Algebraic Grid Generation about Wing-Fuselage Bodies

R. E. Smith,* R. A. Kudlinski,† and E. L. Everton‡
NASA Langley Research Center, Hampton, Virginia
and
M. R. Wiese§
Computer Sciences Corporation, Hampton, Virginia

An algebraic procedure for generating boundary-fitted grids about wing-fuselage configurations is presented. A wing-fuselage configuration consists of two aircraft components specified by cross sections and mathematically represented by Coons' patches. Several grid blocks are constructed to cover the entire region surrounding the configuration, and each grid block maps into a computational cube. Grid points are first determined on the six boundary surfaces of a block and then in the interior. Grid points on the surface of the configuration are derived from the intersection of planes with the Coons' patch definition. Approximate arc length distributions along the resulting grid curves concentrate and disperse grid points. The two-boundary technique and transfinite interpolation are used to determine grid points on the remaining boundary surfaces and block interiors.

Introduction

An important aspect of simulating fluid motion about complex aerodynamic bodies is grid generation. For creating grids about wing-fuselage bodies, an algebraic approach based on Coons' patch surface definition,[1] the two-boundary grid-generation technique,[2-4] and transfinite interpolation[5-7] is presented. The process is divided into five steps:

1) Define the wing and fuselage surfaces and their intersection.
2) Determine grids on the wing and fuselage surfaces.
3) Define the remaining exterior boundary surfaces.
4) Determine grids on the exterior boundary surfaces.
5) Determine the interior grid.

This approach combines with algebraic grid generation aircraft-surface definition concepts that have been developed over many years. For the wing-fuselage grids described herein, four adjoining grids are specified. They are the top-front, bottom-front, top-back, and bottom-back grids; and the step-by-step demonstration of the grid-generation procedure is done with the top-front grid.

Wing-Fuselage Surface Definition

The components of an aircraft can be described by the Harris geometry[8] in terms of cross sections. A fuselage is described by cross sections along the x-body axis, and a wing is described as airfoil sections in the spanwise z direction. In turn, each airfoil section is defined by the coordinates of the camber line and Δy coordinates. An orthographic view of the defining data describing a wing-fuselage configuration is shown in Fig. 1. Cubic splines are fit along and across the specified cross-sectional data for each component.[9] The specified positional and derivative data (obtained from the spline fits) provide corner parameters for a Coons' patch surface definition for a component. Each patch is of the form

$$V(u,w) = UCBC^T W^T \tag{1}$$

where

$$V(u,w) = [x(u,w)\ y(u,w)\ z(u,w)]$$

$$B = \begin{bmatrix} V(0,0) & V(0,1) & V_w(0,0) & V_w(0,1) \\ V(1,0) & V(1,1) & V_w(1,0) & V_w(1,1) \\ V_u(0,0) & V_u(0,1) & V_{uw}(0,0) & V_{uw}(0,1) \\ V_u(1,0) & V_u(1,1) & V_{uw}(1,0) & V_{uw}(1,1) \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$U = [u^3\ u^2\ u\ 1], \qquad W = [w^3\ w^2\ w\ 1]$$

$$0 \le u \le 1, \qquad 0 \le w \le 1$$

The vector-valued function $V(u,w)$ represents the Cartesian coordinates (x,y,z) for a patch definition, and the variables u and w are independent parametric variables. The subscripts u and w matrix B denote differentiation with respect to the parametric independent variables. The patch description is a bicubic representation of the region bounded by the sides of the patch. An ordered set of patches represents the surface of a component, and Ref. 9 describes a widely available computer program for performing the spline fits, saving the patch description, and plotting enriched surface components (Fig. 2).

Wing-Fuselage Intersection

In order to map the wing and fuselage surfaces into two adjoining planar surfaces in a computational coordinate system (Fig. 3), the intersection of the two components must be found. The process we propose is to find the intersection of a sequence of planes with the wing and fuselage components, and then compute the intersection of the planar

Received Oct. 15, 1986; revision received March 27, 1987.

