NASA Technical Memorandum 101046

Surface Grid Generation for Complex Three-Dimensional Geometries

Raymond Ching-Chung Luh

(NASA-TM-101046) SURFACE GRID GENERATION FOR COMPLEX THREE-DIMENSIONAL GEOMETRIES (NASA) 15 p CSCL 20D N89-13747

Unclas

G3/34 0177655

October 1988

National Aeronautics and
Space Administration

NASA Technical Memorandum 101046

Surface Grid Generation for Complex Three-Dimensional Geometries

Raymond Ching-Chung Luh, Ames Research Center, Moffett Field, California

October 1988

NASA
National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035

SUMMARY

This paper presents an outline for the creation of surface grids from primitive geometry data such as obtained from CAD/CAM systems. The general procedure is applicable to any geometry including full aircraft with wing, nacelle, and empennage. When developed in an interactive graphics environment, a code based on this procedure is expected to substantially improve the turn-around time for generating surface grids on complex geometries. Results are shown for a generic hypersonic airplane geometry.

1. INTRODUCTION

With the advances achieved in supercomputer technology and computational fluid dynamics(CFD) over the past number of years, the computation of flow fields around complex three-dimensional geometries has become a greater focus of attention. However, it is increasingly becoming apparent that one of the real obstacles to obtaining flow field solutions is the lack of means to produce high quality grids in a fast and efficient manner. As geometries become ever more complex, such as a full fighter aircraft configuration, the task of grid generation is further complicated by the need to define grids on all surfaces before any attempt to grid the flow field domain can be made. This is because the geometry data one obtains is usually not in a form that the grid generator can readily use. It is common practice for researchers to put together ad hoc generators for their particular surface gridding requirements. As long as the geometries are at a low level of complexity such as bodies of revolution or analytically defined surfaces, the time penalty for such extra effort is not severe. However, this practice is fast becoming obsolete or intractable because of the taxing demands on CFD analysts to obtain solutions, usually not without time constraints, to numerous realistic configurations which are inevitably complex in nature.

Although three-dimensional grid generators abound, users of these codes have found that they are often slowed down because substantial computations external to the codes are necessary either before or during the surface grid generation process. This slowdown is due to the fact that none of these codes are set up to deal with data bases that are frequently obtained from CAD/CAM systems. One has to essentially *prepare* a data base to the level of a two-dimensional array of surface points. But this preparation work is more than likely to take up about 70% of the total effort. It is thus not surprising that many man-months of efforts are common place for generating surface grids associated with complex geometries.

The present work specifically addresses the needs of surface grid generation beginning from the primitive data bases. The procedure allows the systematic processing of primitive or raw geometry data into distributed zonal surface patch grids. This procedure introduces the idea of grid construction from local surface patches — a means by which complex surface geometries are broken down into smaller but simpler ones without artificially creating discontinuities in the grids at any of the patch boundaries. Surface patches are of course very basic to the composite block structure of three-dimensional grids (ref. 1) and the patched mesh systems in two dimensions (ref. 2). The construction or conception of any block structure is highly dependent on the surface geometry and in three dimensions is a difficult process in itself. For the purpose of surface grid generation, it suffices to assume that a global block structure is already setup so that the local surface patching strategy is transparent. The present work then provides the means for an analyst to take the raw geometry data and create the required patched surface grids as quickly as possible.

Section 2 deals with the processing of primitive geometry data which is also the most time consuming portion. The resulting preliminary or defining surface grid with desired patch structure is then redistributed, and reassembled if necessary. This is discussed in Section 3 along with the subject of surface intersections. A code based on the present method has been developed in an interactive environment with some graphical display capabilities. Some results for a generic hypersonic airplane are shown in Section 4.

2. GEOMETRY DATA PROCESSING

The geometry data for a realistic or complex configuration are most likely extracted from some CAD/CAM systems for high quality representation, usually in a series of cross sections, i.e. one-dimensional arrays of x, y, and z coordinates. These data are nevertheless primitive or raw because of a number of deficiencies. The fuselage, for instance, may be represented by point sets or distributions that vary widely among the cross sections, or the wing-body intersections may be undefined. In addition, the data must be grouped into a patch network that is the basis for a global block structure before it is ready for a surface grid generator.

In the present work, all the pre-processing required as implied above except surface intersection can be performed by a geometry data processor (GDP). A geometry data processor works like a data editor performing such functions as inserting, deleting, enriching, and redistributing of data points. It can also segment a sectional curve into any number of pieces. These functions are accomplished by interpolating on curve data fitted with piecewise parametric cubic polynomials. Consider a parameter s which varies from 0 to 1 between two points of a curve data set. A point $\vec{Q}(s)$ is then defined as

$$\vec{Q}(s) = [F(s)][b]^T \tag{1}$$

where the blending functions $F(s)$ are the cubic Hermite basis functions defined as

$$\begin{aligned} F_1(s) &= 1 - s^2(3 - 2s) \\ F_2(s) &= s^2(3 - 2s) \\ F_3(s) &= s(s-1)^2 \\ F_4(s) &= s^2(s-1) \end{aligned} \tag{2}$$

and the boundary condition matrix $[b]$ is given as

$$[b] = (\vec{Q}_0 \;\; \vec{Q}_1 \;\; \vec{Q}_{s,0} \;\; \vec{Q}_{s,1}) \tag{3}$$

where $\vec{Q}_0$ and $\vec{Q}_{s,0}$ stand respectively for $\vec{Q}(s)$ and its first derivative evaluated at $s = 0$, etc. The two end points are given but the derivatives must be evaluated from the entire curve set. Let S be the arc-length parameter over the entire data set. Observing s at the two ends to be the equivalent of indices, the derivative is simply

$$\vec{Q}_s = \frac{\partial \vec{Q}}{\partial S}\frac{\partial S}{\partial s} \tag{4}$$

