INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **30**: 523–540 (1999)

A FINITE VOLUME METHOD FOR NUMERICAL GRID GENERATION

S.B. BEALE*

National Research Council of Canada, Ottawa, Ontario K1A 0R6, Canada

SUMMARY

A novel method to generate body-fitted grids based on the direct solution for three scalar functions is derived. The solution for scalar variables ξ, η and ζ is obtained with a conventional finite volume method based on a physical space formulation. The grid is adapted or re-zoned to eliminate the residual error between the current solution and the desired solution, by means of an implicit grid-correction procedure. The scalar variables are re-mapped and the process is reiterated until convergence is obtained. Calculations are performed for a variety of problems by assuming combined Dirichlet–Neumann and pure Dirichlet boundary conditions involving the use of transcendental control functions, as well as functions designed to effect grid control automatically on the basis of boundary values. The use of dimensional analysis to build stable exponential functions and other control functions is demonstrated. Automatic procedures are implemented: one based on a finite difference approximation to the Cristoffel terms assuming local-boundary orthogonality, and another designed to procure boundary orthogonality. The performance of the new scheme is shown to be comparable with that of conventional inverse methods when calculations are performed on benchmark problems through the application of point-by-point and whole-field solution schemes. Advantages and disadvantages of the present method are critically appraised. Copyright © 1999 National Research Council of Canada.

KEY WORDS: grid generation; finite volume method; body-fitted co-ordinates

1. INTRODUCTION

Numerical methods are applied readily to a wide range of problems in fluid flow, heat, and mass transfer. These may be considered to be governed by the single equation

$$D(\phi)=\underbrace{\frac{\partial}{\partial t}(\rho\phi)}_{\text{Transient}}+\underbrace{\vec{\nabla}\cdot(\rho\vec{u}\phi)}_{\text{Convection}}-\underbrace{\vec{\nabla}\cdot\Gamma\vec{\nabla}\phi}_{\text{Diffusion}}-\underbrace{s}_{\text{Source}}=0. \tag{1}$$

A wide range of engineering problems based on Equation (1) have been solved successfully through the use of computational fluid dynamics (CFD) codes based on established numerical schemes, such as the finite volume method [1–3], where $\phi = u, v, w, h, \ldots$ are solved iteratively. This paper shows how the grid variables may be added to the list of state variables. Although grid smoothing and grid control should be considered as an integral part of the overall solution procedure, in general, they are treated as a separate subject. Several books [4,5] and review articles [6–8] have appeared on the subject.

* Correspondence to: National Research Council of Canada, Montreal Road, Ottawa, Ontario K1A 0R6, Canada. E-mail: steven.beale@nrc.ca

Received December 1997
Revised March 1998

Grid generation involves the stipulative definition of functions denoted by Greek letters ξ^i, or alternatively ξ, η, ζ, which are to be considered a function of position $\xi^i(\vec{r}) = \xi^i(x^j\vec{e}_j)$, or more simply $\xi^i = \xi^i(x^j)$. The relationships between ξ^i and x^j are prescribed through the use of partial differential equations, as follows:

$$\frac{\mathrm{D}(\xi^i)}{\mathrm{D}(x^j)} = 0. \tag{2}$$

The Roman letters x^i (or x, y, z) denote Cartesian components. Equation (2) is sometimes referred to as a 'physical space' formulation. The reader will note that this paper assumes tensor notation [9] and the summation convention in order to present the formulation in a compact fashion. Subsequently, the scalar notation ξ, η, ζ and x, y, z is adopted.

Allen [10], one of the pioneers of the so-called inverse methods, observed that the use of a finite difference scheme with a Cartesian mesh leads to an imprecise prescription of (Neumann) boundary conditions along curved boundaries and to an imperfect solution for ζ^i. Instead, he proposed the analytical inversion of the governing (Laplace) equations, i.e. the interchange of dependent and independent variables

$$\frac{\mathrm{D}(x^i)}{\mathrm{D}(\xi^j)} = 0, \tag{3}$$

so that the Cartesian co-ordinates of the grid are obtained in 'transformed space'. Most grid generation schemes are based on the analytical inversion of Equation (2), although formulations have also been proposed directly in transformed space [11,12]. These early methods caused grid folding, though more recent hyperbolic forms [13,14] are in use. Inverse methods are not the only plausible approach to grid generation. If the governing equations are formulated by means of vector operators, any suitable co-ordinate system may be employed to obtain a solution, and Equation (2) may be written as

$$\mathrm{D}(\xi^i) = \frac{\mathrm{D}(\xi^i)}{\mathrm{D}(\xi^{*j})} = 0. \tag{4}$$

In Equation (4), ξ^{*j} could represent a Cartesian, polar [15,16] or general body-fitted co-ordinate (BFC) system or grid, like the one used here. Equation (4) is the generic form solved in this paper. An initial ξ^{*j} grid is generated algebraically, and the equations for ξ^i are discretised in physical space. The key to the procedure is that values of the solved for scalars are substituted back into the grid as the solution proceeds.

Figure 1 illustrates these ideas pictorially. With scalars specified as a function of position in physical space, interpolation is required along curved boundaries to obtain the solution in Cartesian co-ordinates, $\xi^i = \xi^i(x^j)$. In conventional schemes, the solution obtained for $x^i = x^i(\xi^j)$ corresponds to a rectangular grid in the transformed space. The approach presented in this paper provides the solution in a BFC system, in physical space. Because the system is body-fitted, boundary interpolation is not required, and Allen's problem does not arise.

The governing equations may be parabolic, hyperbolic or elliptic, the latter being the most popular. Early elliptic grid generation equations were based on Laplace systems [17], where only the diffusion term in Equation (1) is non-zero (with constant Γ). With suitable boundary conditions, such a system can generate sets of orthogonal grids [18]. Laplace systems satisfy an *extremum* principle, namely that the mapping be proper (1–1 and monotonic, with boundary values spanning the interior). Numerous systems of equations based on diffusion source (Poisson) equations [19,20], with the generic form

$$\mathrm{D}(\xi^i) = \vec{\nabla}\cdot\vec{\nabla}\xi^i - P^i = 0, \tag{5}$$

have been employed widely in grid generation. The source terms P^i (or P, Q, R) are used as 'control functions' in order to alter the position or slope of grid lines or surfaces. This form corresponds to a simplified version of Equation (1) with $s = -P^i$. Well-established computational methods for this form have existed for years: this suggests that these methods could well be adapted to grid smoothing. Note that the formulation is co-ordinate independent, i.e. it is written in terms of the vector operator $\vec{\nabla}$. The inverse form is

$$g^{jk}\frac{\partial^2 x^i}{\partial\xi^j\,\partial\xi^k} - P^j\frac{\partial x^i}{\partial\xi^j} = 0. \tag{6}$$

This paper considers primarily numerical solutions for Equation (5), not Equation (6).

