Numerical grid generation based on the solution of convection-diffusion-source equations

Beale, S. B.

Numerical Grid Generation Based on the Solution of Convection-diffusion-source Equations

by S.B. Beale

Paper presented at
CFD 95
3rd Annual Conference of the CFD Society of Canada
Banff, Canada, June 25-27, 1995, pp. 31-38

NUMERICAL GRID GENERATION BASED ON THE SOLUTION OF CONVECTION-DIFFUSION-SOURCE EQUATIONS

S.B. Beale
National Research Council of Canada
Ottawa, Ontario K1A 0R6

ABSTRACT

This paper describes a scheme to generate structured body-fitted grids. The equation solved is the standard scalar transport equation. The method is based on a non-inverse solution procedure for the transformation functions using an adaptive scheme, i.e. the grid is moved to conform to the desired solution. The technique may readily be applied using any standard CFD code based on the finite-volume method or control-volume finite-element method.

1. INTRODUCTION

Computational fluid dynamics (CFD) may be categorized as a 3-stage cycle: (1) Grid generation and pre-processing, (2) the main flow solver, and (3) graphics and post-processing. The cycle is typically repeated a number of times in obtaining a refined solution to a given flow problem. Grid generation is treated as a separate subject with a distinct suite of software. In reality, of course, it is an integral part of the overall solution procedure. The work described below is aimed towards integrating grid generation within the framework of the flow solver, the ultimate goal being to replace the 3-stage process with a single-stage process.

The dominant type of grid in use today is the structured body-fitted coordinate (BFC) grid composed of quadrilateral (2D) or hexahedral elements (3D) cells. The grid-generation process involves the definition of two or three functions, ξ^i, (also denoted by ξ, η, ζ) referred to as the contravariant displacement components[1]. These are considered a function of the Cartesian displacement components, x^i, also denoted by x, y, z, i.e. $\xi^i = \xi^i\left(x^1, x^2, x^3\right)$ $i = 1, 2, 3$. The relationships between (ξ^1, ξ^2, ξ^3) and (x^1, x^2, x^3) are often stipulated by means of differential equations,

$$\frac{D\left(\xi^1, \xi^2, \xi^3\right)}{D\left(x^1, x^2, x^3\right)} = 0 \qquad (1)$$

where the symbol D is used to denote some general differential operator. It is, however, the Cartesian coordinates of the grid-points, $x^i = x^i\left(\xi^1, \xi^2, \xi^3\right)$ which are actually required. Hence it is usually the so-called inverse form,

$$\frac{D^{-1}\left(x^1, x^2, x^3\right)}{D\left(\xi^1, \xi^2, \xi^3\right)} = 0 \qquad (2)$$

which is solved.

The conventional grid generation process involves the use of transfinite interpolation to obtain initial grid x^i values This is then followed by 'relaxation' or 'smoothing' based on the solution of (2). These latter schemes are frequently partially-converged solutions, owing to the fact that the combined choice of transformation functions, and boundary conditions do not render the desired mesh.

In this paper a single basic rule will be observed; namely that the D-operator must always be expressible in terms of vector functions. Under these circumstances, the choice of independent variables is inconsequential and (1) may be replaced by the coordinate independent form,

$$D(\phi) = 0\,, \qquad \phi = \xi^i \qquad (3)$$

Among the simplest and most widely-used grids are those based on Laplace's equation, which may be written in the operator form (3) as,

$$D(\phi) = \vec{\nabla} \cdot \vec{\nabla}(\phi) = 0 \qquad (4)$$

where $\phi = \xi^1, \xi^2, \xi^3$. The coordinate dependent form (1) is,

---

[1] The summation convention is assumed in §1 and §2: Any index which appears twice, once as a superscript and once as a subscript is summed. Superscripts in the denominator of fractions are considered as subscripts. In §3 and subsequently, the more conventional notation ξ, η, ζ and x, y, z are employed.

$$\frac{\partial^2 \xi^i}{\partial x^j \partial x^j} = 0 \qquad (5)$$

which may be inverted (2) as,

$$g^{jk} \frac{\partial^2 x^i}{\partial \xi^j \partial \xi^k} = 0 \qquad (6)$$

g^{jk} are the contravariant components of the metric tensor.

