Ann. Rev. Fluid Mech. 1985. 17: 487–522

GRID GENERATION FOR FLUID MECHANICS COMPUTATIONS

Peter R. Eiseman

Department of Applied Physics and Nuclear Engineering,
Columbia University, New York, NY 10027

INTRODUCTION

Fluid mechanics is understood in a descriptive way through experimental observation, mathematical analysis, and numerical simulation. When the understanding is required for flows with complex internal structure and with complicated regional boundaries, insight is gained primarily from experiments and simulations. Numerical simulations are motivated by the prospect of economically obtaining a detailed flow-field description. In each instance, a governing system of flow equations is analytically formulated over the region and is solved in an approximate form on a computer. Various numerical methods have been devised for such approximations, and most depend upon some representation of the flow-field region by means of finitely many points and the interconnections between them. When each regional boundary is given by a sequence of connected points, the efficiency and accuracy of the various methods are enhanced, since boundary conditions can be applied without interpolation. Further enhancement comes when the connectivity pattern is regular. The most regular and thus preferable patterns are those that result from coordinate transformations.

With the application of transformations, regions with topological complexity are consistently treated even when points are in motion. During the course of simulation, motion can be advantageously used to adaptively resolve the significantly varying solution quantities. From a geometric viewpoint, the quantities determine a surface over the physical region. The resolution of the surface as it evolves then determines the adaptive movement.

0066-4189/85/0115-0487$02.00

Because of the typically required generality, most of the useful coordinate transformations are nonorthogonal. While many two-dimensional regions or surfaces can efficiently be given conformal coordinates to match a variety of geometric configurations, pointwise distributions cannot be arbitrarily specified on boundaries. Upon specification, conformality with its simple metric structure is no longer available. The next best structure comes from orthogonal coordinates: Availability is also limited, but not as severely as with conformal coordinates. In higher dimensions both are virtually eliminated because of their severely restricted range of applications.

Two primary categories for arbitrary coordinate generation have been developed. They are algebraic methods and partial differential equation methods. They include both fixed and moving grids, and control over the results is the main issue in the respective developments. Control can be applied precisely and adaptively and in a variety of ways. In this review, these methods are analyzed by examining their strengths and weaknesses. Before embarking upon the detailed methodology involved in coordinate generation, some general observations are first made concerning applications. This establishes the overall setting in terms of geometry, topology, and motion.

THE APPLICATION OF COORDINATE TRANSFORMATIONS

With one transformation, Cartesian coordinates on a rectangular region are typically mapped into coordinates that cover the physical region and match boundaries. When the transformation is applied to a discrete representation, the connectivity pattern is preserved. On a rectangular region, regularity arises naturally from uniform spacing in each Cartesian direction. The simplest pattern occurs when the points are connected only along coordinate directions. This is called a Cartesian grid. Under the mapping, the Cartesian grid goes into a corresponding physical-space grid, where boundaries are defined by discrete coordinate curves. With the transformation, the flow-field computations can be performed entirely on the fixed Cartesian grid, regardless of the geometry or motion of the boundaries. The Cartesian grid then represents the rectangular computational space.

The same computational space can be used to treat multiply connected physical regions that arise when there are a number of solid bodies in the field. The basic methods come from the identification of Cartesian-coordinate sections with either artificial or actual boundaries. Artificial boundaries are used to connect bodies with the physical boundary or with other bodies to essentially reduce the number of bodies. These connections are transmissive boundaries for the fluid and are called branch cuts, patches, or simply cuts. Actual boundaries are determined by the geometry of the physical bodies. These are prescribed in correspondence with either internal or boundary Cartesian sections. The internal sections are used alone as a slit or in a combination to remove a local region. In physical space, the mapped result is wrapped around the given body. With the slit, physical boundary data must be doubly specified so that its image is opened. With the local region, the boundary grid inherits the Cartesian corners or edges even if they do not physically exist. In the contrary sense, any Cartesian section may also map to a physical boundary with corners or edges. Barring irregularities in the physical boundaries, the Cartesian identifications tend to produce further irregularities that appear as coordinate singularities on the boundaries. While special numerical treatment can usually be applied at singularities, the accuracy of a simulation can suffer, particularly if significant properties of the fluid occur near the boundaries. The fundamental reason for the singularities comes directly from the consideration of a multiply connected physical region. The use of a single coordinate transformation and the associated computational space is the primary cause for the placement of the singularities directly on the boundaries.

To move the singularities off the boundaries and away from potentially large flow variations, more than one coordinate system is required. The immediate consequence is an assembly of computational spaces and identifications. Altogether, this may appear as a general rectilinear region covered by a Cartesian grid. With junctures between systems corresponding to the identifications, coordinate transformations are then applied to produce grids for each system, which upon assembly cover the physical region in a patchwise fashion.

On a local level, the physical-space grid retains the simple, regular connectivity pattern of each coordinate grid. The only places where the pattern is altered are those that correspond to the moved singularities and that consequently represent only a minor part of the entire region. On a global level, the grid connectivity is changed in a way that usually represents a higher degree of conformity with the actual topology of the region. This occurs because solid bodies in the field often get individual local coordinate grids wrapped around them. The local grids essentially provide a good structure for the modeling of nearby fluid-mechanical phenomena.

From a global perspective, the outer boundaries of the local systems are considered as a new collection of bodies that can be fewer in number if some of them are joined together. The physical-space grid is typically constructed from such collections with the introduction of further artificial transmissive boundaries, which altogether determine its global topology.

With the selection of grid topology, certain families of coordinate curves or surfaces are selected for enhancement in number on a purely local basis. The local body-oriented grids provide an immediate example: The number of coordinate curves or surfaces that wrap around each body can be adjusted separately. As a consequence, the simulation process can be optimized in the vicinity of each body. A similar optimization is not generally available for one global transformation, since local requirements cause global changes that would be wasteful and possibly conflicting. As in the case of a single transformation, the simulation process can be done entirely with respect to a fixed computational space, even when the physical grid is in motion. The movement is restricted only by the fixed choice of grid topology, which is a possible problem only when distinct bodies collapse upon each other.

Once a grid has been generated to match the physical boundaries and to have some conformity with the regional topology, simulation accuracy can be enhanced through grid movement. While maintaining the simple, regular connectivity pattern, points are pushed into positions that more accurately represent the desired solution to the governing system of fluid equations. Although the entire solution would theoretically be needed to drive the pointwise motion, the intrinsic character is often quite accurately detected by a small number of salient quantities. For example, in supersonic compressible flow, pressure is readily identified as a salient quantity, since shock waves are detected by its rapid variation. Altogether, the collection of salient quantities efficiently summarizes the solution data. On application, a feedback cycle is formed in which the solution is advanced by some computational algorithm and the grid is moved in response. Such cycles that couple movement with solution procedures are called *adaptive grid techniques*. When the motion is supplemented or replaced by arbitrary local changes in the number of points, the adaptive procedures lose the logically regular ordering of grids and inherit a data-management problem. When overlapping grids are employed, the problem becomes one of data transfer and its numerical consequences. The discussion herein, however, is restricted only to coordinate transformations and the resulting grids.

ALGEBRAIC GRID GENERATION

With coordinate transformations defined by algebraic formulas, the physical region can be given a continuous description; this is often done in an explicit manner. From the application of such transformations, arbitrarily large grids can be efficiently generated. The definition and the application form a process known as *algebraic grid generation.*

Classically, transformations have been globally defined by analytic functions of a complex variable and by direct shearing. The analytic functions produce conformal coordinates that are inherently nonsingular and over which the fluid-dynamic equations assume their simplest general form. In addition, ,potential-flow solutions are readily available and are often quite useful. The fundamental limitations, however, are a loss of control over boundary distributions and a practical restriction to two dimensions. For an overview of conformal mapping techniques, the reader is referred to the surveys by Moretti (1980) and Ives (1982).

The limitations of conformal coordinates are removed with the classical shearing transformations. The price is nonorthogonality, which typically adds complication to the fluid-dynamic equations. In a practical sense, the complications have a minor impact and may not even appear in the discrete form. As a consequence, the primary general development starts with shearing transformations. These and the closely related Hermite transformations are just global interpolations in one direction. The first and most fundamental objective is to insert the maximum possible amount of control over the grid in a given direction. This is provided by the multisurface transformation that includes both shearing and Hermite transformations as special cases. The same control is available in all directions by using Boolean sums of multisurface transformations. In the earlier shearing and Hermite cases, the transformations were often called transfinite in view of the fact that a generally infinite number of boundary points were interpolated when directions were combined. This is in comparison with the finite use of only vertex data. Our discussion in this section starts with shearings and proceeds to the multisurface method; we then develop the Boolean operations and conclude with a look at applications.

Shearing and Hermite Transformations

When an arbitrary positive function f over some domain is multiplied by a scale factor η, the result is either above or below the original function, depending upon whether or not the scale factor is greater than or less than unity. In particular, when η varies from 0 to 1, the product ηf varies from 0 to f. At each fixed point in the domain, this variance monotonically traces out a vertical line segment extending from the domain up to the function value. A coordinate system is defined by taking the line segments together with the curves or surfaces corresponding to ηf. Since this process represents a vertical shearing of f onto its domain, it is called a shearing transformation. In two dimensions with Cartesian coordinates (x, y), the transformation is given by

$$\begin{aligned} x &= \xi, \\ y &= \eta f(\xi), \end{aligned} \tag{1}$$

and is illustrated in Figure 1 for the case of a nozzle geometry. In the displayed discrete form, a Cartesian (ξ, η) space grid is mapped onto the nozzle-conforming grid. The first equation determines the vertical lines with constant ξ values. Upon insertion into the second equation, it yields the scaled form with respect to the domain of f.

While the shearing as expressed in Equation (1) can be extended to include a general bottom boundary and arbitrary distributions in each variable, the basic process is done vertically. To perform a shearing in any direction, the transformation must be written in vector form. In this spirit, Equation (1) is rewritten for the position vector (x, y) as

$$(x,y) = (\xi,0)+\eta[(\xi,f(\xi))-(\xi,0)], \tag{2}$$

where the bottom and top boundaries are explicitly given by $(\xi, 0)$ and $(\xi, f(\xi))$, respectively. With bottom and top boundaries replaced by arbitrary vector functions $\mathbf{P}_1(\boldsymbol{\xi})$ and $\mathbf{P}_2(\boldsymbol{\xi})$, the general shearing transformation is given by

$$\mathbf{P}(\boldsymbol{\xi},\eta) = \mathbf{P}_1(\boldsymbol{\xi})+\eta[\mathbf{P}_2(\boldsymbol{\xi})-\mathbf{P}_1(\boldsymbol{\xi})] \tag{3}$$

for the position vector $\mathbf{P}(\boldsymbol{\xi}, \eta)$ with Cartesian components typically for n dimensions. The boundary coordinates are given by the vector $\boldsymbol{\xi}$ of length $n-1$. When $n = 2$, this becomes the curve parameterization ξ. For each fixed boundary coordinate $\boldsymbol{\xi}$, the shearing is done along the line segment from $\mathbf{P}_1(\boldsymbol{\xi})$ to $\mathbf{P}_2(\boldsymbol{\xi})$ as η varies from 0 to 1. Shearing in arbitrary directions is readily illustrated with the choice $\mathbf{P}_1(\xi) = (0,0)$ and $\mathbf{P}_2(\xi) = (\cos\ \xi,\ \sin\ \xi)$, which produces polar coordinates on the unit disk.

