INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 15, 297–311 (1992)

ENHANCING CONTROL OF GRID DISTRIBUTION IN ALGEBRAIC GRID GENERATION

E. STEINTHORSSON AND T. I-P. SHIH

Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213-3890, U.S.A.

AND

R. J. ROELKE

National Aeronautics and Space Administration, Lewis Research Center, Cleveland, OH 44135, U.S.A.

SUMMARY

Three techniques are presented to enhance the control of grid-point distribution for a class of algebraic grid generation methods known as the two-, four- and six-boundary methods. First, multidimensional stretching functions are presented, and a technique is devised to construct them based on the desired distribution of grid points along certain boundaries. Second, a normalization procedure is proposed which allows more effective control over orthogonality of grid lines at boundaries and curvature of grid lines near boundaries. And third, interpolating functions based on tension splines are introduced to control curvature of grid lines in the interior of the spatial domain. In addition to these three techniques, consistency conditions are derived which must be satisfied by all user-specified data employed in the grid generation process to control grid-point distribution. The usefulness of the techniques developed in this study was demonstrated by using them in conjunction with the two- and four-boundary methods to generate several grid systems, including a three-dimensional grid system in the coolant passage of a radial turbine blade with serpentine channels and pin fins.

KEY WORDS Grid generation Transfinite interpolation Stretching functions Blending functions

INTRODUCTION

Even though considerable progress has been made in grid generation, the generation of good quality structured grid systems in geometrically complex, three-dimensional spatial domains remains a difficult problem. This is because, for such domains (e.g. see Figure 1), grid lines or surfaces may need to make many sharp twists and turns in the interior of the domain in addition to being clustered at the right places, and made nearly orthogonal and as smooth as possible. In order to generate grid systems for such domains, the grid generation method must be able to exert precise control over how grid points are to be distributed.

Of the grid generation methods available, control over grid-point distribution is either indirect (and hence not transparent to the user) or direct but inadequate. In elliptic grid generation methods, grid-point distribution is controlled indirectly through source terms in the partial differential equations.[1] Even though these source terms are well defined, they involve user specified parameters that must be fine-tuned for each problem. For three-dimensional geometries, the fine tuning of these parameters can be costly, since these methods require the solution of quasi-linear systems of partial differential equations. In algebraic grid generation methods, the

0271–2091/92/150297–15$12.50

Received May 1991
Revised January 1992

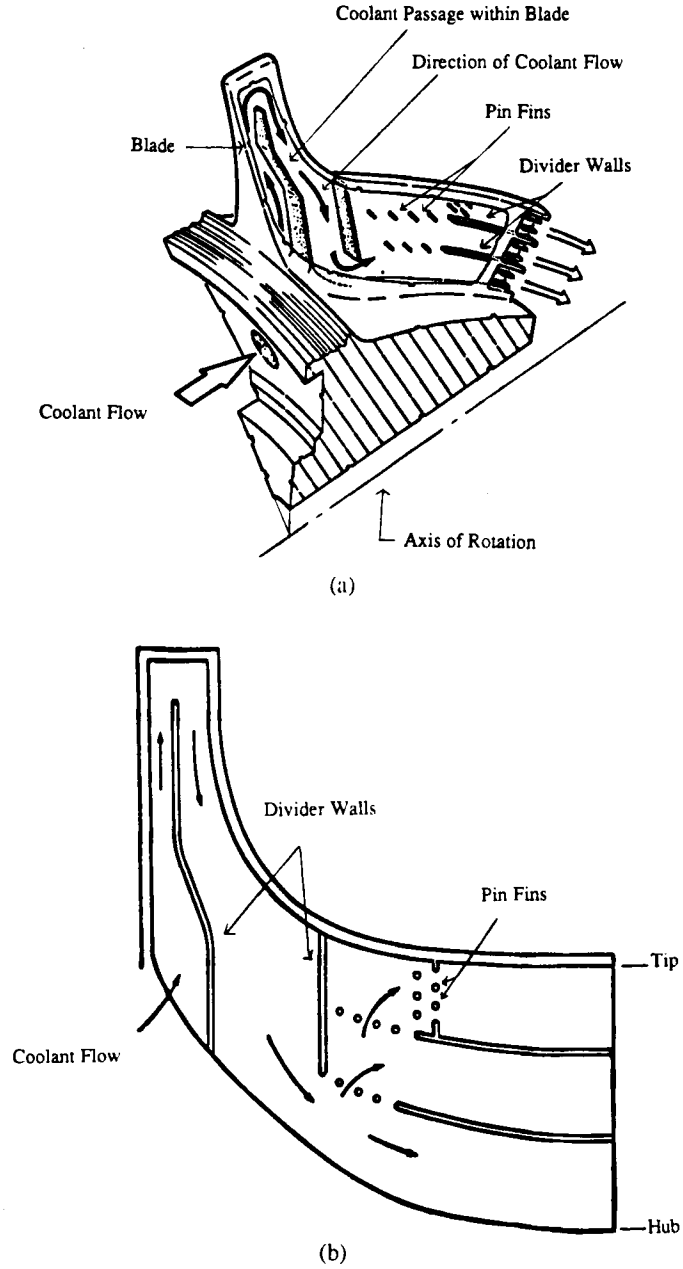

Figure 1. Coolant passage within a radial turbine blade: (a) cutout view; (b) schematic drawing

control over grid-point distribution is more direct (e.g. through the use of stretching functions). Although these methods also involve parameters that need to be fine-tuned for each problem, the fine tuning can be done very efficiently since algebraic methods do not require the solution of partial differential equations. The drawback of algebraic methods is that the control needed to obtain acceptable grid systems is often inadequate.

In this paper, we present several techniques that enhance the control of grid-point distribution for a class of algebraic grid generation methods based on transfinite interpolation which are known as the two-, four- and six-boundary methods.[2] First, multidimensional stretching functions are proposed and a technique is devised to construct them based on the desired grid-point distribution along certain boundaries. Second, a normalization procedure is proposed which enables more effective control over orthogonality of grid lines at boundaries and curvature of grid lines near boundaries. Third, interpolating functions based on tension splines are introduced to control the curvature of grid lines in the interior of the spatial domain. Finally, a set of consistency conditions is derived that must be satisfied by all user-specified data employed in the grid generation process to control grid-point distribution. The usefulness of the techniques presented in this paper was demonstrated by applying them to generate several grid systems including a three-dimensional grid system for the coolant passage of a radial turbine blade with serpentine channels and pin fins.

In the following sections, the three techniques for enhancing control of grid-point distribution are described along with examples of grid systems generated using these techniques. The consistency conditions formulated in this study are given in the Appendix.

This paper assumes that the reader is familiar with algebraic grid generation methods based on transfinite interpolation. Readers unfamiliar with these methods are referred to References 1 and 2.

