Optimization of Three-Dimensional Computational Grids

Richard Carcaillet,* Stephen R. Kennon,* and George S. Dulikravich†
The University of Texas at Austin, Austin, Texas

A method for generating and optimizing arbitrary three-dimensional boundary-conforming computational grids has been developed. The smoothness and local orthogonality of the grid are maximized using a fast iterative procedure, and provision is made for clustering the optimized grid in selected regions. An opimal grid can be obtained iteratively, irrespective of the method used to generate the initial grid. Unacceptable grids and even singular grids (i.e., grids containing regions of overlap) can be made useful for computation using this method. Application of the method to several test cases shows that grids containing regions of overlap are typically untangled in 2-5 iterations and that the conjugate gradient optimization procedure converges to an optimized grid within 25 iterations. Taking advantage of the original properties of this method, a new concept for generating optimal three-dimensional computational grids is proposed. It consists in optimizing a first guess of the desired grid, using an imperfect grid generated by a simple, inexpensive method as input.

I. Introduction

THE quality of the computational grid is essential to the accurate and stable numerical analysis of engineering flow problems. On the one hand, the grid should be smooth, that is, it should exhibit reasonable change in spacing in each of the curvilinear coordinate directions, so that the diffusionlike truncation error introduced by a nonuniform grid[1] is limited. On the other hand, orthogonality of grid lines should be enforced at the boundaries of the grid to allow accurate implementation of boundary conditions.[2]

Methods explicitly using these desirable qualities to generate a grid have been developed, at least conceptually.[3-6] The method of Brackbill and Saltzman has shown some remarkable results and has motivated the present study. In our work, an alternative, more heuristic formulation was adopted and successfully implemented in two dimensions using a nonlinear programming approach.[7] The method, which consists in iteratively optimizing a given arbitrary computational grid, has been extended to three dimensions and further refined. Satisfactory three-dimensional single grids about typical aircraft configurations are very difficult to achieve, since the slope discontinuities of the surface grid will result in highly skewed or even overlapped cells near the surface in most of the existing three-dimensional grid generation techniques. In some simple cases, it is possible to connect corresponding points of individual two-dimensional grids generated on successive planes or curved surfaces[8] to obtain a "stacked" grid. For more complex configurations, the only practical alternative to a single grid consists in generating contiguous subgrids on simpler three-dimensional domains[9] patched together to form the complete grid. This method, however, is quite complex and may create additional grid quality problems at the boundaries between the patches.

The present paper describes a method that will optimize a given arbitrary three-dimensional grid with respect to smoothness and local orthogonality while allowing local, directional clustering of the resulting grid. The optimization technique may be applied in this fashion, as a postprocessor capable of improving an unacceptable grid to the point at which it is useful for flow computation. But its potential is best exploited by using it as an efficient grid generator. In this case, only a rough first guess of the desired grid is needed as input. The resulting grid will possess an optimal combination of smoothness and local orthogonality for the given configuration. A detailed example of application of this original grid generation concept is given in this paper.

Received June 3, 1985; presented as Paper 85-4088 at the AIAA 3rd Applied Aerodynamics Conference, Colorado Springs, CO, Oct. 14-16, 1985; revision received Jan. 9, 1986.

*Graduate Research Assistant, Texas Institute for Computational Mechanics (TICOM), Department of Aerospace Engineering and Engineering Mechanics. Student Member AIAA.

†Assistant Professor, Department of Aerospace Engineering and Engineering Mechanics; presently, Associate Professor of Aerospace Engineering, The Pennsylvania State University, University Park, PA. Senior Member AIAA.

II. Analysis

Outline

The approach taken is one of unconstrained minimization of an appropriate (nonlinear) objective function. The latter is defined as a composite weighted measure of departure from smoothness and orthogonality over the entire grid. The minimization technique is a first-order iterative conjugate gradient method. Each iteration yields corrections dx, dy, dz to the physical coordinates x, y, z of each interior grid point such that the objective function decreases. The procedure is halted when corrections do not bring any further reduction in the objective function. Additional techniques to implement orthogonality at the boundaries and local, directional grid clustering complete the numerical scheme.

The Objective Function

The objective function rests on an alternative, heuristic formulation of the variational grid generation equations of Brackbill and Saltzman.[6] Assuming a computational grid given by the set of its grid point coordinates

$$\{(x,y,z)_{ijk}\},\ 1\le i\le I,\ 1\le j\le J,\ 1\le k\le K \qquad (1)$$

a local grid optimization problem is considered at each interior grid point $P(i,j,k)$ that is the center of a master cell consisting of the eight elementary neighboring grid cells that share $P(i,j,k)$ (Fig. 1). The grid points are assumed to be connected by straight line segments, and six position vectors $r1,\ldots,r6$ are defined that connect $P(i,j,k)$ to its immediate neighbors.

Qualitatively, the desirable properties of the master cell are stated as:

1) Smoothness: The master cell is smooth if the change in cell volume from one of its grid cells to the next is minimal, in all three directions $i=$ const, $j=$ const, $k=$ const.

2) Orthogonality: The master cell is orthogonal if the three grid lines $i=$ const, $j=$ const, $k=$ const intersect at $P(i,j,k)$ at right angles.

Quantitative smoothness and orthogonality measures are then formulated as follows, choosing for the former the most computationally efficient of several possible expressions.[10] Note that both measures are always positive and that they are minimal for a Cartesian master cell:

Smoothness measure:

$$SM_{ijk} = \kappa_1 \|r1\|^2 + \ldots + \kappa_6 \|r6\|^2 \tag{2a}$$

Orthogonality measure:

$$ORT_{ijk} = (r1 \cdot r2)^2 + \ldots + (r6 \cdot r4)^2 \text{ (12 terms)} \tag{2b}$$

The scalars κ ($\kappa \geq 1$) in Eq. (2a) can be seen as variable stiffness coefficients, in clear analogy with a system where grid points are connected to their immediate neighbors by tension springs. According to this analogy, minimizing the departure from smoothness SM of the master cell corresponds to minimizing the energy of the spring system. An immediate useful consequence is that local clustering of the desired grid is easily achieved, with respect to selected lines, surfaces, or three-dimensional regions. It suffices to specify functional relationships for the stiffness coefficients in the appropriate curvilinear direction(s).[10] For example, a relationship of the general form $\kappa = \kappa(k) + 1$, $\kappa(k) \geq 0$ will result in the clustering of grid surfaces $k = \text{const}$ according to the variations of $\kappa(k)$.