The accuracy in interpolation depends largely on the partial derivatives in (4) but their evaluation is not unique. The method developed by Akima (ref. 3) for interpolating on a curve has been found to be very reliable and has been adapted here. Its advantage is that with the use of weighted average of slopes of four adjacent line segments, the method is independent of interval widths. However, the method is applicable at the extremes of the data set only if points are available outside the range. This requires extrapolation which can be rather unreliable especially when point spacings vary wildly. An imaging technique has been developed here whereby the arc-length parameter is imaged outside the range and three-point extrapolations applied.

For a given arc-length location S, the segment containing it provides the boundary conditions in (3), the local parameter s which is assumed to have a linear relationship with S is used in (2), and then the corresponding coordinate values on the curve $\vec{Q}(s) = Q(x, y, z)$ are calculated by (1). Figure $1(a)$ shows a typical curve with arbitrarily distributed data points. The geometry data processor utilizes Vinokur's (ref. 4) two-sided stretching function to obtain distribution functions based on arc-length. Figure 1(b) shows the result for an enriched uniform distribution. A close-up view of the corner with high curvature shown in figure 2(a) reveals the necessity to cluster points locally. This can be accomplished by breaking up the original curve into two or more segments to permit specification of interior spacings. Segmenting of curves is required for the setup of a surface patch network and must be predetermined by closely examining the data base. The insert mode allows for adding points to the curve at user specified locations. For example, several points are inserted within the curved region of the original curve as shown in figure 2(b).

3. SURFACE GRID GENERATION

Once the geometry data processing is completed, the data output is ready for surface grid generation. In addition to having zonal properties, the processed surface data is represented by L surface patches, each containing M by N points with M and N varying from patch to patch. The surface grid generator (SGG) then interpolates each patch to m by n points with distribution functions prescribed on its boundaries. The parameters m and n again depend on the needs of the grid generator and vary between patches. The method of curve interpolation used in section 2 can be extended to the surface case which becomes parametric bicubic interpolation. It is a special case of the Coon's patch method (ref. 5) which takes the boolean sums of the prescribed boundary curves. Consider a quadrilateral element of the patch data set parametrized by u and v from 0 to 1. A point $\vec{P}(u,v)$ is defined as

$$\vec{P}(u,v) = [F(u)][B][F(v)]^T \tag{5}$$

where again $F(u)$ and $F(v)$ are given by (2) and the boundary condition matrix $[B]$ takes the form

$$[B] = \begin{pmatrix} \vec{P}_{00} & \vec{P}_{01} & \vec{P}_{v,00} & \vec{P}_{v,01} \\ \vec{P}_{10} & \vec{P}_{11} & \vec{P}_{v,10} & \vec{P}_{v,11} \\ \vec{P}_{u,00} & \vec{P}_{u,01} & \vec{P}_{uv,00} & \vec{P}_{uv,01} \\ \vec{P}_{u,10} & \vec{P}_{u,11} & \vec{P}_{uv,10} & \vec{P}_{uv,11} \end{pmatrix} \tag{6}$$

where $\vec{P}_{uv,01}$ stands for the partial derivatives of $\vec{P}$ with respect to u and v evaluated at $u = 0, v = 1$, etc. The success of this method again depends entirely on the calculation of the derivative terms in the boundary condition matrix. Experience by others has shown that inaccuracies can cause oscillatory problems (ref. 6). Ferguson (ref. 7) does not show how the derivative terms are obtained and excludes the twist or cross derivative terms. Here, Akima's (ref. 8) method has been adapted as in the cubic curve case. If U and V are the normalized arc-length parameters in the u and v directions respectively, then

$$\vec{P}_u = \frac{\partial \vec{P}}{\partial U}\frac{\partial U}{\partial u} + \frac{\partial \vec{P}}{\partial V}\frac{\partial V}{\partial u} \tag{7}$$

and other terms follow similarly. This is a vast improvement over Luh (ref. 9) where U and V are point index rather than arc-length parameters.

Intersections of two surfaces can be built up from intersections of a curve and a surface. Following a particular curve on one of the surfaces (starting surface), it intersects the other (target surface) when the distance

$$\vec{R}(u,v,s) = \vec{P}(u,v) - \vec{Q}(s) \tag{8}$$

is zero. The three unknowns in three equations above can be solved by the Newton-Raphson method. Thus

$$\vec{R}_i \Delta t_i = -\vec{R} \tag{9}$$

where t_i are (u, v, s) for $i = 1,2,3$, respectively, and $\vec{R}_i = \partial\vec{R}/\partial t_i$ are obtained by differentiating the basis functions in (1) and (5). Convergence is usually obtained within 3 to 4 iterations.

The following example demonstrates the basic functions of SGG. Figure 3(*a*) shows a vertical surface with 8 by 5 points intersecting a horizontal surface with 21 by 5 points. The common intersecting curve shown in figure 3(*b*) is found by interchanging the roles of either surface between that of starting and target. Observe how the new surfaces retain the same number of points as the original ones. Figure 3(*c*) shows the two surfaces enriched with locally clustered distributions that match up at the intersection.

