$, $\frac{dY_1}{d\xi}(\xi,\zeta)$, $\frac{dY_1}{d\zeta}(\xi,\zeta)$, etc., we have

$$\frac{\partial X}{\partial r}(\ell-1,\xi,\zeta)\,\vec{i} + \frac{\partial Y}{\partial r}(\ell-1,\xi,\zeta)\,\vec{j} + \frac{\partial Z}{\partial r}(\ell-1,\xi,\zeta)\,\vec{k} =$$

$$K\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial X_\ell}{\partial \xi}(\xi,\zeta) & \frac{\partial Y_\ell}{\partial \xi}(\xi,\zeta) & \frac{\partial Z_\ell}{\partial \xi}(\xi,\zeta) \\ \frac{\partial X_\ell}{\partial \zeta}(\xi,\zeta) & \frac{\partial Y_\ell}{\partial \zeta}(\xi,\zeta) & \frac{\partial Z_\ell}{\partial \zeta}(\xi,\zeta) \end{vmatrix} \quad \ell = 1,2 \qquad (49)$$

where $\vec{i}$, $\vec{j}$, and $\vec{k}$ are unit vectors and $K$ is the magnitude of the normal vector, the choice of which can be used to apply controls developed in Eiseman [1] for the shape of coordinate curves in $r$ and for the distribution of constant r-surfaces. Applying this procedure will force the mesh to be orthogonal at the boundaries but not necessarily anywhere else.

A globally uniform computational mesh (for linear $S_p$ in Fig. 6) can be mapped onto a physical mesh with the polynomial N-surface transformations given in Eqs. 6, 9, and 12. Concentration of mesh points in the $r$ direction is accomplished by choosing a function $\bar{r}(r)$ such that

$0 \leq r \leq 1$, $0 \leq \bar{r} \leq 1$ and $\frac{d\bar{r}}{dr} > 0$ (Fig. 13). For example Smith and

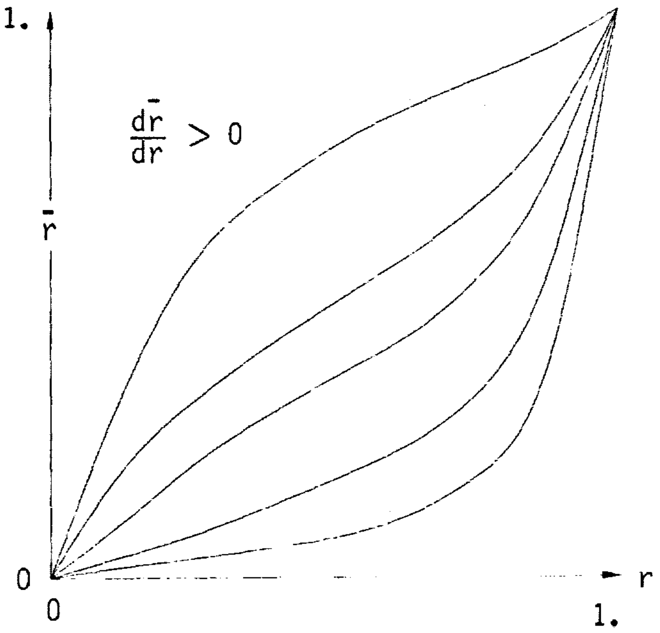

Fig. 13 - Grid control.

Weigel [3] used the function

$$\bar{r} = \frac{e^{kr}-1}{e^{k}-1} \quad ; \quad 0 \leq r \leq 1, \tag{50}$$

to contract the physical grid toward one boundary or the other. The number k is a free parameter whose magnitude dictates the degree of contraction. When r is replaced by $\bar{r}(r)$ in Eq. 6, the contractive function becomes embedded in the linear polynomial part of Eq. 6, which results in

$$\begin{aligned} x &= X_2(\xi,\zeta)\bar{r} + X_1(\xi,\zeta)(1 - \bar{r}), \\ y &= Y_2(\xi,\zeta)\bar{r} + Y_1(\xi,\zeta)(1 - \bar{r}), \\ z &= Z_2(\xi,\zeta)\bar{r} + Z_1(\xi,\zeta)(1 - \bar{r}), \end{aligned} \tag{51}$$

where a uniform partition of $0 \le r \le 1$ yields a desired nonuniform partition of $0 \le \bar{r} \le 1$ that, in turn, proportionately partitions the linear segments of the transformation.

