INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 8, 1165–1181 (1988)

A CONTROL POINT FORM OF ALGEBRAIC GRID GENERATION

PETER R. EISEMAN

Department of Applied Physics and Nuclear Engineering, Columbia University, New York, NY 10027, U.S.A.

SUMMARY

Local control points are established within the context of algebraic grid generation. The method of generation is based upon a multidirectional assembly of multisurface transformations that incorporates the best features of tensor product and Boolean sum constructions. Upon assembly, the resultant capability is the capacity to conform precisely to prescribed boundaries while being able to manipulate the grid through a sparse net of control points.

KEY WORDS Grid Net Control points Curve Interpolation Tensor Transfinite

INTRODUCTION

The development of algebraic grid generation has proceeded in a general manner to either establish a larger amount of control for a given direction or combine various directions with a lesser amount of control in such directions. The basic constructive elements are the multisurface transformation for a high level of control in a given direction and the Boolean sum operation for the combination directions.[1] In the applications the multisurface controls are controls over the shape of the co-ordinate curves which connect the opposing boundaries of a physical region. With the local shape control, the lateral boundaries of the region may be given a reasonably good approximation, although the match there is not precise. To make it precise, the various directions must be combined with the use of Boolean sums. The straightforward combination of general multisurface transformations is possible and results in the requirement to specify a number of intermediate control surfaces for each distinct direction. While this requirement might not present a large burden in two-dimensional applications, it is somewhat excessive in three dimensions. There are simply too many data to be specified for the level of control sought. As a consequence, virtually all previous combinations of directions have been done with the straightforward use of global Lagrange or Hermite polynomial interpolation. Such interpolation procedures are special cases of the multisurface transformation. The corresponding implementation has concentrated exclusively on the boundaries and the derivatives from the boundaries.

With the intent of economically employing local controls throughout the entire field in a multidirectional context, a control point form of algebraic grid generation has been developed. The earlier burden of entire specified surfaces has been removed. It has been replaced by a sparse collection of control points from which the shape and position of co-ordinate curves can be manipulated while the grid conforms precisely to all boundaries, regardless of direction. Methods with complete boundary conformity have typically been called transfinite to reflect the matching

0271–2091/88/101165–17$08.50

Received 2 September 1987
Revised 17 December 1987

of the infinite number of points that generally describes the boundary. This terminology appears in contrast to tensor product interpolation which usually only matches the boundary corners: a finite number of points.

THE CONSTRUCTION OF CURVES

A fundamental part of the control point formulation is the construction of curves. This construction represents algebraic co-ordinate generation in a single direction wherein two opposing boundaries are connected by the newly created curves. With the restriction to only a single curve, the opposing boundaries are each represented by a point. Here the essential nature of the control appears in such an isolated state that its features are more transparently evident. Moreover, the generation process is more readily presented.

The assumed data for the generation of a curve are just a sequence of points in space. In terms of co-ordinate generation, the first and last points lie on opposing boundaries and are the fixed endpoints of the curve. The remaining points are in the interior of the sequence and are used to control the shape of the curve. As such, they are called control points.

From any one control point, the linear connection to the neighbouring point on each side gives a local segment from the entire piecewise linear curve connecting the successive points of the given sequence. The local segment about each control point then defines a change in direction. The rate of change is a curvature measure and thus presents a shape control when the curve follows the directions determined by the two linear connections attached to the given control point. This circumstance arises when the curve successively assumes the respective directions between each pair of control points.

To enforce the successive assumptions of desired direction in a smooth manner, a continuous direction field is obtained by interpolation. The independent variable for the interpolation is simply the curve parametrization. Altogether, the interpolated result defines the field of vectors that are tangent to the desired curve and that is simply stated as an interpolation of first parametric derivatives. This determines a smooth first derivative of the entire curve. The desired curve is then obtained by a parametric integration. The integration here is taken so that the curve connects the specified endpoints.

To state the result mathematically some notation is needed. For this purpose, let $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_N$ be the given sequence of points in space; let r be the curve parametrization; let $\mathbf{P}(r)$ be the position at r along the desired curve; let $r_1, r_2, \ldots, r_{N-1}$ be the successive parametric locations to interpolate the directions of $(\mathbf{C}_2 - \mathbf{C}_1), (\mathbf{C}_3 - \mathbf{C}_2), \ldots, (\mathbf{C}_N - \mathbf{C}_{N-1})$; and let $\psi_1(r), \psi_2(r), \ldots, \psi_{N-1}(r)$ be the corresponding interpolation functions which successively separate each direction by assuming a non-zero value at the asssociated location while vanishing at the remaining locations for interpolation. In two dimensions $\mathbf{C}_\alpha = (x_\alpha, y_\alpha)$ and $\mathbf{P}(r) = (x(r), y(r))$. With this notation the curve is given by

$$\mathbf{P}(r) = \mathbf{C}_1 + \sum_{\alpha=1}^{N-1} \frac{G_\alpha(r)}{G_\alpha(r_{N-1})} (\mathbf{C}_{\alpha+1} - \mathbf{C}_\alpha), \tag{1}$$

where

$$G_\alpha(r) = \int_{r_1}^{r} \psi_\alpha(\mu)\, \mathrm{d}\mu. \tag{2}$$

To witness the basic specifications indicated in our discussion, it is an easy matter to check the end conditions $\mathbf{P}(r_1) = \mathbf{C}_1$ and $\mathbf{P}(r_{N-1}) = \mathbf{C}_N$ and the interpolatory condition that $\mathbf{P}'(r_\alpha)$ is in the direction of $(\mathbf{C}_{\alpha+1} - \mathbf{C}_\alpha)$ for each α from 1 to $N-1$. In the context of co-ordinate generation for two- or three-dimensional regions, the endpoints become boundary surfaces and the interior points become control surfaces. For this reason the transformation generated by curves of the above form has been called a multisurface transformation.

To apply the method, the interpolation functions must be chosen. There is a wide variety of such choices. In broad terms these can be split between global and local functions. With local functions, however, the alteration of a control point (or surface) results in an alteration of the constructed curve (or curves) which is restricted to a local region about the point (or surface). The remaining regions are unaltered. Altogether, this represents an advantage since local sections can be manipulated in an independent manner. The development and application of such local controls is given in References 2 and 3, while a general overview is given in Reference 1.