4. APPLICATIONS

A GDP and SGG have been combined into an integrated surface grid generation package and applied to a number of complex geometries. The initial data base for a configuration is usually given in a number of cross sections which must be closely examined. A surface patching strategy for building the global blocking structure emerges to guide the grid generation process. As an example, figure 4 shows the cross sections that are interfaces to distinct geometry zones for a generic hypersonic aircraft. For a flow solver set up to handle general zonal grids, one may choose a patch network as shown in figure 5(*a*). Besides the singularity at the nose, the surface grid resulting from this patching strategy would require the handling of four more singularity points where a grid line diverges to or converges from a multiplicity of lines. Figure 5(*b*) shows the preliminary surface grid after processing the primitive data with GDP. Notice the arbitrary distribution in the axial direction which is not altered. The work required in arriving at this surface defining two-dimensional arrays of points is perhaps 70% or more of the entire surface grid generation effort. The SGG then interpolates the surfaces to desired grid dimensions and distributions such as shown in figure 5(*c*).

An alternative patching strategy shown in figure 5(*d*) has only the nose singularity and requires no special handling at zonal boundaries. This requires, for the notch, approximating its ends which are points as finite in length so that grid lines can be carried across the patch boundaries. No special treatment is needed in removing the singularities in the back because grid lines can be made to wrap around the highly curved but nondiscontinuous areas with appropriate point distributions. Figures 6(*a*) and 6(*b*) show close up views in the vicinity of the notch origin for the two cases. The non-singular case shown requires 18% more grid points which in turn requires corresponding increase in computation time.

The blunt nose is an important region in the hypersonic flow regime because of the extreme high gradients. The data base defines it very roughly with just four sections as shown in figure 7(*a*). After interpolating to 25 sections with a spacing of 3.0E-6 at the nose, figure 7(*b*) shows a very smooth representation.

5. CONCLUDING REMARKS

A method to quickly and efficiently generate surface grids has been presented. It consists of two steps: geometry processing and grid generation. The geometry processing edits the primitive data and prepares it into a number of surface patches. The surface grid generation is then accomplished by interpolating each patch to desired points and distributions. An interactive code based on the method has been developed with some static graphics capability. Results for the generic hypersonic aircraft shown require roughly one week of intense effort. Further improvement in turn-around time can be achieved in a workstation environment with multi-window and dynamic graphics display capabilities.

REFERENCES

1. Thompson, J. F.: A Survey of Composite Grid Generation for General Three-Dimensional Regions. Numerical Methods for Engine-Airframe Integration, Ed. Murthy, S.N.B and Paynter, G.C., AIAA, 1986.

2. Luh, R. C.-C. et al.: FASTWO - A 2-D Interactive Algebraic Grid Generator. AIAA-88-0516, 1988.

3. Akima, H.: A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures. J. of the ACM, vol. 17, no. 4, 1970, pp. 589-602.

4. Vinokur, M.: On One-Dimensional Stretching Functions for Finite Difference Calculations. NASA CR-3313, 1980.

5. Coons, S. A.: Surfaces for Computer Aided Design of Space Forms. MAC-TR-41, 1967.

6. Barnhill, R. E.: Surfaces in Computer Aided Geometric Design: A Survey with New Results. CAGD, vol. 2,1985, pp. 1-17.

7. Ferguson, J.: Multivariable Curve Interpolation. J. of the ACM, vol. 11, no. 2, 1964, pp. 221-228.

8. Akima, H.: A Method of Bivariate Interpolation and Smooth Surface Fitting Based on Local Procedures. Comm. of the ACM, vol. 17, no. 1, 1974, pp. 18-20.

9. Luh, R. C.-C. et al.: Algebraic Grid Generation in Patched Mesh Systems. Presented at the SIAM Conference on Applied Geometry, July 1987, Albany, New York.