A shortcoming of the form of the transformation defined by (6) was noted by Thompson et al. [1]. Consider a simple rectilinear grid; when cells are concentrated in a geometric progression, $\Delta x \propto 1, s, s^2, \ldots$ the grid is distorted: The solution to the 1D Laplace equation (4) is a set of evenly-spaced lines. The numerical solution of (6) constitutes a set of constant ξ^i lines/surfaces, of fixed interval $\Delta\xi^i$, i.e. the fixed boundary-point distribution is incompatible with the function. This is unfortunate, since it is often important to be able to concentrate grid cells near walls (particularly for turbulent flows where the wall coordinates must be precisely defined). The geometric progression is a good method of achieving this.

There are two possible remedies (a) Generate a set of tangent lines ξ^i = constant, but not necessarily equispaced, $\Delta\xi^i \neq$ constant, or (b) Use a different set of transformation functions. Both options will be considered in this paper.

The next most popular set of equations used in grid generation are Poisson's equations, which may be written in the operator form (3) as,

$$D(\phi) = \vec{\nabla} \cdot \vec{\nabla}\phi - S = 0 \qquad (7)$$

where $S = S^i$ are 'control-functions' (also referred to as P, Q, R). A number of physical analogies, e.g. heat conduction with internal sources, can be constructed to provide a phenomenological basis for (7). The inverse form (2) is easily obtained. Thompson et al. [1] suggest that control-functions of the form $S = -a\,\mathrm{sign}(x - x_0)\, e^{-b|x-x|}$ may readily be used to effect surface attractions. (In this paper, these alone will be considered. Line and point attractions can always be effected by applying two or three surface attractions simultaneously.) The use of control functions is highly effective, especially when combined with sliding (Neumann) boundary conditions

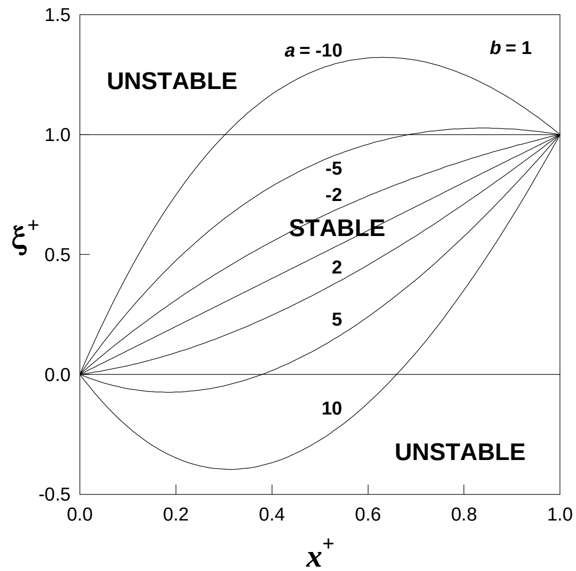

Figure 1 Solution to the 1D diffusion-source equation $d^2\xi^+/dx^{+2} = -ae^{-bx+}$ for $b = 1$ Values outside of the range $0 < \xi^+ < 1$ indicate interior grid points would lie outside the range of the boundary values, suggesting the solution to be unstable.

There are, however, disadvantages to the use of exponential control-functions: (i) The presence of two coefficients, a and b, demands a measure of skill on the part of the programmer in order to concentrate grid cells effectively. (ii) Convergence is by no means guaranteed; Figure 1 shows the solution for a 1D exponential[2] source-term in the range $x^+ \in (0,1)$. Values of ξ^+ outside (0,1) indicate the solution to be potentially unstable.

2. MODEL EQUATIONS

A function which is one-to-one over (0,1) is the normalized exponential function,

$$\xi^+ = \frac{1 - e^{Pe\, x^+}}{1 - e^{Pe}} \qquad (8)$$

Equation (8) is the solution to the 1D convection-diffusion equation, Patankar [2]. Pe is a Peclet number i.e. the ratio of convection to diffusion. This

[2] In order to write the problem over (0,1), x and ξ are non-dimensionalized according to $x^+ = (x - x_{min})/(x_{max} - x_{min})$ and $\xi^+ = (\xi - \xi_{min})/(\xi_{max} - \xi_{min})$.

