

1995/22334

GRID GENERATION ON TRIMMED BEZIER AND NURBS QUILTED SURFACES

Chung-Jin Woan, Willard C. Clever, and Clement K. Tam
 North American Aircraft Division/Rockwell International
 Seal Beach, California 90740-7644

405035
 18P

SUMMARY

This paper presents some recently added capabilities to RAGGS, Rockwell Automated Grid Generation System. Included are the trimmed surface handling and display capability and structured and unstructured grid generation on trimmed Bezier and NURBS quilted surfaces. Samples are given to demonstrate the new capabilities.

INTRODUCTION

As described in Ref. 1, RAGGS (Rockwell Automated Grid Generation System) provides an integrated CFD (Computational Fluid Dynamics) environment where a user can 1) import a CAD (Computer Aided Design) generated configuration or a geometry from other sources, 2) build a CFD grid directly on the imported geometry, 3) set up flow solver input, 4) postprocess the flow solution, and 5) display the flow solution data and geometry simultaneously. The program structure of RAGGS is depicted in Fig. 1. It consists of a series of modules to handle various tasks in the CFD process: starting with a given geometry, usually input as an IGES (Initial Graphics Exchange Specification) file, through construction of CFD geometry database, domain blocking, surface grid generation, volume grid generation, flow simulation, to postprocessing of the flow solution to obtain data useful for engineering design and analysis. A file containing the geometry data read by RAGGS is referred to as a RASCAL (Rockwell Automated System for Computer Aided Lofting) file. As shown in Fig. 1, each module is connected to the common database shown in the central hub. Through the GUI (Graphical User Interface) module and/or grid file module, RAGGS enables user to quickly communicate with some state-of-the-art flow solvers, flow analysis software, and grid generators as listed in the figure. The RAGGS surface grid generation is operated directly on the quilts of rational Bezier patches and/or NURBS (Non Uniform Rational B-Spline), each of arbitrary order. This minimizes the interaction of CAD/CFD groups and eliminates the need of re-constitution of the imported geometry, which often requires hundreds of hours of manipulation to achieve a data format suitable for the grid generation software package.

This paper presents some recently added capabilities to RAGGS. These include trimmed surface/curve handling and display, unstructured surface grid generation, and structured/unstructured grid generation on trimmed Bezier and NURBS quilted surfaces. The materials presented in this paper are intended to complement that given in Ref. 1.

GEOMETRY DATABASE

The coordinates of a surface in space can be described in terms of two parametric variables. A patch is a mapping from a space defined by these two parametric variables to the three Cartesian coordinates in 3-D space. The untrimmed domain of each patch is the rectangular region in parameter space, $s_0 \leq s \leq s_1$ and $t_0 \leq t \leq t_1$, where usually $s_0 = t_0 = 0$ and $s_1 = t_1 = 1$. Additionally, by defining active and inactive regions, trim curves may be used to further restrict the domain of a patch. These trim curves are nonintersecting closed loops of Bezier curves or NURBS in the parametric space of a patch.

For a general configuration, aircraft or otherwise, there is no uniform parameterization which can describe the entire surface. However, the surface can be described by a set of individually parameterized patches,

possibly with trim curves. Any subset of these patches, having surface continuity, can be said to constitute a quilt. Any configuration or any number of configurations can be completely described by a number of such quilts. RAGGS uses such quilts of rational Bezier patches and/or NURBS, each of arbitrary order, to create grids which lie exactly on the parameterized surface. These patches may have 3 or 4 sides and need not be distributed in any type of uniform manner.

Rational Bezier Patch and NURBS

Bezier and NURBS curves in 3-D space are composed of a point which varies parametrically with a single parameter. Surfaces are composed of a curve which itself varies parametrically in 3-D space according to another parameter. Consider a such NURBS curve, of degree n , parameterized in t . The curve may be written in terms of the B-Spline basis functions as follows:

$$\tilde{G}(t) = \frac{\sum_{j=0}^{K_n} v_j \tilde{g}_j b_j^n(t)}{\sum_{j=0}^{K_n} v_j b_j^n(t)} \quad \left\{ \begin{array}{l} v_j \quad - \quad \text{the weight coefficients} \\ \tilde{g}_j \quad - \quad \text{the control points} \\ b_j^n(t) \quad - \quad \text{the B-Spline basis functions} \end{array} \right. \quad (1)$$

A knot vector is defined by the nondecreasing sequence $t_{-n}, \dots, t_0, t_1, \dots, t_{K_n}, t_{K_n+1}$. The B-Spline basis functions are defined by the following recursion,

$$b_i^{n+1}(t) = \frac{t - t_{i-n-1}}{t_i - t_{i-n-1}} b_{i-1}^n(t) + \frac{t_{i+1} - t}{t_{i+1} - t_i} b_i^n(t) \quad \text{where,} \quad b_i^0(t) = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The basis functions have the property that

$$\sum_{j=0}^{K_n} b_j^n(t) \equiv 1 \quad (3)$$

For a rational Bezier curve, $K_n = n$, is equal to the degree of the curve, the parameter domain becomes $[0, 1]$, the knot vector becomes $t_{-n}, t_{-n+1}, \dots, t_0 = 0$, and $t_1, \dots, t_n, t_{n+1} = 1$, and the basis functions become the Bernstein functions,

$$b_k^n(t) = \frac{n!}{k!(n-k)!} (1-t)^{n-k} t^k \quad (4)$$

If we let the weights, v_j , and the control points, \tilde{g}_j , be interpolated functions of another parameter s ,

$$v_j(s) = \sum_{i=0}^{K_m} w_{ji} b_i^m(s) \quad \tilde{g}_j(s) = \frac{\sum_{i=0}^{K_m} w_{ji} \tilde{p}_{ji} b_i^m(s)}{\sum_{i=0}^{K_m} w_{ji} b_i^m(s)} \quad (5)$$

Then Eq. (1) together with Eq. (5) constitutes a surface patch parameterized in s and t as follows:

$$\tilde{P}(s, t) = \frac{\sum_{i=0}^{K_n} v_j(s) \tilde{g}_i(s) b_i^n(t)}{\sum_{i=0}^{K_n} v_j(s) b_i^n(t)} = \frac{\sum_{j=0}^{K_n} \left[\frac{\sum_{i=0}^{K_m} w_{ji} \tilde{p}_{ji} b_i^m(s)}{\sum_{i=0}^{K_m} w_{ji} b_i^m(s)} \right] b_j^n(t)}{\sum_{j=0}^{K_n} \left[\sum_{i=0}^{K_m} w_{ji} b_i^m(s) \right] b_j^n(t)} = \frac{\sum_{j=0}^{K_n} \sum_{i=0}^{K_m} w_{ji} \tilde{p}_{ji} b_i^m(s) b_j^n(t)}{\sum_{j=0}^{K_n} \sum_{i=0}^{K_m} w_{ji} b_i^m(s) b_j^n(t)} \quad (6)$$

with \vec{p}_{ij} being the control points and w_{ij} the weights.

For fixed t , Eq. (6) may be written as

$$\vec{H}(s) = \vec{P}(s, t) = \frac{\sum_{i=0}^{K_m} \left[\sum_{j=0}^{K_n} w_{ji} b_j^n(t) \frac{\sum_{j=0}^{K_n} w_{ji} \vec{p}_{ji} b_j^n(t)}{\sum_{j=0}^{K_n} w_{ji} b_j^n(t)} \right] b_i^m(s)}{\sum_{i=0}^{K_m} \left[\sum_{j=0}^{K_n} w_{ji} b_j^n(t) \right] b_i^m(s)} = \frac{\sum_{i=0}^{K_m} u_i(t) \vec{h}_i(t) b_i^m(s)}{\sum_{i=0}^{K_m} u_i(t) b_i^m(s)} \quad (7)$$

with

$$u_i(t) = \sum_{j=0}^{K_n} w_{ji} b_j^n(t) \quad \vec{h}_i(t) = \frac{\sum_{j=0}^{K_n} w_{ji} \vec{p}_{ji} b_j^n(t)}{\sum_{j=0}^{K_n} w_{ji} b_j^n(t)} \quad (8)$$

Equations (7)-(8) define another set of curves parameterized in s for fixed t , i.e., for fixed t ,

$$\vec{H}(s) = \frac{\sum_{i=0}^{K_m} u_i(t) \vec{h}_i(t) b_i^m(s)}{\sum_{i=0}^{K_m} u_i(t) b_i^m(s)} \quad (9)$$

with u_i being the weights, $\vec{h}_i(t)$ the control points, and $\vec{b}_i^m(s)$ the B-spline basis functions.

