A MULTIGRID METHOD FOR ELLIPTIC GRID GENERATION USING FINITE VOLUME MEHTOD

SHENG LUO, CHAOQUNLIU, HAUSHAN
Department. of Mathematics, University of Texas at Arlington,
Box 19409, Arlington, TX, U.S.A

1. INTRODUCTION

Elliptic and hyperbolic partial differential equations are, at the heart of most mathematical models used in engineering and physics, giving rise to extensive computations. Often the problems that one would like to solve exceed the capacity of even the most powerful computers. On the other hand, the time required is too great to allow inclusion of advanced mathematical models in the design process. Also iterative processes for solving the algebraic equations arising from discretizing partial-differential equations are stalling numerical processes, in which the error has relatively small changes from one iteration to the next. The computer grinds very hard for very small or slow real physical effect with the use of too-fine discretization grids. In this case, in large parts of the computational domain, the mesh size is much smaller than the real scale of solution changes. In particular, convergence of iterative methods for elliptic grid generation based on non-linear grid generation equations is extremely slow.

For the above reason, the authors have worked on the improvement of the non-linear elliptic grid generation. In this paper, the Laplace equation and the algebraic transformation were presented for the domains in 2D and 3D physical space. The second order finite difference scheme was used for the discretization of the grid generating equations. The linear system is solved by the ADI method and the convergence was accelerated by the multigrid FAS scheme. The performance characteristic of the algorithm was discussed and illustrations were made.

2. 2D AND 3D ELLIPTIC GRID GENERATION

The elliptic grid generation method used in this paper is based on the use of a composite mapping [1]. It is a composition of an algebraic transformation and an elliptical transformation based on Laplace equations. The algebraic transformation is a differential one-to-one mapping from computational space onto a parameter space. The parameter space and the computational space are unit squares. The algebraic transformation will only depend on the prescribed boundary grid point distribution. The control functions are defined based on the algebraic transformation. The elliptical transformation is a differential one-to-one mapping from parameter space onto the physical domain. The elliptical transformation depends only on the shape of the domain and is independent of the prescribed boundary grid point distribution. The composition of these two mappings defines the interior grid point distribution and is a differentiable and one-to-one for 2D domains and surfaces and, probably, also for 3D domains.

2.1 Grid Generation Equations

For 2D problem, the grid generating system of elliptic partial differential equations are as follow:

$$Px_{\xi\xi} + 2Qx_{\xi\eta} + Rx_{\eta\eta} + Sx_{\xi} + Tx_{\eta} = 0 \tag{1}$$

Where

$$P = (x_{\eta}, x_{\eta}), Q = -(x_{\xi}, x_{\eta}), R = (x_{\xi}, x_{\xi}),$$
$$S = PP_{11}^{1} + 2QP_{12}^{1} + RP_{22}^{1}, T = PP_{11}^{2} + 2QP_{12}^{2} + RP_{22}^{2} \tag{2}$$

The control functions are given by

$$P_{11} = -T^{-1}\begin{pmatrix} s_{\xi\xi} \\ t_{\xi\xi} \end{pmatrix}, P_{12} = -T^{-1}\begin{pmatrix} s_{\xi\eta} \\ t_{\xi\eta} \end{pmatrix}, P_{22} = -T^{-1}\begin{pmatrix} s_{\eta\eta} \\ t_{\eta\eta} \end{pmatrix} \tag{3}$$

and the matrix T is defined as $T = \begin{pmatrix} s_{\xi} & s_{\eta} \\ t_{\xi} & t_{\eta} \end{pmatrix}$

The six coefficients of the vectors $P_{11} = (P_{11}^{1}, P_{11}^{2})^{T}, P_{12} = (P_{12}^{1}, P_{12}^{2})^{T} and P_{22} = (P_{22}^{1}, P_{22}^{2})^{T}$ are called the control functions. The two algebraic equations that define the transformation are given by

$$s = s_{E_3}(\xi)(1-t) + s_{E_4}(\xi)t \tag{4}$$

$$\text{and } t = t_{E_1}(\eta)(1-s) + t_{E_2}(\eta)s \tag{5}$$

The 3D grid generating system is too complicated to describe here. It can be found in [1].

2.2 Discretization Method

Let's use 2D case as an example and consider a uniform rectangular grid of size $(N+1)(M+1)$ defined as $\xi_{i,j} = \xi_i = i/N$, $\eta_{i,j} = \eta_j = j/M$, $i = 0...N; j = 0...M$.

