COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING 64 (1987) 377–411
NORTH-HOLLAND

A GENERAL THREE-DIMENSIONAL ELLIPTIC GRID GENERATION SYSTEM ON A COMPOSITE BLOCK STRUCTURE*

Joe F. THOMPSON

Department of Aerospace Engineering, Mississippi State University, Mississippi State, MS 39762, U.S.A.

Received 16 March 1987

A general three-dimensional elliptic grid generation system is discussed. This system includes automatic evaluation of control functions from an algebraic grid or interpolated from boundary point distributions. Also included is iterative adjustment of control functions for boundary orthogonality. The solution is by point SOR iteration using a field of optimum acceleration parameters. This system is incorporated in a code that can treat any physical configuration by using a multiblock structure with complete continuity across interfaces.

1. Introduction

Finite difference (or finite volume) and finite element solutions both require a discrete set of points or cells covering the physical field, and the efficiency of the computation is greatly enhanced if there is some organization to this set. This organization can be provided by having the discretization defined by the nodes of a curvilinear coordinate system filling the physical field. Such systems are readily available from handbooks for certain simple configurations such as regions that are cylindrical, spherical, elliptical, etc. For general regions of arbitrary shape, numerical grid generation provides the curvilinear system.

The techniques of numerical grid generation, and its application to the numerical solution of partial differential equations, are covered in detail in a recent text on the subject [1]. Several surveys of the field have also been given [2–5], and four conference proceedings dedicated to the area have appeared [6–9]. The first of these proceedings also contains a number of expository papers and other sources on the subject.

The curvilinear system can be constructed simply by setting values in a rectangular array of position vectors:

$$\boldsymbol{r}_{ijk}\,, \quad i=1,2,\ldots,I\,,\ \ j=1,2,\ldots,J\,,\ \ k=1,2,\ldots,K\,,$$

and identifying the indices i, j, k with the three curvilinear coordinates. The position vector $\boldsymbol{r}$ is a three-vector giving the values of the Cartesian x-, y-, and z-coordinates of a grid point.

*This research was supported by the U.S. Air Force Armament Laboratory, Eglin Air Force Base, under Grant F08635-84-0028, Dr. Lawrence E. Lijewski monitor.

Since all increments in the curvilinear coordinates cancel out of the transformation relations for derivative operators, there is no loss of generality in defining the discretization to be on integer values of these coordinates.

Fundamental to this curvilinear coordinate system is the coincidence of some coordinate surface with each segment of the boundary of the physical region, in the same manner that surfaces of constant radius coincide with the inner and outer boundary segments of the region between two concentric spheres filled with a polar coordinate system. This is accomplished by placing a two-dimensional array of points on a physical boundary segment and setting these values in the array of position vectors with one index constant, e.g., in $\boldsymbol{r}_{ijk}$ with i from 1 to I and j from 1 to J. The curvilinear coordinate k is thus constant on this physical boundary segment. With values set on the sides of the rectangular array of position vectors in this manner, the generation of the grid is accomplished by determining the values of $\boldsymbol{r}_{ijk}$ in the interior of the rectangular array from the specified boundary values on its sides. The set of values $\boldsymbol{r}_{ijk}$ then forms the nodes of a curvilinear coordinate system filling the physical region. A physical region bounded by six generally curved sides can thus be considered to have been transformed to a rectangular computational region on which the curvilinear coordinates (i.e., the indices i, j, k) are the independent variables.

Although in principle it is possible to establish a correspondence between any physical region and a single empty rectangular block for general three-dimensional configurations, the resulting grid is likely to be much too skewed and irregular to be usable when the boundary geometry is complicated. A better approach with complicated physical boundaries is to segment the physical region into contiguous subregions, each bounded by six curved sides (four in 2D) and each of which transforms to a rectangular block in the computational region, with a grid generated within each subregion [10, 11]. Each subregion has its own curvilinear coordinate system irrespective of that in the adjacent subregions.

This then allows both the grid generation and numerical solutions on the grid to be constructed to operate in a rectangular computational region, regardless of the shape or complexity of the full physical region. The full region is treated by performing the solution operation in all of the rectangular computational blocks. With the composite framework, partial differential equation solution procedures written to operate on rectangular regions can be incorporated into a code for general configurations in a straightforward manner, since the code only needs to treat a rectangular block. The entire physical field then can be treated in a loop over all the blocks.

The generally curved surfaces bounding the subregions in the physical region form internal interfaces across which information must be transferred, i.e., from the sides of one rectangular computational block to those of another. These interfaces occur in pairs, an interface on one block being paired with another on the same, or another, block since both correspond to the same physical surface. Grid lines at the interfaces may meet with complete continuity, with or without slope continuity, or may not even meet.

Complete continuity of grid lines across the interface requires that the interface be treated as a branch cut on which the generation system is solved just as it is in the interior of blocks. The interface locations are then not fixed, but are determined by the grid generation system [10]. This is most easily handled in coding by providing an extra layer of points surrounding each block. Here the grid points on an interface of one block are coincident in physical space with those on another interface of the same or another block, and also the grid points on the

surrounding layer outside the first interface are coincident with those just inside the first interface, and vice versa. This coincidence can be maintained during the course of an interative solution of an elliptic generation system by setting the values on the surrounding layers equal to those at the corresponding interior points after each iteration. All the blocks are thus iterated to convergence together, so that the entire composite grid is generated at once.

The construction of flow codes for the solution of partial differential equations (PDE) in complicated regions is greatly simplified by the composite grid structure since, with the use of the surrounding layer of points on each block, a PDE code is only required basically to operate on a rectangular computational region. The necessary correspondence of points on the surrounding layers (image points) with interior points (object points) is set up by the grid code and made available to the PDE solution code.

The present paper discusses the three-dimensional elliptic generation system used in a new general three-dimensional grid generation code [19] based on the block structure. This system allows any number of blocks to be used to cover an arbitrary three-dimensional region. Any block can be linked to any other block (or to itself), with complete (or lesser) continuity across the block interfaces as specified by input. This code uses an elliptic generation system with automatic evaluation of control functions either directly from the initial algebraic grid and then smoothed, or by interpolation from the boundary point distributions. This evaluation by interpolation from boundaries differs from earlier related procedures [11] in that the arc length and curvature contributions to the control functions are evaluated and interpolated separately into the field from the appropriate boundaries. The control function at each point in the field is then formed by combining the interpolated components. This procedure allows very general regions, with widely varying boundary curvature, to be treated. For complex multiblocked configurations, however, the evaluation from the algebraic grid appears to be the most generally applicable.

The control functions can be also determined automatically to provide orthogonality at boundaries with specified normal spacing (related to the GRAPE code [15]). In the present code, the iterative adjustments in the control function are made by increments radiated from boundary points where orthogonality has not yet been attained. This allows the basic control function structure evaluated from the geometry of the configuration to be retained and thus relieves the iterative process from the need to establish this basic form of the control functions.

The elliptic generation system is solved by point SOR iteration using a field of locally-optimum acceleration parameters [12]. These optimum parameters make the solution robust and capable of convergence with strong control functions. The code includes an algebraic three-dimensional generation system based on transfinite interpolation [13, 14] (using either Lagrange or Hermite interpolation) for the generation of an initial algebraic grid to start the iterative solution of the elliptic generation system.

2. Elliptic system

Since elliptic partial differential equations determine a function in terms of its values on the entire closed boundary of a region, such a system can be used to generate the interior values

in the array r_{ijk} from the values on the sides. The properties of elliptic grid generation systems are discussed in [1, 6]. With such systems there is less tendency for grid overlap, and the grids tend to be smoother than algebraic grids.

2.1. General Poisson-type systems

If a curvilinear coordinate system, $\bar{\xi}^i$, $i = 1, 2, 3$, which satisfies the Laplace system,

$$\nabla^2 \bar{\xi}^i = 0 , \quad i = 1, 2, 3 ,$$

is transformed to another coordinate system, ξ^i, $i = 1, 2, 3$, then the new curvilinear coordinates, ξ^i, satisfy the inhomogeneous elliptic system (cf. [1]),

$$\nabla^2 \xi^i = P^i , \quad i = 1, 2, 3 , \tag{1}$$

where

$$P^i = \sum_{j=1}^{3} \sum_{k=1}^{3} g^{jk} P^i_{jk} , \tag{2}$$

where the g^{jk} are the elements of the contravariant metric tensor, and with the P^i_{jk} defined by the transformation from $\bar{\xi}^i$ to ξ^i:

$$P^i_{jk} = \sum_{m=1}^{3} \sum_{n=1}^{3} \frac{\partial \bar{\xi}^m}{\partial \xi^j} \frac{\partial \bar{\xi}^n}{\partial \xi^k} \frac{\partial^2 \xi^i}{\partial \bar{\xi}^m \partial \bar{\xi}^n} . \tag{3}$$

(It may be noted that if the subsequent transformation is one-dimensional, i.e., if $\partial \bar{\xi}^i / \partial \xi^j = \delta^i_j \, \partial f^i / \partial \xi^j$, then only the three functions P^i_{ii}, with $i = 1, 2, 3$, are nonzero.)

These results show that a grid with lines concentrated by applying a subsequent transformation (often called a 'stretching' transformation) to a grid generated as a solution of the Laplace system could have been generated directly as the solution of the Poisson system (1) with appropriate 'control functions', P^i_{jk}, derived from the subsequent concentrating transformation according to (3). Therefore, it is appropriate to adopt this Poisson system (1) as the generation system, but with the control functions specified directly rather than through a subsequent transformation.

Thus, an appropriate generation system can be defined by (1) and (2):

$$\nabla^2 \xi^i = \sum_{j=1}^{3} \sum_{k=1}^{3} g^{jk} P^i_{jk} , \quad i = 1, 2, 3 , \tag{4}$$

with the control functions, P^i_{jk}, considered to be specified. The basis of the generation system (4) is that it produces a coordinate system that corresponds to the subsequent application of a stretching transformation to a coordinate system generated for maximum smoothness. From (3), the three control functions P^i_{ii}, $i = 1, 2, 3$, correspond to one-dimensional stretching in each coordinate direction and thus are the most important of the control functions. In applications, in fact, the other control functions have been taken to be zero, i.e., $P^i_{jk} =$

$\delta^i_j \delta^i_k P_i$, so that the generation system becomes

$$\nabla^2 \xi^i = g^{ii} P_i \,, \quad i = 1, 2, 3 \,. \tag{5}$$

It may be noted that (7) can be written as

$$\nabla^2 \xi^i = \sum_{j=1}^{3} \sum_{k=1}^{3} P^i_{jk} (\nabla \xi^j \cdot \nabla \xi^k) = 0 \,. \tag{6}$$

Actual computation is to be done in the rectangular transformed field where the curvilinear coordinates, ξ^i, are the independent variables, with the Cartesian coordinates, x_i, as dependent variables. The transformation of (6) is,

$$\sum_{i=1}^{3} \sum_{j=1}^{3} g^{ij} \left(\boldsymbol{r}_{\xi^i \xi^j} + \sum_{k=1}^{3} P^k_{ij} \boldsymbol{r}_{\xi^k} \right) = 0 \,. \tag{7}$$

This then is the quasi-linear elliptic partial differential equation which can be solved to generate the coordinate system. (In computation, the Jacobian squared, g, can be omitted from the evaluation of the metric coefficients, g^{ij}, in this equation since it would cancel anyway.)