In two dimensions, Eiseman (1978) presented a thorough geometric analysis of shearing transformations and considered applications to

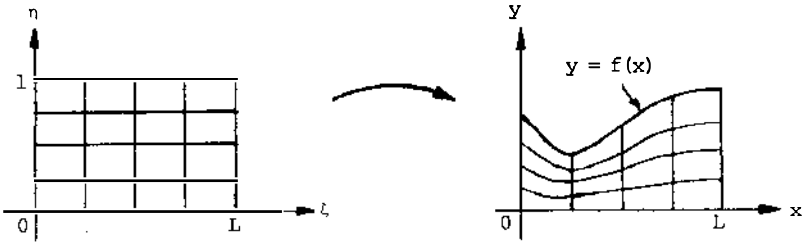


Figure 1 A shearing transformation for a nozzle.

cascades of airfoils. In a polar manner, a family of coordinate curves that looped about an airfoil were generated along lines from the airfoil to an outer boundary on which periodic alignment was enforced in a vertical direction matching top and bottom portions. With a periodic parameterization for the outer loop $\mathbf{P}_2$, the parameterization for the airfoil $\mathbf{P}_1$ was chosen to yield orthogonality at its surface. The purpose was to locally approximate boundary-layer coordinates. However, because of the linear pseudo-radial coordinate curves, the periodic matching was only possible for positions. The derivatives at the periodic juncture were discontinuous. In a more general context, the patching together of shearing transformations gives rise to slope discontinuities at the junctures. While a smoothing can be done as a postprocessing step, a direct algebraic formulation is nicer.

The most direct manner of producing smoothness is to use classical Hermite interpolation. To specify one derivative on each boundary in addition to locations, the cubic polynomials must be employed. In terms of the position vector, the cubic Hermite transformation is given by

$$\begin{aligned}\mathbf{P}(\xi,\eta) &= (1-3\eta^2+2\eta^3)\mathbf{P}(\xi,0)+\eta^2(3-2\eta)\mathbf{P}(\xi,1)\\ &\quad +\eta(1-\eta)^2\frac{\partial \mathbf{P}}{\partial \eta}(\xi,0)+\eta^2(\eta-1)\frac{\partial \mathbf{P}}{\partial \eta}(\xi,1),\end{aligned} \tag{4}$$

where the coefficients are chosen so that evaluations of $\mathbf{P}$ or $\partial\mathbf{P}/\partial\eta$ at the endpoints of η reduce to the given specifications on the right-hand side. For this to happen, each of these four possibilities produces the desired term with a coefficient of unity and the remaining three with coefficients of zero. Higher-order Hermite cases follow this same general pattern, first by allowing more derivatives at the boundaries to be specified. In the simple cubic case, a typical application is the specification of boundary orthogonality. In continuation, the most general Hermite form includes the same type of interpolation at a finite number of η values. Function values and derivatives up to various possibly distinct levels are interpolated as each point is taken in succession in going from boundary to boundary. The algebraic format is clear, but complicated, and thus is not given here. As a final remark on Hermite interpolation, we note that we could consider cases where the interpolation skips either function values or derivatives less than the highest interpolated order at some points. This is called a *defective Hermite interpolation.* For polynomials, it leads to the Hermite-Birkhoff interpolation problem, which is discussed in Prenter (1975, pp. 414–19).

The Multisurface Transformation

A set of constraints upon the grid arises in a natural way when there is a need to enhance accuracy and efficiency or to extend capabilities. Typical constraints come from the requirements for resolution, smoothness, topology, embedding, and uniformity. The shearing and Hermite transformations arose primarily to meet the requirements at the boundaries. These transformations are special cases of the general multisurface transformation that was developed by Eiseman (1979) to provide the necessary control over the grid at all locations and in a very precise sense. To illustrate the level of control and how constraints arise, an example of a grid from the multisurface transformation is displayed in Figure 2. The grid was designed for two flow-field simulations in which the effect of wind-tunnel walls could readily be examined. While the grid globally expands to an exact polar system for the far-field, local Cartesian grids were smoothly embedded about the airfoil. Taking the wind-tunnel walls to be Cartesian segments above and below the airfoil, we can apply boundary conditions directly for the wind-tunnel part, with concurrent grid alignment near the

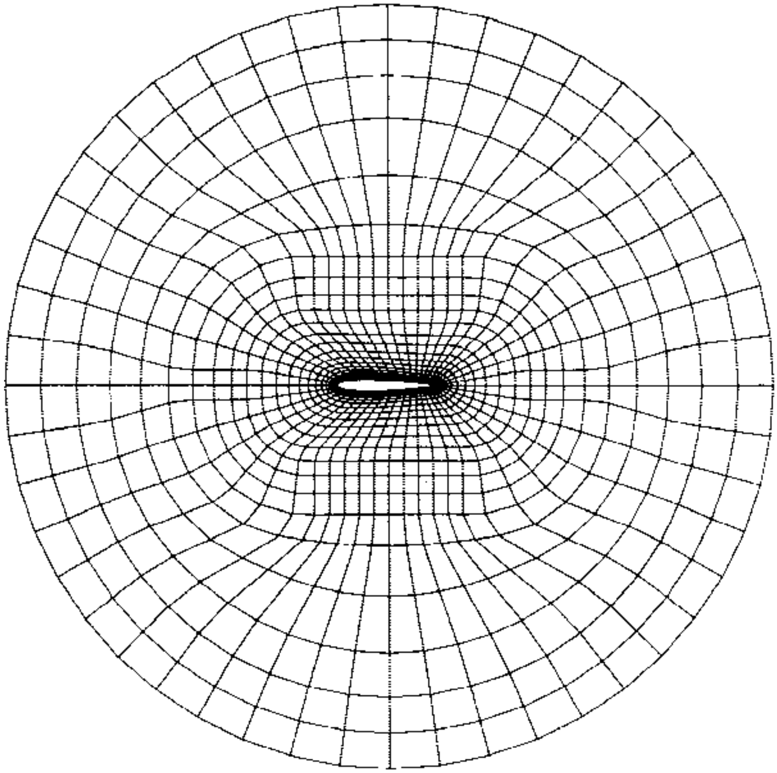


Figure 2 A grid to simulate the effect of wind-tunnel walls upon the flow over an airfoil.

walls. The separate simulation with far-field boundary conditions then produces flow-field data at the same grid points as the simulation with walls included. In addition to the embedding, the grid was clustered near the airfoil on which orthogonality was enforced.

THE CONSTRUCTION The multisurface transformation is designed for coordinate generation from a boundary $\mathbf{A}(\xi)$ to a boundary $\mathbf{B}(\xi)$ by means of an independent variable η and a smooth vector field established by interpolation of only first derivatives. From the vantage point of the general position vector $\mathbf{P}(\xi,\eta)$, the interpolation may be interpreted as a deficient form of Hermite interpolation. For interpolation points $\eta_1 < \eta_2 < \cdots < \eta_{N-1}$, the vector field is given by

$$\frac{\partial \mathbf{P}}{\partial \eta}(\xi,\eta) = \sum_{k=1}^{N-1} \phi_k(\eta)\frac{\partial \mathbf{P}}{\partial \eta}(\xi,\eta_k), \tag{5}$$

where the functions ϕ_k vanish at all interpolation points except $\eta = \eta_k$ (where the value is unity). By integration from $\mathbf{A}(\xi)$, the position vector is given by

$$\mathbf{P}(\xi,\eta) = \mathbf{A}(\xi) + \sum_{k=1}^{N-1} G_k(\eta)\frac{\partial \mathbf{P}}{\partial \eta}(\xi,\eta_k),$$

where

$$G_k(\eta) = \int_{\eta_1}^{\eta} \phi_k(\zeta)\, d\zeta. \tag{6}$$

At the endpoint $\eta = \eta_{N-1}$, the final position should be located at $\mathbf{B}(\xi)$. Upon setting

$$\mathbf{V}_k(\xi) = G_k(\eta_{N-1})\frac{\partial \mathbf{P}}{\partial \eta}(\xi,\eta_k), \tag{7}$$

the endpoint condition becomes

$$\mathbf{B}(\xi) = \mathbf{A}(\xi) + \sum_{k=1}^{N-1} \mathbf{V}_k(\xi). \tag{8}$$

Starting with $\mathbf{P}_1(\xi) = \mathbf{A}(\xi)$, successive vector positions are determined by the partial sums

$$\mathbf{P}_i(\xi) = \mathbf{P}_1(\xi) + \sum_{k=1}^{i-1} \mathbf{V}_k(\xi), \tag{9}$$

which, in an inverse sense, yields

$$\mathbf{V}_k(\xi) = \mathbf{P}_{k+1}(\xi) - \mathbf{P}_k(\xi). \tag{10}$$

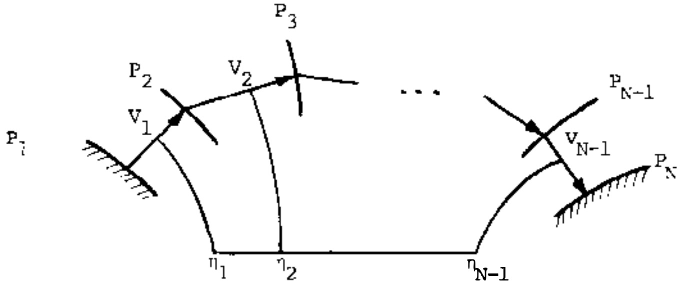


Figure 3 The construction of the multisurface transformation.