1.1. Finite volume equations

The so-called 'mathematical' form [21] of Equation (1) is

$$\mathrm{D}(\phi) = \frac{\partial}{\partial t}(\sqrt{g}\phi) + \frac{\partial}{\partial\xi^{*j}}\left(\sqrt{g}u^j\phi - \sqrt{g}g^{jk}\Gamma\frac{\partial\phi}{\partial\xi^{*k}}\right) - \sqrt{g}s = 0, \tag{7}$$

where $\phi = \xi^i = \xi,\ \eta,\ \zeta$, for $i = 1, 2, 3$. The metric components, g^{jk} and Jacobian $\sqrt{g}$, refer to the ξ^{*i} co-ordinate system. The use of general scalar transport equations offers many interesting possibilities. In order to illustrate the technique, however, attention will be focused on the Poisson system used conventionally in grid generation, which may be considered a subset of Equation (7), as follows:

$$\frac{\partial}{\partial\xi^{*j}}\left(\sqrt{g}g^{jk}\frac{\partial\phi}{\partial\xi^{*k}}\right) - \sqrt{g}P^i = 0. \tag{8}$$

Equation (8) may be discretised as [1–3,22]

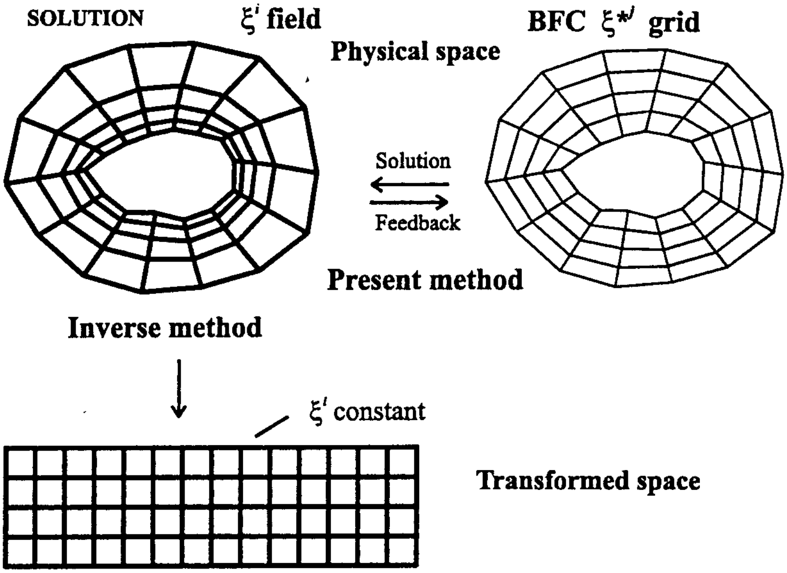

Figure 1. Conceptual schematic of grid generation methodology.

$$\phi_P = \frac{a_W\phi_W + a_E\phi_E + a_S\phi_S + a_N\phi_N + a_L\phi_L + a_H\phi_H + a_T\phi_T + CV}{a_W + a_E + a_S + a_N + a_L + a_H + a_T + C}, \tag{9}$$

where the dependent variables are $\phi = (\xi, \eta, \zeta)$, and the compass notation [1,3] W, E, S, N, L and H refers to the west ξ^*, east ξ^*, south η^*, north η^*, low ζ^*, and high ζ^* neighbours of P. For Poisson equations, the linking coefficients are due to diffusion alone. The T values refer to the previous grid, and are to be considered as a mechanism for inertial relaxation (ϕ is steady, but the grid is not). The source term, S, is linearised according to

$$S = C(V - \phi_P). \tag{10}$$

In a non-orthogonal system, it is necessary to introduce the diagonal terms EN, EH, NH, etc., in the finite volume equations. This is accomplished by treating these as geometric source terms, so that

$$S = S_{BC} + S_{GEOM} - \sqrt{g}P, \tag{11}$$

where S_{GEOM} is the non-orthogonal source term and S_{BC} is a boundary source term. An expression for S_{GEOM} used in the present method is provided in Table I.

1.2. Grid correction

Suppose a BFC grid, with $\vec{r}^*_P = (x^*_P, y^*_P, z^*_P)$, has been generated. Let ξ_{ref}, η_{ref}, ζ_{ref} be the desired reference values, often chosen to be natural numbers 1, 2, 3, . . . If the grid corners are not at the desired places, ξ_P, η_P, ζ_P will differ from ξ_{ref}, η_{ref}, ζ_{ref}. Displacement correction factors $\vec{r}'$ are then added as follows:

$$\vec{r}_P = \vec{r}^*_P + \alpha\vec{r}'_P, \tag{12}$$

and

$$\vec{r}'_P = (\xi_{ref} - \xi_P)\frac{\partial\vec{r}}{\partial\xi} + (\eta_{ref} - \eta_P)\frac{\partial\vec{r}}{\partial\eta} + (\zeta_{ref} - \zeta_P)\frac{\partial\vec{r}}{\partial\zeta}, \tag{13}$$

where α is a linear relaxation coefficient. A central difference grid-correction scheme is used with

$$\frac{\partial\vec{r}}{\partial\xi} = \frac{\vec{r}_E - \vec{r}_W}{2}, \tag{14}$$

$$\frac{\partial\vec{r}}{\partial\eta} = \frac{\vec{r}_N - \vec{r}_S}{2}, \tag{15}$$

$$\frac{\partial\vec{r}}{\partial\zeta} = \frac{\vec{r}_H - \vec{r}_L}{2}, \tag{16}$$

where it is tacitly assumed that $\xi_{ref_E} - \xi_{ref_P} = 1$. These could be chosen as real numbers or values of ξ_P, η_P, ζ_P at particular nodes; in this event, these terms should appear in the denominator of the grid correction equations. Grid correction is applied implicitly over the entire field. After re-zoning the grid, the state variables are re-mapped as follows:

$$\phi = (1 - \alpha)\phi^*_P + \alpha\phi_{ref}, \quad \phi = \xi, \eta, \zeta, \tag{17}$$

and the process is repeated. (For the usual case $\alpha = 1$, ϕ is simply re-set to initial values 1, 2, 3, . . .) The numerical solution of the finite volume equations, together with the grid-correction procedure, constitute a complete description of the methodology.

Table I. Comparison of present and inverse schemes

	Present method $\phi = (\xi, \eta, \zeta)$	Inverse method $\phi = (x, y, z)$
Coefficients	$a_{\mathrm{W}} = (\sqrt{g}g^{11})_w, \quad a_{\mathrm{E}} = (\sqrt{g}g^{11})_e$ $a_{\mathrm{S}} = (\sqrt{g}g^{22})_w, \quad a_{\mathrm{N}} = (\sqrt{g}g^{22})_n$ $a_{\mathrm{L}} = (\sqrt{g}g^{33})_l, \quad a_{\mathrm{H}} = (\sqrt{g}g^{33})_h$	$a_{\mathrm{E}} = a_{\mathrm{W}} = (g^{11})_P$ $a_{\mathrm{N}} = a_{\mathrm{S}} = (g^{22})_P$ $a_{\mathrm{L}} = a_{\mathrm{H}} = (g^{33})_P$
Non-orthogonal terms	$S = (\sqrt{g}g^{12})_e(\phi_{en} - \phi_{es}) + (\sqrt{g}g^{13})_e(\phi_{eh} - \phi_{el})$ $-(\sqrt{g}g^{12})_w(\phi_{wn} - \phi_{ws}) - (\sqrt{g}g^{13})_w(\phi_{wh} - \phi_{wl})$ $+(\sqrt{g}g^{12})_n(\phi_{en} - \phi_{wn}) + (\sqrt{g}g^{23})_n(\phi_{nh} - \phi_{nl})$ $-(\sqrt{g}g^{12})_s(\phi_{es} - \phi_{ws}) - (\sqrt{g}g^{23})_s(\phi_{sh} - \phi_{sl})$ $+(\sqrt{g}g^{13})_h(\phi_{eh} - \phi_{wh}) + (\sqrt{g}g^{23})_h(\phi_{nh} - \phi_{sh})$ $-(\sqrt{g}g^{13})_l(\phi_{el} - \phi_{wl}) - (\sqrt{g}g^{23})_l(\phi_{nl} - \phi_{sl})$	$S = 2g^{12}_P(\phi_{en} + \phi_{ws} - \phi_{es} - \phi_{wn})$ $+2g^{13}_P(\phi_{eh} + \phi_{wl} - \phi_{el} - \phi_{wh})$ $+2g^{23}_P(\phi_{sh} + \phi_{nl} - \phi_{sl} - \phi_{nh})$
Control functions	$S = -\begin{cases} (\sqrt{g}P)_P & \phi = \xi \\ (\sqrt{g}Q)_P & \phi = \eta \\ (\sqrt{g}R)_P & \phi = \zeta \end{cases}$	$S = -[\max(0, P)_P(\phi_{\mathrm{E}} - \phi_{\mathrm{P}})$ $+\min(0, \mathrm{P})_P(\phi_P - \phi_{\mathrm{W}})$ $+\max(0, Q)_P(\phi_{\mathrm{N}} - \phi_P)$ $+\min(0, Q)_P(\phi - \phi_{\mathrm{S}})$ $+\max(0, R)_P(\phi_{\mathrm{H}} - \phi_P)$ $+\min(0, R)_P(\phi_P - \phi_{\mathrm{L}})]$

