JOURNAL OF COMPUTATIONAL PHYSICS 82, 269–288 (1989)

A Laplacian Equation Method for Numerical Generation of Boundary-Fitted 3D Orthogonal Grids

T. THEODOROPOULOS AND G. C. BERGELES

Laboratory of Aerodynamics, Department of Mechanical Engineering, National Technical University of Athens, Greece

Received November 20, 1986; revised July 5, 1988

A method for generating boundary fitted orthogonal curvilinear grids in 3-dimensional space is described. The mapping between the curvilinear coordinates and the Cartesian coordinates is provided by a set of Laplace equations which, expressed in curvilinear coordinates, involve the components of the metric tensor and are therefore non-linear and coupled. An iterative algorithm is described, which achieves a numerical solution. Grids appropriate for the calculation of flow fields over complex topography or in complex flow passages as those found in turbomachinery, and for other engineering applications can be constructed using the proposed method. Various examples are presented and plotted in perspective, and data for the assessment of the properties of the resulting meshes is provided. © 1989 Academic Press, Inc.

1. INTRODUCTION

The need to solve Reynolds equations expressed in domain fitted curvilinear orthogonal coordinate systems has created a demand for an efficient method to generate such meshes for given geometries. Reasons for this include the reduction of numerical diffusion during the solution of Reynolds equations (as one of the grid coordinates can approximate in direction the expected streamlines of the flow), the convenience in the expression of boundary conditions in complicated geometries, and the avoidance of the complexities and the substantial computer time and storage requirements usually associated with the expression of Reynolds equations on non-orthogonal meshes.

Most of the literature on the generation of orthogonal meshes is on 2-dimensional methods. There are also algorithms, in both 2- and 3-dimensional spaces, for the generation of non-orthogonal meshes fitted in the computational domain. Some of these orthogonal and non-orthogonal approaches will be mentioned in the following.

In the non-orthogonal case, of importance is the work of Thompson *et al.* [1, 2] and Thomas *et al.* [3, 4] in which a system of Poisson equations is employed and inverted to give the physical coordinates as functions of the curvilinear coordinates. Amsden and Hirt [5] also constructed such grids through an algebraic procedure.

0021-9991/89 $3.00

For the generation of orthogonal meshes in 2-dimensional space many researchers (Barfield [6], etc.) use conformal mapping. The essential characteristics of this approach are the absence of sufficient flexibility in mesh control and the fact that small changes in the shape of the domain can drastically alter the position of the mapped boundary points. Pope [7] attempted to alleviate the problem of conformal mapping, which is the double requirement of orthogonality and equality of the scale factors, by constructing orthogonal mappings in which the ratio of the scale factors is not unity but an adjustable constant called "scaling factor." Potter and Tuttle [8] and Davies [9] suggested ways of orthogonalisation of non-orthogonal meshes. This technique sometimes leads to discrete orthogonal coordinates, in the sense that there are discontinuities in one family of grid lines. Haussling and Coleman [10] worked on the modification of the technique described in Thompson *et al.* [1, 2], to make it suitable for the generation of orthogonal grids by means of enforcing conditions on the coefficients of the inverted system. Mobley and Stewart [11], Visbal and Knight [12], and Ryskin and Leal [13] included Laplace equations in their generating systems of elliptic equations. Reviews and other articles on orthogonal and non-orthogonal grid generation can be found in [14].

The work presented here follows our previous work [15] on the generation of orthogonal grids in 2-dimensional space. In the following paragraphs we present the mathematical and numerical formulation of the problem and the application of our algorithm to some geometries of practical interest.

2. Mathematical Formulation

A vital problem in the area of 3-dimensional grid generation is the problem of existence. The theoretical statement of the conditions under which there is a Euclidean manifold that can be mapped on a specific region of interest, and satisfy the orthogonality conditions there, meets significant mathematical complexities but there is plenty of evidence (Ryskin and Leal [13], Sokolnikoff [16], etc.) that such a mapping exists in many practical cases. In the present work we assume that there is a Euclidean curvilinear orthogonal coordinate system which can be assigned to a 3-dimensional region of practical interest and describe how mapping functions can be derived.

Bearing the above in mind, we establish the system to be found as our system of reference in the physical space, where the Cartesian coordinates x, y, z are known to satisfy

$$\nabla^2 x = 0, \qquad \nabla^2 y = 0, \qquad \nabla^2 z = 0. \tag{1}$$

We introduce the orthogonal curvilinear coordinates as ξ, η, ζ. The Laplace operator can be expressed in its appropriate form for the coordinates ξ, η, ξ to give x, y, z as functions of ξ, η, ζ.

The formulation of the problem must ensure a solution of the Laplacians which, in its Cartesian form, should read

$$\begin{aligned} \text{grad } x &= (1, 0, 0) \\ \text{grad } y &= (0, 1, 0) \\ \text{grad } z &= (0, 0, 1) \end{aligned} \tag{2}$$

everywhere in the domain of interest.

The Cartesian coordinates on the entire boundary of the physical domain are known and it is important that they satisfy Eq. (2). It is these boundary values for x, y, z that will secure that (2) are satisfied in the interior too. Each of these equations is obviously a solution of the respective Laplacian in the entire domain unless it is not satisfied on the boundaries. This can be easily seen if (2) are substituted in the Cartesian form of (1). The solution will be exactly the same with (1) expressed in our reference system since they are tensor equations! We face a typical boundary value problem for which there exists a unique solution (Courant and Hilbert [17]). Thus, since (2) form a solution, this will be the unique solution. This means that the x, y, z are uniquely expressed in terms of ξ, η, ζ in the desired way. In other words, the solution of (1) on our curvilinear system of reference, the Cartesian coordinates of the physical domain boundaries being known, is certain to be (2) if expressed in the Cartesian coordinates. Of course, (2) will assume a different form in our reference system. We aim to derive this form of the solution because it will provide us with the desired relations which give x, y, z as functions of ξ, η, ζ.

