NASA Technical Memorandum 88240

Elliptic Generation of Composite Three-Dimensional Grids about Realistic Aircraft

Reese L. Sorenson

March 1986

National Aeronautics and
Space Administration

NASA Technical Memorandum 88240

Elliptic Generation of Composite Three-Dimensional Grids about Realistic Aircraft

Reese L. Sorenson, Ames Research Center, Moffett Field, California

March 1986

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

ELLIPTIC GENERATION OF COMPOSITE THREE-DIMENSIONAL GRIDS ABOUT REALISTIC AIRCRAFT

Reese L. Sorenson
NASA Ames Research Center
Moffett Field, California

SUMMARY

An elliptic method for generating composite grids about realistic aircraft is presented. A body-conforming grid is first generated about the entire aircraft by the solution of Poisson's differential equation. This grid has relatively coarse spacing, and it covers the entire physical domain. At boundary surfaces, cell size is controlled and cell skewness is nearly eliminated by inhomogeneous terms, which are found automatically by the program. Certain regions of the grid in which high gradients are expected, and which map into rectangular solids in the computational domain, are then designated for zonal refinement. Spacing in the zonal grids is reduced by adding points with a simple, algebraic scheme. Details of the grid-generation method are presented along with results of the present application, a wing/body configuration based on the F-16 fighter aircraft.

1. INTRODUCTION

Of the several factors pacing the development of three-dimensional computational fluid dynamic (CFD) modeling of flows about realistic aircraft shapes, generating the grid is probably the most limiting [1]. The problem is how to have the grid points dense enough to provide adequate resolution of flow variables in regions of high gradients, while also limiting the total number of points so that the program will fit in the computer and run in a reasonable length of time. When the difficulty of selecting a grid topology appropriate to today's complex aircraft designs is added to the need for the grid to not be too skewed, a formidable challenge is seen.

Dividing the flow field into zones, and solving for the flow concurrently in each of the different zones, is an approach to these problems that has been effective in several recent works [2-8]. The advantages of this approach are several. One is that each zone can be topologically simple, even though complex aircraft shapes can be treated. Generating grids for simple zones is relatively easy, and implementing the flow solver in such zones is straightforward. Another advantage of this approach is that different flow-modeling equations can be used in different zones, as appropriate. For example, viscous effects can be modeled using the Navier-Stokes equations at no-slip body surfaces, while the Euler or full-potential equations can be used in the far field. Another advantage is that the computer storage capability needs to be equal only to the requirements of the largest zone, rather than to meet the needs of the entire problem at once.

The present work is a technique for generating zonal grids about complex and realistic aircraft shapes [9]. It consists first of constructing a coarse, body-conforming, global grid filling the entire flow field. Then fine zonal grids are constructed in regions which are rectangular in the computational domain, and roughly correspond to regions in the physical domain in which high gradients in the flow-field variables are expected. Although generating the coarse grid about a modern fighter airplane is difficult, the benefit of this approach is that the smaller zonal grids are found by simply subdividing the coarse-grid intervals. This is done in a simple and direct algebraic way. Furthermore, when flow-solver results require an adjustment to the size, location, or resolution of the fine zonal grids, the adjustment is usually accomplished without any modification to the coarse grid, and with simple changes to the subdivision which produces the zonal grids.

2. GENERATION OF THE COARSE GLOBAL GRID

The coarse global grid is generated using a three-dimensional extension of the widely distributed, two-dimensional, airfoil grid-generator program called GRAPE[1] [10,11]. This extension to three-dimensions has already been done for an elliptical topology [12]. The

---

[1]The acronym is "GRids about Airfoils using Poisson's Equation."

present work represents an extension to cylindrical topology, although the grid points are located by their Cartesian coordinates x,y,z. The computational variables are as seen in Fig. 1.

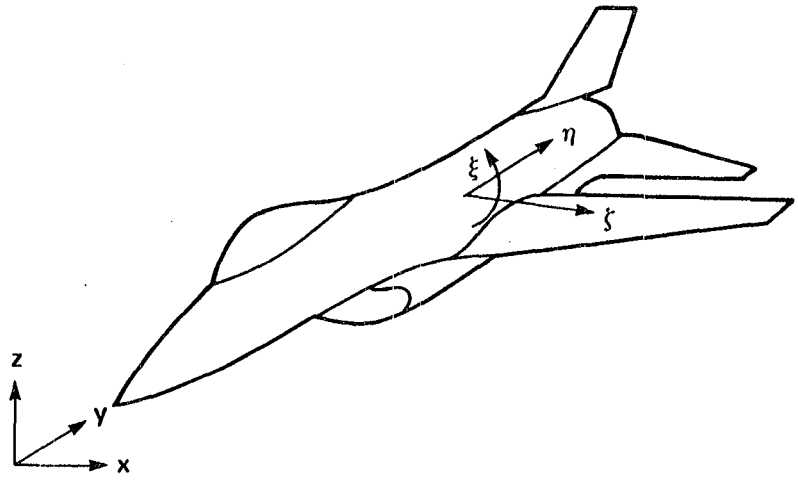

Fig. 1 Generic Airplane Showing Computational Variables in Cylindrical Topology.

It is required that the mapping between ξ,η,ζ and x,y,z satisfy the Poisson equations

$$\xi_{xx} + \xi_{yy} + \xi_{zz} = P(\xi,\eta,\zeta) \tag{1a}$$

$$\eta_{xx} + \eta_{yy} + \eta_{zz} = Q(\xi,\eta,\zeta) \tag{1b}$$

$$\zeta_{xx} + \zeta_{yy} + \zeta_{zz} = R(\xi,\eta,\zeta) \tag{1c}$$

in which P,Q,R are defined below. Equations (1a-c) are solved numerically in the uniform transformed space [13-15] as

$$\alpha_{11}\vec{r}_{\xi\xi} + \alpha_{22}\vec{r}_{\eta\eta} + \alpha_{33}\vec{r}_{\zeta\zeta} + 2(\alpha_{12}\vec{r}_{\xi\eta} + \alpha_{13}\vec{r}_{\xi\zeta} + \alpha_{23}\vec{r}_{\eta\zeta}) = -J^2(P\vec{r}_{\xi} + Q\vec{r}_{\eta} + R\vec{r}_{\zeta}) \tag{2a}$$

where

$$\vec{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \qquad \alpha_{ij} = \sum_{m=1}^{3} \gamma_{mi}\gamma_{mj} \tag{2b,c}$$

γ_{ij} is the ijth cofactor of the matrix M

$$M = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \qquad (2d)$$

and the Jacobian J is the determinant of M.