*Senior Research Engineer, Computer Applications Branch. Member AIAA.
†Research Mathematician, Computer Applications Branch.
‡Research Engineer, Flight Software and Graphics Branch.
§Senior Member of Technical Staff.

curves.[10] The reason for taking this approach is that the intersection of arbitrarily defined planes with the components of a configuration is an existing capability of the surface-definition program described in Ref. 9. The formulation of the plane-patch intersection is obtained by rewriting Eq. (1) as

$$x = UCB(x)C^T W^T$$

$$y = UCB(y)C^T W^T$$

$$z = UCB(z)C^T W^T$$

and substituting the comonents in the equation of a plane

$$ax + by + cz + d = 0$$

where a, b, c, and d are constants derived from a three-point

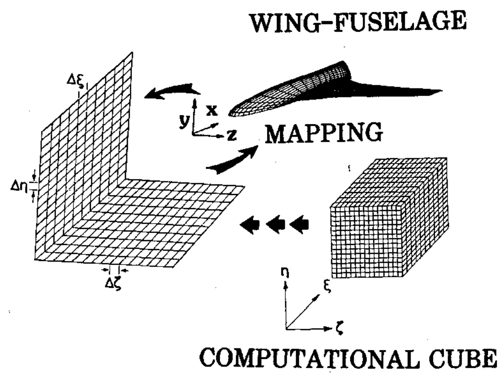

Fig. 3 Wing-fuselage mapping to the computational domain.

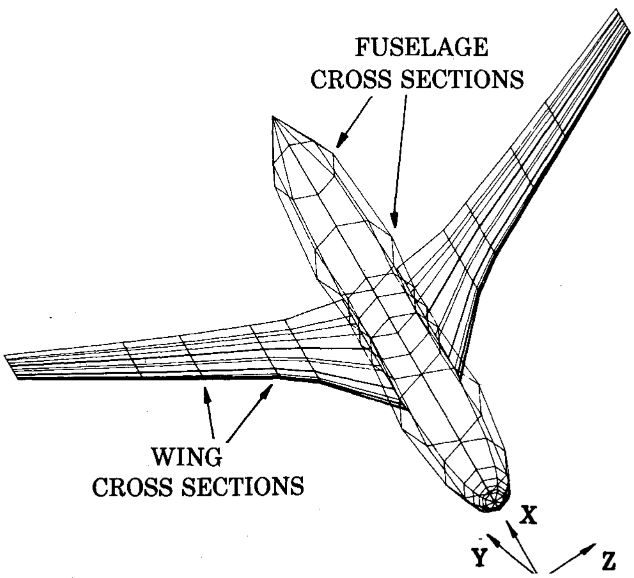

Fig. 1 Initial definition of a wing-fuselage configuration.

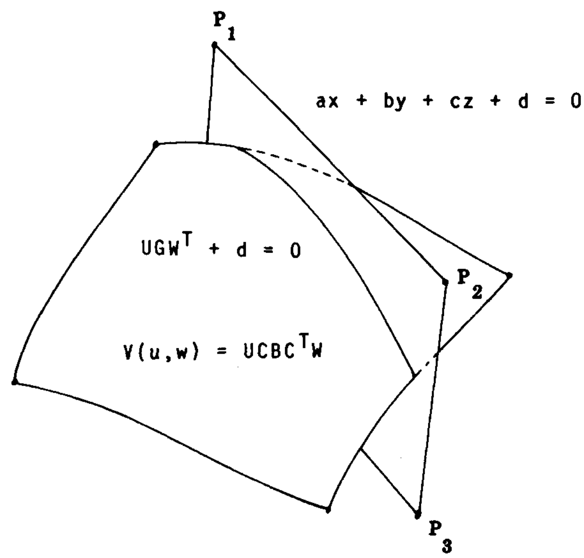

Fig. 4 Patch definition and patch-plane intersection.