In the following we will assume summation on repeated indices. Let h_{ij} be the elements of the covariant metric tensor of a multi-dimensional coordinate system $\xi^1, \xi^2, ..., \xi^n$, sometimes called metric coefficients, which define the length of the element of arc through the fundamental quadratic form

$$ds^2 = h_{ij}\, d\xi^i\, d\xi^j.$$

If h is the determinant of this tensor and h^{ij} are the elements of the corresponding contravariant metric tensor, the Laplace operator in curvilinear coordinates can be written as

$$\frac{\partial}{\sqrt{h}\,\partial\xi^i}\left(\sqrt{h}\,h^{ij}\frac{\partial}{\partial\xi^j}\right). \tag{3}$$

When the metric coefficients are associated with an orthogonal curvilinear frame, (3) reduces to

$$\frac{1}{\sqrt{h_{11}\cdots h_{nn}}}\cdot\frac{\partial}{\partial\xi^j}\left(\frac{\sqrt{h_{11}\cdots h_{j-1j-1}h_{j+1j+1}\cdots h_{nn}}}{h_{jj}}\frac{\partial}{\partial\xi^j}\right). \tag{4}$$

For Cartesian coordinates (4) simply reads

$$\frac{\partial^2}{\partial \xi^j \, \partial \xi^j}.$$

Applied on $x^1, x^2, ..., x^n$, denoting the Cartesian coordinates, (4) will give (1) as strongly coupled non-linear equations with h_{ij} given by

$$h_{ij} = \frac{\partial x^\alpha}{\partial \xi^i} \frac{\partial x^\alpha}{\partial \xi^j}. \tag{5}$$

Equation (4) can be used to create the required mapping and we have indeed done so in some applications. However, a step by step orthogonalisation of an initial grid can provide a practical alternative way to be followed in the establishment of a numerical solution algorithm in order to ensure that convergence problems will be eliminated.

For a stepwise solution procedure simplifications in the form (3) of the Laplace operator are introduced. The cross-derivative terms are omitted and h^{ii} is set equal to $1/h_{ii}$, as for orthogonal coordinates, giving

$$\frac{1}{\sqrt{h}} \frac{\partial}{\partial \xi^i} \left(\sqrt{h}\, h^{ij} \frac{\partial}{\partial \xi^j} \right) \simeq \frac{1}{\sqrt{h}} \frac{\partial}{\partial \xi^i} \left(\sqrt{h}\, h^{ii} \frac{\partial}{\partial \xi^i} \right) \simeq \frac{1}{\sqrt{h}} \frac{\partial}{\partial \xi^i} \left(\frac{\sqrt{h}}{h_{ii}} \frac{\partial}{\partial \xi^i} \right). \tag{6}$$

The above simplifications appear reasonable as we approach orthogonality.

In another way of looking at the simplification of (3), the Laplace operator applied on a scalar $\nabla^2 \phi$ may be considered as $\nabla \cdot (\nabla \phi) \equiv \nabla \cdot \mathbf{F}$. The expression of divergence in curvilinear coordinates is

$$\nabla \cdot \mathbf{F} = \frac{1}{\sqrt{h}} \frac{\partial}{\partial \xi^i} (\sqrt{h}\, F^i).$$

Replacing F^i by $h^{ij}(\partial \phi / \partial \xi^j)$ which is the form for the gradient in curvilinear coordinates we get (3) applied on ϕ. However, if we assume that F^i can be approximated by $1/h_{ii}\, \partial \phi / \partial \xi^i$, which is strictly accurate only in the case of orthogonal coordinates, we arrive at the simplified form appearing in (6).

Hence the task can be carried out using the simpler form (6) and we have found the tedious calculations implied by (3) unnecessary.

We introduce a "deviation factor" D as

$$D = \frac{\sqrt{h}}{\sqrt{h_{11} h_{22} \cdots h_{nn}}}. \tag{7}$$

The simplified Laplace operator for the non-orthogonal system then reads

$$\frac{1}{\sqrt{h}} \cdot \frac{\partial}{\partial \xi^j} \left(D \frac{\sqrt{h_{11} \cdots h_{j-1\,j-1} h_{j+1\,j+1} \cdots h_{nn}}}{h_{jj}} \right) \frac{\partial}{\partial \xi^j}. \tag{8}$$

We define the scale factors as $h_{\xi^i} = \sqrt{h_{ii}}$ and, for three dimensions, if $\xi^1 = \xi$, $\xi^2 = \eta$, $\xi^3 = \zeta$, we get (1) in the form

$$\frac{\partial}{\partial \xi}\left(D\frac{h_\eta h_\zeta}{h_\xi}\frac{\partial \phi}{\partial \xi}\right) + \frac{\partial}{\partial \eta}\left(D\frac{h_\zeta h_\xi}{h_\eta}\frac{\partial \phi}{\partial \eta}\right) + \frac{\partial}{\partial \zeta}\left(D\frac{h_\xi h_\eta}{h_\zeta}\frac{\partial \phi}{\partial \zeta}\right) = 0, \qquad \phi = x, y, z. \tag{9}$$

The solution of these equations will provide the coordinate transformation. Since

$$\cos a_{ij} = \frac{h_{ij}}{\sqrt{h_{ii}h_{jj}}}$$

where a_{ij} is the angle formed by the ξ^i and ξ^j lines in the physical space, (7) can also be written as

$$D = (2\cos a_{\xi\eta}\cos a_{\eta\zeta}\cos a_{\zeta\xi} - \cos^2 a_{\xi\eta} - \cos^2 a_{\eta\zeta} - \cos^2 a_{\zeta\xi} + 1)^{1/2}. \tag{10}$$

Comparing (4) and (8), it can be seen that the stepwise orthogonalisation procedure may be realised by imposing a stepwise approach of D to unity. The imposed D_{imp} for the next step is expressed in terms of the actual D of the previous step as

$$D_{\text{imp}} = D + (1 - D)/s, \qquad s > 1 \tag{11}$$

The scale factors involved in (9) can be obtained through (5) as

$$h_{\xi^i}^2 = \frac{\partial x^\alpha}{\partial \xi^i}\frac{\partial x^\alpha}{\partial \xi^i}. \tag{12}$$

However, since the simple substitution of (12) into (9) was observed to lead to numerical instabilities in some cases, our practice was to use this definition on the boundaries and, for one of the curvilinear coordinates, for example ξ, in the interior of the domain, and employ a kind of "aspect ratio" functions

$$r_{\xi\eta} \equiv \frac{h_\xi}{h_\eta}, \qquad r_{\eta\zeta} \equiv \frac{h_\eta}{h_\zeta} \tag{13}$$

for the rest. Definition (13) was applied for r's all over the domain boundary. Since h_ξ, h_η, h_ζ were observed to vary smoothly in the interior, simple interpolation

$$r(\xi, \eta, \zeta) = \{(1-\xi)\, r(0, \eta, \zeta) + \xi r(1, \eta, \zeta) + (1-\eta)\, r(\xi, 0, \zeta) + \eta r(\xi, 1, \zeta) + (1-\zeta)\, r(\xi, \eta, 0) + \zeta r(\xi, \eta, 1)\}/3, \qquad r = r_{\xi\eta}, r_{\eta\zeta} \tag{14}$$

was used to yield their values there. Then the scale factors could be provided by

$$h_\eta = \frac{h_\xi}{r_{\xi\eta}}, \qquad h_\zeta = \frac{h_\eta}{r_{\eta\zeta}}.$$