The global objective function is then obtained by taking a weighted sum of the local smoothness and orthogonality measures over all the interior grid points. It is written as

$$F = \sum_i \sum_j \sum_k [\alpha ORT_{ijk} + (1-\alpha) SM_{ijk}] \tag{3}$$

where α ($0 \leq \alpha \leq 1$) is a scalar weight parameter that enables a tradeoff between grid smoothness and local orthogonality to be achieved in the optimization procedure. Minimizing F will produce a grid that is optimally smooth ($\alpha = 0$), or optimally locally orthogonal ($\alpha = 1$), or that possesses a desired combination thereof determined by the value of α. Note that in Eq. (3), the range of the summations restrict the optimization procedure to interior grid points only. Points at the grid boundaries are left fixed, or allowed to "float" on the boundary surfaces. The latter option will be considered in a separate section.

The optimization algorithm: The problem of minimizing F is reformulated as an unconstrained nonlinear optimization problem for the function of N variables

$$F = F(V) \tag{4a}$$

where

$$V = \{(x,y,z)_{ijk}\}^T \tag{4b}$$

is the vector of length $N = 3 \times (I \times J \times K)$ containing the physical grid point coordinates of all the grid points in natural ordering. V represents a point in the design space of all the possible grid for the given boundary configuration. Iteratively minimizing F yields V^* (the optimal grid) such that

$$\|\nabla F(V^*)\| \leq \epsilon \tag{5}$$

where ϵ is the specified convergence criterion.

The Fletcher-Reeves conjugate gradient method[11,12] is chosen to solve the problem defined by Eqs. (4). This iterative search procedure possesses favorable convergence properties due to the use of information from the previous iterations or line searches. A true quadratic form will be minimized using this algorithm in at most N iterations. Each iteration consists of a line search in the direction defined by

$$\delta V^0 = -\nabla F(V^0)$$

$$\delta V^n = -\nabla F(V^n) + \beta_n \delta V^{n-1} \tag{6a}$$

where

$$\beta_n = \frac{\|\nabla F(V^n)\|}{\|\nabla F(V^{n-1})\|} \tag{6b}$$

F is minimized in this direction, which reduces the N-dimensional problem to a one-dimensional problem in the variable ω^n that can be stated as

$$\text{Minimize } \psi(\omega^n) = F(V^n + \omega^n \delta V^n) \tag{7}$$

The minimization of ψ^n or line searching is usually the costliest portion of such an algorithm. However, it is seen by inspection of Eqs. (2) that the grid quality measures are simple polynomials in the physical grid point coordinates. The smoothness measure contributes second-order and the orthogonality measure fourth-order terms to the objective function. It follows by substitution of Eq. (6a) into Eq. (7) that ψ^n is a fourth-order polynomial in ω^n. Hence, the line searching defined by Eq. (7) is exact, reducing to finding and testing the three roots of

$$\left(\frac{\partial \psi}{\partial \omega}\right)^n = 0 \tag{8}$$

and selecting the root ω^n that minimizes ψ^n. The grid point coordinates are then updated according to

$$V^{n+1} = V^n + \omega^n \delta V^n \tag{9}$$

This completes a short description of the algorithm employed. See Ref. 10 for more details and for considerations on the scaling of the search direction vector δV and on convergence toward the true minimum of F or toward a relative minimium (that represents a suboptimal grid). Treatment of the boundaries orthogonality at the boundaries is easily enforced for plane boundaries or boundaries that exhibit cylin-

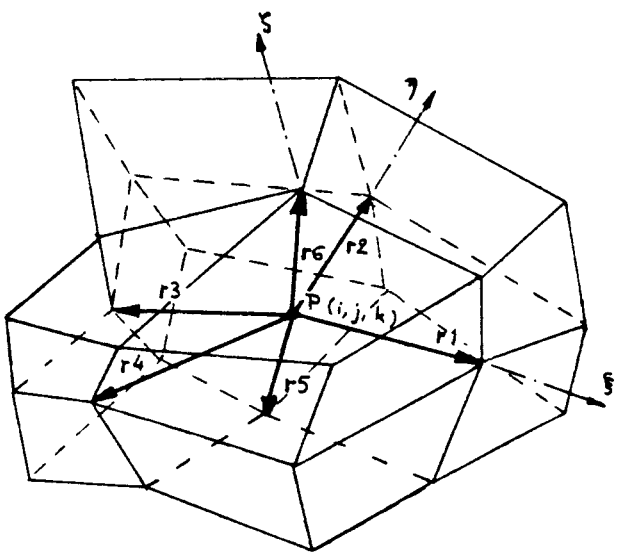

Fig. 1 Master cell.

Table 1 Influence of α on the untangling procedure

Optimization iteration no.	Number of overlapped master cells		
	$\alpha = 0$ (smoothness)	$\alpha = 0.5$	$\alpha = 1$ (orthogonality)
0	271	271	271
1	7	19	108
2	0	4	36
3	0	0	17
4	0	0	9
5	0	0	3
6	0	0	0

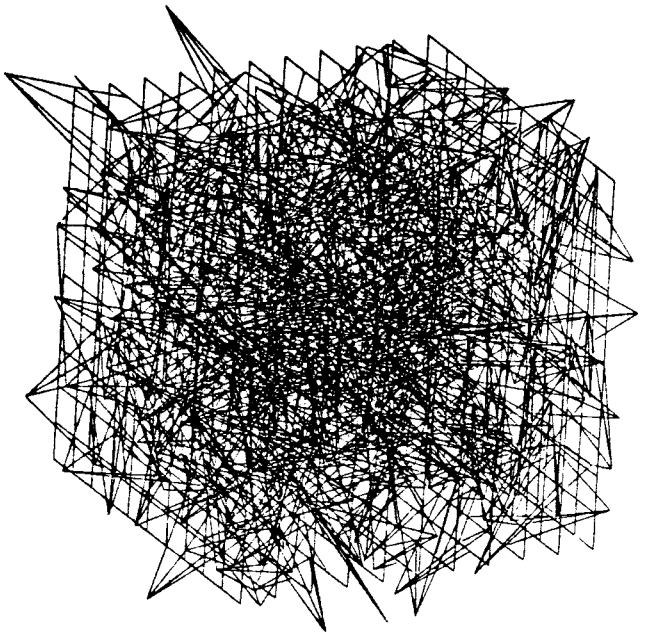

Fig. 2 Randomized initial grid (271 overlapped cells).