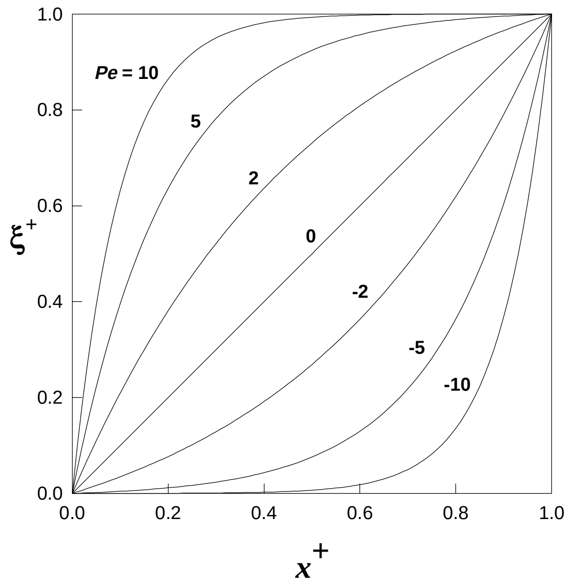

Figure 2 Solution to the 1D convection-diffusion equation $d^2\xi^+/dx^{+2}$-$Pe\xi^+$=0 showing the mapping always to be one-to-one.

exponential function generates a geometric series $\Delta\xi \propto 1, s, s^2,...$ where $s = e^{u_0 \Delta x}$ for Δx = constant.

The proposed model transformation equation $D(\phi) = 0$ is,

$$\frac{\partial \phi}{\partial t} + \vec{\nabla} \cdot (\vec{u}\phi) - \vec{\nabla} \cdot \vec{\nabla}\phi - S = 0 \qquad (9)$$

Equation (9) is the incompressible form of the well-known transport equation for a general scalar, ϕ (with $\Gamma = 1$). The terms $\vec{u} = \vec{u}^i$ and/or $S = S^i$ may be used to control the grid lines/surfaces by moving them in desired directions.

Equation (9) may be written in terms of general curvilinear coordinates (the so-called mathematical form) as,

$$\frac{\partial}{\partial t}\left(\sqrt{g}\phi\right) + \frac{\partial}{\partial \xi^i}\left(\sqrt{g}u^i\phi - \sqrt{g}\, g^{ij}\frac{\partial \phi}{\partial \xi^j}\right) - \sqrt{g}S = 0 \quad (10)$$

where the operation of differentiation $\partial/\partial\xi^i$, the metric coefficients, g^{ij}, and the Jacobian $\sqrt{g}$, all refer to the BFC system (i.e. the current grid configuration).

Henceforth in this paper the indicial notation will be dispensed with, in favour of the more conventional notation for the dependent variables:, $\xi^i = \xi, \eta, \zeta$, independent variables $x^i = x, y, z$ with the convection-fluxes being denoted by $\vec{u}^i = \vec{u}, \vec{v}, \vec{w}$, and the source terms by $S = P, Q, R$.

3. SOLUTION PROCEDURE

The set of equations (9) may readily be discretized to obtain linear algebraic equations $L(\phi) = 0$. These may be written in the form,

$$a_W(\phi_W - \phi_P) + a_E(\phi_E - \phi_P) + a_S(\phi_S - \phi_P) + a_N(\phi_N - \phi_P)$$
$$+ a_L(\phi_L - \phi_P) + a_H(\phi_H - \phi_P) + a_T(\phi_T - \phi_P) + S = 0$$
$$(11)$$

The subscripts W, E, S, N, L, H refer to the west, east, south, north, low, and high neighbours of P, respectively, while T refer to P-values obtained in the previous 'sweep' or 'time-step'[3] (i.e. grid configuration). The linking coefficients in (11) are evaluated by considering the combined influence of convection and diffusion. Methods of achieving this, such as the popular scheme of Patankar and Spalding [2], are well-known, and are therefore excluded from the paper.

The ϕ-distributions are independent of the choice of coordinate system (apart from numerical error). Thus ϕ may readily be solved-for, in a BFC grid which is an initial guess for the desired grid. Furthermore the BFC grid could be moving (adapting). The essence of the procedure is as follows: A number of inner 'iterations' are performed to obtain a solution to $L(\phi) = 0$. At the end of a 'sweep' the a-coefficients are re-computed based on the solution values. The key to this non-inverse procedure is that the grid itself is adapted, based on a comparison of the nodal values of ϕ with a set of reference values, ϕ_{ref}.

Grid correction procedure

Let it be assumed that an initial BFC grid has been generated and that the current Cartesian coordinates of point P are denoted by x^*, y^*, z^*. Suppose $\xi_{ref}, \eta_{ref}, \zeta_{ref}$ are reference values at point P in the grid: These could be integer values, corresponding to $\Delta\xi = \Delta\eta = \Delta\zeta = 1$, or they

[3] The transient term is retained: Even though ϕ is steady it is solved for in a grid which is non-inertial. The transient term may be regarded as real, or as an inertial relaxation factor.

could be arbitrary values based on the boundary-point distribution.

