Advances in Hydro-Science and -Engineering, Volume I
Sam S.Y. Wang (ed.) *ISBN 0-937099-02-3 (2-Part Set)*

presented at First Int. Conf. on Hydro-Science & Engineering, Washington, D.C., June 1993.

COMPUTATIONAL GEOMETRY TOOLS IN GRID GENERATION

Bharat Soni* and Bernd Hamann**

ABSTRACT

The B-spline approximation techniques applicable to surface grid generation pertinent to computational field simulations are presented. A systematic procedure to first correct design process errors and then unite selected surface patches yielding an overall continuous approximation is developed. The application of these techniques to surface grid redistribution, automatic remapping, and elliptic refinement is presented.

INTRODUCTION

Very often, CAD/CAM data files resulting from design processes contain errors, such as gaps between surface patches and intersections of patches. The correction of these errors are extremely crucial in the generation of surface and volume grids applicable to field simulations. Geometric design and computer aided geometric design techniques [1-4] have been utilized to eliminate design errors for proper surface generation. The resulting B-spline surfaces are utilized in development of well-distributed and well-refined surface grids [5-9].

B-SPLINE APPROXIMATION

The creation of a single B-spline cubic patch approximating a part of the given geometry requires four points on the original surface in either clockwise or counter-clockwise order. These points determine a 3D hexahedral block used for the selection of the surfaces to be approximated. The "top" and "bottom" face are computed by considering normal vectors at the specified quadrilateral's corners and the lengths of its edges. A pictorial view of four selected surface points, unit normal vectors at these points, and edge lengths are shown in figure 1a, and the constructed block is shown in figure 1b.

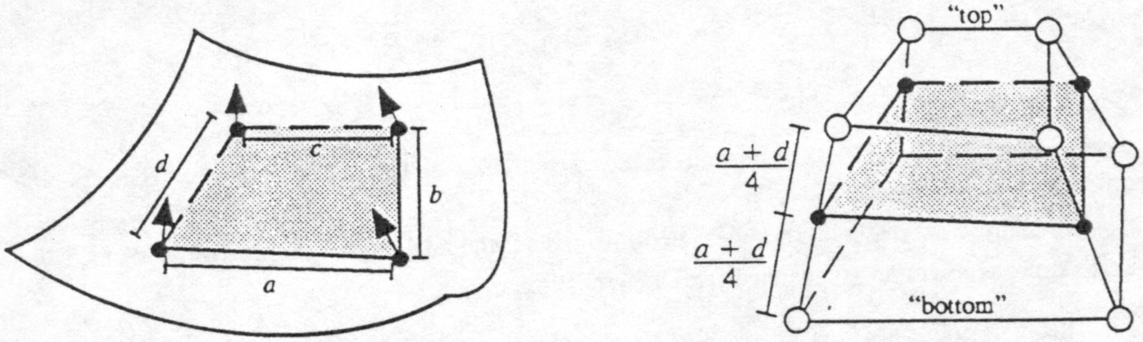

Fig. 1a. Surface and four selected points

Fig. 1b. Implied block

Fig. 1. Block constructed from quadrilateral

Bezier, B-spline, and NURBS surfaces lie within the convex hull of their control points [1,2]. Therefore, such surfaces can not have interior block points if the bounding box of their control points has no intersection

* Associate Professor, Aerospace Engineering, Mississippi State University, Grid Generation Thrust Leader, NSF Engineering Research Center for Computational Field Simulation, Sr. Member, AIAA

** Assistant Professor, Computer Science, Mississippi State University, Visualization Thrust, NSF Engineering Research Center for Computational Field Simulation with the bounding box of the block. All triangles used for the surfaces' discretization lying completely or partly inside a block are kept. The clipping of surface triangles against block faces is illustrated in figure 2.

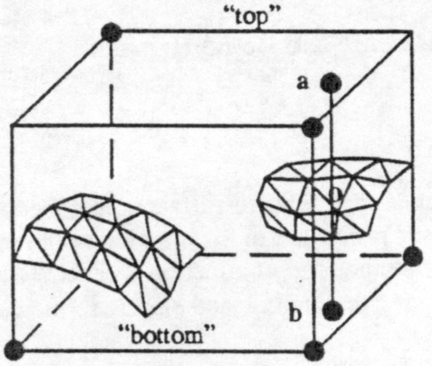

Fig. 2. Clipping surface triangles against block faces

In order to compute a B-spline approximation for the surface lying inside a block, a finite set of $(M + 1)(N + 1)$ points is generated. These points are interpolated yielding the local surface approximation. The finite set of surface points is obtained by intersecting the surface triangles inside the block with line segments. Each line segment $\overline{ab}$ is defined by a pair of corresponding points on opposite block faces, one point on the "top" face, the other one on the "bottom" face. A line segment can intersect exactly one triangle, more than one triangle, or no triangle at all. If at least one intersection point is found the point closest to the block's "top" face is chosen for the approximation.