The example previously presented can be derived with this approach where

$$x(\xi) = X_1(\xi,\zeta) = X_2(\xi,\zeta) = \bar{\xi}X_L,$$

$$Y_1(\xi,\zeta) = Y_1(\xi) = \xi X_L \tan \theta_y$$

$$Y_2(\xi,\zeta) = Y_2(\xi) = \xi X_L(\tan \theta_y + Y_L)$$

$$z(\xi,\zeta) = Z_1(\xi,\zeta) = Z_2(\xi,\zeta) = \xi X_L(\tan \theta_z + Z_L)\bar{\zeta} + \xi X_L \tan \theta_z(1 - \bar{\zeta}),$$

$$\bar{\zeta} = \frac{e^{k_3\zeta} -1}{e^{k_3} -1} \quad , \quad \bar{\xi} = \frac{e^{k_1\xi} - 1}{e^{k_1} - 1} \quad , \tag{52}$$

and

$$x = \bar{\xi}X_L,$$

$$y = Y_2(\xi)\bar{r} + Y_1(\xi)(1 - \bar{r}),$$

$$z = z(\xi,\zeta),$$

where

$$\bar{r} = \frac{e^{k_2 r} - 1}{e^{k_2} - 1} \quad .$$

A fundamental constraint of this approach is the representation of the boundaries. The boundaries can be represented as analytical functions or approximate functions based on discrete data from the boundaries. In either case the representation must be in a form where parametric variables which can be normalized to the unit interval are the independent variables. If the parametric independent variables are chosen to be s and t, then for the two boundaries

$$\begin{aligned}
&X_1(\xi,\zeta) \rightarrow X_1(s,t),\\
&Y_1(\xi,\zeta) \rightarrow Y_1(s,t),\\
&Z_1(\xi,\zeta) \rightarrow Z_1(s,t),\\
&X_2(\xi,\zeta) \rightarrow X_2(s,t),\\
&Y_2(\xi,\zeta) \rightarrow Y_2(s,t), \qquad (53)\\
&Z_2(\xi,\zeta) \rightarrow Z_2(s,t),\\
&0 \le \xi \le 1, \quad s_{min} \le s \le s_{max},\\
&0 \le \zeta \le 1, \quad t_{min} \le t \le t_{max}.
\end{aligned}$$

The choice of parametric variables can vary from problem to problem. A relationship between $(\xi,\zeta)$ and $(s,t)$ is

$$s = \xi(s_{max} - s_{min}) + s_{min},$$
$$t = \zeta(t_{max} - t_{min}) + t_{min}. \qquad (54)$$

This is a linear relation which maps the unit interval onto the parametric variables. Contraction of the physical grid at the boundaries is accomplished in the same manner as for the internal grid distribution.

$$\bar{\xi} = \bar{\xi}(\xi), \quad \frac{d\bar{\xi}}{d\xi} > 0,$$
$$\bar{\zeta} = \bar{\zeta}(\zeta), \quad \frac{d\bar{\zeta}}{d\zeta} > 0,$$
$$0 \leq \bar{\xi} \leq 1, \quad 0 \leq \xi \leq 1, \qquad (55)$$
$$0 \leq \bar{\zeta} \leq 1, \quad 0 \leq \zeta \leq 1.$$

## Approximate Boundary-Fitted Coordinate Systems Using Tension Spline Functions

It is often the case that boundaries in a physical domain are described by discrete sets of points. The boundaries may be open or closed. An approximate boundary-fitted coordinate system can be obtained using the technique described and a tension spline function interpolation to the discrete data defining the boundaries. Tension splines are chosen because standard cubic splines and other higher order approximation techniques often result in wiggles in the approximation. Wiggles on a boundary using the technique propagate into the interior grid. The tension parameter embedded in the tension spline approximation to the curve allows control of the "curvedness" of the approximation. A very large magnitude of the tension parameter corresponds to a linear approximation, whereas a very small value corresponds to cubic splines. Tension splines can be applied in two and three dimensions. An example is presented here that is applicable to a two-dimensional mesh.