Owing to the fact that the grid locations at *P* differ from the ultimate values, *x, y, z,* values of $\phi = \xi$, η, ζ, will also differ from the desired reference values ξ_{ref}, η_{ref}, ζ_{ref}. After a number of iterations, displacement correction factors *x'*, *y'*, *z'* may be added to the previous values of x^*, y^*, z^*, as follows,

$$x = (1-\alpha)x^* + \alpha x' \qquad (12)$$

where α is a linear relaxation coefficient. The displacement-correction-factors are calculated from the $\partial\xi/\partial x$'s, at *P*, namely,

$$x' = (\xi_{ref} - \xi)\frac{\partial x^*}{\partial \xi} + (\eta_{ref} - \eta)\frac{\partial x^*}{\partial \eta} + (\zeta_{ref} - \zeta)\frac{\partial x^*}{\partial \zeta} \qquad (13)$$

with similar expression for *y* and *z*. This ensures that the nodal values ξ, η, ζ converge on the reference values.

Boundary conditions

Boundary conditions are prescribed as linear source terms $S = C(\phi_{ref} - \phi_P)$ in the finite-volume equations. Two conditions are common (a) Fixed values (Dirichlet) and (b) sliding or zero gradient (Neumann) boundary conditions. Both may readily be implemented.

All grids require two fixed values in each direction, for instance ξ may be fixed at the *E* and *W* faces, η at the *S* and *N* faces, and ζ at the *L* and *H* faces. Values of ξ at the *S, N, L, H*, η at the *E, W, L, H*, ζ at the *E, W, S, N* surfaces may either be fixed or be allowed to slide, There is no problem in prescribing either of these conditions; the Neumann condition is the default (*S*=0), while in-cell values of ϕ may be fixed to ϕ_{ref}, by means of a suitably large coefficient, *C*.

The choice of reference values for ξ, η, ζ may either be fixed to (a) values such 1, 2, 3,...*nx*, corresponding to $\Delta\xi$ = constant (b) other non-integer values, e.g. a geometric progression, or (c) actual in-cell values at a boundary or interior location, obtained as part of the procedure.

Choice of control-functions

Grid control may be effected either by means of source or convection terms. The latter were employed here. The function grad ϕ is vector perpendicular to the ϕ–surface at *P*. This suggests the $\vec{u}$ -function could be defined by,

$$\vec{u} = u_0 \vec{\nabla}\xi \qquad (14)$$

with similar terms in η, ζ for $\vec{v}$ and $\vec{w}$.

Summary of the procedure

The grid generation procedure may be summarized as follows.

(1) Transfinite interpolation is used to generate an initial grid.

(2) The metric coefficients are obtained from the grid geometry, and the coefficients in the linear algebraic equations are calculated.

(3) Field values of ξ, η, ζ are set to initial values, ξ_{ref}, η_{ref}, ζ_{ref}.

(4) A few iterations of the elliptic solver are applied to obtain current values for $\phi = \xi, \eta, \zeta$.

(5) Values of ξ, η, ζ are then compared to the desired values ξ_{ref}, η_{ref}, ζ_{ref} the $\partial\xi/\partial x$'s and grid correction factors *x'*, *y'*, *z'* calculated and added to x^*, y^*, z^*.

Steps (2)-(5) are then reiterated until $\xi = \xi_{ref}$, $\eta = \eta_{ref}$, $\zeta = \zeta_{ref}$.

4. ILLUSTRATIVE EXAMPLES

Some simple 2D grids are used to illustrate the method described in this paper. Figures 3-6 show a 31x16 O-grid around a NACA 012 airfoil. The radial lines are of constant ξ while the circumferential lines are of constant η. ξ has been fixed to 1 and *nx* at $\theta = 0$ and π, while η is fixed at the inner-airfoil and the outer circular sections, respectively.

Figure 3 shows the solution for Laplace's equation, with Dirichlet boundary conditions being applied at all boundaries, i.e. the ξ-boundary-points and the η-boundary-points are fixed, with ξ_{ref} = 1, 2, 3,... at 0, $\Delta\theta$, $2\Delta\theta$,.. π where $\Delta\theta = \pi/(nx-1)$ and η_{ref} = 1, 2, 3,... at r_0, $r_0+\Delta r$, $r_0+2\Delta r$...etc.