2.2. Generation system

The elliptic grid generation system used in this code thus is

$$\sum_{m=1}^{3} \sum_{n=1}^{3} g^{mn} \boldsymbol{r}_{\xi^m \xi^n} + \sum_{n=1}^{3} g^{nn} P_n \boldsymbol{r}_{\xi^n} = 0 \,, \tag{8}$$

where the g^{mn} are the elements of the contravariant metric tensor:

$$g^{mn} = \nabla \xi^m \cdot \nabla \xi^n \,. \tag{9}$$

These elements are more conveniently expressed in terms of the elements of the covariant metric tensor, g_{mn},

$$g_{mn} = \boldsymbol{r}_{\xi^m} \cdot \boldsymbol{r}_{\xi^n} \,, \tag{10}$$

which can be calculated directly. Thus,

$$g^{mn} = (g_{ik} g_{jl} - g_{il} g_{jk})/g \,, \quad (m, i, j) \,, (n, k, l) \text{ cyclic} \,,$$

where g, the square of the Jacobian, is given by,

$$g = \det |g_{mn}| = \boldsymbol{r}_{\xi^1} \cdot (\boldsymbol{r}_{\xi^2} \times \boldsymbol{r}_{\xi^3}) \,. \tag{11}$$

In these relations, $\boldsymbol{r}$ is the Cartesian position vector of a grid point, $\boldsymbol{r} = \boldsymbol{i}x + \boldsymbol{j}y + \boldsymbol{k}z$, and the ξ^i, $i = 1, 2, 3$, are the three curvilinear coordinates. The P_n are the three 'control functions' which serve to control the spacing and orientation of the grid lines in the field.

In the absence of control functions, the generation system reduces to Laplace equations, which produce the smoothest possible grid, with a tendency of lines to concentrate over convex boundary regions and to spread out over concave regions.

Negative values of the control function P_n cause grid lines on which ξ^n is constant to tend to move in the direction of decreasing ξ^n, and this feature can be used to concentrate grid lines near other grid lines and/or points or in certain regions of physical space. However, a more automatic procedure is to determine the control functions so as to reflect the spacing of an algebraic grid, or to reflect the boundary point spacing into the field. Both of these approaches are discussed in the sections that follow.

3. Control functions from an algebraic grid

The three components of the elliptic grid generation system (8) provide a set of three equations,

$$\sum_{k=1}^{3} g^{kk}(\boldsymbol{r}_{\xi^k})_l P_k = -\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}(\boldsymbol{r}_{\xi^i\xi^j})_l\,, \quad l = 1, 2, 3\,, \tag{12}$$

that can be solved simultaneously at each point for the three control functions, P_k, $k = 1, 2, 3$. The derivatives here are represented by central differences. This produces control functions which will reproduce the algebraic grid from the elliptic system solution in a single iteration. Thus, evaluation of the control functions in this manner would be of trivial interest except when these control functions are smoothed before being used in the elliptic generation system. This smoothing is done by replacing the control function at each point with the average of the four neighbors in the two curvilinear directions (one in 2D) other than that of the function, see Fig. 1. Thus,

$$\begin{aligned} P_1(\xi^1, \xi^2, \xi^3) = \tfrac{1}{4}[&P_1(\xi^1, \xi^2 + 1, \xi^3) + P_1(\xi^1, \xi^2 - 1, \xi^3) + P_1(\xi^1, \xi^2, \xi^3 + 1) \\ &+ P_1(\xi^1, \xi^2, \xi^3 - 1)]\,, \end{aligned} \tag{13}$$

with analogous equations for P_2 and P_3. No smoothing is done in the direction of the function because to do so would smooth the spacing distribution.

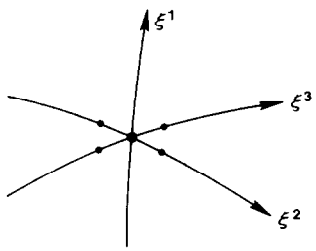

Fig. 1. Smoothing of control functions.

The code generates an algebraic grid by transfinite interpolation from the boundary point distribution to serve as the starting solution for the SOR iteration for the elliptic system. With the boundary point distribution set from the hyperbolic sine or tangent functions (Appendix A), which have been shown to give reduced truncation error, this algebraic grid has a good spacing distribution but may have slope breaks propogated from corners into the field. The use of smoothed control functions evaluated from the algebraic grid, as discussed above, produces a smooth grid that retains essentially the spacing of the algebraic grid.

A variation of this evaluation procedure is to set the off-diagonal metric elements to zero in the evaluation of the control functions from the algebraic grid, as if the grid were orthogonal. This produces a grid from the elliptic system that has the same qualitative line distribution as the algebraic grid but is more orthogonal. A third variation sets only the two off-diagonal metric elements having one index the same as that of the control function to zero. This amounts to relaxing the assumption of orthogonality in the curvilinear coordinate surface associated with each control function. Both of these variations, however, tend to produce skewness at boundaries.

4. Control functions from an algebraic grid

Control functions can be evaluated on the boundaries using the specified boundary point distribution in the generation system, with certain necessary assumptions to eliminate some terms, and can then be interpolated from the boundaries into the field. Earlier approaches [11] interpolated the entire functions from the boundaries in this manner. More general regions can, however, be treated by interpolating elements of the control functions separately. (Some related work along these lines has appeared in [16]).

4.1. General development

Consider first a physical rectangle with equally spaced points on the horizontal sides, but with the same unequal spacing on the two vertical sides. With no control functions, i.e., $P_n = 0$, (8) will produce a grid that attempts to be equally spaced away from the unequal spacing on the vertical sides, as shown in Fig. 2.

A grid of parallel lines for this configuration, reflecting the unequal spacing on the boundaries, as shown in Fig. 3, can only be produced from (8) with $P_2 \neq 0$ (taking ξ^1 to vary on the horizontal sides and ξ^2 to vary on the vertical sides).

The proper values of P_2 needed to accomplish this are determined by evaluating (8)

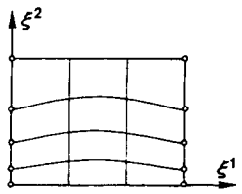

Fig. 2. Grid without control function.

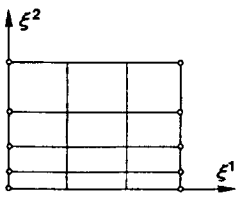

Fig. 3. Grid with control function.

one-dimensionally on the vertical sides, with the result:

$$y_{\xi^2\xi^2} + P_2 y_{\xi^2} = 0 ,$$

so that

$$P_2 = -y_{\xi^2\xi^2}/y_{\xi^2} . \tag{14}$$

If there are J points on the vertical sides and I points on the horizontal sides, the control function P_2 on the vertical sides can then be evaluated from the point distribution thereon as

$$P_2(1, j) = -\frac{[y(1, j+1) - 2y(1, j) + y(1, j-1)]}{\frac{1}{2}[y(1, j+1) - y(1, j-1)]} ,$$

for $j = 2, 3, \ldots, J-1$, with an analogous equation with I for the first argument. The values of P_2 in the interior of the region then can be interpolated between the two vertical sides as shown in Fig. 4. A similar evaluation of the other control function, P_1, on the two horizontal sides from

$$P_1 = -x_{\xi^1\xi^1}/x_{\xi^1} \tag{15}$$

produces a zero in the present case with equal spacing on these sides. In the case of unequal spacing, the values of P_1 in the interior of the region would be evaluated by interpolation between the values on the two horizontal sides as shown in Fig. 5.

With the control functions evaluated in this manner, (8) will produce a grid composed of parallel straight lines for this boundary configuration, thus reflecting the boundary point spacing into the field.

Now consider an O-type grid with two concentric circular boundaries and equally spaced points around the circles. Because of its inherent tendency to cause the grid lines to move closer to convex boundaries, (8) with no control functions will produce a grid with unequal radial spacing of the circumferential lines as shown in Fig. 6. In this case, evaluation of (8) with

$$x(r, \theta) = r(\xi^2)\cos\theta , \qquad y(r, \theta) = r(\xi^2)\sin\theta$$

yields the equation

$$P_2 = -\frac{r_{\xi^2\xi^2}}{r_{\xi^2}} + \frac{r_{\xi^2}}{r} . \tag{16}$$

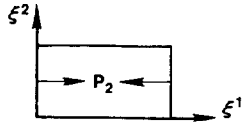

Fig. 4. Interpolation of P_2.

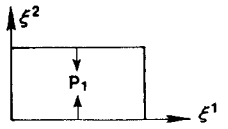

Fig. 5. Interpolation of P_1.

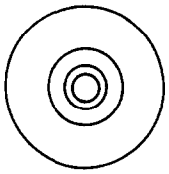

Fig. 6. Grid generated without control functions.

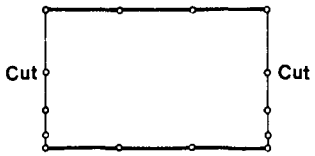

Fig. 7. Cut between two circular boundaries.

Thus, in order to produce a specified radial distribution of lines, the control function P_2 must be evaluated from (16) using the given distribution $r(\xi^2)$. Now, the computational field here has its two vertical sides corresponding to a cut between the two circular boundaries in the physical field, as shown in Fig. 7.

Therefore, with the specified radial distribution placed on the two vertical sides of the computational field, the control function P_2 can be evaluated from (16) on these sides as

$$P_2(1, j) = -\frac{[r(1, j+1) - 2r(1, j) + r(1, j-1)]}{\frac{1}{2}[r(1, j+1) - r(1, j-1)]} + \frac{\frac{1}{2}[r(1, j+1) - r(1, j-1)]}{r(1, j)}, \tag{17}$$

and a similar equation with I as the first argument. The interior values can then be determined by interpolation between the two vertical sides as before. Note that this amounts to interpolation in the circumferential direction in the physical plane, as indicated in Fig. 8. Again, equal spacing around the circles produces a zero value of the other control function, P_1. In general, P_1 would be evaluated on the two circles and interpolated between the two horizontal sides in the computational region, i.e., between the two circles in the physical field, as shown in Fig. 8.

The second term in (16) arises from the curvature of the boundary, and the denominator is the local radius of curvature of the grid line that is to pass through the point where the control function is being evaluated. This term acts to reduce the magnitude of the control function in order to allow for the natural tendency of the grid lines to move toward convex boundaries. Since the lines tend to concentrate near the inner circle even with zero control functions, the use of the first term alone in (16) (in analogy with the flat boundary case, i.e., (14)) would produce a stronger concentration of lines near the inner circle than was intended.

Finally, consider a C-type grid as shown in Fig. 9, with the computational region as shown in Fig. 10.

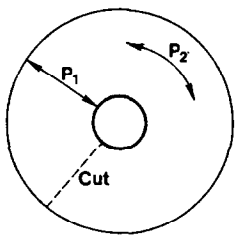

Fig. 8. Interpolation for P_1 and P_2.

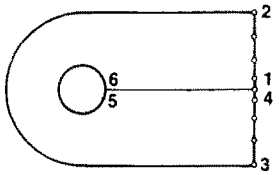

Fig. 9. C-type grid.

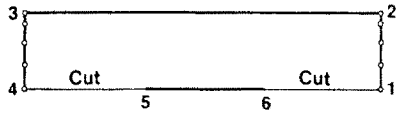

Fig. 10. Computational region for C-type grid.

Now, if the control function P_2 is evaluated on lines 1-2 and 4-3 in the physical field from (14), and the interior values are interpolated between the two vertical sides in the computational field, the resulting control function, while serving well over the right portion of the physical field, will be too strong over the inner body where the line curvature is not zero. The use of (16) on the lines 1-2 and 4-3 would be no better since the r in the denominator is to be interpreted as the local radius of curvature of the crossing line and hence is infinite on these lines so that the second term in (16) vanishes.

This situation can be remedied by interpolating for the local radius of curvature in (16) between the inner and outer boundaries in the physical field, i.e., between the horizontal sides in the computational region. However, since the ξ^2 derivatives in (16) must still be evaluated on the vertical sides, it is necessary to separate (16) into three pieces:

$$P_2 = -\frac{r_{\xi^2\xi^2}}{r_{\xi^2}} + \frac{r_{\xi^2}}{r} = A_2 + \frac{s_2}{\rho_2}\,, \tag{18}$$

where A_2 is the contribution from the rate of change of the arc length spacing,

$$A_2 = -r_{\xi^2\xi^2}/r_{\xi^2}\,,$$

s_2 is the arc length spacing,

$$s_2 = r_{\xi^2}\,,$$

and ρ_2 is the radius of curvature,

$$\rho_2 = r\,.$$

Now the arc length contributions, A_2 and s_2, are evaluated on the vertical sides of the computational region, while the radius of curvature, ρ_2, is evaluated on the two horizontal sides. The control function in the interior is then evaluated by interpolating the arc length contributions between the vertical sides, interpolating the radius of curvature between the horizontal sides, and then evaluating P_2 from (18) using the interpolated values as shown in Fig. 11.

