JOURNAL OF COMPUTATIONAL PHYSICS 113, 112-121 (1994)

Surface Grid Generation in a Parameter Space

JAMSHID SAMAREH-ABOLHASSANI AND JOHN E. STEWART

Computer Sciences Corporation, Hampton, Virginia 23666

Received August 31, 1992

A robust and efficient technique is discussed for surface-grid generation on a general curvilinear surface. This technique is based on a nonuniform parameter space and allows for the generation of surface grids on highly skewed and nonuniform spaced background surface-grids. This method has been successfully integrated into the GRIDGEN software system.

1. INTRODUCTION

Computational fluid dynamics (CFD) has progressed rapidly during the last two decades. It is now an integral part of the aircraft design process. This technology allows a designer to develop aerodynamic configurations in less time and cost compared to wind tunnel tests. The first step in obtaining a CFD solution is the creation of a structured or unstructured computational volume-grid. This step typically requires a considerable amount of time and effort on the part of a designer. The process of transforming the aerodynamic configuration from a computer-aided design (CAD) model to a CFD surface model is referred to as the surface-grid generation process. This process is often a formidable one for complex geometries such as realistic aircraft configurations. The surface-grid generation process is the first step in generating either a structured or unstructured volume-grid.

CAD systems typically represent the surfaces of aerodynamic vehicles with a set of structured points or patches. The initial surfaces constitute the background surface-grid on which the CFD surface-grid is generated. The final CFD surface-grid may or may not have the same number of points, distribution, and/or topology as the background surface-grid.

The process of surface-grid generation in the parameter space may be divided into three steps—forward mapping, grid generation, and backward mapping. The forward mapping is the mapping of the background surface-grid from a three-dimensional physical space to a two-dimensional parameter space. This step is referred to as surface parameterization in the computer-aided geometric design field. Curve and surface parameterization have been utilized in this field to a much greater extent than it has in the field of grid generation [1]. Once the forward mapping is complete, the grid is generated in the parameter space and then mapped back into the physical space (backward mapping).

A rule of thumb in computer-aided geometric design is to incorporate characteristics of the surface geometry into the parameterization as much as possible. It is very important to understand that the type of parameterization will not change the shape of the surface, but it will change the distribution of the points on the surface. A poor parameterization may cause the CFD surface-grid to be highly skewed. In grid generation, the parameter space is traditionally constructed through the use of the surface-grid indices. This method is referred to as a uniform parameter space (UPS) mapping. It is the most common method and produces satisfactory results as long as the background surface-grid is relatively smooth and orthogonal. As discussed in this paper, these shortcomings can be alleviated by transferring the background surface-grid from the physical space to a nonuniform parameter space (NPS) such as an arc-length parameter space (APS). For a fully nonuniform parameter space, the backward mapping is fairly complicated but nevertheless possible.

The following sections describe the forward mapping, algebraic grid generation, backward mapping, patch definition, and patch creation techniques. Results and conclusions concerning these techniques are presented in the last section.

2. FORWARD MAPPING

Transferring the background surface-grid from the physical space ($\bar{R} = \{X, Y, Z\}^T$) to a parameter space ($\bar{W} = \{U, V\}^T$) is referred to as a forward mapping. The background surface-grid is a general curvilinear surface (Fig. 1a) which can be represented by either a bilinear, bicubic, nonuniform rational B-spline (NURBS), or other representation. This process is very similar to the parameterization