When the derivative in Equation (7) is expressed in terms of Equation (10) and is inserted into Equation (6), the position vector is given by

$$\mathbf{P}(\xi,\eta) = \mathbf{P}_1(\xi)+\sum_{k=1}^{N-1}\frac{G_k(\eta)}{G_k(\eta_{N-1})}[\mathbf{P}_{k+1}(\xi)-\mathbf{P}_k(\xi)], \qquad (11)$$

where $\mathbf{P}_1(\xi) = \mathbf{A}(\xi)$ and $\mathbf{P}_N(\xi) = \mathbf{B}(\xi)$. The equation represents a coordinate transformation that is entirely constructed from the curves or surfaces $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N$, each of which is expressed with respect to the same space of curvilinear variables ξ. For this reason, it is called the *multisurface transformation*. In correspondence with the interpolation points of Equation (5) and the vector directions of Equation (10), the constructive surfaces are assumed to be ordered in a monotone sequence that simply partitions the space between the boundaries $\mathbf{P}_1$ and $\mathbf{P}_N$. The intermediate surfaces $\mathbf{P}_2, \mathbf{P}_3, \ldots, \mathbf{P}_{N-1}$ and the choice of interpolants ϕ_k are used to control the grid so that various constraints can be satisfied as the grid is generated from boundary $\mathbf{P}_1$ to boundary $\mathbf{P}_N$. Because of the ratio of integrals G_k in the multisurface transformation, the normalization of the interpolants is not required. In Figure 3, a two-dimensional illustration is given.

INTERPOLANTS When there is only one interpolation, a constant function is the simplest choice. The result is the shearing transformation, which assumes the general form of Equation (3) when the unit interval $0 \leqslant \eta \leqslant 1$ is used. The cubic Hermite transformation of Equation (4) is obtained with two intermediate surfaces and global quadratic polynomial interpolants for the points $\eta_1 = 0, \eta_2 = 1/2$, and $\eta_3 = 1$. The exact Hermite format appears when the two intermediate control surfaces $\mathbf{P}_2$ and $\mathbf{P}_3$ are traded for the boundary derivatives. From the pure Hermite formulation, however, there is virtually no indication of any limitation on the size of the magnitudes. One is simply required to use a trial-and-error approach. By contrast, a natural geometric limitation comes from the assumption that there is some space between $\mathbf{P}_2$ and $\mathbf{P}_3$. This prevents the interpolation of vector directions in Equation (5) that would experience an abrupt backward flip. The corresponding coordinate curves then do not backtrack and thereby cause the grid to fold upon itself. In summary, a direct geometric interpretation of Hermite derivatives is provided in a manner that can be simply used to avoid the folding problem.

Among the various possible interpolation functions for the multisurface transformation, those that vanish off local regions can be used to establish local controls. Such interpolants have been constructed with piecewise polynomials. As a result of the integration for G_k, the level of derivative continuity is one greater than that of the piecewise polynomals. Relative to neighboring points of interpolation, the range of nonzero values for each interpolant is as small as possible subject to the constraints of derivative continuity, of curvature generality, and of uniformity admission. The details are given by Eiseman (1982a,b). An illustration of the general interpolants corresponding to the first two levels of derivative continuity is displayed in Figure 4. The interpolation functions at or near the endpoints appear as truncated versions. The piecewise linear case of Figure 4*a* yields first-derivative continuity and is generally applicable in two dimensions but not in three. For three dimensions, the second-derivative continuity and curvature generality from the piecewise quadratics of Figure 4*b* are required. The essential features of each interpolant, however, are the same. Both provide local control at a very precise level. The basic control stems from a dependency upon only the closest constructive surfaces. To illustrate this fact, we examine the simple piecewise linear case in graphical form without the supporting mathematical theory. In Figure 5, a curve segment corresponding to the interval $\eta_k \leqslant \eta \leqslant \eta_{k+1}$ is displayed at a fixed value of ξ. The segment is determined by the three successive surfaces indicated by dots at the endpoints of the line segments. The curve leaves one line at $\mathbf{P}(\xi, \eta_k)$ with matching tangent direction and enters the next at $\mathbf{P}(\xi, \eta_{k+1})$ in the same manner. A general point within the interval is noted by $\mathbf{P}(\xi, \eta)$. The

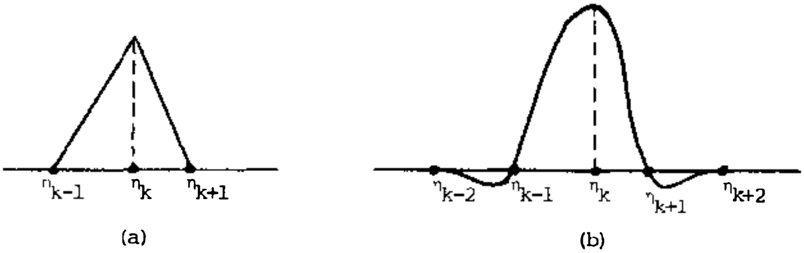


Figure 4 Local interpolation functions for the multisurface transformation. (*a*) Piecewise linear with continuity. (*b*) Piecewise quadratic with continuity up to first derivatives.

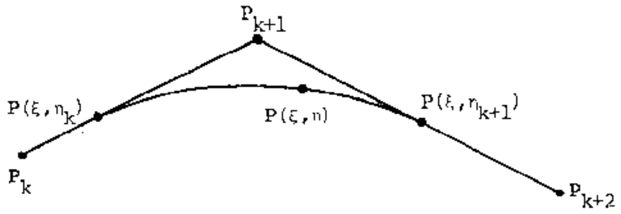


Figure 5 A curve segment determined by the piecewise linear interpolants.

curve segment is depicted without inflection points, and this is supported by the theory. With the local piecewise quadratics, the same pattern is continued by adding surfaces to each end of the construction to produce a dependency upon five consecutive surfaces. The corresponding segment naturally assumes a more arbitrary form. At or near the boundaries, each form is slightly modified to reflect the truncation of the interpolation functions.

With the freedom to choose intermediate control surfaces, the local dependency in η can be used to locally manipulate the grid without altering it elsewhere. Because of the algebraic formalism, the manipulation is done precisely and represents a high level of local control. An example with the local piecewise linear interpolants was given in Figure 2, where local Cartesian grids were smoothly embedded into a polarlike global grid about an airfoil so that the effect of wind-tunnel walls could be examined. Another useful embedding is the smooth and precise inclusion of boundary-layer coordinates within a global system. In addition to the many possible specifications for such local coordinate forms, a further application is to produce smoothness. When the boundaries have slope discontinuities, the classical shearing and Hermite transformations will propagate those discontinuities across the grid unless sufficient clustering is applied along the boundaries at the requisite locations. Without clustering, the local controls can be used to limit the propagation, even to the extent that it is gone before the first inward grid point. In effect, the use of smooth nearby surfaces allows the grid to forget the boundary at a close distance.

UNIFORMITY In the application of the controls, a major degree of precision is achieved with the capability to establish uniform distributions of points, curves, or surfaces in either a local or a global sense. Once established, any distribution function can be composed with the transformation to achieve a desired effect without distortion. An obvious example is the distribution of points along a curve by using its arc length as a measure of uniform conditions. Uniform conditions for families of coordinate curves or surfaces, by contrast, are much more complex. While the direct and simple approach would be to take the arc length of each curve in the η variable, the result could still be a nonuniformly distributed family. This will happen if the curves deviate sufficiently from straight lines. No deviation occurs with the shearing transformation, and with this transformation the desired uniformity appears directly as a result of linearity in η along the line segments. Along the more arbitrarily constructed curves, the same uniformity can also be directly achieved by a projection onto line segments or, more generally, onto prescribed vector directions. With a given vector $\boldsymbol{\tau}(\xi)$ to define the direction of measurement, the projected curve assumes the length

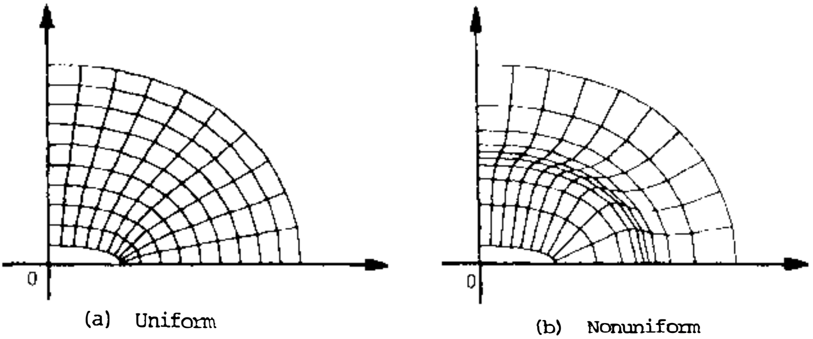
$$s_p(\xi,\eta) = [\mathbf{P}(\xi,\eta) - \mathbf{P}_1(\xi)] \cdot \boldsymbol{\tau}(\xi) \tag{12}$$

starting from $\mathbf{P}_1(\xi)$. A uniform family of curves is then obtained when the length s_p along $\boldsymbol{\tau}$ is linear in η. In a graphic sense, the distance measurements can be viewed as if a sequence of yardsticks were laid across the region, with one for each fixed ξ. If we use the intuitive notion of yardsticks, the distinction between uniform and nonuniform can be visually detected in an almost analytic way. A global example is displayed in Figure 6.

To admit uniformity, the choice of interpolants for the multisurface transformation is restricted to satisfy the relationship

$$\sum_{k=1}^{N-1} \frac{\phi_k(\eta)}{\phi_k(\eta_k)} = 1. \tag{13}$$

Once satisfied, the chosen interpolants lead to conditions upon the placement of intermediate surfaces in order to achieve uniformity either globally or locally. The global polynomials and the local piecewise-linear cases generally satisfy the condition. The local piecewise quadratics have to be constructed to satisfy it. The respective development is given by Eiseman



(a) Uniform

(b) Nonuniform

Figure 6 Grids about an ellipse.

(1979, 1982a,b). On application, the desired uniform distribution of curves or surfaces is prescribed in an a priori sense by precise, simple algebraic formulas. The resulting transformation then has the uniformity analytically built into its formulation. This formalism is particularly valuable in the construction of local coordinate forms.

Boolean Operations

The multisurface transformation represents the synthesis and extension of classical algebraic techniques for coordinate construction in one direction. The capability to combine directions comes with Boolean operations on projectors that correspond to transformations. Although the highest level of control is available with the projector for the general multisurface transformation, the essential properties of combining directions is more readily displayed with the projector for the classical shearing transformation.