In RAGGS, all the surface grid generators are formulated in the 3-D physical space, i.e., the grid points are computed directly in terms of the three Cartesian coordinates, x , y , and z . These space points are then placed exactly on the nearest quilted surface point by iterating for the patch number and parametric variables. A key element in a formulation using discrete patches is the boundary data, which is a description of how the parametric variables change as one crosses from the boundary of one patch to the boundary of an adjacent patch.

Definition Of Active Region Of Patch

As mentioned before, a surface is described by a quilt of Bezier patches or NURBS, possibly with trim curves. Each patch is parameterized in s and t . The untrimmed domain of each patch is $s_0 \leq s \leq s_1$ and $t_0 \leq t \leq t_1$, where usually $s_0 = t_0 = 0$ and $s_1 = t_1 = 1$. Trim curves (if any) are nonintersecting Bezier curves or NURBS forming closed loops in the parametric space of a patch.

$$\left\{ \begin{array}{c} s \\ t \end{array} \right\} = \vec{C}_i(\sigma) \quad i = 1, \dots, n \quad (10)$$

Here, $\vec{C}_i(\sigma)$ is the i th parameterized trim curve, σ is the parameter, and n is the number of trim curves.

The entire domain of an untrimmed patch is defined to be active. Patches which have trim curves have both active and inactive regions. By convention, the active region of a trimmed patch is the region that lies

on the left of each trim curve as a point advances along the curve. Here, left of a trim curve at a point $\vec{C}_i(\sigma)$ is the direction of the vector formed as the cross product (in the order specified) of the surface normal and the tangent vector to the trim curve at $\vec{C}_i(\sigma)$.

Patch Boundary Data

The patch boundary data is a description of how the parametric variables change as one crosses from the boundary of one patch to the boundary of an adjacent patch. Boundary data for each patch is given in terms of a series of parameter ranges for each side. Figure 2 shows a typical boundary connectivity between two adjacent patches (in this case, patches p and q). For patch p , there may be data for ranges,

$$\begin{aligned} s_{b_1} &\leq s \leq s_{b_2} \\ s_{b_3} &\leq s \leq s_{b_4} \\ &\vdots \end{aligned} \tag{11}$$

For each parameter range, an adjacent patch number and a connectivity matrix of six numbers are given. In Fig. 2, we assume that a point has crossed the boundary at (s_b, t_b) with

$$t_b = t_1, \quad s_0 \leq s_b \leq s_1 \tag{12}$$

Then the boundary data for the side $t_b = t_1$ and the range which includes s_b will give the new patch number q and the corresponding parameter values of the boundary point (t_b, s_b) on this patch q as follows:

$$\begin{pmatrix} s_b \\ t_b \end{pmatrix}_{\text{patch } q} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \begin{pmatrix} s_b \\ t_b \\ 1 \end{pmatrix}_{\text{patch } p} \tag{13}$$

where a_{ij} are the transformation matrix coefficients.

Trim Curve Boundary Data

Consider the trim curves of Eq. (10). Analogous to the patch boundaries, boundary data for trim curves is given for a series of parameter ranges of the curve being crossed as follows:

$$\begin{aligned} \sigma_{b_1} &\leq \sigma \leq \sigma_{b_2} \\ \sigma_{b_3} &\leq \sigma \leq \sigma_{b_4} \\ &\vdots \end{aligned} \tag{14}$$

When crossing trim curve boundaries, the boundary data which includes the parameter range of the crossing point σ_b will give the new patch and the corresponding trim curve on the new patch, and the parameter value of the boundary point on the corresponding trim curve as follows:

$$(\sigma_b)_{\text{trim curve on patch } q} = \begin{bmatrix} a_1 & a_2 \end{bmatrix} \begin{pmatrix} 1 - \sigma_b \\ \sigma_b \end{pmatrix}_{\text{trim curve on patch } p} \tag{15}$$

where a_1 and a_2 are the transformation matrix coefficients.

Geometry Creation and IGES/RAGGS Interface

The basic method of geometry input is through IGES files created by any CAD system. The data in the IGES file must be translated into surfaces in the form of NURBS and Bezier Patches. These are the fundamental surface forms used by RAGGS. Translating the data into the RAGGS internal format may take a number of different paths, depending on the type of data in the IGES file. A flow diagram of the process is shown in Fig. 3. The processing of the IGES file yields four different types of information which are stored for later use.

NURBS, entity type 128, contain the only type of surface data which may be used directly. The control points and other data for these entities can be read directly from the IGES file. For other surface entity types such as planes, parametric splines, surfaces of revolution, tabulated cylinders, and ruled surfaces, the data must be converted into a NURBS or rational Bezier patch format. In all cases the parameter conversion results in a surface which is mathematically equivalent to the original surface.

The second type of information used by RAGGS is information about curves. Curve entities may be in the form of parametric splines, NURBS, or point data. The curve data may be used subsequently for trim curves, surface generation, (surfaces of revolution, tabulated cylinders, ruled surfaces, etc.) or they may have other uses. Surface generation using point data involves curve fitting. This may result in a surface which is not exactly equivalent to the surface used to generate the points.

Trimmed surface data is the third type of information which is retained and processed for later use. When combined with the surface and curve data, this data may be used to create the information necessary to define the active and inactive domains on each patch. This allows the trimmed surfaces to be displayed and to create grids which cross the trim boundaries.

Ruled surface information is the fourth type of data which is stored. This data is combined with the curve data to create additional surfaces parameterized in the form of Bezier patches or NURBS.

A file containing the patch information and the related boundary data read by RAGGS is referred to as a RASCAL file.

NUMERICAL TOOLS

Various numerical tools which are fundamental to our grid generations are described herein.

Projecting A Point On A (Trimmed) Surface

A point on a quilted surface may be identified by (p, s, t) with its position vector computed by $\vec{R}(p, s, t)$. p is the patch number. For a given patch number p , s and t are the patch parametric variables and \vec{R} is a Bezier patch or NURBS taking the form of Eq. (6). Let \vec{r} denote a space point. Then, the square of the distance between \vec{R} and \vec{r} is given by

$$d = [\vec{R}(p, s, t) - \vec{r}] \cdot [\vec{R}(p, s, t) - \vec{r}] \quad (16)$$

The projection of \vec{r} is defined to be the point \vec{R} that minimizes d of Eq. (16).

A local maximum or minimum distance is found by taking partial derivatives of d with respect to s and t as follows:

$$\vec{R}_s \cdot [\vec{r} - \vec{R}] = 0 \equiv f(p, s, t) \quad (17)$$

$$\vec{R}_t \cdot [\vec{r} - \vec{R}] = 0 \equiv g(p, s, t) \quad (18)$$

Here, the subscripts s and t denote the derivatives with respect to s and t , respectively. Applying Newton's iteration method, we obtain

$$f_s \Delta s + f_t \Delta t = -f \quad (19)$$

$$g_s \Delta s + g_t \Delta t = -g \quad (20)$$

where Δs and Δt are the changes in s and t , respectively. Equations (19) and (20) are solved for $(\Delta s, \Delta t)$ at fixed p . (s, t) are then updated as follows:

$$s^{Q+1} = \begin{cases} s^Q + [\omega(|\Delta s| - \epsilon) + \epsilon] \text{sign}(\Delta s) & \text{if } |\Delta s| > \epsilon \\ s^Q + \Delta s & \text{if } |\Delta s| \leq \epsilon \end{cases} \quad (21)$$

$$t^{Q+1} = \begin{cases} t^Q + [\omega(|\Delta t| - \epsilon) + \epsilon] \text{sign}(\Delta t) & \text{if } |\Delta t| > \epsilon \\ t^Q + \Delta t & \text{if } |\Delta t| \leq \epsilon \end{cases} \quad (22)$$

Here, ω is the relaxation factor, ϵ (typically, 0.02) is the cut-off parameter and the superscript Q denotes the Q -th iteration.