Figure 4 shows that by prescribing η_{ref} = 1, 2, 3,... at points which are themselves

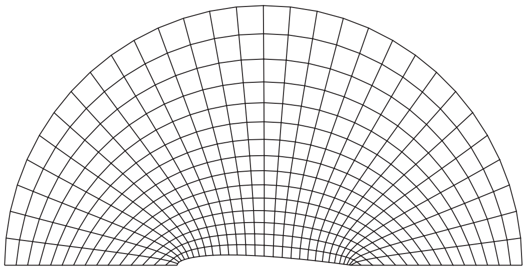

Figure 3 NACA 012 airfoil, Dirichlet boundaries

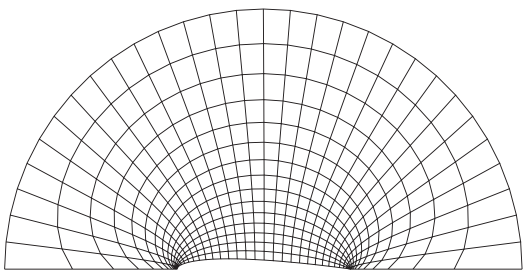

Figure 4 NACA 012 airfoil, Dirichlet boundaries

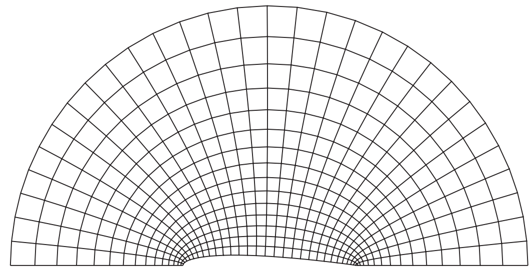

Figure 5 NACA 012 airfoil, Neumann boundaries

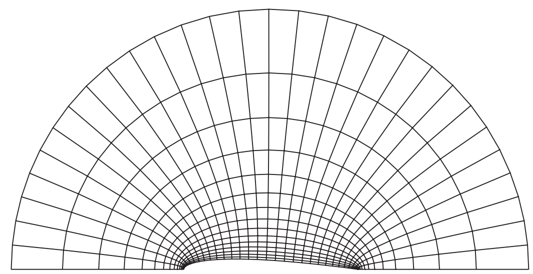

Figure 6 NACA 012 airfoil, Neumann boundaries

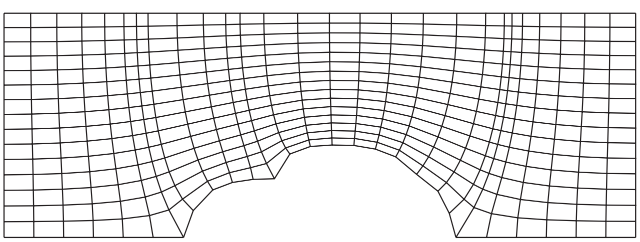

Figure 7 Vehicle body, Laplace's equation (v_0=0)