0021-9991/94 $6.00

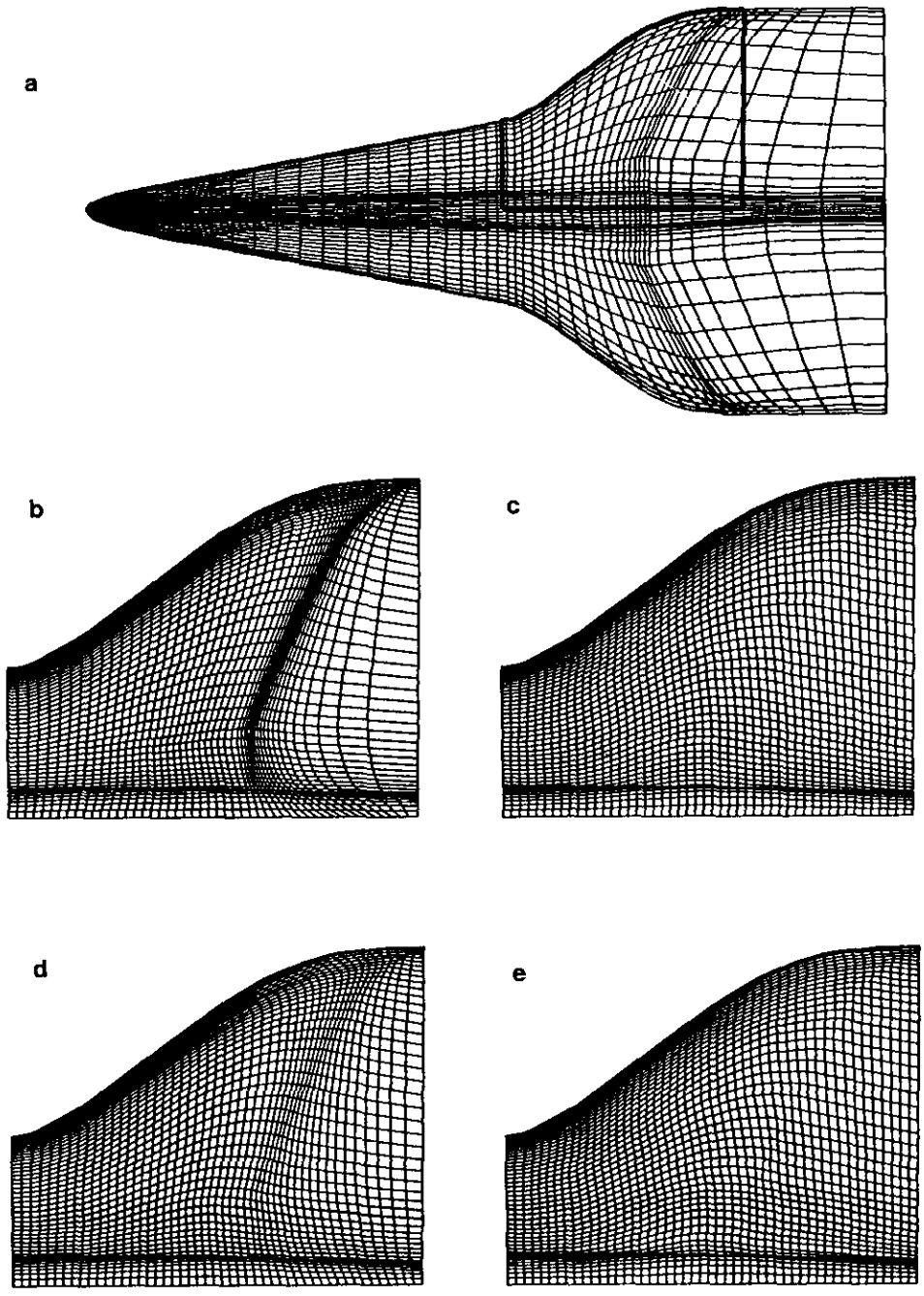

FIG. 1. High-speed civil transport (HSCT): (a) HSCT background surface-grid; (b) CFD grid (UPS); (c) CFD grid (APS); (d) CFD grid (CPS); (e) CFD grid (FPS).

process used in computer-aided geometric design. A suface is represented by a set of patches, each of which has a local coordinate $(0 \leqslant (s_{i,j}, t_{i,j}) \leqslant 1)$, and a global coordinate $(0 \leqslant (U_{i,j}, V_{i,j}) \leqslant 1)$ which covers a set of patches. The values assigned to $U_{i,j}$ and $V_{i,j}$ must be monotone with respect to i and j, respectively. Aside from these conditions, the selection of the parameter space is arbitrary. It is very important to understand that the type of parameterization will not change the shape of the surface but will change the distribution of the points on the surface. There are four techniques discussed below to parameterize the background surface-grid—uniform, arclength, centripetal [2], and Foley [3]. These parameterizations are discussed within the grid generation context.

The uniform parameter space (UPS) is very simple to create, and it is the primary type of parameter space used in

grid generation today. A UPS is constructed by using the background surface-grid indices (Fig. 1b)

$$U_{i,j} = i, \qquad V_{i,j} = j, \tag{1}$$

as the parameter space. The UPS contains no information about the physical characteristics of the background surface-grid itself.

An arc-length parameter space (APS) is constructed by computing the arc-length in each computational direction (Fig. 1c)

$$U_{i,j} = U_{i-1,j} + e_{i,j}, \tag{2}$$

$$V_{i,j} = V_{i,j-1} + f_{i,j}, \tag{3}$$

where $e_{i,j}$ and $f_{i,j}$ are defined as

$$e_{i,j} = \|\bar{R}_{i,j} - \bar{R}_{i-1,j}\|,$$

$$f_{i,j} = \|\bar{R}_{i,j} - \bar{R}_{i,j-1}\|.$$

This parameter space incorporates some information about the relative grid spacing. Lee [2] has developed a variation on the APS which is referred to as the centripetal parameter space (CPS). This parameter space is constructed by computing the square root of the arclength in each computational direction (Fig. 1d)

$$U_{i,j} = U_{i-1,j} + \sqrt{e_{i,j}}, \tag{4}$$

$$V_{i,j} = V_{i,j-1} + \sqrt{f_{i,j}}. \tag{5}$$

All of the above parameterizations contain no information regarding the skewness of the background surface-grid. The Foley parameter space (FPS [3]), however, is based on the arclength *and* the angle formed by neighboring segments in each computational direction (Fig. 1e),

$$U_{i,j} = U_{i-1,j} + \Delta U_{i,j}, \tag{6}$$

where $\Delta U_{i,j}$ is defined as

$$\Delta U_{i,j} = \frac{3}{2} e_{i,j} \left[\frac{2}{3} + \frac{\hat{\Theta}_{i-1,j} e_{i-1,j}}{e_{i-1,j} + e_{i,j}} + \frac{\hat{\Theta}_{i,j} e_{i+1,j}}{e_{i,j} + e_{i+1,j}} \right]$$

and

$$\hat{\Theta}_{i,j} = \min\left(\pi - \Theta_{i,j}, \frac{\pi}{2}\right).$$

The angle, $\Theta_{i,j}$, is the angle formed by the points $\bar{R}_{i-1,j}$, $\bar{R}_{i,j}$, $\bar{R}_{i+1,j}$. This equation is a variation of the Foley equation which permits symmetry about angular deviations in the surface-grid. A similar expression can be written for $V_{i,j}$.

The $U_{i,j}$ and $V_{i,j}$ are normalized such that U and V vary from zero to one along i- and j-coordinate directions, respectively. In general, the NPS is variant under affine transformation, but this will not effect the grid generation process.

3. GRID GENERATION

The grid generation process is the same for all parameter spaces. As shown in Fig. 2a, edges of the CFD surface grid are first determined in the parameter space $[\bar{w}_{l,m} = \{u_{l,m}, v_{l,m}\}^T]$. Once the edges are known, the grid may be calculated in the parameter space by any grid generation method.