Since exactly $(M + 1)(N + 1)$ points are interpolated for the local approximation, additional points are needed in case that certain line segments do not intersect any triangle. Each intersection point can be written as $(1 - t)\ a + tb,\ t \in [0, 1]$. Additional points are generated using Hardy's reciprocal multiquadric method [3] to compute a bivariate function $t\ (u, v)$. The bivariate interpolation constraints are obtained by considering all intersection points found and using a uniform parametrization for the bilinear "top" and "bottom" faces,

$$t_{I,J} = t\ (u_I, v_J) = \sum_{j \in \{0,\dots,N\}} \sum_{i \in \{0,\dots,M\}} c_{i,j} \left(R + (u_I - u_i)^2 + (v_J - v_j)^2\right)^{-0.001}, \qquad I \in \{0,\dots,M\}, J \in \{0,\dots,N\}, \tag{2}$$

where only those values $t_{I,J}$, u_I, u_i, v_J, and v_j are considered for which an intersection point has been found. Using uniform parameter spacing $(\delta_u = u_{i+1} - u_i, i = 0 ... (M - 1),\ \delta_v = v_{j+1} - v_j,\ j = 0 ... (N - 1))$, the values $R = \frac{1}{2}(\delta_u + \delta_v)$ and the exponent -0.001 yield good results. If the analytical definition of all surfaces is known all intersection points found are mapped onto the real surfaces.

The $(M + 1)(N + 1)$ points are interpolated by a C^2 continuous, piecewise bicubic B-spline surface. The resulting B-spline surface of order 4 is given by

$$s(u, v) = \sum_{j=0}^{n} \sum_{i=0}^{m} d_{i,j} N_{i,4}\ (u)\ N_{j,4}\ (v), \qquad u \in [u_3, u_{m+1}], v \in [v_3, v_{n+1}], \tag{3}$$

where d_{ij} is a B-spline control point and $N_{i,4}\ (u)$ and $N_{j,4}\ (v)$ are normalized B-spline basis functions defined on two pre-computed knot sequences $[u_0 < u_1 < ... < u_{m+4}]$ and $[v_0 < v_1 < ... < v_{n+4}]$ [1].

In this application, the domain of a B-spline surface, which is $[u_3, u_{m+1}]\ x\ [v_3, v_{n+1}]$, is normalized by setting $u_3 = v_3 = 0$ and $u_{m+1} = v_{n+1} = 1$. Therefore, the linear system to be solved becomes

$$x_{I,J} = s(u_{3+I}, v_{3+J}) = \sum_{j=0}^{N+2} \sum_{i=0}^{M+2} d_{i,j} N_{i,4}\ (u_{3+I})\ N_{j,4}\ (v_{3+J}), \qquad I = 0 ... M, J = 0 ... N. \tag{4}$$

Once a local B-spline approximation is determined, an error estimate is computed. This estimate is the root-mean-square error estimate based on shortest distances between points on the approximating B-spline surface and the approximated surface(s). This error estimate can be computed, regardless, whether an analytical definition of the original surfaces is known or not.

CONNECTING B-SPLINE SURFACES

Topologically, all approximating B-spline surfaces are four-sided entities having at most four neighbors. Two B-spline surfaces are considered neighbors when sharing a common boundary curve. All B-spline surfaces used in the approximation must satisfy these topological constraints: (i) Each boundary curve of a B-spline surface is shared by at most two surfaces (no bifurcations). (ii) A corner vertex of a B-spline surface can be shared by any number of surfaces. (iii) Each B-spline surface has at least one point along one of its boundary curves in common with another surface ("connectivity"). (iv) If a corner vertex of a B-spline surface is shared by a second surface then this point is also a corner vertex of the second surface ("full-face interface"). Examples of such topologies are illustrated in figure 3.

