Alternating Direction Adaptive Grid Generation

Peter R. Eiseman*
Columbia University, New York, New York

An alternating direction method has been developed to adaptively resolve numerical solutions to physical problems by moving the points of a coordinate grid. With the solution or with the salient quantities derived from it, a surface grid is defined over the grid on the physical region. Grid movement is computed on this abstractly defined solution surface with the use of pointwise weights and is then projected back to determine new locations in the physical region. The weights cause clustering relative to the uniform conditions of surface arc length and are formed with magnitudes of quantities that are used for point attraction. When the salient quantities have been used to form the surface, the gradients are implicitly revolved and surface curvature becomes the primary remaining quantity. In each coordinate direction, the magnitude of normal curvature is used. With arbitrary weights, a movement cycle generally consists of a sweep through the coordinate curves in a given direction on a curve-by-curve basis and then in alternating directions to do the same thing until all directions have been covered.

Introduction

FLUID dynamic phenomena often contain important and sensitive physical quantities that vary widely over short distances. Moreover, such variations can occur simultaneously within a complicated spatial configuration that continually changes. In the numerical simulation, a discrete approximation is usually made to a system of partial differential equations that describe the particular phenomena. When the discretization is done with a fixed mesh, the accurate simulation can be jeopardized if the large solution variations occur on a scale that is too small for the mesh. To assure that the solution is well modeled, various strategies have been proposed for adapting the mesh in some way to the solution.[1-12]

In this study, a solution-adaptive grid generation scheme has been developed in a manner that has a number of favorable characteristics: grid control is directly inserted, the transformation is analytically nonsingular, the algorithm is simple, the algorithm is fast, the algorithm is independent of any particular numerical method, the number of user-specified quantities is minimized, the technique works well in any number of spatial dimensions, and the results are also applicable to grid generation on arbitrary surfaces.

These characteristics are obtained with an alternating direction strategy that extends the work of Dwyer et al.,[5] Gnoffo,[10] Ablow and Schecter,[1] Ablow,[2] and White.[3,4] The scheme is executed on one coordinate curve at a time as first done by Dwyer et al.[5] for one direction and later by Gnoffo[10] for two directions. However, the coordinate curves are in the physical domain, rather than on the actual solution surface as was the case in the work by Ablow and Schecter[1] and White[3,4] for one-dimensional problems or for the more general surfaces considered here to monitor the solution behavior. With curves in the physical domain, the number of quantities required for clustering are greater since the properties of a solution surface must be explicitly inserted. At a basic level, gradient information is desired. Dwyer et al.[5] include it in the form of a direct integral average, while Gnoffo[10] considers springs between the points with chosen constants. Both of these methods for clustering reduce to exactly the same form on any coordinate curve and each can be extended to include further properties. Without a consideration of a full solution surface, a rational selection of properties becomes more difficult since transverse information to any given coordinate curve is also needed. In Gnoffo's setting, there are transverse springs; but with no clear method of application, these were neglected in favor of only coordinatewise springs.

With a solution surface, the previously neglected data are included and, more generally, the surface geometry leads directly to a rational selection of the properties for clustering. At the most elementary level, a uniform surface distribution is the equivalent to a resolution of gradients when it is projected back down to the physical domain. In one dimension, uniformity is uniquely defined (up to normalization) by the surface arc length. An illustration is given in Fig. 1, where x is the physical space coordinate and ψ represents a scalar solution-generated quantity that consolidates the data required for adaptation. The uniform arc length distribution is depicted by the points on the curve. The lines from those points indicate the projection back down to the phsyical space, whereupon gradient clustering is clearly evident. Analytically, constant arc length increments imply that the product of $\sqrt{1+\psi_x^2}$ and dx is constant. When ψ_x^2 is much larger than 1, the radical is well approximated by $|\psi_x|$ which yields a distribution with the physical space increment dx inversely proportional to the gradient. When ψ_x^2 is much less than 1, the radical is nearly unity and a virtually uniform x distribution is obtained. Altogether, a uniform arc length distribution on the surface yields a physical space distribution that smoothly varies between gradient clustering and uniformity.

White[3,4] used arc length for gradient resolution and also as a curvilinear independent variable. The latter use is interesting in the context of nonlinear problems that can exhibit multivalued behavior. When the problem is posed directly in terms of the solution surface, the evolution of the surface is computed without reference to a possibly singular grid in the physical domain. Singularity would appear at the points of vertical tangency where the surface folds over itself. Such folds can be visualized by horizontally pushing the peak in Fig. 1 until some of the vertical lines of projections have multiple intersections with the surface. While the surface grid can be well defined and readily usable, the projected physical

Presented as Paper 83-1937 at the AIAA 6th Computational Fluid Dynamics Conference, Davers, Mass., July 13-15, 1983; received July 30, 1983; revision received May 4, 1984.

*Department of Applied Physics and Nuclear Engineering. Member AIAA.

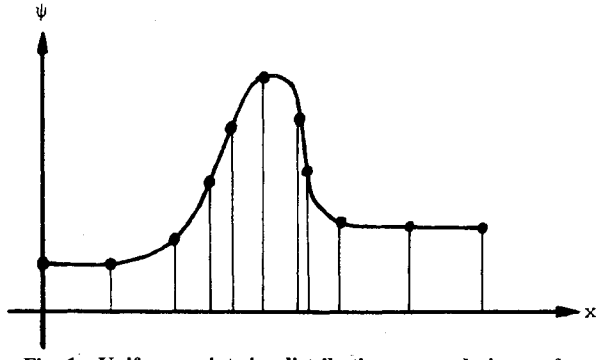

Fig. 1 Uniform pointwise distribution on a solution surface.

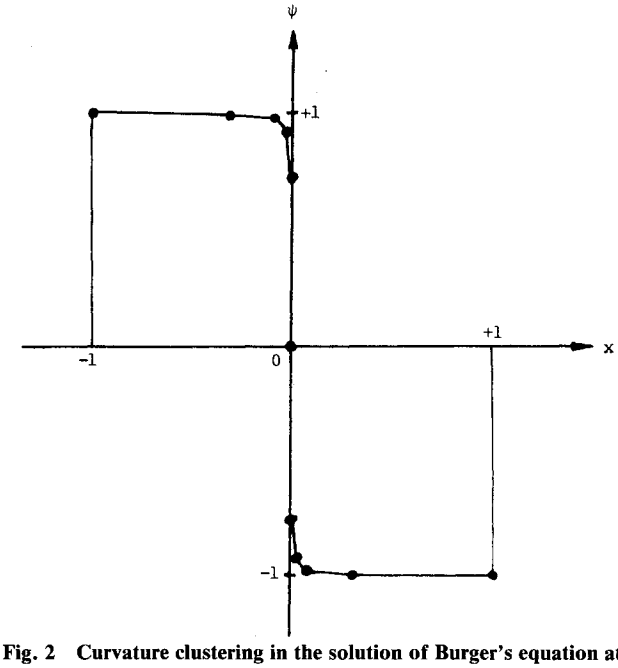

Fig. 2 Curvature clustering in the solution of Burger's equation at a Reynolds number of 200 with 11 mesh points.

space grid would then have undesirable overlaps under the folds. Consequently, nonsingularity in the simulation of multivalued phenomena is important only on the solution-generated surface.