During iteration, the patch number p is treated as a parameter and updated only if (s, t) is outside the active region of patch p . When the calculated (s, t) is outside the active region, the patch boundary data and the trim curve boundary data (if any) are used to find the adjacent patch to update p and (s, t) for the next iteration. In the case of patch boundary, Eq. (13) is used. In the case of trim curve, Eq. (15) is used to compute σ_b on the trim curve of the adjacent patch. The physical space point computed at σ_b on patch p is projected onto the trim curve on the adjacent patch using the computed σ_b as the initial guess. The corresponding (s, t) on the adjacent patch is determined by Eq. (10). Then the computed (s, t) is used to continue the iteration.

Projecting A Point On A Curve And Distance Between Curves

Consider two parametrically defined curves, $\vec{r}(t)$ and $\vec{\xi}(\tau)$, each taking the form of Eq. (1). The square of the distance between $\vec{r}(t)$ and $\vec{\xi}(\tau)$ is given by

$$d = [\vec{r}(t) - \vec{\xi}(\tau)] \cdot [\vec{r}(t) - \vec{\xi}(\tau)] \quad (23)$$

At any point $\vec{\xi}(\tau)$ on the second curve we define the separation between the curves as the minimum distance d of Eq. (23). This is found by taking partial derivative with respect to t while fixing τ as follows:

$$\vec{r}_t(t) \cdot [\vec{r}(t) - \vec{\xi}(\tau)] = 0 \equiv f(t, \tau) \quad (24)$$

Applying the Newton's iteration method, we obtain

$$f_t \Delta t = -f \quad (25)$$

or

$$\Delta t = -\frac{\vec{r}_t \cdot [\vec{r} - \vec{\xi}(\tau)]}{\vec{r}_t \cdot \vec{r}_t + \vec{r}_{tt} \cdot [\vec{r} - \vec{\xi}(\tau)]} \quad (26)$$

where Δt is the change in t .

Along the curves there are a series of extrema where the curves are locally separated by a maximum or minimum distance. At these points, the partial derivative of Eq. (23) with respect to τ vanishes at fixed t as follows:

$$[\vec{r}_\tau(t(\tau)) - \vec{\xi}_\tau(\tau)] \cdot [\vec{r}(t) - \vec{\xi}(\tau)] = 0 \equiv f(t(\tau), \tau) \quad (27)$$

where $t(\tau)$ is defined by the solution of Eq. (24) at fixed τ .

Applying the Newton's iteration method, we obtain

$$f_\tau \Delta\tau = -f \quad (28)$$

or

$$\Delta\tau = \frac{-(\vec{r}_i t_\tau - \vec{\xi}_\tau) \cdot (\vec{r} - \vec{\xi})}{(\vec{r}_i t_{\tau\tau} + \vec{r}_{ii} t_\tau^2 - \vec{\xi}_{\tau\tau}) \cdot (\vec{r} - \vec{\xi}) + (\vec{r}_i t_\tau - \vec{\xi}_\tau) \cdot (\vec{r}_i t_\tau - \vec{\xi}_\tau)} \quad (29)$$

Here, t_τ and $t_{\tau\tau}$ may be evaluated from Eq. (24). For example, t_τ is evaluated as follows:

$$t_\tau = \frac{\vec{r}_i \cdot \vec{\xi}_\tau}{\vec{r}_{ii} \cdot (\vec{r} - \vec{\xi}) + \vec{r}_i \cdot \vec{r}_i} \quad (30)$$

So we can choose a τ , iteratively solve for t and \vec{r} from Eq. (26), and then get $\Delta\tau$ from Eq. (29). This process continues until $\Delta\tau = 0$.

Projecting A Point On A Surface Curve

Given a point in space \vec{r} and a point on a parametrically defined curve, $\vec{R}(s(\tau), t(\tau))$, on a surface $\vec{R}(s, t)$, the square of the distance between \vec{R} and \vec{r} is given by

$$d = [\vec{R}[s(\tau), t(\tau)] - \vec{r}] \cdot [\vec{R}[s(\tau), t(\tau)] - \vec{r}] \quad (31)$$

A local maximum or minimum distance is found by taking partial derivative with respect to τ as follows:

$$\vec{R}_\tau \cdot [\vec{R} - \vec{r}] = 0 \equiv f(\tau) \quad (32)$$

Applying Newton's iteration method, we obtain

$$f_\tau \Delta\tau = -f \quad (33)$$

or

$$\Delta\tau = -\frac{\vec{R}_\tau \cdot (\vec{R} - \vec{r})}{\vec{R}_\tau \cdot \vec{R}_\tau + \vec{R}_{\tau\tau} \cdot (\vec{R} - \vec{r})} \quad (34)$$

where

$$\vec{R}_\tau = \vec{R}_s s_\tau + \vec{R}_t t_\tau \quad (35)$$

$$\vec{R}_{\tau\tau} = \vec{R}_s s_{\tau\tau} + \vec{R}_t t_{\tau\tau} + [\vec{R}_{ss} s_\tau + \vec{R}_{st} t_\tau] s_\tau + [\vec{R}_{ts} s_\tau + \vec{R}_{tt} t_\tau] t_\tau \quad (36)$$

Curve And Surface Intersections

Consider a curve $\vec{C}(u)$, and a surface $\vec{R}(s, t)$. At an intersection between \vec{C} and \vec{R} we have

$$\vec{C}(u) - \vec{R}(s, t) = 0 \equiv \vec{f}(u, s, t) \quad (37)$$

constituting 3 equations in 3 unknowns, u, s , and t . Following the Newton's iteration method, we derive an iterative process as follows:

$$-\vec{C}_u(u)\Delta u + \vec{R}_s(s, t)\Delta s + \vec{R}_t(s, t)\Delta t = \vec{C}(u) - \vec{R}(s, t) \quad (38)$$

Isolating terms with cross and dot products, we have 3 equations,

$$-\vec{R}_s(s, t) \cdot [\vec{R}_t(s, t) \times \vec{C}_u(u)] \Delta u = \vec{R}_s(s, t) \cdot \left\{ \vec{R}_t(s, t) \times [\vec{C}(u) - \vec{R}(s, t)] \right\} \quad (39)$$

$$\vec{C}_u(u) \cdot [\vec{R}_t(s, t) \times \vec{R}_s(s, t)] \Delta s = \vec{C}_u(u) \cdot \left\{ \vec{R}_t(s, t) \times [\vec{C}(u) - \vec{R}(s, t)] \right\} \quad (40)$$

$$\vec{C}_u(u) \cdot [\vec{R}_s(s, t) \times \vec{R}_t(s, t)] \Delta t = \vec{C}_u(u) \cdot \left\{ \vec{R}_s(s, t) \times [\vec{C}(u) - \vec{R}(s, t)] \right\} \quad (41)$$

Solving these equations, we have the iterative process,

$$\Delta u = \frac{\vec{R}_s \cdot [\vec{R}_t \times \vec{f}]}{\vec{C}_u \cdot \vec{n}}, \quad \Delta s = \frac{\vec{C}_u \cdot [\vec{R}_t \times \vec{f}]}{\vec{C}_u \cdot \vec{n}}, \quad \Delta t = \frac{\vec{C}_u \cdot [\vec{R}_s \times \vec{f}]}{\vec{C}_u \cdot \vec{n}} \quad (42)$$

since

$$\vec{C}_u \cdot [\vec{R}_s \times \vec{R}_t] = -\vec{C}_u \cdot [\vec{R}_t \times \vec{R}_s] = \vec{R}_s \cdot [\vec{R}_t \times \vec{C}_u] = \vec{C}_u \cdot \vec{n} \quad (43)$$

where $\vec{n} = \vec{R}_s \times \vec{R}_t$ is the surface normal at (s, t) , and $\vec{f} = \vec{C} - \vec{R}$. For a quilted (trimmed) surface, patch number p will enter the iterative process and play the same role as that in section "Projecting A Point On A (Trimmed) Surface."