To start, we first consider the collection (space) of all transformations $\mathbf{F}(\xi, \eta)$ over the domain of ξ and η variables. When the two boundaries $\mathbf{P}_1(\xi)$ and $\mathbf{P}_2(\xi)$ are given in correspondence with $\eta = 0$ and $\eta = 1$, the collection contains the shearing transformation of Equation (3) and many other transformations that match the two boundaries. The selection of shearing is the same as a projection Q_η from the entire collection onto the shearing element. Analytically, the shearing projector Q_η is defined by

$$Q_\eta[\mathbf{F}] = (1-\eta)\mathbf{F}(\xi, 0) + \eta\mathbf{F}(\xi, 1), \tag{14}$$

where the form of Equation (3) is retrieved upon setting $\mathbf{P}(\xi,\eta) = Q_\eta[\mathbf{F}]$, $\mathbf{P}_1(\xi) = \mathbf{F}(\xi, 0)$, and $\mathbf{P}_2(\xi) = \mathbf{F}(\xi, 1)$. In a formal algebraic context, a projector is an operator whose double application produces the same result as its original application. The shearing operation is readily seen to satisfy this condition, which is symbolically stated as $Q_\eta^2 = Q_\eta$.

At the boundaries of the ξ variables, the shearing projector yields a transformation where the corresponding image boundaries are generated by straight line segments. Such lateral boundaries, however, are often too specialized. They simply arise from a construction in one direction that does not contain any data about lateral variations. To include the requisite data, other directions must be considered in a similar format. In two dimensions with $0 \leqslant \xi \leqslant 1$, a shearing projector for the only other direction is given by

$$Q_\xi[\mathbf{F}] = (1-\xi)\mathbf{F}(0, \eta) + \xi\mathbf{F}(1, \eta), \tag{15}$$

where the specifications for $\mathbf{F}(0, \eta)$ and $\mathbf{F}(1, \eta)$ define an arbitrary pair of lateral boundaries for Q_η. While the simple sum of projectors for the ξ and η directions would include data for all boundaries, the corners would be doubly counted. To correct this situation, a pure corner interpolation must be subtracted from the sum. Such an interpolation comes from the successive application of the two projectors. The result is called the "*tensor product*" *interpolant* and is given by

$$Q_\xi Q_\eta[\mathbf{F}] = (1-\xi)(1-\eta)\mathbf{F}(0,0)+\xi(1-\eta)\mathbf{F}(1,0) \\ +(1-\xi)\eta\mathbf{F}(0,1)+\xi\eta\mathbf{F}(1,1). \tag{16}$$

The order of application is unimportant, since the commutative property $Q_\xi Q_\eta = Q_\eta Q_\xi$ is clearly satisfied. The sum with the corner adjustments is called the "*Boolean sum*," is denoted by $\oplus$, and is defined by

$$Q_\xi \oplus Q_\eta = Q_\xi + Q_\eta - Q_\xi Q_\eta. \tag{17}$$

Using the known commutativity of the product, it is an easy matter to check that the Boolean sum is a projector in a formal algebraic sense. The projected result $Q_\xi \oplus Q_\eta[\mathbf{F}]$ is also readily seen to match all four boundaries. With different projectors, the multidirectional matching of various properties follows when the commutative property holds. In the bidirectional cubic Hermite case, it follows from derivative commutativity at the corners. Within the general multisurface context, it comes from a well-defined control net of constructive surfaces.

Upon application of multidirectional matching, the interpolation can be executed in one direction at a time. With the Boolean sum of Equation (17) in the form $Q_\xi + Q_\eta(1-Q_\xi)$, the two-dimensional process is split into the steps

$$\mathbf{F}_1 = Q_\xi[\mathbf{F}]$$

and (18)

$$\mathbf{F}_2 = \mathbf{F}_1 + Q_\eta[\mathbf{F}-\mathbf{F}_1],$$

where $\mathbf{F}_2(\xi,\eta)$ is the position vector defining the transformation. With (ξ,η,ζ) coordinates for three dimensions, corresponding Q_ξ, Q_η, Q_ζ must be used in some fashion. When all bounding surfaces are specified, the transformation is given by the Boolean sum $Q_\xi \oplus Q_\eta \oplus Q_\zeta[\mathbf{F}]$, which, as before, is similarly split into steps as

$$\mathbf{F}_1 = Q_\zeta[\mathbf{F}], \\ \mathbf{F}_2 = \mathbf{F}_1 + Q_\eta[\mathbf{F}-\mathbf{F}_1], \tag{19} \\ \mathbf{F}_3 = \mathbf{F}_2 + Q_\xi[\mathbf{F}-\mathbf{F}_2].$$

When only edges are specified, Boolean products and sums are used to define the transformation as $(Q_\xi Q_\eta \oplus Q_\xi Q_\zeta \oplus Q_\eta Q_\zeta)\,[\mathbf{F}]$. When only corners are specified, the transformation assumes the pure product form $Q_\xi Q_\eta Q_\zeta[\mathbf{F}]$, which (with the shearing projector) yields standard trilinear interpolation. In continuation, the pattern of specifications can be done in a mixed sense. Interpreting products as intersections and sums as unions, transformations can be readily constructed for specifications of various assortments of bounding surfaces, edges, or corners.

With the interpolation of continuum quantities such as full curves and surfaces, the algebraic methods using Boolean sums have been called transfinite to reflect the generally infinite number of interpolation points. The theory of transfinite interpolation was developed as an outgrowth of the earlier blending-function techniques of Coons (1964) for the mathematical representation of surfaces required in computer-aided geometric design, as in Barnhill & Riesenfeld (1974). The basic development with Boolean operations was done by Gordon (1971). The use in coordinate generation is described by Gordon & Hall (1973) and Gordon & Theil (1982). The stepwise application is explicitly presented by Smith (1982) for the Hermite case of arbitrary degree.

Applications

Algebraic grid-generation techniques have been developed to meet a variety of problem constraints in a precise manner. The principal constructive elements are (*a*) the multisurface transformation for its control in one direction and (*b*) the Boolean operations for extending the control to all directions. Typical previously mentioned applications arose from constraints for geometry, resolution, smoothness, local embedding, and grid patching. Further applications arise in conjunction with other techniques. Through an algebraically defined system of coordinates, orthogonal trajectories can be computed with enhanced accuracy from the algebraic definition. The various orthogonal trajectory techniques for the production of orthogonal grids are reviewed by Eiseman (1982c). Nearly conformal transformations have been obtained by using a sequence of simple algebraic conformal mappings followed by a shearing. The conformal maps bring the geometry nearly onto a line or a circle. The shearing makes the match exact, but it destroys precise conformality. Such cases were examined by Caughey (1978). If we use the local controls within the multisurface transformation, the exact match can be accomplished with a rapid blending into either Cartesian or polar coordinates. The final transformation is then nearly conformal adjacent to the boundaries and exactly conformal elsewhere.

The automation required to systematically generate algebraic grids for a wide variety of applications was established in two dimensions with the software system developed by Eiseman (1982d). The eventual manipulation of the system with interactive graphics will streamline the process for many applications. By contrast, such automation is not available in three dimensions, where grid generation tends to be done on a case-by-case basis.

PARTIAL DIFFERENTIAL EQUATIONS METHODS

The techniques for the generation of arbitrary curvilinear coordinates come from either explicit or implicit definitions. Correspondingly, the coordinates are usually given either algebraically or as the solution to differential equations. In the latter category, a numerical solution procedure is required to find the grid-point locations from the defining system and the associated boundary conditions that must represent the physical geometry.

Hyperbolic Methods

When only one physical boundary is specified, hyperbolic partial differential equations may be used to obtain a grid by spatially marching from the given boundary. The remaining boundaries are determined by the solution and are geometrically unimportant in cases such as the external flow about a single object. With the requirement to prescribe only the object and the spacing along it, the hyperbolic methods proceed in marching steps that give the spacing in the outward direction. The inherent efficiency of the methods is due to the use of a single sweep through physical space. In two dimensions, orthogonal grids are obtained and are smooth provided that the boundary is free from slope discontinuities. A fundamental development was given by Stadius (1977), and one which was well suited to body concavity was presented by Steger & Sorenson (1980). The hyperbolic methods, however, are restricted in the amount of control that can be applied.

Elliptic Methods and Concavity Controls

The element of control over the grid has been more fully developed with the methods based upon elliptic partial differential equations. These we shall refer to as *elliptic methods*. As in the previous algebraic development, conformal mapping can be done in two dimensions with specifications of only boundary geometry. Typically, a pair of Laplace equations is solved subject to Cauchy-Riemann boundary conditions. The basic departure from conformal mapping comes with the specification of pointwise distributions along the boundaries. From this departure, some resolution requirements for the flow can be inserted. In comparison, the boundary distributions conflict with conformal conditions and produce nonorthogonal coordinates.

The earliest successful development was formally reported by Winslow (1967), who started with a Laplace system that determines a conformal mapping from the physical region into the space of curvilinear variables. Since the grid had to be generated in the physical space, the system had to be reformulated by an interchange of dependent and independent variables. With the reformulation, the physical boundaries are then represented by pointwise (Dirichlet) boundary conditions on the rectangular region of curvilinear variables. When the basic Laplace system determines a nonsingular mapping, the inverse mapping is biased toward conformal conditions. This tends to impose a global smoothness. The conflicting Dirichlet data produce nonorthogonality, which is most intense at the boundaries and gradually decays toward conformality as the interior is approached. From the discrete viewpoint of grid mappings, conformality is observed when small Cartesian squares are mapped approximately into squares of varying size.

The next major step in the development of elliptic methods was given by Thompson et al. (1974), who added periodic boundary conditions to produce branch cuts for various topological configurations and who also suggested that control over the grid could be accomplished by altering the original Laplace system. The alteration is to consider a pair of Poisson equations by including specifications for the right-hand sides. These are called *forcing terms* and are general functions of the curvilinear variables. The particular form to be used was established later by Thompson et al. (1977). Without forcing terms, Mastin & Thompson (1978) were able to show that the two-dimensional system analytically defined a nonsingular transformation. In higher dimensions or with forcing functions, nonsingularity is not generally assured. However, in two dimensions, the implication is that the basic system will reliably produce nonoverlapping grids, and that departures therefrom can be balanced against a solidly established reliability factor.

ONE DIMENSION To examine the basic Poisson formulation in the simplest possible manner, our discussion starts in one dimension and then extends to higher dimensions. For a physical-space location x and a curvilinear variable ξ, the one-dimensional Laplacian is just the second derivative ξ_{xx}. The Laplace equation is given by its vanishing. An equally spaced grid in ξ then yields a corresponding equally spaced grid in x, since the analytic solution produces a linear relationship. To deviate from linearity and the associated uniform grid, the second derivative is forced to have prescribed positive and negative values. This gives the function $\xi(x)$ respective upward and downward concavity in amounts that increase with the size of the force. The result is a Poisson formulation. The force as a control over concavity is employed for the purpose of clustering points in the physical space. Clustering is done at fixed locations in the curvilinear space, with

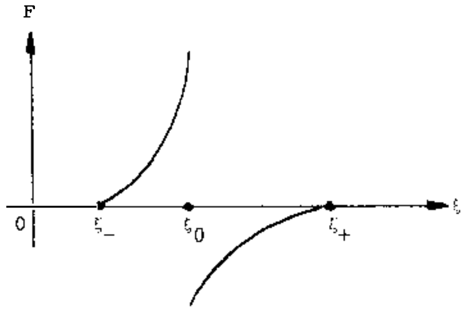


Figure 7 A one-dimensional forcing function for one cluster.

corresponding forcing terms each given as functions of ξ. The physical clustering locations x are then determined by the solution.