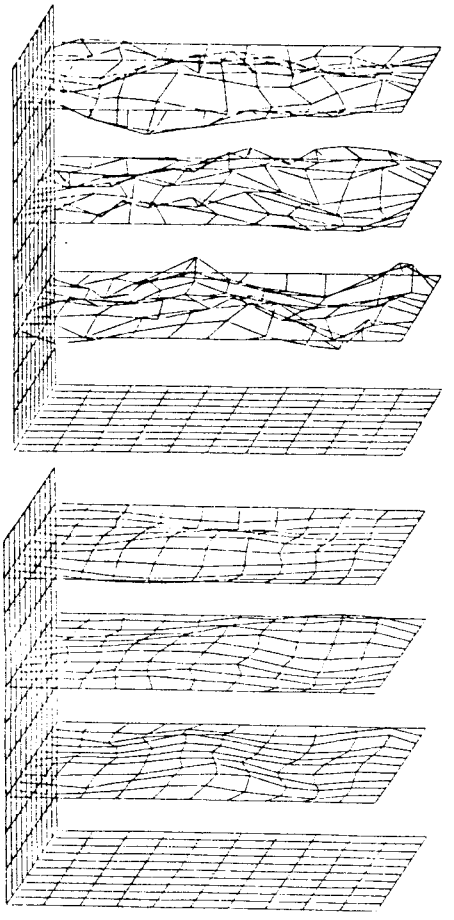

Fig. 3 Grid after one iteration (19 overlapped cells, above) and after three iterations (no overlapped cells, below).

drical symmetry. For general three-dimensional boundary surfaces, explicit reference to a mapping between the physical plane (x, y, z) and the computational plane (ξ, η, ζ) can be made, using, say, Lagrange interpolation for three-dimensional surfaces. Two techniques have been devised. The first technique is capable of treating complex boundary intersection configurations[10] and consists in including boundary points in the optimization process. This involves generating

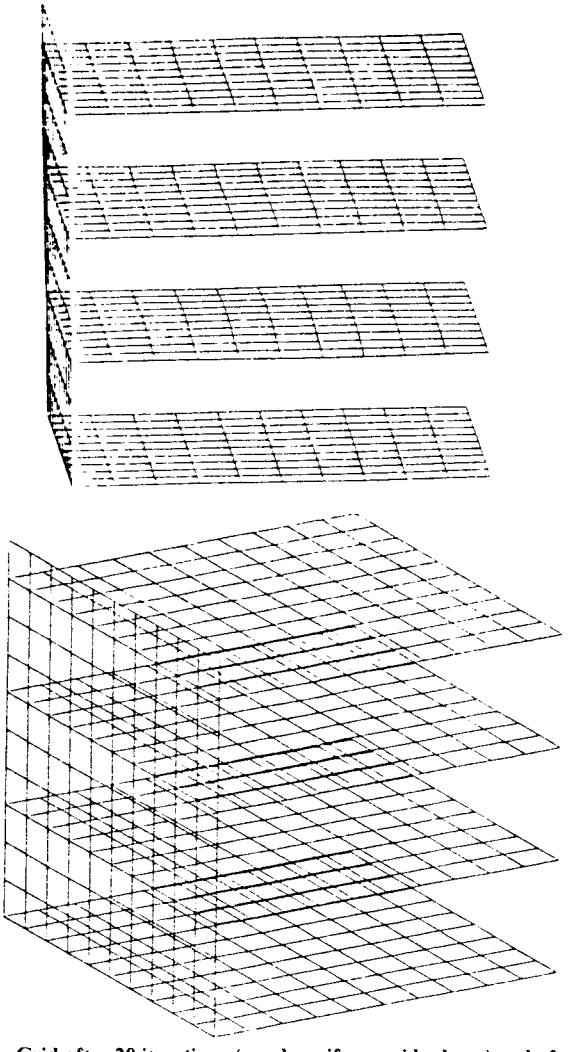

Fig. 4 Grid after 20 iterations (nearly uniform grid, above) and after 50 iterations (Cartesian grid is recovered, below).

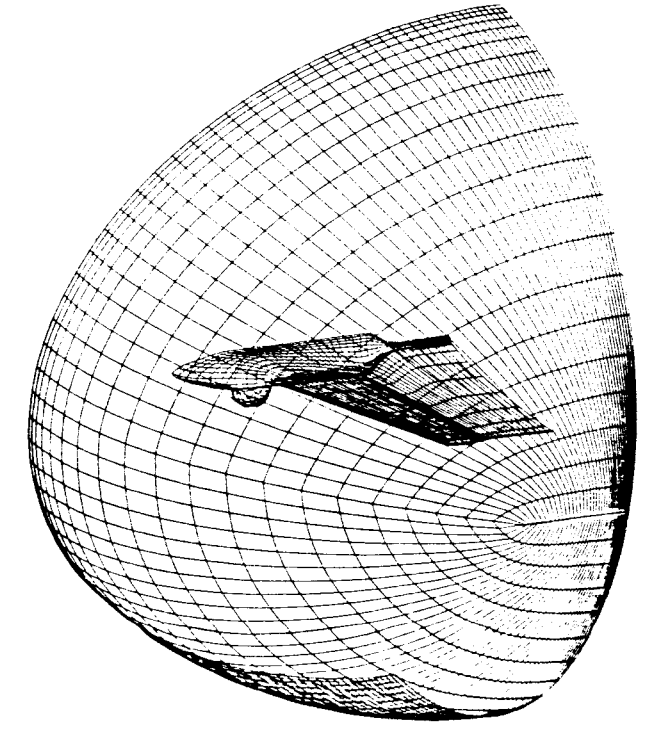

Fig. 5 Inner and outer grid surfaces for the "Aquila" configuration.