MULTIDIMENSIONAL STRETCHING FUNCTIONS

In algebraic grid generation, stretching functions are important for controlling the distribution of grid points in the physical domain. To date, all stretching functions have been one-dimensional.[1-3] The use of one-dimensional stretching functions has the major limitation that all grid lines in the same direction have the same amount of stretching. For many complex-shaped geometries, these stretching functions do not provide the control that is needed to obtain acceptable grid systems.

To improve the control over grid-point distribution in algebraic grid generation, this study proposes the construction of multidimensional stretching functions to replace the one-dimensional stretching functions. Multidimensional stretching functions will allow the stretching to vary from one grid line to the next. Since grid systems generated by algebraic methods are strongly influenced by the grid-point distribution on the boundaries of the spatial domain, we present here a method for constructing multidimensional stretching functions based on the desired grid-point distribution on the edges of the spatial domain. As used here, the edges are the four plane or twisted curves that define the boundary of a surface.

To illustrate the method devised here for constructing multidimensional stretching functions, we construct a stretching function for the ξ-grid lines in a three-dimensional grid system. In this grid system, x, y and z are the co-ordinates in the physical domain; and ξ, η, and ζ are the co-ordinates in the transformed domain. The transformed domain is taken as the region in which the co-ordinates ξ, η and ζ vary between zero and one. For the ξ-grid lines, the edges which affect grid-point distribution are the ones located at $(\eta=0, \zeta=0)$, $(\eta=1, \zeta=0)$, $(\eta=0, \zeta=1)$ and $(\eta=1, \zeta=1)$. With the relevant edges identified, suppose $\bar{\xi}_{00}(\xi)$, $\bar{\xi}_{10}(\xi)$, $\bar{\xi}_{01}(\xi)$, and $\bar{\xi}_{11}(\xi)$ are the one-dimensional stretching functions which give the desired grid-point distribution along the edges at $(\eta=0, \zeta=0)$, $(\eta=1, \zeta=0)$, $(\eta=0, \zeta=1)$ and $(\eta=1, \zeta=1)$, respectively. Since each edge is a line and hence one-dimensional, these stretching functions can easily be obtained. Once the one-dimensional stretching functions for the relevant edges have been determined, the multidimensional stretching function for the ξ-grid lines in the entire domain is constructed by interpolation of the

one-dimensional stretching functions at the edges. If linear interpolation is used, then we obtain the following multidimensional stretching function:

$$\hat{\xi}(\xi, \eta, \zeta)=[\hat{\xi}_{00}(\xi)(1-\eta)+\hat{\xi}_{10}(\xi)\eta](1-\zeta)+[\hat{\xi}_{01}(\xi)(1-\eta)+\hat{\xi}_{11}(\xi)\eta]\zeta. \tag{1}$$

If cubic polynomials as in Hermite interpolation are used, then we obtain

$$\hat{\xi}(\xi, \eta, \zeta)=[\hat{\xi}_{00}(\xi)h_1(\eta)+\hat{\xi}_{10}(\xi)h_2(\eta)]h_1(\zeta)+[\hat{\xi}_{01}(\xi)h_1(\eta)+\hat{\xi}_{11}(\xi)h_2(\eta)]h_2(\zeta), \tag{2}$$

where

$$h_1(s)=2s^3-3s^2+1, \qquad h_2(s)=-2s^3+3s^2. \tag{3a, b}$$

The flexibility of the approach described above for constructing multidimensional stretching functions can be increased further if one realizes that it is unnecessary to work with analytical stretching functions for the edge. This is because we only need the values of the stretching functions at the grid points on the edges. Thus, one can, for example, specify the location of the grid points along an edge directly, then use arc length to calculate an equivalent stretching function that corresponds to the distribution of grid points along the edge, and finally, use the equivalent stretching function to construct a multidimensional stretching function as shown above.

As an example of how an equivalent stretching function can be obtained, suppose (x_i, y_i, z_i) with $i=1, \ldots, IL$ are the specified grid-point locations on the edge located at $(\xi=0, \eta=0)$, and we want to construct an equivalent stretching function for that edge. In this case, the required stretching function $\hat{\xi}_{00}$ can be constructed from the specified grid-point distribution by using approximate arc length as follows:

$$\hat{\xi}_{00}(\xi_i)=0 \qquad \text{for } i=1, \tag{4a}$$

$$\hat{\xi}_{00}(\xi_i)=\frac{d_i}{d_{IL}} \qquad \text{for } i=2, 3, 4, \ldots, IL, \tag{4b}$$

where

$$d_i=\sum_{n=2}^{i}[(x_n-x_{n-1})^2+(y_n-y_{n-1})^2+(z_n-z_{n-1})^2]^{1/2}. \tag{4c}$$

The approach described above for deriving the multidimensional stretching function for the ξ-grid lines [i.e. $\hat{\xi}(\xi, \eta, \zeta)$] can also be applied to derive the multidimensional stretching functions for the η- and ζ-grid lines [i.e. $\hat{\eta}(\xi, \eta, \zeta)$ and $\hat{\zeta}(\xi, \eta, \zeta)$]. The approach used to compute $\hat{\xi}_{00}$ from arc length can also be applied to compute equivalent one-dimensional stretching functions for any edge or grid line.

Above, we have illustrated how multidimensional stretching functions can be constructed from one-dimensional stretching functions at the edges of a three-dimensional grid system. This approach can be extended to include one or more one-dimensional stretching functions for grid lines that lie in between the edges or in the interior of the grid system in order to further increase control over grid-point distribution throughout the domain. An approach different from the one described above is to construct multidimensional stretching functions by applying a single one-dimensional stretching function to all grid lines but vary the parameters that control the location and amount of clustering from grid line to grid line. Stretching functions such as those developed by Vinokur[3] can be implemented in this fashion.

The multidimensional stretching functions introduced above can be used with two-, four- and six-boundary methods. Below, we illustrate how they can be used with the four-boundary method via the following six steps (see Reference 2 for details of the four-boundary method):

1. Obtain parametric description of the four boundary surfaces of the spatial domain that are to be mapped correctly; e.g.

$$\mathbf{r}_1(\eta,\zeta) \quad \text{at } \xi=0, \qquad \mathbf{r}_2(\eta,\zeta) \quad \text{at } \xi=1,$$
$$\mathbf{r}_3(\xi,\zeta) \quad \text{at } \eta=0, \qquad \mathbf{r}_4(\xi,\zeta) \quad \text{at } \eta=1.$$