While the one-dimensional uniformity of arc length is adequate for many problems, there are often further requirements for resolution. For example, if a solution has one very large peak relative to both the remaining solution and the physical domain, then virtually all of the points would go to the peak, leaving too few points to detect a smooth transition. In this case, as in many others, the transition regions are important. In terms of the surface, there is a change of direction that is analytically expressed as a high-curvature region. Curvature clustering in one dimension has been considered by Ablow and Schecter[1] and Eiseman.[13] An example of curvature clustering is given in Fig. 2 where the steady-state solution of the Burgers equation at a Reynolds number of 200 was obtained from a linear start and a uniform mesh distribution in physical space $[-1,1]$ with nine movable points. The final configuration of points is plotted on the solution-generated surface that was taken as the solution itself. The curvature regions are clearly resolved by observing the clusters where the solution changes direction by 90 deg. As a consequence, the solution analytically known to be a hyperbolic tangent is well represented. In this example, a Crank-Nicolson procedure was used in a straightforward manner with the weight of Eq. (18) below, which in one dimension reduces to the curvature of the curve.

In higher dimensions, by contrast, there is no unique definition for surface uniformity or curvature clustering since the distinct directions become a fundamental part of the grid distribution. The concept of uniformity can be extended in terms of equal cell volumes, equal arc lengths, or some other mapping property as formally expressed in Ref. 14 (pp. 147-166). Directional effects are evident on comparing volumes with arc lengths. Cells with fixed volumes can become elongated in one direction, while cells with fixed edge lengths can be collapsed. Various other candidates for the definition of uniformity can be similarly compared and each can be seen to have some defect. On balance, however, arc lengths and volumes do provide reasonable definitions for the grid as a whole in most circumstances. As with uniformity, there are also distinct choices for curvature clustering. Along any curve in the surface, curvature is split into a geodesic part and a normal part. The geodesic part is a measure of directional changes within the surface, but not of the surface. The normal part gives the rate at which the surface tangent planes vary as the curve is traversed and, consequently, is a measure of the directional changes of the surface. In two dimensions, both Gaussian and mean curvatures are defined in terms of (principal) normal curvatures (Ref. 14, p. 54). The direct use of either Gaussian or mean curvatures, particularly in the alternating direction context, would possibly neglect clustering for some desired direction. For example, the Gaussian curvature of a conical surface vanishes at every point except the vertex and could not be used for clustering along an arbitrary curve in the cone. Similarly, mean curvature vanishes on minimal surfaces (Ref. 14, pp. 163-166) and cannot be used for clustering on such surfaces. A more reasonable and easily applied approach is to use normal curvatures directly. This is done by the choice weights in the alternating direction method.

The Alternating Direction Method

An alternating direction method is obtained when a transformation of surface coordinates is iteratively created by separate transformations along the coordinate curves that are grouped by direction and cycled toward convergence. With the separation into curves, a sequence of one-dimensional transformations results. The independent variable in each transformation is taken to be the curve arc length. There is no loss of generality in this choice since all curve parameterizations can be expressed relative to arc length. In particular, the new parameterization is a function of arc length that, in turn, is a known function of the old one. Moreover, when these functions are strictly monotone, the composition, which is the transformation from old to new, is also strictly monotone and consequently nonsingular. Collectively, when all curves in a given direction are nonsingularly reparameterized, a nonsingular transformation results under the modest requirement that the consequent transverse tangents are not aligned with curve tangents. Altogether, after any finite number of repetitions of the complete iterative cycle, nonsingularity is usually preserved when there is strict monotonicity in each one-dimensional transformation relative to arc length.

Along each coordinate curve in the ith direction, the general transformation t_i as a function of arc length s_i is determined by the proportionality

$$dt_i \propto w_i(s_i)ds_i \tag{1}$$

relating differential elements dt_i to ds_i by means of a nonzero weight function $w_i(s_i)$. To get a pointwise distribution along the curve, a fixed dt_i is used in correspondence with an assumed uniform distribution in the curvilinear variable t_i. When the weight is equal to unity, the differential elements are in a direct proportion and a uniform distribution of points results. Deviations from unity then yield deviations from arc length uniformity. Since a proportionality statement has an implied scale factor, there is no loss in generality in assuming that the weight is never less than unity. Under this assumption, the weight can be expressed as the sum of unity and a nonnegative but finite term. The separated term then represents clustering relative to uniform conditions. At locations where

the term exceeds zero, the weight exceeds unity and the arc length increment of Eq. (1) must shrink in order to match a fixed increment in t_i The result is a pointwise distribution along the surface curve with a spacing that is inversely proportional to the weight. Consequently, the weight is equally distributed with respect to the arc length. This is often referred to as an "equidistribution" of the weight.

To form the clustering term, we first specify the quantities requiring resolution as they increase from zero in magnitude. Given the magnitudes of m_i such quantities, the simplest form for the term is simply a linear combination. The resulting weight is then of the form

$$w_i(s_i) = 1 + \sum_{j=1}^{m_i} c_{ij} M_{ij}(s_i) \tag{2}$$

where $M_{ij}(s_i)$ is the magnitude of the jth quantity and c_{ij} a non-negative adjustable constant; the size of which gives the level of importance attached to the jth quantity relative to the others and to uniformity.

When the variables t_k in directions other than i are given specific values, a fixed i-direction curve is selected and the proportionality factor for Eq. (1) is a constant adjusted to the assumed minimum of unity for the w_i of Eq. (2). The factor for all i-direction curves is then a function of all variables except t_i. Denoting it by A_i, the proportionality statement of Eq. (1) is converted into the equation

$$A_i \mathrm{d}t_i = w_i \mathrm{d}s_i \tag{3}$$

where for notational convenience the functional dependence has been deleted. At this stage, A_i can be determined either implicitly in the form of boundary conditions or explicitly by an integration over the entire curve.

The implicit determination comes from a conversion into a partial differential equation obtained from Eq. (3) with a division by $\mathrm{d}t_i$ and a differentiation of the result. Before the division, the differential arc length $\mathrm{d}s_i$ of an i curve is expressed in terms of the differential $\mathrm{d}t_i$ by using the metric expression

$$(\mathrm{d}s)^2 = g_{jk} \mathrm{d}t_j \mathrm{d}t_k \tag{4}$$

for the arc length s of an arbitrary curve on the surface. Here, j and k are each summed from 1 to the number of spatial dimensions for the surface. When the surface is contained in an m-dimensional Euclidian space with Cartesian coordinates $(u_1, u_2, \ldots, u_m)$, the metric coefficients g_{jk} of Eq. (4) are given by

$$g_{jk} = \sum_{\ell=1}^{m} \frac{\partial u_\ell}{\partial t_j} \frac{\partial u_\ell}{\partial t_k} \tag{5}$$

where Eq. (4) can be interpreted as a Euclidian distance by means of the chain rule. Along the i curves, all $\mathrm{d}t_j$ vanish except $\mathrm{d}t_i$ and the metric of Eq. (4) reduces to

$$\mathrm{d}s_i = \sqrt{g_{ii}}\, \mathrm{d}t_i \tag{6}$$

upon taking square roots. When this is inserted into Eq. (3), the element $\mathrm{d}t_i$ can be removed by division, with the result that

$$w_i \sqrt{g_{ii}} = A_i \tag{7}$$

Upon differentiation with respect to t_i, the original proportionality factor A_i vanishes. After some minor manipulation, the partial differential equation for the ith direction becomes

$$\frac{\partial g_{ii}}{\partial t_i} = -2 g_{ii} \frac{\partial}{\partial t_i} (\log w_i) \tag{8}$$

where w_i is given by Eq. (2).