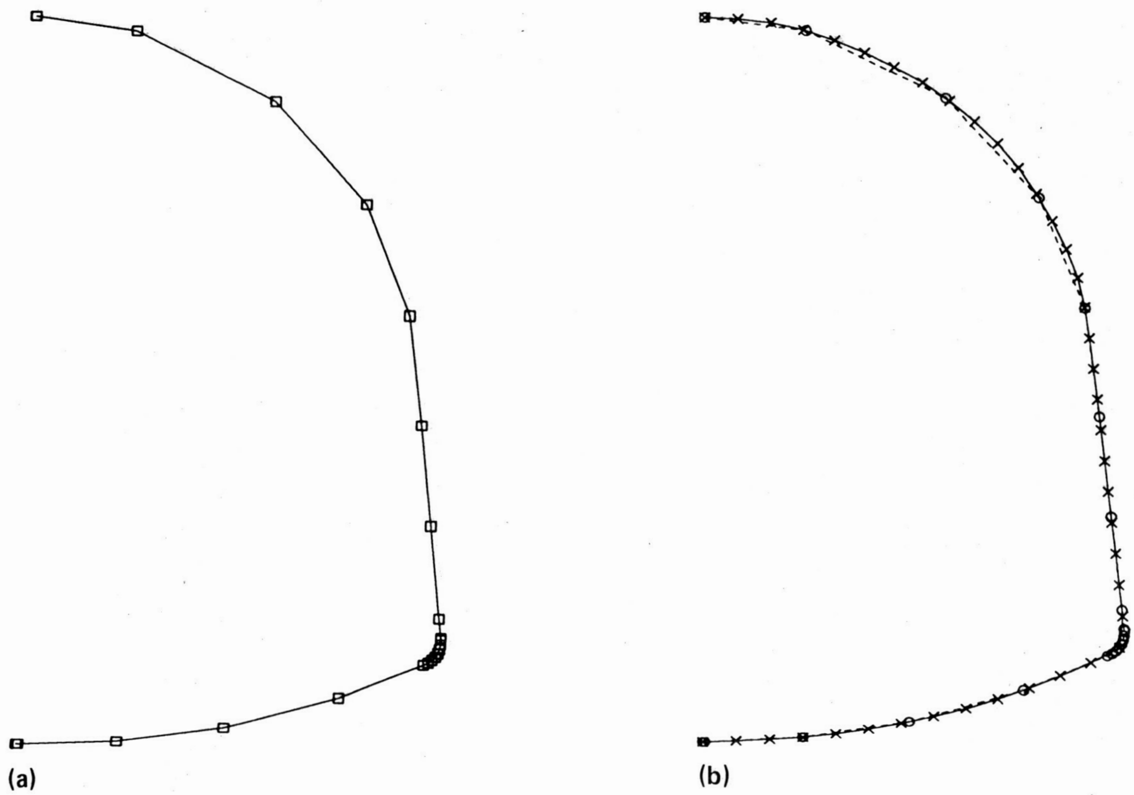


Figure 1.– (*a*) A curve set, (*b*) enriched with uniform distribution.

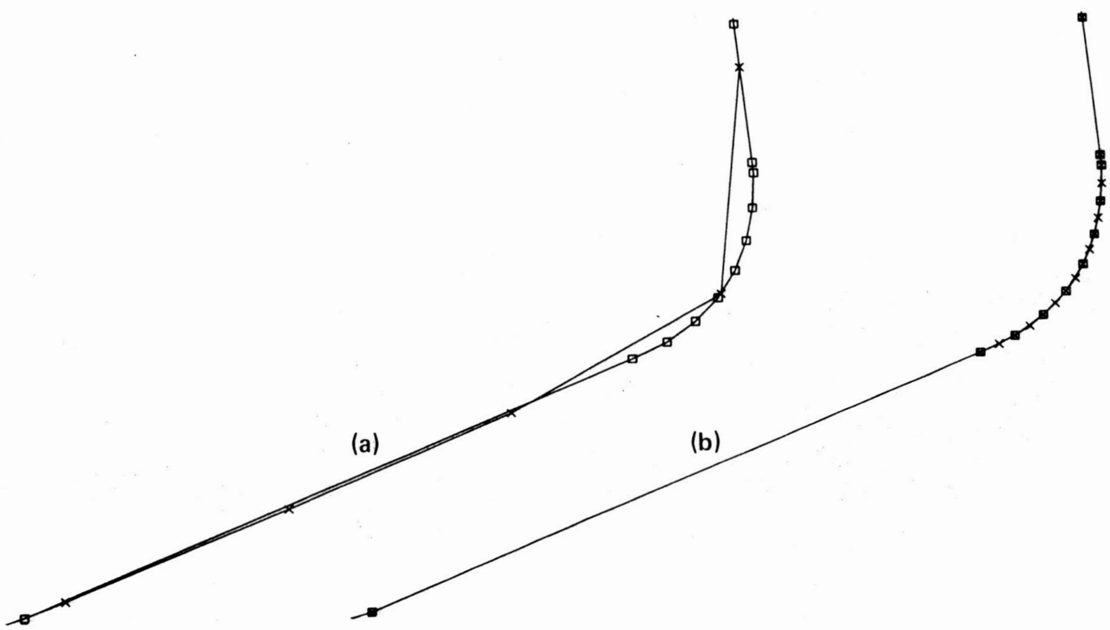


Figure 2.– (*a*) Enlarged view of high-curvature region, (*b*) points locally inserted to original curve.