2. Select stretching functions that give the desired grid-point distribution along the edges of the spatial domain; i.e. select the functions $\hat{\xi}_{00}$, $\hat{\xi}_{10}$, $\hat{\xi}_{01}$ and $\hat{\xi}_{11}$ in equations (1) and (2), and similar functions for the η- and ζ-co-ordinate directions.
3. Construct a multidimensional stretching function $\hat{\xi}(\xi,\eta,\zeta)$ for the ξ-grid lines by using equation (1) or (2). Similarly, construct a multidimensional stretching function $\hat{\eta}(\xi,\eta,\zeta)$ for the η-grid lines and $\hat{\zeta}(\xi,\eta,\zeta)$ for the ζ-grid lines.
4. Define new parametric descriptions of the boundary surfaces as follows:

$$\tilde{\mathbf{r}}_1(\eta,\zeta)=\mathbf{r}_1[\hat{\eta}(\xi=0,\eta,\zeta),\hat{\zeta}(\xi=0,\eta,\zeta)], \tag{5a}$$
$$\tilde{\mathbf{r}}_2(\eta,\zeta)=\mathbf{r}_2[\hat{\eta}(\xi=1,\eta,\zeta),\hat{\zeta}(\xi=1,\eta,\zeta)], \tag{5b}$$
$$\tilde{\mathbf{r}}_3(\xi,\zeta)=\mathbf{r}_3[\hat{\xi}(\xi,\eta=0,\zeta),\hat{\zeta}(\xi,\eta=0,\zeta)], \tag{5c}$$
$$\tilde{\mathbf{r}}_4(\xi,\zeta)=\mathbf{r}_4[\hat{\xi}(\xi,\eta=1,\zeta),\hat{\zeta}(\xi,\eta=1,\zeta)]. \tag{5d}$$

5. Select $\partial\mathbf{r}(\xi=0,\eta,\zeta)/\partial\xi$, $\partial\mathbf{r}(\xi=1,\eta,\zeta)/\partial\xi$, $\partial\mathbf{r}(\xi,\eta=0,\zeta)/\partial\eta$, and $\partial\mathbf{r}(\xi,\eta=1,\zeta)/\partial\eta$, such that they are orthogonal to $\tilde{\mathbf{r}}_1(\eta,\zeta)$, $\tilde{\mathbf{r}}_2(\eta,\zeta)$, $\tilde{\mathbf{r}}_3(\xi,\zeta)$ and $\tilde{\mathbf{r}}_4(\xi,\zeta)$, respectively (except near the edges as discussed in the Appendix).
6. Generate the grid points in the physical domain as follows:

$$\begin{pmatrix} x_{ijk} \\ y_{ijk} \\ z_{ijk} \end{pmatrix} = \mathbf{r}(\xi_i,\eta_j,\zeta_k), \tag{6}$$

where

$$\mathbf{r}(\xi,\eta,\zeta)=\mathbf{r}'(\xi,\eta,\zeta)+\Delta\mathbf{r}(\xi,\eta,\zeta), \tag{7}$$

$$\mathbf{r}'(\xi,\eta,\zeta)=\tilde{\mathbf{r}}_1(\eta,\zeta)h_1(\hat{\xi})+\tilde{\mathbf{r}}_2(\eta,\zeta)h_2(\hat{\xi})+\frac{\partial\mathbf{r}(\xi=0,\eta,\zeta)}{\partial\xi}h_3(\hat{\xi})+\frac{\partial\mathbf{r}(\xi=1,\eta,\zeta)}{\partial\xi}h_4(\hat{\xi}), \tag{8}$$

$$\begin{aligned}\Delta\mathbf{r}(\xi,\eta,\zeta)=&[\tilde{\mathbf{r}}_3(\eta,\zeta)-\mathbf{r}'(\xi,\eta=0,\zeta)]h_1(\hat{\eta})+[\tilde{\mathbf{r}}_4(\xi,\zeta)-\mathbf{r}'(\xi,\eta=1,\zeta)]h_2(\hat{\eta})\\&+\left[\frac{\partial\mathbf{r}(\xi,\eta=0,\zeta)}{\partial\eta}-\frac{\partial\mathbf{r}'(\xi,\eta=0,\zeta)}{\partial\eta}\right]h_3(\hat{\eta})+\left[\frac{\partial\mathbf{r}(\xi,\eta=1,\zeta)}{\partial\eta}-\frac{\partial\mathbf{r}'(\xi,\eta=1,\zeta)}{\partial\eta}\right]h_4(\hat{\eta}),\end{aligned} \tag{9}$$

$$\xi_i=\frac{i-1}{IL-1}, \qquad i=1,\ldots,IL, \tag{10a}$$
$$\eta_i=\frac{j-1}{JL-1}, \qquad j=1,\ldots,JL, \tag{10b}$$
$$\zeta_k=\frac{k-1}{KL-1}, \qquad k=1,\ldots,KL. \tag{10c}$$

In the above equation, IL, JL and KL are the number of grid points in the ξ-, η- and ζ-direction, respectively.

Grid generation with multidimensional stretching functions as described above offers significantly more control over how grid points can be distributed than with one-dimensional stretching functions. This increased control comes with only a minor increase in complexity.

SPECIFICATION OF DERIVATIVES AT THE BOUNDARIES

When the two-, four- and six-boundary methods are used to generate grid systems, first-order derivatives at boundaries [e.g. $\partial \mathbf{r}(\xi=0,\eta,\zeta)/\partial\xi$ and $\partial \mathbf{r}(\xi,\eta=0,\zeta)/\partial\eta$ in equations (8) and (9)] must be specified. Since grid lines should be orthogonal at boundaries, previous investigators have specified these derivatives as follows:

$$\frac{\partial \mathbf{r}(\xi=0,\eta,\zeta)}{\partial\xi}=K_1(\eta,\zeta)\mathbf{t}_1(\eta,\zeta), \tag{11a}$$

$$\frac{\partial \mathbf{r}(\xi,\eta=0,\zeta)}{\partial\eta}=K_3(\xi,\zeta)\mathbf{t}_3(\xi,\zeta), \tag{11b}$$

where

$$\mathbf{t}_1(\eta,\zeta)=\frac{\partial \mathbf{r}_1}{\partial\eta}\times\frac{\partial \mathbf{r}_1}{\partial\zeta}, \tag{12a}$$

$$\mathbf{t}_3(\xi,\zeta)=-\frac{\partial \mathbf{r}_3}{\partial\xi}\times\frac{\partial \mathbf{r}_3}{\partial\zeta}. \tag{12b}$$

In equation (11), the factors $K_1(\eta,\zeta)$ and $K_3(\xi,\zeta)$—henceforth referred to as K-factors—are specified by the user. The K-factors are intended to control the magnitudes of the first-order derivatives which in turn control the curvature of the grid lines near boundary surfaces. However, the magnitudes of these derivatives depend not only on the K-factors but also on the magnitudes of the vectors $\mathbf{t}_1$ and $\mathbf{t}_3$, which in turn depend on the geometry of the boundary surfaces at $\xi=0$ and $\eta=0$ and on the grid spacings on those surfaces. Therefore, for complex geometries, the effectiveness of K-factors as control parameters is greatly reduced.