In the case of two-dimensional problems, the metric is completely determined by using Eq. (8) to get g_{11} and g_{22} and by using the equation for Gaussian curvature to get g_{12}, which gives a measure of orthogonality on the surface. With the metric, the surface grid can be constructed by means of line integrals. For detailed discussions on the use of such line integrals, the interested reader is referred to Refs. 15-17.

Alternatively, the Cartesian coordinate expressions for g_{ii} can be used directly from Eq. (5). With the physical space given in terms of the Cartesian coordinates, $u_1 = x$ and $u_2 = y$, and with a surface defined by values ψ in the perpendicular direction, the equation $u_3 = \psi$ together with Eq. (8) gives three equations to determine the Cartesian locations x, y, and ψ. When there is no surface clustering, $w_i = 1$ and the two equations from Eq. (8) then become

$$\frac{\partial}{\partial t_1}\left[\left(\frac{\partial x}{\partial t_1}\right)^2 + \left(\frac{\partial y}{\partial t_1}\right)^2 + \left(\frac{\partial \psi}{\partial t_1}\right)^2\right] = 0$$

$$\frac{\partial}{\partial t_2}\left[\left(\frac{\partial x}{\partial t_2}\right)^2 + \left(\frac{\partial y}{\partial t_2}\right)^2 + \left(\frac{\partial \psi}{\partial t_2}\right)^2\right] = 0 \tag{9}$$

which are the grid movement equations Ablow[2] solved by an alternating direction implicit (ADI) technique. With the direct use of either metric coefficients or Cartesian locations, partial differential equations are solved subject to boundary conditions representing the effect of the proportionality factor A_i that was removed by differentiation.

In contrast, the explicit determination of A_i can be accomplished by a direct integration of Eq. (3) over each i curve. For convenience, let the curve coordinate t_i vary from 0 to 1 as the curve arc length varies from 0 to a maximum length. An integration over the entire curve then gives A_i as the total weight distributed along the curve. The one-dimensional transformation is determined by a subsequent integration up to t_i and the corresponding s_i. With the A_i established, the result is given by

$$t_i = F_i(s_i)/F_i(L_i) \tag{10a}$$

where

$$F_i(s) = \int_0^s w_i(r)\mathrm{d}r \tag{10b}$$

and L_i is the maximum arc length along the curve. Specific cases of this form have been used by Dwyer et al.[5] Gnoffo,[10] Eiseman,[13] Ablow and Schecter,[1] and White.[4,5] Since the weights of Eq. (2) in the integrand of $F_i(s)$ in Eq. (10b) are never less than unity and since the accumulation of arc length is already a strictly monotone process, the transformation of Eq. (10a) is also strictly monotone, which means that it is nonsingular in one dimension. Collectively, then, the application to all i-direction coordinate curves yields a full transformation that preserves nonsingularity, assuming that transverse and i direction tangents are nonaligned. Upon alternating directions and, in general, upon cycling through all directions i a finite number of times, the resulting transformation is nonsingular. Upon convergence, the alternating direction strategy will then provide a solution to the implicit formulation of Eq. (8) for all directions i that are cycled. Although each step in the approach to convergence may be nonsingular, there is no mathematical guarantee of nonsingularity for the converged result. As a practical matter, however, convergence is rapid enough to provide the desired results with a small number of cycles that we know then to be nonsingular.

Algorithmic Construction

In the general discussion given above on the alternating direction method, nonsingular transformations were observed to result from a finite number of cycles through the various coordinate directions. Implicit in the observation was the

assumption that there was a continuum of curves in each such direction and moreover that each curve, after being transformed, remained unchanged except for its parameterization. When an algorithm is to be constructed, however, these assumptions are no longer valid. The loss of validity comes from the fact that the transformation at each step is ultimately defined by a finite number of points. The algorithmic objective is to move the finite number of points of an existing nonsingular grid into better positions that collectively form another nonsingular grid. Here, the grid nonsingularity is distinguished from that of the continuous transformations because the determination is based on finite and infinite computations respectively. To examine this distinction and ultimately the appropriate precautions that are needed in a construction, the most straightforward algorithm is considered first.

In the first algorithmic construction, each curve in sequence is parameterized by an approximation of the integral in Eq. (10) with a cumulative sum of weighted cordal lengths between its successive grid points. The sum is just a trapezoidal quadrature rule in which each cordal length is multiplied by the average of its end point weight in order to produce a centered weighting. To illustrate the parameterization process, suppose that $T(J)$, $W(J)$, and $S(J)$ are given arrays for the transformation, the weights, and the arc length, respectively, as the index J runs from the first curve point 1 to the last point J_{MAX}. Starting with $T(1)=0$, the cumulative sum is generated at each curve point with the basic loop on the expression

$$DELS = S(J) - S(J-1)$$

$$T(J) = T(J-1) + 0.5*(W(J) + W(J-1))*DELS \qquad (11)$$

as J runs from 2 to J_{MAX}. The transformation array T can then be normalized to conform to Eq. (10a), although this is not necessary since the only substantive requirement is that the values of T must be uniformly partitioned to determine the new parametric locations. For each J, the corresponding point in the uniform partition is given by

$$U = [(J-1)/(J_{MAX}-1)]*T(J_{MAX}) \qquad (12)$$

and is generally observed to fall between successive points in the T array. To find the T interval containing U, a search is performend. Most simply, although not optimally, it consists of a loop on the statement

$$\text{IF}(U.GT.T(K-1).\text{AND}.U.LE.T(K))L=K \qquad (13)$$

to determine the interval by the index L as K runs from 2 to J_{MAX}. Given the parametric interval, the next step is to get the new position P_{new} on the curve from the old positions P_{old} by the linear interpolation

$$P_{new}(J) = A\,P_{old}(L) + (1-A)\,P_{old}(L-1)$$

where

$$A = [U - T(L-1)]/[T(L) - T(L-1)] \qquad (14)$$

The interpolation is executed in a componentwise fashion with Cartesian coordinates. When this process is done for each J, the new curve is given entirely by the array P_{new}. In physical space, Dwyer et al.[5] used this simple algorithm on all curves in one direction, while Gnoffo[10] used it for two directions. The present author has used it on surfaces with gridded definitions, as would be generated by a numerical solution to a physical problem.