To illustrate the control, we consider a single cluster that starts at ξ_-, is centered at ξ_0, and ends at ξ_+. The associated forcing function $F(\xi)$ for the Poisson equation

$$\xi_{xx} = F(\xi) \tag{20}$$

is depicted in Figure 7. The solution is qualitatively displayed in Figure 8 and under the assumption of nonsingularity is a strictly monotone function. Upon approaching ξ_- from below, the solution is linear, since F is zero. At some physical point x_- corresponding to ξ_-, the solution starts to rise in the concave-upward direction with increasing intensity until ξ_0 is reached

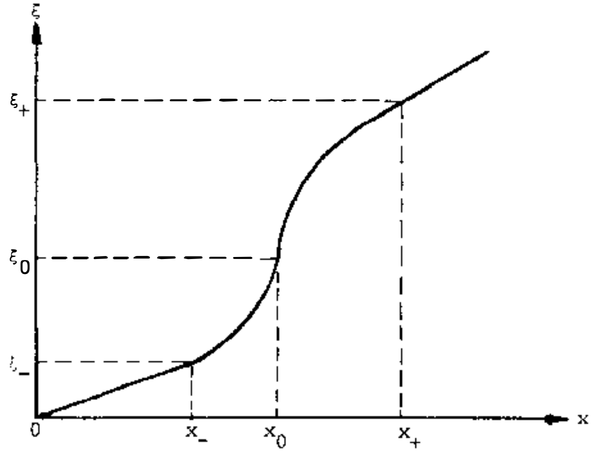


Figure 8 A one-dimensional Poisson solution for one cluster.

for some x_0. The forcing function F then changes sign to flip concavity to the downward direction. This decreases in intensity upon approaching x_+, which corresponds to the point ξ_+ where F reaches zero. Above ξ_+ and beyond x_+, the vanishing F causes the solution to return to linearity. With the uniform grid for ξ, the physical-space grid is most tightly clustered at the location x_0, where the solution has its greatest slope. On either side the slope steadily decreases toward constant conditions with increasing distance from x_0. The constant slopes on each side may be different, and if they are, the associated grid sections will be uniform but will have different spacing.

The requirements for equal spacing on each side are evident when the Poisson equation is transformed by using the chain rule to interchange the dependent and independent variables. The transformed equation, which is also needed for the numerical solution, is given by

$$x_{\xi\xi} = -x_\xi^3 F(\xi). \tag{21}$$

A direct integration, however, yields

$$\frac{1}{x_\xi^2} = c+2\int F(\xi)\, d\xi \tag{22}$$

for some constant c. A general plot of the right-hand side is presented in Figure 9. The constant of integration c gives the constant slope for the uniform grid before ξ_-. The uniform grid following ξ_+ will have different spacing if the second constant value is distinct from c. In the figure, a smaller value is depicted. This would correspond to a larger spacing Δx, since for constant $\Delta\xi$, the plotted values give $(\Delta\xi/\Delta x)^2$. To match the spacing, the values on each side must be the same, and the forcing function must then integrate to zero over the range from ξ_- to ξ_+. If the function is

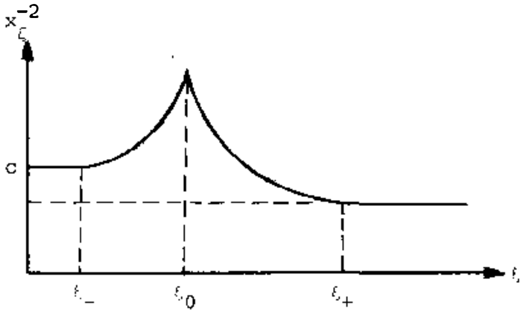


Figure 9 The first Poisson integral for one dimension.

antisymmetric about ξ_0, then the zero integral is assured, but ξ_- and ξ_+ must be at equal distances from ξ_0.

The inclusion of a cluster point, while preserving the original uniform grid elsewhere, is accomplished with a forcing function defined separately for the four successive segments determined by ξ_-, ξ_0, and ξ_+. Upon relaxing the requirement for exact preservation, the function can be defined to asymptotically decay on the two segments adjoining ξ_0. The form is very close to the previous case displayed in Figure 7. The ξ_- and ξ_+ can now be interpreted as the points where the decay has just become virtually complete. With the assumption of antisymmetry, the choice of simple exponential segments leads to the Poisson equation

$$\xi_{xx} = -\operatorname{sgn}(\xi-\xi_0)ae^{-b|\xi-\xi_0|}, \tag{23}$$

which was suggested in two dimensions by Thompson et al. (1977). The sign function sgn is $+1, 0$, and -1 corresponding to arguments that are positive, zero, and negative, respectively. The intensity of the cluster is controlled by the coefficient a; the size, by the rate of decay b. When periodic boundary conditions are imposed for the equation, the decay must be carried periodically through the grid a requisite number of times. This can be shortened by a sufficiently rapid decay. When multiple clusters are desired, the same considerations apply, and the right-hand side of Equation (23) is extended with a cluster summation index i for a, b, and ξ_0.

TWO DIMENSIONS The further extension to two dimensions assumes the same format, although the curvilinear locations for clustering are now both points and lines rather than just points. For curvilinear variables (ξ, η), the Poisson system assumes the form

$$\begin{aligned} \xi_{xx}+\xi_{yy} &= P(\xi,\eta), \\ \eta_{xx}+\eta_{yy} &= Q(\xi,\eta). \end{aligned} \tag{24}$$

Clustering is accomplished relative to the uniform conditions defined by the pure Laplace system where the right-hand sides vanish. The forcing function P for the first equation controls the curves in physical space that correspond to constant values of ξ. As in the one-dimensional case from Equation (23), the function $P(\xi,\eta)$ is conveniently taken as a sum of exponential-type terms. To uniformly cluster curves of constant ξ about the curve for ξ_0, a term of the form

$$a \operatorname{sgn}(\xi-\xi_0) \exp(-c|\xi-\xi_0|) \tag{25}$$

is included. To locally cluster curves of constant ξ toward the point with parameter value η_0 on a curve of fixed ξ_0, a term of the form

$$b \operatorname{sgn}(\xi-\xi_0) \exp(-d[(\xi-\xi_0)^2+(\eta-\eta_0)^2]^{1/2}) \tag{26}$$

is also included. As earlier, a and b give the clustering intensity, while c and d give the decay rates to establish the range of clustering. Each controlling coefficient a, b, c, and d is positive and upon inclusion in $P(\xi,\eta)$ would be appropriately subscripted along with the desired point locations (ξ_0,η_0) and line locations ξ_0. Negative values for a and b would produce the opposite effect of repulsion. The forcing function $Q(\xi,\eta)$ for the curves with constant η is similarly structured by interchanging the above roles for ξ and η in our discussion of $P(\xi,\eta)$. The analytic solution to the generating system [Equation (24)] has derivative continuity to all orders except along the lines where clustering is imposed. There, continuity is obtained only up to first derivatives because of the flipping of concavity from the sign functions in Equations (25) and (26).

At the boundaries, only one direction of concavity is needed, and it is determined by a simple truncation of the general form upon translation to either side. When the pointwise distribution on a boundary is fixed as a result of Dirichlet conditions, the effect of forcing functions is to change intersection angles and spacings from the adjacent curve. For a boundary with constant ξ, the forcing function P changes the spacing, while Q alters the angles of the intersecting constant-η curves. The actual specification of the angles and spacings requires an iterative determination of P and Q. This was done by Steger & Sorenson (1979).

Rather than adjustments purely at the boundary, the spacing along the boundary may also be desired within the interior of the region. To propagate boundary distributions toward the interior, there must be a suitable definition of the forcing functions. As a simple illustrative example, consider the Cartesian coordinates on a rectangle with an x distribution $\xi = x^3$ and a uniform y distribution $\eta = y$. Upon substitution into the Poisson system of Equation (24), the forcing functions for these coordinates are found to be $P = 6\xi^{1/3}$ and $Q = 0$. Along with the distributions prescribed only on the boundaries, the Poisson system has been constructed from its solution. If the same boundary data were given to the Laplace system obtained by setting $P = 0$, then the x distribution would not be propagated across the rectangle. Instead, points would spread out in adjusting themselves to approximate conformal conditions in the interior.

In the general setting, the propagation of distributions from the boundaries is accomplished by an interpolation of forcing functions from the boundaries. The boundary forces are obtained from the Poisson system by using the pointwise distribution and the assumption of boundary orthogonality. The orthogonality assumption replaces the transverse derivative data in the system with the boundary curvature distribution. From the boundary forces, the interpolation is done between opposing faces in the (ξ,η) space. Once the Poisson system has been established with the interpolated forcing functions, the boundary orthogonality assumption is discarded. The solution then proceeds directly from the usual boundary data. The propagation of distributions by forcing-function interpolation is primarily due to Middlecoff & Thomas (1979) and Warsi & Thompson (1976).

In the propagation technique and in the motivating Cartesian example, the Poisson system is used to determine the forcing functions. With a view toward modifying existing coordinate systems, the same determination can be more arbitrarily applied to establish corresponding Poisson systems. Their modification then results in a corresponding modification of the coordinates. Clustering would then be done relative to the determined system.

With the forcing functions depending only upon the curvilinear variables, the application of clustering is relative to only curvilinear location. The internal points or curves about which clusters are desired cannot be given specified locations in the physical region. Rather, each cluster affects the others, and altogether the Poisson solution determines their respective locations. This can be an advantage, however, since possibly unimportant locations would not have to be specified and would thereby relieve the user of the effort. This determination can also be a disadvantage if the location is important. A potential remedy is suggested by Thompson (1982b), who considers a forcing-function dependence on physical rather than curvilinear variables.