In order to enhance the effectiveness of the K-factors as control parameters, an alternative way is proposed for specifying the derivative terms $\partial \mathbf{r}(\xi=0,\eta,\zeta)/\partial\xi$ and $\partial \mathbf{r}(\xi,\eta=0,\zeta)/\partial\eta$. This alternative approach ensures that the K-factors alone determine the magnitude of the first-order derivatives. This is achieved by normalizing the vectors orthogonal to the boundary surfaces; i.e. the derivatives are specified as

$$\frac{\partial \mathbf{r}(\xi=0,\eta,\zeta)}{\partial\xi}=K_1(\eta,\zeta)\mathbf{e}_1(\eta,\zeta), \tag{13a}$$

$$\frac{\partial \mathbf{r}(\xi,\eta=0,\zeta)}{\partial\eta}=K_3(\xi,\zeta)\mathbf{e}_3(\xi,\zeta), \tag{13b}$$

where $\mathbf{e}_1$ and $\mathbf{e}_3$ are unit vectors orthogonal to the boundary surfaces at $\xi=0$ and $\eta=0$, respectively. These normalized vectors are given by

$$\mathbf{e}_1=\frac{\mathbf{t}_1}{|\mathbf{t}_1|}, \tag{14a}$$

$$\mathbf{e}_3=\frac{\mathbf{t}_3}{|\mathbf{t}_3|}, \tag{14b}$$

where $\mathbf{t}_1$ and $\mathbf{t}_3$ are given by equation (12).

At this point, it is important to note that while the specification of the first-order derivatives is relatively straightforward for the two-boundary method, special attention is required for the four- and six-boundary methods. This is true in particular if the boundary surfaces in the physical domain intersect non-orthogonally. In such cases, if proper care is exercised when the first-order derivatives are specified, then the grid can be made orthogonal at the boundary surfaces except in narrow regions next to the edges where boundary surfaces intersect non-orthogonally. If proper care is not exercised, the orthogonality may not be achieved anywhere on the boundary surface, even in regions far away from the edges. In order to achieve orthogonality away from the edges, the first-order derivatives that are specified must satisfy certain geometrically imposed consistency conditions that ensure continuity of these derivatives at the edges. The set of consistency conditions for the four-boundary method is derived in the Appendix, and a similar set of conditions can be derived for the six-boundary method.

To illustrate how first-order derivatives should be specified to ensure orthogonality, consider again the terms $\partial \mathbf{r}(\xi=0, \eta, \zeta)/\partial \xi$ and $\partial \mathbf{r}(\xi, \eta=0, \zeta)/\partial \eta$ in equations (8) and (9). When the four-boundary method is used, these derivatives must be chosen such that the geometrically imposed consistency conditions given by equations (26), (27), (29) and (30) in the Appendix are satisfied. This can be accomplished by specifying the first-order derivative terms as follows:

$$\frac{\partial \mathbf{r}(\xi=0, \eta, \zeta)}{\partial \xi} = \begin{cases} \dfrac{\partial \mathbf{r}_3(\xi=0, \zeta)}{\partial \xi} h_1(\eta/\eta_1) + K_1(\eta_1, \zeta)\mathbf{e}_1(\eta_1, \zeta) h_2(\eta/\eta_1) & \text{for } 0 \leqslant \eta \leqslant \eta_1, \\ K_1(\eta, \zeta)\mathbf{e}_1(\eta, \zeta) & \text{for } \eta_1 \leqslant \eta \leqslant \eta_2, \\ K_1(\eta_2, \zeta)\mathbf{e}_1(\eta_2, \zeta) h_1\left(\dfrac{\eta-\eta_2}{1-\eta_2}\right) + \dfrac{\partial \mathbf{r}_4(\xi=0, \zeta)}{\partial \xi} h_2\left(\dfrac{\eta-\eta_2}{1-\eta_2}\right) & \text{for } \eta_2 \leqslant \eta \leqslant 1, \end{cases} \tag{15}$$

$$\frac{\partial \mathbf{r}(\xi, \eta=0, \zeta)}{\partial \eta} = \begin{cases} \dfrac{\partial \mathbf{r}_1(\eta=0, \zeta)}{\partial \eta} h_1(\xi/\xi_1) + K_3(\xi_1, \zeta)\mathbf{e}_3(\xi_1, \zeta) h_2(\xi/\xi_1) & \text{for } 0 \leqslant \xi \leqslant \xi_1, \\ K_3(\xi, \zeta)\mathbf{e}_3(\xi, \zeta) & \text{for } \xi_1 \leqslant \xi \leqslant \xi_2, \\ K_3(\xi_2, \zeta)\mathbf{e}_3(\xi_2, \zeta) h_1\left(\dfrac{\xi-\xi_2}{1-\xi_2}\right) + \dfrac{\partial \mathbf{r}_2(\eta=0, \zeta)}{\partial \eta} h_2\left(\dfrac{\xi-\xi_2}{1-\xi_2}\right) & \text{for } \xi_2 \leqslant \xi \leqslant 1, \end{cases} \tag{16}$$

where h_1 and h_2 are given by equation (3), and $\mathbf{e}_1$ and $\mathbf{e}_3$ are given by equation (14). With the above approach, each boundary surface is divided into three regions: an interior region, a narrow interval or strip that separates the interior region from the edges of the boundary surface, and the edges themselves. In the interior region, $\partial \mathbf{r}(\xi=0, \eta, \zeta)/\partial \xi$ and $\partial \mathbf{r}(\xi, \eta=0, \zeta)/\partial \eta$ are specified such that they satisfy equations (13) and (14) which ensure orthogonality in that region. At the edges where the boundary surfaces intersect, the terms are specified such that the consistency conditions given by equations (26c), (26d) and (27c)–(27h) are satisfied. On the intervals adjacent to the edges, the terms are constructed as functions that bridge between the specified terms at the edges and those in the interior in a manner such that the consistency conditions given by equations (29) and (30) are satisfied. When equations (15) and (16) are used, grid lines will intersect boundary surfaces orthogonally everywhere except on the narrow strips near the edges where boundary surfaces intersect non-orthogonally.

BLENDING FUNCTIONS BASED ON TENSION SPLINE INTERPOLATION

One difficulty frequently encountered when using algebraic grid generation methods is that the cubic polynomials used as blending functions in Hermite interpolation to define connecting

curves often produces grid lines with too much curvature. Too much curvature can lead to problems such as excessive skewness of the grid or overlapping grid lines. To overcome this difficulty, new blending functions based on tension-spline interpolation are developed in this study to replace the cubic polynomials in the Hermite interpolation. The most attractive feature of the tension-spline blending functions developed here is that as the tension parameter is increased from zero to infinity, the blending functions vary from being cubic polynomials to being linear polynomials. Thus, the new tension-spline interpolation offers significantly increased control over the shape of grid lines in the grid system. We first derive the new blending functions for an arbitrary variable, and then its application to algebraic grid generation is illustrated.

