Comparison of direct and inverse grid generation schemes

Beale, Steven

COMPARISON OF DIRECT AND INVERSE GRID GENERATION SCHEMES

S.B. Beale
National Research Council
Ottawa, Ontario K1A 0R6
steven.beale@nrc.ca

ABSTRACT

A comparison is made of a grid generation scheme based on an inverse method with one based on a direct-solution, using the same computer code. Two test cases involving H-type and O-type grids, with fixed boundary conditions, are considered. Calculations are performed using an explicit point-by-point and a strongly-implicit whole-field solution procedure with control-functions. The results of the study are critically appraised with respect to grid quality, convergence, and other performance measures.

1. INTRODUCTION

The majority of algorithms used for body-fitted grid-generation are based on the application of elliptic smoothing techniques. Diffusion equations [1] have the desirable property of smoothing discontinuities generated using algebraic techniques, and diffusion-source equations [2,3] are often used for additional control. The governing equations for ξ^i or (ξ,η) are typically formulated in the form,

$$\text{div grad } \xi^i + s^i = 0 \qquad (1)$$

and then inverted to obtain expressions for x^i or (x,y),

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

The source terms may be control-functions, $s^i = -P^i = -(P,Q)$, or other devices. Beale [4] demonstrated that it is possible to directly solve Eq. (1) for (ξ,η) in a curvilinear co-ordinate system, $\xi^* = (\xi^*,\eta^*)$,

$$\frac{\partial}{\partial\xi^{*j}}\left(\sqrt{g}\,g^{jk}\frac{\partial\xi^i}{\partial\xi^{*k}}\right) + \sqrt{g}\,s^i = 0 \qquad (3)$$

and re-mesh the grid based on the solution. Because this methodology involves an additional round of iteration for re-meshing; it might be anticipated that a major performance reduction would result. In this paper an evaluation is made of the relative performance of direct and inverse methods. The same computer code, using identical numerical methods was used to solve linear algebraic equations having the form,

$$\phi_P = \frac{a_W\phi_W + a_E\phi_E + a_S\phi_S + a_N\phi_N + a_T\phi_T + CV + S}{a_W + a_E + a_S + a_N + C} \qquad (4)$$

ϕ_W, ϕ_E, ϕ_S, ϕ_N are neighbour values, a_W, a_E, a_S, a_N are linking coefficients, a_T is an inertial relaxation factor, ϕ_T is the previous value of ϕ_P, C and V are linearized-source-term coefficient and value (often used to prescribe boundary conditions), and S is a source-term, composed of terms due to grid non-orthogonality and control-functions,

$$S = S_{nonort} + S_{control} \qquad (5)$$

2. INVERSE METHOD

The dependent variables are the Cartesian components of the grid nodes at P,

$$\phi = (x,y) \qquad (6)$$

The linking coefficients are,

$$\begin{aligned} a_E &= a_W = \left(g^{11}\right)_P \\ a_N &= a_S = \left(g^{22}\right)_P \end{aligned} \qquad (7)$$

The built-in source terms are,

$$S = 2g_P^{12}\left(\phi_{en} + \phi_{ws} - \phi_{es} - \phi_{wn}\right) \qquad (8)$$

where

$$\phi_{en} = \frac{\phi_P + \phi_E + \phi_N + \phi_{EN}}{4} \qquad (9)$$

etc., as illustrated in Fig. 1.

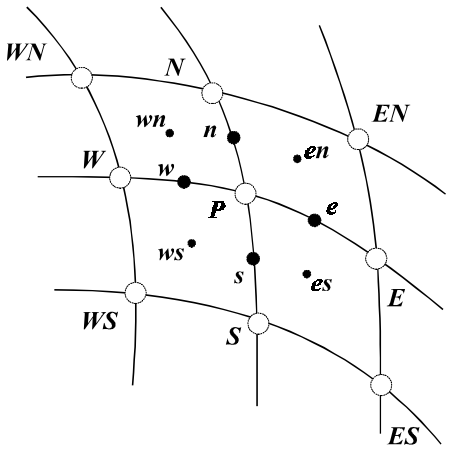

Figure 1. Compass notation.