1.3. Solution procedure

A variety of algorithms has been established for the solution of Equation (9). These include point-by-point, line, slab and whole-field solvers, which are documented in various reports and theses [2,23]. These have been used in a number of CFD codes over the years. A series of inner 'iterations' are performed over the block for each field variable, on the basis of fixed values of linking coefficients and source terms. After calculations have been performed for all field variables, the grid is re-zoned implicitly. The metric components, g^{ij}, linking coefficients and built-in sources are then updated. The entire process is repeated for a number of outer iterations, or 'sweeps', until convergence is achieved.

1.4. Boundary conditions

Technically, the term 'boundary condition' applies to the scalar variables ξ, η, ζ, not the current grid. These are prescribed as source terms, as described below.

(i) Neumann problem: the normal gradients $\partial\phi/\partial n$ are equivalent to fixed sources. The situation $\partial\phi/\partial n = 0$ corresponds to the default $S = 0$. (ii) Dirichlet problem: the ϕ values are fixed to V with a large coefficient, C (method of Payne and Irons [24]). The variable V should be consistent at opposing nodes in two dimensions (2D) or around an entire 'slab' in three dimensions (3D). (iii) Mixed Dirichlet–Neumann problem: corresponding to the 'natural' boundary value problem; this will produce good results under most circumstances. Two fixed-value Dirichlet boundary conditions are required in addition to two (2D) or four (3D) Neumann boundary conditions. Practical considerations often prevent this formulation. Other combinations are encountered: for example in 3D, the six boundary surfaces may be treated as mixed Dirichlet–Neumann, as above, with the 12 bounding space curves fixed.

Grid correction is not necessary at Dirichlet boundaries. At Neumann boundaries, grid correction is applied subject to an additional constraint: for example, $\xi(x, y, z) = \text{constant}$, unless boundary orthogonality is induced by means of variable source terms, in which case grid correction should not be applied.

1.5. Control functions

Control functions are often introduced as source terms and, typically, are formulated (1) to concentrate cells in boundary layers, (2) to effect a measure of orthogonality (i.e. reduce distortion) at grid boundaries, or both. Early control functions were transcendental functions designed to concentrate cells in appropriate regions. So-called 'automatic' procedures were introduced subsequently to compute the control functions from the boundary geometry. These are discussed further below.

1.5.1. Transcendental control functions. The exponential forms of Thompson *et al.* [25] were among the first used control functions:

$$P = a\ \text{sign}(\xi - \xi_0) \times \exp(c|\xi - \xi_0|), \tag{18}$$

and

$$P = b\ \text{sign}(\xi - \xi_0) \times \exp(-d[(\xi - \xi_0)^2 + (\eta - \eta_0)^2]^{1/2}). \tag{19}$$

These forms were used to effect attractions to surfaces $\xi = \xi_0$ and space curves $\xi = \xi_0$, $\eta = \eta_0$ respectively. Frequently, surfaces are chosen to correspond to the boundary values $\xi = 1$ or $\xi = n$, although they could correspond to interior regions. A known problem of these functions is that it is easy to generate values of P that cause numerical divergence. In one dimension (1D), it can be shown that a diffusion source formulation will generate proper functions satisfying the *extremum* principle if and only if

$$-2 \leq \frac{PL^2}{\xi_{\max} - \xi_{\min}} \leq 2, \tag{20}$$

where $\xi_{\min}$ and $\xi_{\max}$ are values at $x = 0$ and $x = L$ respectively. It is easy to derive the conditions under which coefficients a and c in Equation (18) will violate this condition (e.g. for $\xi_0 = 1$). It is, however, possible to generate exponential-type control functions that do not violate the *extremum* principle through the use of dimensional analysis. An example is a 1D stretching function $x^+(\xi^+)$,

$$x^+ = \frac{1 - \exp(a\xi^+)}{1 - \exp(a)}, \tag{21}$$

where $x^+ = x - x_{\min}/x_{\max} - x_{\max}$, $\xi^+ = \xi - 1/n - 1$, and $n - 1$ is the number of cells in the ξ-direction. This function congregates points in the form of a geometric progression for constant $\Delta\xi$, and the sign of a determines to which boundary the grid lines are attracted. Figure 2 shows the required property for the $\mathrm{d}\xi^+/\mathrm{d}x^+$ distribution is that of skewness for asymmetric stretching functions and kurtosis for the symmetric case. The sharper the spike, the more effective the concentration. Conversely, $\mathrm{d}\xi^+/\mathrm{d}x^+$ should not approach zero elsewhere, which would indicate that cells are being pulled out excessively. The control function is obtained from $\mathrm{d}^2\xi^+/\mathrm{d}x^{+2}$, and may be written as

$$P = -\left(\frac{n-1}{L^2}\right) \times \frac{(e^a-1)^2}{a} \exp\left(-2a\frac{\xi-1}{n-1}\right). \tag{22}$$

This function does not suffer from the deficiencies discussed above. A Poisson equation, with P prescribed according to Equation (22) satisfies Equation (21) in 1D. The length scale $L = |\vec{r}_{\max} - \vec{r}_{\min}|$ should always be a maximum to assure compliance with the *extremum* principle. Other control function terms may be used [26] readily: for example, to concentrate cells at both ends or the centre of the domain.

1.5.2. Automatic control functions. Transcendental control functions have been supplanted, to some degree, by so-called 'automatic' schemes. The way these are usually computed from the grid geometry is described below. Poisson's equation may be rearranged to solve for P^i (P, Q, R), where

$$P^i = g^{jk}\begin{Bmatrix} i \\ jk \end{Bmatrix} = g^{il}g^{jk}[jk, l] \tag{23}$$

and Christoffel's symbols are given by

$$[jk, l] = \frac{\partial \vec{e}_j}{\partial x^k}\cdot\vec{e}_l = \frac{1}{2}\left(\frac{\partial g_{jl}}{\partial \xi^k} + \frac{\partial g_{kl}}{\partial \xi^l} - \frac{\partial g_{jk}}{\partial \xi^l}\right). \tag{24}$$

Existing techniques invariably invoke this tautology, modified by the presumption of local boundary orthogonality:

$$P^i = \frac{[jj, i]}{g_{ii}g_{jj}} \quad (i, \text{ free}; \ j, \text{ summed}). \tag{25}$$

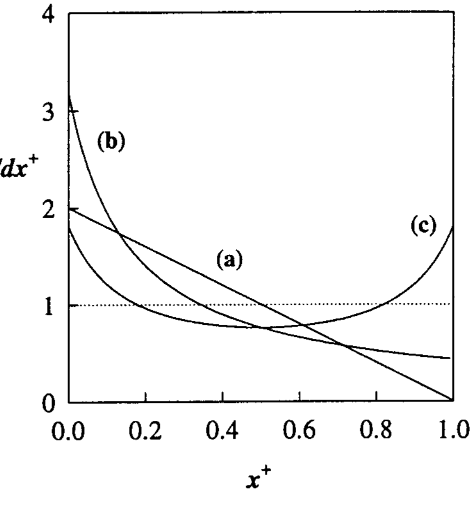

Figure 2. 1D stretching functions: the skewness or kurtosis controls the mesh distribution: (a) constant source P; (b) one-sided attraction according to Equation (22); (c) two-sided attraction.