2.1 Controlling cell size and skewness on a typical boundary surface

Zonal grids are constructed by a simple subdivision of grid intervals in rectangular subregions of the coarse grid. While this method has many benefits, its efficacy depends critically upon whether or not the coarse grid is well-behaved at the boundaries. The meaning of "well-behaved" in this context is illustrated in Fig. 2. This figure shows a sketch of a grid cell that is tangent to a typical boundary surface, here the $\zeta = 0$ boundary surface. Since the vector $\vec{r}$ in Eq. (2) simply gives the location of a typical point,

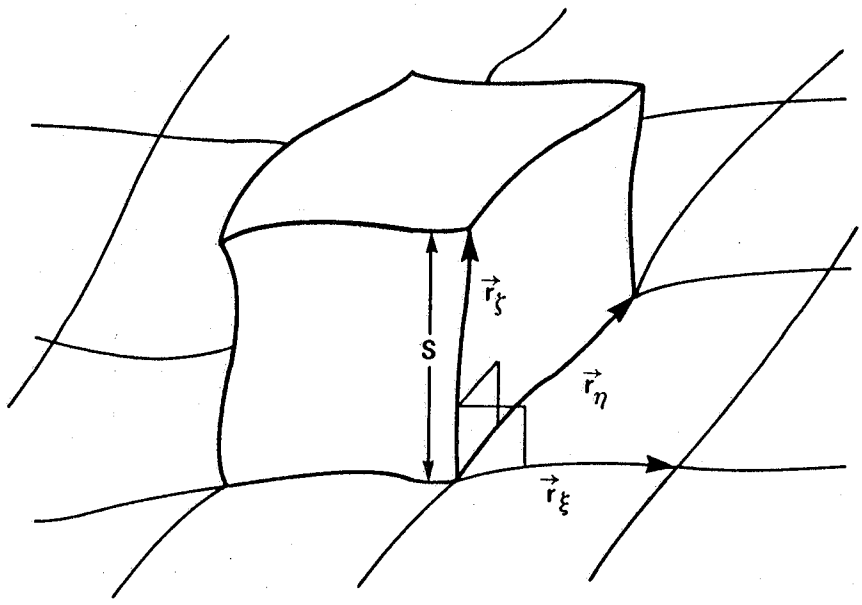

Fig. 2 Three Geometric Conditions Imposed at Typical Boundary Surface $\zeta = 0$.

derivatives of $\vec{r}$ could be thought of as unit-normal vectors in the indicated directions. For the subject grid cell to be well behaved, it is required that (1) the unit-normal vector $\vec{r}_\xi$ be orthogonal to $\vec{r}_\zeta$, (2) $\vec{r}_\eta$ be orthogonal to $\vec{r}_\zeta$, and (3) the height of the grid cell, here called S, be controlled. These three geometric constraints can be stated algebraically as

$$\vec{r}_\xi \cdot \vec{r}_\zeta = 0 \tag{3a}$$

$$\vec{r}_\eta \cdot \vec{r}_\zeta = 0 \tag{3b}$$

$$\vec{r}_\zeta \cdot \vec{r}_\zeta = S^2 \tag{3c}$$

Equations (3a-c) are side conditions which are added to the Poisson equations that generate the grid.

Continuing with a treatment of the cells on the typical $\zeta = 0$ boundary surface, the form chosen for the inhomogeneous term P is

$$P(\xi,\eta,\zeta) = p_1(\xi,\eta)e^{-a_1\zeta} \tag{4}$$

Q and R are similar. Note that the exponential factor is one (1) at the $\zeta = 0$ boundary, and that it decreases rapidly toward zero as ζ increases toward the middle of the grid. Thus the effect of that clustering term is at a maximum at the boundary, and it decays rapidly toward the middle of the grid. As a result, the controlling effect on cell size and skewness is greatest at the boundary, and the control decays rapidly toward the middle of the grid.

One view of the transformed Poisson equation, Eq. (2a), is that it is nothing more than a mass of derivatives on the left-hand side, and a linear combination of P, Q, and R on the right. Therefore, if numerical values could be obtained for all of the derivatives, a simple 3-by-3 linear system in P, Q, and R would result. Many of the derivatives required in Eq. (2a) can be obtained numerically from the fixed shape of the subject boundary ($\zeta = 0$), and the distribution of points on it. More of those derivatives are obtained from the side conditions, Eq. (3a-c). The only derivatives not found are the second partial derivatives with respect to ζ. Those derivatives are found by differencing the grid at the current computational time step.

The main computational loop for this method can be summarized in steps as follows:

1. Difference the grid at the current time step to obtain the derivatives.

2. Combine those derivatives with all of the other derivatives, which are fixed for all computational

time, and back-solve the transformed Poisson equations (2a) to get numerical values for P, Q, and R.

3. Using those values for P, Q, and R take one numerical solution step toward finding the x,y,z.

The above procedure is iterated to convergence. Thus the values for P, Q, and R which cause the grid to be well behaved at the subject $\zeta = 0$ boundary are found automatically as the solution for x,y,z proceeds.

2.2 A more general derivation of the inhomogeneous terms

The foregoing is a derivation of the inhomogeneous terms, and the resulting main solution loop, for the case wherein the grid cells tangent to one boundary surface, the $\zeta = 0$ surface, are controlled. However, it is generally not adequate, and control of the cells on several boundary surfaces is frequently needed. Inhomogeneous terms such as those above, but affecting cells tangent to different boundary surfaces, can be superimposed. Such terms can be simply added together because their controlling effects decay rapidly toward zero with distance away from their respective boundaries. Thus there are few, if any, regions wherein their controlling effects would be in conflict.