To numerically solve the Poisson system with any of the above forcing functions, the dependent and independent variables must be interchanged by using the matrix inverse to the chain rule. The transformed Poisson system is given by

$$\begin{aligned} g_{22}x_{\xi\xi}-2g_{12}x_{\xi\eta}+g_{11}x_{\eta\eta} &= -g[x_\xi P-x_\eta Q], \\ g_{22}y_{\xi\xi}-2g_{12}y_{\xi\eta}+g_{11}y_{\eta\eta} &= -g[y_\xi P-y_\eta Q], \end{aligned} \tag{27}$$

where

$$g_{11}=x_\xi^2+y_\xi^2, \qquad g_{12}=x_\xi x_\eta+y_\xi y_\eta, \qquad g_{22}=x_\eta^2+y_\eta^2$$

are metric coefficients and $g=\det(g_{ij})$ is the square of the Jacobian $x_\xi y_\eta - x_\eta y_\xi$. Geometrically, the coefficients come from the differential expansion of squared arc length along arbitrary curves, with measurement relative to the given coordinate system. On the rectilinear boundary for the curvilinear variables, the boundary data for various bodies are now explicitly inserted as Dirichlet boundary conditions. Branch cuts are directly given by periodic boundary conditions and, as in the case of clustering, have locations that are determined by the solution and that cannot be specified in advance.

The same Poisson format, the clustering, and the interchange of variables can be given in three dimensions. The format is simple: The clustering is for points, curves, and surfaces, and the interchange is more detailed. The entire process is much more complicated, is done on a case-by-case basis, and depends upon grid generation for two-dimensional surfaces.

ADAPTIVE GRID MOVEMENT

The numerical simulation of fluid-mechanical phenomena often involves solutions with rapid variations over short distances and at unpredictable locations. Moreover, a solution may even be multivalued. When the grid is predetermined by the physical region and the resolution requirements at expected locations, the accurate simulation can be jeopardized if the large solution variations should appear somewhere on a scale that is too small for the prescribed spacing of points. To model such phenomena, the grid must be adaptively moved into positions that most accurately represent the solution as it evolves.

The Consolidation of Adaptive Data—A Monitor Surface

The severe behavior in the solution is usually well monitored by a small number of salient physical quantities. Their use clearly obviates the need to create a monitoring quantity by means of a formal error estimate. Altogether, the salient quantities form a vector that can be evaluated at each point in physical space. Collectively, the evaluations determine a surface over the physical space of the same dimension as the space. The surface then simply consolidates the resolution requirements into a single object. When salient quantities have large variations at distinct or nonconflicting locations, their combined behavior can be adequately followed by using some monotone scalar function of the quantities. The effect of employing suitable scalar functions is to define a monitoring surface in a space of the lowest possible dimension. When all quantities are collapsed into one scalar function, the dimension is only one greater than the physical space. Once established, the evolving monitor surface becomes the object that must be accurately resolved by the appropriately adapting grid.

The methods for adaptive movement consider the surface either as a geometric entity or as a function from which derivative data are taken. With the function viewpoint, gradients are directly given by first derivatives, and curvature is roughly approximated by second derivatives. With the geometric viewpoint,·gradients are implicitly detected by surface arc-length measurements, and the actual curvatures are employed directly. Moreover, the treatment of multivalued solutions can be done in a direct surface-oriented fashion.

The comparison between the two viewpoints is most readily seen when the surface is defined by a single scalar quantity u. In one dimension, with location x, the assumption of constant arc-length increments means that $(1+u_x^2)^{1/2}dx$ is constant. Large gradients u_x increase the first factor and cause dx to shrink to maintain constancy. When the increase is substantial, the first factor is essentially the magnitude of the gradient itself. In the discrete form, the one-dimensional surface grid is uniform, and the projection down into physical space produces a corresponding grid that resolves gradients. If the surface changes direction on a scale that is too small for the uniform arc-length grid, then points must be moved to provide adequate resolution. The precise measure of direction changes is the curvature. In terms of the physical-space location x, it is given by $k = u_{xx}(1+u_x^2)^{-3/2}$. From the surface viewed as a function, it is usually approximated by u_{xx}, which is accurate only when u_x is much less than unity in magnitude. In an application where u_x varies rapidly from zero to a large value, the estimate u_{xx} of k correspondingly varies from a highly accurate one to one that is much too large because of a missing large denominator. The consequence is that the desired resolution is pulled off-center toward the steep gradient region.

In higher dimensions, the use of arc length and curvature is required for the same purposes. The application of each, however, is less well defined, since there is no unique measure of a uniform distribution of surface points, nor is there a universal choice for curvature. The complicating feature comes from the combination of distinct directions. The measure of uniformity can be established in terms of equal cell volumes, equal arc lengths along curves, or some other desirable mapping property such as conformality. Directional properties are witnessed by noting that cells with fixed volumes can become elongated, while cells with fixed edge lengths can be collapsed. In contrast, conformality produces well-structured cells that can rapidly dilate in size. On balance, these uniformity measures are reasonable for a wide variety of situations, albeit each has distinct strengths and weaknesses. 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 to separately describe curve and surface bending rates. The geodesic part gives rates only for the curve, while the normal part gives rates only for the surface. For adaptive movement, complications from the physical boundaries are treated with geodesic curvature, while complications from the solution are treated with normal curvature.

The basic surface geometry is given by the normal curvature, which is formally a measure of the rate at which surface-tangent planes vary as the curve is traversed. In two dimensions, both Gaussian and mean curvatures are defined in terms of the two principal normal curvatures, which are the maximum and minimum values, respectively, when all angular directions are considered. The Gaussian and mean curvatures are the product and the average, respectively. When the surface has a fold corresponding to a disturbance such as a straight shock front, the normal curvature has a minimum of zero along the front and some large maximum in the perpendicular direction. The Gaussian curvature vanishes as a result of the factor of zero. Consequently, it cannot be used for clustering to the front. Similarly, mean curvature vanishes on minimal surfaces and cannot be used for clustering on such surfaces (e.g. Scherk's minimal surface: $e^z \cos x = \cos y$). A more reasonable and easily applied approach is to use normal curvatures directly.

When the surface is viewed as a function and second derivatives are taken, a situation similar to the previously discussed one-dimensional case arises. As in one dimension, large gradients tend to shift clustering regions off-center. In addition, the sole use of derivatives along coordinate curves in physical space fails to detect the full bends in the surface. The neglected parts correspond to the sideways tilting motion of the tangent planes as the curve is traversed.

For a space of any dimension, the same function and geometric views of the surface can be taken and compared. To accurately represent the surface, the geometric view is required and is consistent with local Taylor series estimates on the surface. The function viewpoint approximates the geometric quantities in varying degrees of quality and is consistent with local Taylor series estimates from physical space. To continue in more detail, basic differential geometry is needed and is available in Laugwitz (1965). The application to adaptive movement is discussed further in Eiseman (1983a).

Underlying each viewpoint is the presumption that the surface is sufficiently smooth to yield reasonable approximations to the various derivatives and geometric quantities. With the representation in the discrete form of a grid, a numerical filter can usually be applied to remove spurious oscillations and to round off sharp discontinuities. When a filter is needed and the solution must be exactly preserved, the surface must be stored separately and then smoothed. In addition, the temporal evolution of the surface may also require a smoothing operation to prevent wiggles in time. The corresponding filter then requires surface data over a time period up to at most the new solution level.

Movement Strategies

With the salient solution properties consolidated into a smooth monitor surface, the creation of accurate grid representations produces general movement strategies that are not tied to a particular numerical solution procedure or physical problem. The generality is needed to address a wide variety of simulations in a nearly optimal way. Further optimization would be expected when special techniques are developed for special situations. The moving finite-element method described by Miller & Miller (1981) and the modified-equation technique proposed by Klopfer & McRae (1981) are both optimal in their respective measures, and they are restricted to distinct types of solution procedures. For a class of convection-diffusion problems, the transformation into simpler diffusion equations was simultaneously considered by Piva et al. (1982) and Ghia et al. (1983). Under this physical restriction, the movement is determined by the Poisson system of Equation (27) with velocity-dependent forces. While further such studies are available for specific situations, the discussion here is for general strategies that not only are simpler to implement in many situations but also address the more complicated simulations that typically arise in fluid mechanics.

The general adaptive grid simulations directly use the monitor surface to feed the solution properties back into the movement part of the processes. The primary objective of the feedback process is to maximize accuracy for the given number of points at the lowest possible cost for movement should that be a significant part of the simulation. Attempts to mathematically formulate the objective were made by Rheinboldt (1983) and Babuška et al. (1983b).

PROPORTIONALITY With the stated objective, the grid motion is established directly in the physical region or on the monitor surface by using weight functions. The typical application is to require that the product of the weight and some measure of cell size be constant or nearly so. An increase in the weight then causes the cell sizes to shrink and to produce a desired clustering of points. Balanced against the arbitrary application of clustering is the requirement that the grid be well structured, in the sense that deviations from orthogonality are moderate and that global smoothness, or equivalently uniformity, appears when clustering is absent or has moved elsewhere. When smoothness is explicitly included, the simplest and most useful form of the weight is given by the linear combination

$$w = 1 + \sum_{k=1}^{m} c_k M_k, \tag{28}$$

where the M_k are the magnitudes of attracting quantities and the corresponding c_k are the coefficients that determine the respective levels of significance. The smoothness is provided by the choice of unity for the first term, which replicates the measure of cell size under the intended multiplication.

In one dimension or for curves in higher dimensions, the cell size is the differential element of arc length ds that is taken either in the physical region or on the overlying monitor surface. The transformation to a curvilinear variable ξ is defined when the product $w\,ds$ is proportional to $d\xi$. The grid motion comes from the assumed constant value of $d\xi$, which forces ds to vary inversely with w. The result is called an equidistribution of the weight. When w is a function of s, the resolution is prescribed at fixed physical or surface locations. If we assume that ξ is in the unit interval, a direct integration yields the transformation

$$\xi = \frac{F(s)}{F(s_{\max})}, \tag{29}$$

where

$$F(s) = s + \sum_{k=1}^{m} c_k \int_0^s M_k(z)\, dz. \tag{30}$$

When all the coefficients c_k are set to 0, the transformation is linear and a uniform grid results. With the normalization by $F(s_{\max})$, the proportionality constant is explicitly given. Had differentiation been performed instead, the constant would then have been implicitly determined from the boundary conditions to the resulting differential equation $w\xi_{ss} - w_s\xi_s = 0$.

With the linearity of the weight carried over into $F(s)$, the coefficients can be given a precise interpretation by using the evaluation at $s_{\max}$. Dwyer (1983) observed that the total amount of a quantity M_k was represented by its total integral I_k, and that the ratio $r_k = c_k I_k / F(s_{\max})$ would then be the fractional amount. The prescription of fractions is more natural, since precise numbers of points can be assigned for each quantity once the total number of points is given. The conversion of the prescribed fractions r_k into weighting coefficients $c_k = (r_k/I_k)F(s_{\max})$ is done once $F(s_{\max})$ is determined. The determination comes from Equation (30) with an evaluation at $s_{\max}$ and a substitution for each c_k. Upon solution, we obtain $F(s_{\max}) = s_{\max}/(1 - r_1 - \cdots - r_m)$. Underlying the expression for c_k is the assumption that the total amount of each quantity I_k is not zero. Otherwise, there would be a division by 0. In practice, such a division is avoided by adding a very small positive quantity to each I_k.