The method described here is based on the transfinite interpolation,

$$\begin{aligned}\bar{w}_{l,m} = {} & a_{l,m}\bar{w}_{l,m1} + b_{l,m}\bar{w}_{l,m2} + c_{l,m}\bar{w}_{l1,m} + d_{l,m}\bar{w}_{l2,m} \\ & - a_{l,m}c_{l,m}\bar{w}_{l1,m1} - b_{l,m}c_{l,m}\bar{w}_{l1,m2} \\ & - a_{l,m}d_{l,m}\bar{w}_{l2,m1} - b_{l,m}d_{l,m}\bar{w}_{l2,m2}.\end{aligned} \tag{7}$$

The subscripts $(l1, l2)$ and $(m1, m2)$ are the minimum and maximum indices of the CFD grid in the u and v directions, respectively. The variables a, b, c, and d are the blending functions subjected to the following conditions,

$$a_{l,m1} = c_{l1,m} = b_{l,m2} = d_{l2,m} = 1,$$

$$a_{l,m2} = c_{l2,m} = b_{l,m1} = d_{l1,m} = 0.$$

The most robust blending functions have been proposed by Soni [4] which are defined as

$$a_{l,m} = 1 - \eta_{l,m}, \qquad b_{l,m} = \eta_{l,m},$$

$$c_{l,m} = 1 - \xi_{l,m}, \qquad d_{l,m} = \xi_{l,m},$$

where

$$\xi_{l,m} = \frac{g_{l,m1} + h_{l1,m}\{g_{l,m2} - g_{l,m1}\}}{P_{l,m}}, \tag{8}$$

$$\eta_{l,m} = \frac{h_{l1,m} + g_{l,m1}\{h_{l2,m} - h_{l1,m}\}}{P_{l,m}}, \tag{9}$$

$$P_{l,m} = 1 - \{g_{l,m2} - g_{l,m1}\}\{h_{l2,m} - h_{l1,m}\},$$

where $g_{l,m}$ and $h_{l,m}$ are the normalized arclength of the edges in the physical space. Once this interpolation is completed, the grid can be transferred from the parameter space to the physical space.

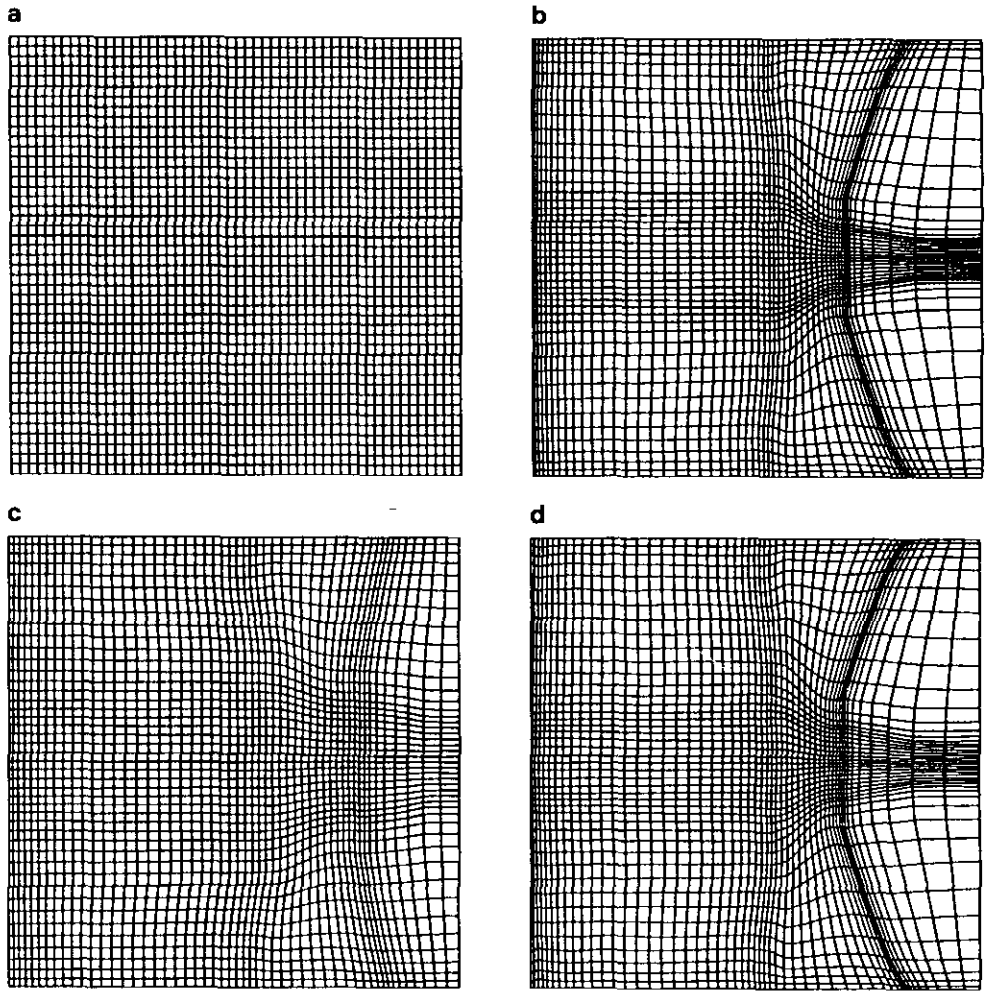

FIG. 2. Parameter spaces for Fig. 1a: (a) UPS; (b) APS; (c) CPS; (d) FPS.

4. BACKWARD MAPPING

Backward mapping is the transferring of grid points from the parameter space $[\bar{w}_{l,m} = \{u_{l,m}, v_{l,m}\}^T]$ to the physical space $[\bar{r}_{l,m} = \{x_{l,m}, y_{l,m}, z_{l,m}\}^T]$. Three steps are used for this backward mapping. The first step is to search for a patch in the background parameter space $(\bar{W}_{i,j})$ that surrounds the grid point $(\bar{w}_{l,m})$. Since the parameter space is two-dimensional, the search is performed in two dimensions. The second step is to compute the local coordinates of the patch using bilinear interpolation. The third step is to compute the coordinates of the grid points in the physical space. This is accomplished using bilinear interpolation if the background surface-grid is a collection of grid points, bicubic interpolation if the background surface-grid is a collection of patches, or other appropriate approximations depending on the background surface-grid type. Each of these steps are outlined for a UPS and a NPS.