The inhomogeneous terms P, Q, and R are chosen to be

$$P(\xi,\eta,\zeta) = p_1 + p_2 + \ldots p_n + \ldots + p_{10} \qquad (5a)$$

$$Q(\xi,\eta,\zeta) = q_1 + q_2 + \ldots q_n + \ldots + q_{10} \qquad (5b)$$

$$R(\xi,\eta,\zeta) = r_1 + r_2 + \ldots + r_n + \ldots + r_{10} \qquad (5c)$$

The first constituent term in P, p_1, is identical in form to what is seen in Eq. (4)

$$p_1 = p_1(\xi,\eta)e^{-a_1\zeta} \qquad (6a)$$

The grid cells are controlled on the opposing $\zeta = \zeta_{max}$ boundary surface by the following term having similar form to Eq. (6a)

$$p_2 = p_2(\xi,\eta)e^{-a_2(\zeta_{max}-\zeta)} \qquad (6b)$$

Likewise, cells on the $\eta = 0$ and $\eta = \eta_{max}$ boundaries are controlled by

$$P_3 = P_3(\xi,\zeta)e^{-a_3\eta} \tag{6c}$$

$$P_4 = P_4(\xi,\zeta)e^{-a_4(\eta_{max}-\eta)} \tag{6d}$$

and cells on the $\xi = 0$ and $\xi = \xi_{max}$ boundaries are controlled by

$$P_5 = P_5(\eta,\zeta)e^{-a_5\xi} \tag{6e}$$

and

$$P_6 = P_6(\eta,\zeta)e^{-a_6(\xi_{max}-\xi)} \tag{6f}$$

Those six constituent terms together control grid cells on all six sides of the computational cube. However, that is not adequate for the present application, because of the presence of the wing in a slit. Two coordinate surfaces, which conform to the lower and upper surfaces of the wing, are coincident everywhere in the planform plane off of the wing. Grid cells are controlled on the lower and upper surfaces of the wing by

$$P_7 = \begin{cases} P_7(\eta,\zeta)e^{-a_7(\xi_w-\xi)} & \text{if } \xi \le \xi_w \\ 0 & \text{if } \xi > \xi_w \end{cases} \tag{6g}$$

$$P_8 = \begin{cases} P_8(\eta,\zeta)e^{-a_8(\xi-\xi_w)} & \text{if } \xi \ge \xi_w \\ 0 & \text{if } \xi < \xi_w \end{cases} \tag{6h}$$

in which ξ_w denotes not a derivative, but the value of ξ corresponding to the double-stored wing slit. In addition, terms must be added to control the grid cells in front of and behind the wing (i.e., terms must be added to attract grid lines to the leading and trailing edges). Those terms need act only within the $\xi = \xi_w$ coordinate surface. Thus they can be a degenerate two-dimensional form of the constituent terms as seen above

$$P_9 = \begin{cases} P_9(\zeta)e^{-a_9(\eta_{le}-\eta)} & \text{if } \eta \le \eta_{le} \text{ and } \xi = \xi_w \\ 0 & \text{if } \eta > \eta_{le} \text{ or } \xi \ne \xi_w \end{cases} \tag{6i}$$

$$P_{10} = \begin{cases} P_{10}(\zeta)e^{-a_{10}(\eta-\eta_{te})} & \text{if } \eta \ge \eta_{te} \text{ and } \xi = \xi_w \\ 0 & \text{if } \eta < \eta_{te} \text{ or } \xi \ne \xi_w \end{cases} \tag{6j}$$

It should be noted that the terms in Eqs. (6i) and (6j) are identical in form to the terms used in the two-dimensional grid generator called GRAPE.

The constituent terms q_n and r_n of inhomogeneous terms Q and R are similar to the p_n illustrated in Eqs. (6a-j). The main iterative loop is the same as what is delineated above, except that in each pass, steps 1 and 2 must be performed at each surface represented by a constituent term.

The coefficients a_n, b_n, and c_n in the constituent terms are positive constants. Values of 0.4 are typical. The point was made above that the clustering effect at each boundary surface dissipates rapidly with increasing distance away from that surface. The coefficients a_n, b_n, and c_n determine the rate at which that control dissipates. Increasing their value causes the control to dissipate more rapidly, and decreasing their value causes the control to be propagated far out into the field.

At the corners of the computational domain, or at the intersection of any two surfaces tangent to which cells are being controlled, problems can arise. It would be possible to have cell-height specifications for the two adjoining sides, combined with distributions of points on those boundary surfaces, which are mutually incompatible. This leads to a geometric impossibility, and to mathematically "overspecified" boundary conditions. This, in turn, causes nonconvergence of the grid-generation equations. The solution to this problem is to locally increase the coefficients a_n, b_n, and c_n at the corners to make the controlling effects decay with extreme rapidity. This causes the grid at the intersection to be locally Laplacian (i.e., locally uncontrolled).

After work with the present application it was determined that certain simplifications could be introduced. The present application is the construction of a grid, of cylindrical topology (Fig. 1), about a wing-body configuration based on the F-16 aircraft. The cylindrical axis is coincident with the long axis of the body, and one family of grid lines wraps around the aircraft from bottom to top. The grid covers only one side of the aircraft, since bilateral symmetry is assumed (no yaw). The wing resides in a slit, a surface of constant ξ. It was found that there was no need to control grid cells on the inflow and outflow boundaries where $\eta = 0$ and $\eta = \eta_{max}$. A completely adequate grid resulted by simply allowing the grid there to be uncontrolled (i.e., locally

Laplacian). Therefore, terms p_3 and p_4 were discarded, along with the corresponding q and r terms. Likewise, there was no need to control cells on the outer boundary $\zeta = \zeta_{max}$. Thus, term p_2 was also set to zero. In addition, there was no need to use the method above to control cells on the symmetry plane. A simple reflection-type boundary condition is applied there, with p_5 and p_6 set to zero.