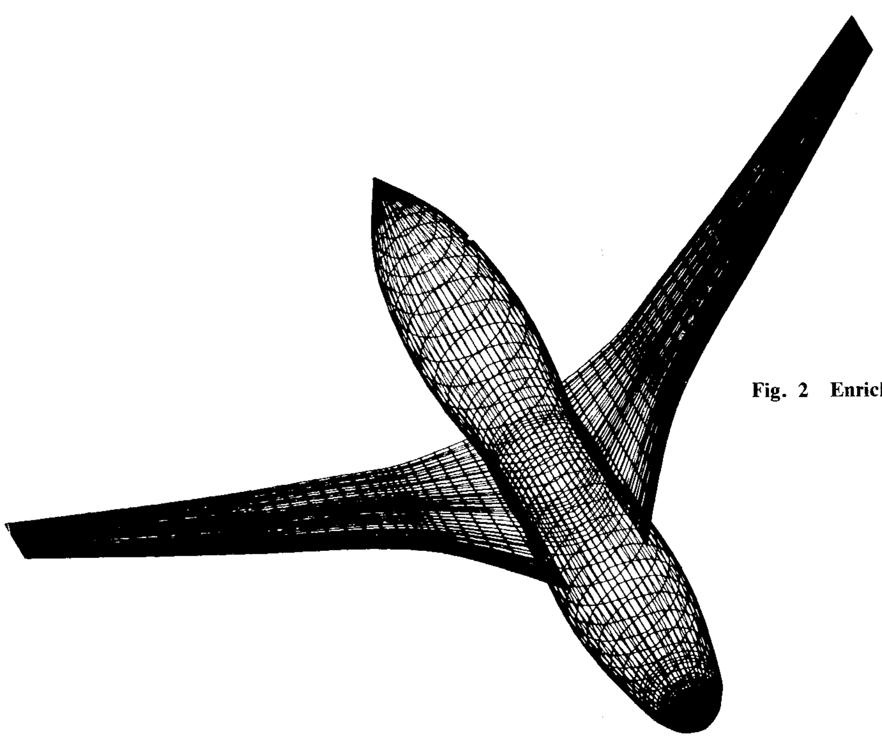

Fig. 2 Enriched plot of the wing-fuselage configuration.

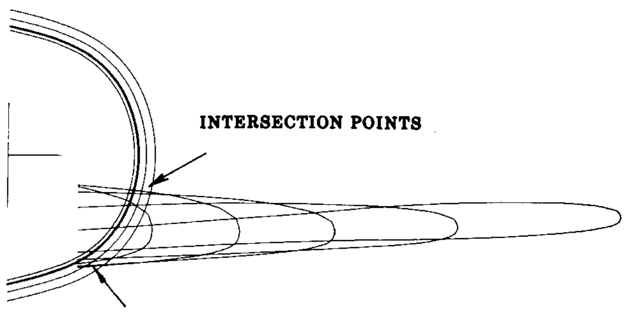

Fig. 5 Intersection of planes orthogonal to the x-axis with the wing and fuselage.

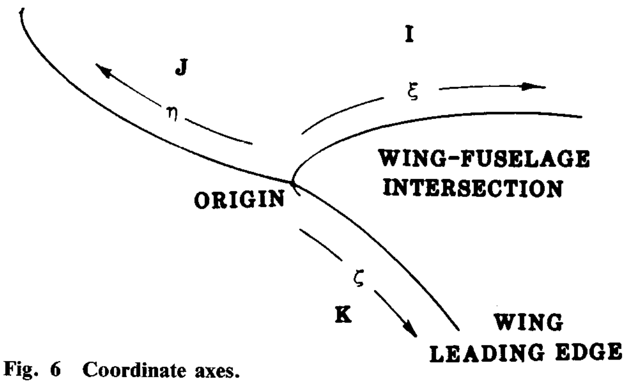

Fig. 6 Coordinate axes.

description. This leads to the equation

$$UGW^T + d = 0 \tag{2}$$

where

$$G = aCB(x)C^T + bCB(y)C^T + cCB(z)C^T$$

is a matrix of constants. Assigning a value to one of the parametric variables u or w results in Eq. (2) becoming a cubic polynomial with respect to the other parametric variable. The real root of Eq. (2), interior to the patch boundaries for a sequence of values of the specified paramteric variable, defines a patch-plane intersection curve (Fig. 4).

The intersection of a wing-surface component and fuselage-surface component is found by first determining the intersection of the patch definition of each component with a sequence of planes along, and orthogonal to, the x-body axis (Fig. 5). This is accomplished by using the computer code described in Ref. 9 as a kernel that is driven by another code (driver program), which specifies the planes, orders the coordinates, deletes multiple points at the intersection of neighboring patches, and searches to find the intersection of the planar curves from the two components. The leading and trailing points of the wing-fuselage intersection occur where there is only one intersection point on the plane-patch intersection curves. A search is performed by the driver code to define these terminating points accurately.