The search is straightforward for a UPS. The grid point $(\bar{w}_{l,m})$ on the parametric surface is surrounded by the background surface-grid points (i, j), $(i+1, j)$, $(i+1, j+1)$, and $(i, j+1)$. The indices, i and j, are the integer portions of $u_{l,m}$ and $v_{l,m}$, respectively.

The search for a NPS is more involved. One way to proceed is to perform an exhaustive angular search (explained later) over the entire background surface-grid. This is very inefficient. A much more efficient method is outlined here. The procedure is based on an iterative search and hence requires an initial guess. The initial guess for the first point is found using a blind search. In this search, the entire background surface-grid is searched to identify the closest grid point to $(\bar{w}_{l,m})$. For subsequent points, the previously found grid point in the background surface-grid serves as an initial guess. Once the closest point has been found, four patches are formed which consist of nine points surrounding and including the closest point. Then, an angular search is performed to find the patch that contains the grid point $(\bar{w}_{l,m})$. The angular search computes the sum of the angles

formed by $(\bar{w}_{l,m})$ and the four corners of each patch in the parameter space. If the sum is equal to $2\pi \pm \varepsilon$ where ε is a small approximation error, then $(\bar{w}_{l,m})$ is within the patch. For highly skewed grids, $(\bar{w}_{l,m})$ may not be within the region enclosed by the four initial patches. If this is true, then the angular search is extended outward to the closest neighboring patches until the whole region is covered. With the current method, each patch is searched only once. If this search fails, $(\bar{w}_{l,m})$ is not on the surface. With this method, an average of two patches is searched per surface-grid point.

In the second step, the coordinates of the parameter space $(\bar{w}_{l,m})$ are converted to the coordinates of the local patch ($s_{l,m}$ and $t_{l,m}$). This is relatively simple for a UPS, where

$$s_{l,m} = u_{l,m} - U_{i,j}, \tag{10}$$

$$t_{l,m} = v_{l,m} - V_{i,j}. \tag{11}$$

For a NPS, the local coordinates of the patch are calculated using a set of quasi-linear equations. The values of $U_{i,j}$ and $V_{i,j}$ are known for each corner of the patch and the grid point $(u_{l,m}, v_{l,m})$ is known for the interior of the identified patch. With this information, two equations and two unknowns are formed, where

$$\begin{aligned} u_{l,m} &= (1-s_{l,m})(1-t_{l,m})\,U_{i,j} + s_{l,m}(1-t_{l,m})\,U_{i+1,j} \\ &\quad + (1-s_{l,m})\,t_{l,m}U_{i,j+1} + s_{l,m}t_{l,m}U_{i+1,j+1}, \\ v_{l,m} &= (1-s_{l,m})(1-t_{l,m})\,V_{i,j} + s_{l,m}(1-t_{l,m})\,V_{i+1,j} \\ &\quad + (1-s_{l,m})\,t_{l,m}V_{i,j+1} + s_{l,m}t_{l,m}V_{i+1,j+1}. \end{aligned} \tag{12}$$

The solution to these equations gives the local coordinates of the patch $(s_{l,m}, t_{l,m})$. This set of quasi-linear equations is solved using the Newton–Raphson method. A value of $\frac{1}{2}$ is used as in initial guesses for both s and t. It typically takes two to three iterations to obtain a converged solution. These values are then used to obtain the physical coordinates $(\bar{r}_{l,m})$ of the parametric grid point $(\bar{w}_{l,m})$ of the identified patch. This is accomplished using either a bilinear or bicubic interpolation techniques as described in the following section.

5. SURFACE PATCHES

A background surface-grid is commonly represented by a set of surface patches. A surface patch contains four boundary segments. Each patch is generated by mathematically blending the shapes of the individual segments into each other to form the patch. A surface is formed by collecting a group of patches that have some degree of continuity along their common boundaries.

For a linear patch, only the coordinates of the four corners are needed. This patch is mapped from parameter space $(\bar{w}_{l,m})$ to the physical space $(\bar{r}_{l,m})$ using a bilinear equation of the form

$$\begin{aligned} r_{l,m} &= (1-s_{l,m})(1-t_{l,m})\,\bar{R}_{i,j} + s_{l,m}(1-t_{l,m})\,\bar{R}_{i+1,j} \\ &\quad + (1-s_{l,m})\,t_{l,m}\bar{R}_{i,j+1} + s_{l,m}t_{l,m}\bar{R}_{i+1,j+1}. \end{aligned} \tag{13}$$

For a dense background surface-grid with small variations in curvature, a bilinear surface patch produces satisfactory results. For a coarse background surface-grid with large variations in curvature, bilinear surface patches produce grids that are not satisfactory. A bicubic surface patch can be used in this case to maintain the continuity in curvature. A bicubic surface patch is represented by