Let $X_{i,j}$ be prescribed on the boundary of this grid. We are going to compute $X_{i,j}$ in the interior of the computational grid based on the solution of the Poisson system defined by Eq.1.

The solution of this system of nonlinear elliptical equations is obtained by Picard iteration.

$$P^{k-1}\mathbf{x}^{k}_{\xi\xi} + 2Q^{k-1}\mathbf{x}^{k}_{\xi\eta} + R^{k-1}\mathbf{x}^{k}_{\eta\eta} + S^{k-1}\mathbf{x}^{k}_{\xi} + T^{k-1}\mathbf{x}^{k}_{\eta} = 0,$$

$$\text{Where, } P^{k-1} = (\mathbf{x}^{k-1}_{\eta}, \mathbf{x}^{k-1}_{\eta}),\ Q^{k-1} = -(\mathbf{x}^{k-1}_{\xi}, \mathbf{x}^{k-1}_{\eta}),\ R^{k-1} = (\mathbf{x}^{k-1}_{\xi}, \mathbf{x}^{k-1}_{\xi}),$$

$$S^{k-1} = P^{k-1}\,P^{1}_{11} + 2Q^{k-1}\,P^{1}_{12} + R^{k-1}\,P^{1}_{22},\ T^{k-1} = P^{k-1}\,P^{2}_{11} + 2Q^{k-1}\,P^{2}_{12} + R^{k-1}\,P^{2}_{22}.$$

The six control functions P^{1}_{11}, P^{1}_{12}, P^{1}_{22}, P^{2}_{11}, P^{2}_{12}, P^{2}_{22} are computed according to the equations given by Eq.2 and by applying second order central difference schemes for the discretizations of $s_{\xi\xi}, s_{\eta\eta}, s_{\xi\eta}, t_{\xi\xi}, t_{nn}$, $t_{\xi\eta}, s_{\xi}, s_{\eta}, t_{\xi}, t_{\eta}$

The arc length normalized variables $(s_{i,j}, t_{i,j})$ at the boundary are computed as follows:

- Compute the distance (*l*) between succeeding points at the each boundary.
- Define the length of the edge (*L*) along each boundary as the sum of the distances between succeeding points along that boundary.
- The normalized distance (*d*) between succeeding points is then given by *l/L*.
- The arc length normalized variables $s_{i,j}$ and $t_{i,j}$ at the boundary are defined by

$$s_{0,j} = 0, \quad s_{N,j} = 1, \quad j = 0...M, \quad t_{i,0} = 0, \quad t_{i,M} = 1, \quad i = 0...N,$$

$$\text{and } s_{i,0} = s_{i-1,0} + d_{i,0}, \quad s_{i,M} = s_{i-1,M} + d_{i,M}, \quad i = 1...N,$$

$$t_{0,j} = t_{0,j-1} + d_{0,j}, \quad t_{N,j} = t_{N,j-1} + d_{N,j}, \quad j = 1...M.$$

The arc length normalized variables $(s_{i,j}, t_{i,j})$ in the interior of the grid are computed according to the algebraic straight-line transformation given by Eq.(3) and (4). Simultaneously solving the two linear algebraic equations yields $(s_{i,j}, t_{i,j})$.

$$s_{i,j} = s_{i,0}(1 - t_{i,j}) + s_{i,M}t_{i,j}, \quad t_{i,j} = t_{0,j}(1 - s_{i,j}) + t_{N,j}s_{i,j},$$

at each node $(i, j) \in (1...N-1; 1...M-1)$

The steps for an iteration to improve the current approximation x^{k-1} is as follows:

- The coefficients $P^{k-1}, Q^{k-1}, R^{k-1}, S^{k-1}, T^{k-1}$ are computed by applying central difference schemes for the discretization of x^{k-1}_{ξ} and x^{k-1}_{η}. The six control functions remain unchanged during the iterative process.

- Using central difference schemes to discretize $x^k_{\xi\xi}, x^k_{\eta\eta}, x^k_{\xi}, x^k_{\eta}$. The discretization of the mixed derivative using central difference schemes is done in a way described in [2].
- We then obtain a linear system of equations for the unknowns $x^k_{i,j}, i = 0...N; j = 0...M$ with Dirichlet boundary conditions. A Multigrid solver is used to solve this linear system. The solver is called twice to compute the two components $x^k_{i,j}$ and $y^k_{i,j}$.