For the discussion herein a simple choice shall be made with the underlying understanding that other choices are possible. By restricting our attention to two-dimensional applications, it is sufficient to consider local piecewise linear interpolants. These produce curves that are continuous only up to first derivatives. Moreover, a unit spacing in the interpolation points r_α shall be assumed by setting $r_\alpha = \alpha$. This causes both the interpolation functions and their integrals G_α to be translations of a function about the origin. The overall consequence is a simply stated curve definition. In an analytical form the origin-centred interpolation function is given by

$$\psi(r)=\begin{cases} 1-|r| & \text{for } -1\leq r\leq 1, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and its integral is given by

$$\Omega(r)=\begin{cases} 0 & \text{for } r<-1, \\ \frac{1}{2}(r+1)^2 & \text{for } -1\leq r<0, \\ 1-\frac{1}{2}(r-1)^2 & \text{for } 0\leq r\leq 1, \\ 1 & \text{for } 1<r. \end{cases} \tag{4}$$

With Ω increasing to a value of unity, the ratio of integrals in equation (1) reduces to just the integral in each numerator. This is yet another simplification. By employing the Ω of equation (4), the desired curve is now generated by

$$\mathbf{P}(r)=\mathbf{C}_1+\sum_{\alpha=1}^{N-1} G_\alpha(r)(\mathbf{C}_{\alpha+1}-\mathbf{C}_\alpha), \tag{5}$$

where

$$G_1(r)=2\Omega(r-1)-1, \quad G_\alpha(r)=\Omega(r-\alpha), \quad G_{N-1}(r)=2\Omega(r-N+1) \tag{6}$$

for $\alpha=2, 3, \ldots, N-2$. Since the coefficients represented by equations (4) and (6) are known functions, the curve of equation (5) depends only upon the sequence of points $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_N$. Moreover, the more general form of equation (1) can always be reduced to the form of equation (5) by scaling each interpolation function so that $G_\alpha(r_{N-1})=1$. In the subsequent development herein this scaling shall be assumed. In the applications presented the functions in equations (4) and (6) are employed. These specific functions also lead to local quadradic Bezier segments, a coincidence that does not occur with other choices. A general schematic view of the curve construction is given in Figure 1.

At this point it is important to take note of the distinctive properties of the multisurface construction that separate it from other methods such as the Bezier and B-spline techniques. To start, we shall first examine the assembly process. While the above noted specialization to local Bezier quadradic segments occurred, the process by which those segments were assembled was automatically determined so that the consequent grid spacing along the entire curve would vary

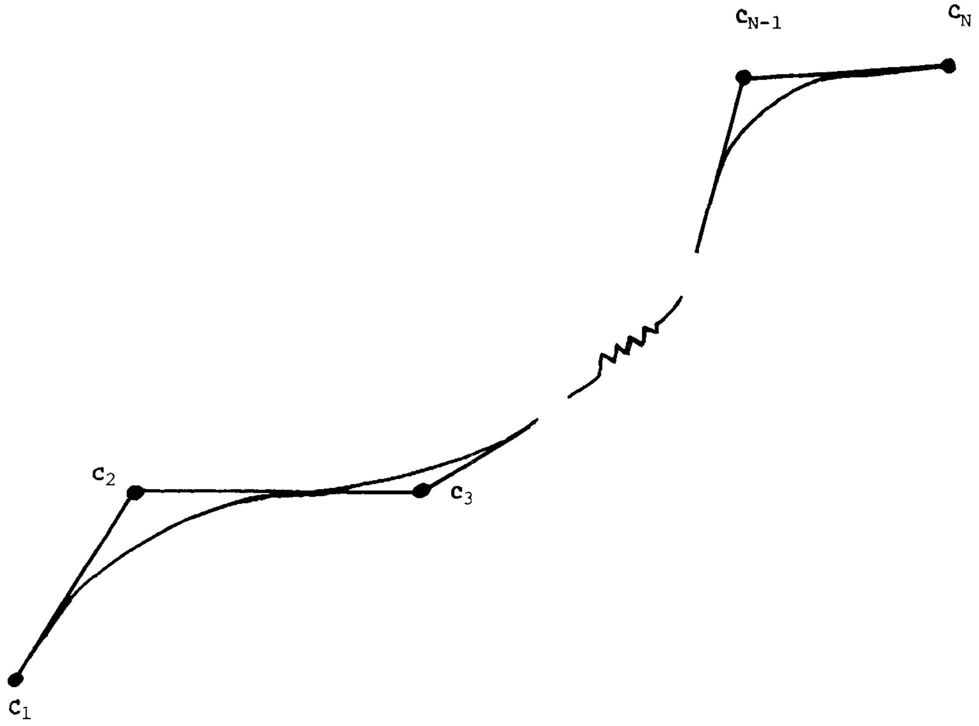

Figure 1. Curve construction from equations (4)–(6)

continuously. This result comes from the fact that tangent vectors to the desired curve were interpolated. As such, the tangent field is uniquely determined in a smooth fashion. When the associated curve parameter is then uniformly partitioned to determine a grid, this smoothness is immediately reflected in the form of smooth variations in the spacing of points. By contrast, the direct Bezier construction can have distinct tangents at the junctures between segments: the only requirement there is that the tangents be parallel. From a global view of the grid on the curve, however, there would be discontinuities in the grid spacing at each juncture where the magnitudes of the tangents are distinct; the size of the jumps there would simply be given by the amount of discrepancy in tangent magnitudes.

Aside from the assembly process, there is also a basic geometric distinction. As we have seen in our sketch of the fundamental multisurface construction, it is as if our current location were given by a vehicle that we were smoothly steering across a field to leave a curve demarked by tire tracks. As we have noted in the previous paragraph, the smoothness of our motion also yields the smoothness in spacing. In an even more fundamental way, however, we are in direct control of the curve shape for we can steer in any direction that we please and accordingly develop virtually any path that we want. Thus we clearly have a direct control over the curvature of the path and, in particular, over the convexity of the path. In a similar but distinctly weaker sense, both Bezier and B-spline techniques have a convexity control. In each of these techniques the path is locally seen to lie within the convex hull of the control points that determine the given local segment. The convex hull of a set of points is simply defined to be the smallest convex set which contains them. The techniques which satisfy this condition are said to possess a convex hull property. In the specific case with local quadradic Bezier segments, each such local segment is determined by only three control points. In terms of the convex hull property, the segment must lie within the triangle defined by the three points. In terms of actively steering the curve, the tangent vectors at each end must assume the respective directions determined by the two successive vectors obtained by taking the differences between successive control points. In this special case the endpoints of the curve segment also lie on the two line segments which join the successive control points. Altogether, the curve segment then leaves the first line segment with the same tangent direction and then linearly changes direction to reach the second line segment in the same manner. Clearly this linear direction change causes the curve to stay within the triangle and thus to satisfy the convex hull property. While the same convexity property arose from the two distinct methods in this special case, the differences rapidly appear as we leave this case. In the most direct sense, as we increase the local control point dependency beyond three, the size of the convex hull can grow rapidly. This growth is common and is often quite dramatic. The concurrent requirement that the curve lie within it, however, is then greatly weakened. By contrast, the direct control of tangent directions does not experience such a weakening. It also does not necessarily satisfy a convex hull property, although departures therefrom are not large.

In addition to the above noted distinctions, there is also the issue of generality. This arises from the somewhat rigid confines of Bezier and B-spline methods in comparison with the flexibility that is available with the multisurface method. The rigidity comes from the restrictions to the use of only Bernstein polynomials for Bezier methods and (usually cubic) B-splines for B-spline methods. These specific choices are also tied to the number of local control points and appear as a further restriction; namely, that the degree of the polynomial or assembled polynomials is set. In contrast, the multisurface method is not restricted in this manner. There the process of vector field interpolation is performed without tying down the choice of interpolation functions. The only constraining condition is that the choice be taken in the canonical fashion whereby the function for a given interpolation point is non-zero there but vanishes at all other interpolation points. The inherent flexibility is clearly evident: the choice can be taken in many forms. This can appear as polynomial or piecewise polynomial constructs as in the Bezier and B-spline methods. While there are a multitude of possibilities with polynomials and pieces of them, there are yet further choices which, for example, can be put together with trigonometric and hyperbolic functions, just to mention a few other possible constructive elements.