The control-function source terms are evaluated using,

$$S = -\Big[\max(0,P)_P(\phi_E - \phi_P) + \min(0,P)_P(\phi_P - \phi_W) + \max(0,Q)_P(\phi_N - \phi_P) + \min(0,Q)_P(\phi_P - \phi_S)\Big] \quad (10)$$

where P and Q are control-functions.

3. DIRECT METHOD

The dependent variables are the scalars,

$$\phi = (\xi, \eta) \quad (11)$$

The diffusion coefficients are evaluated inter-nodally,

$$a_W = \left(\sqrt{g}g^{11}\right)_w,\ a_E = \left(\sqrt{g}g^{11}\right)_e \qquad a_S = \left(\sqrt{g}g^{22}\right)_s,\ a_N = \left(\sqrt{g}g^{22}\right)_n \quad (12)$$

The non-orthogonal source terms are,

$$S = \left(\sqrt{g}g^{12}\right)_e(\phi_{en} - \phi_{es}) - \left(\sqrt{g}g^{12}\right)_w(\phi_{wn} - \phi_{ws}) + \left(\sqrt{g}g^{12}\right)_n(\phi_{en} - \phi_{wn}) - \left(\sqrt{g}g^{12}\right)_s(\phi_{es} - \phi_{ws}) \quad (13)$$

The control-function source terms are simply,

$$S = -\begin{cases} \left(\sqrt{g}P\right)_P & \phi = \xi \\ \left(\sqrt{g}Q\right)_P & \phi = \eta \end{cases} \quad (14)$$

In the direct method, grid-correction is applied, using a central difference scheme,

$$\mathbf{r}_P = \mathbf{r}_P^* + \alpha\left((i - \xi_P)\frac{\mathbf{r}_E - \mathbf{r}_W}{2} + (j - \eta_P)\frac{\mathbf{r}_N - \mathbf{r}_S}{2}\right) \quad (15)$$

where $\mathbf{r}_P = (x_P, y_P)$, and α is a linear relaxation factor.

4. CONTROL FUNCTIONS

Control-functions are often introduced as source terms and are typically formulated to (1) concentrate cells in boundary layers and/or (2) effect orthogonality at grid boundaries.

4.1 Control-functions from grid geometry

A number of methods have been proposed [5,6,7,8,9]. Rearranging Eq. (2);

$$P^i = g^{jk}\begin{Bmatrix} i \\ jk \end{Bmatrix} = g^{il}g^{jk}[jk,l] \quad (16)$$

where the Christoffel symbols are,

$$[jk,l] = \frac{\partial \mathbf{e}_j}{\partial x^k}\cdot\mathbf{e}_l = \frac{1}{2}\left(\frac{\partial g_{jl}}{\partial \xi^k} + \frac{\partial g_{kl}}{\partial \xi^l} - \frac{\partial g_{jk}}{\partial \xi^l}\right) \quad (17)$$

Existing methods assume local orthogonality[1], $g^{ii} = 1/g_{ii}$ (unsummed), and $g^{ij} = g_{ij} = 0,\ i \neq j$. Thus $P^i = [jj,i]/g_{ii}g_{jj}$ (i free, j summed), which may be computed from Eq. (17) as,

$$P = \frac{1}{\mathbf{r}_{,\xi}\cdot\mathbf{r}_{,\xi}}\left(+\frac{\mathbf{r}_{,\xi\xi}\cdot\mathbf{r}_{,\xi}}{\mathbf{r}_{,\xi}\cdot\mathbf{r}_{,\xi}} - \frac{\mathbf{r}_{,\eta\xi}\cdot\mathbf{r}_{,\eta}}{\mathbf{r}_{,\eta}\cdot\mathbf{r}_{,\eta}}\right)$$
$$Q = \frac{1}{\mathbf{r}_{,\eta}\cdot\mathbf{r}_{,\eta}}\left(-\frac{\mathbf{r}_{,\xi\eta}\cdot\mathbf{r}_{,\xi}}{\mathbf{r}_{,\xi}\cdot\mathbf{r}_{,\xi}} + \frac{\mathbf{r}_{,\eta\eta}\cdot\mathbf{r}_{,\eta}}{\mathbf{r}_{,\eta}\cdot\mathbf{r}_{,\eta}}\right) \quad (18)$$

where the comma denotes the partial-derivative, $\mathbf{r}_{,\xi} = \mathbf{e}_1$, $\mathbf{r}_{,\eta} = \mathbf{e}_2$. Eq. (18) is similar to the formulation of Barron [10]. The [*ii*,*i*] terms are due to divergence (dilatation), while the [*jj*,*i*] *i*≠*j* are curvature terms. The latter are frequently used to control line spacing near the wall. Values of P and Q are obtained at the walls, interior source-terms being obtained by interpolation, prior to conducting any calculations. This non-iterative prescription was used with both inverse and direct methods, assuming fixed-Dirichlet boundary conditions.