Although the simplest algorithmic construction outlined above works well in a significant number of cases, there is also a significant problem in the construction that can severely limit its range of application. The problem arises from the

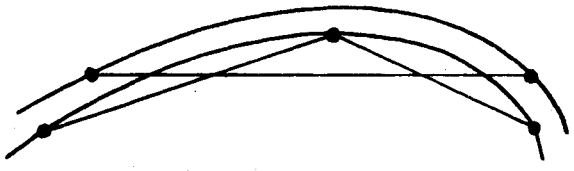

Fig. 3 Problem of closely spaced curves with curvature.

piecewise linear approximation of the curves and is observed when there are closely spaced curves with some curvature. To examine what can happen, an illustration of two such curves is locally displayed about part of a piecewise linear approximation in Fig. 3. In this figure, the piecewise linear approximation of the top curve is seen to intersect that of the bottom one, regardless of how well the original curves may be defined. This intersection represents a grid singularity even if the continuous transformation is nonsingular. With the grid movement restricted to just one direction, a continuous transformation was used by Dwyer et al.[5] to define a finite number of smooth fixed curves. The singularity, however, was not observed since the fixed curves never appeared simultaneously with both curvature and tight spacing. With the full alternating direction strategy giving adaptive movement in all directions, all curves must be defined by the grid points since the theoretically underlying continuous transformation is unavailable. Assuming the underlying continuum, the curvature can be estimated by finite differences and is a measure of the directional changes in the resulting piecewise linear curves. Together with tight spacing, a certain amount of curvature can cause the same overlap singularity as depicted in Fig. 3. To visualize the effect, a pair of nonoverlapping piecewise linear curves would simply be used for the fixed curves in place of the pair of curved arcs. The other piecewise linear curves would then represent the respective one-dimensional transformations from the first algorithm. Upon alternating directions, the overlap singularity is usually amplified since the transverse curves would then double back upon themselves. The result is a collapse of the procedure. The collapse has been observed by the present author and is almost certainly the reason that Gnoffo[10] had trouble with convergence in certain cases.

At the verge of collapse, the grid becomes excessively skewed and virtually unusable. However, skewness can be corrected by choosing weights that bias toward an equidistribution of deviations from orthogonality. On application, the other quantities would be balanced against both uniformity and orthogonality within the general weight of Eq. (2). Here, the correct balance is the best remedy against the collapse of the movement procedure since it insures a well-structured grid. However, it does not give an absolute guarantee against collapse.

To recapture the guarantee, a second algorithm can be constructed from the first by giving some transverse movement to the curves. Although there are various options as to when and how the movement is to be done, a specific and simple method is described here. After the first algorithm is applied in a given direction, movement is supplied to untangle the curves and then to compress the result should there be some boundary overlap caused by the untangling process. The untangling sweep is done by taking the curves in sequence and using the previous curve, which is then part of an untangled and thus nonsingular grid.

Curve entanglement is illustrated in Fig. 4. Note the two entangled points in Fig. 4b with the nonentanglement in Fig. 4a. At each point (K,J) on the current curve K, the corresponding point $(K-1,J)$ of the previous curve $K-1$ is used to define a transverse vector direction (which is indicated by an arrow). In going from the first point $J=1$ to the last point $J=J_{MAX}$, direction changes relative to the tangent vector of curve $K-1$ are detected by sign changes in the Jacobian and are saved in index form. In two dimensions, the cross product of the

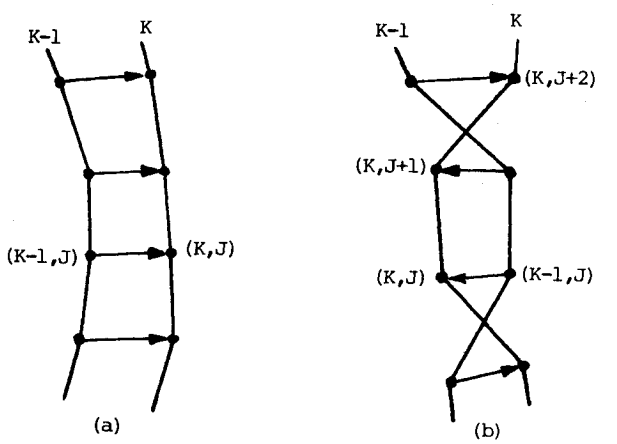

Fig. 4 Curve entanglement: a) no entanglement, b) two entangled points.

transverse and tangent vectors changes direction. Since both the first and last points are on the boundary, they are correctly directed, which means that the direction changes between them must occur in pairs. If there are such changes, the current curve must then be untangled from its predecessor between the first and second index locations and, continuing in sequence, between odd and even locations. In each instance, the odd location is included, while the even is not.

In Fig. 4b, the odd location J is followed by the even location $J+2$ that falls just outside of the entanglement. To keep the curve spacing in scale, the overlapping sections are moved to the current side of the previous curve by creating new transverse vector directions from the correct directions at the section end points and from the tangents of the previous curve. In the example of Fig. 4b, the vectors at $J-1$ and $J+2$ are used. The new section is then generated with a linear interpolation between the vectors and a subsequent addition of the results to the corresponding points of the previous curve. The interpolation is for magnitudes and angles relative to the tangents. When the existing overlapping sections have been corrected for each curve in sequence, the untangling sweep is complete and the resulting grid is nonsingular, but may possibly overlap the original boundary represented by the last curve. Such an overlap is detected if the last curve differs from the given boundary. Upon detection, the grid can then be compressed in a nonsingular fashion by proportionately shrinking each transverse increment.

The Choice of Weights

In the alternating direction method, the coordinate movement and structure is controlled by the choice of weights. The weights for a given direction determine the clustering criteria along that direction and thereby determine the structure of the transverse coordinate surfaces. Transverse phenomena are then resolved with these coordinates, which may also be appropriately aligned over significant regions. When all directions are considered, the control for both resolution and alignment is at its most flexible level. In addition to the use of different transverse coordinate surfaces, different weighting may also be applied to some advantage. For this reason, the general weight function of Eq. (2) appears with the subscript i. When the solution surface is taken to coincide with the physical region, the solution gradients are usually resolved by employing the magnitudes M_{il} of the first derivatives with respect to curve arc length in the i direction. With $m_i=1$ in Eq. (2), the case examined by Gnoffo[10] is then obtained. With $m_i=2$ and the inclusion of second derivatives by M_{i2}, the unidirectional adaptive technique of Dwyer et al.[5] is obtained by setting $i=1$. However, second derivatives give at best a linearized approximation to the curvature of the solution surface curve. This does not take into account the fact that a solution surface exists around the curve as would have been measured had the normal curvature been approximated. The problem there is that clustering can occur because of curves changing direction within the surface (geodesic curvature) rather than the surface itself changing direction. However, in the unidirectional context, this problem is not as sensitive as it is in an alternating direction approach.

When the fundamental adaptive data are established in the form of a solution surface, the resolution of the salient solution gradients is accomplished with a uniform surface grid. The grid is generated with unit weights and may be used either directly or after a projection straight down to the physical space below it. In one dimension, White[3,4] employed a direct use for the curve determined by the entire solution vector. Also, to resolve the direction changes in the curve, the weight from Eq. (2) is determined with M_{il} as the magnitude of curvature and with $m_{il}=1$ where $i=1$ is the only direction. This yields the cases examined by Ablow and Schecter[1] in projected form for two-point boundary value problems and by Eiseman[13] for arbitrary curves that represent the boundaries for two-dimensional regions.

In contrast to the resolution of curves, the resolution of two-dimensional solution surfaces is more subtle. The basic requirement is to cluster points where the surface changes direction, regardless of how the curves change direction within the surface. Otherwise, internal clustering could wastefully appear due to the movements of coordinate curves during the alternating direction cycles. Along with the proper treatment of the internal coordinate structure, the next requirement arises from the need to adequately resolve the physical boundaries that, when lifted to the surface, become surface boundaries. At nontrivial boundaries, the need then shifts toward the resolution of the curve within the surface. Upon combination, the mechanism for boundary treatment must then be damped as the interior region is penetrated. Both requirements and the shift between them are accomplished with the splitting of curvature into two parts.