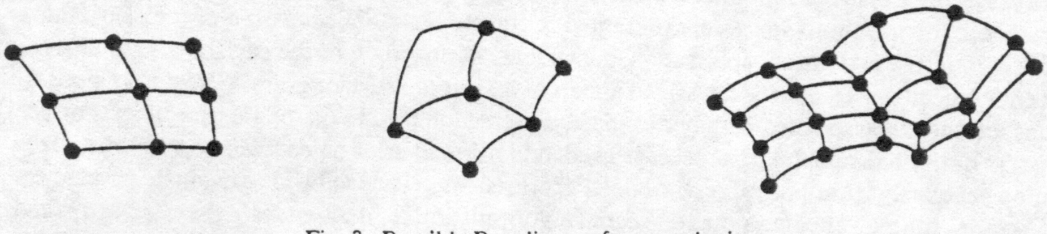

Fig. 3. Possible B-spline surface topologies

A connectivity table stores the (up to) four neighbors of each surface. Two surfaces are identified as neighbors if an estimate for the "distance" between pairs of boundary curves of two surfaces is smaller than some tolerance. Once determined, this connectivity table is used in the enforcement of continuity along edges of neighbor surfaces and at surface corner vertices shared by multiple surfaces.

All surfaces constructed must be compatible, i.e., they must be piecewise bicubic surfaces, all having the same number of control points and the same knots along common boundary curves. If this condition is violated surfaces are made compatible by performing degree raising and knot insertion [4]. Once all surfaces are compatible, continuity is enforced along the common boundary curves of two surfaces by averaging rows (or columns) of control points along the common curve. Continuity is enforced at a common corner vertex of multiple surfaces by averaging control points "around" this vertex. Currently, each B-spline surface must be represented by at least six rows and six columns of control points.

The figures 4 and 5 demonstrate the surface approximation of different geometries. A single B-spline surface approximating parts of multiple parametric surfaces of a fighter aircraft is represented in figure 4. Multiple approximating surfaces on the space shuttle geometry are shown in figure 5. Original surfaces are shaded darker than the approximating surfaces.

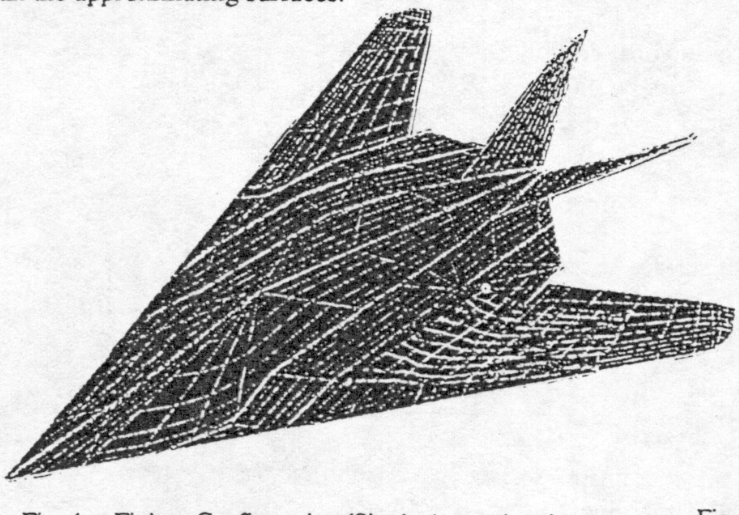

Fig. 4. Fighter Configuration (Single Approximating B-Spline Surface)

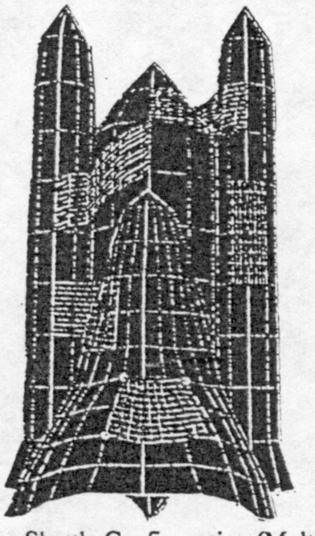

Fig. 5. Space Shuttle Configuration (Multiple Approximating B-Spline Surfaces)

SURFACE GRID REDISTRIBUTION AND REMAPPING:

Let $\underline{r} = (x_1(s,t),\ x_2(s,t),\ x_3(s,t))$ denote a parametric surface with Euclidean coordinates (x_1, x_2, x_3) and parameter values (s,t). The parametric space associated with a NURBS surface is transformed as the distribution (normalized arc length based) mesh [5]. The redistributed surface grid is obtained by evaluating the B-spline surface at the respective parameter associated with the desired distribution space.