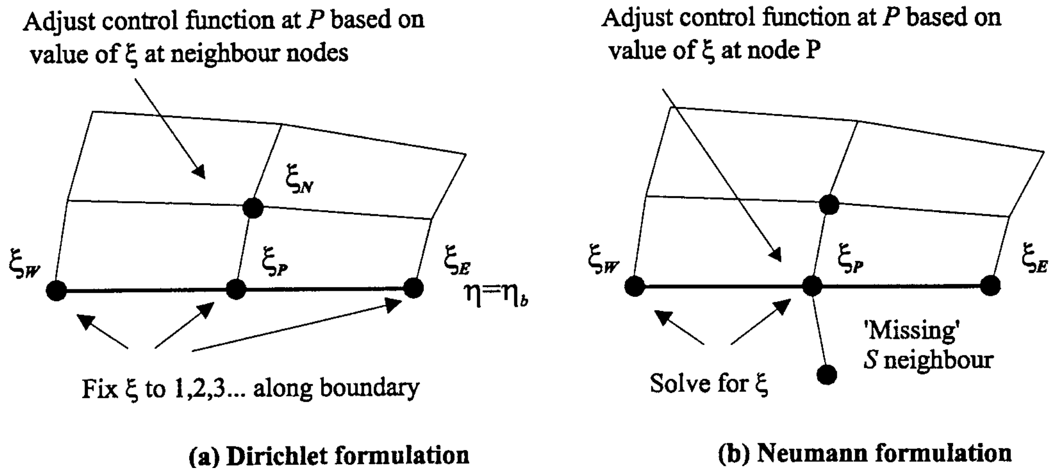
Adjust control function at P based on value of ξ at neighbour nodes
ξ_N
ξ_W
ξ_P
ξ_E
η=η_b
Fix ξ to 1,2,3... along boundary
(a) Dirichlet formulation
Adjust control function at P based on value of ξ at node P
ξ_W
ξ_P
ξ_E
'Missing' S neighbour
Solve for ξ
(b) Neumann formulation

Figure 3. Possible implementation schemes for 'automatic' control functions: (a) boundary ξ values fixed to desired values; (b) ξ solved at the boundary, and source term adjusted until ξ reaches desired value; no grid correction is applied at the wall.

The $[ii, i]$ terms are due to divergence [27] (dilatation), while the $[jj, i]$ $i \neq j$ are curvature terms, frequently introduced in efforts to control line spacing near walls [28–31]. A number of methods have been proposed for computing $[jj, i]$, usually based on finite difference approximations to the spatial derivatives of the metric tensor components

$$g_{ij} = \frac{\partial x^k}{\partial \xi^i}\frac{\partial x^k}{\partial \xi^j}, \tag{26}$$

or equivalent. In 2D, P and Q are thus obtained from the grid geometry at the boundary as

$$P = \frac{1}{\vec{r}_{,\xi}\cdot\vec{r}_{,\xi}}\left(+\frac{\vec{r}_{,\xi\xi}\cdot\vec{r}_{,\xi}}{\vec{r}_{,\xi}\cdot\vec{r}_{,\xi}} - \frac{\vec{r}_{,\eta\xi}\cdot\vec{r}_{,\eta}}{\vec{r}_{,\eta}\cdot\vec{r}_{,\eta}}\right), \tag{27}$$

and

$$Q = \frac{1}{\vec{r}_{,\eta}\cdot\vec{r}_{,\eta}}\left(+\frac{\vec{r}_{,\xi\eta}\cdot\vec{r}_{,\xi}}{\vec{r}_{,\xi}\cdot\vec{r}_{,\xi}} - \frac{\vec{r}_{,\eta\eta}\cdot\vec{r}_{,\eta}}{\vec{r}_{,\eta}\cdot\vec{r}_{,\eta}}\right). \tag{28}$$

This form is similar to that of Barron [32]. (*N.B.* the comma denotes the partial derivative, $\vec{r}_{,\xi} = \vec{e}_1$, $\vec{r}_{,\eta} = \vec{e}_2$.) In transformed space, boundary values of (x, y, z) corresponding to integer values of (ξ, η, ζ) are prescribed. The physical space analogue, as shown in Figure 3(a), is to fix (ξ, η, ζ) to integer values along the boundary. A related, though different method [26] involves the prescription of source terms corresponding to the 'missing' boundary terms. With reference to Figure 3(b), the value of the source at the south wall is set according to

$$S_P = S_P^* + a_{\mathrm{N}}(i - \xi_P), \tag{29}$$

where i is the desired value at P and S^* is the previous value of S. This procures rather than presumes orthogonality, and at the same time ensures $\xi_{\mathrm{P}} = i$, etc., in the fully converged state. For reasons discussed below, the Neumann implementation (Figure 3(b)) is preferred to the more usual Dirichlet formulation. This former approach can be implemented only by using the physical space formulation described in this paper.

A similar treatment is applied to ξ at the north boundary, η at the east and west boundaries, and ζ at the low and high ends. Interior source terms (control functions) are interpolated from the six sets of boundary values weighted according to the Jacobian $\sqrt{g}$.

2. EXAMPLES

Figure 4(a)–(d) shows four elementary O-grids. Figure 4(a) is a Laplace system corresponding to the mixed boundary value problem, with $\Delta\xi_{\text{ref}} = \Delta\eta_{\text{ref}} = 1$. In Figure 4(b), η cells have been concentrated. This was achieved equivalently in two different ways: (a) by using the transcendental control function defined in Equation (22), with $\eta_{\text{ref}} = 1, 2, 3, \ldots$, and (b) by prescribing η_{ref} as a set of non-integer values according to a geometric progression (i.e. no control function). Figure 4(c) shows a Dirichlet–Laplace system. Figure 4(d) demonstrates the use of automatic control functions according to Equation (29) to obtain boundary orthogonality. The same end may also be achieved by varying reference values. Figure 5 shows a 3D bend, where variable source terms are automatically prescribed with fixed values along all boundary lines, but surface values are allowed to slide.

Figure 6 shows a C-grid around an aircraft. In this case, only η was solved as a Dirichlet problem, and the ξ lines are simply the algebraically generated initial values. Figure 6(a) shows how divergence (distortion) will occur if the fixed boundary point distribution is not consistent with the choice of control function. In Figure 6(b), this distortion has been eliminated.

Figure 7 shows an H-grid over a 2D car body. The grid was allowed to slide at the sides and upper boundary, but was fixed at the lower wall. The ξ_{ref} values were set to ξ_{P} at $j = 1$. It can be seen from the inset that the exponential control function can be used to concentrate cells

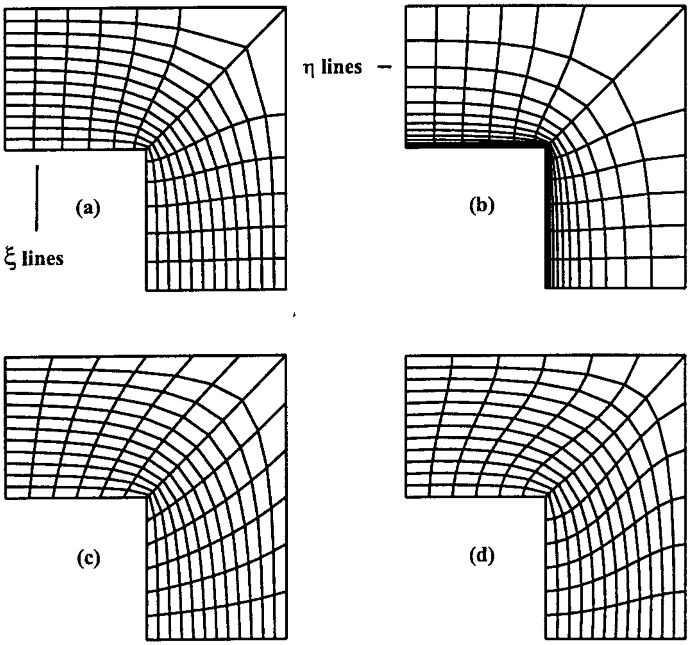

Figure 4. 2D bend: (a) Laplace system: mixed (Neumann) problem; (b) Poisson system; (c) Laplace system: Dirichlet problem; (d) use of 'automatic' control functions to effect boundary orthogonality.