To analytically formulate the splitting, the solution surface must first be prescribed. In the algorithm, it is given in embedded form by the position vector

$$\boldsymbol{P}(u,v)=[x(u,v),y(u,v),\psi(u,v)] \tag{15}$$

where u and v are surface coordinates that determine the Cartesian locations (x,y,ψ) in three dimensions. The physical region is defined in the $x-y$ plane, while a norm of the solution quantities in need of dynamic resolution determine the ψ values. An arbitrary curve on the surface can now be described by its preimage in (u,v) space as $[u(t),v(t)]$ for some parameter t. With a slight notational abuse, the curve on the surface will be expressed with a single variable as $\boldsymbol{P}(t)$ rather than $\boldsymbol{P}[u(t),v(t)]$. Moreover, upon choosing t to be the arc length, the curve will be denoted more simply by $\boldsymbol{P}$, with any derivative indicated by a prime. The first derivative yields the curve tangent vector $\boldsymbol{P}'$, which is of unit length due to the definition of arc length and which gives the direction of the curve. The derivative of $\boldsymbol{P}'\cdot\boldsymbol{P}'=1$ then yields $\boldsymbol{P}''\cdot\boldsymbol{P}'=0$, which means that $\boldsymbol{P}''$ is normal to the curve. In the Frenet theory of curves,[14] the direction of the second derivative $\boldsymbol{P}''$ is given by a unit normal vector $\boldsymbol{n}$, while the magnitude is defined to be the curvature κ. This gives the rate of change in the direction $\boldsymbol{P}'$ of the curve. The cross product of $\boldsymbol{n}$ with $\boldsymbol{P}'$ completes a set of orthogonal unit vectors known as the Frenet frame. This frame conforms only to the curve and is used solely to measure the curve properties by means of the Frenet formulas, the first of which we have already derived as $\boldsymbol{P}''=\kappa\boldsymbol{n}$. Since the curve lies in the surface, a unit normal $\boldsymbol{N}$ to the surface is also normal to the curve and can be used to form a frame that can be employed to measure surface properties. In particular, as the curve is traversed, the rate at which surface tangent planes change direction is just the rate of change

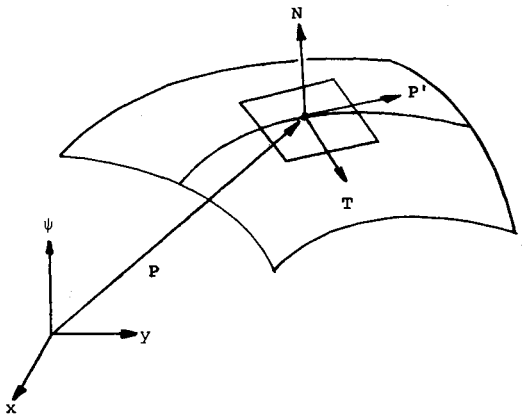

Fig. 5 Surface adjusted frame for a curve in the surface.

of N and is called the normal curvature κ_N. Within each tangent plane, the normal to the curve direction P' is given by the cross product $T = P' \times N$. Its rate of change is a measure of direction changes for the curve within the surface and is referred to as the geodesic curvature κ_g. Altogether, the rate of change of the curve direction P' can be expressed both in the Frenet frame and in the surface adjusted frame comprising T', P, and N. This yields the two equalities

$$P'' = \kappa n = \kappa_g T + \kappa_N N \tag{16}$$

corresponding to the two frames, respectively. An illustration of the surface frame is given in Fig. 5. Both normal and geodesic curvature are now obtained from the frame properties under dot products and are given by

$$\kappa_N = N \cdot P'' \tag{17a}$$

and

$$\kappa_g = T \cdot P'' \tag{17b}$$

where for completeness,

$$T = P' \times N \tag{17c}$$

and

$$N = B/\sqrt{B \cdot B} \tag{17d}$$

with

$$B = \frac{\partial P}{\partial u} \times \frac{\partial P}{\partial v} \tag{17e}$$

determined from Eq. (15). Upon considering all curves through a point, the principal curvatures are defined by the maximum and minimum values of the normal curvature κ_N. Their product and average are, respectively, the Gaussian and mean curvatures.[14] However, both have limitations for clustering, since both vanish on surfaces where some resolution is needed. Respective examples are given by cones and saddle-shaped surfaces where linear segments and opposing signs cause the vanishing. By contrast, normal curvatures will always provide clustering along the current direction rather than being effected by the properties in other directions. As a consequence, the direct application of normal curvature is preferred. On application for the alternating direction algorithm, finite differences are used to estimate all derivatives in Eqs. (17), remembering that P' and P'' are arc length derivatives along the direction of the current sweep, which may be either u or v.

In cases where the boundaries of the physical region are mildly varying, the normal curvature from Eq. (17a) is used alone to cluster the points relative to arc length uniformity along each surface coordinate curve taken in the iterative sequence to complete a cycle. On any such curve, the balance between uniformity and curvature is accomplished with the weight

$$w(s) = 1 + c\,|\kappa_N(s)| \tag{18}$$

which is a special case of Eq. (2) where, for notational convenience, the directional subscript i has been dropped. The intensity of curvature clustering is controlled with the value of c, which is increased from zero to produce an increase from arc length uniformity. The uniformity measure is represented by the unity in Eqs. (2) and (18) and produces the same effect as the smoothness integral in Brackbill and Saltman's variational technique.[8] By contrast, the excess gradient method of Rai and Anderson[7] does not have such an effect built into it. As a consequence, their method will not automatically remove clustering when it is no longer needed. To correct for this problem they had to augment their scheme by linearly including spatial distance under their gradients.

The weight of Eq. (18) next determines the transformation of Eq. (10) in the form

$$t = \frac{s + cI(s)}{s_{\max} + cI(s_{\max})} \tag{19}$$

where

$$I(s) = \int_0^s |\kappa_N(r)|\,dr \tag{20}$$

$s_{\max}$ is the total arc length of the curve, and the directional index i is implicitly assumed. Following a development of Dwyer,[6] the controlling constant c can be replaced by the fraction

$$f = \frac{cI(s_{\max})}{s_{\max} + cI(s_{\max})} \tag{20}$$

of points that are assigned to curvature. Upon inversion it is given by

$$c = \frac{fs_{\max}}{(1-f)I(s_{\max}) + \epsilon} \tag{21}$$

when $\epsilon = 0$. The number ϵ is introduced here as a small positive number to guarantee a finite c in cases where no curvature is present. In a straightforward manner, the use of suitable fractions can be established for the general weights of Eq. (2). The use of fractions is particularly helpful when precise numbers of points are to be placed along a high-curvature region that cuts across all coordinate curves of a given family. The limitations arise, however, when the region misses some of the coordinate curves. As a consequence, the fraction or equivalently c must then be dynamically chosen to make the transition smooth.