Coordinate Axes, Grids, and Indices

For the grids described herein, the physical point corresponding to the origin of the computational coordinate system is the leading point of the wing-fuselage intersection. The wing-fuselage intersection curve corresponds to the ξ axis. A curve that extends from the leading wing-fuselage intersection point to the leading nose point on the fuselage corresponds to the η axis, and a curve along the leading edge of the wing from the wing-fuselage intersection point to the wing tip corresponds to the ζ axis (Fig. 6). A grid is represented by $F_i(I,J,K)$, where

$$F_i(I,J,K) = [X_i(I,J,K) \; Y_i(I,J,K) \; Z_i(I,J,K)]$$

and $I = 1,2,...L$, $J = 1,2,...M$, and $K = 1,2,...N$. The top-front fuselage grid is denoted by $F_1(I,J,K)$. The computational coordinates are related to indices I, J, and K by

$$\xi = (I-1)/(L-1)$$

$$\eta = (J-1)/(M-1)$$

$$\zeta = (K-1)/(N-1)$$

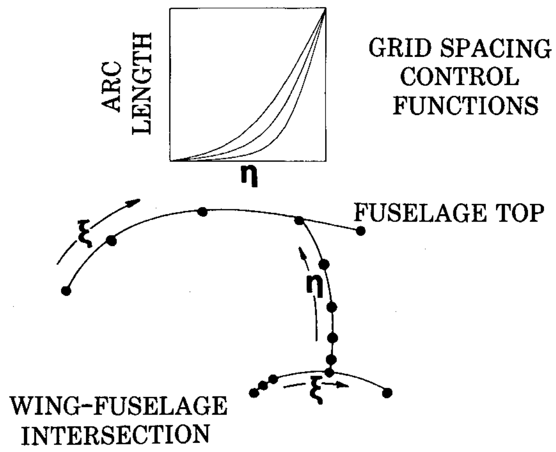

Fig. 7 Boundary curves for fuselage grid definition.

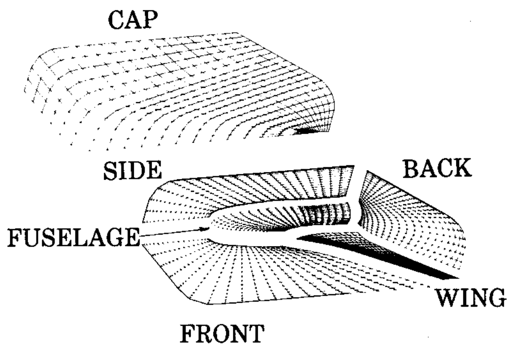

Fig. 8 Top-front boundary surface grids.

Fuselage Grid

The fuselage grid computation is preceded by the computation of a distribution of grid points (Fig. 7) on the wing-fuselage intersection curve $\eta = 0(F_1(I,1,1)$, $I = 1,2,...L)$ and along a curve on top of the fuselage at $\eta = 1(F_1(I,M,1)$, $I = 1,2,...L)$. The computational ξ coordinate is uniformly discretized on the unit interval $0 \leq \xi \leq 1$, and the ξ coordinate is related by one-to-one functions to the normalized approximation arc length along the wing-fuselage intersection and top boundary curves. The physical coordinates of the intersection curve and the top boundary curve are related to the normalized approximate arc lengths by linear interpolation into previously stored tables of coordinates vs arc lengths. Grid spacing[3,4,11] in the physical domain is controlled by single-valued functions relating computational