Now, consider a sculptured surface (figure 5a). The surface grid is mapped as an O-grid on the surface. However, if the user is interested in mapping a A-type grid on the surface then a re-mapping process is accomplished by essentially creating a distribution space which, when evaluated, results in a surface grid of figure 5d. The associated parametric (distribution) spaces are presented in figures 5b-c. The creation of the distribution space for remapping is intuitive and highly application-dependent. Remapping of the surface grid is also required when the interior object(s) is (are) to be kept fixed as part of the interior surface grid. For example, consider the sculptured surface and the associated parametric space presented in figures 6a-b. Interior objects on the surface are presented in dark. The remapping process is needed to blend interior objects as a part of the surface grid. An interpolation search algorithm based on the NURBS presentation is developed to evaluate parameters associated with the interior object. The interpolation/search algorithm utilizes derivatives $\underline{r}_s, \underline{r}_t, r_{ss}, \underline{r}_{st}$, and $\underline{r}_{tt}$ and Taylor's expansion to inversely evaluate parameters. An automatic algorithm based on the weighted transfinite interpolation [5,7] in two dimensions is developed which blends the parameters associated with interior object into an overall distribution space. The resulting re-parametrized distribution space is demonstrated in figure 7a. The surface grid is then evaluated with respect to this new distribution space. The resulting, re-mapped surface grid is presented in figure 7b.

ELLIPTIC REFINEMENT:

Consider a three-dimensional elliptic grid generation system [8,9]

$$\sum_{i=1}^{3}\sum_{j=1}^{3} g^{ij}\underline{r}_{\xi^i\xi^j} + \sum_{k=1}^{3}\phi_k \underline{r}_{\xi k} = 0 \ ,$$

where

$$g^{il} = \frac{1}{g}\ (g_{jm}g_{km} - g_{jn}g_{kn})\ , \qquad (5)$$

$$i = 1,2,3; j = 1,2,3\ ,$$

and $\underline{r} = (x_1, x_2, x_3)$ and $\underline{r} = (\xi^1, \xi^2, \xi^3)$ denote the physical and computational space, respectively.

The control functions $\phi_k, k = 1,2,3$, for an orthogonal grid [6] can be formulated as

$$\hat{\phi}_k = \frac{1}{2}\ \frac{d}{d\xi^k}\ (\ln \frac{g_{kk}}{g_{ii}g_{jj}})\ ,\ \ (i,j,k)\ cyclic\ ,$$

and

$$\phi_k = -\frac{g\hat{\phi}_k}{g_{ii}g_{jj}} \qquad , g_{ii} \neq \text{ for } i = 1,2,3.$$

A quasi two-dimensional elliptic system for surface generation can be formulated [9] as

$$g_{22}(r_{\xi\eta} - \phi_1 r_\xi) - 2g_{12}r_{\xi\eta} + g_{11}(r_{\eta\eta} - \phi_2 r_\eta) = k|r_\xi x r_\eta|(r_\xi x r_\eta). \qquad (6)$$

$$where\ k = \frac{g_{22}r_{\xi\xi}\cdot(r_\xi x r_\eta) - 2g_{12}r_{\xi\eta}\cdot(r_\xi x r_\eta) + g_{11}r_{\eta\eta}\cdot(r_\xi x r_\eta)}{|r_\xi x r_\eta|^3}. \qquad (7)$$

Here, the surface is represented as a coordinate surface $\xi^3 =$ constant ($\xi = \xi^1$ and $\eta = \xi^2$), ξ^1 - and ξ^2 -lines are assumed to be perpendicular to ξ^3 lines ($g_{13} = g_{23} = 0$), and the principal curvatures of

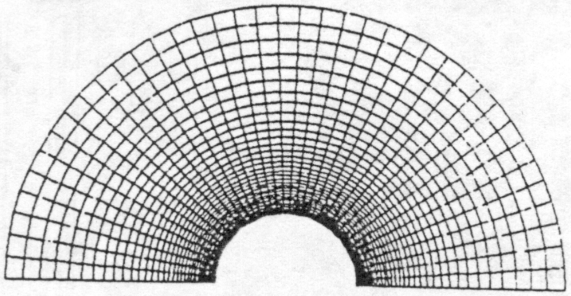

Fig. 5a. Initial O-type grid

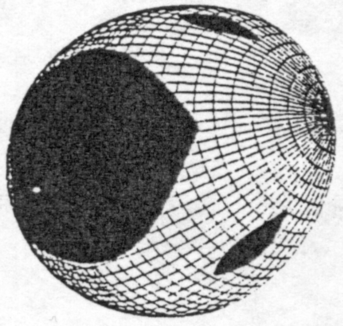

Fig. 6a. Sculptured surface with interior objects

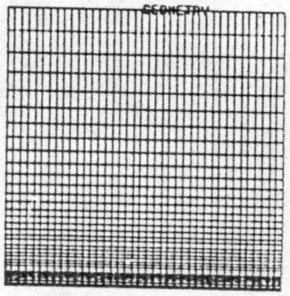

Fig. 5b. Parametric space associated with initial O-type grid

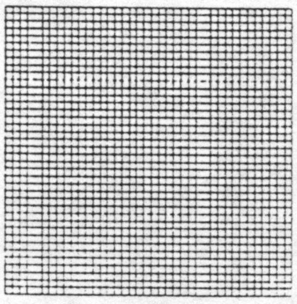

Fig. 6b. Parametric space associated with NURBS representation of the sculptured surface