3. Finite Difference Discretization

The integration of (9) over the finite volume shown in Fig. 1 provides their finite difference expression, which is suitable for the numerical solution. The result is of the form

$$C_P\phi_P = C_E\phi_E + C_W\phi_W + C_N\phi_N + C_S\phi_S + C_U\phi_U + C_D\phi_D, \tag{15}$$

where $\phi = x, y, z$ and

$$C_E = \left(\frac{Dh_\eta h_\zeta}{h_\xi}\right)_e \frac{\delta\eta\,\delta\zeta}{\Delta\xi_{PE}}, \qquad C_W = \left(\frac{Dh_\eta h_\zeta}{h_\xi}\right)_w \frac{\delta\eta\,\delta\zeta}{\Delta\xi_{WP}}$$

$$C_N = \left(\frac{Dh_\zeta h_\xi}{h_\eta}\right)_n \frac{\delta\zeta\,\delta\xi}{\Delta\eta_{PN}}, \qquad C_S = \left(\frac{Dh_\zeta h_\xi}{h_\eta}\right)_s \frac{\delta\zeta\,\delta\xi}{\Delta\eta_{SP}}$$

$$C_U = \left(\frac{Dh_\xi h_\eta}{h_\zeta}\right)_u \frac{\delta\xi\,\delta\eta}{\Delta\zeta_{PU}}, \qquad C_D = \left(\frac{Dh_\xi h_\eta}{h_\zeta}\right)_d \frac{\delta\xi\,\delta\eta}{\Delta\zeta_{DP}}$$

$$C_P = C_E + C_W + C_N + C_S + C_U + C_D.$$

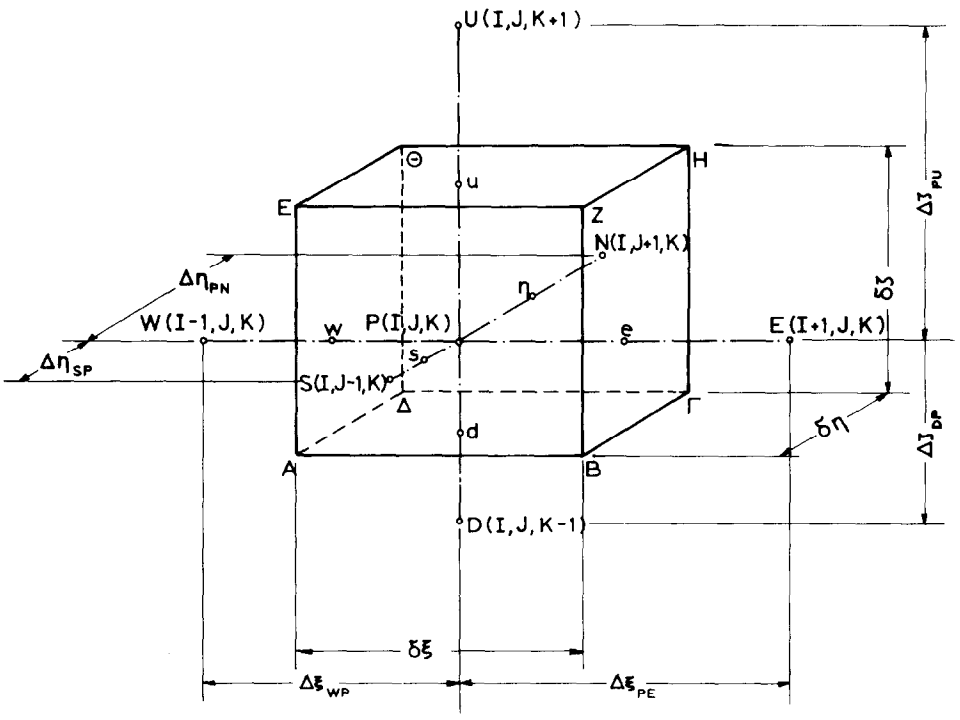

FIG. 1. The computational cell for the f.d. equations.

The defining of the scale factors equation (12), where used, was discretized as

$$h_{\xi^i} = \frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)^{1/2}_{\xi^i \text{line}}}{\Delta \xi^i}. \tag{16}$$

4. Boundary Conditions

The typical domain under consideration was, in the physical space, bounded by six curved surfaces of freely determined shape with the single restriction that they intersect each other orthogonally (Fig. 2). If this requirement is not satisfied, the problem is apparently not well posed. In practice, however, sensible deviations from this rule will not prohibit convergence.

Let $x^3 = f(x^1, x^2)$ be the equation which describes one of the boundary surfaces ($x = x(y, z)$, $y = y(z, x)$, $z = z(x, y)$ as appropriate). It is obvious that, for reasons of economy in computing, a very fast and efficient method is needed to impose the orthogonality of the mesh lines to the boundaries.

Consider the calculation of the normal from a point (x^{10}, x^{20}, x^{30}) to the $x^3 = f(x^1, x^2)$ surface. The system

$$\frac{\partial((x^{10} - x^1)^2 + (x^{20} - x^2)^2 + (x^{30} - x^3)^2)}{\partial x^1} = 0$$

$$\frac{\partial((x^{10} - x^1)^2 + (x^{20} - x^2)^2 + (x^{30} - x^3)^2)}{\partial x^2} = 0$$

$$f(x^1, x^2) - x^3 = 0$$

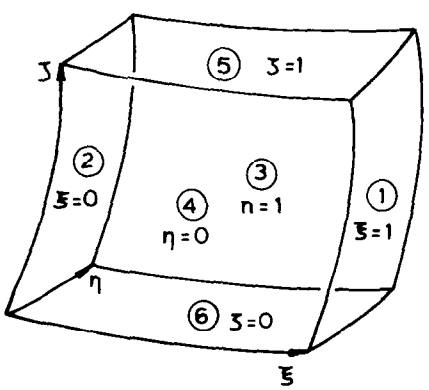

FIG. 2. Domain of computation.