Using the tension spline technique, a point set on boundary one is defined by $\{x_i, y_i\}_{i=1}^{i=n}$ and on boundary two by $\{x_j, y_j\}_{j=1}^{j=m}$. Approximate arc length is used as a parametric independent variable. The approximate arc length is:

$$s_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2]^{1/2} + s_{i-1},$$

$$s_j = [(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2]^{1/2} + s_{j-1},$$

$$i = 1 \ldots n$$

$$j = 1 \ldots m \tag{56}$$

$$s_1 = 0$$

$$0 \leq s_i \leq s_n$$

$$0 \leq s_j \leq s_m.$$

From the computational coordinate system the unit interval ($0 \leq \xi \leq 1$) must be mapped onto each boundary; that is:

$$\begin{aligned} & s = s(\xi), \\ & 0 \leq \xi \leq 1, \\ & 0 \leq s \leq s_n, \\ & 0 \leq s \leq s_m. \end{aligned} \qquad (57)$$

This is accomplished by letting

$$\begin{aligned} & s = \xi s_n \text{ on boundary one and} \\ & s = \xi s_m \text{ on boundary two.} \end{aligned} \qquad (58)$$

The tension spline functions are piecewise continuous hyperbolic functions on each boundary such that

$$\begin{aligned} x = {} & g''(s_\ell)\,\frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\ & + g''(s_{\ell+1})\,\frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]} \\ & + \left[x_\ell - \frac{g''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right) \\ & + \left[x_{\ell+1} - \frac{g''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right) \quad , \end{aligned} \qquad (59a)$$

$$y = h''(s_\ell) \frac{\sinh[\sigma(s_{\ell+1} - s)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]}$$
$$+ h''(s_{\ell+1}) \frac{\sinh[\sigma(s - s_\ell)]}{\sigma^2 \sinh[\sigma(s_{\ell+1} - s_\ell)]}$$
$$+ \left[y_\ell - \frac{h''(s_\ell)}{\sigma^2}\right]\left(\frac{s_{\ell+1} - s}{s_{\ell+1} - s_\ell}\right)$$
$$+ \left[y_{\ell+1} - \frac{h''(s_{\ell+1})}{\sigma^2}\right]\left(\frac{s - s_\ell}{s_{\ell+1} - s_\ell}\right) , \qquad (59b)$$

$\ell = i$ on boundary one,

$\ell = j$ on boundary two,

$\sigma$ = tension parameter.

The unknowns in these equations are $g''(s_\ell)$ and $h''(s_\ell)$ which are second derivatives at the data points $\{x_\ell, y_\ell\}_{\ell=1}^{\ell=L}$ and are obtained through enforcement of the continuity of the first derivatives at the data points, and the specification of two end conditions. A tridiagonal system of linear equations results for each set of unknowns. The solutions of the tridiagonal systems yield $g''(s_\ell)$ and $h''(s_\ell)$.

A cubic polynomial and the contracting function $\bar{r} = \frac{e^{kr} - 1}{e^k - 1}$ provide the relationship between the computational domain and physical domain. The derivatives $\frac{dX_\ell}{d\eta}$ and $\frac{dY_\ell}{d\eta}$ are:

$$\frac{dX_\ell}{d\eta} = K \frac{dY_\ell}{ds} ,$$
$$(60)$$
$$\frac{dY_\ell}{d\eta} = -K \frac{dX_\ell}{ds} .$$

By defining a grid with constants $\Delta\xi$ and $\Delta r$ in the computational domain a corresponding grid is explicitly defined in the physical domain.

An example of an airfoil grid is presented (Fig. 14). Data points on each boundary, magnitude of the normal derivative, and contract parameter values define the grids. Boundary data for the airfoil is shown in Fig. 15. Also, for closed boundaries such as the airfoil, periodic conditions are applied in the spline functions.