A discussion of Bezier and B-spline techniques can be readily witnessed in the computer graphics literature and corresponding texts. The most notable source there is the text by Foley and van Dam.[4] Their emphasis is reflective of their field and, accordingly, is placed within the context of 4 × 4 matrix operations. In that format the interest is in fast recursive operations to evaluate the successive points of a cubic curve. The mathematics there is brief and is clearly restricted to cubic polynomials. Aside from some illustrations, the convex hull property is not at all developed there. To examine the spirit of the analysis behind the convex hull property, the reader is referred to the work of Gordon and Riesenfeld[5] where the property is developed for B-spline methods.

THE CONTROL POINT FORMULATION IN TWO DIMENSIONS

The control point array is a sparse grid-type arrangement of locations in physical space with an index for each direction. In two dimensions it will be denoted by $(\mathbf{C}_{ij})$. Along each sequence of control points, a multisurface construction can be performed, and this in effect creates sequences of curves: there is one for each index value of i or j. With the multisurface interpolants leading to functions $G_\alpha(r)$ and $H_\beta(t)$ respectively, the constructed curves are given by

$$\mathbf{E}_j(r)=\mathbf{C}_{1j}+\sum_{\alpha=1}^{N-1} G_\alpha(r)[\mathbf{C}_{\alpha+1,j}-\mathbf{C}_{\alpha j}] \tag{7}$$

and

$$\mathbf{F}_i(t)=\mathbf{C}_{i1}+\sum_{\beta=1}^{M-1} H_\beta(t)\,[\mathbf{C}_{i,\beta+1}-\mathbf{C}_{i\beta}] \tag{8}$$

respectively for $j=1, 2, \ldots, M$ and $i=1, 2, \ldots, N$. The tensor product form also depends only upon the $\mathbf{C}_{ij}$ and is given by

$$\mathbf{T}(r, t)=\mathbf{E}_1(r)+\sum_{\beta=1}^{M-1} H_\beta(t)[\mathbf{E}_{\beta+1}(r)-\mathbf{E}_\beta(r)] \tag{9}$$

or alternatively by

$$\mathbf{T}(r, t)=\mathbf{F}_1(t)+\sum_{\alpha=1}^{N-1} G_\alpha(r)\,[\mathbf{F}_{\alpha+1}(t)-\mathbf{F}_\alpha(t)]. \tag{10}$$

The equivalence of the two expressions for the tensor product can be easily checked by inserting equations (7) and (8) and reducing each to arrive at the same expansion in terms of the $\mathbf{C}_{\alpha\beta}$. The equivalence means that the tensor product is independent of the order in which the multisurface constructions are performed.

In either form the tensor product is seen to match the $\mathbf{E}_j$ or $\mathbf{F}_i$ at the extremities of i and j. From the t-direction construction with the curves $\mathbf{E}_j(r)$ in equation (9), the end conditions immediately produce $\mathbf{T}(r, 1)=\mathbf{E}_1(r)$ and $\mathbf{T}(r, M-1)=\mathbf{E}_M(r)$. At the lateral boundary $r=1$, each curve $\mathbf{E}_j$ reduces to the corresponding point $\mathbf{C}_{1j}$ as determined by equation (7). As a consequence, equation (9) then reduces to the expression of equation (8) with $i=1$. This shows that $\mathbf{T}(1, t)=\mathbf{F}_1(t)$. In the same manner $\mathbf{T}(N-1, t)=\mathbf{F}_N(t)$. Alternatively, the same results can be derived from the r-direction construction of equation (10) or directly from the end conditions of both the constructions of equations (9) and (10).

When boundaries are to be specified, the corresponding data appear at the extremities of the values for r and t. Since the co-ordinate transformations are generally expressed in the form of a vector $\mathbf{P}(r, t)$ for the desired positions of all points in physical space, it is also convenient to express the boundary specifications in terms of the position vector. Thus the boundaries are denoted by $\mathbf{P}(1, t)$, $\mathbf{P}(N-1, t)$, $\mathbf{P}(r, 1)$ and $\mathbf{P}(r, M-1)$. To include the boundaries, the multisurface transformation is performed again as above, but now with the actual boundaries inserted. The above tensor product forms are then replaced by

$$\begin{aligned}\mathbf{A}(r, t)=\mathbf{P}(1, t)&+G_1(r)\,[\mathbf{F}_2(t)-\mathbf{P}(1, t)]\\&+\sum_{\alpha=2}^{N-2} G_\alpha(r)\,[\mathbf{F}_{\alpha+1}(t)-\mathbf{F}_\alpha(t)]+G_{N-1}(r)\,[\mathbf{P}(r_{N-1}, t)-\mathbf{F}_{N-1}(t)]\end{aligned} \tag{11}$$

and

$$\begin{aligned}\mathbf{B}(r, t)=\mathbf{P}(r, 1)&+H_1(t)\,[\mathbf{E}_2(r)-\mathbf{P}(r, 1)]\\&+\sum_{\beta=2}^{M-2} H_\beta(t)\,[\mathbf{E}_{\beta+1}(r)-\mathbf{E}_\beta(r)]+H_{M-1}(t)\,[\mathbf{P}(r, M-1)-\mathbf{E}_{M-1}(r)]\end{aligned} \tag{12}$$

for the r- and t-direction respectively, so that the actual boundaries appear as end conditions. The alteration here was just a replacement of the control point boundaries by the actual boundaries. As such, the tensor product form is retained in the interior: this appears as the summation in equations (11) and (12). Accordingly, these transformations should then also appear as direct boundary variations to the tensor product transformation. To algebraically establish the anticipated variations, the interior summations must first be expressed in terms of the full tensor product transformation. With equations (9) and (10) they become

$$\sum_{\alpha=2}^{N-2} G_\alpha(r)\,[\mathbf{F}_{\alpha+1}(t)-\mathbf{F}_\alpha(t)]=\mathbf{T}(r,t)-G_1(r)\,[\mathbf{F}_2(t)-\mathbf{F}_1(t)]-G_{N-1}(r)\,[\mathbf{F}_N(t)-\mathbf{F}_{N-1}(t)] \quad (13)$$

and

$$\sum_{\beta=2}^{M-2} H_\beta(t)\,[\mathbf{E}_{\beta+1}(r)-\mathbf{E}_\beta(r)]=\mathbf{T}(r,t)-H_1(t)\,[\mathbf{E}_2(r)-\mathbf{E}_1(r)]-H_{M-1}(t)\,[\mathbf{E}_M(r)-\mathbf{E}_{M-1}(r)] \quad (14)$$

respectively. Upon substitution, the tensor product transformation is explicitly rendered. With some modest cancellation of terms, the desired variations are obtained and are given by

$$\mathbf{A}(r,t)=[1-G_1(r)]\,[\mathbf{P}(1,t)-\mathbf{F}_1(t)]+\mathbf{T}(r,t)+G_{N-1}(r)\,[\mathbf{P}(N-1,t)-\mathbf{F}_N(t)] \quad (15)$$

and

$$\mathbf{B}(r,t)=[1-H_1(t)]\,[\mathbf{P}(r,1)-\mathbf{E}_1(r)]+\mathbf{T}(r,t)+H_{M-1}(t)\,[\mathbf{P}(r;M-1)-\mathbf{E}_M(r)] \quad (16)$$

respectively. In this form the extent of each variation is directly displayed. This appears with an adjustment term for each boundary. For example, at the boundary of $r=1$, the first term in equation (15) yields $\mathbf{P}(1,t)-\mathbf{F}_1(t)$, the tensor product reduces to $\mathbf{F}_1(t)$ and the last term is 0. The result is then a match with $\mathbf{P}(1,t)$. On the same boundary each adjustment term in the transverse construction of equation (16) must vanish because $\mathbf{P}(1,1)$ and $\mathbf{E}_1(1)$ match the corner $\mathbf{C}_{11}$ and similarly $\mathbf{P}(1, M-1)$ and $\mathbf{E}_M(1)$ match the corner $\mathbf{C}_{lM}$. This leaves only the tensor product contribution which is $\mathbf{F}_1(t)$.