may be used which finally adopts the form

$$\begin{pmatrix} x^1 \\ x^2 \\ x^3 \end{pmatrix} = \begin{pmatrix} x^{10} + x^{30}\dfrac{\partial f(x^1, x^2)}{\partial x^1} - x^3\dfrac{\partial f(x^1, x^2)}{\partial x^1} \\ x^{20} + x^{30}\dfrac{\partial f(x^1, x^2)}{\partial x^2} - x^3\dfrac{\partial f(x^1, x^2)}{\partial x^2} \\ f(x^1, x^2) \end{pmatrix} = g\begin{pmatrix} x^1 \\ x^2 \\ x^3 \end{pmatrix}$$

or, concisely,

$$\mathbf{x} = g(\mathbf{x}). \tag{17}$$

For a given initial $\mathbf{x}^0$, we can construct a sequence of vectors $\mathbf{x}^\kappa$ with the formula

$$\mathbf{x}^\kappa = g(\mathbf{x}^{\kappa-1}). \tag{18}$$

It can be mathematically proved that this sequence converges for an appropriate choice of $\mathbf{x}^0$ if

$$\left|\frac{\partial f}{\partial x^1}\right| + \left|\frac{\partial f}{\partial x^2}\right| < 1$$

is true in a neighbourhood of the foot of the normal to be calculated. Introducing under relaxation into (18), through an under relaxation factor ω, we arrive at

$$\mathbf{x}^\kappa = (1-\omega)\mathbf{x}^{\kappa-1} + \omega g(\mathbf{x}^{\kappa-1}) \tag{19}$$

and convergence is assured for much greater slopes of the boundary surface, even greater than 70° for both directions x^1, x^2 as we found in our applications of the method, where we used

$$\omega = 0.2.$$

The main advantage of this procedure is its rapid convergence, and, hence, the very little computational effort required. However, if greater slopes relative to the independent variable plane for the surface descriptive functions f (near 90°) are involved in regions of the domain boundaries, and if the method described proves computationally inefficient, another method should be used on these regions. The expression of f in other coordinates or the representation of the surface by a sufficiently large number of plane elements can provide such alternatives.

In some engineering applications there is need for more direct control over the position of the boundary nodes than Neumann type boundary conditions would

allow. The imposition of Dirichlet boundary conditions is a topic for investigation. Clearly, orthogonal quasi-2-dimensional grids on curved surfaces will have to be generated on the boundaries of the 3-dimensional domain, where Dirichlet boundary conditions will be used. This can be achieved through a simple extention of a 2-dimensional method of orthogonal grid generation, for example, the method described in [15].

5. Polynomial Approximation of a Curved Surface through the Least Square Method with Two Independent Variables

In complex topographies we often know the shape of a surface through the coordinates of a set of its points but there is no analytic expression to establish the boundary conditions in the way mentioned in Section 4. We proceed with the task of constructing analytic approximations for these cases.

Consider that m surface descriptive vectors (x^*, y^*, z^*) are known. We will approximate the values z^* with a polynomial $z(x, y)$ so that for any (x^*, y^*, z^*),

$$z^* \simeq z(x^*, y^*).$$

Let the maximum power of each independent variable be n. Then the polynomial is of the form

$$\begin{aligned} z = {} & a_{00} + a_{01}\, y + a_{02}\, y^2 + \cdots + a_{0n}\, y^n \\ & + a_{10} x + a_{11} xy + a_{12} xy^2 + \cdots + a_{1n} xy^n \\ & + a_{n0} x^n + a_{n1} x^n y + a_{n2} x^n y^2 + \cdots + a_{nn} x^n y^n, \qquad (n+1)^2 < m. \end{aligned}$$

The quantity

$$E = \sum_{r=1}^{m} (z_r - z_r^*)^2$$

is minimum if the $(n+1)^2$ equations

$$\frac{\partial E}{\partial a_{pq}} = 0, \qquad p, q = 0, 1, \ldots, n,$$

are satisfied.

Thus we have a system of $(n+1)^2$ equations with $(n+1)^2$ unknowns, the coefficients a_{pq}. After the calculations, the kth equation of this system becomes

$$\begin{aligned}\sum z^* x^{*\lambda} y^{*\mu} = {} & a_{00} \sum x^{*\lambda} y^{*\mu} + a_{01} \sum x^{*\lambda} y^{*\mu+1} \\ & + a_{02} \sum x^{*\lambda} y^{*\mu+2} + \cdots + a_{0n} \sum x^{*\lambda} y^{*\mu+n} \\ & + a_{10} \sum x^{*\lambda+1} y^{*\mu} + a_{11} \sum x^{*\lambda+1} y^{*\mu+1} + a_{12} \sum x^{*\lambda+1} y^{*\mu+2} \\ & + \cdots + a_{1n} \sum x^{*\lambda+1} y^{*\mu+n} + a_{n0} \sum x^{*\lambda+n} y^{*\mu} + a_{n1} \sum x^{*\lambda+n} y^{*\mu+1} \\ & + a_{n2} \sum x^{*\lambda+n} y^{*\mu+2} + \cdots + a_{nn} \sum x^{*\lambda+n} y^{*\mu+n},\end{aligned}$$

where

$$\lambda = \text{integer}\left(\frac{k-1}{n+1}\right)$$

$$\mu = k - \lambda(n+1) - 1.$$

6. Selection of the ξ, η, ζ Distributions

An inspection of the coefficients of (15) will verify that we have freedom in selecting convenient $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$, to meet the requirements of the specific application by effectively controlling the spacing of the resulting mesh in the physical domain through the particular choice of the useful curvilinear coordinate surfaces of the mapping. Each of these surfaces is associated with one particular value of one of the ξ, η, ζ and, therefore, by specifying appropriate $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$ distributions we can effectively choose the mesh surfaces which will be calculated.

Possibilities include

- $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$ constant throughout the field.
- $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$ terms of geometric progressions.
- $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$ polynomially distributed.

7. Initial Guess

Execution time depends on the properties of the initial grid to be orthogonalized. It is sufficient to apply as initial values for the solution of Eq. (9) the results obtained from the solution of (9) with the assumptions $h_\xi = h_\eta = h_\zeta = 1$ and $D = 1$. In other words, the initial values of x, y, z are found by solving

$$\frac{\partial^2 \phi}{\partial \xi^2} + \frac{\partial^2 \phi}{\partial \eta^2} + \frac{\partial^2 \phi}{\partial \zeta^2} = 0 \tag{20}$$

for $\phi = x, y, z$.