Surface And Surface Intersections

When surface intersections are not given by trim curves, an intersection must be computed. Two surfaces generally intersect along a curve or curves in space. Initial points on any intersection curves may be found by creating a mesh of curves, usually constant parameter curves, on one of the surfaces, and then finding the intersection of these curves with the other surface. The mesh must be of sufficient density to find at least one point on all curves of intersection, but not too dense and causing an inordinate number of computations. When a point on any intersection curve is found, further intersection points are easily created along this curve. This curve may then be described parametrically by fitting an interpolating function through the generated points. This procedure is followed for each curve of intersection.

Consider two surfaces given by the equations

$$\vec{R}_1 = \vec{P}(s, t) \quad (44)$$

$$\vec{R}_2 = \vec{Q}(u, v) \quad (45)$$

each taking the form of Eq. (6). At an intersection point, we have 3 equations in 4 unknowns s, t, u , and v .

$$\vec{P}(s, t) - \vec{Q}(u, v) = 0 \equiv \vec{f}(s, t, u, v) \quad (46)$$

Following Newton's iteration method, we have

$$\vec{P}_s(s, t)\Delta s + \vec{P}_t(s, t)\Delta t - \vec{Q}_u(u, v)\Delta u - \vec{Q}_v(u, v)\Delta v = -\vec{f} \quad (47)$$

Since the system is overspecified, we need another equation for a unique solution.

Holding one of the parameters constant reduces the problem to a curve and surface intersection. Holding a parameter constant is also useful for finding a point on a particular intersection curve, such as the patch boundary. We could also hold the parameter constant which maximizes the denominator of Eq. (42) for the curve and surface intersection. Another way is to approach perpendicularly to the intersection curve along one of the surfaces. The tangent vector to the intersection curve $\vec{\tau}$, is given by,

$$\vec{\tau} = \vec{n}_P \times \vec{n}_Q \quad (48)$$

where \vec{n}_P and \vec{n}_Q are the local normals at (s, t) and (u, v) , respectively,

$$\vec{n}_P = \vec{P}_s \times \vec{P}_t, \quad \vec{n}_Q = \vec{Q}_u \times \vec{Q}_v \quad (49)$$

The movement vector on surface \vec{P} is given by $\vec{P}_s\Delta s + \vec{P}_t\Delta t$. The trace of the movement may be written in parameterized form as follows:

$$\vec{C}_s\Delta s = \vec{P}_s\Delta s + \vec{P}_t\Delta t \quad (50)$$

with the direction of movement constrained to the normal of the intersection curve, i.e.,

$$\vec{\tau} \cdot (\vec{P}_s\Delta s + \vec{P}_t\Delta t) = 0 \quad (51)$$

Now, the problem has been reduced to the intersection of curve $\vec{C}(s)$ and surface $\vec{Q}(u, v)$. Following section "Curve And Surface Intersection" and from Eq. (42), we have

$$\Delta s = -\frac{\vec{Q}_u \cdot [\vec{Q}_v \times \vec{f}]}{\vec{C}_s \cdot \vec{n}_Q}, \quad \Delta u = -\frac{\vec{C}_s \cdot [\vec{Q}_v \times \vec{f}]}{\vec{C}_s \cdot \vec{n}_Q}, \quad \Delta v = \frac{\vec{C}_s \cdot [\vec{Q}_u \times \vec{f}]}{\vec{C}_s \cdot \vec{n}_Q} \quad (52)$$

Here, \vec{C}_s may be calculated from Eqs. (50)-(51) as follows:

$$\begin{aligned} (\vec{\tau} \cdot \vec{P}_t)\vec{C}_s\Delta s &= (\vec{\tau} \cdot \vec{P}_t)(\vec{P}_s\Delta s + \vec{P}_t\Delta t) \\ &= [(\vec{\tau} \cdot \vec{P}_t)\vec{P}_s - (\vec{\tau} \cdot \vec{P}_s)\vec{P}_t] \Delta s \end{aligned} \quad (53)$$

$$(\vec{\tau} \cdot \vec{P}_t)\vec{P}_s - (\vec{\tau} \cdot \vec{P}_s)\vec{P}_t = [(\vec{n}_P \times \vec{n}_Q) \times \vec{n}_P] \Delta s \quad (54)$$

Since

$$(\vec{\tau} \cdot \vec{P}_t)\vec{C}_s \cdot \vec{n}_Q = [(\vec{n}_P \times \vec{n}_Q) \times \vec{n}_P] \cdot \vec{n}_Q = (\vec{n}_P \times \vec{n}_Q) \cdot (\vec{n}_P \times \vec{n}_Q) \quad (55)$$

we cannot have a nonzero denominator unless $\vec{n}_P \times \vec{n}_Q = 0$, which means the surfaces are locally parallel and there is no intersection anyway. Substituting in the above we get the iteration process,

$$\Delta s = -\frac{[(\vec{n}_P \times \vec{n}_Q) \cdot \vec{P}_t] [\vec{Q}_v \times \vec{f}] \cdot \vec{Q}_u}{(\vec{n}_P \times \vec{n}_Q) \cdot (\vec{n}_P \times \vec{n}_Q)} \quad (56)$$

$$\Delta t = \frac{[(\vec{n}_P \times \vec{n}_Q) \cdot \vec{P}_s] [\vec{Q}_v \times \vec{f}] \cdot \vec{Q}_u}{(\vec{n}_P \times \vec{n}_Q) \cdot (\vec{n}_P \times \vec{n}_Q)} \quad (57)$$

$$\Delta u = -\frac{[(\vec{n}_P \times \vec{n}_Q) \times \vec{n}_P] \cdot [\vec{Q}_v \times \vec{f}]}{(\vec{n}_P \times \vec{n}_Q) \cdot (\vec{n}_P \times \vec{n}_Q)} \quad (58)$$

$$\Delta v = \frac{[(\vec{n}_P \times \vec{n}_Q) \times \vec{n}_P] \cdot [\vec{Q}_u \times \vec{f}]}{(\vec{n}_P \times \vec{n}_Q) \cdot (\vec{n}_P \times \vec{n}_Q)} \quad (59)$$

As previously defined, \vec{n}_P and \vec{n}_Q are in the direction of the local normals.

$$\vec{n}_P = \vec{P}_s \times \vec{P}_t, \quad \vec{n}_Q = \vec{Q}_u \times \vec{Q}_v, \quad \vec{f} = \vec{P}(s, t) - \vec{Q}(u, v) \quad (60)$$

When the convergence is satisfied, the next point may be found by marching along the surfaces in the direction of the tangent vector $\vec{r} = \vec{n}_P \times \vec{n}_Q$, and iterating again. Along the tangent vector,

$$\Delta \vec{R} = \vec{P}_s \Delta s + \vec{P}_t \Delta t = \frac{\vec{r}}{|\vec{r}|} \Delta \sigma \quad (61)$$

or

$$\Delta \vec{R} = \vec{Q}_u \Delta u + \vec{Q}_v \Delta v = \frac{\vec{r}}{|\vec{r}|} \Delta \sigma \quad (62)$$

where σ is the arc length. Then Δs and Δt along the intersection curve may be calculated by the vector cross product of Eq. (61) and \vec{P}_s and \vec{P}_t , respectively,

$$-(\vec{P}_s \times \vec{P}_t) \Delta s = \vec{P}_t \times [(\vec{P}_s \times \vec{P}_t) \times \vec{n}_Q] \frac{\Delta \sigma}{|\vec{r}|} = (\vec{P}_s \times \vec{P}_t) (\vec{P}_t \cdot \vec{n}_Q) \frac{\Delta \sigma}{|\vec{r}|} \quad (63)$$

$$\Delta s = -(\vec{P}_t \cdot \vec{n}_Q) \frac{\Delta \sigma}{|\vec{r}|} \quad (64)$$

$$\Delta t = (\vec{P}_s \cdot \vec{n}_Q) \frac{\Delta \sigma}{|\vec{r}|} \quad (65)$$