$$[\bar{R}] = [S][M][\bar{B}][M]^{\mathrm{T}}\,[S]^{\mathrm{T}},$$

$$[S] = [s_{l,m}^3\ s_{l,m}^2\ s_{l,m}\ 1],$$

$$[T] = [t_{l,m}^3\ t_{l,m}^2\ t_{l,m}\ 1]^{\mathrm{T}},$$

$$[M] = \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$

$$[\bar{B}] = \begin{bmatrix} \bar{R}_{i,j} & \bar{R}_{i,j+1} & \dfrac{\partial \bar{R}_{i,j}}{\partial j} & \dfrac{\partial \bar{R}_{i,j+1}}{\partial j} \\ \bar{R}_{i+1,j} & \bar{R}_{i+1,j+1} & \dfrac{\partial \bar{R}_{i+1,j}}{\partial j} & \dfrac{\partial \bar{R}_{i+1,j+1}}{\partial j} \\ \dfrac{\partial \bar{R}_{i,j}}{\partial i} & \dfrac{\partial \bar{R}_{i,j+1}}{\partial i} & \dfrac{\partial^2 \bar{R}_{i,j}}{\partial i\,\partial j} & \dfrac{\partial^2 \bar{R}_{i,j+1}}{\partial i\,\partial j} \\ \dfrac{\partial \bar{R}_{i+1,j}}{\partial i} & \dfrac{\partial \bar{R}_{i+1,j+1}}{\partial i} & \dfrac{\partial^2 \bar{R}_{i+1,j}}{\partial i\,\partial j} & \dfrac{\partial^2 \bar{R}_{i+1,j+1}}{\partial i\,\partial j} \end{bmatrix}, \tag{14}$$

where $[S][M]$ and $[M]^{\mathrm{T}}\,[S]^{\mathrm{T}}$ are the blending functions, and $[\bar{B}]$ is the boundary matrix. The $[\bar{B}]$ matrix contains the coordinates, the parameter rates, and the twist vectors of the four corners of the patch. This bicubic patch representation is referred to as a Coons' patch [5]. It is possible to create a patch with infinite slope in the physical space using this definition.

The $[\bar{B}]$ matrix is created by first approximating the derivatives at each corner. The simplest way to evaluate

these derivatives is to use a centered difference approximation [6]:

$$\frac{\partial \bar{R}_{i,j}}{\partial i} = \frac{\bar{R}_{i+1,j} - \bar{R}_{i-1,j}}{2}. \tag{15}$$

This approximation produces satisfactory results as long as the background surface-grid is dense and smooth. For a general background surface-grid, derivatives should be approximated by cubic splines [7]. The derivatives at each corner may be calculated by splining ($\bar{R}$) with respect to arclength in the i- and j-coordinate directions. The arclength in the i-coordinate direction is defined as

$$E_{i,j} = E_{i-1,j} + e_{i,j}. \tag{16}$$

The derivative of ($\bar{R}$) is then multiplied by the average spacing at each grid point [7]

$$\frac{\partial \bar{R}}{\partial i} = \frac{\partial \bar{R}}{\partial E_{i,j}} \frac{e_{i,j} + e_{i+1,j}}{2}. \tag{17}$$

A slightly different approach is proposed in this paper which works well for a background surface-grid with discontinuous spacing. The arclength ($E_{i,j}$) is splined with respect to the i- and j-coordinate directions. Then, the derivatives of ($\bar{R}$) are multiplied by the derivative of ($E_{i,j}$) at each grid point:

$$\frac{\partial \bar{R}}{\partial i} = \frac{\partial \bar{R}}{\partial E_{i,j}} \frac{\partial E_{i,j}}{\partial i}. \tag{18}$$

The standard cubic spline and Akima [8] methods are used for the splining algorithms. Each method has its merits. The standard cubic spline algorithm uses a C^2 spline and produce satisfactory results as long as there are no slope discontinuities in the surface. For a surface with a discontinuous slope, the standard cubic spline may over- or under-shoot the actual surface. In these cases, one may use the Akima technique which is C^1-continuous near the region of discontinuous slope.

6. RESULTS

Three different test cases are used to demonstrate the influence of surface curvature, nonuniform spacing, and grid orthogonality on grid generated in a parameter space. The first test case is the surface of a generic high-speed civil transport (HSCT) vehicle which is being analyzed at NASA Langley Research Center. The second and third test cases are planar surfaces used to demonstrate the influence of background surface-grid discontinuity and orthogonality on CFD surface-grids.

There are four steps in generating a CFD surface-grid in a parameter space. The first step is to construct the four edges of the grid in the physical space (Fig. 1a). In the second step, the background surface-grid is mapped to a parameter space using one of the techniques described in Section 4. In the third step, the interior grid is generated using transfinite interpolation with Soni's blending functions as described in Section 5. Finally, the grid is transformed from the parameter space to the physical space. The edges of the second and third test cases are uniformly distributed. In the ideal parametric transformation, the interior surface-grid is controlled by only the edge distribution and edge shape. Therefore, a uniformly distributed CFD surface-grid should result for these cases.

The first test case uses the HSCT surface as the background surface-grid. This surface is relatively simple and is representative of a typical CFD surface-grid. Figures 1b–e show the CFD surface-grids generated in UPS, APS, CPS, and FPS, respectively. The CFD surface-grid generated in a UPS (Fig. 1b) is neither orthogonal or uniform and has concentrations in the same regions as the background surface-grid. The grid generated in an APS (Fig. 1c) is uniform an orthogonal. The CFD grids generated in a CPS and FPS (Figs. 1d–e) are relatively orthogonal, uniform, and have concentrations in the same regions as the background surface-grid.

Of the four grids generated for the HSCT surfaces, the APS is the best surface-grid in terms of uniform spacing and degree of skewness. The UPS, on the other hand, is the worst. These diagrams demonstrate that the background surface curvature and skewness have little or no effect on the surface parameterization. The background surface-grid spacing, however, does affect the surface parameterization. In order to understand why this is the case, one should examine the parameter spaces for the HSCT surface (Figs. 2a–d). These figures show that the quality of the CFD grid is directly dependent on how much information has been carried from the background surface-grid to the parameter space. The APS brings the most information to the parameter space because it concentrates grid points in the parameter space in the same regions that they are concentrated in the physical space. This parameter space is most representative of the background grid.