Suppose the variable X is a function of the parameter s on an interval [0, 1], but only $X(0)$, $X(1)$, $X'(0)$ and $X'(1)$ (X' denotes dX/ds) are known, and a tension-spline interpolation of $X(s)$ on the interval [0, 1] is sought. A tension-spline interpolation of $X(s)$ is traditionally written in terms of $X(0)$, $X(1)$, $X''(0)$ and $X''(1)$, where $X''=d^2X/ds^2$, as follows:

$$X(s)=\frac{X''(0)\sinh[\sigma(1-s)]}{\sigma^2\sinh(\sigma)}+\left[X(0)-\frac{X''(0)}{\sigma^2}\right](1-s)+\frac{X''(1)\sinh(\sigma s)}{\sigma^2\sinh(\sigma)}+\left[X(1)-\frac{X''(1)}{\sigma^2}\right]s, \tag{17}$$

where σ is the tension parameter. By differentiating equation (17) and evaluating the resulting equation at the end points $s=0$ and $s=1$, we obtain

$$X'(0)=-\frac{X''(0)\cosh(\sigma)}{\sigma\sinh(\sigma)}-\left[X(0)-\frac{X''(0)}{\sigma^2}\right]+\frac{X''(1)}{\sigma\sinh(\sigma)}+\left[X(1)-\frac{X''(1)}{\sigma^2}\right], \tag{18a}$$

and

$$X'(1)=-\frac{X''(0)}{\sigma\sinh(\sigma)}-\left[X(0)-\frac{X''(0)}{\sigma^2}\right]+\frac{X''(1)\cosh(\sigma)}{\sigma\sinh(\sigma)}+\left[X(1)-\frac{X''(1)}{\sigma^2}\right]. \tag{18b}$$

The above two simultaneous equations can be solved to give expressions for $X''(0)$ and $X''(1)$ in terms of $X(0)$, $X(1)$, $X'(0)$ and $X'(1)$. Substituting the resulting expressions in equation (17) gives

$$X(s)=X(0)h_1(s)+X(1)h_2(s)+X'(0)h_3(s)+X'(1)h_4(s), \tag{19}$$

where

$$h_1(s)=c_1(1-s)+c_2s+c_2\left[\frac{\sinh[\sigma(1-s)]-\sinh(\sigma s)}{\sinh(\sigma)}\right], \tag{20a}$$

$$h_2(s)=c_1s+c_2(1-s)-c_2\left[\frac{\sinh[\sigma(1-s)]-\sinh(\sigma s)}{\sinh(\sigma)}\right], \tag{20b}$$

$$h_3(s)=c_3\left[(1-s)-\frac{\sinh[\sigma(1-s)]}{\sinh(\sigma)}\right]+c_4\left[s-\frac{\sinh(\sigma s)}{\sinh(\sigma)}\right], \tag{20c}$$

$$h_4(s)=-c_4\left[(1-s)-\frac{\sinh[\sigma(1-s)]}{\sinh(\sigma)}\right]-c_3\left[s-\frac{\sinh(\sigma s)}{\sinh(\sigma)}\right], \tag{20d}$$

$$c_1=1-c_2, \tag{21a}$$

$$c_2=\frac{\sinh(\sigma)}{2\sinh(\sigma)-\sigma\cosh(\sigma)-\sigma}, \tag{21b}$$

$$c_3=\frac{-\alpha}{(\beta^2-\alpha^2)}\sinh(\sigma), \tag{21c}$$

$$c_4=\frac{\beta}{(\beta^2-\alpha^2)}\sinh(\sigma), \tag{21d}$$

$$\alpha = \sigma \cosh(\sigma) - \sinh(\sigma), \tag{21e}$$

$$\beta = \sinh(\sigma) - \sigma. \tag{21f}$$

Equations (19)–(21) can be used to interpolate any function on the interval [0, 1], when the function's values and its first derivatives are known at the end points of the interval. The application of these equations to algebraic grid generation is straightforward. For example, with the two-boundary method,[2] we obtain

$$\mathbf{r}(\xi, \eta, \zeta) = \mathbf{r}(\xi, \eta = 0, \zeta) h_1(\eta) + \mathbf{r}(\xi, \eta = 1, \zeta) h_2(\eta) + \frac{\partial \mathbf{r}(\xi, \eta = 0, \zeta)}{\partial \eta} h_3(\eta) + \frac{\partial \mathbf{r}(\xi, \eta = 1, \zeta)}{\partial \eta} h_4(\eta), \tag{22}$$

where h_1, h_2, h_3 and h_4, are given by equations (20) and (21). The above equations have the same form as if Hermite interpolation was used except for the definition of the blending functions. However, the control over curvature of grid lines is increased considerably because as $\sigma \to 0$, h_1, h_2, h_3 and h_4 approach cubic polynomials, and as $\sigma \to \infty$, $h_1(s) \to (1-s)$, $h_2(s) \to s$, $h_3(s) \to 0$ and $h_4(s) \to 0$, giving rise to linear blending functions.

RESULTS

In this section, two grid systems generated by using the techniques developed in this study are presented.

Figure 2 shows a three-dimensional grid system where grid points are clustered towards one corner of the domain. This grid system was generated with the four-boundary technique in

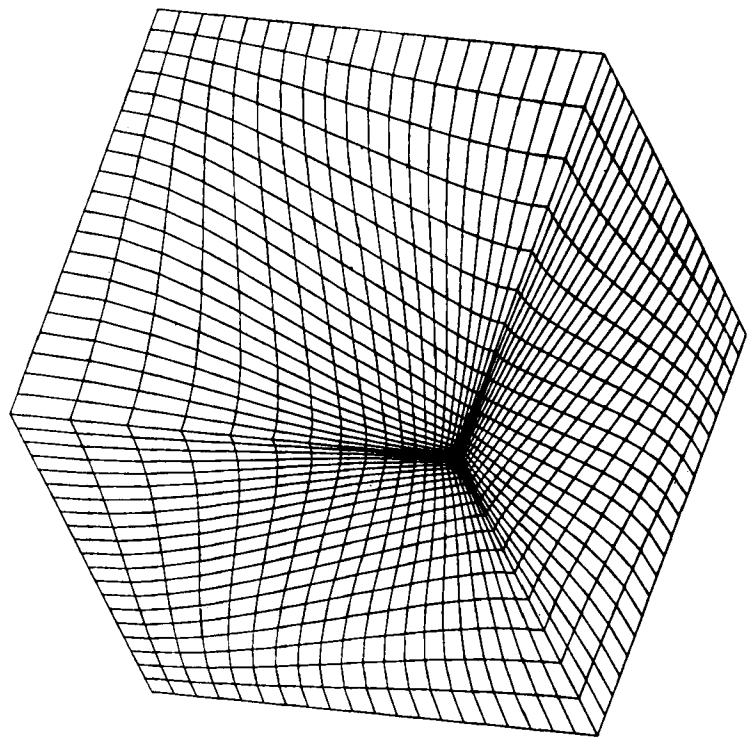

Figure 2. Multidimensional stretching function for clustering about a corner point in a three-dimensional grid system

conjunction with multidimensional stretching functions that were constructed using equation (1). Note that in this grid system, grid lines intersect the boundary surfaces orthogonally. Also note that this grid system could not have been generated by using one-dimensional stretching functions.