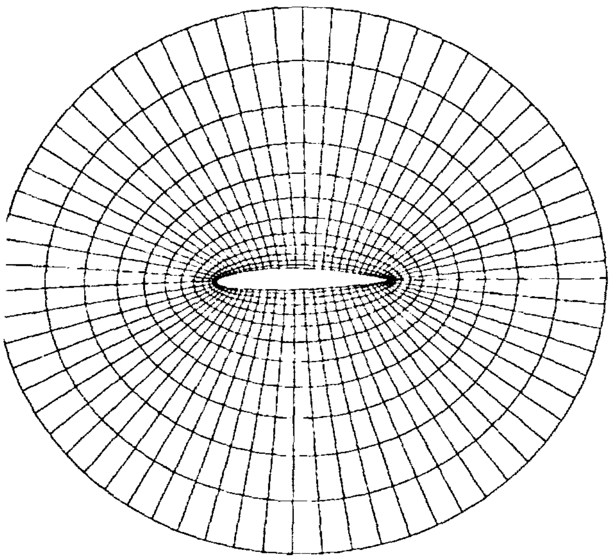

Fig. 14 - Mesh for Kármán Trefftz airfoil using splines under tension.

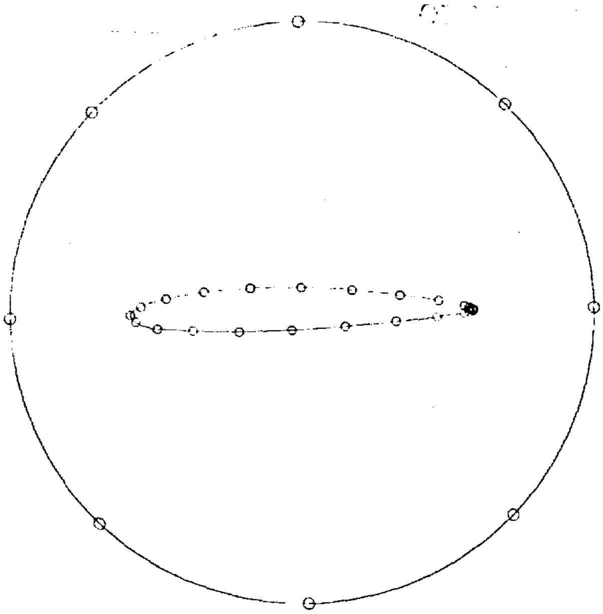

Fig. 15 - Data definition for boundaries of Kármán Trefftz airfoil domain.

## Complex Three-Dimensional Mesh Generation

The development of three-dimensional meshes where mesh lines are free to move in all three coordinate directions is extremely difficult. The reader is directed to reference 9 for examples of such unconstrained three-dimensional meshes. An expedient approach for complex three-dimensional geometries is to attempt to simplify the problem by rendering the three-dimensional geometry quasi two-dimensional. This is the essence of the frustrum pyramid mesh previously presented. Also when there is axis-symmetry in a problem, two-dimensional rendering of the geometry is possible. This is demonstrated with the spike-nosed configuration shown in Fig. 16. Figure 17 shows an algebraic generated mesh for one plane of the geometry. The mesh is made three dimensional by rotating the mesh in movements about the axis of symmetry.