The initial approximation x^0 is obtained by an algebraic grid generation. The above iterative process is repeated until a required approximation to the solution has been obtained.

The discretization method for 3D problem can also be found in [1].

3. FINITE DIFFERENCE AND FINITE VOLUME METHOD

In this paper, the second order finite difference scheme was used to discretize the equations and the boundary conditions. For 2D problem, Eq. (1) becomes:

$$P\frac{x_{i+1,j}-2x_{i,j}+x_{i-1,j}}{\Delta x^2}+2Q\frac{-x_{i-1,j}-x_{i+1,j}-x_{i,j-1}-x_{i,j+1}+x_{i+1,j+1}+x_{i-1,j-1}+2x_{i,j}}{2\Delta x\Delta y}$$

$$+R\frac{x_{i,j+1}-2x_{i,j}+x_{i,j-1}}{\Delta y^2}+S\frac{x_{i+1,j}-x_{i-1,j}}{2\Delta x}+T\frac{x_{i,j+1}-x_{i,j-1}}{2\Delta y}=0$$

Grids obtained by the nonlinear elliptic Poisson grid generation system defined by Eq.(1) are grid folding free and have an excellent interior gird point spacing distribution. However, the computed grids are in general not orthogonal at the boundary, but we need grids to be orthogonal at the boundary, especially for Navier-Stokes computations. The orthogonality of the gird in a boundary layer is often desired.

The s coordinate in parameter space P satisfies the linear second-order elliptic equation $(Ja^{11}s_\xi + Ja^{12}s_\eta)_\xi + (Ja^{12}s_\xi + Ja^{22}s_\eta)_\eta = 0$. The t coordinate is obtained in the same way. A finite-volume cell-centered nine-point stencil approach is used to obtain the discretized equations for boundary orthogonality of 2D problem and seven-point stencil for that of 3D problem.

4. MULTIGRID TECHNIQUE

As we know, Gauss-Seidel relaxation for solving Eq.6 typically stalls after a few iterations. This is because Gauss-Seidel, though effective for high-frequency errors components, has very little effect on low-frequency components. Multigrid capitalizes on this "smoothing" property of Gauss-Seidel by visiting coarser grids to resolve smooth errors. But the regular multigrid correction scheme is only valid for linear problem. In order to accommodate the nonlinearities in Eq.6, we applied the Full Approximation Scheme (FAS) version of multigrid with bilinear interpolation and full weighting of residuals and approximations.

4.1. Full Approximation Scheme (FAS)

In non-linear problem, the difference $L_h u_h - L_h \bar{u}_h$ can no longer be replaced by $L_h v_h$, where v_h is the truncation error on fine grid. Hence, we introduce a new coarse grid variable $\bar{u}_{2h}$ defined as $\bar{u}_{2h} = \bar{I}_h^{2h} \bar{u}_h + v_{2h}$, where $\bar{I}_h^{2h}$ represents a so-called restriction operator which interpolates fine grid solution variables to the coarse grid and v_{2h} is the truncation error on coarse grid. The coarse grid equation can be written as $L_{2h} \bar{u}_{2h} - L_{2h} \bar{I}_h^{2h} \bar{u}_h = -I_h^{2h} r_h$, I_h^{2h} represents the restriction operator which transfers residuals from fine to coarse grids and r_h is the residue on fine grid. The operators $\bar{I}_h^{2h}$ may in principle be different from each other.

It is useful to rewrite the above equation as:

$$L_{2h} \bar{u}_{2h} = S_{2h}, \text{ where } S_{2h} = L_{2h} \bar{I}_h^{2h} \bar{u}_h - I_h^{2h} r_h .$$

In this form, the coarse gird equation is seen to take on a similar structure to the original fine grid equation, with a modified source term. This enables us to use similar techniques for solving both the coarse and fine grid problems. Once the coarse grid equations have been solved, either exactly or approximately, the fine grid variables are updated as:

$$\bar{u}_h^{new} = \bar{u}_h^{old} + I_{2h}^h (\bar{u}_{2h}^{new} - \bar{I}_h^{2h} \bar{u}_h^{old})$$

which can also be written as $\bar{u}_h^{new} = \bar{u}_h^{old} + I_{2h}^h v_{2h}$. I_{2h}^h is called interpolation operator.