As a more stringent test for the techniques developed in this study, a grid system was generated for the coolant passage geometry shown in Figure 1. This geometry is very complicated with turning and twisting in three-dimensions coupled with serpentine passages and pin fins. The grid system sought for this complicated geometry was a partially continuous composite grid. Composite grids with the degree of continuity sought here are the most difficult to generate, but once they have been generated, they are the easiest to obtain solutions on with finite-difference or finite-volume methods.[2]

Figure 3 shows how the coolant passage was partitioned into 19 blocks or zones for the purpose of grid generation. The partitioning that is shown was deemed necessary in order to generate an acceptable grid system. A single grid system was generated for each of the partitions and then patched together to form a partially continuous composite grid. Figure 4 shows the grid system for the partition number 18. The entire grid system for the coolant passage is shown in Figures 5 and 6. A detailed description of the grid generation process for this complicated

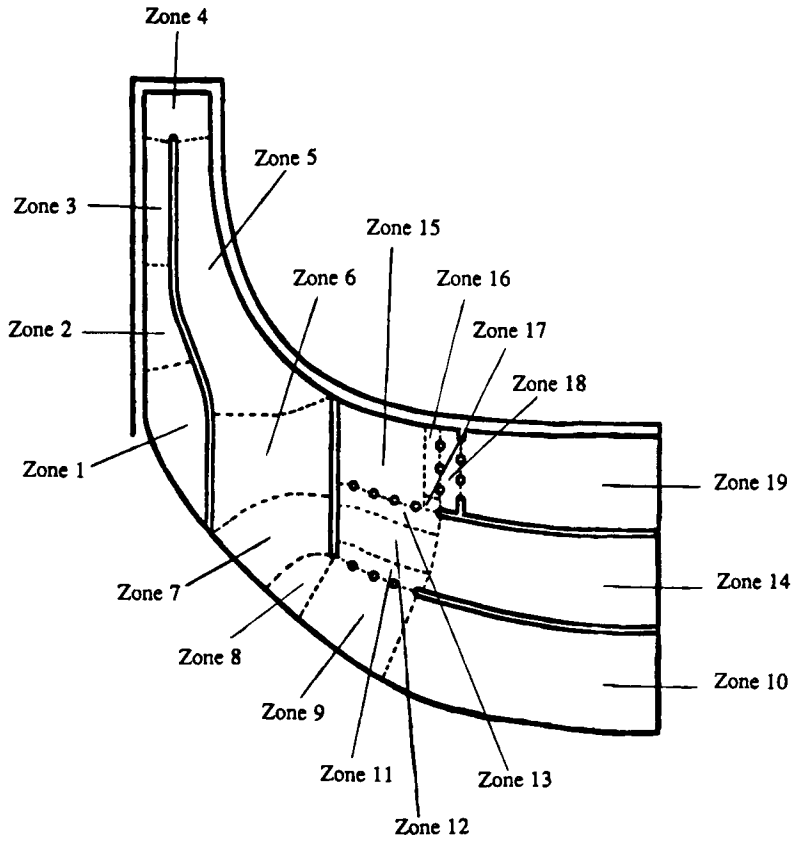

Figure 3. Partitioning of the coolant passage geometry into zones for grid generation

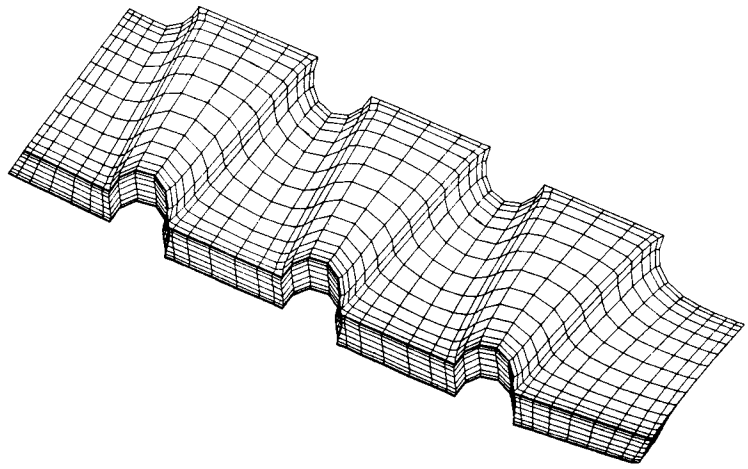

Figure 4. Grid system generated for zone 18 (see Figure 3)