In the algorithm, the weight function coefficients are dynamically chosen to change from curve to curve. The purpose is to give the correct balance between properties. In the case of the curvature balance in Eq. (18), curves that miss a severe peak should clearly be treated differently than curves that do not. Otherwise, the peak could consume most of the curve arc length and could then take all movable grid points along the curve. In such a predicament, the transition from the peak to the remaining region can be abrupt, not to mention the lack of resolution for other significant but less severe peaks. The transitional regions, however, are more important here since the climb to great peaks is usually quite smooth and well resolved with few points when viewed as a surface grid. As a consequence, the balance is shifted to pull the grid points

more strongly to the high-curvature regions that are just the transitional regions. The shift is accomplished with the actual distance and an estimate of the shortest possible surface distance along a coordinate curve that connects the boundary points. Upon a projection into physical space, the underlying curve may be viewed as a curved bottom leg of a distorted right triangle. With the vertical leg as the change in altitude $\Delta\psi$ between the boundary points, the estimate is given by an application of the Pythagorean theorem using the total arc length $p_{\max}$ of the projected curve. The ratio of the actual distance $s_{\max}$ to the estimated shortest distance is the critical shifting mechanism and is given by

$$R = s_{\max}/\sqrt{p_{\max}^2 + (\Delta\psi)^2} \tag{22}$$

When there is little or no vertical variation in the surface curve, the ratio is near its minimum, which is unity. In such cases, there are no large peaks, direction changes, and transitional regions; hence, there is no real need for curvature attraction. As large peaks are formed relative to the mildest conditions connecting the boundaries, the surface arc length will substantially increase relative to the minimal distance as expressed by the ratio. This increase in R is applied in a saturated form by using a hyperbolic tangent. On combination with the fraction of Eq. (20), the curvature weighting coefficient of Eq. (21) is modified to give

$$c = \frac{f s_{\max} \tanh D(R-1)}{(1-f)I(s_{\max}) + \epsilon} \tag{23}$$

The number f is interpreted here as the maximum fraction of points that can be assigned to resolve normal curvature along any curve of a given direction. The constant D gives the rate at which this maximum fraction is approached as the surface changes in shifting from curve to curve.

With the weights established and dynamically adjusted for the varying requirements of normal curvature, the next class of problems to be considered are those in which geodesic curvature becomes important. These are cases when boundary curvature in the physical region is in need of resolution. A typical example is given by the numerical simulation of compressible flow about an airfoil in which there is some pressure disturbance on top. With values of pressure defining the solution surface, the airfoil contour is lifted to form a surface boundary that cannot be detected with normal curvature and that is still in need of refinement in order to accurately model the solution properties about the highly curved leading edge. The geodesic curvature is required for such boundaries since it is the curvature of the curves within a surface. Thus to resolve both the surface and its boundaries, the general weight of Eq. (2) becomes

$$w(s) = 1 + c_1 |\kappa_N(s)| + c_2 |\kappa_g(s)| \tag{24}$$

deleting the directional index i. As before, to dynamically adjust the requirements for surface resolution, the parameter c_1 is assumed to contain the same hyperbolic tangent factor, which is applied to a constant determined by a specified fraction for maximal conditions. Since the influence of the geodesic curvature is desirable only near the boundaries, the parameter c_2 must decay upon approaching the interior in order to avoid unnecessary internal clustering because of direction changes inherent in the curves but not the surface. In the example of an airfoil, the geodesic curvature is needed only for the local region adjacent to the airfoil and only for the surrounding curves, not the outward ones. On these curves, the effect can be achieved when c_2 is proportional to a quantity of the type $(1 - d/d_{\max})^m$, where d is a distance measure from the airfoil and where increases in m will shrink the region of influence for geodesic curvature. Along the radially outward curves, by contrast, c_2 is taken to be zero, which yields the weight of Eq. (18) with the dynamic adjustment of Eq. (23).

In addition to the resolution of various disturbances in the solution, the normal curvature also serves the need to align coordinate curves with such disturbances, regardless of other weighting quantities such as the geodesic curvature considered above. The pull toward alignment, while restricted by the basic grid topology, is a pull of entire coordinate curves or significant parts of them into regions where the surface changes direction. Since the disturbances are represented by folds in the solution surface, the corresponding regions of direction change give the alignment upon which the pull is based. The alignment problem has previously been considered by Rai and Anderson,[7] who established the capability within their adaptive method and, moreover, demonstrated the inherent advantages for the numerical simulation of shock waves.

In addition to the resolution and alignment strategies with curvature, other weighting quantities may also be employed to some advantage. One fairly broad collection of such quantities consists of all smooth positive functions of the curvilinear variables. In each case, the typical application would be to shift a family of curves in a given direction or to cluster curves or points about a specific curve or point, regardless of the location on the solution surface or in physical space. As mentioned earlier, the structural integrity of the grid can also be enhanced by using deviations from orthogonality as a weighting quantity. A further possibility comes from volume weighting by using the Jacobian of the transformation. By including the Jacobian as a quantity in any of the weights, the definition of uniformity can be altered. As the weighting coefficient is increased from zero, the definition shifts from equal arc length along coordinate curves toward equal volume for all cells. Under a complete shift, the entire transformation becomes an essentially volume-preserving mapping as defined in Laugwitz.[14] In another study, Eiseman[11] has directly used volume elements to develop a mean value relaxation technique for adaptive movement. The technique is based upon a local difference molecule that is constructed with weights of the form considered here applied to local volume elements. With actual volumes as factors, the unity in the general weight of Eq. (2) then represents a pull toward an essentially volume-preserving map as the measure of uniformity. This use of the measure is distinguished from the present case by observing that the Jacobian is essentially a factor applied to the whole weight rather than as a separate quantity within the weight.

Applications

The alternating direction movement algorithm is applied to adapt a grid to surface data and weights that are given with respect to the grid itself. For the movement to be smooth, both the surface and the weights must also be smooth. Otherwise, ripples in either the surface or the weights will be translated into ripples in the grid and most likely into an unstable situation. As a consequence, a smoothing step or steps is recommended if the gridded data are not already smooth. In the overall solution adaptive context, movement is first applied to conform more closely to initial conditions and, subsequently, to resolve the solution as it evolves. During the evolutionary part, however, a grid that follows the significant solution properties will often tend to produce smooth results regardless of the numerical method of solution. With weights in physical space, Dwyer[6] has observed this tendency and has supplied some supporting theory for the cell Reynolds number problem. Assuming the worst possible case in which ripples arise for some unforeseen reason, a smoothing step should be applied as frequently as needed. The smoothing steps in the overall procedure can be accomplished by a short iteration on local averages to filter out small fluctuations in the data while retaining its basic character. It should be emphasized, here, that only a solution-generated surface and the corresponding weights are smoothed; the solution itself is not, for then it would be unreasonably diffused.

The basic part of the solution adaptive process is the grid movement scheme. To test the alternating direction method

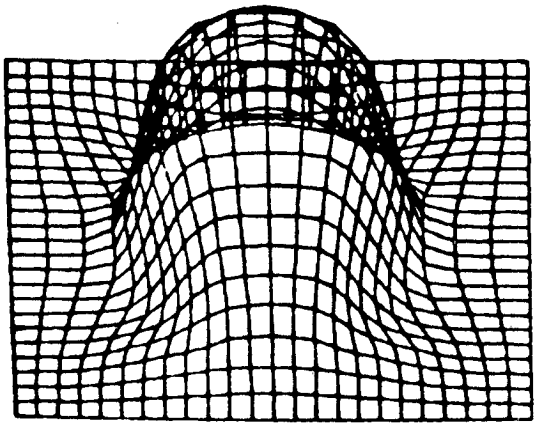

Fig. 6 Three-dimensional view of uniform surface grid.