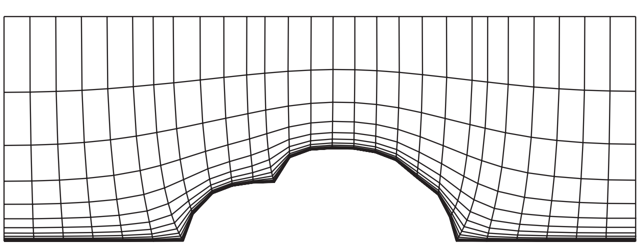

Figure 8 Vehicle body, v_0=-0.3

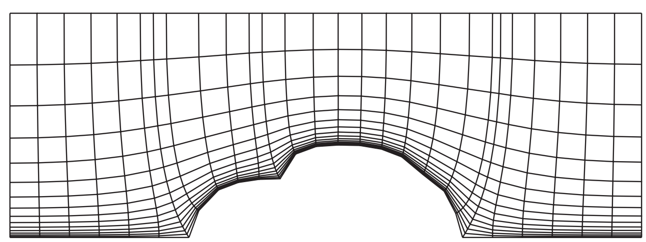

Figure 9 Vehicle body, v_0=-0.5

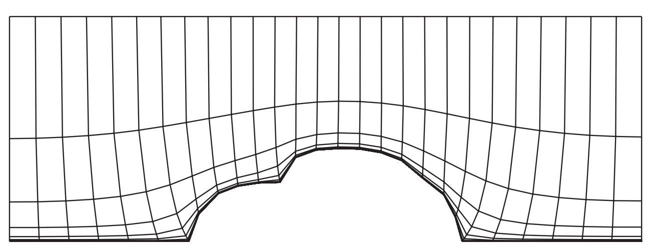

Figure 10 Vehicle body, v_0=--1.0

concentrated in a geometric progression $\Delta r \propto$ 1, s, s^2,...the grid lines may be concentrated at the front and rear edges of the airfoil. There is, however little improvement to the distribution elsewhere, e.g. along the airfoil section.

Figure 5 shows the solution for Laplace's equation, this time with Neumann boundary conditions being applied, and $\Delta\xi_{ref} = \Delta\eta_{ref} = 1$. Grid-nodes are allowed to slide until $\xi = \xi_{ref}$, and $\eta = \eta_{ref}$. The grid is now orthogonal.

Figure 6 shows that by applying an inward 'convection-flux' according to (14), effective grid control of the η-lines is possible. It can be seen that the grid-cells are concentrated in the airfoil boundary-layer in a desirable fashion. Grid-cells may also be concentrated at the leading and trailing edges by use of a similar function in the ξ equation.

Further use of convection-functions is illustrated in Figs. 7-10. These show an H-grid around a 2D vehicle. Figure 7 shows the solution to Laplace's equation ($v_0 = 0$). The grid lines are coincident with stream-lines and parallel to iso-potentials for streaming potential-flow. Values of the η-lines (stream-lines) were allowed to slide at both the west and east boundaries with $\eta_{ref} =1, 2, 3,...$ as above. ξ–values (parallel to the iso-potentials) were allowed to slide at the north boundary, but were fixed at the south wall. This was achieved by setting the ξ_{ref} values to the current nodal values at the south boundary corresponding to $j = 1$ (i.e. the car surface). It can be seen that the η-lines diverge at the front and rear stagnation points, a problem typical of H-grids.

Figures 8-10 show the effects on the η-lines of progressively increasing the magnitude of the convection-flux term, v_0.

5. DISCUSSION

These results and others demonstrate that it is possible to generate grids in a stable and efficient manner, by means of the solution of the general scalar equation (9) combined with the grid-correction procedure (13).

Inspection of Figs. 8-10 reveals that convective-terms may be used to control grid-lines in a stable and predictable fashion. The Laplace-type grid shown in Fig. 7 would render a poor flow-field solution owing to the coarse nature of the horizontal η-lines in the boundary-layer around the vehicle. As the magnitude of v_0 is increased (Figs. 8-9) the η-lines become concentrated in the boundary layer around the vehicle. Figure 10 shows that the solution is highly stable even when the grid-lines are almost all concentrated in the boundary layer. The technique may readily be used to pack grid-lines close to internal and external steps (discontinuities) where Laplace-grids render poor resolution. Inspection of Figs. 7-10 reveals the vertical ξ-lines to be unaffected by the values of v_0: Because ξ and η are solved for independently, they may be controlled in an independent fashion.

Boundary conditions can be applied in a number of different ways. In 2D three distinct scenarios may be identified: Consider, say the ξ-lines, and let it be assumed that they are fixed at the *E* and *W* boundaries. (a) ξ-points can slide at both the *N* and *S* boundaries (Neumann boundary conditions) see Figs. 5-6. (b) ξ-points are fixed at say the *S* boundary but remain free to slide at the *N* boundary, see Figs. 7-10. (c) ξ-points are fixed at both *N* and *S* boundaries, see Figs. 3-4. These cases are discussed further below.

Use of convection-functions for grid-control is best combined with (Neumann) boundary conditions: If grid-lines can slide freely at both ends (Figs. 5-6) iso-ξ lines corresponding to $\xi_{ref} = 1,2,3...$ (or other suitable values) may be generated. Under these circumstances, control-functions are effectively used to concentrate grid-lines by manipulating the actual ξ-functions. An alternative procedure (not shown) is to simply prescribe ξ_{ref} as non-integer values, according to an appropriate bunching law; Rather than change the form of the ξ function, create a grid which is parallel to ξ-function, but of arbitrary distribution.