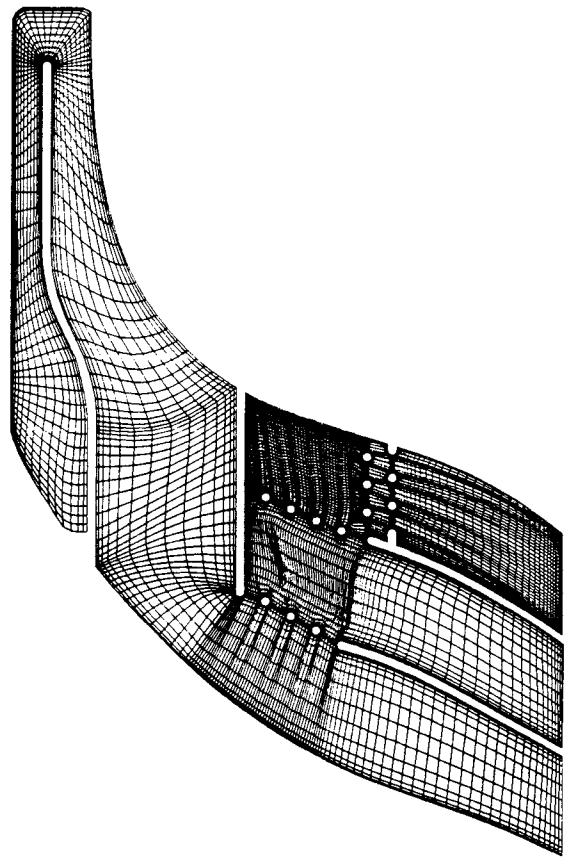

Figure 5. A two-dimensional view of the grid system for the coolant passage

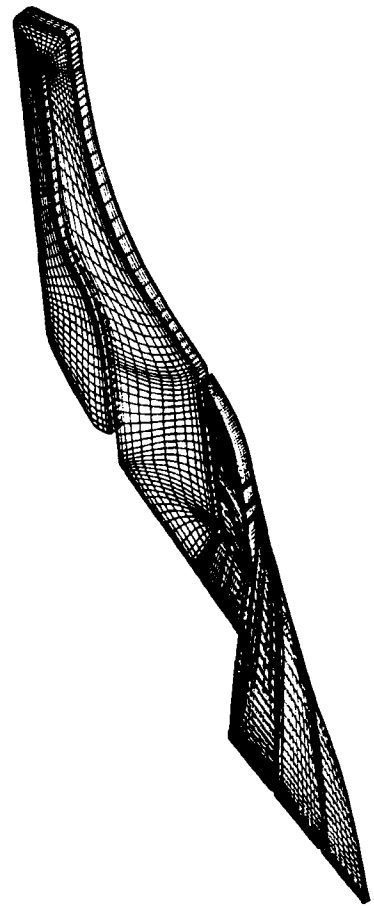

Figure 6. A three-dimensional view of the entire grid system for the coolant passage

geometry is given in Reference 4. This example demonstrates the usefulness of the three methods presented in this study to enhance control of grid-point distribution.

CONCLUDING REMARKS

In this paper, three techniques have been presented to enhance control of grid-point distribution in a class of algebraic grid generation methods known as the two-, four- and six-boundary methods. In addition to these three techniques, a set of consistency conditions was derived that must be satisfied by all user-specified data employed in the grid generation process. The usefulness of the techniques developed in this study was demonstrated by using them to generate two grid systems, including a three-dimensional grid system for the coolant passage of a radial turbine blade with serpentine channels and pin fins.

ACKNOWLEDGEMENT

This work was supported by NASA grant NAG 3-929. The authors are grateful for this support.

APPENDIX: CONSISTENCY CONDITIONS FOR USER-SPECIFIED DATA IN ALGEBRAIC GRID GENERATION

When using the four- and six-boundary methods, we need to specify curves or surfaces that describe the boundaries of the geometry and the derivatives at those boundaries. Since, for these two methods, the boundary curves or surfaces intersect, the data that we specify should satisfy certain consistency conditions. These conditions for the four-boundary method will be derived in this appendix. Note, while some of the conditions are obvious, others are not so obvious. Nonetheless, it is necessary to be aware of all these conditions in order to exercise maximum control over grid-point distribution. The consistency conditions formulated here have not been reported before.

Consider the four-boundary method given by equations (6)–(10). To facilitate discussion, equations (8) and (9) are recast here as

$$\mathbf{r}'(\xi,\eta,\zeta)=\mathbf{r}_1(\eta,\zeta)h_1(\xi)+\mathbf{r}_2(\eta,\zeta)h_2(\xi)+\mathbf{s}_1(\eta,\zeta)h_3(\xi)+\mathbf{s}_2(\eta,\zeta)h_4(\xi), \tag{23a}$$

$$\Delta\mathbf{r}(\xi,\eta,\zeta)=[\mathbf{r}_3(\xi,\zeta)-\mathbf{r}'(\xi,\eta=0,\zeta)]h_1(\eta)+[\mathbf{r}_4(\xi,\zeta)-\mathbf{r}'(\xi,\eta=1,\zeta)]h_2(\eta)+\left[\mathbf{s}_3(\xi,\zeta)-\frac{\partial\mathbf{r}'(\xi,\eta=0,\zeta)}{\partial\eta}\right]h_3(\eta)+\left[\mathbf{s}_4(\xi,\zeta)-\frac{\partial\mathbf{r}'(\xi,\eta=1,\zeta)}{\partial\eta}\right]h_4(\eta), \tag{23b}$$

where

$$\mathbf{s}_1(\eta,\zeta)=\frac{\partial\mathbf{r}(\xi=0,\eta,\zeta)}{\partial\xi}, \qquad \mathbf{s}_2(\eta,\zeta)=\frac{\partial\mathbf{r}(\xi=1,\eta,\zeta)}{\partial\xi}, \tag{24a, b}$$

$$\mathbf{s}_3(\xi,\zeta)=\frac{\partial\mathbf{r}(\xi,\eta=0,\zeta)}{\partial\eta}, \qquad \mathbf{s}_4(\xi,\zeta)=\frac{\partial\mathbf{r}(\xi,\eta=1,\zeta)}{\partial\eta}. \tag{24c, d}$$

The terms $\mathbf{r}_1$, $\mathbf{r}_2$, $\mathbf{r}_3$ and $\mathbf{r}_4$ are parametric representations of boundary surfaces, and $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$ and $\mathbf{s}_4$ are derivatives transverse to the boundary surfaces. These terms are the data that need to be specified, and this data must satisfy certain conditions for consistency. All of these consistency conditions can be found by evaluating $\mathbf{r}(\xi,\eta,\zeta)$ and its derivatives at boundaries of the domain. For example, evaluating $\mathbf{r}(\xi,\eta,\zeta)=\mathbf{r}'(\xi,\eta,\zeta)+\Delta\mathbf{r}(\xi,\eta,\zeta)$ at $\xi=0$ gives [note that $h_1(\xi=0)=1$, $h_2(\xi=0)=0$, $h_3(\xi=0)=0$, $h_4(\xi=0)=0$]

$$\mathbf{r}(\xi=0,\eta,\zeta)=\mathbf{r}_1(\eta,\zeta)-[\mathbf{r}_1(\eta=0,\zeta)-\mathbf{r}_3(\xi=0,\zeta)]h_1(\eta)-[\mathbf{r}_1(\eta=1,\zeta)-\mathbf{r}_4(\xi=0,\zeta)]h_2(\eta)-\left[\frac{\partial\mathbf{r}_1(\eta=0,\zeta)}{\partial\eta}-\mathbf{s}_3(\xi=0,\zeta)\right]h_3(\eta)-\left[\frac{\partial\mathbf{r}_1(\eta=1,\zeta)}{\partial\eta}-\mathbf{s}_4(\xi=0,\zeta)\right]h_4(\eta). \tag{25}$$

Since $\mathbf{r}_1(\eta,\zeta)=\mathbf{r}(\xi=0,\eta,\zeta)$ by definition, the data $\mathbf{r}_1$, $\mathbf{r}_3$, $\mathbf{r}_4$, $\mathbf{s}_3$ and $\mathbf{s}_4$ must satisfy the following conditions in order for equation (25) to give the correct results:

$$\mathbf{r}_1(\eta=0,\zeta)=\mathbf{r}_3(\xi=0,\zeta), \tag{26a}$$

$$\mathbf{r}_1(\eta=1,\zeta)=\mathbf{r}_4(\xi=0,\zeta), \tag{26b}$$

$$\mathbf{s}_3(\xi=0,\zeta)=\frac{\partial\mathbf{r}_1(\eta=0,\zeta)}{\partial\eta}, \tag{26c}$$

$$\mathbf{s}_4(\xi=0,\zeta)=\frac{\partial\mathbf{r}_1(\eta=1,\zeta)}{\partial\eta}, \tag{26d}$$