3. SUBDIVISION OF THE GLOBAL GRID INTO ZONES

In regions of the flow field where high gradients are expected, zonal grids are constructed by the simple subdivision of the coarse grid intervals (see Ref. 2 for details). The domain of the zone is defined by choosing upper and lower limits on each of the three indices. This designates a region which is a rectangular solid in the computational space, but which can conform to body shapes in the physical space.

Herein lies a reason why the coarse grid near boundaries must be well-behaved, as discussed above. If the zone for refinement is to be a certain number of coarse cells high, wide, and long, then those measures of distance must have some consistency. If not, the physical boundaries of the zone are meaningless.

After defining the boundaries of the zone, the zonal grid is assumed to consist of the points of the coarse grid within the zone, with extra points added to them by simply dividing the coarse grid intervals. In directions tangent to the surface, those intervals are divided in halves, thirds, or quarters, etc. In the direction normal to the surface, where viscous effects are prevalent, the coarse grid intervals are divided even further by a spline-fitting technique. Each of the x, y, and z is fitted as a function of the arc length, and then those spline fits are evaluated at arbitrarily decreasing intervals. By this method the minimum spacing in the direction normal to the wing surface in the zonal grid (i.e., the viscous spacing) can be reduced by orders of magnitude.

Another need for a well-behaved coarse grid can be seen in the matter of the viscous spacing. If the viscous spacing over the entire wing surface is to be, for example, 0.001 times the height of the coarse cells tangent to the surface, and if that measure is to have any meaning, then those coarse cells must be nearly equal in height. Growth in the boundary layer can be accommodated by varying the viscous spacing over the wing surface. For the same reason, the coarse cells

tangent to the surface must not be skewed. If the coarse grid lines in the direction which is ostensibly normal to the surface are not near normal, but are instead highly skewed, then the true resultant viscous spacing will actually be far less than expected. In the above example, the viscous spacing is to be 0.001 times the "height" of the coarse cell. If those coarse grid lines, supposedly normal to the surface, were instead skewed at 45°, then the true resulting viscous spacing would be $0.001/\sqrt{2}$, or aproximately 0.0007.

The intervals in the coarse grid can be subdivided differently in each of the three directions. For instance, the intervals for some zone might be retained unchanged in one direction, halved in a second direction, and reduced to viscous spacing in the third direction. Thus the zonal grids are closely tailored to the needs of the flow solver. The sizes of the zones, and the degree to which the coarse grid intervals are subdivided, determine the number of points in each zone. The number of points per zone is adjusted so that the flow-solver's memory requirements for each zone are within the computer's memory limitations.

4. RESULTS

The present method is applied to a wing/body configuration based upon the F-16 fighter airplane. The fuselage, including the canopy, the strakes, and the "shelf" aft of the main wing are fully modeled [16]. The main wing and horizontal tail are also fully modeled. The ventral inlet is faired over, and the vertical tail, the ventral fins, and the missile rails on the wingtips have been deleted. A tapering sting of a circular cross section is added to the rear. Figure 3 is a "wire frame" figure, showing grid lines on the body and wings.

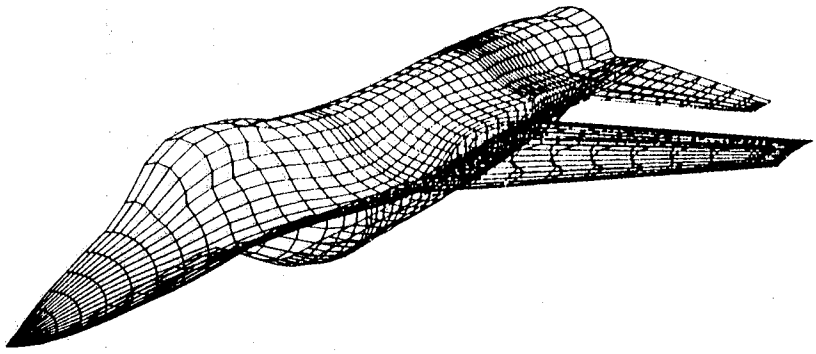

Fig. 3a Front-Quarter View.

Fig. 3 Grid Lines on Surface of Wing-Body Configuration Based on F-16.

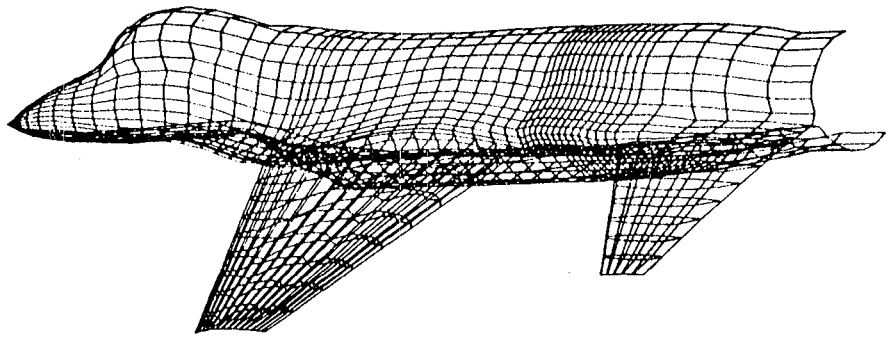

Fig. 3b Rear-Quarter View.

Fig. 3 Concluded.

Figure 4 is a view of the fuselage and main wing ahead of the mid-chord, showing the coarse global grid wrapping around the body. The grid cells on the fuselage surface are nearly orthogonal. The grid spacing normal to the fuselage (i.e., the height of the cells) is a nearly constant 20 cm, based upon the length of the entire aircraft being approximately 14 m. Likewise the cells on the wing surface are nearly orthogonal and are a nearly constant 10 cm high. The grid in the interior is smoothly varying, as is typical of a Laplacian grid. The outer boundary, not shown here, is a cylinder.