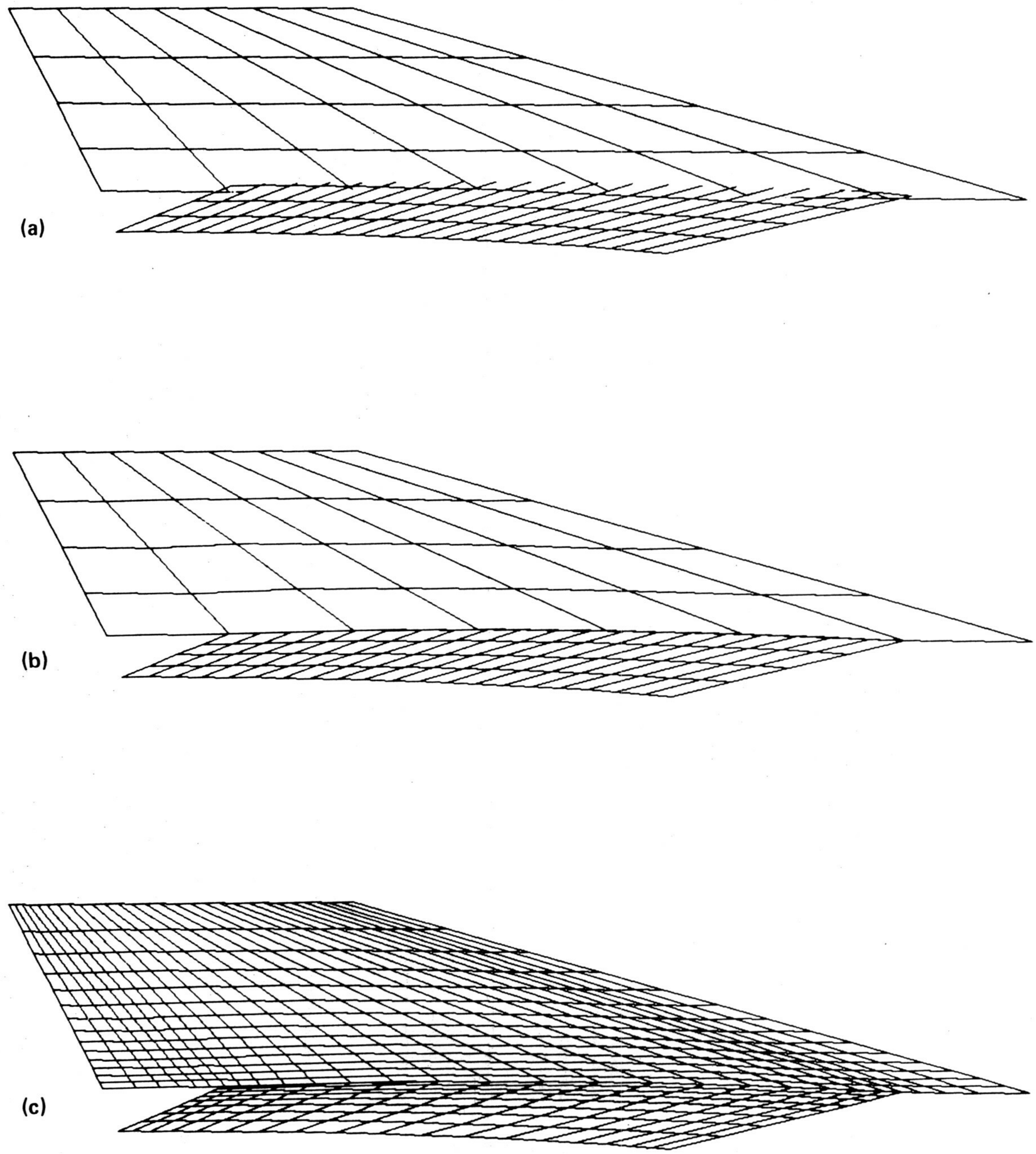


Figure 3.– Two surfaces (a) with unknown intersecting curve, (b) intersecting curve defined, and (c) enriched and matched up at intersection.

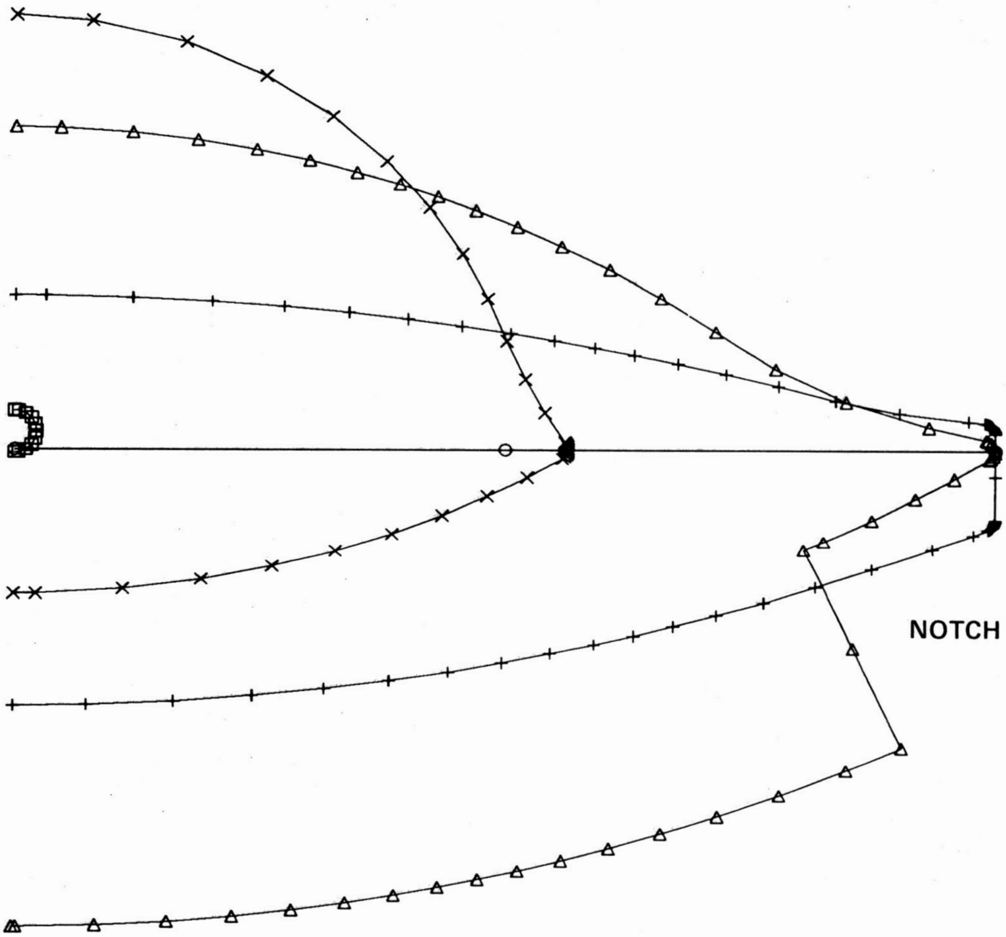


Figure 4.– Cross section interfaces to distinct geometry zones.