Similarly, Δu and Δv are obtained by the vector cross product of Eq. (62) and \vec{Q}_u and \vec{Q}_v , respectively,

$$\Delta u = (\vec{Q}_v \cdot \vec{n}_P) \frac{\Delta \sigma}{|\vec{r}|} \quad (66)$$

$$\Delta v = -(\vec{Q}_u \cdot \vec{n}_P) \frac{\Delta \sigma}{|\vec{r}|} \quad (67)$$

Here, $\Delta \vec{r}$ is computed as follows:

$$\begin{aligned} \Delta \vec{r} &= \vec{r}_s \Delta s + \vec{r}_t \Delta t + \vec{r}_u \Delta u + \vec{r}_v \Delta v \\ &= \left[-\vec{r}_s (\vec{P}_t \cdot \vec{n}_Q) + \vec{r}_t (\vec{P}_s \cdot \vec{n}_Q) + \vec{r}_u (\vec{P}_v \cdot \vec{n}_P) - \vec{r}_v (\vec{P}_u \cdot \vec{n}_P) \right] \frac{\Delta \sigma}{|\vec{r}|} \end{aligned} \quad (68)$$

Surface And Plane Intersections

For the user's convenience, a space plane is specified by a set of three non-colinear points or by a reference point and a normal to the plane. Intersection between a quilted surface and the user-specified plane is done exactly in the same manner as that described in section "Surface And Surface Intersections" once the plane is represented by a quilt containing a single patch.

SURFACE GRID GENERATION

Both structured and unstructured grid generators used in RAGGS are briefly described in this section. References will be cited for readers who are interested in details of the numerical algorithms actually implemented for the surface grid generations.

Structured Surface Grid Generation

TFI Method.—The transfinite interpolation (TFI) of Ref. 2 is coupled with the space point projection, Eqs. (16)-(22), to determine the interior grids of a 4-sided block. A subsection approach is employed. The approach allows a 4-sided block grid be calculated subsection by subsection. Each subsection is defined by its two opposite diagonal points and can easily be specified interactively by point-and-click. A subsection grid is computed in two steps. First, the 4-sided subsection grid is calculated by TFI based on the sectional edge grids. Second, Eqs. (19)-(20) are used to project the TFI grids onto the quilted surface. At each point, a set of initial guesses (p, s, t) is formed by gathering all the available values at immediately adjacent grid points. Additional sets of (p, s, t) may also be specified by user. The solution is taken to be the one that has the minimum distance among all the calculated solutions.

When a grid is generated on a single patch, TFI may be optionally applied to the surface parametric variables. In this case, the grid position vectors are evaluated directly from the patch considered using the TFI interpolated values of parametric variables.

PDE Method.—The elliptic grid generation system on arbitrary curved space surface can be written, in tensor form³, as follows:

$$\sum_{m=1}^2 \sum_{n=1}^2 gg^{mn} \bar{r}_{\xi^m \xi^n} + \sum_{n=1}^2 gg^{nn} P_n \bar{r}_{\xi^n} - gH \bar{n} = 0 \quad (69)$$

Here \bar{r} is the grid position vector, ξ^i are the curvilinear coordinates, P_n are the grid control functions, g^{ij} is the contravariant metric tensor, and $g = \det(g_{ij})$ with g_{ij} the covariant metric tensor. H is the twice of the surface mean curvature and \bar{n} is the unit surface normal. Equation (69) is solved numerically using the relaxation methods described in Ref. 4 once the differential operators are replaced by central-difference operators. Equations (19)-(20) are used to snap the solution grid points onto the quilted surface at the end of each iteration.

The grid control functions P_n are evaluated at the 4-sided block edges and the boundary values are interpolated into the field. Both Thomas/Thomas-Middlecoff's method^{5,6} and Sorenson's method^{7,8} have been implemented for the construction of orthogonal boundary grids. However, Sorenson's method is implemented based on Thompson's method of Ref. 2 with the following two modifications. One, the central-difference operator is used to approximate the second-order derivatives normal to the boundary instead of the one-sided difference operator. Second, the grid control functions are updated immediately after the contributions from a given orthogonal boundary section have been evaluated and the control function increments are evaluated against the most current values. The first modification provides a consistent numerical method in the sense that both the grid position vector and control functions satisfy the same governing equation as the solution converges, and a grid initially orthogonal will remain and be returned as the solution. The second modification has provided a more stable iterative process.

Similarity Method.—An existing grid may be fine-tuned to have a similar or identical distribution in a specified base variable, the arc length or one of the three Cartesian coordinates, along a specified coordinate direction. To accomplish this, the edge distributions of the specified base variable are interpolated into the field. Then, the grid points are re-evaluated line by line in the specified coordinate direction and projected onto the quilted surface. This can provide a simple means of achieving a cylindrical-coordinate-like grid. Also, it can provide panels on an aircraft configuration with constant spanwise or axial sections as required by some panel codes.

Marching Method.—A grid may be constructed on a quilted surface from a given line grid (open or

closed) by marching in the direction normal to the line segment with a specified step size. The marching direction is the direction of the vector formed as the cross product (in the order specified) of the surface normal and the tangent vector to the line segment.

Smoothing.—An existing grid may be smoothed by replacing each of the interior points by the averaged value of its adjacent grid points. The number of passes of smoothing is specified by the user.

Unstructured Surface Grid Generation

Details of our unstructured grid generation algorithm may be found in Ref. 9. Briefly, the technique is based closely on the Löhner-Parikh's¹⁰ data structures and the 2-D advancing front triangulation scheme described in Ref. 11. However, the implementation is made in the 3-D physical space coupled with the space point projection, Eqs. (16)-(22). Interactively, a grid generation starts with a rectangular octree box specified just big enough to cover the entire region to be triangulated. The advancing front is constructed automatically as the user is pointing and clicking line segments on the screen. During triangulation, every new nodal point is introduced to form a nearly equilateral triangle and satisfy the mesh consistency requirement with a base selected from the front. Equations (19)-(20) are used to iteratively locate the new nodal point on the quilted surface.

Since the unstructured grid generation is based on the same geometry database as that of the structured grid generation, geometry manipulators and domain blocking tools developed for structured grid generation can be used directly. With this new capability, RAGGS provides a unique environment where both structured and unstructured grids can be generated and manipulated on a common geometry database.

SAMPLE CASES

Sample cases are presented herein to demonstrate the capabilities of RAGGS in handling trimmed surfaces and generating grids on an unstructured quilt of patches and/or trimmed surfaces.

Figure 4 demonstrates the capability of RAGGS to display trimmed surfaces. Shown in the figure is a portion of an impeller. The geometry is defined by a series of trimmed surfaces. Figures 4-(a) and (b) show the shaded displays of the untrimmed and trimmed patches, respectively. One can hardly picture what the geometry would look like from the untrimmed surface display in Fig. 4-(a). But it is clearly shown in Fig. 4-(b). Figures 4-(c) and (d) show how the two patches corresponding to the front face and the right side of the geometry are discretized for display purposes. In RAGGS a structured grid is used to discretize the patch in the parametric space with the grid size adaptive to the local curvature in each direction of the parametric variable as shown in Fig. 4-(c). For a trimmed patch, the active region is automatically triangulated in parametric space, using the advancing front technique of Ref. 9, with the trim curves as the initial front. Figure 4-(d) shows the triangulation of the active regions of the two patches shown in Fig. 4-(c). It may be noted that in RAGGS the trimmed and untrimmed displays can easily be toggled on and off. Also noted is that no additional patch definition is given to the trimmed patch and the definition of the original untrimmed patch is used for the geometry.