Note that this procedure supplies a finite radius of curvature over the inner body, thus reducing the control function appropriately in this region.

A problem arises, however, when the radius of curvature is of opposite sign on the two boundaries between which it is interpolated, see Fig. 12, since then the interpolation will

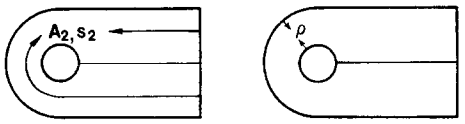

Fig. 11. Interpolations on C-type grid.

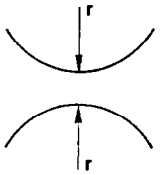

Fig. 12. Radii of opposite signs.

produce a zero value at some point in between, and at such a point the second term of (18) is infinite. Special measures are then necessary as discussed in Appendix C.

Although the exact equations for the general case are more complicated, they still may be separated into terms arising from the spacing along a boundary and terms arising from the local curvature of the crossing lines, as discussed below, with the spacing terms, A_n and s_n, for the control function P_n being interpolated between the four sides on which ξ^n varies, see Fig. 13, and the curvature term, ρ_n, for P_n interpolated between the two sides on which ξ_n is constant, see Fig. 14.

The control function is then evaluated from

$$P_n = A_n + s_n/\rho_n \,. \tag{19}$$

4.2. Block sides

For a block side on which ξ^m and ξ^n vary (on which ξ^l is constant), see Fig. 15, this routine calculates the arc length contributions, A_m and A_n, the spacings, s_m and s_n, the radius of curvature of the surface, ρ_l, from the separate pieces of (14) in Appendix B:

$$A_m = -\frac{\boldsymbol{r}_{\xi^m}\cdot\boldsymbol{r}_{\xi^m\xi^m}}{g_{mm}}\,, \qquad A_n = -\frac{\boldsymbol{r}_{\xi^n}\cdot\boldsymbol{r}_{\xi^n\xi^n}}{g_{nn}}\,, \tag{20}$$

$$s_m = \sqrt{g_{mm}}\,, \qquad s_n = \sqrt{g_{nn}}\,, \tag{21}$$

$$\rho_l = -\left[\frac{\boldsymbol{n}\cdot\boldsymbol{r}_{\xi^m\xi^m}}{g_{mm}} + \frac{\boldsymbol{n}\cdot\boldsymbol{r}_{\xi^n\xi^n}}{g_{nn}}\right]^{-1}, \tag{22}$$

where $\boldsymbol{n}$ is the unit normal to the surface, calculated from

$$\boldsymbol{n} = \frac{\boldsymbol{r}_{\xi^m}\times\boldsymbol{r}_{\xi^n}}{|\boldsymbol{r}_{\xi^m}\times\boldsymbol{r}_{\xi^n}|}\,. \tag{23}$$

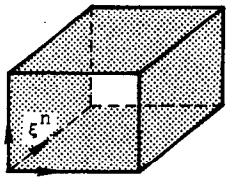

Fig. 13. Interpolation of A_n and s_n.

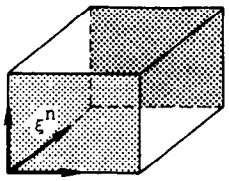

Fig. 14. Interpolation of ρ_n.

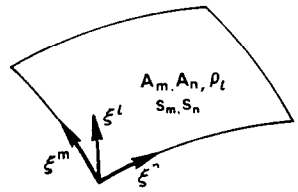

Fig. 15. Block side, varying ξ^m and ξ^n.

4.3. Block edges

On the edge of a block side in the 3D case, only the arc length contribution from (14) of Appendix B,

$$A_l = -\frac{\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^l\xi^l}}{g_{ll}} , \tag{24}$$

is evaluated. In 2D, the spacing,

$$s_l = \sqrt{g_{ll}} , \tag{25}$$

and the radius of curvature,

$$\rho_l = -\left[\frac{\boldsymbol{n} \cdot \boldsymbol{r}_{\xi^l\xi^l}}{g_{ll}}\right]^{-1} , \tag{26}$$

are also calculated on the line.

There are two normals to the line here, one obtained as

$$\boldsymbol{n}^{(1)} = \frac{\boldsymbol{r}_{\xi^n} \times \boldsymbol{r}_{\xi^l}}{|\boldsymbol{r}_{\xi^n} \times \boldsymbol{r}_{\xi^l}|} , \tag{27a}$$

and the other as

$$\boldsymbol{n}^{(2)} = \frac{\boldsymbol{r}_{\xi^l} \times \boldsymbol{r}_{\xi^m}}{|\boldsymbol{r}_{\xi^l} \times \boldsymbol{r}_{\xi^m}|} . \tag{27b}$$

Here central differences are used along the line and also in the two directions off the line if possible, with one-sided differences being used otherwise.

4.4. Interpolation

The arc length spacing, s_i, and the arc length contribution, A_i, to the control function are interpolated into the interior of the block from the four sides on which they are known by two-dimensional transfinite interpolation using linear blending functions, see Fig. 16.

The radius of curvature, ρ_i, is interpolated into the interior from the two sides on which it is known by one-dimensional interpolation using blending functions based on the hyperbolic sine as described in Appendix C, see Fig. 17.

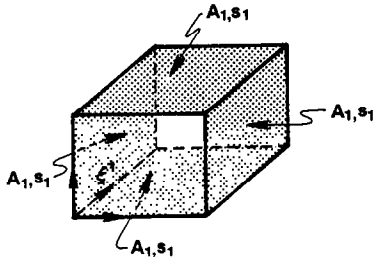

Fig. 16. Interpolation of s_i and A_i.

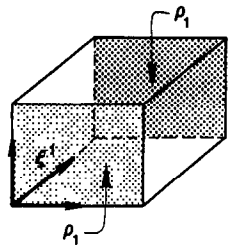

Fig. 17. Interpolation of ρ_i.

The control function is finally formed by adding the arc length spacing divided by the radius of curvature to the arc length contribution according to (19).

5. Iterative adjustment of control functions

A second-order elliptic generation system allows either the point locations on the boundary or the coordinate line slope at the boundary to be specified, but not both. It is possible, however, to iteratively adjust the control functions in the generation system of the Poisson type discussed above until not only a specified line slope, but also the spacing of the first coordinate surface off the boundary is achieved, with the point locations on the boundary specified (cf. [15]), see Fig. 18.

In three dimensions the specification of the coordinate line slope at the boundary requires the specification of two quantities, e.g., the direction cosines of the line with two tangents to the boundary, as shown in Fig. 19.

The specification of the spacing of the first coordinate surface off the boundary requires one more quantity, and therefore the three control functions in the system (8) are exactly sufficient to allow these three specified quantities to be achieved, while the one boundary condition allowed by the second-order system provides for the point locations on the boundary to be specified. The present procedure is a generalization of that of [15].

5.1. General development

To illustrate this development, an iterative procedure can be constructed for the determination of the control function in two dimensions as follows [15]. Consider the generation system

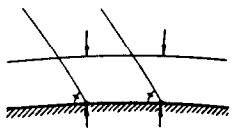

Fig. 18. Specified slope and spacing.

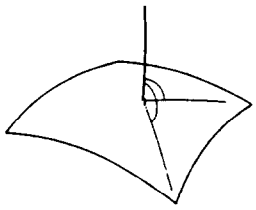

Fig. 19. Coordinate line slope at the boundary.

given by (8) in two dimensions (with $\xi^1 = \xi$, $\xi^2 = \eta$, $x_1 = x$, and $x_2 = y$ here). On a boundary segment that is a line of constant η we have $\boldsymbol{r}_\xi$ and $\boldsymbol{r}_{\xi\xi}$ known from the specified boundary point distribution, as shown in Fig. 20. Also $|\boldsymbol{r}_\eta|$, the spacing off this boundary, is specified, as shown in Fig. 21, as is the condition of orthogonality at the boundary, i.e., $\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\eta = 0$, as shown in Fig. 22. But specification of $|\boldsymbol{r}_\eta| = \sqrt{x_\eta^2 + y_\eta^2}$, together with the condition $\boldsymbol{r}_\xi \cdot \boldsymbol{r}_\eta = x_\xi x_\eta + y_\xi y_\eta = 0$, provides two equations for the determination of x_η and y_η in terms of the already known values of the x_ξ and y_ξ. Therefore, $\boldsymbol{r}_\eta$ is known on the boundary.

Because of the orthogonality at the boundary, (8) reduces to the following equation on the boundary:

$$|\boldsymbol{r}_\eta|^2(\boldsymbol{r}_{\xi\xi} + P\boldsymbol{r}_\xi) + |\boldsymbol{r}_\xi|^2(\boldsymbol{r}_{\eta\eta} + Q\boldsymbol{r}_\eta) = 0\,. \tag{28}$$

Dotting $\boldsymbol{r}_\xi$ and $\boldsymbol{r}_\eta$ into this equation, and again using the condition of orthogonality, yields the following two equations for the control functions on the boundary:

$$P = -\frac{\boldsymbol{r}_\xi \cdot \boldsymbol{r}_{\xi\xi}}{|\boldsymbol{r}_\xi|^2} - \frac{\boldsymbol{r}_\xi \cdot \boldsymbol{r}_{\eta\eta}}{|\boldsymbol{r}_\eta|^2}\,, \tag{29}$$

$$Q = -\frac{\boldsymbol{r}_\eta \cdot \boldsymbol{r}_{\eta\eta}}{|\boldsymbol{r}_\eta|^2} - \frac{\boldsymbol{r}_\eta \cdot \boldsymbol{r}_{\xi\xi}}{|\boldsymbol{r}_\xi|^2}\,. \tag{30}$$

All of the quantities in these equations are known on the boundary except $\boldsymbol{r}_{\eta\eta}$. (On a boundary that is a line of constant ξ, the same equations for the control functions result, but now with $\boldsymbol{r}_{\xi\xi}$ the unknown quantity.)

The iterative solution thus proceeds as follows:

Step (1) Assume values for the control function on the boundary.
Step (2) Solve (28) to generate the grid in the field.
Step (3) Evaluate $\boldsymbol{r}_{\eta\eta}$ on η-line boundaries, and $\boldsymbol{r}_{\xi\xi}$ on ξ-line boundaries, from the result of Step (2), using one-sided difference representations. Then evaluate the control functions on the boundary from (29) and (30). Evaluate the control functions in the field by interpolation from the boundary values.

Steps (2) and (3) are then repeated until convergence.

5.2. Implementation

An iterative solution procedure for the determination of the three control functions for the general three-dimensional case can be constructed as follows. Equation (B.18) gives the two

Fig. 20. Boundary point distribution.

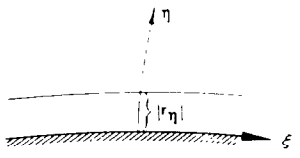

Fig. 21. Spacing off boundary.