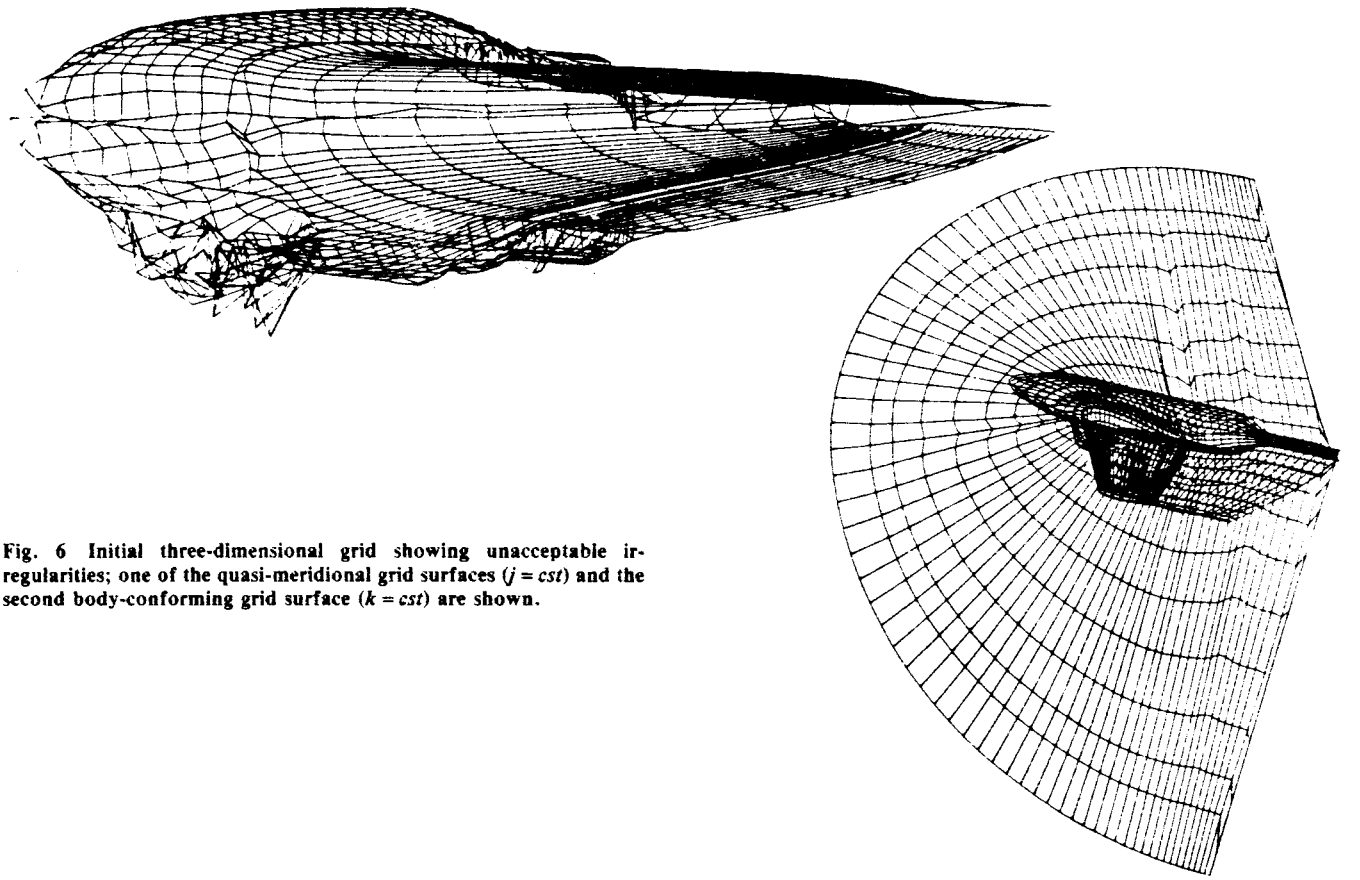

Fig. 6 Initial three-dimensional grid showing unacceptable irregularities; one of the quasi-meridional grid surfaces ($j = cst$) and the second body-conforming grid surface ($k = cst$) are shown.