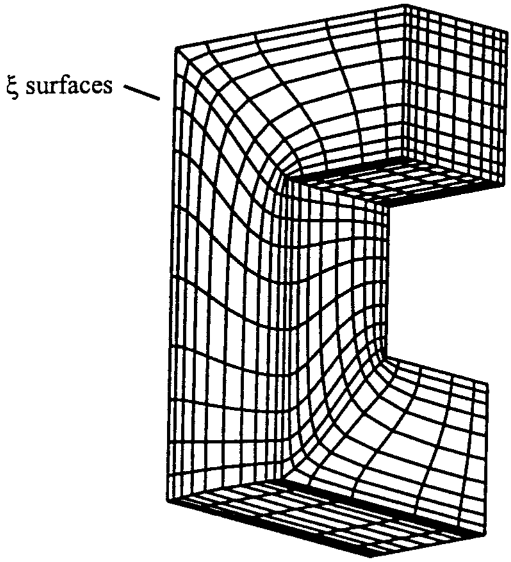

Figure 5. 3D bend with sliding (Neumann) boundary conditions on six boundary surfaces with control functions designed to procure orthogonality along 12 fixed boundary curves (lines).

highly without the occurrence of grid folding. Figure 8 is a similar 3D H-grid. In Figure 9, an initially folded O-grid has been unfolded and concentrated by using the method described in this paper.

3. DISCUSSION

The results demonstrate grid generation by means of the finite volume method and the grid correction procedure, Equation (12). The main differences between the present method and a conventional inverse scheme are highlighted in Table I. The following points are noteworthy:

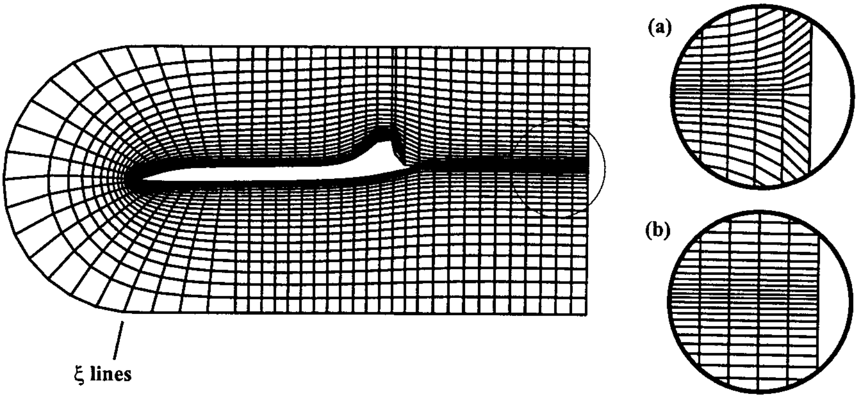

Figure 6. O-grid around an aircraft: ξ lines algebraic, η lines elliptic (Dirichlet problem). Insets: boundary point distribution (a) incompatible and (b) compatible with exponential control function (no divergence).

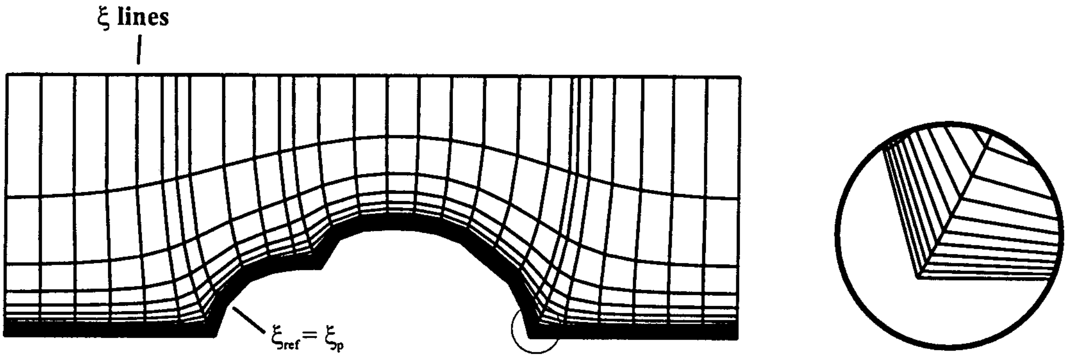

Figure 7. H-grid around a 2D vehicle with Neumann boundary conditions at sides; inset shows that no grid folding has occurred even when the grid is highly concentrated.

(1) the dependent variables are ξ, η, ζ, not x, y, z; (2) diffusion coefficients are based on a conservative discretisation, $(\sqrt{g}g^{11})_w$, $(\sqrt{g}g^{11})_e$, etc., not node-centred values g_P^{ii} (the same is true for the non-orthogonal terms); and (3) the source terms are simply $-\sqrt{g}P$, $-\sqrt{g}Q$, etc., and no cross-terms (such as η source in the ξ equation) are required, as they would be in an inverse scheme. Treated this way, ξ, η and ζ are independent, and it is possible to implement a segregated solver.

The introduction of control functions as source terms in the governing equations allows for effective control when these functions are prescribed to satisfy the *extremum* principle. There is no requirement to code control functions as diffusion source equations. Alternative forms exist, including variable Γ diffusion equations [33], convection–diffusion equations [34], etc. (Figure 7 was coded by means of a convection–diffusion formulation). The same is true for 'automatic' control functions, i.e. the 'velocities' $u = u^* + u'$ used in place of Equation (29). Another alternative to the use of control functions is to prescribe $\xi_{\rm ref}$ according to a bunching law, or to use current nodal values instead of integers. Under these circumstances, boundary

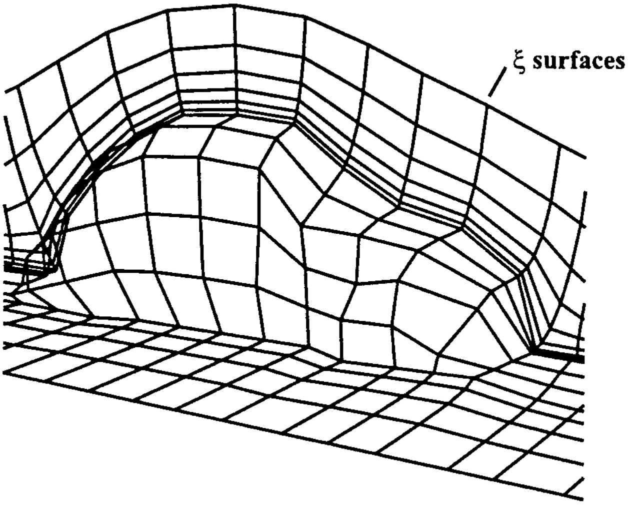

Figure 8. H-grid around a 3D vehicle.