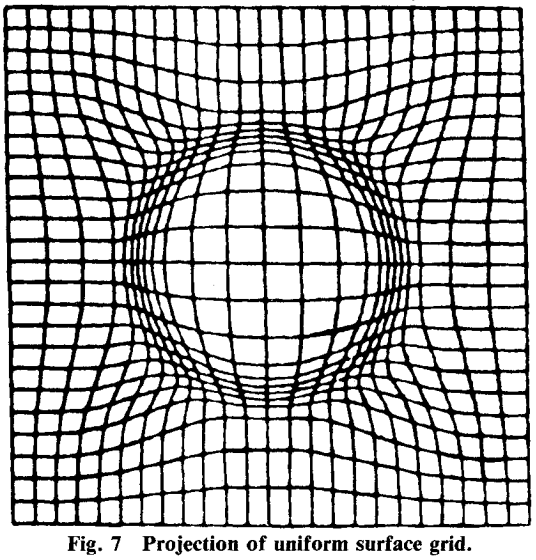

Fig. 7 Projection of uniform surface grid.

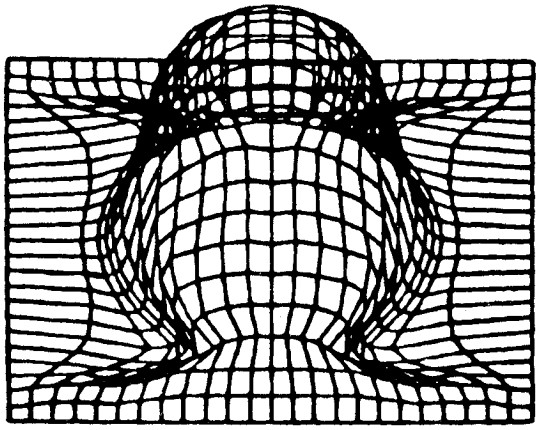

Fig. 8 Three-dimensional view of surface grid with transverse normal curvature attraction.

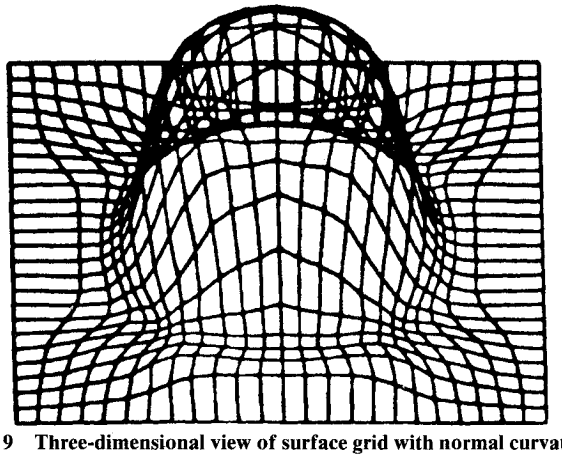

Fig. 9 Three-dimensional view of surface grid with normal curvature attraction in both directions.

and, in particular, the effect of normal curvature weighting, a sequence of test cases has been selected to examine the movement without outside interference from a particular physical problem or numerical method. Since the main purpose of adaptivity is to adequately resolve disturbances, a smooth artificially created disturbance is sufficient for our purposes. In the cases examined, first, a single disturbance on the unit circle is considered by setting

$$\psi(x,y) = \tfrac{1}{2}[1+\tanh[3(1-x^2-y^2)]] \qquad (25)$$

to model a two-dimensional solution surface over the region defined with x and y varying from -2 to $+2$. In going from the outside to the inside of the circle, the function ψ experiences a rapid change from approximately 0 to 1. In all cases, the initial surface grid is a 25×25 uniformly spaced Cartesian grid in x and y with corresponding vertical locations determined by evaluations of $\psi(x,y)$. The movement in each example comes from three cycles with the alternating direction algorithm. This is usually sufficiently close to a final state. To measure the degree of closeness, a formal error analysis would be required to compare the discrete positions against the analytic formulation represented by Eq. (8). In the first case, the weight function was set to unity. The resulting grid on the surface is displayed in Fig. 6. In this three-dimensional plot without hidden line removal, the view is toward one side of the square for a grid that is the same for all sides. The covering of the surface appears to be uniform in front and, because of symmetry, it must then be uniform everywhere. Upon projection, the uniform surface grid yields the gradient clustered grid in Cartesian (x,y) space given in Fig. 7. When the weight is unity in the viewing direction and is the normal curvature balance of (Eq. 18) in the transverse direction, the coordinate curves along the direction of sight become clustered and aligned with the circular disturbance. This effect can be observed by a direct examination of Fig. 8. Along the direction of sight, the base of the disturbance is most easily viewed, from which the clustering and alignment is apparent. The alignment in this case is sufficiently strong to display the tendency for a single family of coordinate curves to follow most of the base region. The opposing force against a complete following of the base comes from the choice of grid topology, which is a Cartesian rather than a polar-type format. By using the same normal curvature in both directions, symmetry is restored while the alignment is retained, but with curves that equally share the base of the disturbance. The comparable three-dimensional plot is given in Fig. 9 and its projection is given in Fig. 10. With the resolution of normal curvature, the expected direction changes in the surface are resolved at both the base and the top rim of the disturbance. In the three-dimensional plot, only the base resolution is readily evident when compared with the uniform case displayed in Fig. 6. The complete resolution is more evident in the projected grid, which has a distinctly broader band of resolution than in the corresponding noncurvature case of Fig. 7. The broadening is clearly viewed on both sides of the circular disturbance, indicating rim and base resolution on the inside and outside respectively.

From the above circular cases, one could reasonably expect to use normal curvature to resolve such disturbances as shock waves in a well-aligned manner. As an illustration, an artificial parabolically shaped disturbance was prescribed by the surface

$$\psi(x,y) = \tfrac{1}{2}\tanh 3(x-0.3y^2+1.5) \qquad (26)$$

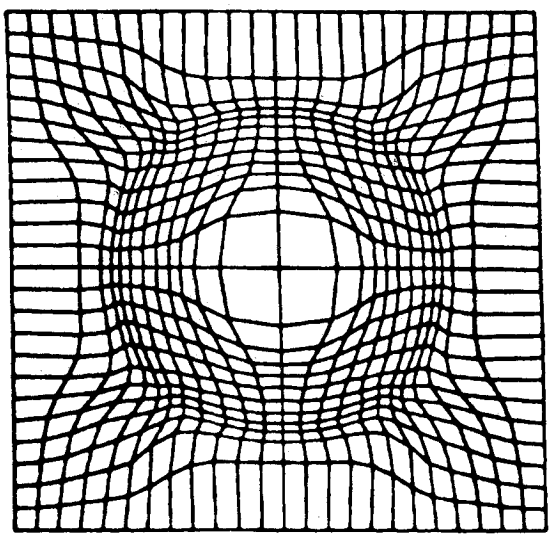

Fig. 10 Projection of curvature clustered surface grid.