The second background surface-grid is a planar surface with spacing discontinuities (Fig. 3a). Figures 3b–e show grids generated in the UPS, APS, CPS, and FPS, respectively. Since the background surface grid is completely orthogonal, the APS and FPS produce the same result. This is because the FPS reduces to the APS when the back-

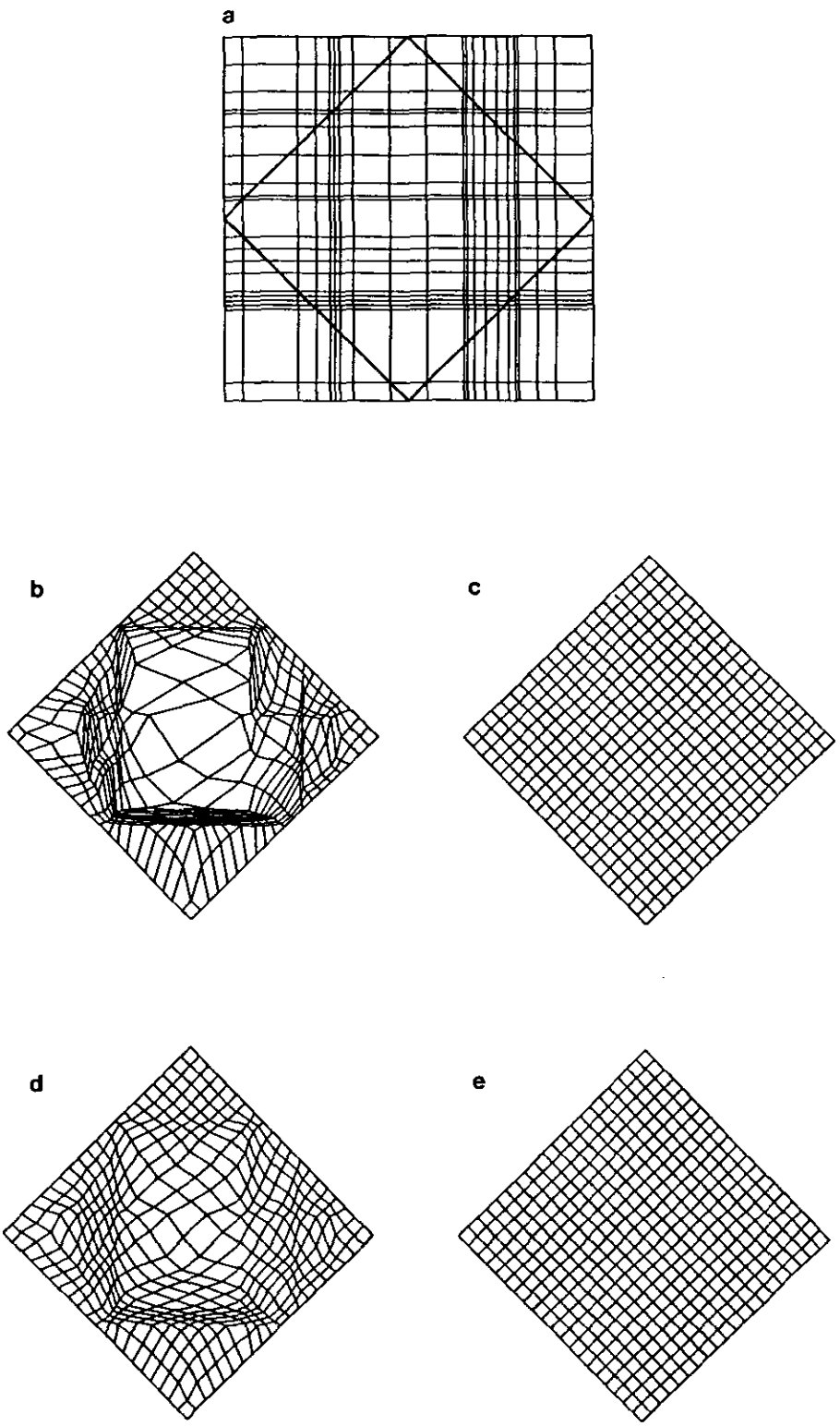

FIG. 3. Background and CFD grid for Test Case Number 2: (a) background grid for Test Case Number 2; (b) UPS; (c) APS; (d) CPS; (e) FPS.