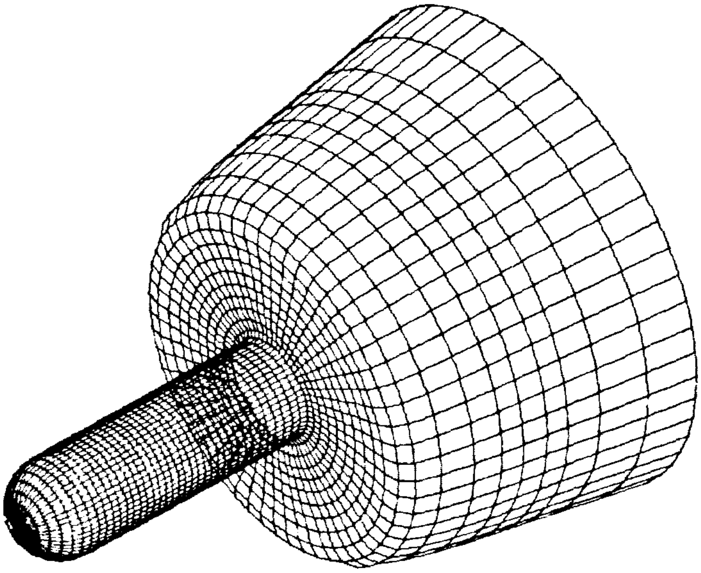

Fig. 16 - Surface for a spike-nosed body.

For aircraft surfaces the problem is more difficult. There are, however, several good software packages ([12] is a good example) for surface definition of aircraft aerospacecraft geometries. In [12]

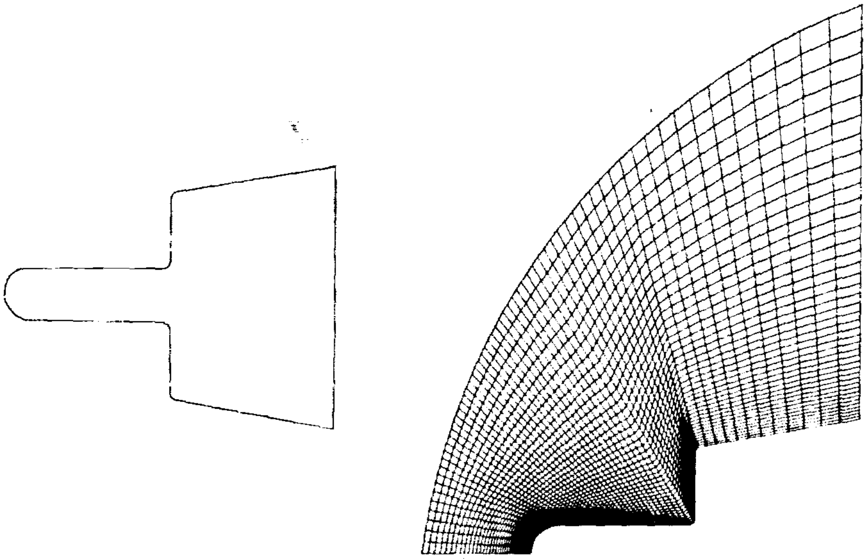

Fig. 17 - Mesh for spike-nosed body.

the Harris geometry is used to establish coordinates for Coon's surface definition of a configuration. Spline functions are used to compute the derivatives for the Coon's surface definition. Figure 18 shows the input description for a wing-fuselage configuration where the wing and fuselage are defined separately. Figure 19 shows an enriched definition of the configuration using the Coon's surfaces. A part of the available software described in [12] is the ability to compute the intersection of an arbitrarily defined plane and the surface definition. For the configuration shown in Fig. 19 planes perpendicular to the $x$ axis and at a constant increment in the $x$ direction are shown in Fig. 20. If an outside boundary is defined the corresponding intersection with the