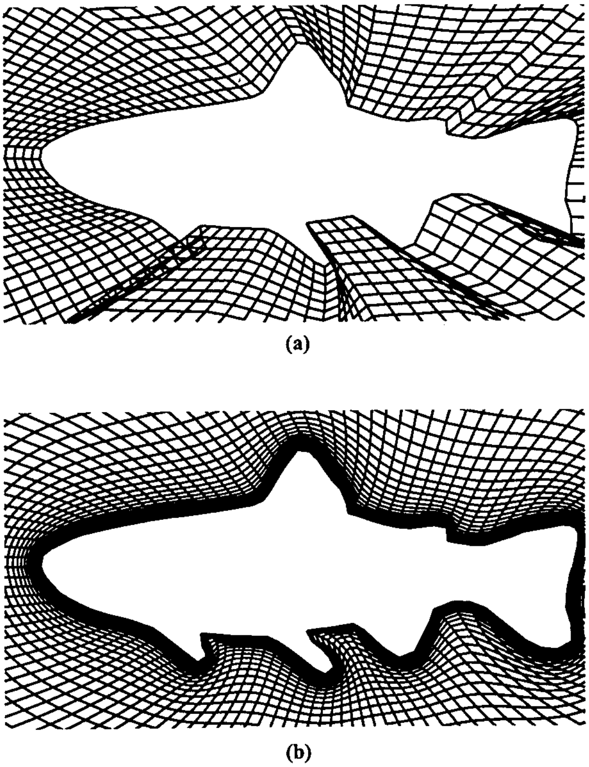

Figure 9. (a) Initially folded grid is (b) unfolded and concentrated with the method presented in this paper.

ξ values are not fixed in the linear algebraic equations. If the same ξ_{ref} is used at opposite boundaries (2D) or around an entire slab (3D), the grid will be parallel to the scalar field. Even if ξ, η ζ are not parallel to ξ_{ref}, η_{ref}, ζ_{ref}, the same results as those achieved by using variable source terms may be obtained in a rapid and stable manner.

The solution of mixed boundary value problems with sliding boundaries is repeatable, grid-independent and allows for effective grid control. The variables ξ, η, ζ are solved for, and may be controlled independently. Stability is seldom a problem. With Neumann conditions, the boundary (x, y, z) co-ordinates slide subject to, say, the constraint $\xi(x, y, z) = \text{constant}$, by locating the point on the ξ surface a minimum from $(x^* + x', y^* + y', z^* + z')$. No distinction need be made between surface and regular grid generation in Euclidean space. However, a general procedure for complex shapes is not trivial, and there may be constraints for the grid to pass through specific points. There will always be situations where the user is obliged to implement fixed boundary nodes. Initial grids then may be highly distorted because of the combination of grid bunching and trans-finite interpolation. Also, stability becomes a matter for concern. Because care was taken to generate proper functions satisfying the *extremum* principle, grid folding [11,12] was not a problem, even when highly concentrated grids were produced (Figures 7 and 8), and initially folded grids were unfolded (Figure 9). This operation was facilitated by imposing limits $1 \leq \xi_P \leq n$ on field variables, and ensuring that $\sqrt{g}$ was always positive. In the solution of Dirichlet problems, the choice of boundary point distribution and control functions must be made consistently, or local dilatation will occur near the boundary and control will be lost, as shown in Figure 6(a). For this example, only η was solved. The ξ values were generated by means of trans-finite interpolation, i.e. each variable may be treated independently. In many problems it is difficult to prescribe *a priori* boundary points consistent with the natural solution. A feature of the code is the facility to obtain results for scalars ξ, η, ζ in a fixed grid ξ^*, η^*, ζ^* by setting $\alpha = 0$ in Equation (12). These results provide an idea of where boundary points should be located.

Tests showed the new procedure is comparable in speed with inverse methods under most circumstances. Figure 10 shows the benchmark grids used to evaluate the performance of the code. The meshes shown in Figure 10(a) and (b) were generated by using both the present method and an inverse method, with linear coefficients, etc., prescribed according to Table I. 'Automatic' control functions were calculated on the basis of finite difference approximations to the metric coefficients, as given in Equations (27) and (28). The grids shown in Figure 10(c) and (d) were computed by using the present method only, according to Equation (29).

Figures 11 and 12 show monitor point data for the results of Figure 10. A Jacobi point-by-point solver and Spalding's strongly implicit whole-field solver [2,23] were used in

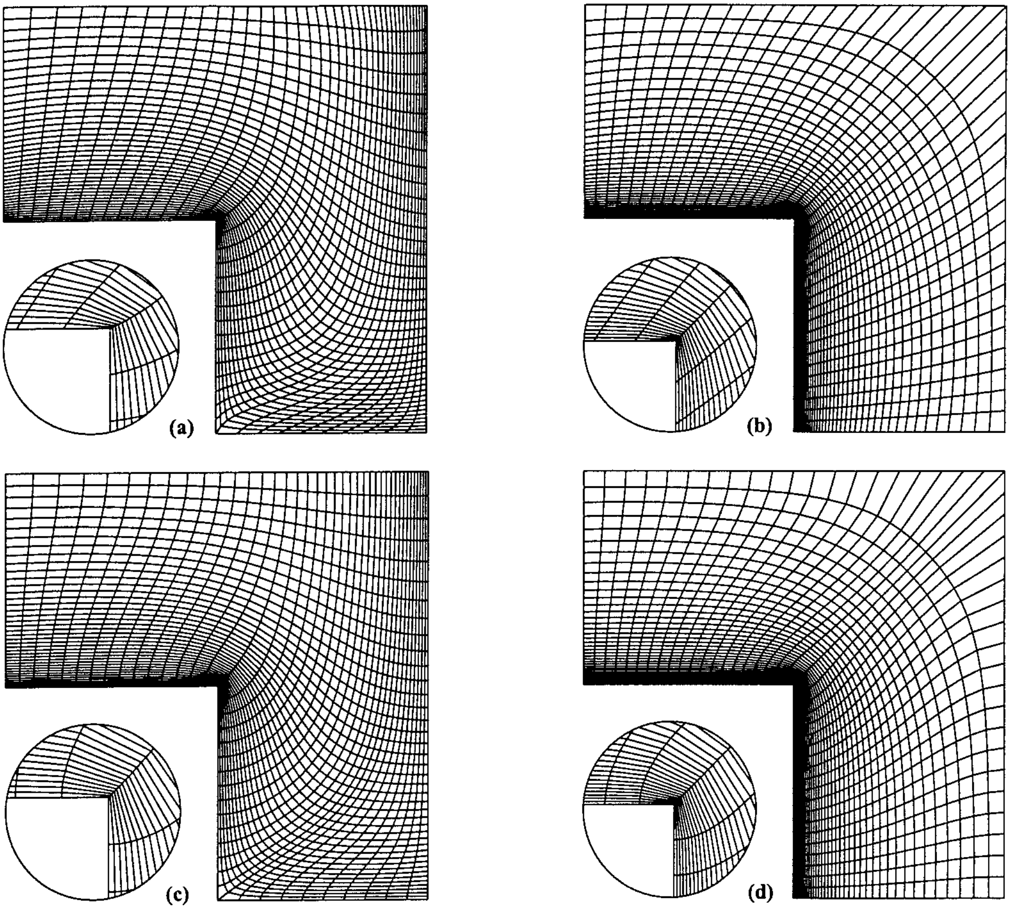

Figure 10. Benchmark problems: (a) and (b) H- and O-grids with 'automatic' control functions computed from metric coefficients according to Equations (27) and (28): identical grids were generated by using both present and inverse methods; (c) and (d): control functions prescribed from missing source terms according to Equation (29), present method only.