Figure 5 demonstrates the capability of RAGGS for generating structured and unstructured grids on unstructured quilts of patches. Figure 5-(a) shows a B-1B forebody with fins. Four quilts are shown here: the upper and lower body quilts and the upper and lower fin quilts. No trim curves were provided so a surface intersection was computed between the body and fin quilts. The irregular arrangement of patches is clearly indicated by the wireframe display in Fig. 5-(a). The calculated structured and unstructured grids are shown in Figs. 5-(b) and (c), respectively.

Figure 6 demonstrates the capability of RAGGS for gridding on trimmed surfaces. Figure 6-(a) shows a plane and a bump (part of semi-spherical surface represented by NURBS). The four edge grids are specified entirely on the flat plane, and the interior grids are generated while considering the trimmed surfaces. As shown in Fig. 6-(b), the interior grid points have been properly positioned on the trimmed patches. It is noted that the untrimmed surfaces are the only surfaces used to define the geometry.

CONCLUSIONS

Capability of generating structured and unstructured grids directly on CAD generated geometry with/without trimmed surfaces has been described and demonstrated. Both geometry integrity and patch structure are retained from IGES file to CFD geometry database.

REFERENCES

1. Woan, C-J, Clever, W. C., and Tam, C. K., "RAGGS: Rockwell Automated Grid Generation System," AIAA Paper AIAA-94-0206, AIAA 32nd Aerospace Sciences meeting & Exhibit, January 10-13, 1994.
2. Thompson, Joe. F. and Gatlin Boyd, "Program EAGLE User's Manual: Vol III-Grid Generation Code," AFATL-TR-88-117, Vol III, Airforce Armament Laboratory, Eglin Air Force Base, Florida, September 1988.
3. Thompson, Joe. F., Warsi, Z. U. A., and Mastin, C. Wayne, *NumericalGridGeneration*, North-Holland-New York-Oxford, 1985.
4. Woan, C-J, "Multi-Block Multigrid Calculations of a System of Elliptic Grid Generators," AIAA Paper 88-0312, AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, January 11-14, 1988.
5. Thomas, P. D. and Middlecoff, J. F., "Direct Control of the grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA Journal, Vol. 18, No. 6, pp. 652-656, June, 1980.
6. Thomas, P. D., "Composite Three-Dimensional Grids Generated by Elliptic Systems," AIAA Journal, Vol. 20, No. 9, pp. 1195-1202, September 1982.
7. Sorenson, Reese L., "A Computer Program to Generate Two-Dimensional Grids about Airfoils and Other Shapes by Use of Poisson's Equation," NASA TM-81198, May 1980.
8. Sorenson, Reese L., "Three-Dimensional Elliptic Grid Generation about Fighter Aircraft for Zonal Finite-Difference Computations," AIAA Paper 86-0429, AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 6-9, 1986.
9. Woan, C-J, "Unstructured Surface Grid Generation on Unstructured Quilt of Patches," AIAA Paper AIAA-95-2202, AIAA 26th Fluid Dynamics Conference, June 19-22, 1995.
10. Löhner, R. and Parikh, P., "Generation of Three-Dimensional Unstructured Grids by the Advancing-Front Method," AIAA Paper AIAA-88-0515, AIAA 26th Aerospace Sciences Meeting, January 11-14, 1988.
11. Peraire, J., Peiro, J., Formaggia, L., Morgan, K., and Zienkiewicz, O. C., "Finite Element Euler Computation in Three Dimensions," AIAA Paper AIAA-88-0032, AIAA 26th Aerospace Sciences Meeting, January 11-14, 1988.

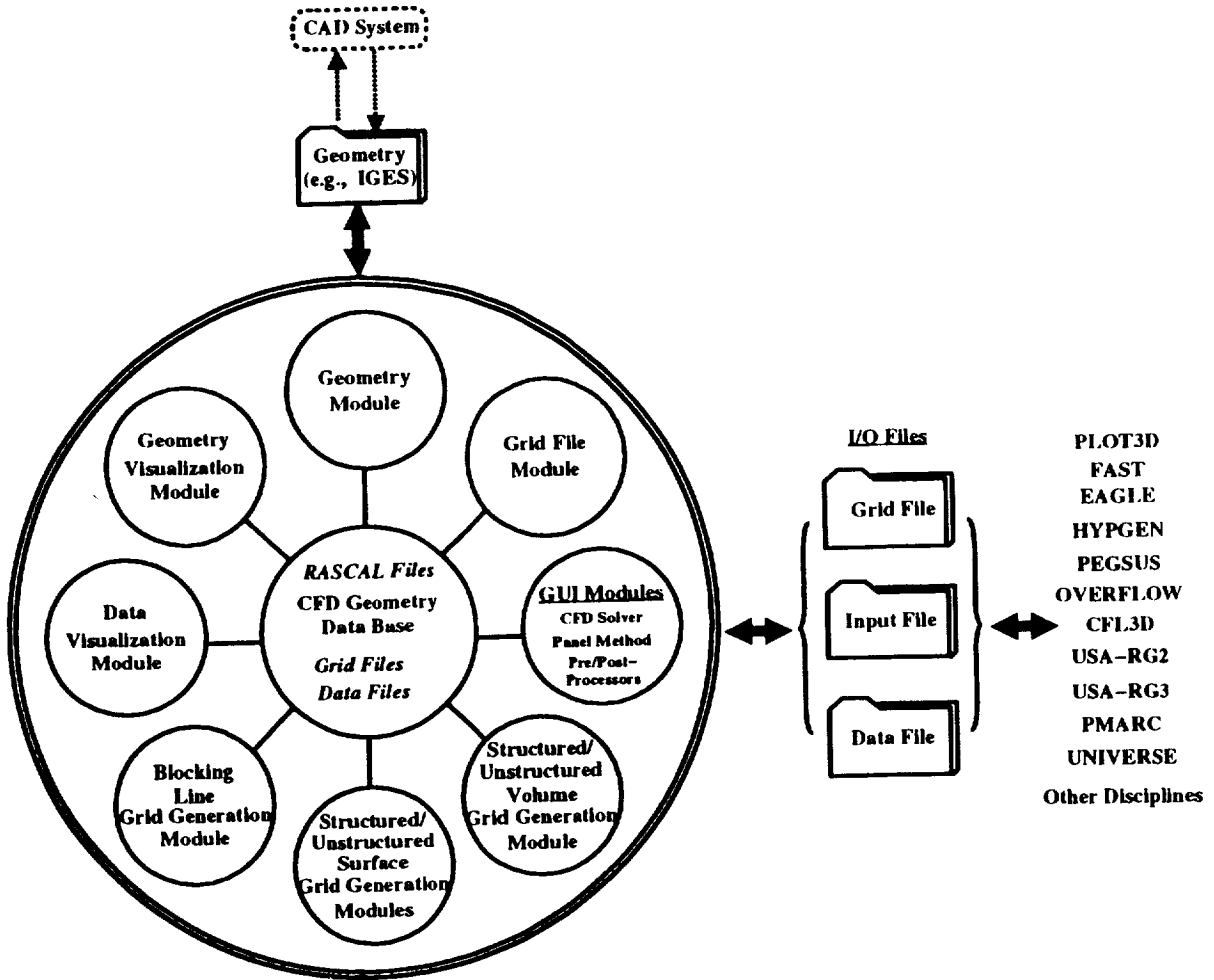


Figure 1.-The program structure of RAGGS.