Proceeding in the same spirit, all of the boundary evaluations are established for **A** and **B**. These are displayed along with those of **T** in Figure 2. In the display each box about **A**, **B** and **T** represents the domain of curvilinear variables. This is just the rectangle $1\leq r\leq N-1$ by $1\leq t\leq M-1$. The corresponding boundary value is indicated at each side of the rectangle for **A**, **B** and **T**. From the values in the figure it is readily witnessed that the simple sum of $\mathbf{A}+\mathbf{B}$ includes a specified **P** and a control point form at each boundary, while **T** always produces a control point form. Thus to remove the control point form from the simple sum, one only needs to deduct the tensor product form. The result is the Boolean sum

$$\mathbf{Q}=\mathbf{A}+\mathbf{B}-\mathbf{T}, \quad (17)$$

which then matches the specified values of **P** at all boundaries.

Upon assembly of the constituent parts represented by equations (15) and (16), the Boolean sum stated in equation (17) reduces to a particularly simple form. Altogether, it is given by

$$\begin{aligned}\mathbf{Q}(r,t)=\mathbf{T}(r,t)&\\ &+[1-G_1(r)]\,[\mathbf{P}(1,t)-\mathbf{F}_1(t)]\\ &+G_{N-1}(r)\,[\mathbf{P}(N-1,t)-\mathbf{F}_N(t)]\\ &+[1-H_1(t)]\,[\mathbf{P}(r,1)-\mathbf{E}_1(r)]\\ &+H_{M-1}(t)\,[\mathbf{P}(r,M-1)-\mathbf{E}_M(r)],\end{aligned} \quad (18)$$

which explicitly bears the interpretation of an adjusted tensor product construction. Each of the four terms following the tensor product $\mathbf{T}(r, t)$ represents a transfinite conformity to a boundary. In the order of appearance, the boundaries are for $r=1$ and $r=N-1$ and then for $t=1$ and $t=M-1$.

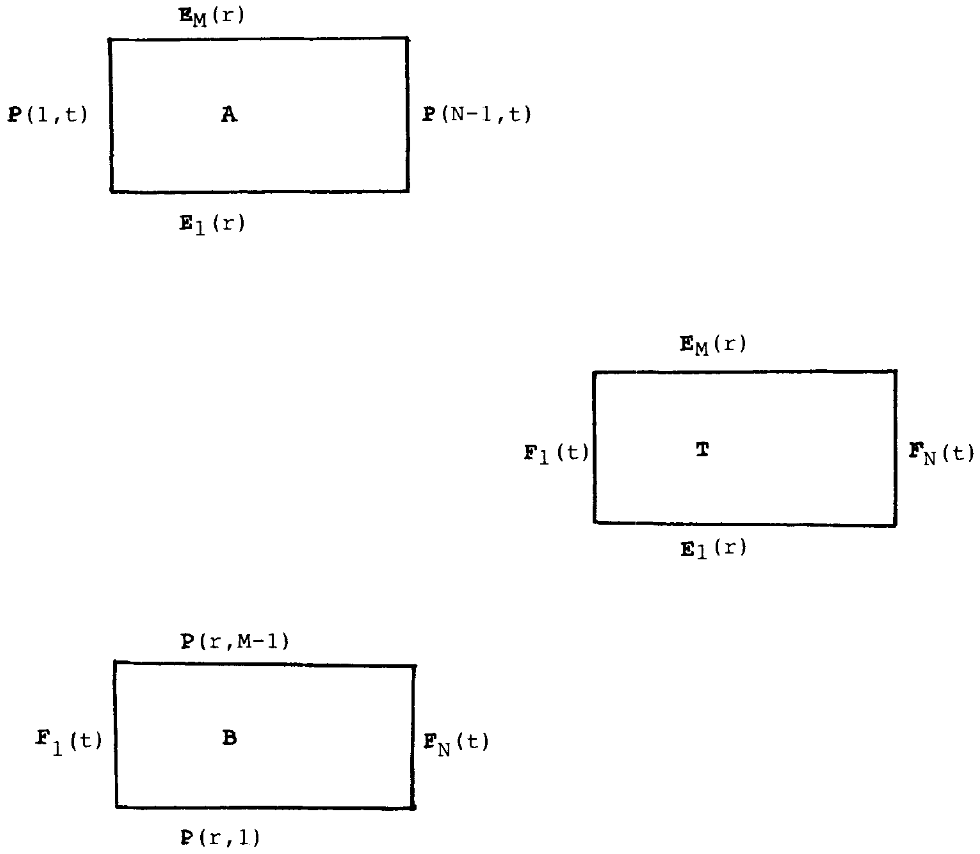

Figure 2. Boundary values for **A**, **B** and **T** on the domain $1 \leq r \leq N-1$, $1 \leq t \leq M-1$

When one employs the local interpolants described earlier and in References 2 and 3, the functions $G_k(r)$ and $H_k(t)$ have only local variations. In the application of equation (18) the effect is a local treatment of the boundary conformity, thus leaving a local tensor product treatment in the interior. This leads to the utilization of only the control points $\mathbf{C}_{ij}$. Variations of equation (18) now arise naturally: boundary conformity can be applied selectively. By dropping any conforming term, the associated boundary can be manipulated into any other desired shape.

APPLICATIONS

To explicitly examine the control point form of algebraic grid generation, some example applications are considered. The first example is that of a control net for a uniform distribution of points over a duct type of region. This is displayed in Figure 3. All boundaries are specified and held fixed, while the control points are determined from an attachment to a known co-ordinate transformation. The known transformation was taken as a transfinite interpolation with linearity in both the r- and t-direction. While this is a special case of the general theory here, it is also the simplest case that just provides us with a relatively uniform grid. The simplicity derives from the use of only boundary data. In the example, the control net determined by the simple linear–linear