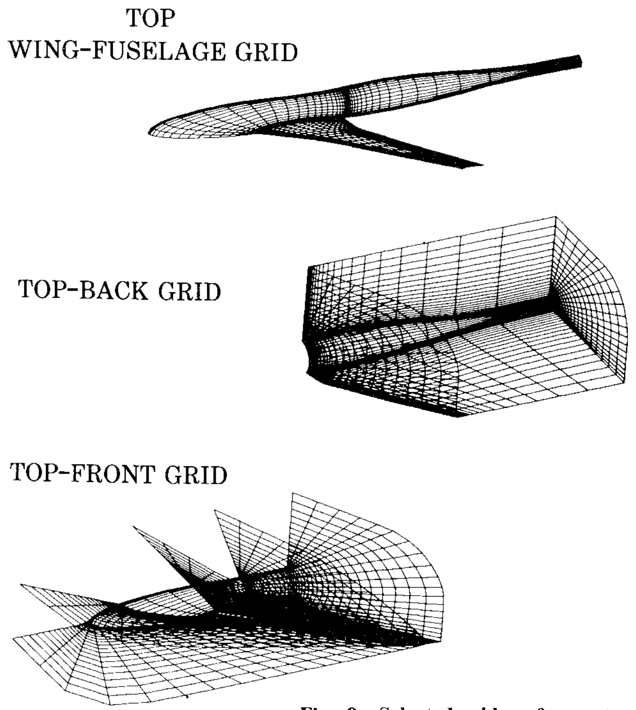

Fig. 9 Selected grid surfaces—top.

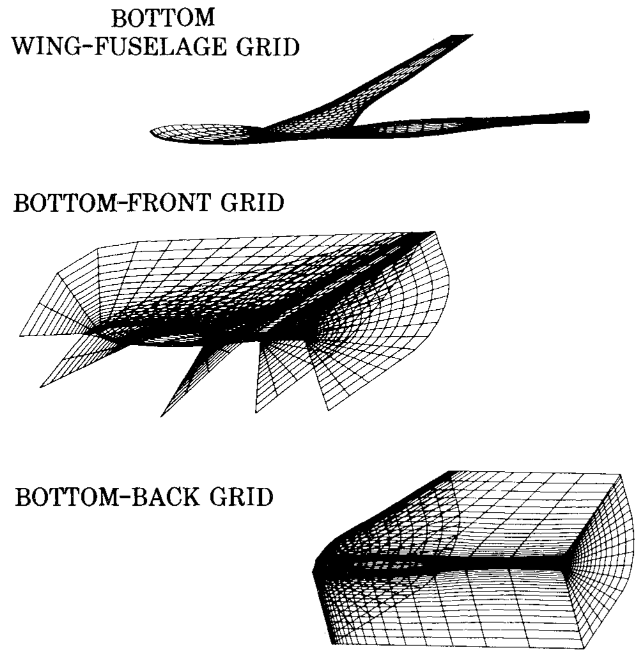

Fig. 10 Selected grid surfaces—bottom.

coordinates to normalized approximate arc lengths. Low slopes in these functions correspond to a physical grid concentration, and high slopes correspond to physical grid dispersions.

Having the distributions of grid points along the wing-fuselage intersection curve and the fuselage-top boundary curve, grid curves (dense sets of points) are computed on the fuselage surface and stored. This is done with Coons' patch surface representation of the fuselage and plane-patch intersection capability described earlier. In this case, the planes are defined by three points of which two are on the boundary curves at $\eta = 0$ and $\eta = 1$. The third point is defined to be on the x-body axis with the x coordinate equal to the x coordinate of the wing-fuselage grid point. With the computer program described in Ref. 8 as a kernel, a driver program specifies the points defining the plane, orders the plane-patch intersection points, deletes multiple points, computes the approximate arc length along the curves from the wing-fuselage intersection boundary to the fuselage-top boundary, and stores the coordinates of the curves as a function of the normalized approximate arc length. The distribution of the grid points in the η direction is obtained in the same manner as previously discussed for the wing-fuselage intersection and the top boundary curve. Figure 8 shows a fuselage grid with concentration toward the leading wing-fuselage intersection curve. Figure 9 shows the same configuration but with concentration toward both the leading and trailing wing-fuselage intersection points. The fuselage surface grid is represented by $F_1(I,J,1)$, $I = 1,2,\ldots L$, $J = 1,2,\ldots M$.

Wing Grid

The plane-patch intersection capability and arc-length spacing-control functions are again used to determine the wing grid. Also, the concept of transfinite interpolation[5-7] is used to conform the grid points near the wing-fuselage intersection to the shape of the wing-fuselage intersection curve. The process is to establish the normalized approximate arc lengths along the leading and trailing edges of the wing in the spanwise direction and to express the coordinates as tabular functions of the arc lengths. A grid spacing-control function[11] relating the ζ coordinate to the normalized approximate arc length is determined. Given the spacing distributions and a discrete set of ζ coordinates, physcial coordinates $F_1(1,1,K)$, $K = 1,2,\ldots N$, $F_1(L,1,K)$, $K = 1,2,\ldots N$ are computed on the leading and trailing edges of the wing. for each ζ coordinate, a plane-patch intersection of the wing is computed with the third point for the plane specified by the x and z coordinates of the leading-edge point and a distinctly different y coordinat.e The procedures used in the driver program for the fuselage are applied to the wing. The grid-spacing distribution function for the wing-fuselage intersection curve in the ξ coordinate direction is used to compute an intermediate wing-surface grid $E(I,K)$, $I = 1,2,\ldots L$, $K = 1,2,\ldots N$. In order to confrom the wing-surface grid to the shape of the winge-fuselage intersection curve, the final wing-surface grid is computed by