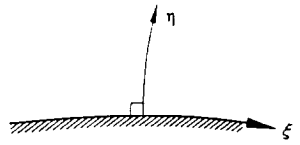

Fig. 22. Condition of orthogonality.

control functions, P_m and P_n, for a coordinate surface on which ξ^l is constant (l, m, n cyclic) for the case where the coordinate line crossing the surface is normal to the surface. Taking the projection of the generation equation (8) on the coordinate line along which ξ^l varies, we have on this same surface,

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\boldsymbol{r}_{\xi^l}\cdot\boldsymbol{r}_{\xi^i\xi^j} + g^{ll}g_{ll}P_l = 0\,,$$

since $g_{lm} = g_{ln}$ are to be zero on the surface. Proceeding as in Appendix B, this equation reduces to

$$P_l = -\frac{1}{g_{ll}}\boldsymbol{r}_{\xi^l}\cdot\boldsymbol{r}_{\xi^l\xi^k} - \frac{\boldsymbol{r}_{\xi^l}}{g_{mm}g_{nn}-g_{mn}^2}\cdot\left(g_{nn}\boldsymbol{r}_{\xi^m\xi^m} + g_{mm}\boldsymbol{r}_{\xi^n\xi^n} - 2g_{mn}\boldsymbol{r}_{\xi^m\xi^n}\right). \tag{31}$$

Since the coordinate line intersecting the surface is to be normal to the surface, we may write

$$\boldsymbol{r}_{\xi^l} = \boldsymbol{a}_l = \sqrt{g_{ll}}\,\frac{\boldsymbol{a}_m\times\boldsymbol{a}_n}{|\boldsymbol{a}_m\times\boldsymbol{a}_n|}\,,$$

since

$$|\boldsymbol{a}_m\times\boldsymbol{a}_n|^2 = g_{mm}g_{nn} - g_{mn}^2\,.$$

Equation (31) can then be written as

$$P_l = -\frac{(g_{ll})_{\xi^l}}{2g_{ll}} - \frac{\sqrt{g_{ll}}}{(g_{mm}g_{nn}-g_{mn}^2)^{3/2}}(\boldsymbol{a}_m\times\boldsymbol{a}_n) \cdot\left[g_{nn}(\boldsymbol{a}_m)_{\xi^m} + g_{mm}(\boldsymbol{a}_n)_{\xi^n} - 2g_{mn}(\boldsymbol{a}_m)_{\xi^n}\right]. \tag{32}$$

With the spacing along the coordinate line intersecting the surface specified at the surface, we have $|\boldsymbol{r}_{\xi^l}| = \sqrt{g_{ll}}$ known on the surface. Since all the quantities subscripted m or n in (B.18) and (32) can be evaluated completely from the specified point distribution on the surface, we then have all quantities in these equations for the three control functions on the surface known except for $(g_{ll})_{\xi^l}$ and $(\boldsymbol{a}_l)_{\xi^l}$. These two quantities are not independent, and using (31), we have,

$$(g_{ll})_{\xi^l} = 2\boldsymbol{a}_l \cdot (\boldsymbol{a}_l)_{\xi^l} = \frac{2\sqrt{g_{ll}}}{\sqrt{g_{mm}g_{nn} - g_{mn}^2}} (\boldsymbol{a}_m \times \boldsymbol{a}_n) \cdot (\boldsymbol{a}_l)_{\xi^l} . \tag{33}$$

Recall also that $(\boldsymbol{a}_l) = \boldsymbol{r}_{\xi^l\xi^l}$.

Therefore, with the control functions in the field determined from the values on the boundary by interpolation, as discussed in the preceeding section, (B.18) and (32) can be applied to determine the new boundary values of the control functions in terms of the new values of $(\boldsymbol{a}_l)_{\xi^l}$ in an interative solution. Upon convergence, the coordinate system will then have the coordinate lines intersecting the boundary normally at fixed locations and with the specified spacing on these lines off the boundary.

The three equations for the control functions on the surface can be written in the following notation, taking advantage of common terms,

$$P_m = -\boldsymbol{B}_1 \cdot \boldsymbol{A} , \tag{34a}$$

$$P_n = -\boldsymbol{B}_2 \cdot \boldsymbol{A} , \tag{34b}$$

$$P_l = -\boldsymbol{C} \cdot \boldsymbol{A} , \tag{34c}$$

where

$$\boldsymbol{A} = \boldsymbol{r}_{\xi^l\xi^l} + \boldsymbol{a} ,$$

$$\boldsymbol{a} = \frac{g_{ll}}{g_{mm}g_{nn} - g_{mn}^2} (g_{nn}\boldsymbol{r}_{\xi^m\xi^m} + g_{mm}\boldsymbol{r}_{\xi^n\xi^n} - 2g_{mn}\boldsymbol{r}_{\xi^m\xi^n}) ,$$

$$\boldsymbol{B}_1 = \frac{1}{g_{ll}} \left(\boldsymbol{r}_{\xi^m} - \frac{g_{mn}}{g_{nn}} \boldsymbol{r}_{\xi^n} \right) ,$$

$$\boldsymbol{B}_2 = \frac{1}{g_{ll}} \left(\boldsymbol{r}_{\xi^n} - \frac{g_{mn}}{g_{mm}} \boldsymbol{r}_{\xi^m} \right) ,$$

$$\boldsymbol{C} = \frac{\boldsymbol{r}_{\xi^m} \times \boldsymbol{r}_{\xi^n}}{\sqrt{g_{ll}(g_{mn}g_{nn} - g_{mn}^2)}} .$$

Here, all quantities except the $\boldsymbol{r}_{\xi^l\xi^l}$ that appears in $\boldsymbol{A}$ can be evaluated from the boundary point distribution and the specified off-boundary spacing. In previous applications the relations have been applied on the boundary and the control function increments at the boundary have been interpolated into the field [15]. In the present code, these relations are applied on each successive coordinate surface off the boundary, with the off-surface spacing determined by a hyperbolic sine distribution from the spacing specified at the boundary. The control function increments are attenuated away from the boundary, and contributions are accumulated from all orthogonal boundary sections.

Since the iterative adjustment of the control functions is a feedback loop, it is necessary to limit the acceleration parameter for stability as discussed in Appendix D.

6. Iterative solution

All of the first and second coordinate derivatives are first calculated, using second-order central differences, and are placed in an array DR(3, 0:3, 0:3) as follows:

$$(\boldsymbol{r}_{\xi^i})_l = \mathrm{DR}(l, i, 0) = \tfrac{1}{2}[\boldsymbol{r}(\xi^i + 1) - \boldsymbol{r}(\xi^i - 1)]_l ,$$

$$(\boldsymbol{r}_{\xi^i\xi^j})_l - 2\delta_{ij}\boldsymbol{r} = \mathrm{DR}(l, i, j) = [\boldsymbol{r}(\xi^i + 1) + \boldsymbol{r}(\xi^i - 1)]_l \quad \text{for } i = j ,$$

or

$$= \tfrac{1}{4}[\boldsymbol{r}(\xi^i + 1, \xi^j + 1) + \boldsymbol{r}(\xi^i - 1, \xi^j - 1) - \boldsymbol{r}(\xi^i + 1, \xi^j - 1) - \boldsymbol{r}(\xi^i - 1, \xi^j + 1)]_l \quad \text{for } i \neq j ,$$

where δ_{ij} is the Kroniker delta and only the arguments that vary have been shown. Next, the covariant metric elements are calculated and placed in the array $G(3, 3)$:

$$g_{ij} = \boldsymbol{r}_{\xi^i} \cdot \boldsymbol{r}_{\xi^j} = G(i, j) .$$

The products of the contravariant metric elements with the square of the Jacobian are placed in the array GG(3, 3):

$$gg^{il} = g_{jm}g_{kn} - g_{jn}g_{km} = \mathrm{GG}(i, l) , \quad (i, j, k), (l, m, n) \text{ cyclic.}$$

(In 2D, g^{33} is set to zero.)

The elliptic grid generation system is (8), restated here as

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\boldsymbol{r}_{\xi^i\xi^j} + \sum_{k=1}^{3} g^{kk}P_k\boldsymbol{r}_{\xi^k} = 0 ,$$

where the P_k are the control functions.

With central differences, the first derivatives in this equation are simply, see Fig. 23,

$$\boldsymbol{r}_{\xi^k} = \mathrm{DR}(-, k, 0) ,$$

where the first subscript gives the component (1, 2, or 3). Provision is also made for one-sided differences, dependent on the sign of the control function P_k (backward for $P_k < 0$ and forward for $P_k > 0$), see Fig. 24,

$$(\boldsymbol{r}_{\xi^k})_l = [\boldsymbol{r}(\xi^k) - \boldsymbol{r}(\xi^k - 1)]_l , \quad P_k < 0 ,$$

$$(\boldsymbol{r}_{\xi^k})_l = [\boldsymbol{r}(\xi^k + 1) - \boldsymbol{r}(\xi^k)]_l , \quad P_k > 0 .$$

The product $P_k\boldsymbol{r}_{\xi^k}$ can then be represented as

$$P_k\boldsymbol{r}_{\xi^k} = P_k\,\mathrm{DR}(-, k, 0) + A|P_k|[\tfrac{1}{2}\,\mathrm{DR}(-, k, k) - \boldsymbol{r}] ,$$

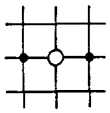

Fig. 23. Central differences.

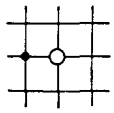

Fig. 24. One-sided differences.

with $A = 0$ for central differences and $A = 1$ for one-sided differences. The switching coefficient, A, can be variable:

$$A = \begin{cases} \frac{1}{2}|P_k|, & |P_k| < 2, \\ 1, & |P_k| \geqslant 2, \end{cases}$$

which creates a return to central differences when the control function vanishes.

Also, with central differences the cross derivatives are, see Fig. 25,

$$\boldsymbol{r}_{\xi^i\xi^j} = \mathrm{DR}(-, i, j),$$

while the skewed cross derivatives are given by

$$(\boldsymbol{r}_{\xi^i\xi^j})_l = \tfrac{1}{2}[\boldsymbol{r}(\xi^i + 1, \xi^j + 1) + \boldsymbol{r}(\xi^i - 1, \xi^j - 1) - \boldsymbol{r}(\xi^i + 1, \xi^j) - \boldsymbol{r}(\xi^i - 1, \xi^j) - \boldsymbol{r}(\xi^i, \xi^j + 1) - \boldsymbol{r}(\xi^i, \xi^j - 1) + 2\boldsymbol{r}(\xi^i, \xi^j)], \quad g^{ij} > 0,$$

and

$$(\boldsymbol{r}_{\xi^i\xi^j})_l = \tfrac{1}{2}[\boldsymbol{r}(\xi^i + 1, \xi^j - 1) + \boldsymbol{r}(\xi^i - 1, \xi^j + 1) - \boldsymbol{r}(\xi^i + 1, \xi^j) - \boldsymbol{r}(\xi^i - 1, \xi^j) - \boldsymbol{r}(\xi^i, \xi^j + 1) - \boldsymbol{r}(\xi^i, \xi^j - 1) + 2\boldsymbol{r}(\xi^i, \xi^j)], \quad g^{ij} < 0.$$

These can be incorporated into one form for the product $g^{ij}\boldsymbol{r}_{\xi^i\xi^j}$ as

$$g^{ij}\boldsymbol{r}_{\xi^i\xi^j} = g^{ij}\,\mathrm{DR}(-, i, j) + B|g^{ij}|\tfrac{1}{2}[\mathrm{DC}(-, i, j) - \mathrm{DR}(-, i, i) - \mathrm{DR}(-, j, j) + \boldsymbol{r}]_l,$$

with $B = 0$ for central differences and $B = 1$ for skewed differences and with

$$\mathrm{DC}(l, i, j) = \tfrac{1}{2}[\boldsymbol{r}(\xi^i + 1, \xi^j + 1) + \boldsymbol{r}(\xi^i - 1, \xi^j - 1) + \boldsymbol{r}(\xi^i + 1, \xi^j - 1) + \boldsymbol{r}(\xi^i - 1, \xi^j + 1)]_l,$$

for $i \neq j$.

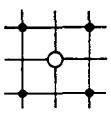

Fig. 25. Cross derivatives.

Fig. 26. Skewed cross derivatives.

The generation equation can then be written as

$$\begin{aligned}\Big[\sum_k \mathrm{GG}(k,k)(2+A|P_k|) - B\sum_{\substack{i\\(i\neq j)}}\sum_j |\mathrm{GG}(i,j)|\Big]\boldsymbol{r} \\ = \sum_k \mathrm{GG}(k,k)[(1+\tfrac{1}{2}A|P_k|)\mathrm{DR}(-,k,k) + P_k\mathrm{DR}(-,k,0)] \\ + \sum_{\substack{i\\(i\neq j)}}\sum_j \{\mathrm{GG}(i,j)\mathrm{DR}(-,i,j) \\ + \tfrac{1}{2}B|\mathrm{GG}(i,j)|[\mathrm{DC}(-,i,j) - \mathrm{DR}(-,i,i) - \mathrm{DR}(-,j,j)]\}\,, \end{aligned} \tag{35}$$

where $\mathrm{GG}(i,j)$ is g^{ij} times the square of the Jacobian.