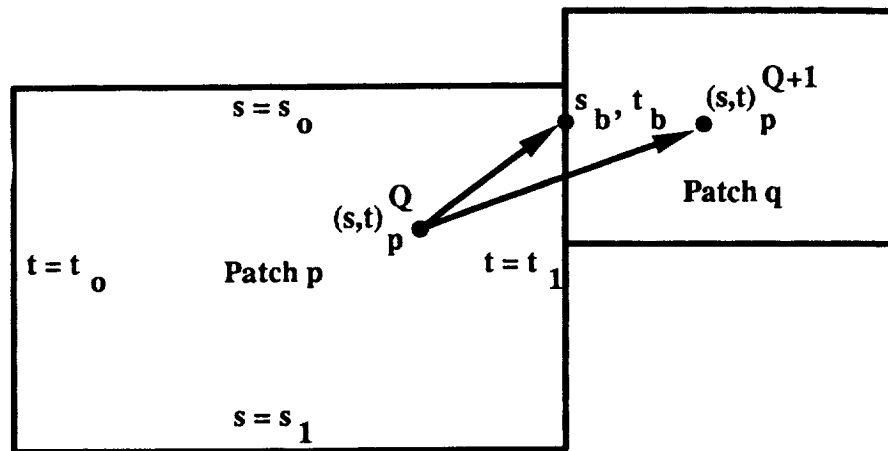


Figure 2.-Typical boundary connectivity between two adjacent patches and determination of new patch and parametric variables during iteration.

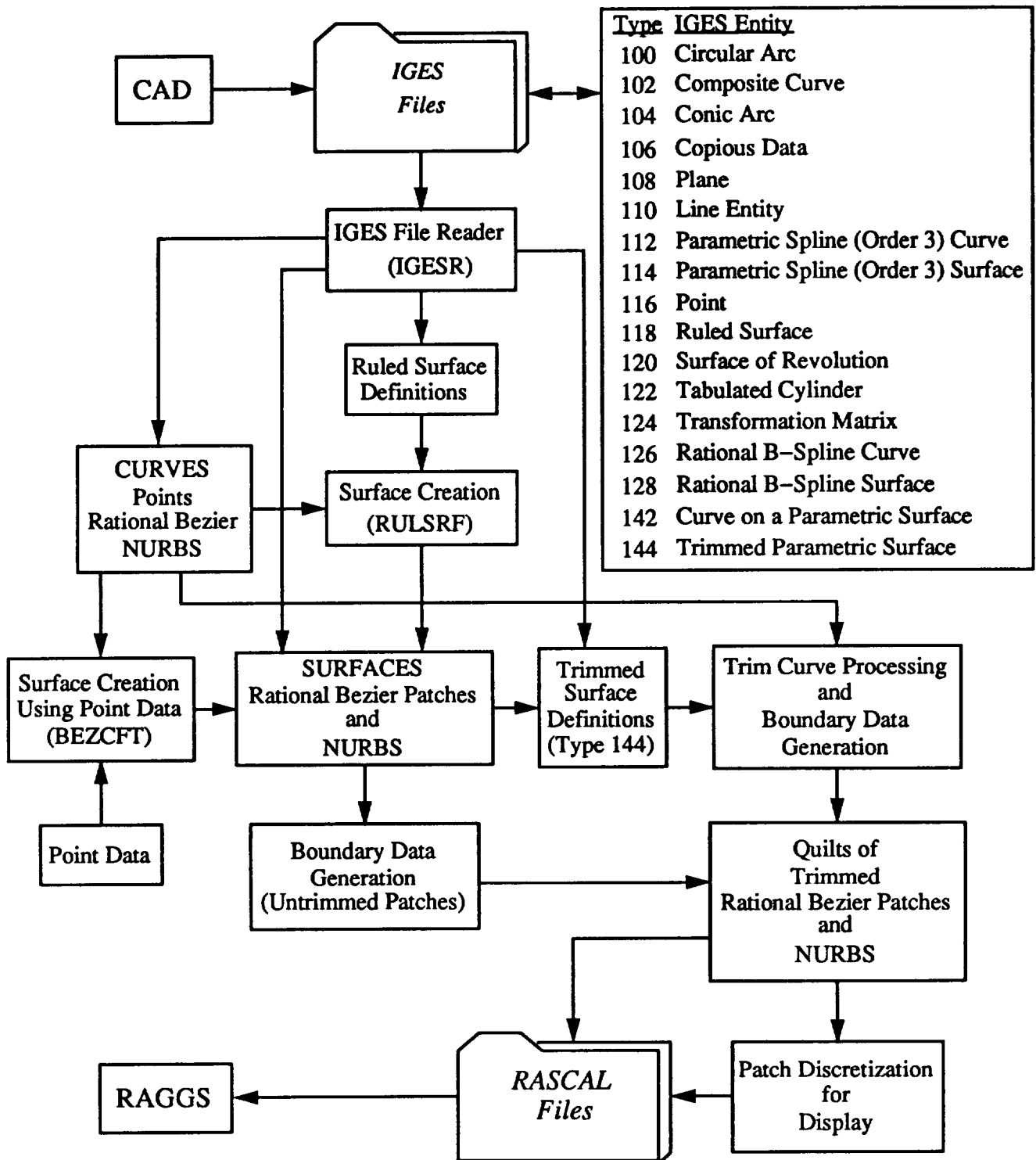
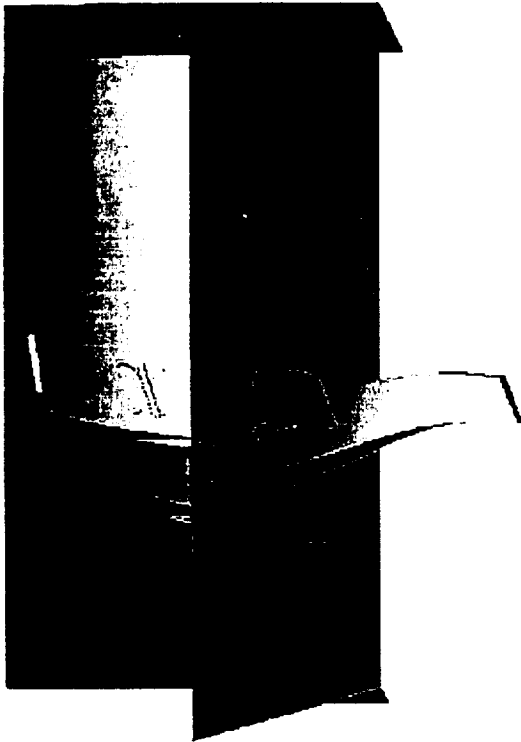
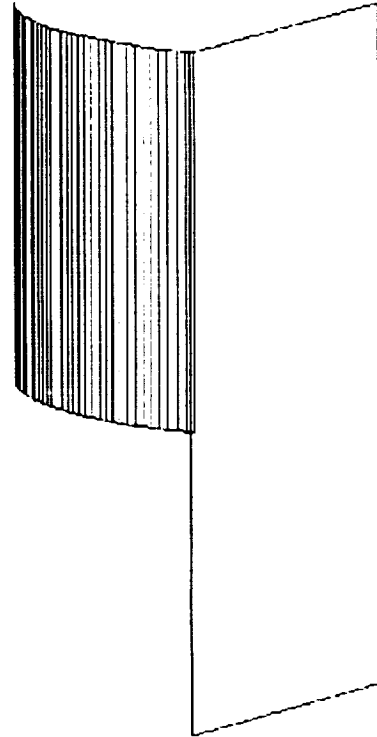


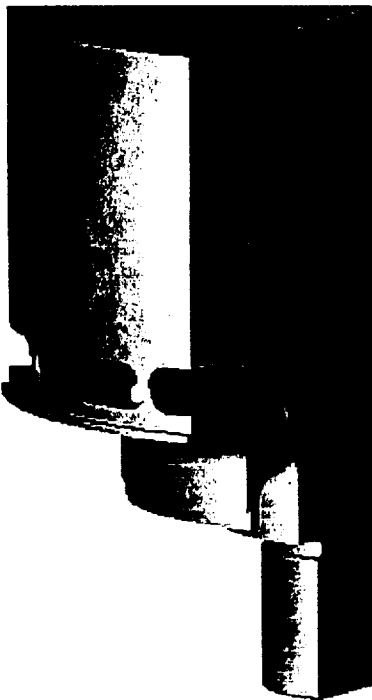
Figure 3.-The IGES/RAGGS interface.