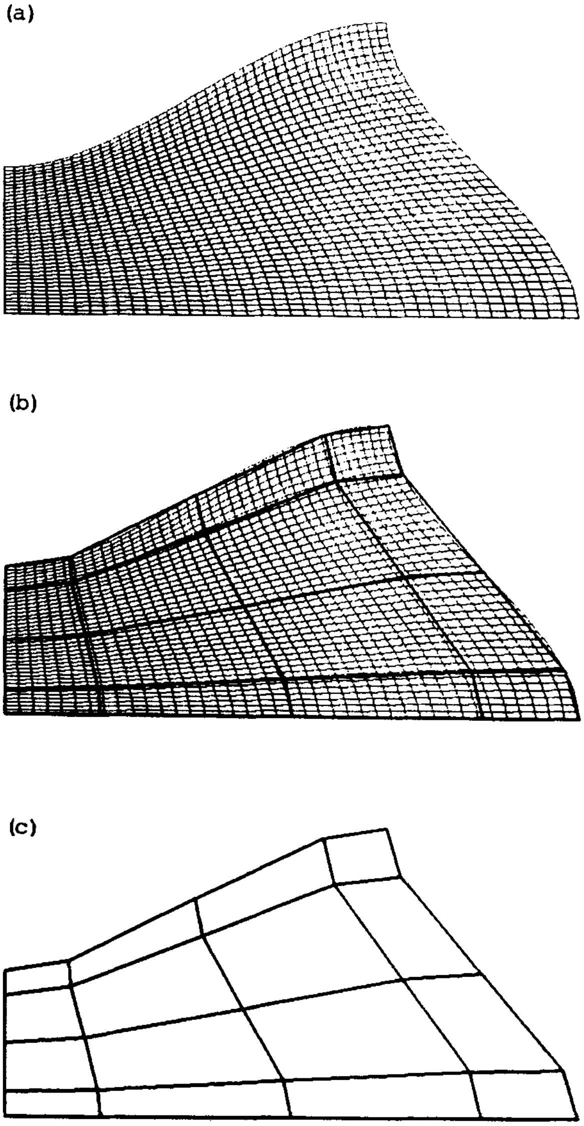

Figure 3. A duct grid (a) and control net (c) that are superimposed in (b). The control net is for a uniform distribution

transfinite transformation is seen as the heavy lines in parts (b) and (c) in Figure 3, while the associated grid is seen as the narrower lines in parts (a) and (b). Because of the boundary truncations for $G_1(r)$, $G_{N-1}(r)$, $H_1(t)$ and $H_{M-1}(t)$, the determined control net is uniform only in the interior and assumes a half of the uniform spacing to reach the boundaries. From the 5×5 control net this half spacing at the ends is readily observed along each control point curve. The consequent grid in parts (a) and (b) of Figure 3 is simply a fairly good approximation to the given linear–linear transfinite transformation.

To depart significantly from the linear–linear transfinite case, one control point is moved towards the centre in Figure 4; the other control points and the boundaries remain the same. In a

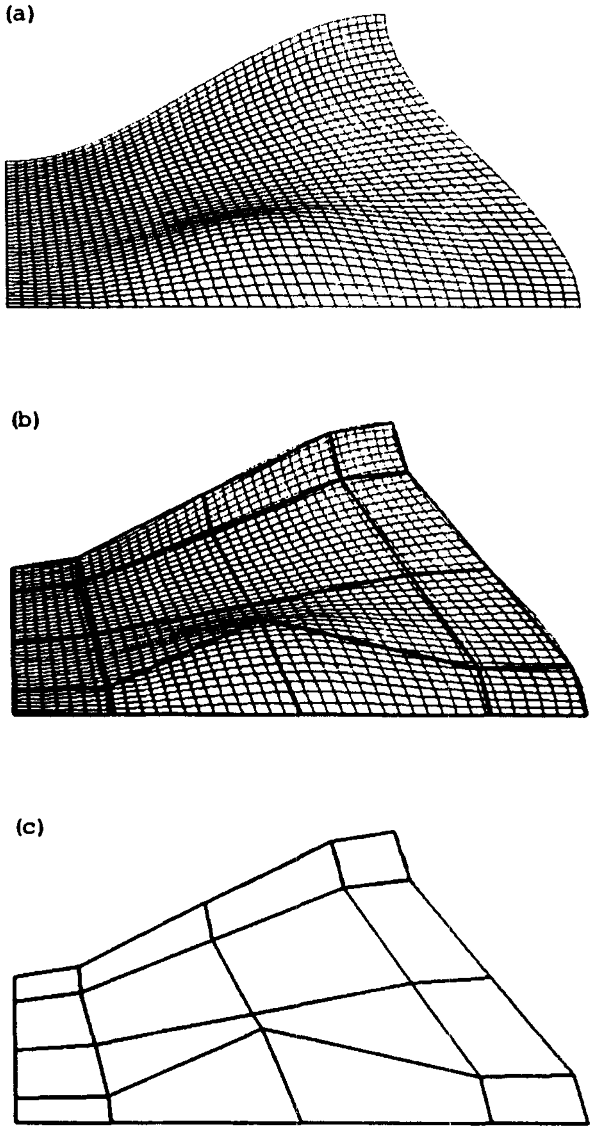

Figure 4. A cluster in the duct grid from the motion of one control point

parallel fashion the parts of Figure 4 correspond to the earlier parts in Figure 3. The effect of the control point motion is seen directly in the grid as the formation of an internal cluster for the horizontal co-ordinate curves. Similarly, vertical clusters, pointwise clusters, local Cartesian forms and boundary orthogonality may be considered, not to mention other possibilities.

To display some of the practical capability in a more complex case, a centre body is inserted. It is given in Figure 5. Here the bottom boundary is allowed to move. This is accomplished by simply dropping the associated blending term in equation (18). Similarly, the adjoining lateral boundaries are also permitted to move but only for the purpose of redistribution, not for a new geometry. As can be seen, the 12 × 5 control net has points which merge to form corners for the

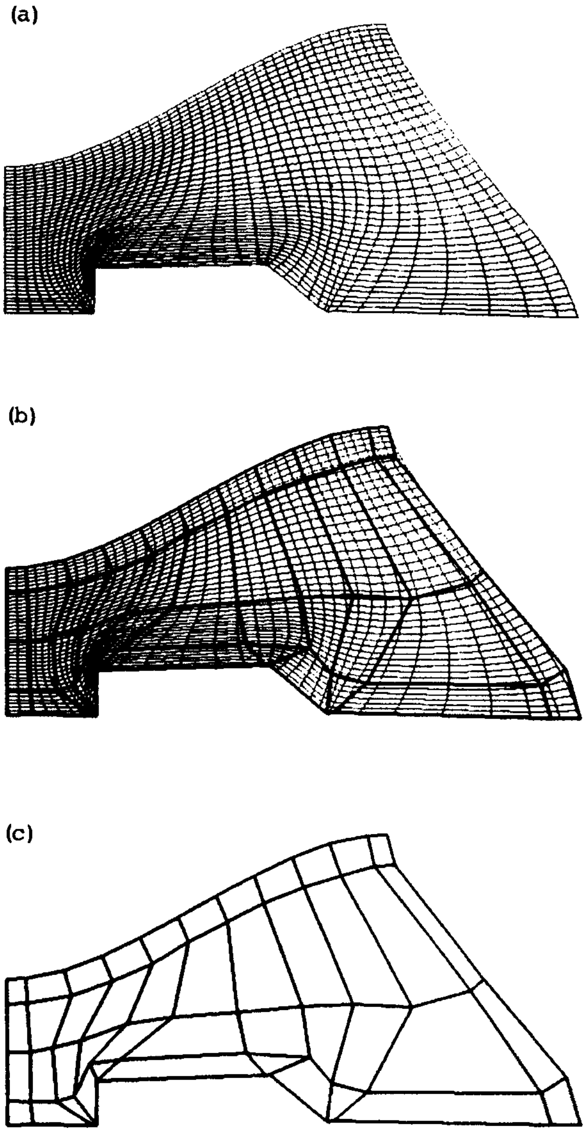

Figure 5. A slope-discontinuous internal body inside the duct and a larger control net to accommodate the body

centre body. In addition, the control points adjacent to the top fixed boundary are chosen to provide boundary orthogonality, as are the control points adjacent to the lateral boundaries. Altogether, a smooth grid is generated with boundary orthogonality on upper and lateral boundaries and with a modest pointwise concentration about an inner body that possess four slope discontinuities.