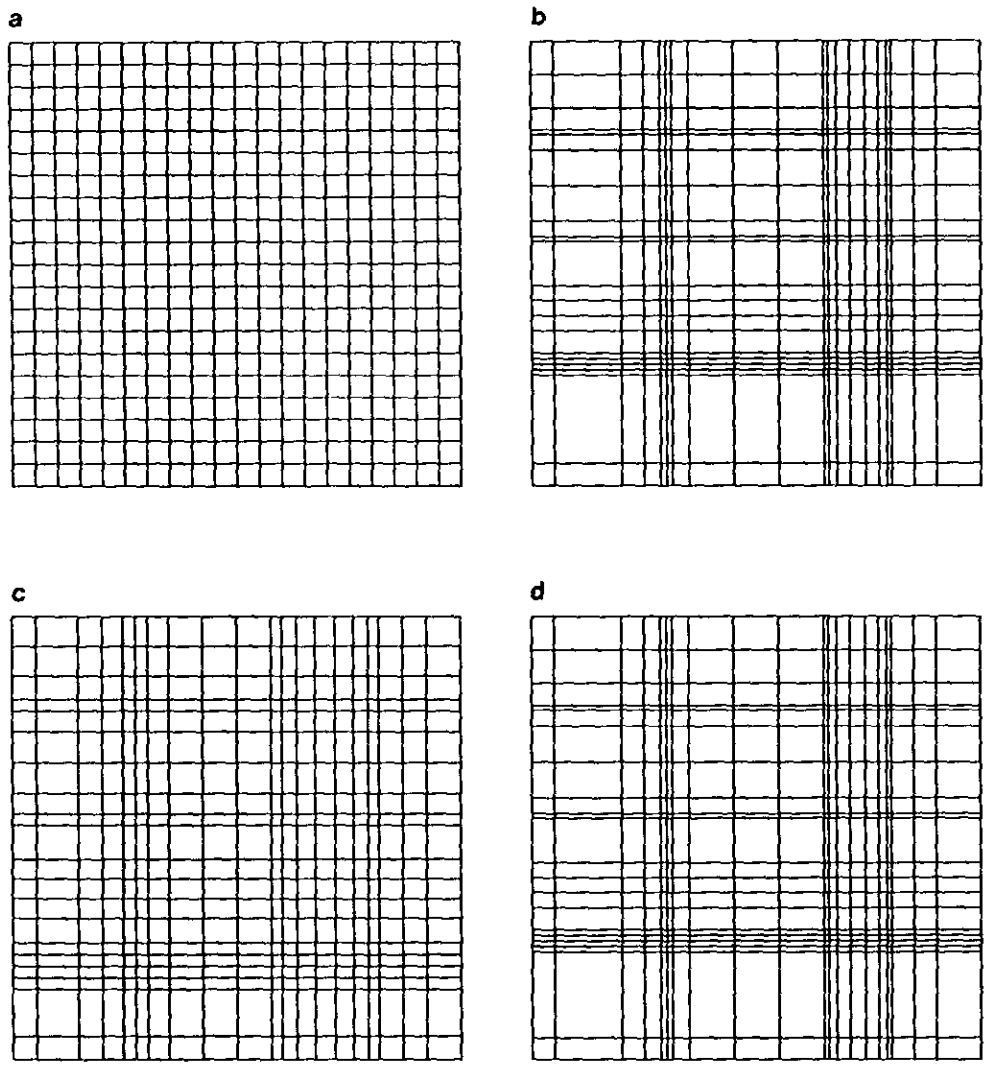

FIG. 4. *Parameter spaces for Fig. 3a: (a) UPS; (b) APS; (c) CPS; (d) FPS.*

ground grid is orthogonal. Once again, the grid generated in an APS is superior to the grid generated in a UPS or CPS. The reason behind the large discrepancy in the CFD grids can be explained when one examines the parameter spaces. Figures 4a–d show the parameter spaces for the UPS, APS, CPS, and FPS, respectively. Again, the APS is the most representative of the background grids because it brings the most information from the background surface-grid to the parameter space. In this case, the APS is identical to the background surface-grid.

The third background surface-grid is also a planar surface but it has discontinuities in angle rather than spacing (Fig. 5a). Figures 5b–e show grids generated in a UPS, APS, CPS, and FPS, respectively. Again, the grid generated in an APS is superior to the grids generated in a UPS, CPS, and FPS. Figures 6a–d show the parameter spaces for a UPS, APS, CPS, and FPS. As in the first two cases, the APS is most representative of the background surface-grid because it brings the skewness of the background grid into the parameter space.

7. CONCLUSIONS

If the background surface-grid is smooth and orthogonal, the CFD surface may be generated in any of the parameter spaces discussed in this paper. If, however, the background surface-grid is not smooth or it is skewed, then the CFD surface-grid should be generated in an APS. Among four parameterization discussed in this paper, the APS is the most representative of the background surface-grid because it has concentrations in grid spacing and skewness in the same regions as the background surface-grid. For CFD applications, the APS is the best among the parameter spaces examined here.