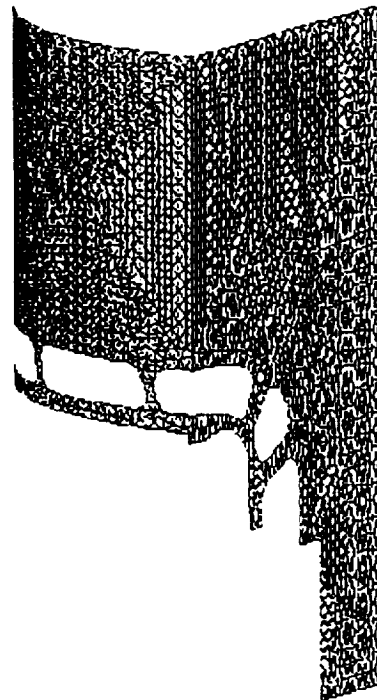
(a) Shaded display of untrimmed surfaces



(c) Structured Polygons for untrimmed patches

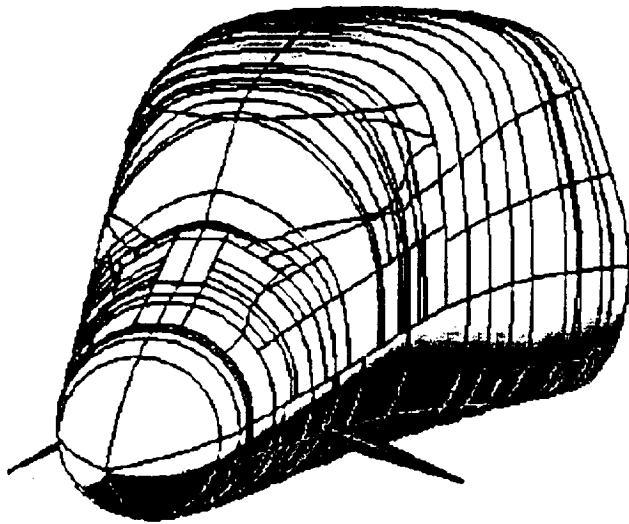


(b) Shaded display of trimmed surfaces

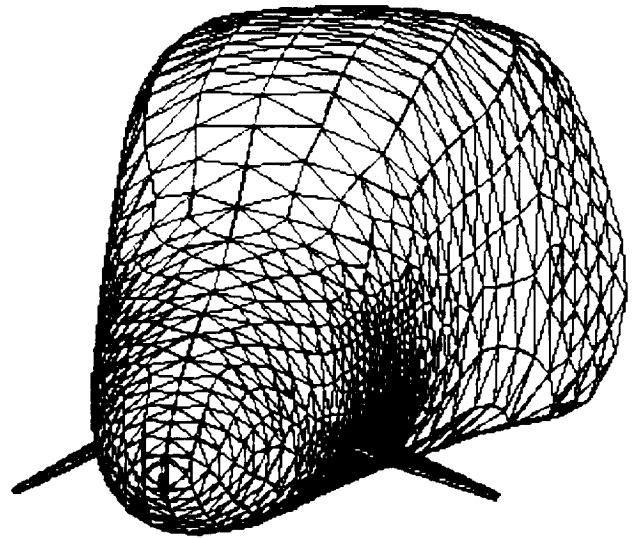


(d) Triangulation of trimmed patches

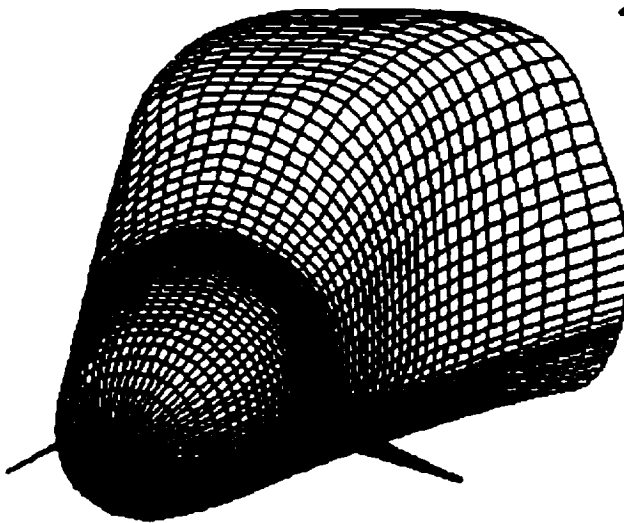
Figure 4.-Capability of displaying trimmed and untrimmed surfaces.



(a) Bezier patch-defined B-1b forebody showing unstructured quilt of patches

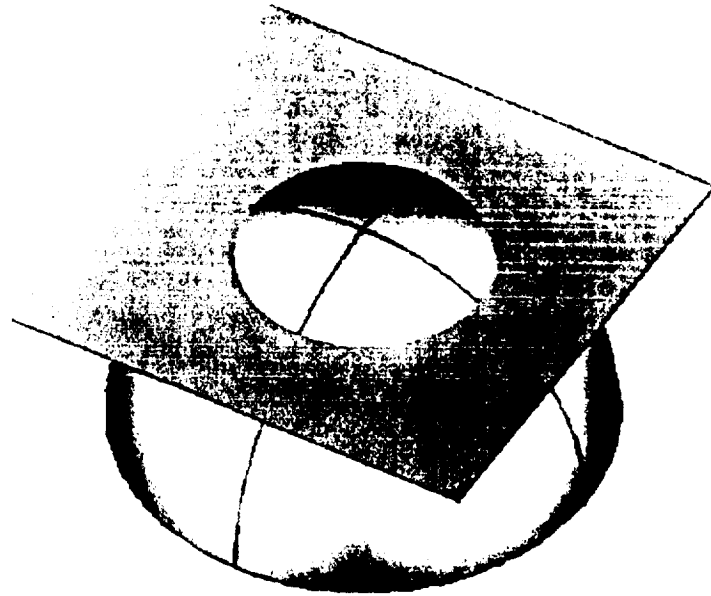


(c) Unstructured surface grid

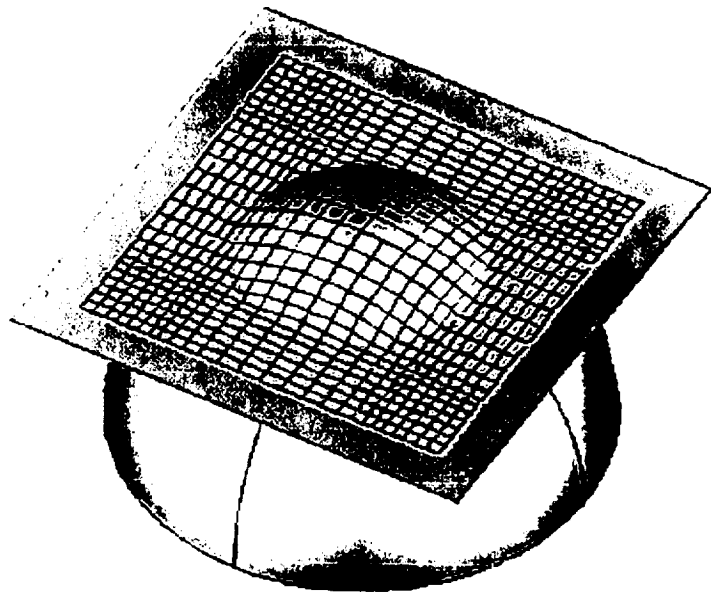


(b) Structured surface grid

Figure 5.—Capability of structured and unstructured grid generations on unstructured quilt of patches.



(a) Patch-defined plane and bump



(b) A structured grid on the trimmed surface

Figure 6.-Capability of gridding on trimmed surfaces.