$$F_1(I,1,K) = E(I,K) + [1 - \zeta(K)]\,[F_1(I,1,1) - E(I,1)] \tag{3}$$

where

$$\zeta(K) = \frac{\exp\{-k[(K-1)/(N-1)]\} - 1}{e^{-k} - 1}$$

$I = 1,2,\ldots L$, $K = 1,2,\ldots N$, and k is a positive constant. Figure 8 shows a wing surface grid with grid concentration toward the wing-fuselage intersection curve.

Unconstrained Surface Grids

Six physical grid surfaces correspond to the six sides of the computational cube. Two of the surfaces are constrained to lie on the wing and fuselage surfaces. The remaining four grid surfaces are not prespecified and are referred to as the side, front, back, and cap grid surfaces.

We have tried many approaches for the next step, which is the computation of the cap grid. What seems to work best is to establish the cap surface and grid using the same procedure applied to the fuselage. The cap surface is defined with cross sections exactly as the fuselage is defined, and the cap grid is computed exactly as the fuselage grid is, with the same software except that the grid curve at $\eta = 0$ is now along the wing tip. The top-front cap grid is denoted by $F_1(I,J,N)$, $I = 1,2,\ldots L$, $J = 1,2,\ldots M$ and is shown in Fig. 8.

The side grid surface is defined in the x-y plane down the center of the fuselage; this grid is planar. The bottom boundary grid curve is specified from the fuselage grid, and the top boundary grid curve is specified by the cap grid. A two-dimensional version of the two-boundary technique[4] is applied directly with minor changes to pre-existing software. The only requirement is choosing the clustering control for the cubic-connecting function. The top-front side boundary grid surface is defined by $F_1(I,M,K)$, $I=1,2,\ldots L$, $K=1,2,\ldots N$.

The remaining exterior boundary grid surfaces are the front and the back. The boundary grid curves on all four sides of both surfaces are specified by the fuselage, wing, cap, and side grids. At this point, the interior grid surfaces could be computed using linear blending with transfinite interpolation. Instead, we have chosen to use the cubic blending found in the two-boundary technique for an intermediate grid between the fuselage and cap grids and exponential blending with the side and wing grids. A three-dimensional version of the two-boundary technique is incorporated into the concept of transfinite interpolation to compute the front and back grid surfaces. The two-boundary technique is used to compute intermediate grid surfaces $[E(1,J,K)$ and $E(L,J,K)$, $J=1,2,\ldots M$, $K=1,2,\ldots N]$, connecting the fuselage-boundary grid curves to the outer-boundary grid curves. The only additional information that must be specified is a control function that defines the grid clustering along the cubic-connecting function. Transfinite interpolation is used in a manner similar to the wing-grid definition. In this instance, it is used to conform the front and back grid surfaces to the side-boundary grid curve and the wing-boundary grid curve. The formulation is

$$F_1(I,J,K)=E(I,J,K)+\alpha(J)\left[F_1(I,1,K)-E(I,1,K)\right]$$
$$+\beta(J)\left[F_1(I,M,K)\;E(I,M,K)\right] \tag{4}$$

where

$$\alpha(J)=1-\frac{\exp\{-k[(J-1)/(M-1)]\}-1}{e^{-k}-1}$$

$$\beta(J)=\frac{\exp\{k[(J-1)/(M-1)]\}-1}{e^{k}-1}$$

and $I=1$ and L. The parameter k controls the effect of the side boundaries ($J=1$ and M) on the interior grid ($J=2,3,\ldots M-1$). The front and back grids are shown in Fig. 8.