This is implemented as

$$x_l = \frac{\mathrm{SUM1} + \mathrm{SUM2}}{\mathrm{SUM}}\,,$$

where

$$\mathrm{SUM} = \sum_k \mathrm{GG}(k,k)(2+A|P_k|) - B\sum_{\substack{i\\(i\neq j)}}\sum_j |\mathrm{GG}(i,j)|\,,$$

$$\mathrm{SUM1} = \sum_k \mathrm{GG}(k,k)[P_k\mathrm{DR}(-,k,0) + \tfrac{1}{2}A|P_k|\mathrm{DR}(-,k,k)]\,,$$

$$\begin{aligned}\mathrm{SUM2} = \sum_k \mathrm{GG}(k,k)\mathrm{DR}(-,k,k) + \sum_{\substack{i\\(i\neq j)}}\sum_j \{\mathrm{GG}(i,j)\mathrm{DR}(-,i,j) \\ + \tfrac{1}{2}B|\mathrm{GG}(i,j)|[\mathrm{DC}(-,i,j) - \mathrm{DR}(-,i,i) - \mathrm{DR}(-,j,j)]\}\,.\end{aligned}$$

The switching coefficient A is obtained from the relation

$$A = A_1 + A_2 \min(1, \tfrac{1}{2}|P_k|)\,,$$

where A_1 and A_2 assume the values depending on the differences as given in Table 1.

The values of the Cartesian coordinates given by this equation are taken as intermediate values, and the acceleration process yields the new values at the current iteration as

$$x_l^{\mathrm{new}} = \omega x_l + (1-\omega)x_l^{\mathrm{old}}\,,$$

where ω is the acceleration parameter.

Table 1
Values of A_1 and A_2

	Difference		
	Centered	One-sided	Variable
A_1	0	1	0
A_2	0	0	1

7. Acceleration parameters

In (8) the coefficient of the central value of $\boldsymbol{r}$ is,

$$a_0 = \sum_k g^{kk}(2 + A|P_k|) - B \sum_{\substack{i \\ (i \neq j)}} \sum_j |g^{ij}| ,$$

and the coefficients of $\boldsymbol{r}$ at $\xi^i + 1$ and at $\xi^i - 1$ on the right-hand side are

$$a^i_{\pm} = g^{ii}[(1 + \tfrac{1}{2}A|P_i|) \pm \tfrac{1}{2}P_i] - B(|g^{ij}| + |g^{ik}|) , \quad i = 1, 2, 3 ,$$

with (i, j, k) cyclic in this last equation. Also the product $a^i_+ a^i_-$ is given by

$$a^i_+ a^i_- = [g^{ii}(1 + \tfrac{1}{2}A|P_i|) - B(|g^{ij}| + |g^{ik}|)]^2 - \tfrac{1}{4}(g^{ii}P_i)^2 .$$

From [12], for the difference equation

$$\sum_i a^i_+ x(\xi^i + 1) + a^i_- x(\xi^i - 1) - a_0 x = 0 ,$$

(where only the functional dependence on the argument that varies from the central value is shown) a field of optimum acceleration parameters for accelerated point Gauss–Seidel iteration can be calculated as follows. The coefficients A and B that determine the form of the first and cross derivatives are as used in the present system.

The Jacobi eigenvalue (i.e., for point Jacobi iteration) with the largest magnitude is given by

$$\mu = \mu_r + \mathrm{i}\mu_\mathrm{i} = \frac{2}{a_0} \sum_i \sqrt{a^i_+ a^i_-} \cos \frac{1}{N^i + 1} ,$$

where N^i is the number of points in the i-direction involved in the solution. (The i on the left side here is, of course, $\sqrt{-1}$, and μ_i is the imaginary part of μ, while on the right, i indicates one of the three directions.)

The locally optimum acceleration parameter, ω, is then given by

$$\omega = -\tfrac{1}{2}[\bar{\omega} \mp \sqrt{\bar{\omega}^2 + 4\bar{\omega}}] ,$$

with the upper sign for $\alpha^2 > \beta$ and the lower for $\alpha^2 < \beta$, where

$$\begin{aligned} &\alpha = \mu_r^2 + \mu_\mathrm{i}^2 , \qquad \beta = \mu_r^2 - \mu_\mathrm{i}^2 , \\ &a = \alpha^2 - \beta^2 , \qquad b = \alpha^2 - \beta , \\ &c = \sqrt{a + b^2} , \\ &\bar{\omega} = \frac{1}{\alpha^2 b} \{[(3b + c)(c - b)^{1/3} - (3b - c)(c + b)^{1/3}]a^{1/3} + \alpha^2 + 3\beta^2 - 4\alpha^2\beta\} . \end{aligned} \tag{36}$$

8. Applications

This elliptic generation system has been incorporated in a general three-dimensional code [19] based on a composite multiblock structure with complete continuity across the block interfaces. A section (all or part) of any block side can be paired with another section on the same or another block to form a branch cut with complete continuity, or can be designated for boundary orthogonality through the iterative adjustment of control functions, as discussed here, or simply through the application of Neumann boundary conditions. In the latter case the boundary points move on a surface spline.

The evaluation of this system continues, of course, as more different configurations are treated, and some early results were reported in [21].

While the evaluation of the control functions by interpolation from the boundary point distribution gives good results in many cases, for configurations having part, but not all, of a block side set up as a cut, there can be problems near concave regions on the edges of the cut. (Figure 27 shows one side of a multiple block configuration. The central portion of the lower half is a fixed boundary; the rest is a cut.) By contrast, evaluation from the algebraic grid followed by smoothing gives good resolution in such areas, as shown in Fig. 28. The effect of the smoothing of the control functions is evident upon comparison of these figures with the algebraic grid given in Fig. 29. The evaluation from the algebraic grid with the nonorthogonality terms dropped gives good results in some cases but not in others, as is shown in Fig. 30.

The effect of the iteratve adjustment of the control functions for boundary orthogonality is shown in Fig. 31, in comparison with the algebraic grid shown in Fig. 32.

In general, the following details have been found to be advantageous. During the iteraton, cuts on block sides are updated immediately after the block has been swept, since updating all of the cuts together after all of the blocks are swept can lead to oscillations near the cuts. The SOR iteration is implemented in a symmetric manner, reversing the sweep direction after each iteration since this gives better symmetry with Neumann boundary conditions. The optimum acceleration parameters are essential to making the system robust. When the control functions

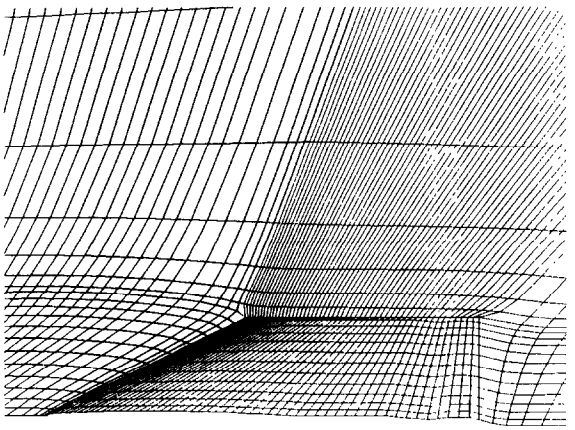

Fig. 27. Control functions from boundary point distribution.

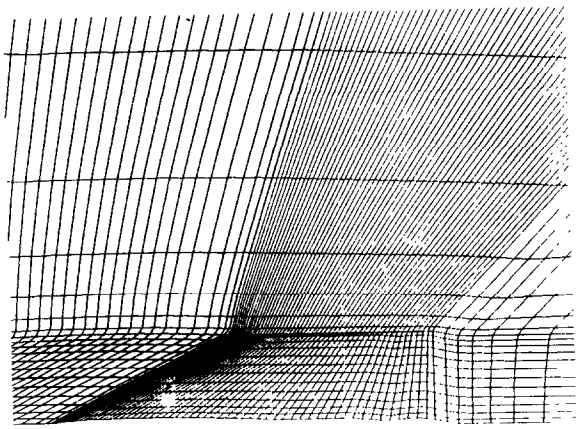

Fig. 28. Control functions from algebraic grid with smoothing.

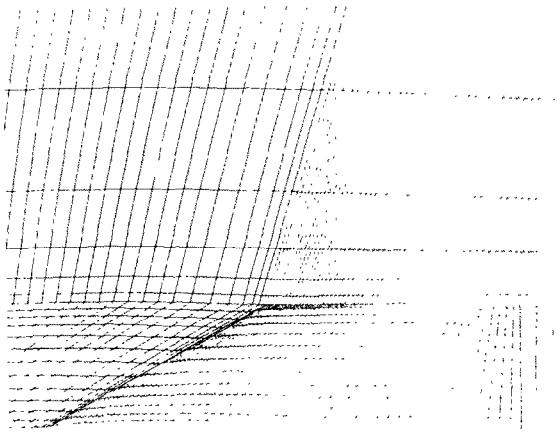

Fig. 29. Algebraic grid.

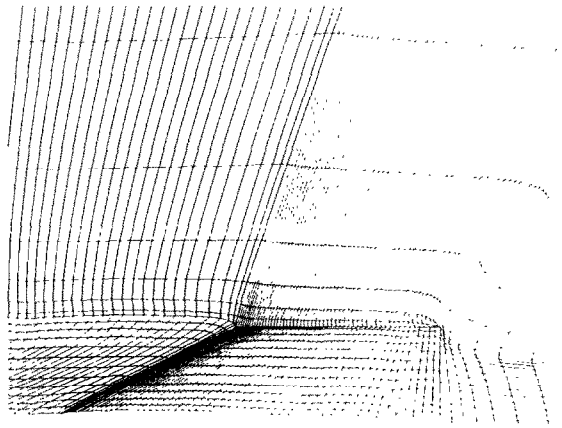

Fig. 30. Control functions from algebraic grid without nonorthogonality terms.

are iteratively adjusted for boundary orthogonality, the use of one-sided, directed first derivatives is appropriate since the changes in the control functions can initially be quite large. Central differences are used in all other cases. The skewed cross derivatives, however, have shown little value. Finally, the evaluation of the control functions from the algebraic grid, followed by smoothing, has proved to be the most generally applicable approach, particularly in complicated configurations.

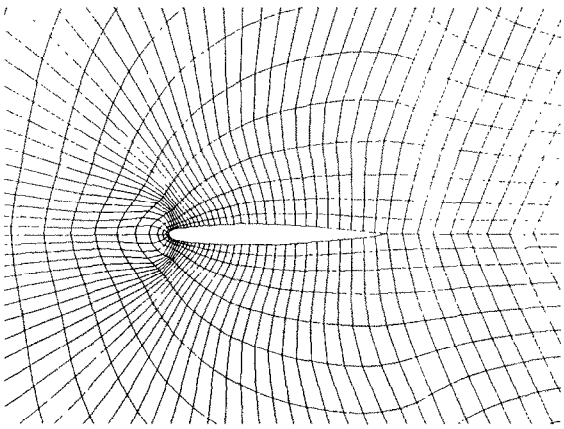

Fig. 31. Grid with iterative adjustment of the control functions.

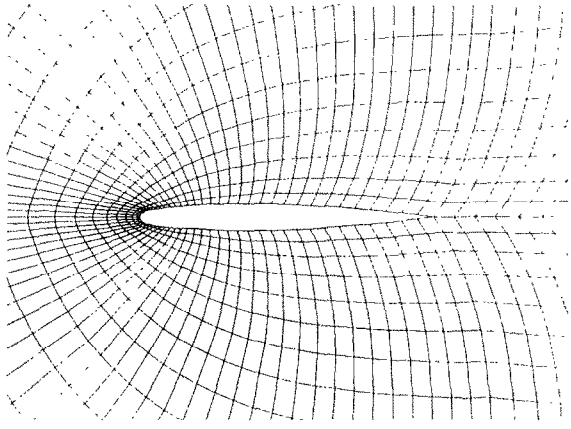

Fig. 32. Algebraic grid.