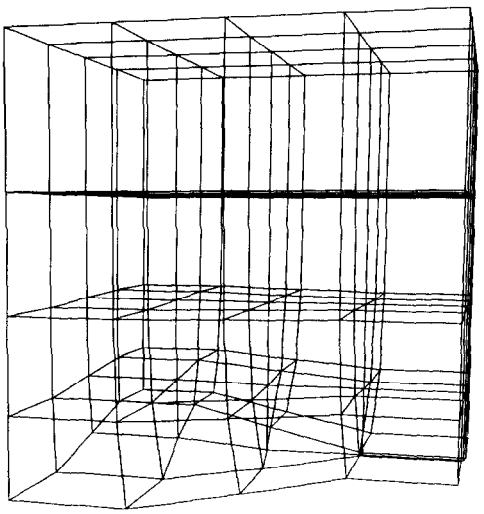

FIG. 3. Orthogonal mesh over a gentle topography.

8. NUMERICAL SOLUTION PROCEDURE

The numerical algorithm, which applies the method under discussion, has the following steps:

1. With arbitrary preliminary values, solution of (20) leads to the initial guess for $x(\xi, \eta, \zeta)$, $y(\xi, \eta, \zeta)$, $z(\xi, \eta, \zeta)$ and, through them, for $D(\xi, \eta, \zeta)$, $h_\xi(\xi, \eta, \zeta)$, $h_\eta(\xi, \eta, \zeta)$, $h_\zeta(\xi, \eta, \zeta)$ by use of (10), (12), (13), (14).

2. These values of the metrics being held constant, (9) are effectively decoupled and each of them is solved iteratively, in combination with the boundary conditions described in the respective chapter, for the computation of new $x(\xi, \eta, \zeta)$, $y(\xi, \eta, \zeta)$, $z(\xi, \eta, \zeta)$.

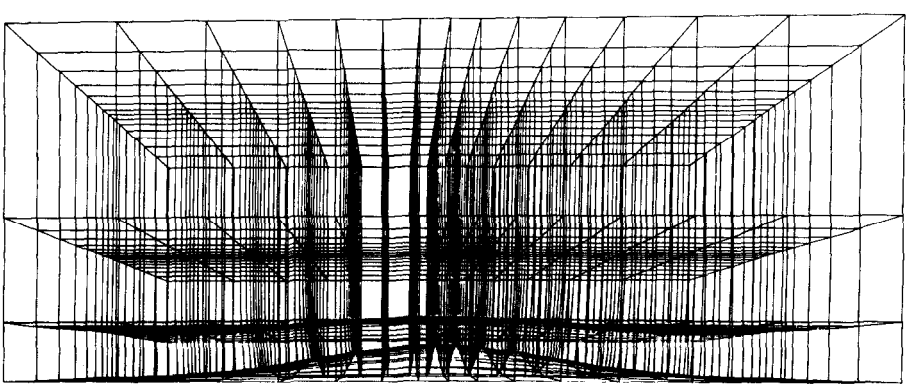

FIG. 4. Orthogonal mesh over an axisymmetric Gaussian hill.

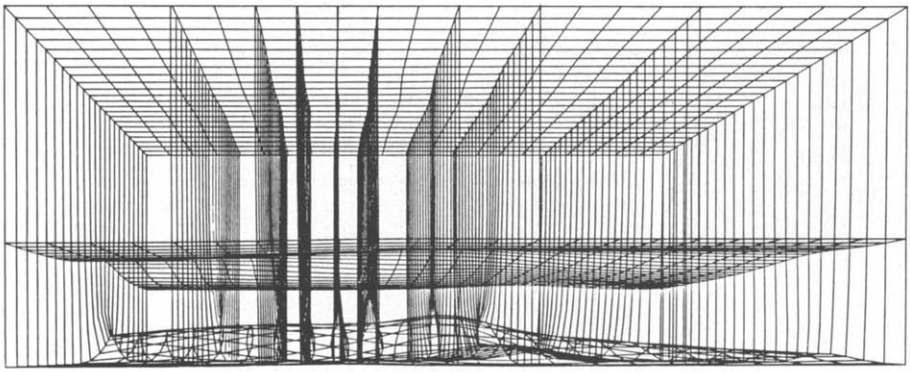

FIG. 5. Orthogonal mesh in a terrain composed of Gaussian hills.

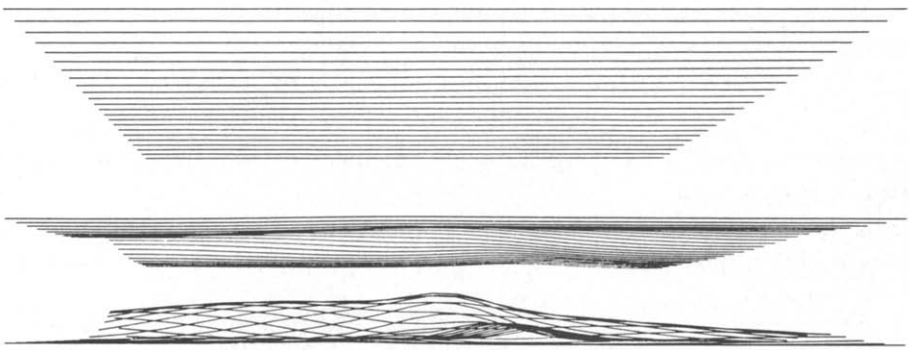

FIG. 6. Partial view of the mesh of Fig. 5.

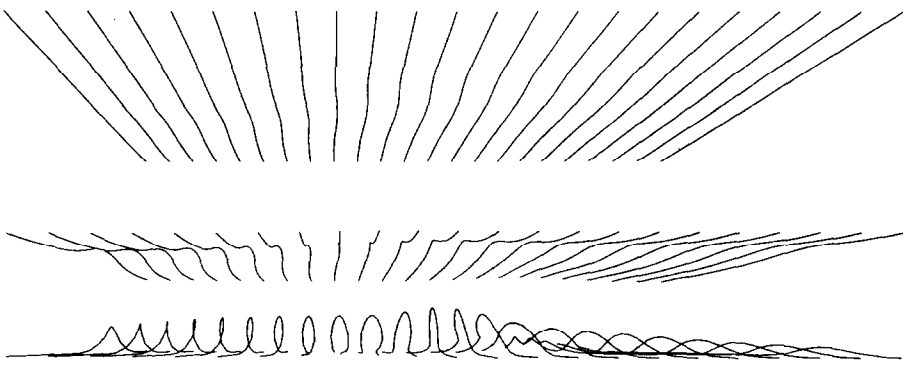

FIG. 7. Partial view of the mesh of Fig. 5.