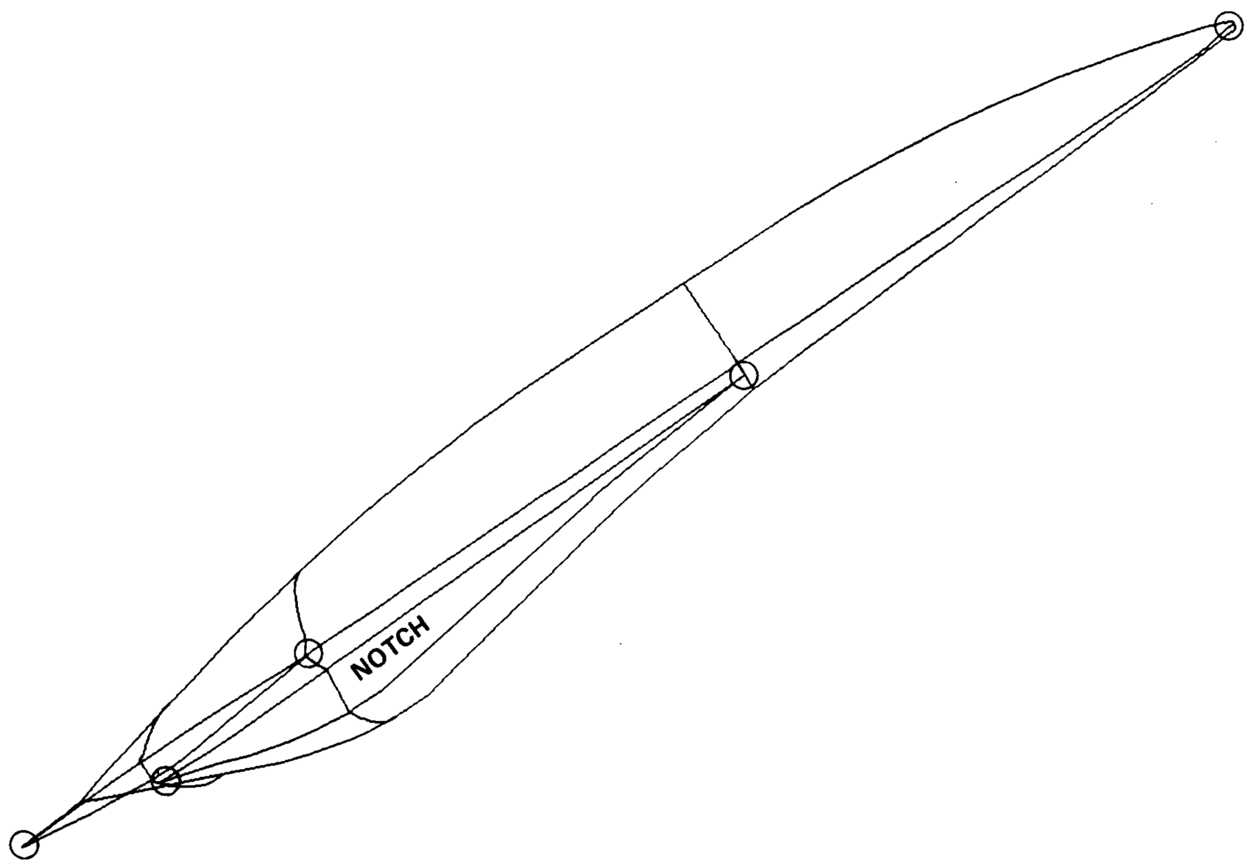


Figure 5.– (a) Patch network with 5 singular points.