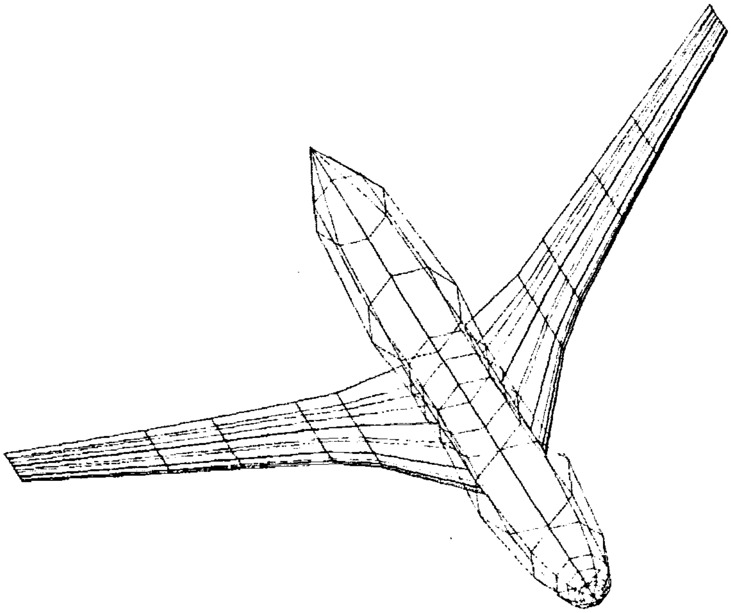

Fig. 18 - Data definition for a wing-fuselage configuration.

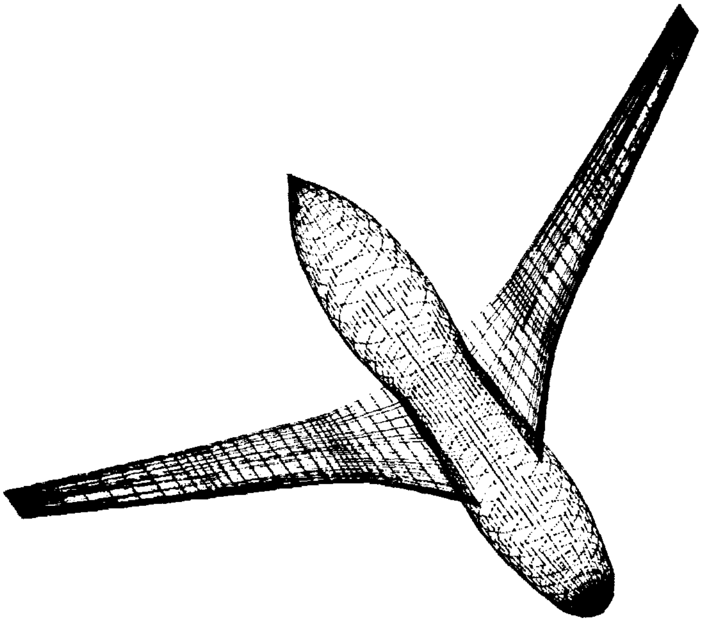

Fig. 19 - Enriched surface definition for a wing-fuselage configuration.

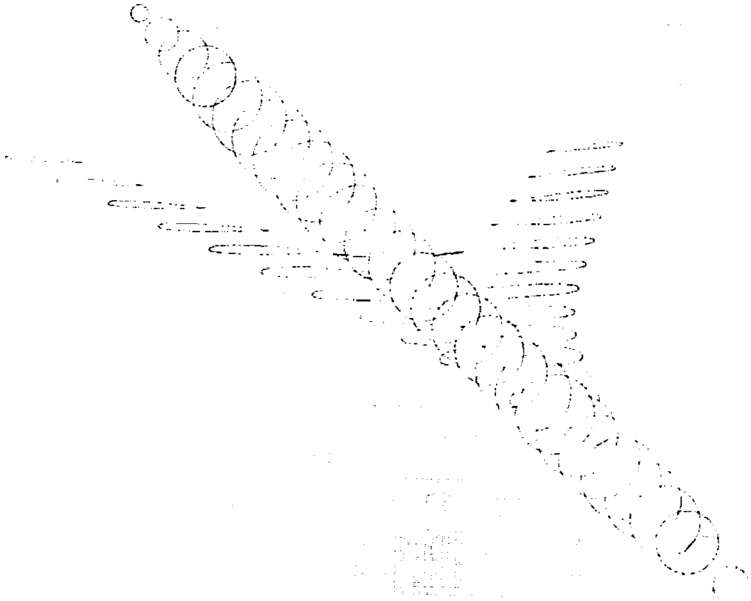

Fig. 20 - Planar intersections with a wing-fuselage configuration.

planes is computable. With an inside and outside boundary the techniques previously described can be employed in two dimensions. Complexities of multiconnected regions is introduced but this can be attacked with branch cuts (Fig. 21).