While the integrals I_k represent the total amount of a given quantity M_k, it should be noted that the contribution to clustering must occur somewhere. At the very least, the associated locations are spread over some of the arc length. Moreover, contributions from distinct quantities may also occur at some of the same places. When the places for clustering are localized and are each the result of a single quantity, the total fraction of the points for the places can be precisely specified. This is accomplished by adding the appropriate local arc length to each integral, by removing that total added length from $s_{\max}$ to get an adjusted maximal arc length, and then by proceeding as before. When there are intervals where clustering comes from more than one quantity, the same type of adjustments are made by simply accumulating all contributions for the interval and removing the appropriate amounts from the previous totals. Such adjustments are only possible when the quantities vanish or almost vanish everywhere except on localized intervals.

A similar development leading up to such functions occurs when the weight is a function of only ξ. This means that clustering is done relative to curvilinear, rather than physical, locations and is thus free to move as in the elliptic method. The weight then appears in reciprocal form, and the roles of ξ and s are formally reversed. The consequent transformation gives s as a function of ξ. Since equal spacing is prescribed for ξ, the desired grid in s is evaluated without interpolation, in contrast to the transformation of Equation (29).

Although further generality would come with weights defined by arbitrary monotonic functions of the magnitudes M_k, the adaptive methods derived from proportionality statements typically use the linear form and thereby inherit the interpretation of coefficients in terms of fractions. White (1979, 1982) takes the monitor surface as the actual solution, chooses s as its arc length, and effectively has $w = 1$. The simulations are also performed with respect to s rather than the physical distance x. This permitted White (1982) to smoothly simulate multivalued behavior in the nonlinear wave equation $u_t + uu_x = 0$. Relative to the arc length of the actual solution, Ablow & Schecter (1978) consider curvature clustering with the curvature magnitude M_1 in Equation 28 for $m = 1$. For arbitrary curves bounding two-dimensional regions, Eiseman (1979) employs the same transformation in the equivalent geometric format of approximate circular normal images. In two dimensions, Dwyer et al. (1980) examine a unidirectional adaptation along a fixed family of coordinate curves in the physical region. By taking the curves one at a time, they develop the movement as a sequence of one-dimensional problems. Relative to the arc length s of each curve, gradient and curvature clustering are inserted with M_1 and M_2 for the estimated size of the first and second derivatives of the monitor surface. Also in physical space, Gnoffo (1982) studies gradient adaptation in two directions with first derivatives for M_1. The sequence of curves now extends through one coordinate family at a time in a cyclic fashion that is iterated until convergence. Returning to the geometric viewpoint, Ablow (1982) considers solution surface arc length along coordinate curves. The partial differential equations describing equal arc length are iteratively solved.

As a synthesis and extension of the previous methods, Eiseman (1983a) presented a general alternating-direction adaptive strategy that uses the linear weight of Equation (28) for arbitrary surfaces. The process is executed sequentially on coordinate curves that are grouped by direction. A cyclic alternation between an arbitrary number of directions is iterated until convergence. Relative to surface arc length, the resolution of basic geometry is done with normal curvature. The coefficient for the weight with normal curvature is determined by a maximum fraction of points and a factor that varies from 0 to almost unity as curve arc length goes from a minimal value to a large one. The dynamic shift of coefficient from curve to curve ensures that a large increase in arc length would more strongly pull points into the bending locations that otherwise might be missed. The remaining locations in such a circumstance are well represented by arc length. With the adaptation being done on a curve-by-curve basis, the use of normal curvature is particularly important, since no clustering is done for curves that may bend within the surface during the iterative process. Rather, clustering is done only for the surface, and (as a favorable by-product) coordinate curves are pulled into alignment with folds in the surface. The importance of alignment has been accented by Anderson & Rai (1982). The geodesic curvature is the measure of curve bending within the surface and is employed only for surface boundaries that can nontrivially arise from the physical region. To resolve surface boundaries, its magnitude is included in the weight of Equation (28) with a coefficient that decays in the inward direction to shift clustering toward surface geometry. With clustering relative to arc-length uniformity, the grid structure can become skewed in places. To prevent excessive skewness, the deviations from orthogonality can be used in the weight to pull toward their equidistribution. While the 1 in the expression for w gives the uniformity measure of arc length, the measure can be essentially changed to equal volumes with a weight term for the Jacobian. Further weighting choices are possible, and these result in a balancing problem between the relative importance of the desired properties.

When the same weight is applied to surface volume elements, the uniformity measure is directly given by equal volumes. With weighted volume elements on the surface, Eiseman (1983b) geometrically constructs a local molecule that is relaxed in a point iterative style. For a two-dimensional surface grid, the molecule about each point is determined by cells corresponding to the four quadrants surrounding it. A separate balance along each coordinate curve is performed in a mean-value sense. For a given curve, this balance is between the total weighted volume on either side of the transverse coordinate curve. In each quadrant, the volume is that of the triangle that contains the point. The weight is applied at the barycenter. The weight center for a given side is then the vector sum for two adjacent triangles. Upon projection to the given curve, a weight is established on the curve at some location that is nearly a third of the outward distance to the neighboring grid point. The mean value is then taken along the curve to determine a new curvilinear location. From the new curvilinear locations, the new surface location is determined by bilinear interpolation. With the bilinear communication, the molecule can also be created directly in the space of curvilinear variables by using the weighted surface volumes. The Cartesian format then provides some algebraic simplification. With the molecule in either location, the transformation is always nonsingular as a continuum, and this is a distinct advantage. However, since it is evaluated at only one point in each molecule, the nonsingularity can be destroyed for the grid. This can occur only when the four quadrilateral cells about a surface point form a patch with severe concavity. To correct this possible situation, the sides adjacent to a point of concavity can be extended to determine values that limit movement along the facing curve. The use of such limiters assures that the grid will never overlap during the adaptive movement. In the context of a general connectivity triangular mesh, a robustness similar to that from nonoverlap comes with restructuring strategies [as reported by Fritts & Boris (1979), who considered Lagrangian movement]. Full adaptive movement was developed by Erlebacher (1984) and Erlebacher & Eiseman (1984). The basic movement is in physical space, curvature attraction is done with an approximation to the mean curvature of the monitor surface, and the movement molecule is a surface-area-weighted average over the triangles about a point. Global movement is done in a point iterative sense. With such iteration on geometrically constructed local molecules, the movement is determined by an implicitly defined elliptic system of partial differential equations.

ELLIPTIC SYSTEMS The Poisson system of Equation (24) is also directly applicable to adaptive movement. Mastin & Thompson (1983) considered the inclusion of monitor-surface derivatives in the forcing functions. Each derivative quantity is with respect to curvilinear variables and can be separately prescribed for each Poisson equation. The forcing function for each direction is a scaled first derivative of the quantity with respect to the same direction. To restrict the possibility of grid overlap, bounds were established for the scaling coefficients.

From a geometric view of the monitor surface, the Poisson system can be expressed with respect to the surface. The previous Laplacians are simply replaced by the Laplace-Beltrami operator, which is described by Warner (1970) in the context of Hodge theory. Proceeding directly from the vector expression for the inverse form of Equation (27), we may use the Gauss equations for second derivatives to suitably state the system. This was done by Warsi (1982). Another related development for surfaces was presented by Thomas (1982), who reasoned from a full three-dimensional Poisson system under the assumptions that curves entered the surface orthogonally and with no curvature. In each development, mean curvature appears as a source term. Although the surface grids were considered as parts of three-dimensional transformation techniques, they can be applied for adaptive movement. Conversely, the surface grids established for adaptive purposes can also be applied within the context of fixed higher-dimensional transformations.

THE VARIATIONAL SETTING A natural framework for elliptic methods is provided by a variational setting. The main distinction is that the parameters for grid control appear in global integrals, rather than in direct proportionality statements or explicitly in source terms. The previous controls were for direct elemental relationships and concavity. When a global linear combination of integrals is considered, the controlling coefficients are for the average quantities represented by integrals over the entire physical region rather than only parts of it. The grid is determined by taking the extremum of the integral combination, which produces the generating system in the form of Euler equations. On comparing the methods in the format of differential equations, we can observe the distinctive locations for control. The primary difference arises from the differential elements employed and the measure of uniformity.

In an isolated sense, the integral of a weighted volume is stationary over the physical region with physical differentials when the weighted volume is constant. The result is a proportionality statement for volumes. Upon returning to the integral, however, a change to curvilinear differentials indicates that the weight is applied to squared volumes rather than to simple weighted volumes. With the general linear weight of Equation (28), the leading term of unity would produce uniformity in the form of equal volumes. Other terms could shift it toward other measures such as conformality. Conformality would appear if M_1 is taken as the sum of the square residual of the Cauchy-Riemann equations. Given a large enough coefficient c_1 to dwarf the effect of the leading unit term, the integral would be essentially split into a Cauchy-Riemann part and the remaining weight without unity. Since the conformal part depends upon transformation derivatives, the stationary condition is now more complex than the previous proportionality statement. In terms of the metric coefficients, the integrand for the conformal part is $h_s = (g^{11} + g^{22})$, where the g^{ij} are the matrix inverse elements to g_{ij} in Equation (27). When the conformal part alone is stationary, the Laplace system from Winslow (1967) is obtained. This fact was first observed by Yanenko et al. (1978), who balanced the conformal part against a weighted volume $h_V = wJ^{\alpha}$ and a Lagrangian part $h_L = \|\mathbf{V}_F - \mathbf{V}_G\|^2$ (where $\mathbf{V}_F$ and $\mathbf{V}_G$ are fluid and grid velocities, respectively). With the integrals over the space-time volume element $dx\ dy\ dt$ conveniently denoted by I_s, I_V, and I_L, the total integral for the grid is given by $I_s + \lambda_V I_V + \lambda_L I_L$, where λ_V and λ_L are adjustable constants that with increased size bias toward weighted volume adaptivity and Lagrangian tracking, respectively.

The next development beyond Yanenko et al. (1978) was the inclusion of a separate orthogonality control, which in light of the conformality measure for uniformity might appear to be redundant. The main utility, however, is that orthogonality can be enhanced without the effect of dilating cell volumes, as would happen with conformality. Removing the Lagrangian measure I_L and the associated dt, Brackbill & Saltzman (1982) considered integrands with the cross metrics g_{12}^2 and $(g^{12})^2$, which lead to integrals I_0 and I'_0 and parameters λ_0 and λ'_0. If we drop the prime notation, then Euler equations are generated from $I_s + \lambda_0 I_0 + \lambda_V I_V$ with $\alpha = 1$. The conformal part I_s was called smoothness. Relative to smooth conditions from I_s, the level of adaptivity is adjusted with λ_V, and the grid structure is improved with increases in λ_0.