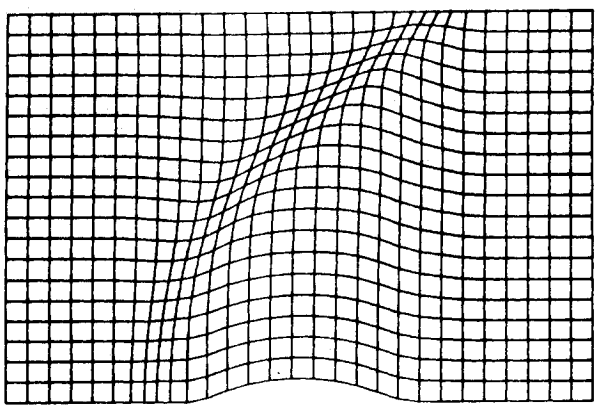

Fig. 11 Parabolic disturbance over a biconvex airfoil without curvature clustering.

over the rectangular region $[-3,3] \times [0,4]$ containing a biconvex airfoil at the origin. From an original sheared coordinate grid, points were moved without and with the use of normal curvature. The moved grids are displayed in Figs. 11 and 12, respectively. The noncurvature case of Fig. 11 resolves the disturbance in a cross-hatched manner as compared to the well-aligned and more finely resolved curvature case in Fig. 12. One's expectations for both resolution and alignment are then substantiated.

Summary

A general alternating direction method has been established to move a surface grid into better positions and thereby to adapt more closely to a solution that is used to create the grid. The application of the adapted grid was viewed to be either direct or upon projection back into the underlying physical space. The movement into better positions was determined primarily by the use of uniformity and normal curvature, which in the projected cases represents a resolution of gradients and their rates of change, respectively. By using surfaces, the movement is more fully and accurately determined than would have been the case with only curves. Upon restriction, the method reduces to the cases examined by prior investigators who separately considered one-dimensional surfaces, one adaptive direction in physical space, two adaptive directions in physical space, and a partial differential equation formulation without curvature on two-dimensional surfaces.

On an analytical basis, the method was seen to move points in a nonsingular fashion. On a discrete basis, a corresponding algorithm was formulated first to separately move points along each curve in the global cycle, only to find that closely

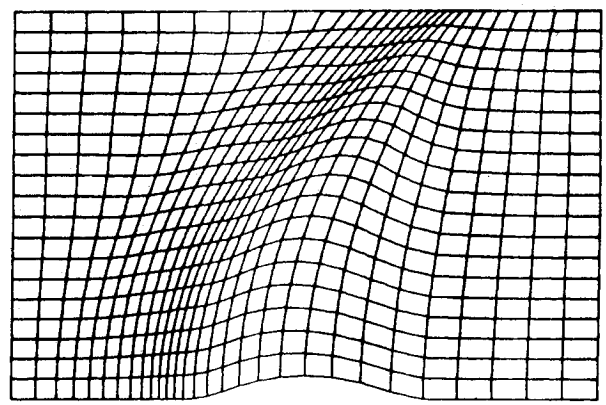

Fig. 12 Parabolic disturbance over a biconvex airfoil with normal curvature clustering in both directions.

spaced, highly curved curves could bump into each other and thereby lose the inherent nonsingularity. With a modification, the previous guarantee of nonsingularity was restored.

The weighting functions for grid control were examined first in the general context and then in particular. The basic requirement for surface resolution was established with the application of normal curvature. This was also augmented with a geodesic curvature balance in cases where boundary geometry required resolution. Relative to each type of clustering, Jacobian weights were seen as a mechanism to shift between arc length and volume measures of uniformity. On a two-dimensional example of a circular disturbance, the basic method was tested in a sequence of cases that clearly demonstrated the fundamental capabilities, including the predicted beneficial effect or normal curvature for both resolution and alignment. This was further illustrated with a parabolic disturbance over a biconvex airfoil.

Acknowledgments

This study was sponsored at different stages by the U.S. Air Force Office of Scientific Research and NASA under Grants AFOSR-82-0176 and NASA NAG 1-355, respectively. The interest, encouragement, and support from Carl Edward Oliver of AFOSR and Douglas L. Dwoyer of NASA Langley Research Center are gratefully acknowledged.

References

[1]Ablow, C. M. and Schecter, S., 'Campolytropic Coordinates," *Journal of Computational Physics,* Vol. 27, 1978, pp. 351-362.

[2]Ablow, C. M., "Equidistant Mesh for Gas Dynamic Calculations," *Numerical Grid Generation,* edited by J. F. Thompson, North-Holland Publishing Co., Amsterdam, 1982, pp. 859-864.

[3]White, A. B., "On the Selection of Equidistributing Meshes for Two-Point Boundary-Value Problems," SIAM *Journal on Numerical Analysis,* Vol. 16, 1979, pp. 472-502.

[4]White, A. B., "On the Numerical Solution of Initial/Boundary-Value Problems in One-Space Dimension," SIAM *Journal on Numerical Analysis,* Vol. 19, 1982, pp. 683-697.

[5]Dwyer, H. A., Smooke, M. D., and Kee, R. J., "Adaptive Gridding for Finite Difference Solutions to Heat and Mass Transfer Problems," *Numerical Grid Generation,* edited by J. F. Thompson, North-Holland Publishing Co., Amsterdam, 1982, pp. 339-356.

[6]Dwyer, H. A., "Grid Adaption for Problems in Fluid Dynamics," *AIAA Journal,* Vol. 22, Dec. 1984, pp. 1705-1712.

[7]Rai, M. M. and Anderson, D. A., "Grid Evolution in Time Asymptotic Problems," *Journal of Computational Physics,* Vol. 43, 1981, pp. 327-344.

[8]Brackbill, J. U. and Saltzman, J. S., "Adaptive Zoning for Singular Problems in Two Dimensions," *Journal of Computational Physics,* Vol. 46, 1982, pp. 342-368.

[9]Yanenko, N. N., Kovenya, V. M., Lisejkin, V. D., Fomin, V. M., and Vorozhtsov, E. V., "On Some Methods for the Numerical Simulation of Flows with Complex Structure," *Lecture Notes in Physics,* Vol. 90, Springer-Verlag, Berlin and New York, 1978, 565-578.

[10]Gnoffo, P. A., "Vectorized, Finite-Volume, Adaptive-Grid Algorithm for Navier-Stokes," *Numerical Grid Generation,* edited by J. F. Thompson, North-Holland Publishing Co., Amsterdam, 1982, pp. 819-836.

[11]Eiseman, P. R., "Adaptive Grid Generation by Mean Value Relaxation," *Advances in Grid Generation,* edited by K. N. Ghia and U. Ghia, ASME FED, Vol. 5, 1983, pp. 29-34 (also to appear in *ASME Journal of Fluids Engineering*).

[12]Mastin, C. W. and Thompson, J. F., "Adaptive Grids Generated by Elliptic Systems," AIAA Paper 83-0451, Jan. 1983.

[13]Eiseman, P. R., "A Multi-Surface Method of Coordinate Generation," *Journal of Computational Physics,* Vol. 33, 1979, pp. 118-150.

[14]Laugwitz, D., *Differential and Riemannian Geometry,* Academic Press, New York, 1965.

[15]Schindler, G. M., "The Intrinsic Geometrical and Physical Properties of Plane Steady Gas Flow," SIAM *Journal on Applied Mathematics,* Vol. 42, No. 1, 1982, pp. 10-26.

[16]Eiseman, P. R., "Orthogonal Grid Generation," *Numerical Grid Generation,* edited by J. F. Thompson, North-Holland Publishing Co., Amsterdam, 1982, pp. 193-233.

[17]Warsi, Z.U.A. and Thompson, J. F., "Numerical Generation of Two-Dimensional Orthogonal Curvilinear Coordinates in an Euclidian Space," NASA CP2166, 1980, pp. 519-544.