If the grid is fixed at one end, but free at the other (vertical lines, Figs. 7-10) iso-ξ lines may be generated but with arbitrary $\Delta\xi$'s. Control-functions of the form (14) cannot readily be used to concentrate grid-lines. Under these circumstances ξ_{ref} would be set to the current in-cell value of ξ, at the fixed boundary. Note that boundary ξ-values need not be fixed in the linear algebraic equations (11). The *x*-locations of the free-end slide to the appropriate position corresponding to constant $\xi = \xi_{ref}$. Convection functions suitable for altering the shape of the ξ -distribution would therefore have to have different values at either end.

If the grid is fixed at both ends, iso-ξ values may still be generated, although the ξ-function will not, in general, be the same as for the Neumann problem (compare Figs. 3 and 5). Under these circumstances the in-cell values of ξ must be fixed at one or more boundaries, in order that a single consistent ξ_{ref} be employed along a given line. In the event that these reference values are chosen as 1, 2, 3,..*nx* (Fig. 3) the solution obtained is that which would arise from a conventional inverse solution procedure (2). Convection-terms could be used to move the grid-lines around in a gross fashion, but probably not as a means to concentrate grid-lines/surfaces. In fact it is only necessary to fix in-cell values of ξ at one end, in the finite-volume equations (i.e. to the values ξ_{ref} at the other end), Under these circumstances at least one boundary would be orthogonal.

In the context of the Dirichlet problem, control-functions are normally designed to satisfy the condition of orthogonality at all boundaries [1]. Time prevented a convection-flux-based formulation from being implemented in time for inclusion in this paper, however the basis for one possible procedure will be briefly discussed: Consider neighbour points *E*, *P*, *W*, say, along one of the boundaries and let $x^+ = (x_\mathrm{P} - x_\mathrm{E})/(x_\mathrm{W} - x_\mathrm{E})$. If $x^+ \neq$ ½, Laplace's equation cannot be satisfied. It is however possible to compute the Peclet number, *Pe*, which will generate $\xi^+ =$ ½ corresponding to any value of x^+ (Fig. 2). Thus a set of convection fluxes can be added on a cell-by-cell basis which will correspond to the solution of the problem. Interior values for the convection-fluxes may be obtained by interpolation from the boundary value(s).

It would have been preferable if constant *u*-values could have been used in place of (14). Inspection of Figure 2 reveals the reason for the choice (14): While the exponential function (8) generates a set of ξ-values corresponding to a geometric series $\Delta\xi \propto 1,\ s,\ s^2,...$ if $\Delta x =$ constant, the converse is not true. The function which can be used to generate a geometric-series $\Delta x \propto 1,\ s,\ s^2 ...$ for $\Delta\xi$□□= constant, is the inverse-logarithmic-function, which satisfies a diffusion-source equation (7) provided,

$$S = u_0\left(\vec{\nabla}\xi \cdot \vec{\nabla}\xi\right) \qquad (15)$$

Values of u_0 are such that *Pe* does not normally exceed unity. Thus a central-difference scheme can and should be employed (of course other schemes [2] may readily be employed if they default to this limit). Since a central-difference scheme is used, whether the control-functions are coded as convection-terms (14) or as source-terms (15) is largely a matter of preference. (15) would normally imply a node-based difference-scheme i.e. non-conservative. The form (14) implies a conservative formulation with $\vec{u}$-values being interpreted as inter-nodal (staggered) quantities. The covariant values are easily computed from neighbour values $u_w = (\xi_\mathrm{W} - \xi_\mathrm{P})/(x_\mathrm{W} - x_\mathrm{P})$ etc. and may thus be incorporated into the scheme in a fashion similar to the diffusion terms. Some possible variations of (14) may be generated based on the use of the divergence vs. convective forms of (9), in the event $\nabla \cdot \vec{u} \neq 0$. Formulation (14) renders a $\vec{u}$ vector which is proportional to the contravariant basis vectors, in the fully-converged state, i.e. the convection flux is being applied in a direction perpendicular to the iso-ξ values. (The notation $\vec{u} = u_0 \vec{e}^{\,1}$ is avoided, since it constitutes a grid-dependent formulation.)