[1] The analysis is appropriate to 2D problems; in 3D, only one curve is assumed orthogonal to the others.

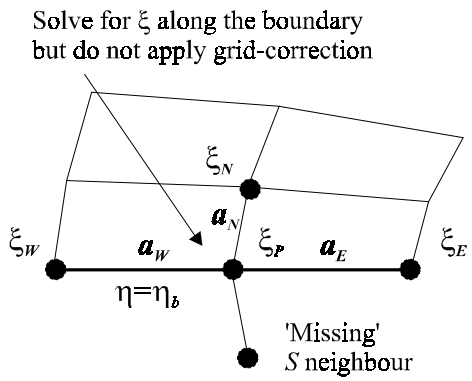

Figure 2. Iterative determination.

4.2 Iterative determination from nodal values

A second method developed by the author [11] involves the introduction of source-terms corresponding to the 'missing' boundary-terms. With reference to Fig. 2, the value of the source at the South wall is set according to,

$$S = S^* + a_N(i - \xi_P) \tag{19}$$

where i is the desired value at P, and S^* is the previous value of S. A similar treatment is applied to the ξ equations at the north boundary, and the η equations at the east and west boundaries, interior sources being interpolated from the boundaries. This procedure guarantees symmetry, i.e. zero-flux Neumann conditions at all boundaries, and at the same time ensures $\xi_P = i$ etc., in the fully-converged state. This method was only implemented using the direct method described in §3.

5. SOLUTION PROCEDURE

A variety of algorithms for the solution of Eq. (4) have been established. These include; point-by-point, line, slab, and whole-field solvers, which are documented in various reports and theses [12,13] and have been used in a variety of CFD codes.

A number of inner 'iterations' are performed over the block for each field variable, based on fixed values of the linking coefficients and source terms. After calculations have been performed for all field variables, the metric components, g^{ij}, linking coefficients, and built-in sources are updated. (For the direct scheme only, the grid is re-meshed using Eq. (15), prior to this stage.) The entire process is repeated for a number of outer iterations or 'sweeps' until convergence is achieved.

6. BENCHMARK PROBLEMS

Figure 3 shows the grids used in this study. Figure 3(a) to (c) are of a 40x40 cell H-type grid while 3(d) to (f) are a similar O-type mesh. Figures 3(a) and (d) show the initial algebraically-generated grids. Figures 3(b) and (e) were generated using the elliptic solvers based on the control-function rationale detailed in §4.1 above. Figures 3(c) and (f) show the effect of introducing boundary source terms according to §4.2. It can be seen that true orthogonality at the wall not attained in Figs. 3(b) and (e) has been achieved in Figs. 3(c) and (f).

Figures 4 and 5 show monitor-point data in the form of x-location of the grid centre-point vs. sweep number. Figure 4 shows results generated for Figs. 3(b) and (e) using the inverse method with non-iterative control-functions, Eq. (18): Two different schemes were used with both the H-grid and the O-grid: (i) a Jacobi point-by-point solver and (ii) a strongly-implicit whole-field solver. In all cases 5 iterations per sweep were performed, with $a_T = 0$. In addition, calculations were performed using the whole-field solver with only 1 iteration per sweep (H-grid only). The two schemes ultimately converge on the same values but, convergence is much more rapid with the whole-field procedure.

Figure 5 is a similar plot for results generated using the direct procedure. Convergence is similar to the Fig. 4 results. Also shown are data for Figures 3(c)(f), using the iterative scheme, generated using a point-by-point scheme.