The FAS method is carried out in the following steps. [4]

- Restrict the current approximation and its fine-grid residual to the coarse grid: $r_{2h} = I_h^{2h}(f_h - L_h(v_h))$ and $v_{2h} = I_h^{2h} v_h$, where f_h is the right-hand term of nonlinear equation system on fine grid.
- Solve the coarse-grid problem $L_{2h}(u_{2h}) = L_{2h}(v_{2h}) + r_{2h}$.
- Compute the coarse-grid approximation to the error: $e_{2h} = u_{2h} - v_{2h}$.
- Interpolate the error approximation up to the fine grid and correct the current fine-grid approximation: $v_h \quad v_h + I_{2h}^h e_{2h}$, where h: the mash space on finest grid, 2h: the mesh space on coarse grid.

4.2. The Solver for Linear Systems

The linear systems were solved by applying *alternating direction implicit* methods (ADI). The basic idea for ADI method is as follows: a complete iteration consists of two scans. The first scan goes from the smallest index number to the largest one from one row to the other followed by the same in the column direction. The second scan is almost the same as the first one, except for going from the largest index number to the smallest one. This method can increase the convergence rate at the cost of some complication in the computational algorithm. [3]

To illustrate the advantage of FAS method, the author compared ADI method with FAS and ADI method.

5. ILLUSTRATIONS

5.1. For 2-D Case

In Fig. 2(a), the body-fitted C-grid around a NACA 0012 airfoil is displayed. The close-up of mesh near the airfoil is shown in Fig. 2(b) where grids are orthogonal at the boundaries. The mesh is clustered near the wall in the wall-normal direction (η) as well as in the vicinity of the leading and trailing edge of the airfoil in the direction parallel to the wall (ξ). The grid contains 769 points in the ξ direction and 129 points in the η direction.

As seen in the figures, the grid generated is grid folding free and the interior grid point distribution is a good reflection of the prescribed boundary grid point distribution. The initial grid is obtained with algebraic grid generation and is required as the initial solution for the non-linear elliptical Poisson system. The final grid is independent of the initial grid. The quality of the initial grid is unimportant and severe grid folding of the initial grid is allowed.

Fig. 3. shows the Log (residue) vs. Work Units of the ADI and the multigrid method for 2D case. From the figure, the multigrid method reduced the residue significantly, which is faster than the ADI method for the above problem. The multigrid method saves at least 23 times in CPU times (or work unit) than the ADI method.

5.1. For 3-D Case

The geometry of the delta wind, taken from the experimental work of Rieley & Lowson (1998), is shown in Fig. 4. The sweep angle denoted by Λ is 85° and the leading-edge angle denoted by σ is 30°. The chord length is taken as the characteristic length L, such that the non-dimensional chord length is $c = 1.0L$. The non-dimensional thickness of the delta wing is $h = 0.024L$.

In Fig.5, the grids around the delta wind are being shown. The mesh is H-type in the meridian section and C-type in the cross section. The grids are orthogonal on the delta wing surface. The meshes are $129 \times 65 \times 65$, where the sequence of numbers is corresponding to the axial, the spanwise and the wall-normal direction, respectively.

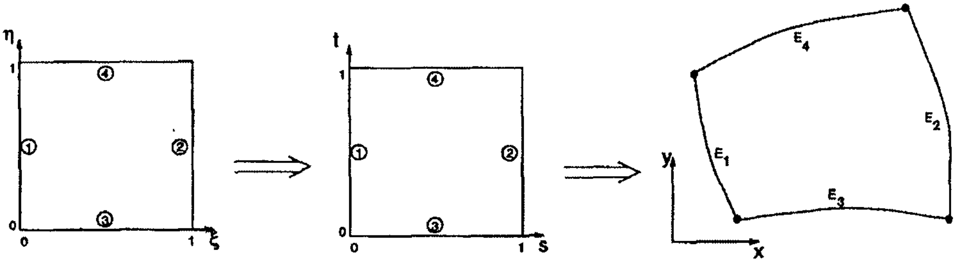
Fig. 6. shows the Log (residue) vs. Work Units of the ADI and the multigrid method for 3D case. Great improvement of convergence speed can also been seen from this figure, although it is not as apparent as that in 2D case. The multigrid method saves at least 2 times in CPU times (or work unit) than ADI method.