To examine the interactive manipulation of a boundary, a succession of control point movements is followed up to the stage where a protrusion with a slope discontinuity is created and the grid distribution is fairly uniform. Here the action starts from the previous initial grid displayed in Figure 3(a), but now with a control net that has eight points across rather than five as before. The net and grid appear at the top of Figure 6. The first action is to drop the term which transfinitely blends the bottom boundary in the control point transformation of equation (18). Although the grid has not yet changed, this action permits the free-form modelling of the bottom boundary. The modelling starts with the downward motion of the fourth point across: $\mathbf{C}_{14}$. The result is given at the bottom of Figure 6 where a smooth protrusion of the grid has been obtained. The next movement creates a slope discontinuity at the bottom of the protrusion as well as broadening it. This is displayed at the top of Figure 7. There the fifth control point along the bottom, $\mathbf{C}_{15}$, has been moved to coincide with the previous fourth point, $\mathbf{C}_{14}$, as displayed at the bottom of Figure 6. Because of the movement, the original bottom boundary segment on the right-hand side is now represented by points $\mathbf{C}_{16}$ through $\mathbf{C}_{18}$ rather than $\mathbf{C}_{15}$ through $\mathbf{C}_{18}$. The

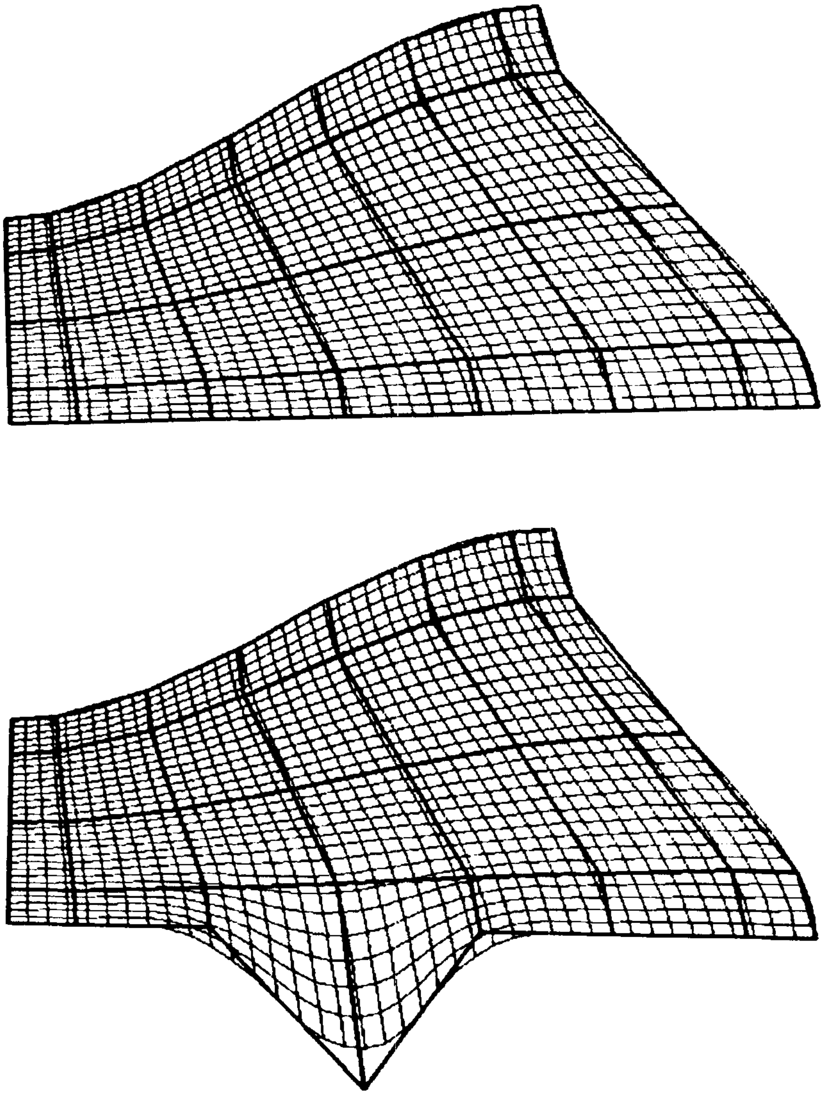

Figure 6. Free-form modelling actions on the bottom boundary

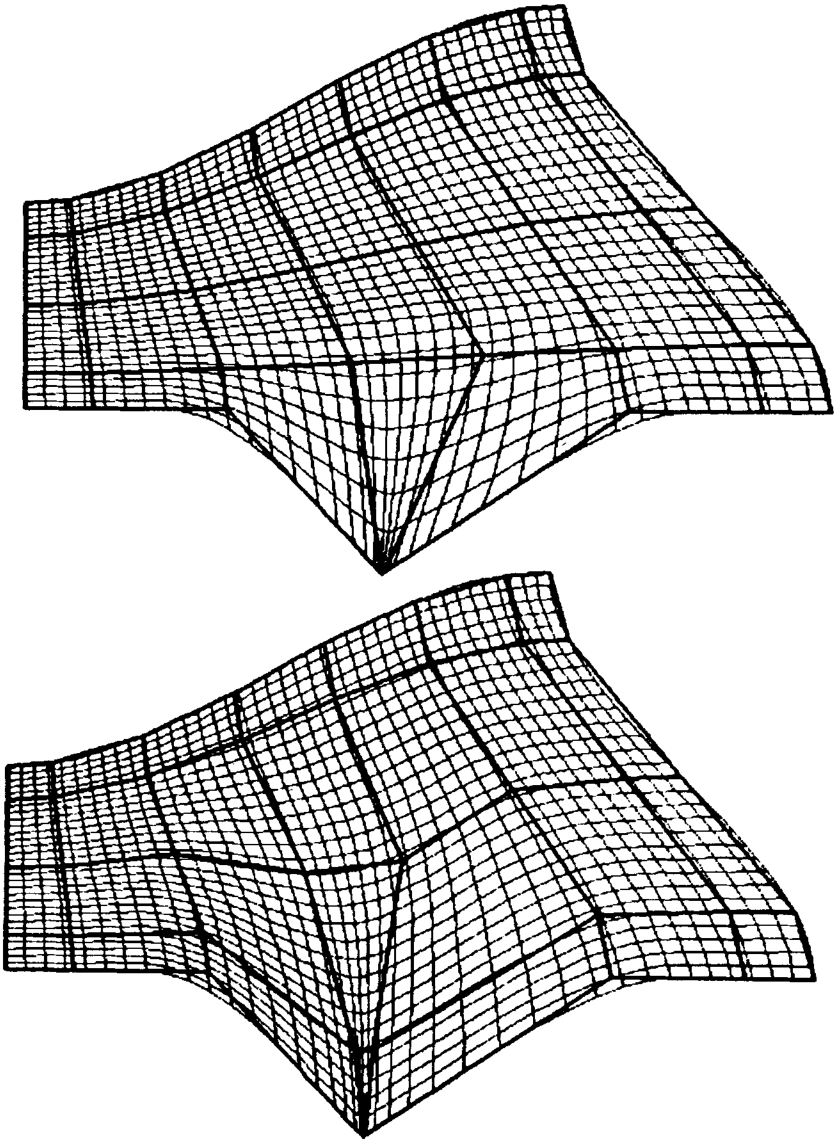

Figure 7. Free-form modelling actions on the bottom boundary

consequence is the broadened protrusion and the shorter original segment. Upon inspection, however, the grid is not as dense in the protrusion as it is elsewhere. A more uniform grid is obtained by moving control points C_{42}, C_{52}, C_{43} and C_{53} into the positions indicated at the bottom of Figure 7. Accordingly the grid there is certainly more uniform than the grid above it. With continued interactive manipulations, the more intricate grid given in Figure 8 is created, in which two slope discontinuities have been successively inserted at the bottom so that a flat connection appears between them. The consequent grid appears in the form of a bidirectional nozzle: flow can enter both the left and bottom boundaries and exit through the right boundary.