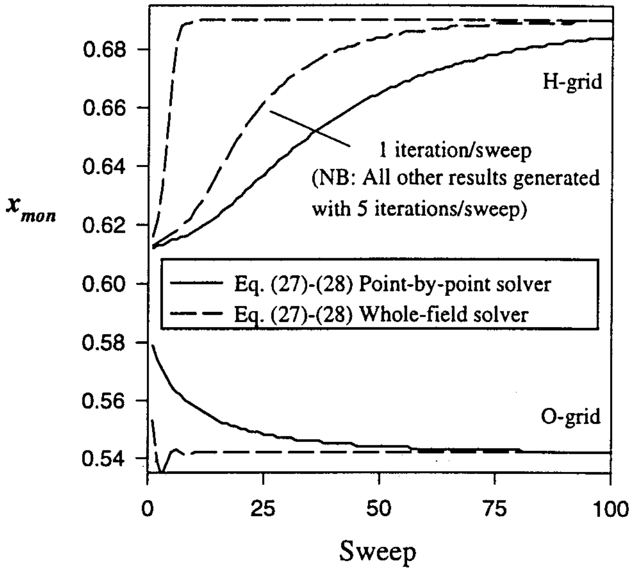

Figure 11. Monitor point data for benchmark problems: inverse method.

both cases. For the point-by-point procedure, the convergence history is nearly identical to the inverse scheme. With the whole-field solver, there appears to be a slight penalty associated with the present method for one of the two cases considered, probably because grid correction was being applied only at the end of each sweep (not at every iteration). The benefits of strongly implicit schemes in grid generation are the subject of debate. For example, Jordan and

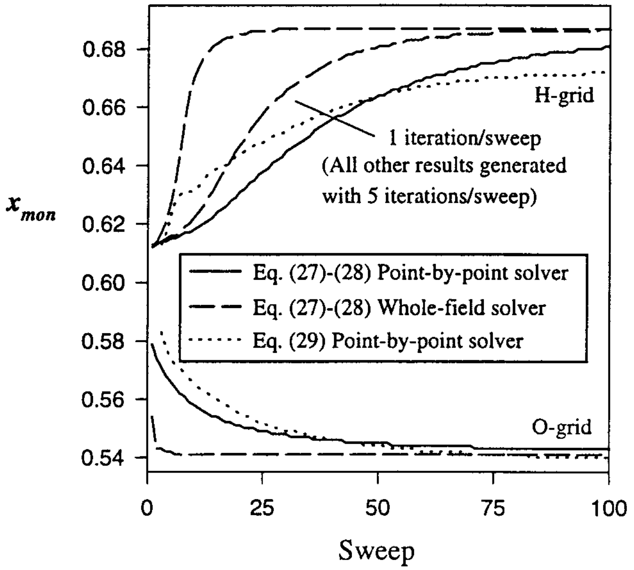

Figure 12. Monitor point data for benchmark problems: present method.

Spaulding [35] suggested this to be highly problem-specific. In fluid mechanics, however, such schemes have been used routinely for decades to solve Poisson equations (e.g. the pressure correction equation [1–3]). Here, the whole field scheme converged about ten times faster than the Jacobi procedure. For the present method, the $\partial x/\partial \xi$ terms needed for grid correction were obtained by means of a central difference scheme, Equations (14)–(16). Improvements could possibly be made by using a one-sided (upwind) difference scheme, or a higher-order non-linear scheme based on an analytical solution to the governing (e.g. diffusion source) equation.

'Automatic' control function schemes based on metric coefficients (Equations (27) and (28)) produced grids that were not orthogonal at the boundaries, as shown in Figure 10(a) and (b). The divergence terms [27] do circumvent major distortions at the boundary, as in Figure 6(a), and the addition of curvature terms may improve overall grid quality. However, boundary point control was found not to be precise. This was true whether the results were obtained by using the present method or a conventional inverse method.

When variable source terms (see Equation (29)) were used to produce boundary orthogonality, the equations were solved as mixed (Neumann) problems, with S eliminating the discrepancy between the Neumann and Dirichlet solutions. Technically, the ξ distribution is a function of changes in η and ζ; however, the effect is minor: only interior weightings, not end values of S, are affected. This method does effect local orthogonality (Figure 10(c) and (d)), but does not control the point distribution out from the wall, other than by way of influencing the end points. A second control function strategy is required to do this. Although the latter method was found to be applicable to highly concentrated grids, convergence was not obtained with the whole-field solver: it required use of the point-by-point procedure, with associated performance penalty. This effect appears to be due to fluctuations in ξ arising from the strong coupling between the source terms and nodal (values with Neumann boundary conditions). Attempts to impose Dirichlet conditions by fixing ξ_P to i at η_b, and prescribing the source term, S, from the ξ-values at the north node (see Figure 3(a)), or

$$S_P = S_P^* + a_{\mathrm{N}}(i - \xi_{\mathrm{N}}), \tag{30}$$

did not generate orthogonality. In fact, this procedure produced results similar to those obtained by using the metric coefficient method (see Equations (27) and (28)). This outcome suggests that Dirichlet-based boundary prescriptions may not be capable of generating orthogonality unless off-boundary nodes are somehow adjusted.

The present method requires additional memory for ξ, η, ζ (only one of which is required at any given time), when it is compared with a conventional code. Storage for x, y, z, metric coefficients, etc., is required here, as elsewhere. The current implementation was node-based, but may be adapted readily for cell-centred procedures. The same software may be used, because the same algorithm as the one for the flow solver is used. One can take advantage of available CFD codes with specific features like multi-block and fine grid embedding, octree and unstructured hexahedral elements, use of multi-grid acceleration, and built-in memory management techniques. Modification of existing codes to include grid correction is a relatively simple task, and many important problems in grid generation (user interface, domain decomposition, boundary condition prescription and automation) are common to CFD flow solvers. Modification of the method for solution-based grid adaptation [33] through the use of redistribution is clearly possible, as the scheme is inherently adaptive.

4. CONCLUSIONS

In previous grid generation methods, the governing equations are typically formulated in physical space, inverted analytically, then solved by means of a finite difference approximation on a uniform mesh in transformed space. In the present method, a numerical technique was developed by using the idea that both the formulation and the solution occur in physical space.

A particular numerical scheme based on the solution of Poisson's (diffusion source) equation, which may be considered a subset of the scalar transport equation, was used to generate grids by means of a vertex-centred finite volume method, which was combined with a grid-correction (re-zone) scheme. The system of linear equations was solved through the use of an explicit Jacobi point-by-point scheme as well as an implicit whole-field solver. Both fixed and sliding conditions for the boundary grid points were considered, as were transcendental and 'automatic' control functions. The use of sliding boundary conditions, when possible, allows the generation of highly concentrated grids without the occurrence of grid folding initially, although under many circumstances these were unfolded successfully with the present method.

Control functions that satisfy the requisite *extremum* condition from an appropriate 1D stretching function were derived by means of dimensional analysis. Automatic control functions computed from metric coefficients based on the presumption of local orthogonality, as well as functions designed to procure local orthogonality, were both implemented within the scheme. The latter could be employed successfully only with the point-by-point procedure. Grid control was also facilitated through the use of non-integer reference values for some 2D grids, with several distinct stability and speed of convergence advantages over control-function source terms. This technique may be extended readily to 3D meshes. In 3D, however, the grid and solution fields are no longer parallel.