Appendix A. Distribution functions

In general, interpolation between $\boldsymbol{r}_1$ at $\xi = 0$ and $\boldsymbol{r}_2$ at $\xi = I$ can be written as

$$\boldsymbol{r}(\xi) = \varphi(\xi/I)\boldsymbol{r}_2 + [1 - \varphi(\xi/I)]\boldsymbol{r}_1 , \tag{A.1}$$

where φ can be any function such that $\varphi(0) = 0$ and $\varphi(1) = 1$. The linear polynomial case is obtained here with $\varphi(\xi/I) = \xi/I$. The function φ in this form may contain parameters which can be determined so as to match the slope at the boundary, or to match interior points and slopes.

The interpolation function, φ, in this form is often referred to as a 'stretching' function, and the most widely used function has been the exponential:

$$\varphi(\xi/I) = \frac{\exp(\alpha\xi/I) - 1}{\exp(\alpha) - 1} , \tag{A.2}$$

where α is a parameter that can be determined to match the slope at a boundary. Thus, since from (A.1),

$$\boldsymbol{r}_\xi = \frac{1}{I}(\boldsymbol{r}_2 - \boldsymbol{r}_1)\varphi' , \tag{A.3}$$

we can determine α from the equation,

$$(\boldsymbol{r}_\xi)_1 = \frac{\boldsymbol{r}_2 - \boldsymbol{r}_1}{I} \frac{\alpha}{\exp(\alpha) - 1} , \tag{A.4}$$

with $(\boldsymbol{r}_\xi)_1$ specified.

The truncation error is strongly affected by the point distribution, and studies of distribution functions have been made in that regard. The exponential, while reasonable, is not the best choice when the variation of spacing is large, and polynomials are not suitable in this case. The better choices are the hyperbolic tangent and the hyperbolic sine. The hyperbolic sine gives a more uniform distribution in the immediate vicinity of the minimum spacing, and thus has less error in this region, but the hyperbolic tangent has the better overall distribution [17, 18]. These functions are implemented as follows (following [17]), with the spacing specified at either or both ends, or a point in the interior, of a point distribution on a curve.

Let the arc length, s, vary from 0 to 1 as ξ varies from 0 to I: $s(0) = 0$, $s(I) = 1$. Then let the spacing be specified at $\xi = 0$ and $\xi = I$:

$$s_\xi(0) = \Delta s_1 , \qquad s_\xi(I) = \Delta s_2 . \tag{A.5}$$

The hyperbolic tangent distribution is then constructed as follows.

First,

$$A = \sqrt{\Delta s_2}/\sqrt{\Delta s_1}\,, \tag{A.6}$$

$$B = I\sqrt{\Delta s_1 \Delta s_2}\,. \tag{A.7}$$

Then the following nonlinear equation is solved for δ:

$$\frac{\sinh \delta}{\delta} = \frac{1}{B}\,, \tag{A.8}$$

The arc length distribution then is given by

$$s(\xi) = \frac{u(\xi)}{A + (1-A)u(\xi)}\,, \tag{A.9}$$

where

$$u(\xi) = \tfrac{1}{2}\left\{1 + \frac{\tanh[\delta(\xi/I - \tfrac{1}{2})]}{\tanh(\tfrac{1}{2}\delta)}\right\}. \tag{A.10}$$

If this is applied to a straight line on which $\boldsymbol{r}$ varies from $\boldsymbol{r}_0$ to $\boldsymbol{r}_I$, we have for the point locations:

$$\boldsymbol{r}(\xi) = \boldsymbol{r}_0 + (\boldsymbol{r}_I - \boldsymbol{r}_0)s(\xi)\,. \tag{A.11}$$

The points are then located by taking integer values of ξ:

$$\xi = 0, 1, 2, \ldots, I\,.$$

Clearly, the arc length distribution, $s(\xi)$, is here the function φ of (A.1).

Note that B is essentially the ratio of the specified spacing to the linear spacing, $1/I$. If B is greater than unity, i.e. if the specified spacing exceeds the linear spacing, the hyperbolic functions all revert to circular functions in all the relations of this appendix.

With the spacing Δs specified at only $\xi = 0$, the construction proceeds as follows. First B is calculated from

$$B = I\,\Delta s\,, \tag{A.12}$$

and (A.8) is solved for δ. The arc length distribution then is given by

$$s(\xi) = 1 + \frac{\tanh[\tfrac{1}{2}\delta(\xi/I - 1)]}{\tanh(\tfrac{1}{2}\delta)}\,. \tag{A.13}$$

With the spacing specified only at $\xi = I$ the procedure is the same, except that (A.13) is replaced by

$$s(\xi) = \frac{\tanh(\tfrac{1}{2}\delta\xi/I)}{\tanh(\tfrac{1}{2}\delta)}\,. \tag{A.14}$$

If the spacing Δs is specified at only an interior point $s = \sigma$, B is again calculated from (A.12), and then δ is determined as the solution of

$$1 + \left(\frac{B}{\sigma\delta}\right)^2 = \left(\frac{\cosh\delta - 1 + 1/\sigma}{\sinh\delta}\right)^2 . \tag{A.15}$$

The value of ξ at which $s = \sigma$ is obtained by solving the nonlinear equation,

$$\chi = \frac{I}{\delta}\tanh^{-1}\left(\frac{\sinh\delta}{1/\sigma + \cosh\delta - 1}\right). \tag{A.16}$$

The arc length distribution then is given by

$$s(\xi) = \sigma\left\{\frac{1 + \sinh[\delta(\xi - \chi)/I]}{\sinh(\delta\chi/I)}\right\}. \tag{A.17}$$

This last distribution is based on the hyperbolic sine. From this, a distribution based on the hyperbolic sine with the spacing specified at one end can be derived. Here B is evaluated from (A.12), and then δ is determined as the solution of,

$$\frac{\sinh\delta}{\delta} = \frac{1}{B} . \tag{A.18}$$

The arc length distribution then is given by,

$$s(\xi) = \frac{\sinh(\delta\xi/I)}{\sinh\delta} , \tag{A.19}$$

if the spacing is specified at $\xi = 0$. With the specification at $\xi = I$, the distribution is

$$s(\xi) = 1 - \frac{\sinh[\delta(1 - \xi/I)]}{\sinh\delta} . \tag{A.20}$$

Appendix B. Surface control functions

B.1. Evaluation along a coordinate line

The projection of (8) along a coordinate line on which ξ^l varies is found by forming the dot product of this equation with the covariant base vector $\boldsymbol{a}_l = \boldsymbol{r}_{\xi^l}$, which is tangent to the line, see Fig. B.1.

Thus we have,

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\boldsymbol{r}_{\xi^l}\cdot\boldsymbol{r}_{\xi^i\xi^j} + \sum_{k=1}^{3} g^{kk}P_k\boldsymbol{r}_{\xi^l}\cdot\boldsymbol{r}_{\xi^k} = 0 . \tag{B.1}$$

Now assume for the moment that the two coordinate lines crossing the coordinate line of interest do so orthogonally, see Fig. B.2. Then on this line we have,

$$\boldsymbol{r}_{\xi^l}\cdot\boldsymbol{r}_{\xi^k} = \boldsymbol{a}_l\cdot\boldsymbol{a}_k = g_{lk} = \delta_{lk}g_{ll} ,$$

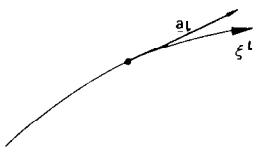

Fig. B.1. Evaluation along a coordinate line.

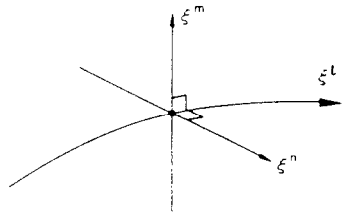

Fig. B.2. Orthogonal crossing of the line of interest.

which leads to an explicit equation for P_l on the coordinate line of interest:

$$P_l = -\frac{1}{g^{ll}g_{ll}}\, \boldsymbol{r}_{\xi^l} \cdot \sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\boldsymbol{r}_{\xi^i\xi^j} . \tag{B.2}$$

If it is further assumed for the moment that the two coordinate lines crossing the coordinate line of interest are also orthogonal to each other, i.e., complete orthogonality on the line of interest, see Fig. B.3, we have on this line $g^{ij} = \delta_{ij}g^{ii}$ and $g_{ij} = \delta_{ij}g_{ii}$. Also,

$$g^{ll} = \frac{1}{g}\,(g_{mm}g_{nn} - g_{mn}^2) = \frac{1}{g}\, g_{mm}g_{nn} , \quad (l, m, n) \text{ cyclic} ,$$

since $m \neq n$. But also $g = g_{ll}g_{mm}g_{nn}$ so that $g^{ll}g_{ll} = 1$. Then (B.2) becomes

$$P_l = -\sum_{i=1}^{3} \frac{1}{g_{ii}}\, \boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^i\xi^i} , \quad l = 1, 2, 3 , \tag{B.3}$$

which can also be written as,

$$P_l = -\sum_{i=1}^{3} \frac{1}{g_{ii}}\, \boldsymbol{a}_l \cdot (\boldsymbol{a}_i)_{\xi^i} . \tag{B.4}$$

The derivative of arc length along the coordinate line on which ξ^l varies is

$$s_{\xi^l} = |\boldsymbol{r}_{\xi^l}| . \tag{B.5}$$

Then,

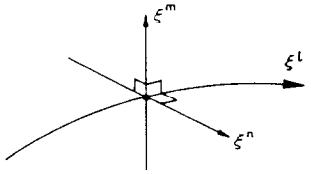

Fig. B.3. Complete orthogonality on the line of interest.

$$s_{\xi^l\xi^l} = |\boldsymbol{r}_{\xi^l}|_{\xi^l} = \frac{\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^l\xi^l}}{|\boldsymbol{r}_{\xi^l}|} , \tag{B.6}$$

so that the logarithmic derivative of arc length along this coordinate line is given by

$$S_l = \frac{s_{\xi^l\xi^l}}{s_{\xi^l}} = \frac{\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^l\xi^l}}{|\boldsymbol{r}_{\xi^l}|^2} = \frac{\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^l\xi^l}}{g_{ll}} , \tag{B.7}$$

which is exactly the term $i = l$ in the summation in (B.3).

The unit tangent to a coordinate line on which ξ^m varies is

$$\boldsymbol{T}^m = \boldsymbol{r}_{\xi^m}/|\boldsymbol{r}_{\xi^m}| = \boldsymbol{a}_m/|\boldsymbol{a}_m| , \tag{B.8}$$

and the derivative of this unit tangent with respect to arc length is a vector that is normal to this line, the magnitude of which is the curvature, K, of the line. The unit vector in this normal direction is the principal normal, N, to the line, see Fig. B.4.

Thus, using (B.5),

$$K^m N^m = (\boldsymbol{T}^m)_s = (\boldsymbol{T}^m)_{\xi^m} \xi^m_s = (\boldsymbol{T}^m)_{\xi^m}/|\boldsymbol{r}_{\xi^m}| . \tag{B.9}$$

Then,

$$K^m N^m = \frac{1}{|\boldsymbol{r}_{\xi^m}|} \left(\frac{\boldsymbol{r}_{\xi^m}}{|\boldsymbol{r}_{\xi^m}|} \right)_{\xi^m}$$

$$= \frac{|\boldsymbol{r}_{\xi^m}|^2 \boldsymbol{r}_{\xi^m\xi^m} - (\boldsymbol{r}_{\xi^m} \cdot \boldsymbol{r}_{\xi^m\xi^m}) \boldsymbol{r}_{\xi^m}}{|\boldsymbol{r}_{\xi^m}|^4} , \tag{B.10}$$

so that the curvature is

$$K^m = \frac{1}{|\boldsymbol{r}_{\xi^m}|^3} |\boldsymbol{r}_{\xi^m}|^2 |\boldsymbol{r}_{\xi^m\xi^m}|^2 - (\boldsymbol{r}_{\xi^m} \cdot \boldsymbol{r}_{\xi^m\xi^m})^2 . \tag{B.11}$$

The component of $K^m N^m$ along the coordinate line on which ξ^l varies is given by, see also Fig. B.5,

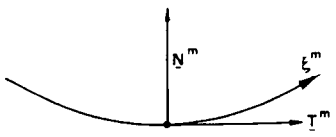

Fig. B.4. Unit tangent and normal.