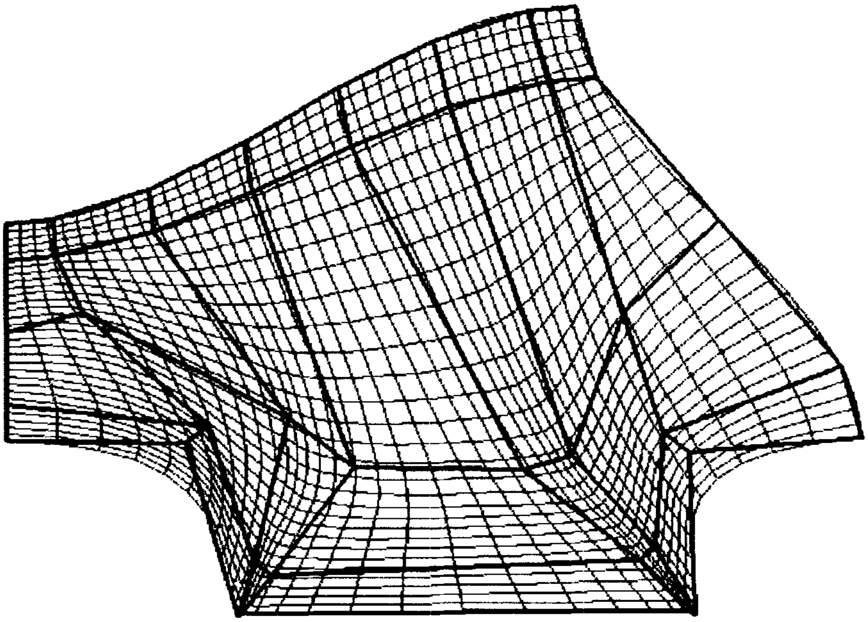

Figure 8. A bidirectional nozzle grid

To model a more general configuration, a typical template from a waterways problem is considered. This could be either an inlet or a section of a river. The generic requirements are that grids conform to various boundary irregularities and permit depth clustering. In these examples twelve control points are employed horizontally to treat the irregularities and an extra control point is added vertically to bring the total up to six so that lengthwise depth clustering can be inserted. To start, the left boundary, $r = 1$, is designated as transfinite, while the others are taken to be open for free-form modelling. This occurs by dropping three of the four transfinite blending terms in the transformation of equation (18). After an interactive sequence of movements, the configuration in Figure 9 is obtained. On the top the grid and control net are simultaneously displayed, while only the grid appears on the bottom so that it can be more clearly seen. The three free-form boundaries are wavy, as can be typically expected in this circumstance. Upon inspection, it is noticed that two slope discontinuities have been inserted. These occur at about the centre of the top boundary and about a quarter of the way across the bottom boundary. They arose from the control point merges $\mathbf{C}_{67} = \mathbf{C}_{68}$ and $\mathbf{C}_{14} = \mathbf{C}_{15}$. In addition, near orthogonality has also been imposed over most of the boundary. This can be witnessed by the approximate boundary orthogonality of the control net together with the corresponding near orthogonality at the associated locations. Upon assumptions of deeper water occurring down the centre of the region, a horizontal cluster has been inserted. Here the extra control point is employed. This is needed because the top and bottom boundaries each require a layer of points for their definition and yet another layer for the grid structure (such as orthogonality and spacing) in their vicinity. Thus with the extra layer of control points there are two layers available for depth clustering. The clustering action comes from bringing these two layers close together. With twelve control points in each layer, the clustering can be made to follow a wavy path if desired. A slightly more modest choice appears in the figure.