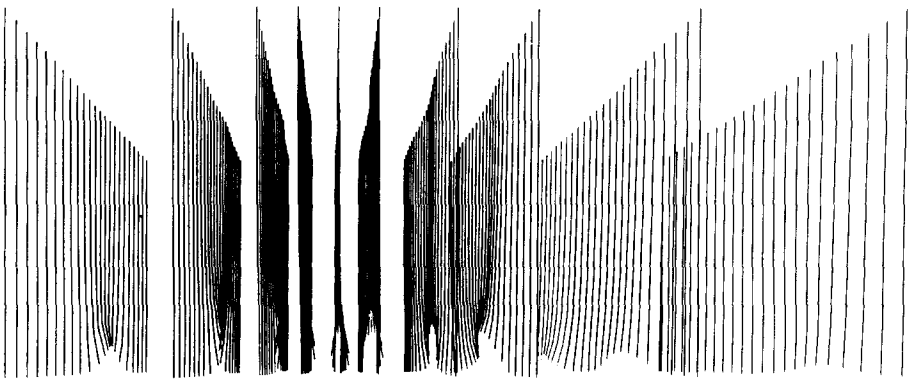

FIG. 8. Partial view of the mesh of Fig. 5.

3. The values of the metrics are computed with the latest data through the corresponding relations (10), (11), (12), (13), (14).

4. Step (2) is reentered unless convergence has already been reached. The latter is assumed when $x(\xi, \eta, \zeta)$, $y(\xi, \eta, \zeta)$, $z(\xi, \eta, \zeta)$ do not change more than a small number e, even when the new metrics are inserted in the coefficients of (9).

For the calculation of the angles involved in (10) we have used spline interpolation through the grid points to approximate the grid line shape and, hence, the slopes at these points.

It is clear that the iterative scheme described here cannot converge if the metrics involved in the equations do not converge. The deviation factor can only converge to unity because of (11).

Consider that a solution has been reached and, thus, $D = 1$. Algebraically manipulating expression (10) for the deviation factor we see that this can be satisfied if, and only if, the three angles $a_{\xi\eta}$, $a_{\eta\zeta}$, $a_{\zeta\xi}$ are 90°. In other words, if the method converges, then $D = 1$ and the coordinate system is orthogonal, as D are calculated on the finally generated mesh.

What we have not provided is a full mathematical investigation of the conditions under which the algorithm should always converge, in the sense that for a practical domain of interest, of the type discussed in paragraph (4), there does or does not exist a Euclidean manifold to be yielded, which will be orthogonal there. Such an attempt seems to meet many theoretical obstacles and may be the scope of future work. However, we will mention that the procedure converged in all of our applications.

9. EXAMPLES OF APPLICATION

In this paragraph, we present orthogonal grids for various geometries as generated with the algorithm described. For the computations we have used a CYBER 171 CDC computer and for the plots a BENSON 2320 plotter.

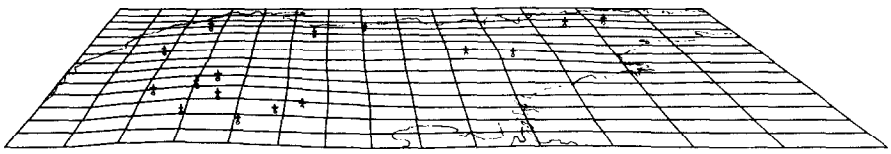

FIG. 9. The Athens topography in perspective.

Figure 3 corresponds to the numerical solution over a gentle topography, mathematically presented as the product of harmonic functions

$$z = \sin(2\pi x)\sin(2\pi y)/20.$$

The differences $\Delta\zeta$ are terms of a geometric progression with common ratio $\omega_\zeta = 1.3$, and $\Delta\xi$, $\Delta\eta$ are equally distributed.

Figure 4 shows the results of the algorithm over an axisymmetric Gaussian hill. The equation which describes the surface of a hill of this kind is

$$z = a \cdot \exp(-r^2/b^2),$$

where r is the horizontal distance from the axis of symmetry of the hill. For this application the maximum height and the characteristic length are $a = 50$ m and $b = 250$ m, respectively. It should be mentioned here that for every three $\zeta = \text{const}$ grid surfaces, only one is presented for reasons of figure clarity. $\Delta\xi$, $\Delta\eta$, $\Delta\zeta$ form geometric progressions with $\omega_\xi = 0.8$, $\omega_\eta = 0.8$, $\omega_\zeta = 1.25$, respectively.

The solution obtained over a composite topography is shown in Fig. 5. The associated equation is

$$z = 60\exp(-((x+2y-200)^2 + y^2)/250^2) + 30\exp(-((2x-y+100)^2 + (x+150)^2/200^2)$$

which means that the boundary surface is a superposition of two non-axisymmetric Gauss hill surfaces. Figure 5 has been broken up to produce Fig. 6, 7, and 8, each related to a set of grid lines.

In the following Fig. 9 the topography of the Athens vicinity is shown in perspective, where the crosses mark mountain tops of height greater than 600 m. The plotted area is 70000 by 50000 m. The surface geometry has been polynomially approximated, according to the theory described in paragraph 5. The maximum power of each independent variable of the polynomial is 7 and its coefficients have been calculated with 121 surface descriptive vectors as data. For greater accuracy

FIG. 10. Orthogonal mesh in the Athens topography.

in the representation, the surface can be better approximated by employing greater powers in the polynomial and more descriptive vectors, at the expense however of computer time, mainly during the application of the boundary conditions, as the surface function involved is then more demanding. The choice is dictated by the user needs but we should mention that the computer time requirements are in any case moderate. Figure 10 includes some of the generated grid surfaces and shows the ground anomalies. In the example shown of the Athens complex, $\Delta\eta$ are equally distributed, $\Delta\zeta$ are terms of a geometric progression of $\omega_\zeta = 1.5$ and $\Delta\xi$ are according to the polynomial

$$\xi = 0.84226107\xi|_{eq}^3 - 1.68292657\xi|_{eq}^2 + 1.86095466\xi|_{eq}$$

with $\xi|_{eq}$ denoting the value of ξ for equidistributed $\Delta\xi$.

The last example presented here is a channel of a type which is met in axial turbomachinery. The domain is bounded by two cylindrical surfaces of radii $r_u = 1$, $r_d = 0.5$, respectively, two plane surfaces at distance $x = 1$ (equations $x_w = 0$, $x_e = 1$, respectively), and two radially extending at angles ϕ_n and ϕ_s from a plane which contains the axis of the cylindrical boundaries. These angles change polynomially.

$$\phi_n = 0.5235988 + 0.0887974x - 1.837189x^2 + 1.224792x^3 \quad \text{(rad)}$$

$$\phi_s = 0.0887974x - 1.837189x^2 + 1.224792x^3 \quad \text{(rad)}.$$

Figure 11, which shows the resulting mesh, is accompanied by its component Figs. 12 and 13.