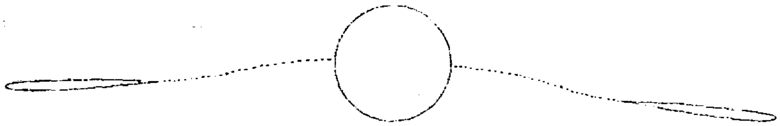

Fig. 21 - Branch cuts for multi-connected sections.

Another attack on complex three dimensional problems is to define several computational domains (Fig. 22) with mutual boundaries. The mapping function transforms the computational domains to parts of the physical domain with mutual intersections.

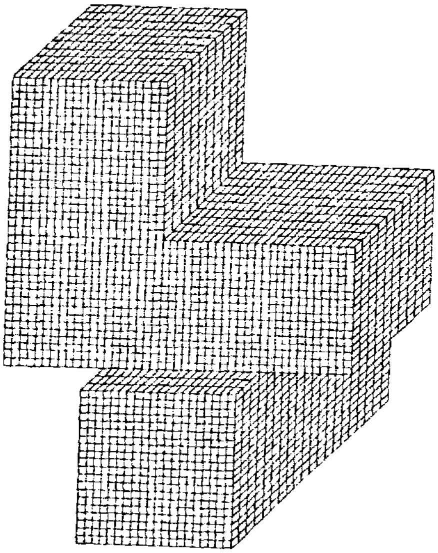

Fig. 22 - Multiple computational domains with mutual boundaries.

## Conclusions

Algebraic methods provide precise controls for mesh generation. Methodologies for mesh construction can be based on a parameterized description of surfaces which consist of bounding surfaces and intermediate control surfaces. The surface locations determined by the respective surface parameterizations determine the nature of the transverse coordinate curves which connect the bounding surfaces. Relative to uniform conditions, precise control over the mesh placement in the physical domain can be accomplished by embedding distribution control functions in the surface parameterizations or in the transverse direction. Complex bounding topologies, especially in three dimensions, cause mesh construction difficulties. It is proposed that whenever feasible, the complex topology be simplified such as rendering the geometry quasi two-dimensional. As a final remark, precise controls are one of the major advantages of algebraic methods: they give the capability to prescribe specific desirable and helpful mesh formations.

## References

1. Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," J. Comp. Phys. 33 (1979), pp. 118-150.

2. Eiseman, P. R., "A Coordinate System for a Viscous Transonic Cascade Analysis," J. Comp. Phys. 26 (1978), pp. 307-338.

3. Smith, R. E. and Weigel, B. L., "Analytical and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation," AIAA Paper 80-0192 (1980).

4. Moretti, G. and Abbett, "A Time-Dependent Computational Method for Blunt Body Flows," AIAA J., Vol. 4, No. 12 (1966) pp. 2136-2141.

5. Eiseman, P. R., "Geometric Methods in Computational Fluid Dynamics," von Karman Institute Lecture Series and ICASE Report 80-11 (1980).

6. Eiseman, P. R., "Coordinate Generation with Precise Controls," Seventh International Conference on Numerical Methods in Fluid Dynamics, held at Stanford Univ. and NASA Ames in June 1980.

7. Eiseman, P. R., "Coordinate Generation with Precise Controls Over Mesh Properties." ICASE Report 80-30 (1980).

8. Gordon, W. J. and Hall, C. A., "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," Intl. J. Num. Meth. in Engr. 7 (1973), pp. 461-477.

9. Eriksson, L., "Three-Dimensional Spline-Generated Coordinate Transformations for Grids Around Wing-Body Configurations." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 17 of this compilation.)

10. Forsey, C. R., Edwards, M. G., and Carr, M. P., "An Investigation Into Grid Patching Techniques." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 18 of this compilation.)

11. Anderson, P. G., Spradley, L. W., "Finite Difference Grid Generation by Multivariate Blending Function Interpolation." Numerical Grid Generation Techniques, NASA CP-2166, 1980. (Paper 6 of this compilation.)

12. Craidon, C. B., "A Computer Program for Fitting Smooth Surfaces to an Aircraft Configuration and Other Three-Dimensional Geometries," NASA TMX-3206, June 1975.