The wing/fuselage juncture is a corner where two controlled surfaces adjoin. As sometimes occurs, the grid cell height and skewness specifications, combined with the surface distributions, are inconsistent at the corner. This produces a mathematically overspecified problem, and leads to convergence difficulties in the grid generator. The solution here is to locally increase the values of a_n, b_n, and c_n so as to make the grid locally Laplacian at the corner.

Figure 5 is a planform view. The wing and horizontal tail are embedded in a slit, or a "double-stored" surface. The two grid surfaces are coincident everywhere except on the wing and horizontal tail. There the two surfaces separate, one modeling the upper surface and one modeling the lower surface. Outboard of the wing and horizontal tail the grid points are fixed, as though the wing extends outboard with zero thickness. This was found to be necessary to achieve reasonable behavior of grid lines just outboard of the tip. This wing extension is considered to be permeable by the flow solver, but that permeability is irrelevant to the grid generator.

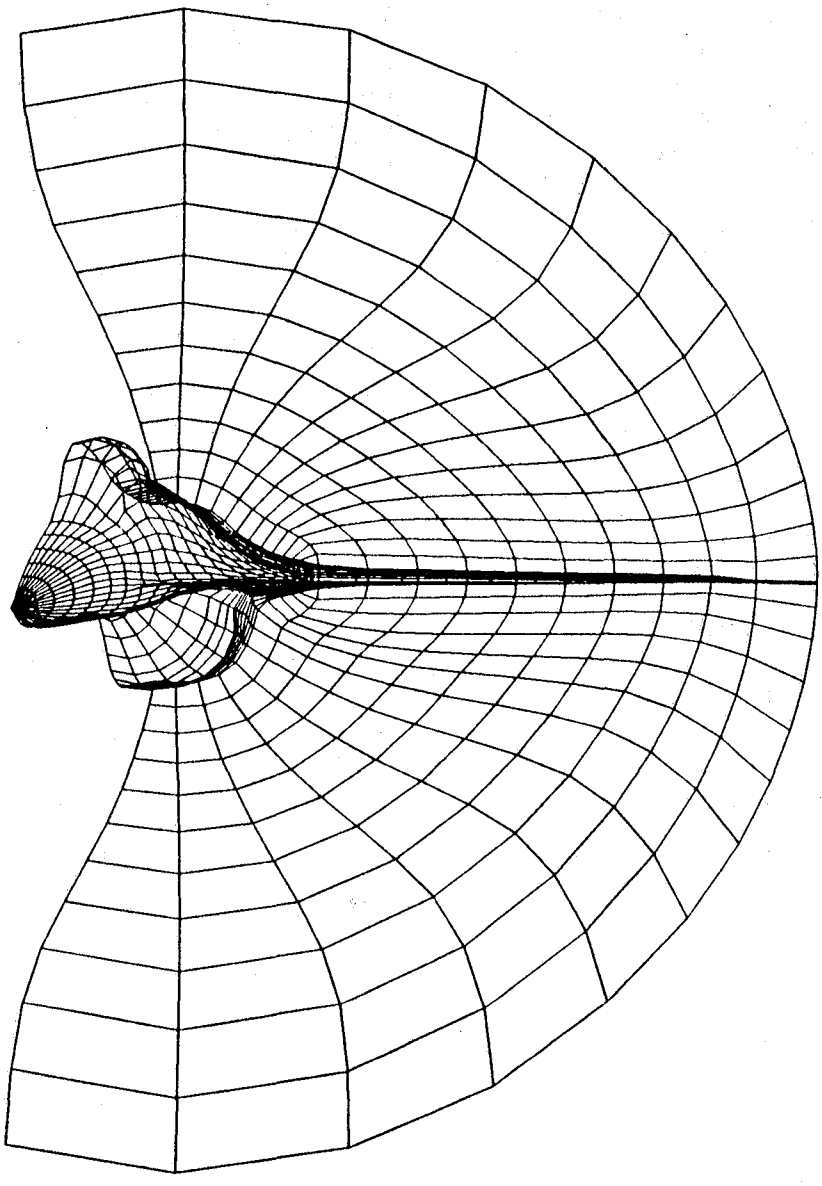

Fig. 4 Forebody and a Portion of a Streamwise-Normal Grid Surface at Mid-Chord. Outer Boundary Not Shown.

In this particular case, the "chordwise" lines on the wing surface are seen to be somewhat jagged. This figure, with its obvious error, was included to make the point that this grid generator is robust. The jagged lines are the result of a minor problem with the body-fitting, which has since been corrected. It is the philosophy of this work, and of the GRAPE program which preceeded it, that body fitting is not part of the grid generation effort, and that the fitted body is an input to the grid generator. The fact that the grid generator did converge, and did generate a smooth grid

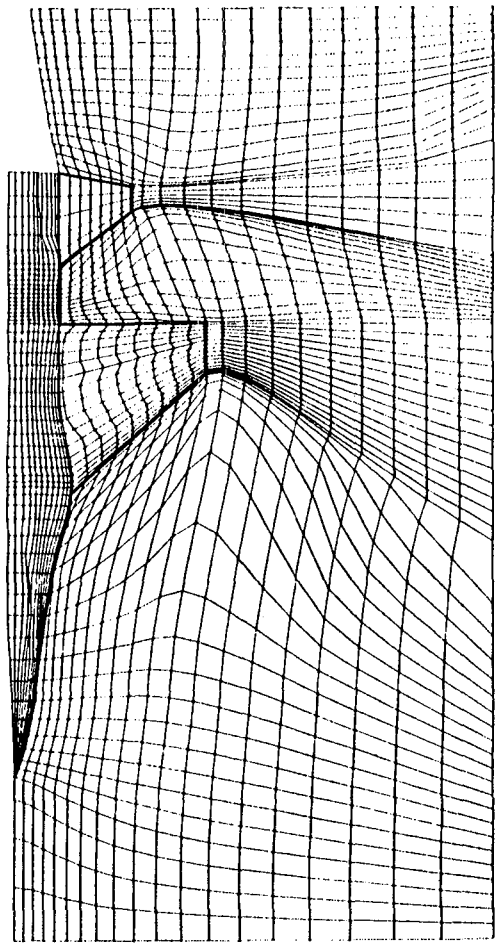

Fig. 5a Entire Planform Surface.