7. DISCUSSION

The results demonstrate that the direct and inverse methods may be used to generate equivalent results with similar performance in terms of speed of convergence, under most circumstances. In the case of the point-by-point procedure, near-identical convergence was noted for the two methods, for these and other examples. With the whole-field solver, there appears to be a penalty associated with the direct method in the case of the H-grid when 5 iterations per sweep are performed: This is due to grid-correction being applied only at the end of each sweep (not every iteration), whereas in the inverse method x and y are updated implicitly. When the number of iterations per sweep was decreased from 5 to 1, virtually identical performance is obtained, as indicated in Figs. 4 and 5. For the O-grid, the direct method actually converges more rapidly than the inverse method, which is apparently under-damped.

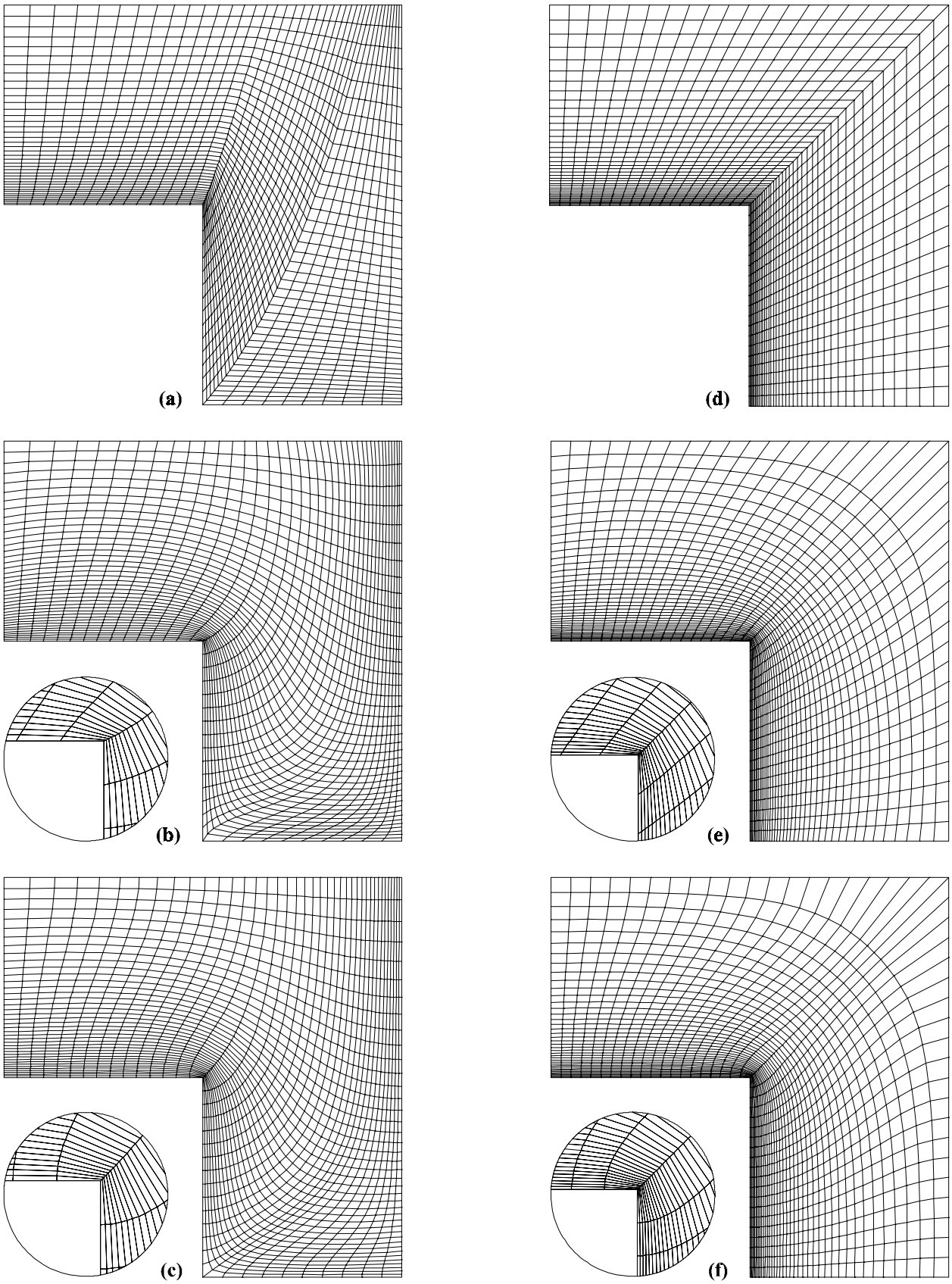

Figure 3. Test cases

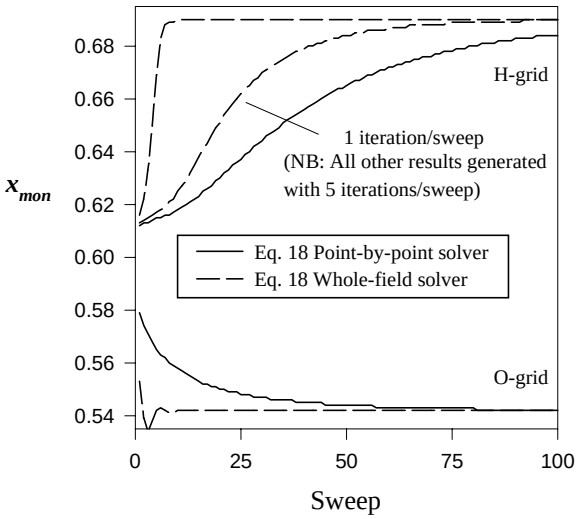

Figure 4. Monitor-point data, inverse method.

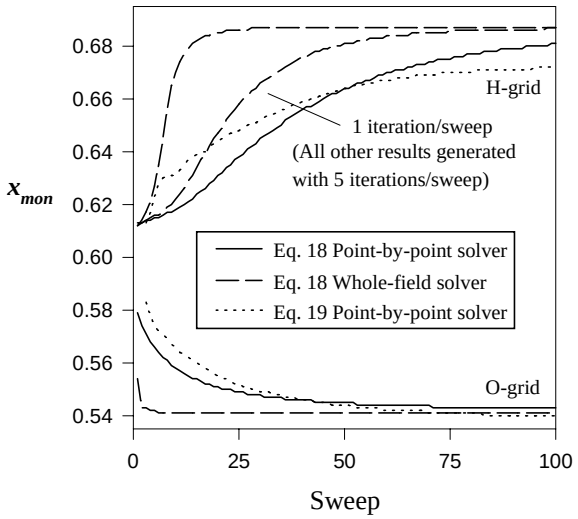

Figure 5. Monitor-point data, direct method.

For the examples considered, the whole-field solver converged in only 5-10 sweeps, about 10 times faster than the Jacobi point-by-point procedure. Jordan and Spaulding [14] indicate the benefits of strongly-implicit schemes to be problem-specific, though based on the fluid-mechanics experience, maximum benefit is to be expected for systems of equations which are not strongly-coupled via the source-terms.