The metric forms in the integrals can be carried directly over to arbitrary surfaces as well as into higher dimensions. In the process with surfaces, an additional complication arises, since the chain rule must be employed to adjust to fixed local surface coordinates that enter by composition. In particular, the variational formulations can be done directly on the monitor surface.

SUMMARY

The need for grid generation arose from the discrete requirements for the numerical simulation in fluid mechanics with the geometric and topological complexity of physical regions and with the possibility of rapid solution variations at unpredictable locations. The general topological setting is examined to establish the overall formats for the desired grids without addressing the specific details required of the methods to meet the consequent variety of constraints.

The element of control over the grid is the most fundamental aspect needed to satisfy the constraints from topology, geometry, and solution variations. The most precise level of control is available from algebraic methods. This is established with the multisurface transformation in one direction and with the extension into multiple directions by means of Boolean operations. From a more relaxed level of precision, elliptic partial differential equation methods are developed from the Laplace system representing conformal conditions up to the Poisson system, where convexity controls are established and examined.

The general feedback cycle for adaptive movement is considered by using a monitor surface to consolidate the feedback data into a single object. With the natural objective of accurately representing this object, strategies for movement are considered. These cover methods from direct proportionality statements, Poisson systems, and variational formulations by taking the monitor surface either as a function over physical space or as a geometric entity. The basic control in the various methods is in the form of weight functions. With the placement in distinct spots in distinct formats, the consequent effects, although similar, are also distinct. How, when, and where control is applied give the separation between the methods and the important effects on the grid in the varying levels of response, directness, and precision.

ACKNOWLEDGMENT

This paper was written under US Air Force sponsorship with Grant No. AFOSR-82-0176A, monitored by John P. Thomas, Jr.

Literature Cited

Ablow, C. M., Schechter, S. 1978. Campylotropic coordinates. *J. Comput. Phys.* 27:351–62

Ablow, C. M. 1982. Equidistant mesh for gas dynamic calculations. See Thompson 1982a, pp. 859–64

Anderson, D. A., Rai, M. M. 1982. The use of solution adaptive grids in solving partial differential equations. See Thompson 1982a, pp. 317–38

Babuška, I., Chandra, J., Flaherty, J. E., eds. 1983a. *Adaptive Computational Methods for Partial Differential Equations*. Philadelphia: SIAM. 251 pp.

Babuška, I., Miller, A., Vogelius, M. 1983b. Adaptive methods and error estimation for elliptic problems of structural mechanics. See Babuška et al. 1983a, pp. 57–73

Barnhill, R. E., Riesenfeld, R. F., eds. 1974. *Computer Aided Geometric Design*. New York: Academic. 326 pp.

Brackbill, J. U., Saltzman, J. S. 1982. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.* 46: 342–68

Caughey, D. A. 1978. A systematic procedure for generating useful conformal mappings. *Int. J. Numer. Methods Eng.* 12: 1651–57

Coons, S. A. 1964. Surfaces for computer-aided design of space forms. *Project MAC*, Design Div., Dept. Mech. Eng., Mass. Inst. Technol., Cambridge

Dwyer, H. A. 1983. *Grid adaption for problems with separation, cell Reynolds number, shock–boundary layer interaction, and accuracy*. Presented at AIAA Aerosp. Sci. Meet., 21st, Reno, Nev. *Pap. AIAA-83-0449*

Dwyer, H. A., Kee, R. J., Sanders, B. R. 1980. An adaptive grid method for problems in fluid mechanics and heat transfer. *AIAA J.* 18:205–12

Eiseman, P. R. 1978. A coordinate system for a viscous transonic cascade analysis. *J. Comput. Phys.* 26:307–38

Eiseman, P. R. 1979. A multi-surface method of coordinate generation. *J. Comput. Phys.* 33:118–50

Eiseman, P. R. 1982a. Coordinate generation with precise controls over mesh properties. *J. Comput. Phys.* 47:331–51

Eiseman, P. R. 1982b. High level continuity for coordinate generation with precise controls. *J. Comput. Phys.* 47:352–74

Eiseman, P. R. 1982c. Orthogonal grid generation. See Thompson 1982a, pp. 193–234

Eiseman, P. R. 1982d. Automatic algebraic coordinate generation. See Thompson 1982a, pp. 447–64

Eiseman, P. R. 1983a. Alternating direction adaptive grid generation. *Proc. AIAA Comput. Fluid Dyn. Conf., 6th, Danvers, Mass.*, pp. 339–48

Eiseman, P. R. 1983b. Adaptive grid generation by mean value relaxation. See Ghia & Ghia 1983, pp. 29–34

Erlebacher, G. 1984. *Solution adaptive triangular meshes with application to plasma equilibrium.* PhD thesis. Columbia Univ., New York, N.Y. 208 pp.

Erlebacher, G., Eiseman, P. R. 1984. *Adaptive triangular mesh generation.* Presented at AIAA Fluid Dyn., Plasmadyn., Lasers Conf., 17th, Snowmass, Colo. *Pap. AIAA-84-1607*

Fritts, M. J., Boris, J. P. 1979. The Lagrangian solution of transient problems in hydrodynamics using a triangular mesh. *J. Comput. Phys.* 31:173–215

Ghia, K. N., Ghia, U., eds. 1983. *Advances in Grid Generation*, Vol. 5. New York: ASME. 219 pp.

Ghia, K., Ghia, U., Shin, C. T. 1983. Adaptive grid generation for flows with local high gradient regions. See Ghia & Ghia 1983, pp. 35–48

Gnoffo, P. A. 1982. A vectorized, finite volume, adaptive-grid algorithm for Navier-Stokes. See Thompson 1982a, pp. 819–36

Gordon, W. J. 1971. Blending-function methods of bivariate and multivariate interpolation and approximation. *SIAM J. Numer. Anal.* 8:158–77

Gordon, W. J., Hall, C. A. 1973. Construction of curvilinear coordinate systems and applications to mesh generation. *Int. J. Numer. Methods Eng.* 7:461–77

Gordon, W. J., Thiel, L. C. 1982. Transfinite mappings and their application to grid generation. See Thompson 1982a, pp. 171–92

Ives, D. C. 1982. Conformal grid generation. See Thompson 1982a, pp. 107–36

Klopfer, G. H., McRae, D. S. 1981. The nonlinear modified equation approach to analyzing finite difference schemes. *Proc. AIAA Comput. Fluid Dyn. Conf., 5th, Palo Alto, Calif.*, pp. 317–33

Laugwitz, D. 1965. *Differential and Riemannian Geometry.* New York: Academic. 238 pp.

Mastin, C. W., Thompson, J. F. 1978. Elliptic systems and numerical transformations. *J. Math. Anal. Appl.* 62:52–62

Mastin, C. W., Thompson, J. F. 1983. *Adaptive grids generated by elliptic systems.* Presented at AIAA Aerosp. Sci. Meet., 21st, Reno, Nev. *Pap. AIAA-83-0451*

Middlecoff, J. F., Thomas, P. D. 1979. Direct control of the grid point distribution in meshes generated by elliptic equations. *Proc. AIAA Comput. Fluid Dyn. Conf., 4th, Williamsburg, Va.*, pp. 175–79

Miller, K., Miller, R. 1981. Moving finite elements I. *SIAM J. Numer. Anal.* 18:1019–32

Moretti, G. 1980. Grid generation using classical techniques. See Smith 1980, pp. 1–36

Piva, R., DiCarlo, A., Favini, B., Guj, G. 1982. Adaptive curvilinear grids for large Reynolds number viscous flows. In *Lecture Notes in Physics*, 170:414–19. Berlin/Heidelberg/New York: Springer-Verlag

Prenter, P. M. 1975. *Splines and Variational Methods.* New York: Wiley Interscience. 323 pp.

Rheinboldt, W. C. 1983. Feedback systems and adaptivity for numerical computations. See Babuška et al. 1983a, pp. 3–19

Smith, R. E., ed. 1980. *Numerical Grid Generation Techniques, NASA CP 2166.* 574 pp.

Smith, R. E. 1982. Algebraic grid generation. See Thompson 1982a, pp. 137–70

Stadius, G. 1977. Construction of orthogonal curvilinear meshes by solving initial value problems. *Numer. Math.* 28:25–48

Steger, J. L., Sorenson, R. 1979. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *J. Comput. Phys.* 33:405–16

Steger, J. L., Sorenson, R. L. 1980. Use of hyperbolic partial differential equations to generate body fitted coordinates. See Smith 1980, pp. 463–78

Thomas, P. D. 1982. Numericalgeneration of composite three-dimensional grids by quasilinear elliptic systems. See Thompson 1982a, pp. 667–86

Thompson, J. F., ed. 1982a. *Numerical Grid Generation*. New York: North Holland. 909 pp.

Thompson, J. F. 1982b. Elliptic grid generation. See Thompson 1982a, pp. 79–106

Thompson, J. F., Thames, F. C., Mastin, C. W. 1974. Automatic numerical generation of body-fitted curvilinear coordinate system for field containing any number of arbitrary two-dimensional bodies. *J. Comput. Phys.* 15:299–319

Thompson, J. F., Thames, F. C., Mastin, C. W. 1977. TOMCAT—A code for numerical generation of boundary-fitted curvilinear coordinate systems on fields containing any number of arbitrary two-dimensional bodies. *J. Comput. Phys.* 24:274–302

Warner, F. W. 1971. *Foundations of Differentiable Manifolds and Lie Groups*. Glenview, Ill./London: Scott, Foresman. 270 pp.

Warsi, Z. U. A. 1982. Basic differential models for coordinate generation. See Thompson 1982a, pp. 41–78

Warsi, Z. U. A., Thompson, J. F. 1976. Machine solutions of partial differential equations in the numerically generated coordinate systems. *Rep. MSSU-EIRS-ASE-77-1*, Miss. State Univ., Miss. State

White, A. B. Jr. 1979. On selection of equidistributing meshes for two-point boundary-value problems. *SIAM J. Numer. Anal.* 16:472–502

White, A. B. Jr. 1982. On the numerical solution of initial boundary-value problems in one space dimension. *SIAM J. Numer. Anal.* 19:683–97

Winslow, A. M. 1967. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.* 2:149–72

Yanenko, N. N., Kovenya, V. M., Lisejkin, V. D., Fomin, V. M., Vorozhtsov, E. V. 1978. On some methods for the numerical simulation of flows with complex structure. In *Lecture Notes in Physics*, 90:565–78. Berlin/Heidelberg/New York: Springer-Verlag