A typical example of the way the method converges is shown in Fig. 14 which is related to the Gaussian hill of Fig. 4. Convergence was reached in 14 computations of the coupling parameters, and 91 iterations in total for the solution of the artificially decoupled equations.

As measures of deviation from orthogonality, the following parameters are introduced

$$D_{\xi\eta} = \frac{|a_{\xi\eta} - \pi/2|}{\pi/2} \cdot 100\%$$

$$D_{\eta\zeta} = \frac{|a_{\eta\zeta} - \pi/2|}{\pi/2} \cdot 100\%$$

$$D_{\zeta\xi} = \frac{|a_{\zeta\xi} - \pi/2|}{\pi/2} \cdot 100\%.$$

Information about the mean and maximum deviation from orthogonality is provided in Table I, where the following equations have been used for a *lxmxn* grid

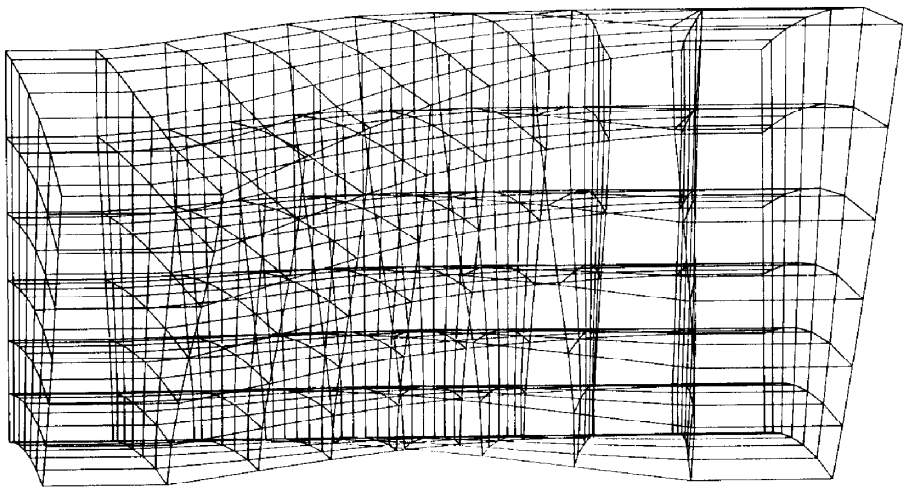

FIG. 11. Orthogonal mesh in a channel of axial turbomachinery.

$$D_{\xi^i\xi^j\text{MEAN}} = \frac{\overline{D_{\xi\eta}} + \overline{D_{\eta\zeta}} + \overline{D_{\zeta\xi}}}{3} = \sum_{\substack{i=1,\ldots,l \\ j=1,\ldots,m \\ k=1,\ldots,n}} \frac{D_{\xi\eta}(i,j,k) + D_{\eta\zeta}(i,j,k) + D_{\zeta\xi}(i,j,k)}{3lmn}$$

$$D_{\xi^i\xi^j\text{MAX}} = \text{MAX}\Big\{ \underset{\substack{i=1,\ldots,l \\ j=1,\ldots,m \\ k=1,\ldots,n}}{\text{MAX}} D_{\xi\eta}(i,j,k),\ \underset{\substack{i=1,\ldots,l \\ j=1,\ldots,m \\ k=1,\ldots,n}}{\text{MAX}} D_{\eta\zeta}(i,j,k),\ \underset{\substack{i=1,\ldots,l \\ j=1,\ldots,m \\ k=1,\ldots,n}}{\text{MAX}} D_{\zeta\xi}(i,j,k) \Big\}.$$

Finally, Table II includes parameters which provide information about the rate of convergence and computer execution time and storage requirements.

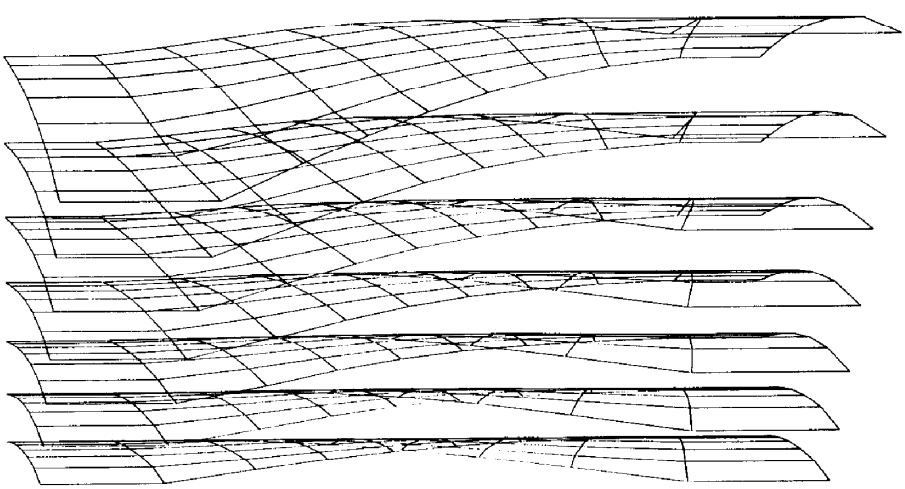

FIG. 12. Partial view of the mesh of Fig. 11.

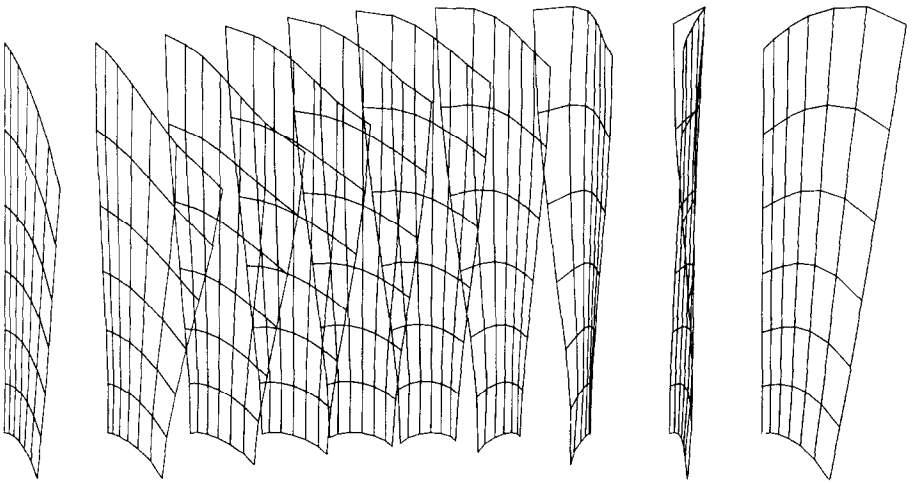

FIG. 13. Partial view of the mesh of Fig. 11.