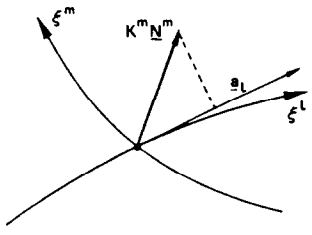

Fig. B.5. The component of $K^m N^m$ along the coordinate line.

$$(K^m N^m)^{(l)} = K^m N^m \frac{\boldsymbol{r}_{\xi^l}}{|\boldsymbol{r}_{\xi^l}|} = \frac{\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^m \xi^m}}{|\boldsymbol{r}_{\xi^l}||\boldsymbol{r}_{\xi^m}|^2} , \tag{B.12}$$

since $\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^m} = 0$ for $m \neq l$. Then the two terms of the summation in (B.3) for which $m \neq l$ can be written as

$$\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^m \xi^m} / g_{mm} = |\boldsymbol{r}_{\xi^l}|(K^m N^m)^{(l)} = \sqrt{g_{ll}}(K^m N^m)^{(l)} , \quad m \neq l . \tag{B.13}$$

Thus (B.3) can be written as

$$P_l = -S_l - \sqrt{g_{ll}}[(K^m N^m)^{(l)} + (K^n N^n)^{(l)}] , \quad l = 1, 2, 3 , \tag{B.14}$$

where (l, m, n) are cyclic.

The first term in this equation, $-S_l$, is the arc length contribution to the control function, $\sqrt{g_{ll}}$ is the spacing, and the two terms in the bracket (with the minus sign) are the curvature contributions.

It should be noted that the assumptions of orthogonality, and perhaps vanishing curvature, that were made in the course of the development of these expressions for the control functions on a coordinate line are not actually enforced on the resulting coordinate system, but merely served to allow some reasonable relations for these control functions corresponding to a specified point distribution on a coordinate line to be developed. This should not be considered a source of error since the control functions are arbitrary in the generation system (8).

B.2. Evaluation on a coordinate surface

In a similar fashion, expressions for the control functions on a coordinate surface on which ξ^l is constant can be obtained from the projections of (8) along the two coordinate lines lying on the surface, i.e., the lines on which ξ^m and ξ^n vary, (l, m, n) being cyclic, see Fig. B.6.

These projections are given by (B.1) with l replaced by m and n, respectively. If it is assumed for the moment that the coordinate line crossing the coordinate surface of interest is orthogonal to the surface, see Fig. B.7, then $\boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^m} = \boldsymbol{r}_{\xi^l} \cdot \boldsymbol{r}_{\xi^n} = 0$, so that P_l is removed from both of these two equations to yield the equation,

$$\sum_{i=1}^{3} \sum_{j=1}^{3} g^{ij} \boldsymbol{r}_{\xi^m} \cdot \boldsymbol{r}_{\xi^i \xi^j} + g^{mm} g_{mm} P_m + g^{nn} g_{nn} P_n = 0 , \tag{B.15}$$

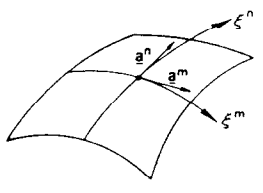

Fig. B.6. Projection onto two coordinate lines.

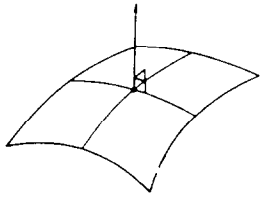

Fig. B.7. Coordinate line crossing the surface.

and an analogous equation with m and n interchanged. Solution of these two equations for P_m and P_n then yields

$$P_m = \frac{-(g_{nn}\boldsymbol{r}_{\xi^m} - g_{mn}\boldsymbol{r}_{\xi^n}) \cdot \sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\boldsymbol{r}_{\xi^i\xi^j}}{g^{mm}(g_{mm}g_{nn} - g_{mn}^2)}, \tag{B.16}$$

with an analogous equation for P_n with m and n interchanged. Since $g_{lm} = g_{ln} = 0$, it follows that $g^{lm} = g^{ln} = 0$. Therefore only the five terms, ll, mm, nn, mn, nm are nonzero in the summation. Also,

$$g^{mm} = \frac{g_{nn}}{g_{mm}g_{nn} - g_{mn}^2},$$

since here $g = \det|g_{ij}| = g_{ll}(g_{mm}g_{nn} - g_{mn}^2)$. An analogous equation for g^{nn} is obtained by interchanging m and n.

Then (B.16) can be rewritten as

$$P_m = -\left(\boldsymbol{r}_{\xi^m} - \frac{g_{mn}}{g_{nn}}\boldsymbol{r}_{\xi^n}\right) \cdot \left(\sum_{i=1}^{3} g^{ii}\boldsymbol{r}_{\xi^i\xi^i} + 2g^{mn}\boldsymbol{r}_{\xi^m\xi^n}\right), \tag{B.17}$$

and an analogous equation for P_n with m and n interchanged. But,

$$g^{ll} = \frac{1}{g_{ll}} \quad \text{and} \quad g^{mn} = \frac{-g_{mn}}{g_{mm}g_{nn} - g_{mn}^2}.$$

Therefore,

$$\begin{aligned} P_m = &-\frac{1}{g_{ll}}\left(\boldsymbol{r}_{\xi^m} - \frac{g_{mn}}{g_{nn}}\boldsymbol{r}_{\xi^n}\right) \cdot \boldsymbol{r}_{\xi^l\xi^l} \\ &-\left(\frac{\boldsymbol{r}_{\xi^m} - \dfrac{g_{mn}}{g_{nn}}\boldsymbol{r}_{\xi^n}}{g_{mm}g_{nn} - g_{mn}^2}\right) \cdot (g_{nn}\boldsymbol{r}_{\xi^m\xi^m} + g_{mm}\boldsymbol{r}_{\xi^n\xi^n} - 2g_{mn}\boldsymbol{r}_{\xi^m\xi^n}), \end{aligned} \tag{B.18}$$

and an analogous equation with m and n interchanged.

All of the terms, except the first, in the above equations can be evaluated completely from the point distribution on the coordinate surface of interest.

Appendix C. Interpolation for radius of curvature

The radius of curvature in the expressions for control functions requires some care in order to ensure generality. The construction of the control functions from arc length and curvature contributions arises from consideration of concentric spheres. Therefore, the baseline distribu-

tion for the radius of curvature is linear. However, in general configurations, the two surfaces between which the radius is interpolated may have greatly differing curvatures which may even be of opposite signs. Flat surfaces have infinite radius of curvature, and such surfaces will occur in the field between two surfaces with curvature of opposite signs, as well as on boundaries. Therefore, simple linear interpolation for the radius is out of the question in general, but must still be approached in many cases. The construction used is as follows.

The interpolation is between the two ends of a curvilinear coordinate line between two surfaces, as shown in Fig. C.1. This interpolation is based on the hyperbolic sine function and depends on the sign of the curvature on the two surfaces.

In the following discussion, the radius of curvature of the surface at the lower value of ξ^l is ρ_1, and that on the other surface is ρ_2. The arc length of the curvilinear coordinate line between the surfaces is A, and the relative arc length (range 0–1) at the point on this line where the radius of curvature is interpolated is s.

The interpolation is constructed as a departure from the linear distribution that would prevail between two spherical surfaces. The departure from linearity is made to be slight for some distance from the surface with the smaller radius in order to allow for flat surfaces.

When both surfaces have positive curvature, see Fig. C.2, the construction is as follows. If ρ_1 is the smaller, and ρ_2 is equal to the value that would occur for two spherical surfaces separated by a distance equal to the arc length A of the connecting coordinate line, then the distribution is linear:

$$\rho(s) = \rho_1 + As\ .$$

If ρ_2 exceeds $\rho_1 + A$, the distribution is

$$\rho(s) = (\rho_1 + As) + f(s)\ ,$$

where $f(0) = 0$, $f(1) = \rho_2 - (\rho_1 + A)$, and $f_s(0) = 0$. The function $f(s)$ is made to vary very slowly away from its initial value of zero by using the hyperbolic sine:

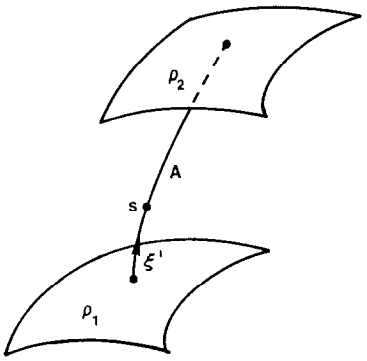

Fig. C.1. Interpolation between two surfaces.

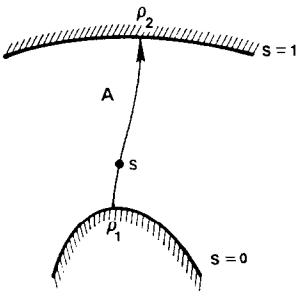

Fig. C.2. Two surfaces with positive curvature.

$$f(s) = A\left(\frac{\sinh \delta s}{\delta} - s\right),$$

where the parameter δ is determined by the condition on $f(1)$:

$$\frac{\sinh \delta}{\delta} = \frac{\rho_2 - \rho_1}{A}.$$

This construction is diagrammed in Fig. C.3.

Substituting for $f(s)$ in $\rho(s)$, the distribution for this case is then given by

$$\rho(s) = \rho_1 + (\rho_2 - \rho_1)\frac{\sinh \delta s}{\sinh \delta}.$$

If, however, ρ_2 is less than $\rho_1 + A$, the construction is similar, but with

$$\rho(s) = (\rho_1 + As) - f(s),$$

and now $f(1) = (\rho_1 + As) - \rho_2$, so that δ is determined by

$$\frac{\sinh \delta}{\delta} = \frac{\rho_1 - \rho_2}{A} + 2.$$

This diagram is shown in Fig. C.4.

Substitution for $f(s)$ yields the distribution,

$$\rho(s) = \rho_1 + 2As - (\rho_1 - \rho_2 + 2A)\frac{\sinh \delta s}{\sinh \delta}.$$

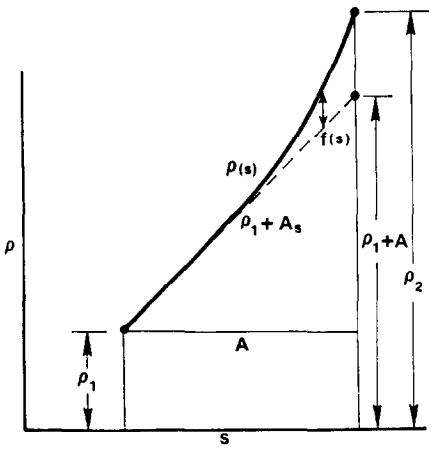

Fig. C.3. Construction with $\rho_2 > \rho_1 + A$.

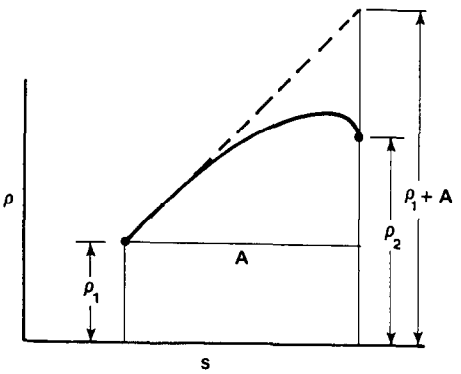

Fig. C.4. Construction with $\rho_2 < \rho_1 + A$.

Note that this form applies even for $\rho_2 < \rho_1$, which occurs, for example, for a circle inside a box, as shown in Fig. C.5, where the radius of curvature at the corner is much less than that on the circle.