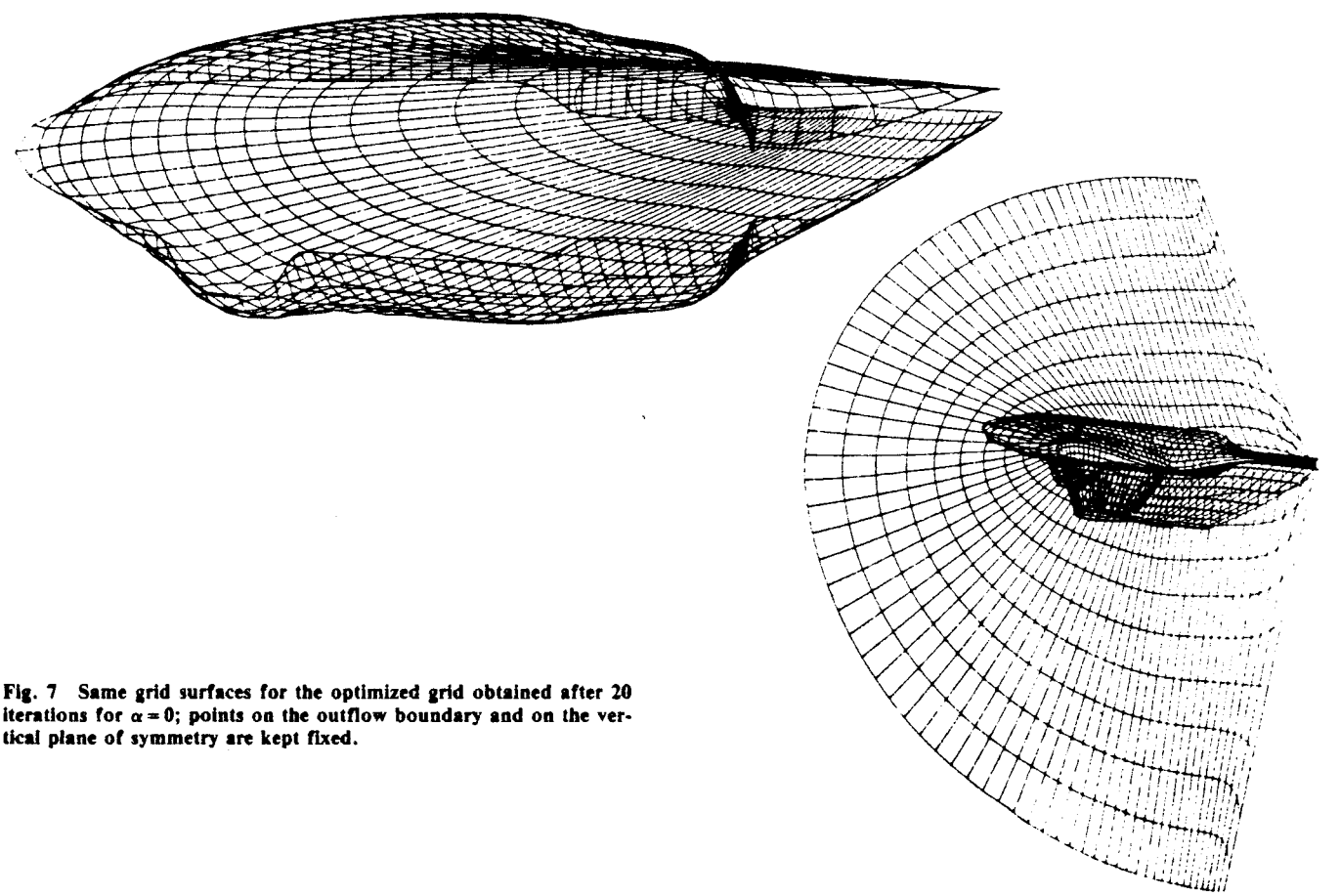

Fig. 7 Same grid surfaces for the optimized grid obtained after 20 iterations for $\alpha = 0$; points on the outflow boundary and on the vertical plane of symmetry are kept fixed.

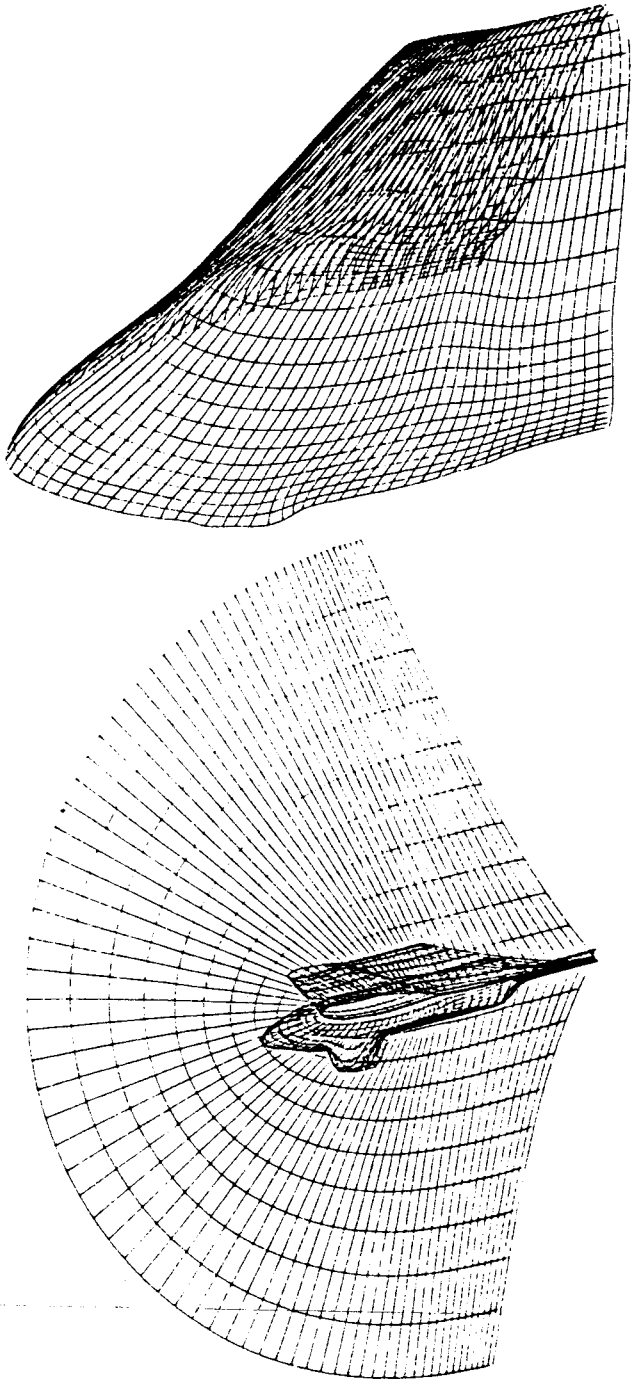

Fig. 8 Same grid surfaces for the optimized grid obtained after 20 iterations for $\alpha = 0$ with orthogonality enforced at the outflow boundary and at the vertical plane of symmetry.