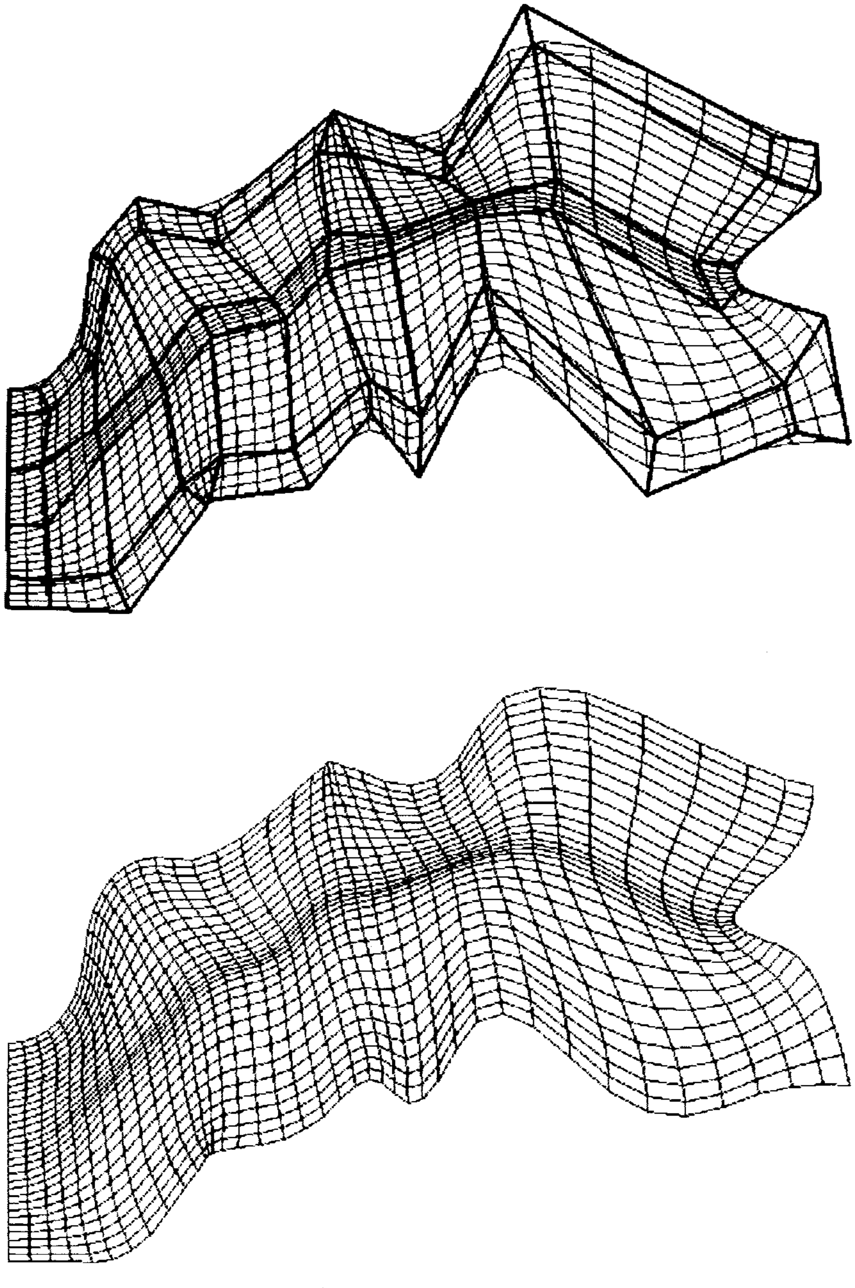

Figure 9. A waterways grid

As a further illustration of the local interactive evolution of this general configuration, further modelling is performed with the results appearing in Figure 10. The localness is clearly seen, since the grid is only changed above the horizontal cluster which is clearly left intact. At the top of Figure 10 most of the last part of the top boundary has been changed. The most noticeable features there are the creation of a more severe protrusion, the presence of two slope dis-

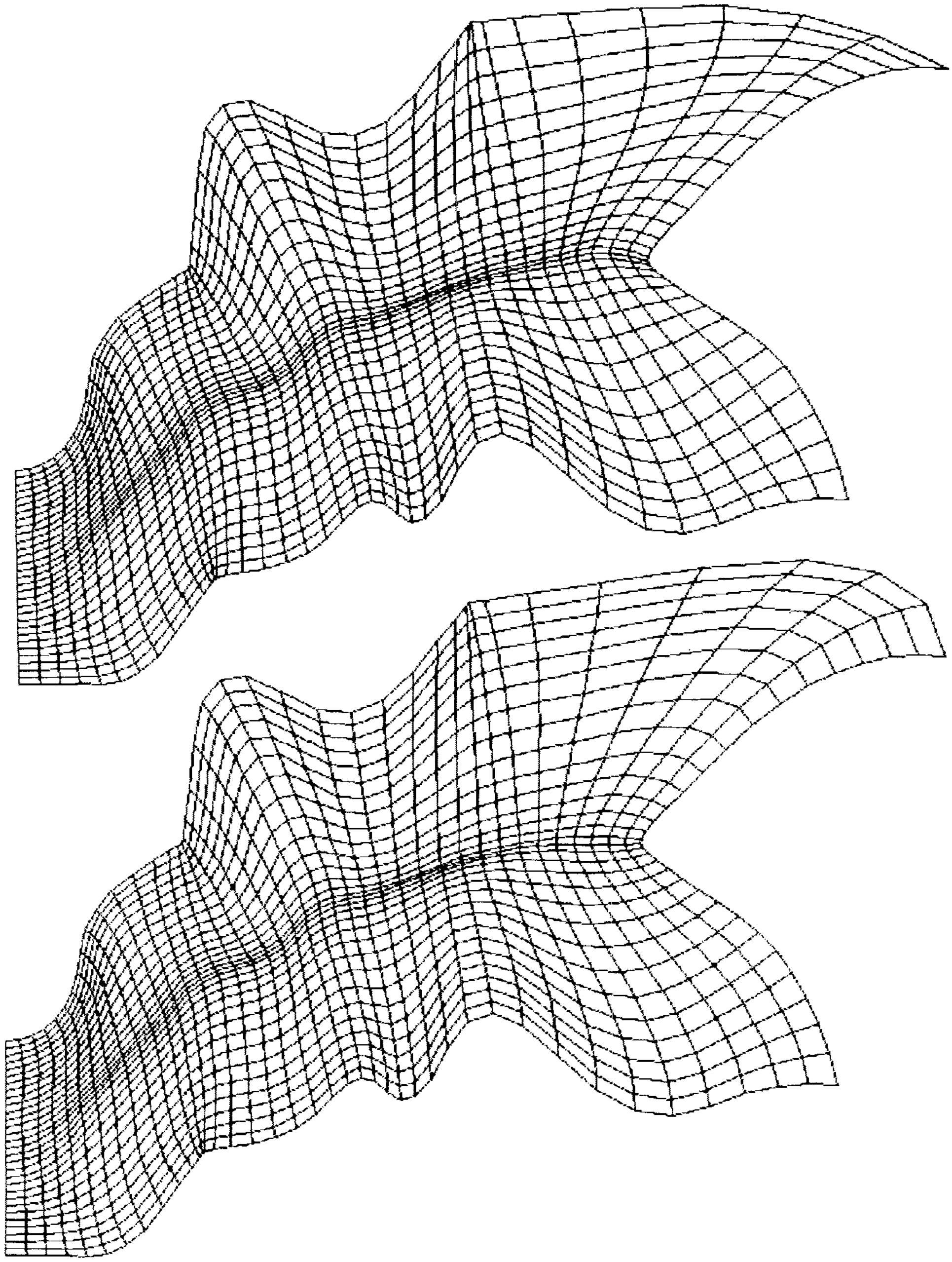

Figure 10. Further free-form modelling on the waterways grid of Figure 9

continuities and the lifting of the far right-hand corner. At the bottom of Figure 9 only one more control point, $C_{6,11}$, is moved to restructure the lifted corner. The localness of that final motion is clearly witnessed in the grid: only the grid points within the small domain of dependence of that control point are altered.

CONCLUSION

A basic control point form of algebraic grid generation has been established. While the actual development only considered specified boundaries, the developmental pattern can be continued. The most direct continuation is the inclusion of exact spacing and angle control from the boundaries. In most circumstances, however, the control point format here may be quite sufficient, since it usually provides a reasonably good approximation to such desired specifications.

Moreover, the three-dimensional form follows the same format. With the capability for a distributed field control from rather modest specification requirements, the greatest advantage of the method may appear in the context of three-dimensional applications. However, since the basic character is most easily seen in two dimensions, we restricted our discussion to two dimensions.

ACKNOWLEDGEMENT

This work was supported by the U.S. Air Force under Grant AFOSR-86-0307 with technical monitors Major John P. Thomas, Jr. and Dr. James D. Wilson and by NASA under Grant NAG-1-427 with technical monitor Dr. Robert E. Smith.

REFERENCES

1. P. R. Eiseman, 'Grid generation for fluid mechanics computations', *Ann. Rev. Fluid Mech.*, **17**, 487–522 (1985).
2. P. R. Eiseman, 'Coordinate generation with precise control over mesh properties', *J. Comput. Phys.*, **47**, 331–351 (1982).
3. P. R. Eiseman, 'High level continuity for coordinate generation with precise controls', *J. Comput. Phys.*, **47**, 352–374 (1982).
4. J. D. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, MA, 1984.
5. W. J. Gordon and R. F. Riesenfeld, 'B-spline curves and surfaces', in R. E. Barnhill and R. F. Riesenfeld (eds), *Computer Aided Geometric Design*, Academic Press, New York, 1974.