When implementing Neumann (sliding) boundary conditions, the grid correction procedure must be modified to allow the (*x*,*y*,*z*) coordinates at the boundary locations to slide subject to the geometrical constraint, $\xi(x,y,z) =$ constant. This was achieved by locating the point on the ξ-surface which is a minimum distance from the point (x^*+x', y^*+y', z^*+z'). Because of the explicit nature of the grid-correction procedure, it is actually easier to implement Neumann conditions here than would be the case if an inverse procedure were employed. While all necessary surface geometry-information must be available in order to generate the initial grid, in practice the development of a general-purpose boundary-point-sliding procedure suitable for application to complex shapes is not trivial. Moreover there may be additional constraints requiring the grid to pass through specific points such as the leading and trailing edges of the car body in Figs. 7-10. Much can be achieved by dividing the domain into sub-domains or zones, however there will probably always be situations where the user is obliged to implement Dirichlet boundary conditions. A feature of the method above is that, because it is not an inverse procedure, field values $\phi = \xi, \eta, \zeta$ are available. Thus it is possible to look at the solution to the Neumann problem, by solving (11) in the fixed initial grid configuration, without ever implementing the grid correction procedure. Plots of iso-ϕ values can give the user a good idea of where the boundary points should be located.

At first sight it might seem that the non-inverse procedure is not efficient due to the presence of an additional outer grid-correction loop. However if a fully-converged solution for ξ, η, ζ is obtained, grid adaptation results in only minor changes to the solution, provided field-values are re-interpolated. At the other end of the scale, by appropriate choice of the false time-step factor, a_T, in (11), it is possible to decrease the number of iterations to 1, i.e. convert the implicit procedure to a fully-explicit scheme, entirely eliminating the inner loop. Under these circumstance the linear relaxation factor in the grid correction procedure (13) can normally be set to unity, $\alpha = 1$. In general, convergence was found to be most rapid when just a few (around 5) iterations were performed per sweep.

It is true that the non-inverse procedure described in this paper requires additional memory allocation in comparison to inverse schemes, owing to the storage requirements for ξ, η, ζ. However these memory-requirements are still less than those required in most CFD codes, where the momentum equations are solved. (CFD codes often have sophisticated memory-management schemes built in.) A significant advantage of the non-inverse grid-generation procedure is that since it uses precisely the same algorithm as conventional flow solvers, the researcher is able to use the same computer program to generate the grid as is used to solve the flow-field (subject to the addition of the grid-correction procedure, Eq. 13). The method described in this paper may be directly applied to so-called control-volume finite-element codes and, with a few alterations, to conventional cell-centred finite-volume procedures. Several general-purpose CFD codes are in use today with the specific internal features. These may readily be exploited by the grid generation procedure: For example, codes designed to handle multi-block and fine-embedded grids could easily be modified to generate the same. Multi-grid solvers could be usefully exploited, and advantage could be taken of existing memory management techniques.

Solution-adaptive grids may be generated, with ξ, η, ζ being solved for at the same time as p, u, v, w. i.e. conduct grid generation and the flow-field solution concurrently. (Such techniques are popular for compressible flow problems involving the use of unstructured grids.) The use of convection-functions (14) proportional to grad ϕ is but one example of grid adaptation, these or other functions could be attenuated or filtered according to other criteria such as the field-values of pressure gradient, entropy etc., as the solution proceeds. Grid-lines could be also be added as necessary. The grid may thus be concentrated towards shock-fronts, and other regions of interest.

Although the illustrative examples given in this paper were simple 2D problems, all of the concepts discussed above may be extended, without difficulty, to 3D, and the same methods applied to complex shapes and geometries.

6. CONCLUSIONS AND RECOMMENDATIONS

A grid generation scheme based on the solution of the scalar transport equation was described. It was shown that by implementing a grid-adaptation procedure grids could be generated using a non-inverse scheme, with both fixed and sliding boundary conditions.. The scheme was found to be quite stable. The practical use of convection-based control functions was demonstrated It was shown that the use of sliding (Neumann) boundary conditions combined with gradient-oriented convection-functions may be used to generate effective grids.

Future research should focus upon the synthesis of generalized procedures for sliding boundary values along arbitrary surfaces, suitable methods for controlling the grid-lines when fixed (Dirichlet) boundaries are imposed, and solution-based grid adaptation procedures.

7. ACKNOWLEDGMENTS

The author wishes to thank S. Bohme and G. Wood for their assistance in preparing the material presented in this paper.

REFERENCES

1. Thompson, J.F., Warsi, Z.U.A., and Mastin, C.W. Numerical Grid Generation, Foundations and Applications, Elsevier, New York (1985).

2. Patankar S.V. Numerical Heat Transfer and Fluid Flow. Hemisphere Publishing Corporation. Washington (1980).