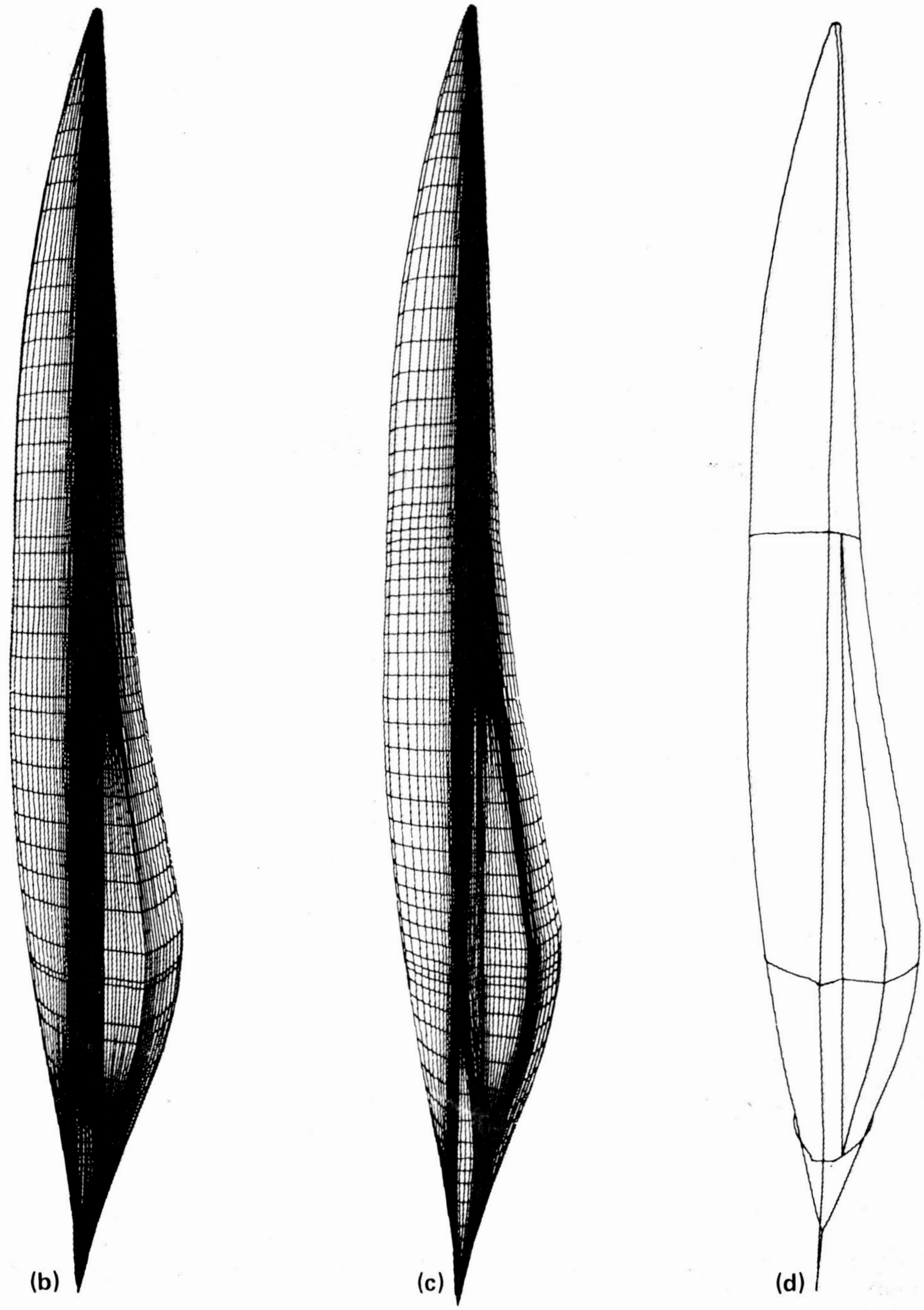


Figure 5.– (*b*) Preliminary surface grid, (*c*) enriched with local clustering, (*d*) patch network with singularity only at the nose.