Similarly, evaluating $\mathbf{r}(\xi, \eta, \zeta)$ at other boundaries gives additional conditions, namely,

$$\mathbf{r}_2(\eta=0, \zeta)=\mathbf{r}_3(\xi=1, \zeta), \tag{27a}$$

$$\mathbf{r}_2(\eta=1, \zeta)=\mathbf{r}_4(\xi=1, \zeta), \tag{27b}$$

$$\mathbf{s}_3(\xi=1, \zeta)=\frac{\partial \mathbf{r}_2(\eta=0, \zeta)}{\partial \eta}, \tag{27c}$$

$$\mathbf{s}_4(\xi=1, \zeta)=\frac{\partial \mathbf{r}_2(\eta=1, \zeta)}{\partial \eta}, \tag{27d}$$

$$\mathbf{s}_1(\eta=0, \zeta)=\frac{\partial \mathbf{r}_3(\xi=0, \zeta)}{\partial \xi}, \tag{27e}$$

$$\mathbf{s}_2(\eta=0, \zeta)=\frac{\partial \mathbf{r}_3(\xi=1, \zeta)}{\partial \xi}, \tag{27f}$$

$$\mathbf{s}_1(\eta=1, \zeta)=\frac{\partial \mathbf{r}_4(\xi=0, \zeta)}{\partial \xi}, \tag{27g}$$

$$\mathbf{s}_2(\eta=1, \zeta)=\frac{\partial \mathbf{r}_4(\xi=1, \zeta)}{\partial \xi}. \tag{27h}$$

Note, the conditions given by equations (26) and (27) are conditions of continuity of the geometry at the edges where two boundary surfaces intersect. Obviously, the boundary of the geometry must be continuous at the edges, which is the condition given by equations (26a), (26b), (27a) and (27b). The meaning of conditions given by equations (26c), (26d) and (27c)–(27h) is that at the edges, the derivative that we specify transverse to one of the boundary surfaces must be tangent to the intersecting boundary surface.

The rest of the consistency conditions that the data must satisfy can be found by evaluating the derivatives of $\mathbf{r}(\xi, \eta, \zeta)$ at the boundaries, for example,

$$\frac{\partial \mathbf{r}(\xi=0, \eta, \zeta)}{\partial \xi}=\mathbf{s}_1(\eta, \zeta)+\left[\frac{\partial \mathbf{r}_3(\xi=0, \zeta)}{\partial \xi}-\mathbf{s}_1(\eta=0, \zeta)\right] h_1(\eta)+\left[\frac{\partial \mathbf{r}_4(\xi=0, \zeta)}{\partial \xi}-\mathbf{s}_1(\eta=1, \zeta)\right] h_2(\eta)$$
$$-\left[\frac{\partial \mathbf{s}_1(\eta=0, \zeta)}{\partial \eta}-\frac{\partial \mathbf{s}_3(\xi=0, \zeta)}{\partial \xi}\right] h_3(\eta)-\left[\frac{\partial \mathbf{s}_1(\eta=0, \zeta)}{\partial \eta}-\frac{\partial \mathbf{s}_4(\xi=0, \zeta)}{\partial \xi}\right] h_4(\eta). \tag{28}$$

Since by definition, $\mathbf{s}_1(\eta, \zeta)=\partial \mathbf{r}(\xi=0, \eta, \zeta)/\partial \xi$, the data must satisfy the conditions given by equations (27e) and (27g), and in addition

$$\frac{\partial \mathbf{s}_1(\eta=0, \zeta)}{\partial \eta}=\frac{\partial \mathbf{s}_3(\xi=0, \zeta)}{\partial \xi}, \tag{29a}$$

$$\frac{\partial \mathbf{s}_1(\eta=1, \zeta)}{\partial \eta}=\frac{\partial \mathbf{s}_4(\xi=0, \zeta)}{\partial \xi}. \tag{29b}$$

Note, the conditions given by equation (29) are simply that

$$\frac{\partial^2 \mathbf{r}(\xi=0, \eta=0, \zeta)}{\partial \xi \partial \eta}=\frac{\partial^2 \mathbf{r}(\xi=0, \eta=0, \zeta)}{\partial \eta \partial \xi},$$

and

$$\frac{\partial^2 \mathbf{r}(\xi=0, \eta=1, \zeta)}{\partial \xi \partial \eta}=\frac{\partial^2 \mathbf{r}(\xi=0, \eta=1, \zeta)}{\partial \eta \partial \xi}.$$

By evaluating the derivatives of $\mathbf{r}(\xi, \eta, \zeta)$ at other boundaries, we arrive at additional conditions that the data must satisfy, namely,

$$\frac{\partial \mathbf{s}_2(\eta=0, \zeta)}{\partial \eta} = \frac{\partial \mathbf{s}_3(\xi=1, \zeta)}{\partial \xi}, \tag{30a}$$

$$\frac{\partial \mathbf{s}_2(\eta=1, \zeta)}{\partial \eta} = \frac{\partial \mathbf{s}_4(\xi=1, \zeta)}{\partial \xi}. \tag{30b}$$

The conditions implied by equations (26) and (28)–(30) are the consistency conditions that the data used to define the transformation $\mathbf{r}(\xi, \eta, \zeta)$ must satisfy in the four-boundary method. If any of these conditions is not satisfied, then one can have difficulties in achieving the desired results in the grid generation. For example, if the conditions given by equations (29) and (30) are not satisfied, then one can have difficulty in achieving orthogonality at the boundaries.

By using the approach illustrated above for the four-boundary method, consistency conditions for the six-boundary method can also be derived. There are no consistency conditions for the two-boundary method since it does not involve intersecting boundary curves or surfaces.

REFERENCES

1. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation*, Elsevier, Amsterdam, 1985.
2. T. I-P. Shih, R. T. Bailey, H. L. Nguyen and R. J. Roelke, 'Algebraic grid generation for complex geometries', *Int. j. numer. methods fluids*, **13**, 1–31 (1991).
3. M. Vinokur, 'On one-dimensional stretching functions for finite-difference calculations', *J. Comp. Phys.*, **50**, 215–234 (1983).
4. E. Steinthorsson, 'Numerical simulations of complex three-dimensional viscous flows', *Ph.D. Thesis*, Department of Mechanical Engineering, Carnegie Mellon University, 1991.