Interior Grid Computation

Once the exterior grid surfaces corresponding to the six sides of the computational cube have been found, an interior grid is described by transfinite interpolation using the following linear-blending functions.

$$D(I,J,K)=[1-(K-1)/(N-1)]\,[F_1(I,J,1)]$$
$$+[(K-1)/(N-1)]\,[F_1(I,J,N)] \tag{5}$$

$$E(I,J,K)=D(I,J,K)+[1-(J-1)/(M-1)]\,[F_1(I,1,K)$$
$$-D(I,1,K)]+[(J-1)/(M-1)]\,[F_1(I,M,K)-D(I,M,K)] \tag{6}$$

$$F_1(I,J,K)=E(I,J,K)+[1-(I-1)/(L-1)]\,[F_1(1,J,K)$$
$$-E(1,J,K)]+[(I-1)/(L-1)]\,[F_1(1,J,K)-E(L,J,K)] \tag{7}$$

where $I=1,2,\ldots L$, $J=1,2,\ldots M$, and $K=1,2,\ldots N$. Selected interior-grid surfaces for the top-front and top-back grids are shown in Fig. 9. Figure 10 shows the bottom-front and bottom-back grids.

Other Configurations

Many topological complexities that can occur on real wing-fuselage configurations have not been considered in this paper. For instance, placing the wing very high or very low relative to the fuselage can create too much grid concentration on the part of the fuselage near the wing-fuselage intersection. Also, if the wing has a sharp leading edge, the C-type grid may not be satisfactory. On the other hand, the component parts of the procedure, such as surface representation, patch-plane intersections, and grid spacing control, are applicable to a wide variety of grid-generation problems.

Conclusion

A boundary-fitted wing-fuselage grid with several grid clusterings has been produced using Coons' patch surface definition, plane-patch intersections, and algebraic interpolation. Several conclusions can be drawn from this demonstration:

1) It is feasible to specify an aircraft surface independent of the grid-genration process, and highly developed techniques and software developed for linear aerodynamics and model making can be used.

2) A building-block concept in which several adjoining physical grids map to a computational cube simplifies the three-dimensional grid-generation process.

3) The intersection of a simple surface (plane) with the configuration surface established curves along which grid points are distributed.

4) Once the grids on the configuration surface are defined, algebraic grid-generation techniques are readily applied to obtain the remaining boundary and interior grids.

Acknowledgment

The authors wish to thank Ms. Joan Pitts for her technical support and editorial comments on this paper.

References

[1]Coons, S. A., "Surfaces for Computer-Aided Design of Space Forms," MAC-TR-41, Massachusetts Institute of Technology, Cambridge, MA, June 1967.

[2]Smith, R. E. and Weigel, B. L., "Analytical and Approximate Boundary-Fitted Curvilinear Coordinate Systems for Fluid Flow Simulation," AIAA Paper 80-0192, Jan. 1980.

[3]Smith, R. E., "Two-Boundary Grid Generation for the Solution of the Three-Dimensional Navier-Stokes Equations," NASA TM-83123, May 1981.

[4]Smith, R. E. and Wiese, M. R., "Interactive Algebraic Grid-Generation Technique," NASA TX-3206, March 1986.

[5]Gordon, W. J. and Hall, C., "Construction of Curvilinear Coordinate Systems and Applications of Mesh Generation," *International Journal of Numerical Methods in Engineering*, Vol. 7, July 1973, pp. 461-477.

[6]Eriksson, L.-E., "Practical Three-Dimensional Mesh Generation Using Transfinite Interpolation," *Journal of Fluid Mechanics*, Vol. 148, 1984, pp. 45-71.

[7]Smith, R. E., "Algebraic Grid Generation," *Numerical Grid Generation*, Elsevier, New York, 1982, pp. 137-168.

[8]Craidon, C. B., "Description of a Digital Computer Program for Airplane Configuration Plots," NASA TM X-2047, 1970.

[9]Craidon, C. B., "A Computer Program for Fitting Smooth Surfaces to an Aircraft Configuration and Other Three-Dimensional Geometries," NASA TM X-3206, June 1975.

[10]Smith, R. E., Kudlinski, R. A., and Pitts, J. I., "Surface Grid Generation for Wing-Fuselage Bodies, NASA CP 2272, April 1983.

[11]Smith, R. E., Kudlinski, R. A., and Everton, E. L., "A Grid Spacing Control Technique for Algebraic Grid Generation Methods," AIAA Paper 82-0226, Jan. 1982.