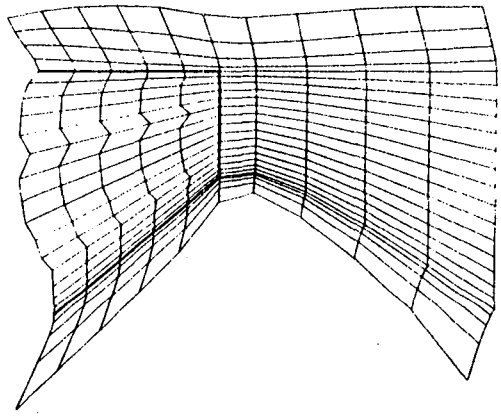

Fig. 5b Close-up of Wingtip Region.

Fig. 5 Planform View.

despite the jagged body fitting, indicates that the grid generator is robust, and can generate a well-behaved grid about bodies which are truly non-smooth.

Inhomogeneous terms P,Q,R were used to cluster points to the fuselage and to the wing's upper and lower surfaces. In addition, two-dimensional clustering terms, as in the grid-generation program GRAPE, which cluster points to a boundary, were added in the planform surface to cluster points to the leading and trailing edges within that surface.

Figure 6 shows an example of the third family of grid surfaces, namely those approximating cylinders in this cylindrical topology. It is a portion of the fifth such surface outward from the body, and passes through the wing at roughly mid-span. It can be seen here that the grid cell height is controlled and surface orthogonality is imposed at the wing's upper and lower surfaces.

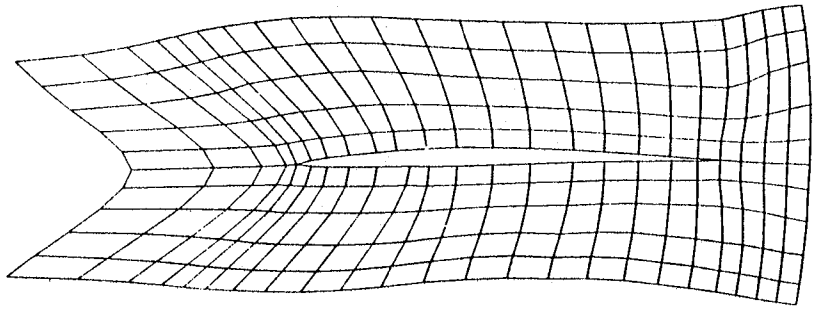

Fig. 6 A Portion of a Cylinder-Like Grid Surface, Shown Intersecting Wing at Mid-Span.

In generating a grid about such an aircraft using cylindrical topology, a problem arises along the cylindrical axis, upstream of the nose. The grid generator requires derivatives with respect to the computational variable ξ which proceeds around the axis. But since the points which would be differenced to find such derivatives are coincident, those derivatives are undefined. A simple but effective solution was suggested by Dr. Joseph L. Steger of NASA Ames Research Center. The boundary conditions and initial conditions are first specified. A transformation is then applied in which the points on the axis are moved out along radial lines by a small increment, turning the axis into what Steger calls a "soda straw," as seen in Fig. 7. In this transformation, all points are located temorarily in the cylindrical coordinates ρ, z, and θ where ρ is the radius in the x-z plane. Then the transformation is

$$\hat{\rho} = \rho + \Delta\rho(\rho_{max} - \rho)/\rho_{max} \qquad (7)$$

for $\Delta\rho$ being the radius of the soda straw, and ρ_{max} being the radius of the outer boundary. Note that the outer boundary points are undisturbed. This only applies to x-z coordinate surfaces which intersect the axis upstream of the nose of the aircraft; surfaces which intersect the aircraft, instead, are left unchanged. The points $\hat{\rho}$, z, and θ are transformed back into x, y, and z coordinates, and the grid-generation equations are then numerically solved, using derivatives which do exist at the "straw." The inverse of the above transformation is then applied, giving points on the cylindrical axis again.

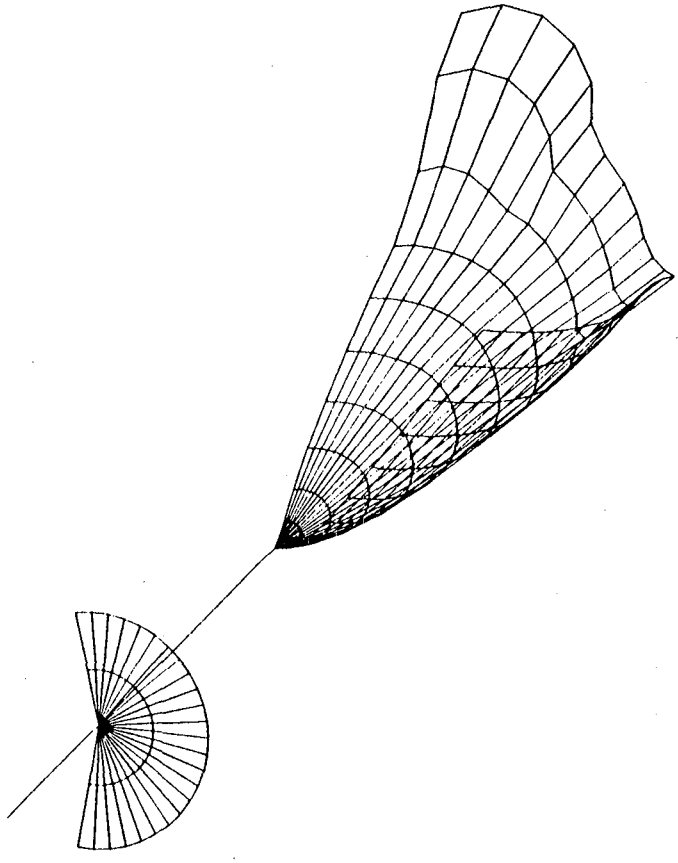

Fig. 7a Nose of Aircraft, With Axis Ahead of It, and Part of an Axis-Normal Grid Surface.