The convergence factor is defined as

$$C = e/(v/(l-1)(m-1)(n-1))^{1/3},$$

where e is the maximum difference $|x^i_{\text{new}} - x^i_{\text{old}}|$, $x^i = x, y, z$ allowed for the determination of convergence, and v is the total volume of the domain of the mapping in the physical space.

The total number of iterations needed for the solution of (9), treated as decoupled and with the coefficients only varying in a stepwise way, is presented under the column of Laplace iterations.

The maximum execution field length is an octal number and shows the maximum number of words which were required in the central memory of the computer during execution. Each word consists of 10 characters.

The CPU computer time requirements could be regarded as modest, however, they are specific to each case considered and the complexity of the geometry.

10. CONCLUSIONS

This study has addressed the problem of generating orthogonal curvilinear coordinates in spaces of practical importance. The gradual orthogonalisation of non-orthogonal meshes with a system of non-linear and coupled covariant Laplace equations was achieved through the development of a flexible and economic numerical algorithm. This algorithm has allowed the generation of orthogonal curvilinear meshes in various geometries. Questions regarding the existence or uniqueness of solution, however, have to be theoretically considered. The existence

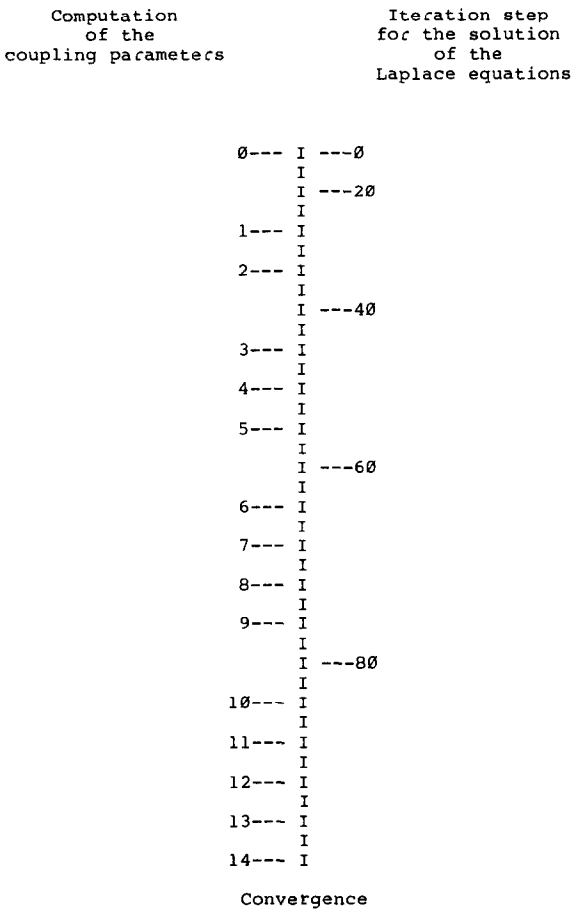

FIG. 14. An illustration of the iterative solution sequence.

TABLE I

Figure	(%) $D_{\xi^i\xi^j/\mathrm{MEAN}}$	(%) $D_{\xi^i\xi^j/\mathrm{MAX}}$	D_{MEAN}	D_{MAX}	D_{MIN}
3	—	—	1	1	0.9993
4	—	—	1	1	0.9999
5	—	—	1	1	0.9981
9	1	6.48	1	1	0.9995
11	1.7556	9.5433	0.9999	1	0.9986

TABLE II

Figure	Grid	$C \times 10^4$ convergence factor	s control parameter of D modification	Iterations. Initial guess	Iterations. Laplace solution	Execution time CPU s	Maximum execution field length
3	$5 \times 5 \times 5$	8.0	1.1	6	168	30	40000
4	$15 \times 15 \times 10$	6.2	1.1	90	91	312	105400
5	$23 \times 23 \times 7$	11.6	1.1	205	128	5612	131700
9	$15 \times 15 \times 7$	23.6	1.1	319	145	3107	76300
11	$10 \times 7 \times 7$	12.2	1.1	58	240	367	57100

of solution in the example cases considered has been indirectly proved by the convergence of the iterative solution procedure and the generation of orthogonal curvilinear meshes in 3-dimensional spaces.

REFERENCES

1. J. F. THOMPSON, F. C. THAMES, AND C. W. MASTIN, *J. Comput. Phys.* **15**, 299 (1974).
2. F. C. THAMES, J. F. THOMPSON, C. W. MASTIN, AND R. L. WALKER, *J. Comput. Phys.* **24**, 245 (1977).
3. P. D. THOMAS AND J. F. MIDDLECOFF, *AIAA J.* **18**, No. 6, 652 (1980).
4. P. D. THOMAS, *AIAA J.* **20**, No. 9, 1195 (1982).
5. A. A. AMSDEN AND C. W. HIRT, *J. Comput. Phys.* **11**, 348 (1973).
6. W. D. BARFIELD, *J. Comput. Phys.* **5**, 23 (1970).
7. S. B. POPE, *J. Comput. Phys.* **26**, 197 (1978).
8. D. E. POTTER AND G. H. TUTTLE, *J. Comput. Phys.* **13**, 483 (1973).
9. C. W. DAVIES, *J. Comput. Phys.* **39**, 164 (1981).
10. H. J. HAUSSLING AND R. M. COLEMAN, *J. Comput. Phys.* **43**, 373 (1981).
11. C. D. MOBLEY AND R. J. STEWART, *J. Comput. Phys.* **34**, 124 (1980).
12. M. VISBAL AND D. KNIGHT, *AIAA J.* **20**, No. 3, 305 (1982).
13. G. RYSKIN AND L. G. LEAL, *J. Comput. Phys.* **50**, 71 (1983).
14. J. F. THOMPSON, *Numerical Grid Generation* (North-Holland, Amsterdam, 1982).
15. T. THEODOROPOULOS, G. BERGELES, AND N. ATHANASSIADIS, in *Proceedings, Fourth International Conference in Numerical Methods in Laminar and Turbulent Flow, Swansea, U.K.*, 1985, edited by C. Taylor *et al.* (Pineridge, Swansea, UK, 1985), p. 1747.
16. I. S. SOKOLNIKOFF, *Tensor Analysis* (Wiley, New York, 1964).
17. R. COURANT AND D. HILBERT, *Methods of Mathematical Physics* Vol. 2 (Wiley-Interscience,