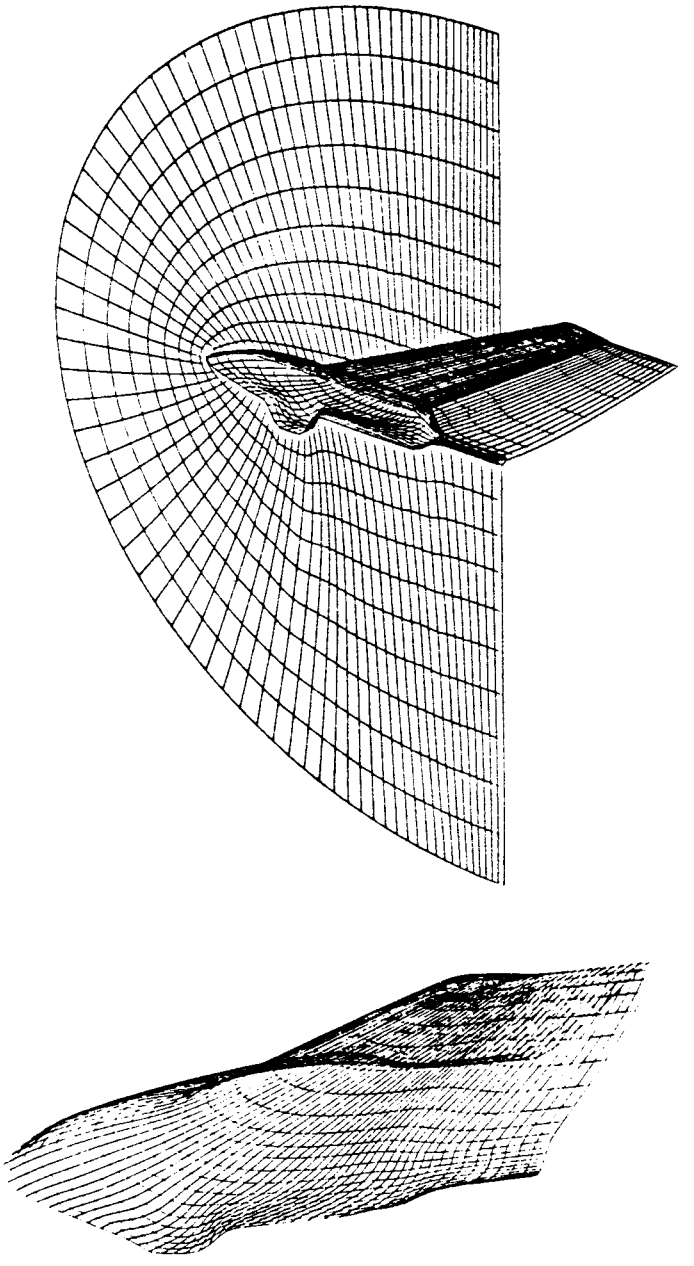

Fig. 9 For the same optimized grid: vertical plane of symmetry and first body-conforming grid surface, showing high degree of smoothness.

so-called imaginary points by reflection or extrapolation of the first boundary-conforming grid surface. The second technique, applicable to simple configurations, consists in a passive updating of boundary points that identifies them at each iteration with the Euclidian projection on the boundary surface of the corresponding grid points on the first boundary-conforming grid surface.[10]

III. Results

The input to the computer code that was developed to implement the grid optimization method is made of: 1) the set of the physical coordinates, normalized to unity, defining the initial grid; 2) the value of the scalar weight parameter α; (3) the convergence criterion ϵ; and 4) the value of some scalar clustering parameter that controls the variations of the functionals, say, $\kappa(k)$. In all the test cases investigated, the optimization procedure was shown to converge fast toward the optimal grid (within fifty iterations for $\epsilon = 10^{-4}$).

Randomized Cube Example

Given a Cartesian grid consisting of $10 \times 10 \times 10$ cubic cells forming the unit cube, random perturbations are added to the coordinates of each interior grid point, resulting in the severely overlapped grid shown in Fig. 2. Note that some grid points even lie outside the cube. This randomized grid is optimized for $\alpha = 0.5$. The optimization procedure easily untangles the 271 overlapped cells detected in the input grid in only two iterations. The grid is nearly uniform after 20 iterations, and the original Cartesian grid is retrieved in 50 iterations. Figures 3 and 4 illustrate the evolution of the grid in the course of optimization.

Tests with $\alpha = 0$ and $\alpha = 1$ demonstrate the influence of α on the untangling process, as documented in Table 1. The

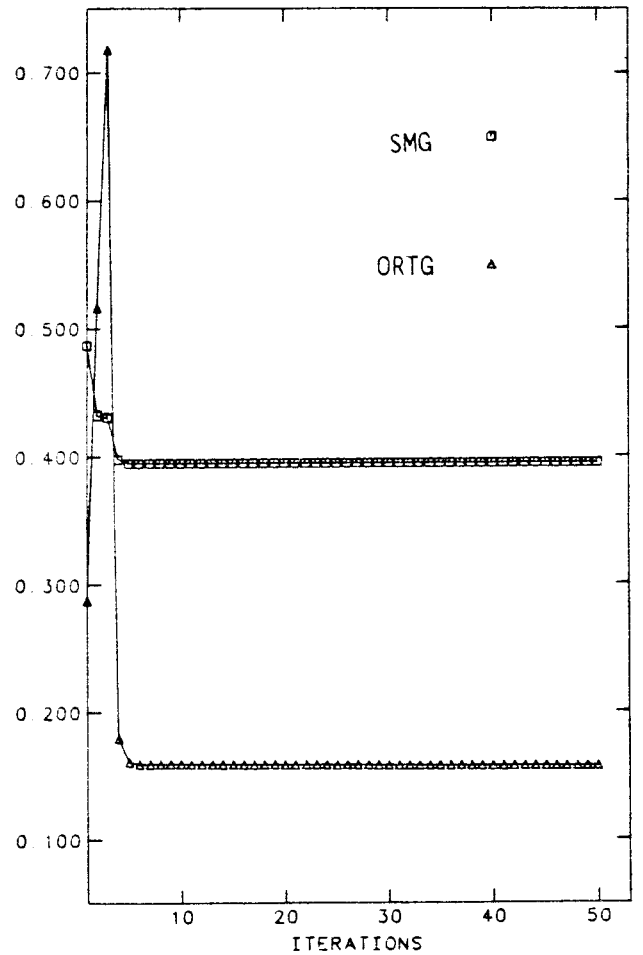

Fig. 10 Convergence history for the optimization procedure of Figs. 8 and 9 (SMG = global smoothness measure, ORTG = global orthogonality measure).