Fig. 7 Axis Singularity Treatment.

The Poisson grid-generation equations, including the side-condition equations which determine the inhomogeneous terms, are solved by a point-successive-over-relaxation (Point-SOR) technique. One-hundred iterations at approximately 1 sec of CPU time per iteration on a CRAY-XMP computer

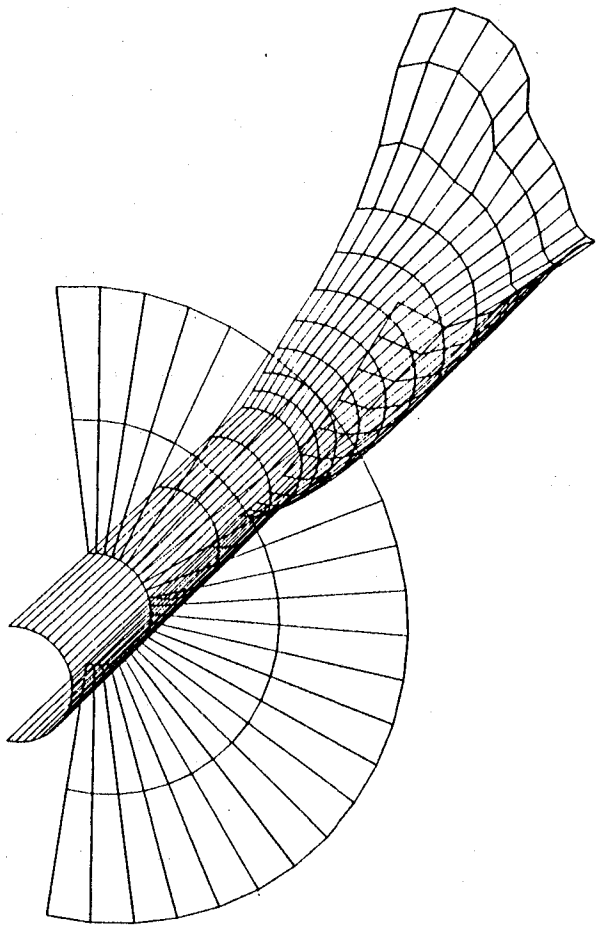

Fig. 7b Nose of Aircraft, With Axis Expanded to "Soda Straw."

Fig. 7 Concluded.

were required to generate this 37,000-point, coarse global mesh. Point-SOR was chosen as the grid-generation solver for its ease of programming, and its inherent flexibility with respect to boundary conditions. No attempt was made to vectorize or otherwise speed up the solver. Recoding for speedup, possibly combined with a faster solver such as ADI, would surely effect a large reduction in CPU time.

This coarse global grid is then divided into zones, as described above. Sixteen different zones for the entire wing/body are used, and the degree in which the coarse grid intervals are subdivided varies from unchanged spacing to viscous spacing. A total of approximately 300,000 points are used in all 16 zones. Since the zonal grids are constructed by simple analytic means, the computer time required to generate them is insignificant. Figure 8 is another plot like

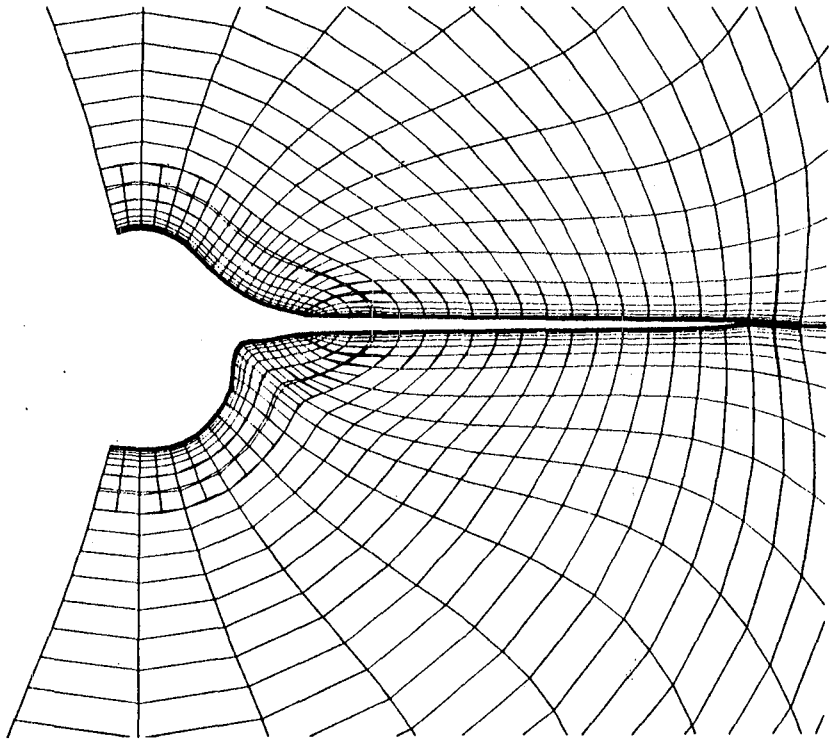

Fig. 8 Streamwise-Normal Grid Surface Showing Zones.

Fig. 4, but showing several of the zones. There are zonal grids with viscous spacing around the fuselage, at the upper and lower wing root, on the upper and lower wing surfaces, and outboard of the tip. Zonal grids with inviscid spacing are seen above and below the fuselage and wing.

Flow-field solutions using mixed Euler/Navier-Stokes flow modeling have been obtained about a wing alone, and are reported in Ref. 2. The grid generation for that effort is a direct precursor to the present work, and indicates this method's utility.