If both surfaces have negative curvature, see Fig. C.6, the construction is the same, but with ρ_1 in the above replaced by $|\rho_2|$, ρ_2 replaced by $|\rho_1|$, and with s replaced by $1 - s$. Thus, with $|\rho_1| > |\rho_2| + A$, we have,

$$\rho(s) = -\left[|\rho_2| + (|\rho_1| - |\rho_2|)\,\frac{\sinh \delta(1-s)}{\sinh \delta}\right],$$

with δ determined by,

$$\frac{\sinh \delta}{\delta} = \frac{|\rho_1| - |\rho_2|}{A},$$

while with $|\rho_1| < |\rho_2| + A$, the distribution is

$$\rho(s) = -\left[|\rho_2| + 2A(1-s) - (|\rho_2| - |\rho_1| + 2A)\,\frac{\sinh \delta(1-s)}{\sinh \delta}\right],$$

with δ from

$$\frac{\sinh \delta}{\delta} = \frac{|\rho_2| - |\rho_1|}{A} + 2.$$

The cases with different curvature on the two surfaces require that the radius of curvature pass through infinity, i.e., that there be a flat surface somewhere in between. With $\rho_1 > 0$ and $\rho_2 < 0$, the location of this flat surface is calculated as

$$s_0 = 1 - \tfrac{1}{2}|\rho_1/\rho_2|$$

if $\rho_1 < |\rho_2|$, see Fig. C.7, and by

$$s_0 = \tfrac{1}{2}|\rho_2/\rho_1|$$

if $\rho_1 > |\rho_2|$, see Fig. C.8.

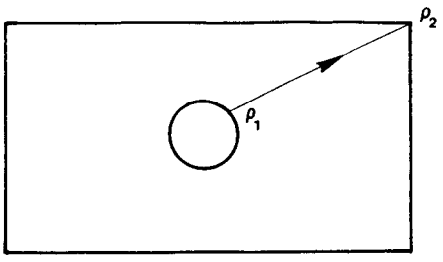

Fig. C.5. Circle in a box.

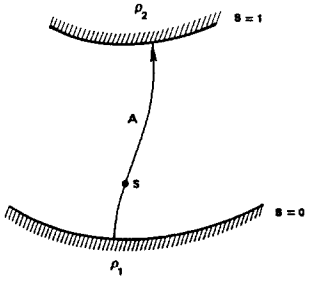

Fig. C.6. Two surfaces with negative curvature.

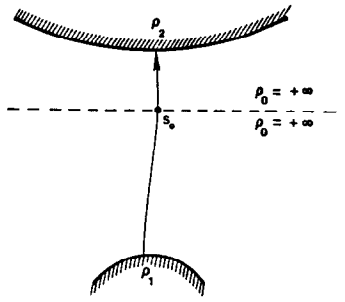

Fig. C.7. Interpolation with $\rho_1 > 0$, $\rho_2 < 0$, and $\rho_1 < |\rho_2|$.

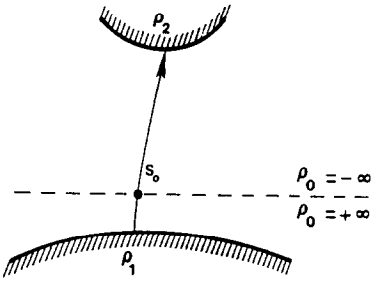

Fig. C.8. Interpolation with $\rho_1 > 0$, $\rho_2 < 0$, and $\rho_1 > |\rho_2|$.

The distribution is then constructed as above, but now separately between ρ_1 and ρ_0 and between ρ_0 and ρ_2 using a coordinate scale for $\pm\infty$.

If $\rho_1 < 0$ and $\rho_2 > 0$, the flat surface location is calculated from

$$s_0 = 1 - \tfrac{1}{2}|\rho_1/\rho_2|$$

if $\rho_2 > |\rho_1|$, see Fig. C.9, or from

$$s_0 = \tfrac{1}{2}|\rho_2/\rho_1|$$

if $\rho_2 < |\rho_1|$, see Fig. C.10.

Again, the distribution is constructed separately between ρ_1 and ρ_0, and between ρ_0 and ρ_2.

Appendix D. Feedback limitation on acceleration parameters

Since the iterative adjustment of the control functions for boundary orthogonality is a feedback loop, the SOR acceleration parameters must be limited for stability. This arises as follows.

From the generation equation (8), since the central terms of the second derivatives are

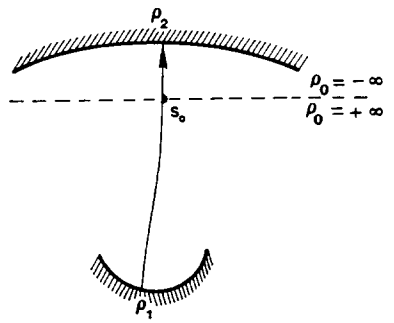

Fig. C.9. Interpolation with $\rho_1 < 0$, $\rho_2 > 0$, and $\rho_2 > |\rho_1|$.

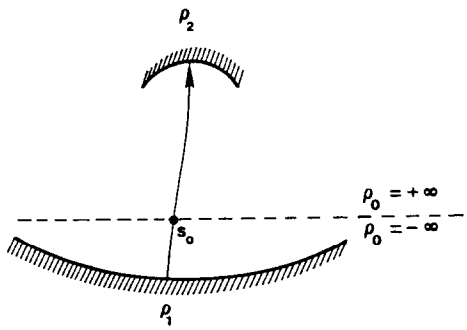

Fig. C.10. Interpolation with $\rho_1 < 0$, $\rho_2 > 0$, and $\rho_2 < |\rho_1|$.

factored to the left for the iterative solution, the change in $\boldsymbol{r}$ due to a change in the control function is approximately

$$\Delta \boldsymbol{r} \sim \omega \frac{\sum_k g^{kk} \Delta P_k \boldsymbol{r}_{\xi^k}}{2 \sum_k g^{kk}},$$

where ω is the acceleration parameter. Then,

$$\boldsymbol{r}_{\xi^i} \cdot \Delta \boldsymbol{r} \sim \frac{\sum_k g^{kk} g_{ik} \Delta P_k}{2 \sum_k g^{kk}}, \quad i = 1, 2, 3.$$

Also, from (34), the change in the control functions due to the change in $\boldsymbol{r}_{ll}$ that occurs when orthogonality is invoked is

$$\Delta P_l \sim \frac{\boldsymbol{n}}{\sqrt{g_{ll}}} \cdot \Delta \boldsymbol{r} \sim \frac{\boldsymbol{r}_{\xi^l}}{g_{ll}} \cdot (2\Delta \boldsymbol{r}),$$

$$\Delta P_m \sim \frac{1}{g_{ll}} \left(\boldsymbol{r}_{\xi^m} - \frac{g_{mn}}{g_{nn}} \boldsymbol{r}_{\xi^n} \right) \cdot (2\Delta \boldsymbol{r}),$$

$$\Delta P_n \sim \frac{1}{g_{ll}} \left(\boldsymbol{r}_{\xi^n} - \frac{g_{mn}}{g_{mm}} \boldsymbol{r}_{\xi^m} \right) \cdot (2\Delta \boldsymbol{r}),$$

where $\boldsymbol{n}$ is the unit normal.

Now, since $g_{lm} = g_{ln} = 0$, we have

$$\begin{bmatrix} \boldsymbol{r}_{\xi^l} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^m} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^n} \cdot \Delta \boldsymbol{r} \end{bmatrix} \sim \frac{\omega}{2 \sum_k g^{kk}} \begin{bmatrix} g^{ll} g_{ll} \Delta P_l \\ g^{mm} g_{mm} \Delta P_m + g^{nn} g_{mn} \Delta P_n \\ g^{mm} g_{mn} \Delta P_m + g^{nn} g_{nn} \Delta P_n \end{bmatrix}$$

$$= \frac{\omega}{2 \sum_k g^{kk}} \begin{bmatrix} \Delta P_l \\ \frac{g_{nn} g_{mm}}{G} \Delta P_m + \frac{g_{mm} g_{mn}}{G} \Delta P_n \\ \frac{g_{nn} g_{mn}}{G} \Delta P_m + \frac{g_{mm} g_{nn}}{G} \Delta P_n \end{bmatrix},$$

with $G = g_{mm} g_{nn} - g_{mn}^2$. Substitution for the changes in the control functions then yields, after some cancellations,

$$\begin{bmatrix} \boldsymbol{r}_{\xi^l} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^m} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^n} \cdot \Delta \boldsymbol{r} \end{bmatrix} \sim \frac{\omega g^{ll}}{\sum_k g^{kk}} \begin{bmatrix} \boldsymbol{r}_{\xi^l} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^m} \cdot \Delta \boldsymbol{r} \\ \boldsymbol{r}_{\xi^n} \cdot \Delta \boldsymbol{r} \end{bmatrix},$$

so that the stability requirement is,

$$\omega < \frac{\sum_k g^{kk}}{g^{ll}} .$$

References

[1] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, Numerical Grid Generation: Foundations and Applications (North-Holland, Amsterdam, 1985).

[2] J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, Boundary-fitted coordinate systems for numerical solution of partial differential equations—A review, J. Comput. Phys. 47 (1982) 1.

[3] J.F. Thompson, Grid generation techniques in computational fluid dynamics, AIAA J. 22 (1984) 1505.

[4] J.F. Thompson, A survey of dynamically-adaptive grids in the numerical solution of partial differential equations, Appl. Numer. Math. 1 (1985) 3.

[5] P.R. Eiseman, Grid generation for fluid mechanics computations, Ann. Rev. Fluid Mech. 17 (1985).

[6] J.F. Thompson, ed., Numerical Grid Generation (North-Holland, Amsterdam, 1982); also: Appl. Math. Comput. 10, 11 (1982).

[7] R.E. Smith, ed., Numerical grid generation techniques, NASA Conference Publication 2166, NASA Langley Research Center, 1980.

[8] K.N. Ghia and U. Ghia, Advances in Grid Generation, FED-5, ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, 1983.

[9] J. Hauser and C. Taylor, ed., Numerical Grid Generation in Computational Fluid Dynamics (Pineridge, Swansea, U.K., 1986).

[10] J.F. Thompson, A survey of composite grid generation for general three-dimensional regions, in: S.N.B. Murthy and G.C. Paynter, eds., Numerical Methods for Engine-Airframe Interpolation (AIAA, New York, 1986).

[11] P.D. Thomas, Composite three-dimensional grids generated by elliptic systems, AIAA J. 20 (1982) 1195.

[12] L.W. Ehrlich, An ad hoc SOR method, J. Comput. Phys 44 (1981) 31.

[13] W.J. Gordon, Blending function methods of bivariate and multivariate interpolation, SIAM J. Numer. Anal. 8 (1971) 158.

[14] W.J. Gordon and L.C. Theil, Transfinite mappings and their application to grid generation, in: J.F. Thompson, ed., Numerical Grid Generation (North-Holland, Amsterdam, 1982).

[15] R.L. Sorenson, Three-dimensional elliptic grid generation about fighter aircraft for zonal finite-difference computations, AIAA-86-0429, AIAA 24th Aerospace Sciences Conference, Reno, NV, 1986.

[16] P.D. Thomas, Stationary interior grids and computation of moving interfaces, in: J.J.H. Miller, ed., Advances in Computational Methods for Boundary and Interior Layers (Boole Press, Dublin, 1984).

[17] M. Vinokur, On one-dimensional stretching functions for finite-difference calculations, J. Comput. Phys. 50 (1983) 215.

[18] J.F. Thompson and C.W. Mastin, Order of difference expressions on curvilinear coordinate systems, in: K.N. Ghia and U. Ghia, eds., Advances in Grid Generation, FED-5, ASME Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, 1983.

[19] J.F. Thompson, Project EAGLE–Numerical grid generation system user's manual, Vol. 3: grid generation system, USAF Armament Laboratory Technical Rept., Eglin Air Force Base (1987) (to appear).

[20] D.M. Belk and D.L. Whitfield, 3-D Euler solutions on blocked grids using an implicit two-pass algorithm, AIAA-87-0450, AIAA 25th Aerospace Sciences Conference, Reno, NV, 1987.

[21] J.S. Mounts, A. Martinez and J.F. Thompson, An analysis of elliptic grid generation techniques using an implicit Euler solver, AIAA-86-1766, AIAA Applied Aerodynamics Conference, San Diego, CA, 1986.