For the direct method, the scalar equations are in the same form as for a conventional flow-solver. Equation (3) could therefore be replaced by the so-called 'physical' form, employed in many flow codes, (though this should not affect the final outcome of the calculations). Scalars ξ and η may also be solved-for independently, using different formulations or solvers, and one may view the scalar-solutions, as output from the direct method without ever applying grid correction (useful during development work).

Inverse methods feature simpler linking coefficients in the linear algebraic equations, and require less memory; however the control-function source-terms are cross-correlated, and contain first-order gradient terms. Generally-speaking, the conservatively-formulated direct method was more stable, especially when highly-concentrated grids were generated. Apart from this, the two methods are by-and-large same: Both methods are capable of uncrossing initially-folded grids (up to a point) by using inertial relaxation $a_T >> 0$, imposing limits on ϕ, and ensuring $\sqrt{g} \geq |\varepsilon|$, a small number, at all times.

Control-functions *P* and *Q* calculated using Eq. (18), (which is based on the assumption of orthogonality) generated grids which were not orthogonal at the boundaries; although the divergence terms [6] circumvent major distortions, and the addition of curvature terms probably improves overall grid quality. Moreover boundary-point control is also imprecise: Inspection of Figs. 3(d) and (e) reveals the grid has been pulled-away from the upper right corner. Re-computing *P* and *Q* at the end of each sweep, based on the current grid, did not improve the situation.