6. CONCLUSIONS

The Multigrid method is applied to elliptic partial differential equations. As seen from the illustrations, convergence of the 2D and 3D elliptic grid generation problems is greatly improved by using the multigrid solver, although convergence of three dimensional elliptic grid generation is still not fast enough and need to be improved. Second order central difference schemes used for discretizing the grid generation equations are close to the exact differentiation and provide better grid point distribution.

References

[1] S. P. Spekreuse, *Elliptic Grid Generation Based on Laplace Equations and Algebraic Transformations*, Journal of Computational Physics 118, 38-61, 1995.
[2] Dale A. Anderson, Johe C. Tannehill, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, 1984.
[3] William F. Ames, *Numerical Methods for Partial Differential Equations*, Third Edition, Academic Press, New York, 1992.
[4] William L. Briggs, Van Emden Henson and Steve F. McCormick, A Multigrid Tutorial, Second Edition, Siam, 2000

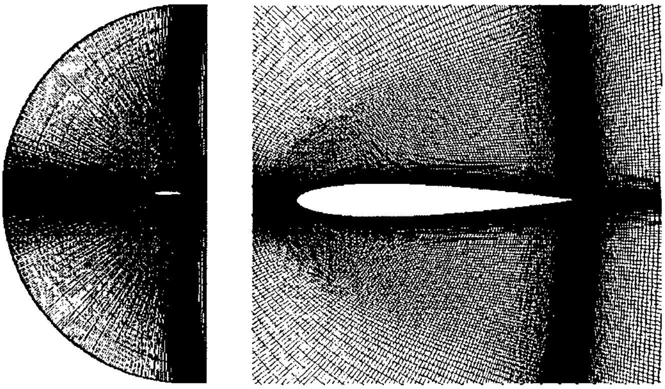


Computational space C

Parameter space P

Domain D

FIG. 1. Transformation from computational (ξ,η) space to a domain D in Cartesian (x, y) space.



(a) overview of the grids

(b) grids near airfoil surface

FIG. 2. C-grid around a NACA 0012 airfoil

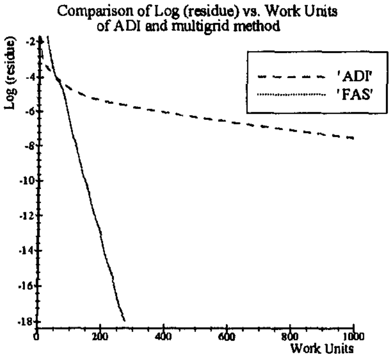


FIG. 3. Comparison of Log (residue) vs. Work Units of ADI and multigrid method in 2D case

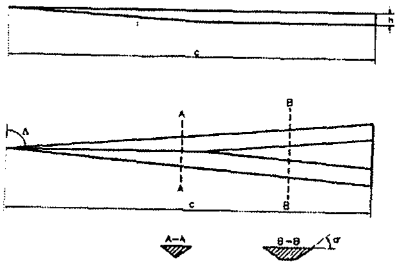


FIG. 4. Schematic of the delta wing

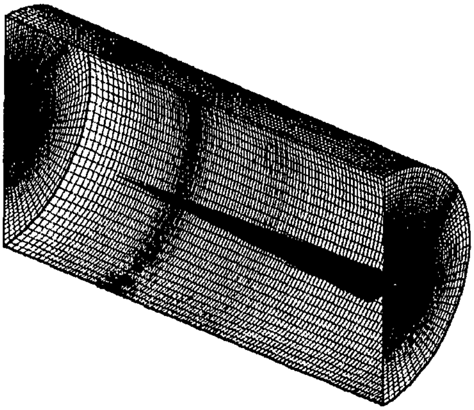


FIG. 5. H-C type grid around a 85°

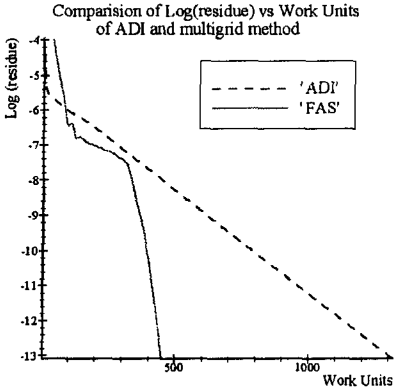


FIG. 6. Comparison of Log(residue) vs. sweep delta wing Work Units of ADI and multigrid method in 3D case