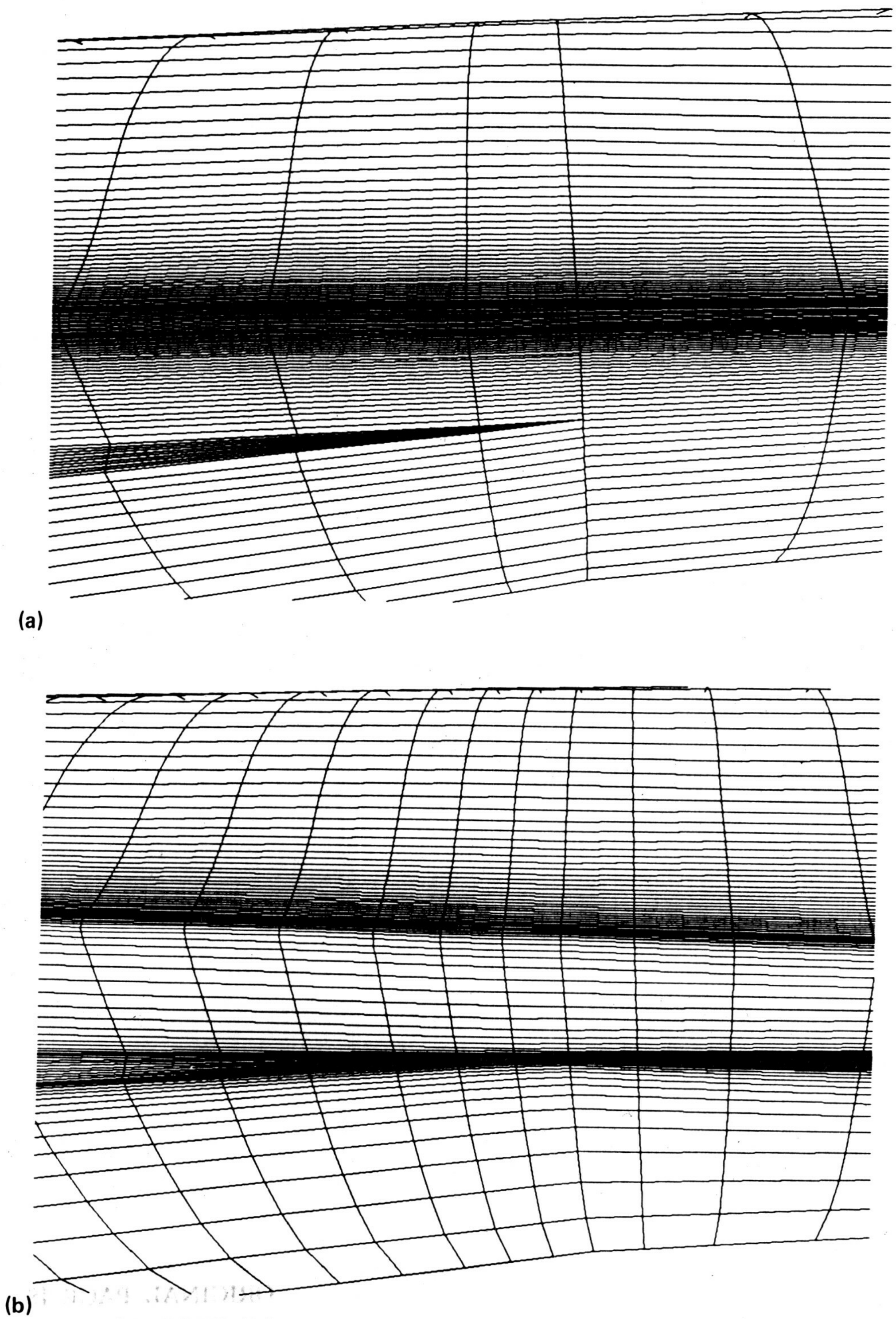
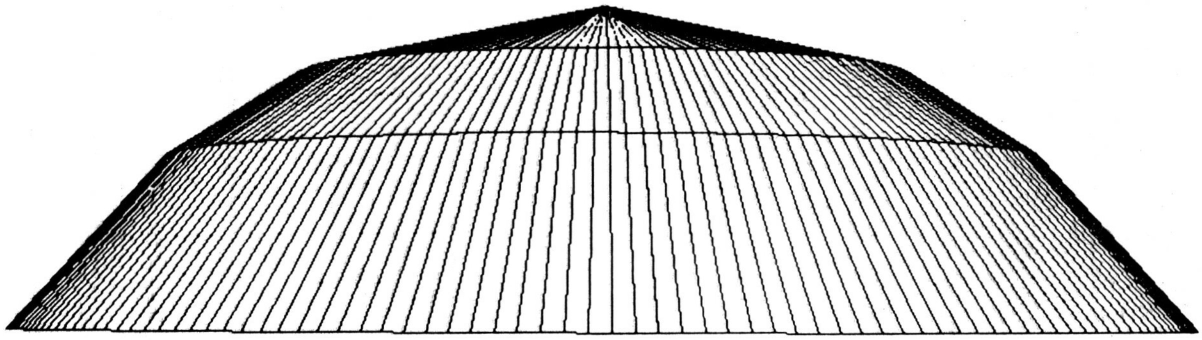
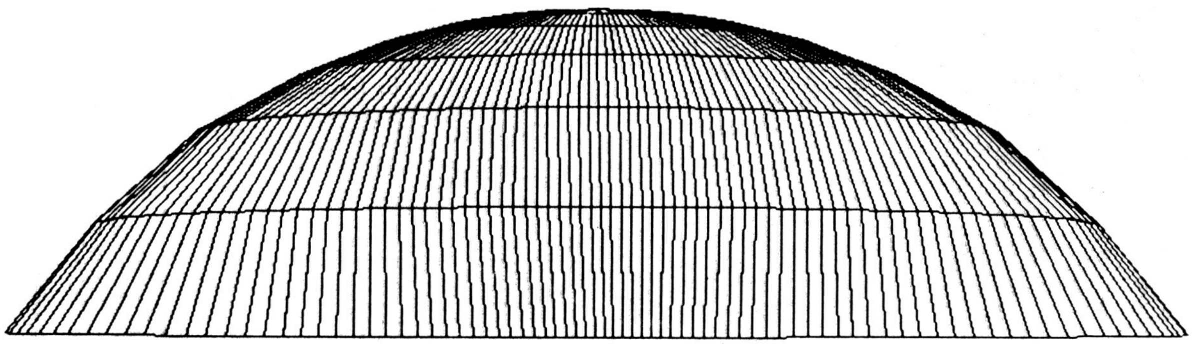


Figure 6.– Enlarged view near the beginning of notch, (a) singular case, (b) non-singular case.



(a)

Figure 7.– (*a*) Blunt nose defined by four sections.



(b)

Figure 7.– (*b*) Blunt nose interpolated to 25 sections.

ORIGINAL PAGE IS
OF POOR QUALITY

Report Documentation Page

National Aeronautics and Space Administration

1. Report No. NASA TM 101046	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Surface Grid Generation for Complex Three-Dimensional Geometries		5. Report Date October 1988
		6. Performing Organization Code
7. Author(s) Raymond Ching-Chung Luh		8. Performing Organization Report No. A88318
		10. Work Unit No. 505-60
9. Performing Organization Name and Address Ames Research Center Moffett Field, CA 94035		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Point of Contact: Raymond Ching-Chung Luh, Ames Research Center, MS 258-1
Moffett Field, CA 94035 (415) 694-4494 or FTS 464-4494

16. Abstract

This paper presents an outline for the creation of surface grids from primitive geometry data such as obtained from CAD/CAM systems. The general procedure is applicable to any geometry including full aircraft with wing, nacelle, and empennage. When developed in an interactive graphics environment, a code based on this procedure is expected to substantially improve the turn-around time for generating surface grids on complex geometries. Results are shown for a generic hypersonic airplane geometry.

[illegible] WINGS
TAIL ASSEMBLIES
NACELLES
[illegible]
AIRCRAFT CONFIGS.

17. Key Words (Suggested by Author(s)) Surface grid generation Bi-cubic interpolation	18. Distribution Statement Unclassified-Unlimited Subject Category - 34

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	16	A02

NASA FORM 1626 OCT 86

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