The iterative determination, Eq. (19) does actually effect local orthogonality, as is evident in the insets to Fig. 3. The approach does not purport to control the point distribution out from the wall, other than by way-of-influence of the end-points, a second control-function/source-term strategy is required to do this. Although the method was found to be readily applicable to highly-concentrated grids, convergence was not achieved when using the whole-field solver under these circumstances, requiring use of the point-by-point procedure, with associated performance penalty discussed above. This appears to be due to fluctuations in ϕ arising from the strong coupling between the source terms and nodal ϕ-values with Neumann boundary conditions. With reference to Fig. 2, attempts to impose Dirichlet conditions, by fixing ξ_P to i at η_b , and prescribing the source term, *S*, from the ϕ-values at the North node,

$$S = S^* + a_N\left(i - \xi_N\right) \qquad (20)$$

generated results quite similar to those obtained using Eq. (18), and suggest that Dirichlet-based boundary-prescriptions may not be capable of generating orthogonality, unless off-boundary nodes are explicitly adjusted.

Source terms are not the only way to effect grid-control; use of solved-for nodal values [4] in the place of integer reference-values in Eq. (15) combined with interpolation, is a proven viable alternative. Tests based on this approach converged rapidly, without the drawbacks of Eq. (19). However the grid is no longer parallel to the scalar fields. The problems with iterative source-term prescription are implementation-related; future improvements in terms of speed-of-convergence and stability are likely.

8. CONCLUSIONS

Inverse and non-inverse grid-generation methods were shown to generate similar results, when solving the same systems of equations. A strongly-implicit whole-field solver converged much more rapidly than an explicit point-by-point procedure. The inverse method did not offer any significant performance improvement over the non-inverse method, provided grid-correction is performed sufficiently frequently, for the latter method. An iterative source-term prescription, introduced using the direct method, was shown to generate precisely orthogonal grid-lines at boundaries, though with an associated penalty in performance. In general, the two methodologies were shown to be quite similar.

REFERENCES

1. A.M. Winslow. Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh. *J. Comput. Phys.*, **2**, pp. 149-172, 1967.

2. S.K. Godunov, and G.P. Prokopov. The Use of Moving Meshes in Gas-dynamical Computations. *Comput. Math. Math. Phys.*, **12**, 182-195, 1972.

3. J.F. Thompson, Z.U.A. Warsi, and C.W. Mastin, *Numerical grid generation, foundations and applications*, Elsevier, New York, 1985.

4. S.B. Beale. Numerical Grid Generation based on the Solution of Convection-diffusion-source Equations. *Proc. CFD Soc. Can.*, Banff, pp. 31-38, 1995.

5. 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**, pp. 405-410, 1979.

6. P. Thomas and J. Middlecoff. Direct control of the grid point distribution in meshes generated by elliptic equations. *AIAA J*., **18**, pp. 652-656, 1980.

7. R.L. Sorenson. Grid generation by elliptic partial differential equations for a tri-element augmentor-wing airfoil. *Numerical grid generation* Ed. J.F. Thompson, North Holland, p. 653, 1982.

8. J.F. Thompson. A General Three-dimensional Elliptic Grid Generation System on a Composite Block Structure. *Comput. Methods. Appl. Mech. Engng*. **64**, pp. 377-411, 1987.

9. B.K. Soni. Elliptic Grid Generation System: Control Functions Revisited-1. *Appl. Math. Comput.*, **59**, pp. 151-163, 1993.

10. R.M. Barron. Improvements to Grid Quality and Cost of Multiblock Structured Grid Generation. *Proc. CFD Soc. Can.*, pp. 303-309, 1996.

11. S.B. Beale. Choice of Control-functions for Numerical grid Generation.. *Proc. CFD Soc. Can.*, Ottawa, pp. 319-326, 1996.

12. D.B. Spalding. Mathematical Modelling of Fluid-mechanics, Heat-transfer and Chemical-reaction Processes: A Lecture Course. HTS/80/1, Imperial College, University of London, 1980.

13. S. Parameswaran. Finite Volume Equations for Fluid Flow Based on Non-Orthogonal Velocity Projections. Ph.D. Thesis, University of London, 1985.

14. S.A. Jordan and M.L. Spaulding. A Fast Algorithm for Grid Generation. *J.Comp. Phys.*, **104**, pp. 118-128, 1993.