5. CONCLUSIONS

A method for constructing zonal grids about realistic aircraft shapes for use in CFD simulations is developed. The method is workable for a variety of different flow-modeling equations, including those treating viscous effects. The method provides body-conforming grids with the smoothness for which elliptic methods are known. The size and orthogonality of grid cells at boundaries and at other selected surfaces are easily controlled. With this method, it is especially easy to

construct and modify the zonal grids, once a coarse global grid is obtained.

As shapes to be gridded become more realistic, and thus require more complex grid topology (i.e., as the computational domain deviates even more from the ideal cube, it becomes more difficult to make the global grid). That difficulty will ultimately require that the entire flow field be divided into several regions, with a coarse grid in each region generated as above. The present method's ability to control grid behavior at boundary surfaces will alleviate any difficulty in patching those regional grids together. Then the joined coarse grids can be subdivided as above. As an example, work is presently under way to model the F-16 with the engine inlet. A regional grid will extend from the inlet to the engine compressor face. Such an approach makes the present method promising for even the most complex aircraft shapes.

6. ACKNOWLEDGMENTS

The author would like to thank Mr. Thomas A. Edwards of NASA Ames Research Center for his help in providing the surface-fitting of the F-16 aircraft; General Dynamics for their generosity in providing the basic data defining that aircraft's shape; Major Steven Reznick, USAF, for his help in subdividing the coarse grid into zones; and Dr. Joseph L. Steger of NASA Ames for his suggestion for treating the axis singularity.

REFERENCES

1. CHAPMAN, D.R. - 'Computational Aerodynamics: Review and Outlook'. AIAA Journal, 1979, 17, 1293.
2. HOLST, T.L., GUNDY, K.L., FLORES, J., and CHADERJIAN, N.M., KAYNAK, U., and THOMAS, S.D. - 'Numerical Solution of Transonic Wing Flows Using an Euler/Navier-Stokes Zonal Approach'. AIAA Paper 85-1640, 1985.
3. ATTA, E.H. - 'Component Adaptive Grid Interfacing'. AIAA Paper 81-0382, 1981.
4. ATTA, E.H. and VADYAK, J.A. - 'Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows about Aircraft Configurations'. AIAA Paper 82-1017, 1982.
5. BENEK, J.A., STEGER, J.L., and DOUGHERTY, F.C. - 'A Flexible Grid Embedding Technique with Application to the Euler Equations'. AIAA Paper 83-1944, 1983.
6. HESSENIUS, K.A. and PULLIAM, T.H. - 'A Zonal Approach to Solution of the Euler Equations'. AIAA Paper 82-0969, 1982.

7. RAI, M.M. - 'A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations'. AIAA Paper 84-0164, 1984.
8. HESSENIUS, K.A. and RAI, M.M. - 'Applications of a Conservative Zonal Scheme to Transient and Geometrically Complex Problems'. AIAA Paper 84-1532, 1984.
9. SORENSON, R.L.. - 'Three-Dimensional Elliptic Grid Generation About Fighter Aircraft for Zonal Finite-Difference Computations'. AIAA Paper 86-0429, 1986.
10. SORENSON, R.L. - 'A Computer Program to Generate Two-Dimensional Grids about Airfoils and Other Shapes by the Use of Poisson's Equation'. NASA TM-81198, May 1980.
11. STEGER, J.L. and SORENSON, R.L. - 'Automatic Mesh-Point Clustering Near a Boundary in Grid Generation with Elliptic Partial Differential Equations'. J. Comp. Phys., 1979, 33, 405.
12. SORENSON, R.L. and STEGER, J.L. - 'Grid Generation in Three Dimensions by Poisson Equations with Control of Cell Size and Skewness at Boundary Surfaces', ASME - FED 5, 1983, p. 181.
13. MASTIN, C.W. and THOMPSON, J.F. - 'Transformations of Three-Dimensional Regions Onto Rectangular Regions by Elliptic Systems'. Numerische Mathematic, 1978, 29, 397.
14. THAMES, F.C. - 'Generation of Three-Dimensional Boundary-Fitted Coordinate Systems for Wing/Wing-Tip Geometries Using the Elliptic Solver Method', Numerical Grid Generation, Ed. Thompson, J.F., Elsevier Science, 1982, p. 695.
15. THOMAS, P.D. - 'Numerical Generation of Composite Three Dimensional Grids by Quasilinear Elliptic Systems', Numerical Grid Generation, Ed. Thompson, J.F., Elsevier Science, 1982, p. 695.
16. EDWARDS, T.A. - 'Definition and Verification of a Complex Aircraft for Aerodynamic Calculations'. AIAA paper 86-0431, 1986.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-88240		

4. Title and Subtitle	5. Report Date
ELLIPTIC GENERATION OF COMPOSITE THREE-DIMENSIONAL GRIDS ABOUT REALISTIC AIRCRAFT	March 1986
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Reese L. Sorenson	A-86165
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Ames Research Center Moffett Field, CA 94035	
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code
	505-60

15. Supplementary Notes

Point of Contact: Reese L. Sorenson, Ames Research Center, M/S 202A-14
Moffett Field, CA 94035 (415) 694-6747 or FTS 464-6747

16. Abstract

An elliptic method for generating composite grids about realistic aircraft is presented. A body-conforming grid is first generated about the entire aircraft by the solution of Poisson's differential equation. This grid has relatively coarse spacing, and it covers the entire physical domain. At boundary surfaces, cell size is controlled and cell skewness is nearly eliminated by inhomogeneous terms, which are found automatically by the program. Certain regions of the grid in which high gradients are expected, and which map into rectangular solids in the computational domain, are then designated for zonal refinement. Spacing in the zonal grids is reduced by adding points with a simple, algebraic scheme. Details of the grid-generation method are presented along with results of the present application, a wing/body configuration based on the F-16 fighter aircraft.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Grid Generation Elliptic Solver Three-Dimensional Grids	Unlimited Subject Category - 01

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	21	A02

*For sale by the National Technical Information Service, Springfield, Virginia 22161

End of Document