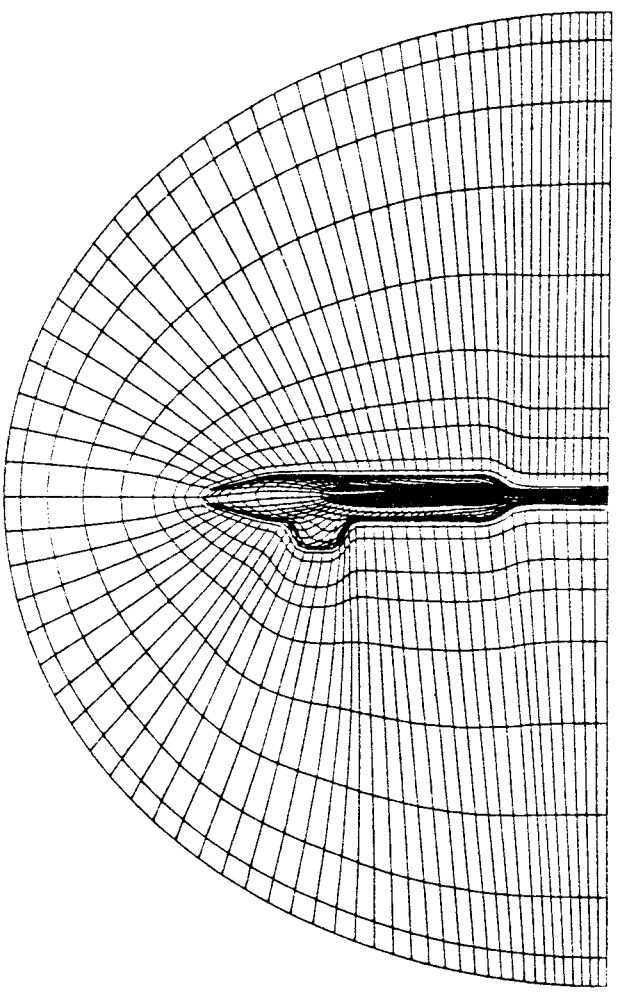

Fig. 11 Optimized grid obtained after 20 iterations for $\alpha = 0.4$, with clustering of the body-conforming grid surfaces toward the body.

number of remaining overlapped master cells is given for the first six iterations. Obviously, optimizing the grid in terms of smoothness only ($\alpha = 0$) allows a much faster untangling of the initial singular grid.

This example proves that the method is able to treat an extremely poor initial grid effectively and that it converges rapidly toward the optimal grid, which in this case was known. The resulting grid is the true minimizer of the global objective function.

Optimal Grid Generation Example

The ability of the method to untangle overlapped grids allows the task of generating a single three-dimensional grid about complex geometries to be recast in the following form:

1) Generate an imperfect initial grid about the configuration of interest, using a fast, simple technique.

2) Optimize this first guess in terms of smoothness and/or local orthogonality, implementing clustering where desirable.

In this example, an existing fast algebraic grid generation technique[13] is employed to generate a three-dimensional grid comprising $80 \times 20 \times 10$ cells between two surface grids, each having the same number of cells. The inner surface grid[14] simulates the Lockheed RPV "Aquila," the outer surface is represented by a hemispherical shape (Fig. 5). Notice the bulge beneath the fuselage, representing a camera pod, and the vortex sheet simulated behind the wing.

Figure 6 shows two surfaces from the initial grid, evidencing unacceptable irregularities. This grid is actually completely useless since 67 overlapped master cells were detected in it. Notice in particular the obvious problems near the bulge.

After 20 iterations of the optimization procedure for $\alpha = 0$, the grid illustrated in Fig. 7 is obtained, for which boundary points were left fixed. The improvement in smoothness is significant, the grid having been untangled in 4 iterations. Implementing orthogonality at the outflow boundary and at the vertical plane of symmetry in the same optimization procedure yields the grid shown in Fig. 8. The grid lines are now leaving these boundaries orthogonally. The same optimized grid is illustrated in Fig. 9, which emphasizes the degree of smoothness achieved. Increasing the number of iterations does not bring significant reduction of the objective function, as demonstrated in Fig. 10, where the global smoothness and orthogonality measures, defined as the sum over the master cells of Eqs. (2), are plotted against the number of iterations. The grid is essentially optimized after 20 iterations. In spite of a nearly constant smoothness, after 30 iterations one singular master cell appears at the nose tip. The simple smoothness measure with uniform stiffness constants fails here, since the first master cell upstream of the nose tip can only reach equilibrium by becoming singular. A more accurate smoothness measure based on differences of exact cell volumes would resolve this difficulty. Alternatively, choosing stiffer springs in this region suppresses the problem. Figure 11 shows the grid obtained after 20 iterations of the optimization procedure for $\alpha = 0.4$ and for stiffness constants taken as

$$\kappa(i) = \kappa(j) = 1, \kappa(k) = K - k \tag{10}$$

where $k = K$ at the outer boundary and $k = 1$ at the wing/body surface. It is seen that the clustering is satisfying, allowing an

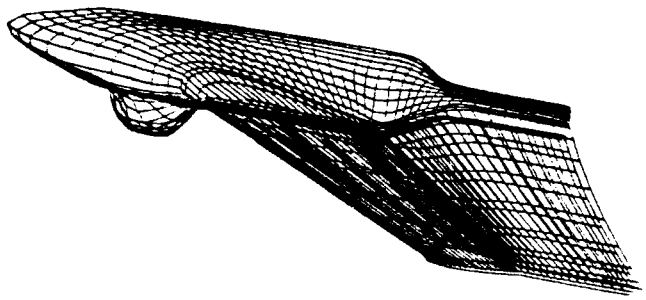

Fig. 12 For the same optimized grid: First body-conforming grid surface, showing high degree of smoothness (compare to Fig. 9).