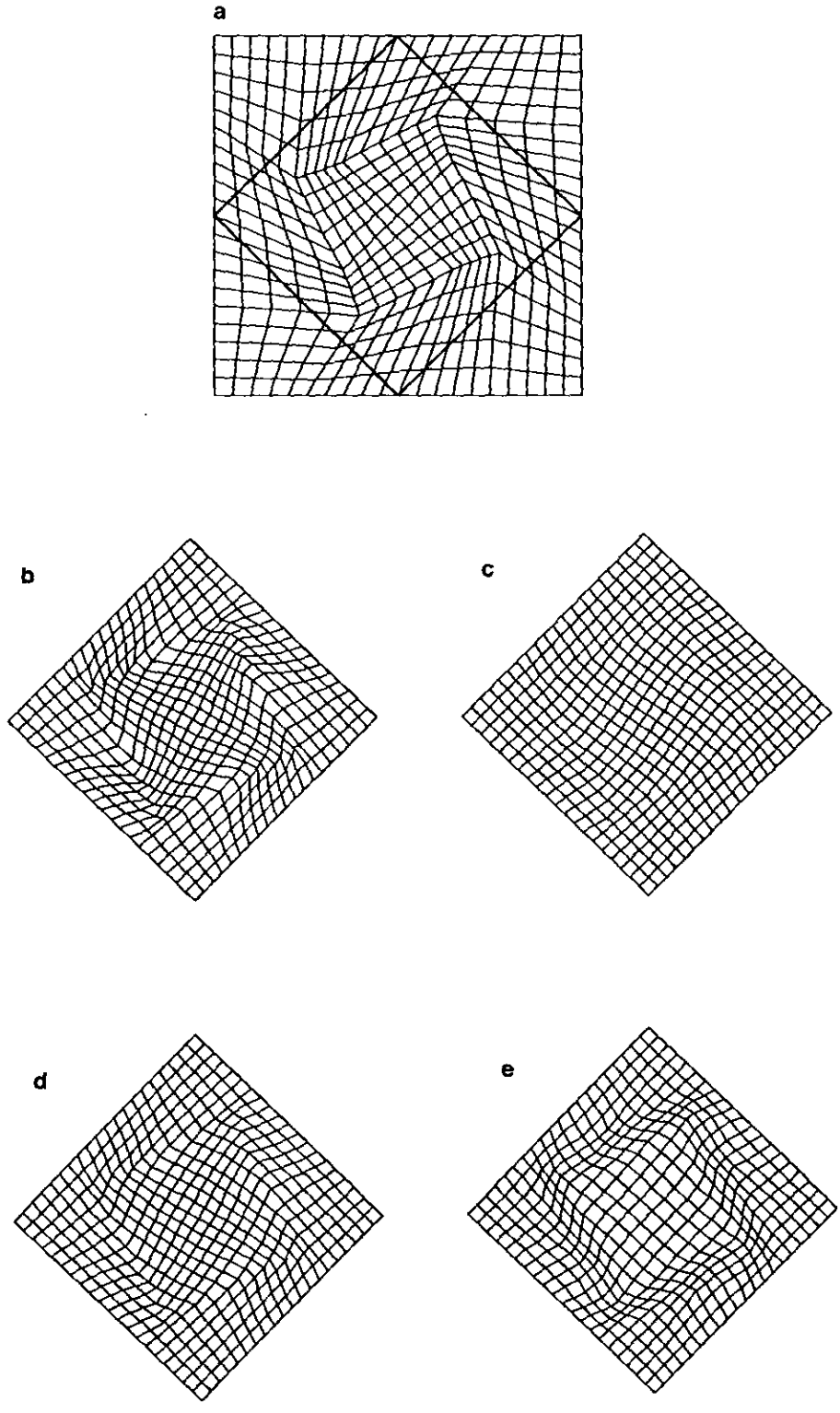

FIG. 5. Background and CFD grids for Test Case 3: (a) UPS; (b) APS; (c) CPS; (d) FPS.

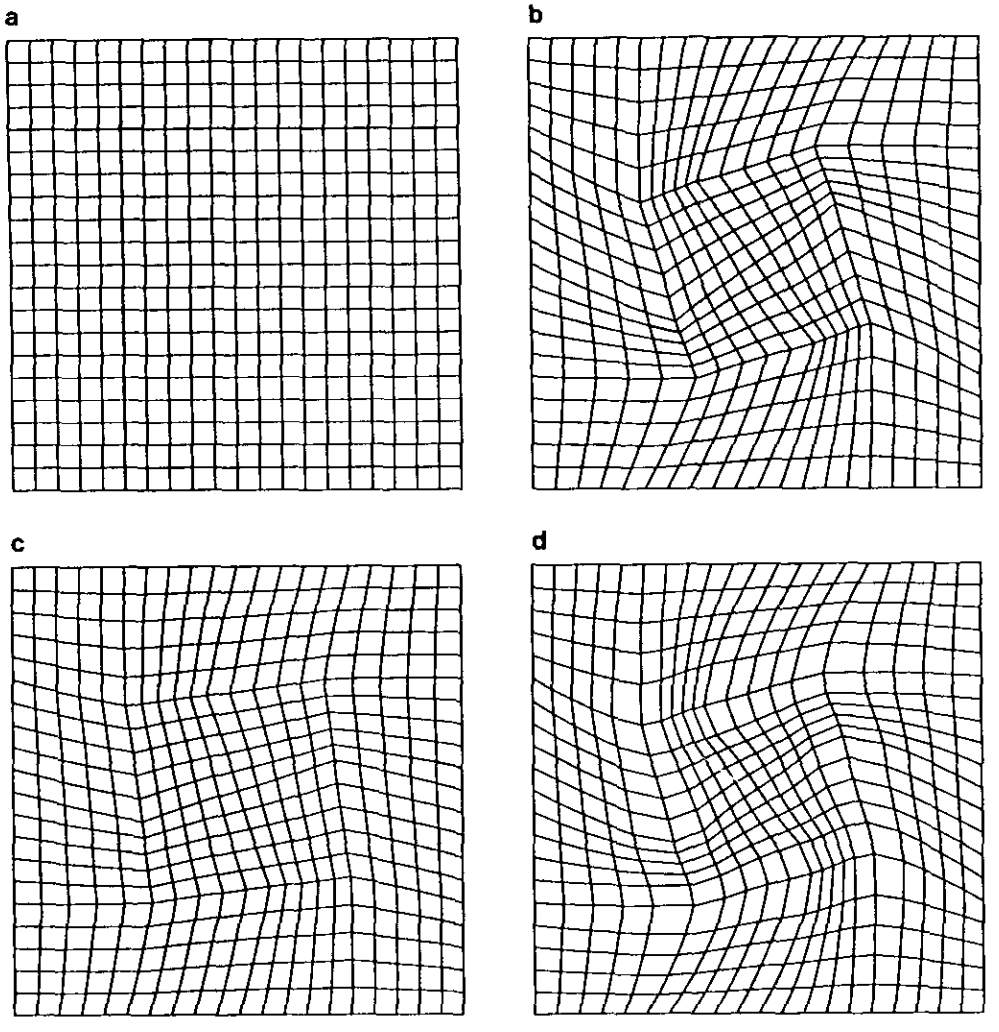

FIG. 6. Parameter spaces for Fig. 5a; (a) UPS; (b) APS; (c) CPS; (d) FPS.

ACKNOWLEDGMENT

The authors thank Dr. Robert E. Smith, Jr. of the Computer Applications Branch of NASA Langley Research Center for providing support and guidance for this research.

REFERENCES

1. G. E. Farin, *Curves and Surfaces for Computer Aided Geometric Design, A Practical Guide*, 2nd ed. (Academic Press, New York, 1989).
2. E. Lee, *Comput. Aided Des.* **21**, No. 6 (1989).
3. G. Nielson and T. Foley, "A Survey of Applications of Affine Invariant Norm," in *Mathematical Aspects in Computer Aided Geometric Design*, edited by T. Lyche and L. Schmaker (Academic Press, New York, 1989), p. 445.
4. B. K. Soni, AIAA-85-1526, 1985.
5. S. Coons, Technical Report, Project MAC-TR 41, MIT, 1967.
6. J. P. Steinbrenner, J. R. Chawner, and C. L. Fouts, Contract Report F33615-87-C-3003, General Dynamics, July 1990.
7. C. B. Craidon, NASA TM X-3206, June 1975.
8. H. Akima, *J. Assoc. Comput. Mach.* **17**(4), 589 (1970).