The discretisation scheme described in this paper is conservative, inherently stable and versatile. It can replicate the state of the art in inverse methodologies, and also can be used to explore possibilities that cannot otherwise be achieved. No distinction need be made between surface and volume grid generation: everything occurs in Euclidean space. The control-function source terms are quite simple (see Table I). Scalars ξ, η, ζ may be solved (and controlled) one by one, i.e. independently. So-called 'physical' forms of partial differential equations, as solved in various CFD codes, may be employed, instead of the 'mathematical' form used currently. The present method can be modified easily for grid adaptation, and is in fact a 'self-adaptive' scheme. Any governing differential equations may be adopted; there is no need to invert analytically prior to solution. Solving for ξ, η, ζ in a fixed grid (no grid correction) gives the user an idea of where boundary points should be located. One can, in effect, view the mesh without creating it, a useful feature at the developmental stage and in problem solving. One works with the same algorithm as the one used in the flow solver, so that with the addition of a grid correction utility, existing CFD codes may be modified to do grid generation. This minimises redundancy in software libraries. Some disadvantages of the present approach include the fact that the linking coefficients and non-orthogonal source terms are more complex than in an inverse method. Slightly more memory may be required for the storage of one or more of ξ, η, ζ in addition to the usual grid geometry, etc. The rate of convergence of the present method is slightly below, or equal to, that of an inverse method for two benchmark problems when re-zoning by means of a central difference scheme. It is to be anticipated that with higher-order grid correction procedures, the present method may generate superior performance. The whole-field solver converged much more rapidly than the Jacobi point-by-point scheme, as might be expected for a Poisson system of equations.

The use of pseudo-Poisson equations to describe grid generation equations mathematically is somewhat arbitrary, in view of the non-linearity and interdependence of the control function source terms. Non-linear source terms degrade the performance of linear equation solvers significantly, and alternative formulations should be considered in the future. The governing grid generation equations need not be in the form of Equation (5) or even Equation (1). Any suitable partial differential equations may be adopted (parabolic, hyperbolic or elliptic). The present method may be employed to obtain a numerical solution for the desired grid, whether the governing equations are inverted analytically or not.

REFERENCES

1. S.V. Patankar and D.B. Spalding, 'A calculation procedure for heat, mass, and momentum transfer in three-dimensional parabolic flows', *Int. J. Heat Mass Transf.*, **15**, 1787–1806 (1972).
2. D.B. Spalding, 'Mathematical modelling of fluid mechanics, heat transfer and chemical reaction processes: a lecture course', *HTS/80/1*, Imperial College, London, 1980.
3. S.V. Patankar, *Numerical Heat Transfer*, Hemisphere, New York, 1980.
4. J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, *Numerical Grid Generation*, *Foundations and Applications*, Elsevier, New York, 1985.
5. P.M. Knupp and S. Steinberg, *Fundamentals of Grid Generation*, CRC Press, Boca Raton, FL, 1994.
6. J.F. Thompson, Z.U.A. Warsi and C.W. Mastin, 'Boundary fitted co-ordinate systems for numerical solution of partial differential equations—A review', *J. Comput. Phys.*, **47**, 1–108 (1982).
7. J.F. Thompson and N.P. Weatherill, 'Aspects of numerical grid generation: current science and art', *AIAA Paper 93-3539-CP*, 1993, pp. 1029–1070.
8. B.K. Soni and N.P. Weatherill, 'Geometry grid generation', in A.B. Tucker (ed.), *The Computer Science and Engineering Handbook*, CRC Press, Boca Raton, FL, 1996, pp. 791–819.
9. R. Aris, *Vectors*, *Tensors*, *and the Basic Equations of Fluid Mechanics*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
10. D.N. de G. Allen, 'Relaxation methods applied to conformal transformations', *Q. J. Mech. Appl. Math.*, **15**, 35–42 (1962).
11. A.A. Amsden and C.W. Hirt, 'A simple scheme for generating general curvilinear grids', *J. Comput. Phys.*, **11**, 348–359 (1973).
12. W.D. Barfield, 'An optimal mesh generator for Lagrangian hydrodynamic calculations in two space dimensions', *J. Comput. Phys.*, **6**, 417–429 (1970).
13. J.L. Steger and R.L. Sorensen, 'Use of hyperbolic partial differential equations to generate body fitted co-ordinates', in R.E. Smith, (ed.), *Numerical Grid Generation Techniques*, NASA-CP-2166, 1980, pp. 463–478.
14. J.L. Steger and D.S. Chausee, 'Generation of body fitted co-ordinates using hyperbolic partial differential equations', *SIAM J. Sci. Stat. Comput.*, **1**, 431–437 (1990).
15. K.N. Ghia, U. Ghia and C.J. Studerus, 'Analytical formulation of three-dimensional laminar viscous flow through turbine cascades using surface-oriented co-ordinates', *ASME Paper 76-FE-22*, *ASME Gas Turbines and Fluids Engineering Conference*, New Orleans, 1976.
16. U. Ghia and K.N. Ghia, 'Three-dimensional incompressible flow in straight polar ducts', *Comput. Fluids*, **5**, 205–218 (1977).
17. A.M. Winslow, 'Numerical solution of the quasi-linear Poisson equation in a non-uniform triangle mesh', *J. Comput. Phys.*, **2**, 149–172 (1967).
18. D.E. Potter and G.H. Tuttle, 'The construction of discrete orthogonal co-ordinates', *J. Comput. Phys.*, **13**, 483–501 (1973).
19. S.K. Godunov and G.P. Prokopov, 'The use of moving meshes in gas-dynamical computations', *Comput. Math. Math. Phys.*, **12**, 182–195 (1972).
20. J.F. Thompson, F.C. Thames and C.W. Mastin, 'Automatic numerical generation of a body-fitted curvilinear co-ordinate system for field containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **15**, 299–319 (1974).
21. Z.U.A. Warsi, 'Conservation form of the Navier–Stokes equations in general non-steady co-ordinates', *AIAA J.*, **19**, 240–242 (1981).
22. I.A. Demirdzic, 'A finite volume method for computation of fluid flow in complex geometries', *Ph.D. Thesis*, University of London, 1982.
23. S. Parameswaran, 'Finite volume equations for fluid flow based on non-orthogonal velocity projections', *Ph.D. Thesis*, University of London, 1985.
24. O.C. Zienkiewicz, *The Finite element Method*, McGraw-Hill, London, 1977.
25. J.F. Thompson, F.C. Thames and C.W. Mastin, 'TOMCAT—A code for numerical generation of boundary-fitted curvilinear co-ordinate systems on fields containing any number of arbitrary two-dimensional bodies', *J. Comput. Phys.*, **24**, 274–302 (1977).

26. S.B. Beale, ‘Choice of control functions for numerical grid generation’, *Proc. 4th Conf: CFD Society*, Canada, 1996, pp. 31–38.
27. P.D. Thomas and J.F. Middlecoff, ‘Direct control of the grid point distribution in meshes generated by elliptic equations’, *AIAA J.*, **18**, 652–656 (1980).
28. J.L. Steger and R.L. Sorenson, ‘Automatic mesh point clustering near a boundary in grid generation with elliptic partial differential equations’, *J. Comp. Phys.*, **33**, 405–410 (1979).
29. R.L. Sorenson, ‘Grid generation by elliptic partial differential equations for a tri-element augmentor-wing airfoil’, in J.F. Thompson (ed.), *Numerical Grid Generation*, North Holland, Amsterdam, 1982, p. 653.
30. J.F. Thompson, ‘A general three-dimensional elliptic grid generation system on a composite block structure’, *Comput. Methods Appl. Mech. Eng.*, **64**, 377–411 (1987).
31. B.K. Soni, ‘Elliptic grid generation system: control functions revisited-1’, *Appl. Math. Comput.*, **59**, 151–163 (1993).
32. R.M. Barron, ‘Improvements to grid quality and cost of multi-block structured grid generation’, *Proc. 4th Conf: CFD Society*, Canada, 1996, pp. 303–309.
33. A.M. Winslow, ‘Adaptive mesh zoning by the equipotential method’, *UCID-19062*, Lawrence Livermore National Laboratory, 1981.
34. S.B. Beale, ‘Numerical grid generation based on the solution of convection–diffusion source equations’, *Proc. 3rd Conf: CFD Society*, Canada, 1995, pp. 31–38.
35. S.A. Jordan and M.L. Spaulding, ‘A fast algorithm for grid generation’, *J. Comp. Phys.*, **104**, 118–128 (1993).