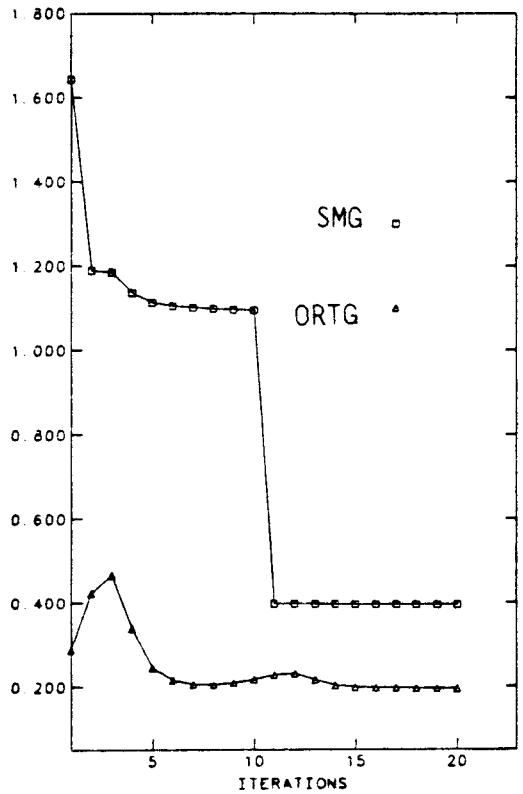

Fig. 13 Convergence history for the optimization procedure of Figs. 11 and 12 (SMG = global weighted smoothness measure, ORTG = global orthogonality measure).

accurate implementation of boundary conditions. The clustering effect is evident when comparing Fig. 9 to Fig. 12, where the same first body-conforming grid surface is shown. Finally, the convergence history presented in Fig. 13 shows that enforcing local grid clustering results in a smaller reduction of the orthogonality measure, although the latter was explicitly considered here. Numerous test cases have shown that best results are obtained for $\alpha \leq 0.5$, in terms of grid quality improvement as well as of convergence rate. Notice also how the stiffer springs effectively prevent large displacements of the grid points in the initial stage of the optimization procedure (Fig. 9).

In this last test case, which represents a typical application of the optimal three-dimensional grid generation concept, the optimal grid was obtained in 20 iterations, each iteration processing the 18711 grid points in 15 s of CPU time. The initial grid was generated in 12 s of CPU time. Both figures are for a Harris 800-II computer.

IV. Conclusions

The method presented in this paper has been shown to improve the quality of poor three-dimensional grids significantly. It is not tied to any particular grid generation method and allows an optimal grid to be obtained, even when the initial grid contains overlapped regions. This makes the grid optimization method an efficient, widely applicable tool capable of alleviating some of the difficulties inherent in three-dimensional grid generation. It can be easily extended to treat embedded or patched grids.[15] The development of optimal solution-adaptive three-dimensional grids is foreseen, based on recent research results in the two-dimensional case.[10]

References

[1]DeRivas, E. K., "On the Use of Nonuniform Grids in Finite-Difference Equations," *Journal of Computational Physics*, Vol. 10, 1972, pp. 202-210.

[2]Thompson, J. F. and Warsi, Z.U.A., "Boundary-Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations—A Review," *Journal of Computational Physics*, Vol. 47, 1982, pp. 1-108.

[3]Winslow, A. M., "Numerical Solution of the Quasilinear Poisson Equation in a Nonuniform Triangle Mesh," *Journal of Computational Physics*, Vol. 1, 1967, pp. 149-172.

[4]Barfield, W. D., "An Optimal Mesh Generator for Lagrangian Hydrodynamic Calculations in Two Space Dimensions," *Journal of Computational Physics*, Vol. 6, 1970, pp. 417-429.

[5]Brackbill, J. U. and Saltzman, J. S., "Adaptive Zoning for Singular Problems in Two Dimensions," *Journal of Computational Physics*, Vol. 46, 1982, pp. 342-368.

[6]Saltzman, J. and Brackbill, J., "Applications and Generalization of Variational Methods for Generating Adaptive Meshes," *Numerical Grid Generation*, edited by J. F. Thompson, Elsevier Science Publishing Co., 1982, pp. 865-884.

[7]Kennon, S. R. and Dulikravich, G. S., "A Posteriori Optimization of Computational Grids," AIAA Paper 85-0483, Jan. 1985.

[8]Dulikravich, G. S., "Fast Generation of Three-Dimensional Computational Boundary-Conforming Periodic Grids of C-Type," *Numerical Grid Generation*, edited by J. F. Thompson, Elsevier Science Publishing Co., 1982, pp. 563-584.

[9]Rubbert, P. E. and Lee, K. D., "Patched Coordinate Systems," *Numerical Grid Generation*, edited by J. F. Thompson, Elsevier Science Publishing Co., 1982, pp. 235-252.

[10]Carcaillet, R., "Generation and Optimization of Flow Adaptive Computational Grids," M.Sc. Thesis, Dept. of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, Aug. 1985.

[11]McCormick, G. P., *Nonlinear Programming—Theory, Algorithms and Applications*, J. Wiley & Sons, Inc., New York, 1983.

[12]Vanderplaats, G. N., *Numerical Optimization Techniques for Engineering Design*, McGraw-Hill Book Co., New York, 1983.

[13]Kennon, S. R., "Three-Dimensional Computational Grid Generation Using Polynomial Representation," Dept. of Aerospace Engineering and Engineering Mechanics, University of Texas at Austin, TICOM Report, to be published.

[14]Sommerfield, D. S. and Dulikravich, G. S., "WBCTG 31—FORTRAN Program for Efficient Three-Dimensional Computational Grid Generation for Wing-Body-Canard-Tail Realistic Aircraft Configurations—User's Manual," Bureau of Engineering Research, University of Texas at Austin, Rept. 84-100 1984.

[15]Kennon, S. R. and Dulikravich, G. S., "Composite Computational Grid Generation Using Optimization," *Proceedings of the First International Conference on Numerical Grid Generation for Computational Fluid Dynamics*, edited by J. Häuser, College of Landshut, Landshut, Germany, 1986.