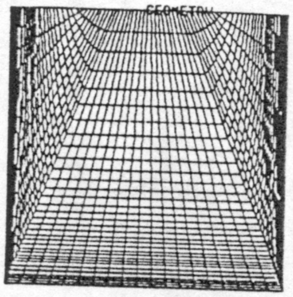

Fig. 5c. Re-mapped parametric space for the desired H-type grid

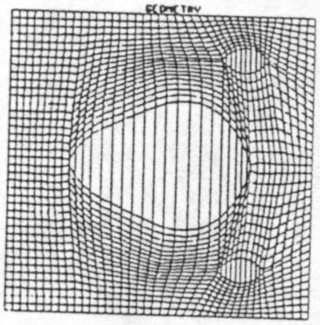

Fig. 7a. Re-parametrized distribution space

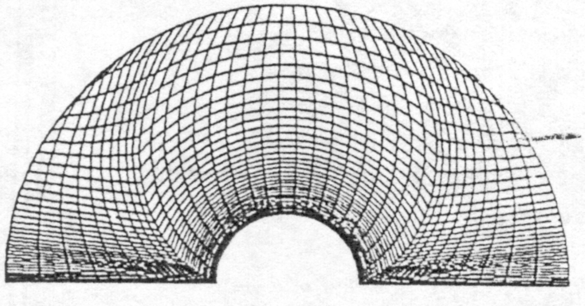

Fig. 5d. Resulting H-type grid

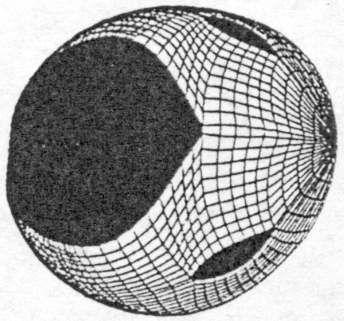

Fig. 7b. Resulting surface grid

the ξ^3-lines are assumed to be zero on the surface. The elliptic system of equations (6) – (7) can be transformed to the parametric space as follows:

$$g_{22}(s_{\xi\xi} - \phi_1 s_\xi) - 2g_{12}s_{\xi\eta} + g_{11}(s_{\eta\eta} - \phi_2 s_\eta) + \frac{J_2 A \cdot (\alpha^1 r_s - \beta^1 r_t)}{|r_s x\ r_t|^2} = 0\ , \tag{8}$$

$$g_{22}(t_{\xi\xi} - \phi_1 t_\xi) - 2g_{12}t_{\xi\eta} + g_{11}(t_{\eta\eta} - \phi_2 t_\eta) + \frac{J_2 A \cdot (\gamma^1 r_t - \beta^1 r_s)}{|r_s x\ r_t|^2} = 0\ , \tag{9}$$

where $J_2 = s_\xi t_\eta - s_\eta t_\xi$,

$$A = \alpha^1 r_{ss} - 2\beta^1 r_{st} + \gamma^1 r_{tt}\ ,$$

$$\beta^1 = x_s x_t + y_s y_t + z_s z_t\ ,$$

$$\gamma^1 = x_s^2 + y_s^2 + z_s^2, \text{ and}$$

$$|r_s x r_t|^2 = (x_s y_t - y_s x_t)^2 + (x_s z_5 - z_s x_t)^2 + (y_s z_t - z_s y_t)^2.$$

The elliptic surface grid refinement is accomplished by solving (8)-(9) with proper forcing functions for the distribution mesh (s,t) and then evaluating the associated B-spline surface at refined parameter values.

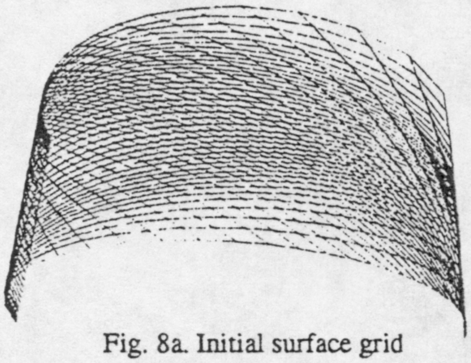

Fig. 8a. Initial surface grid

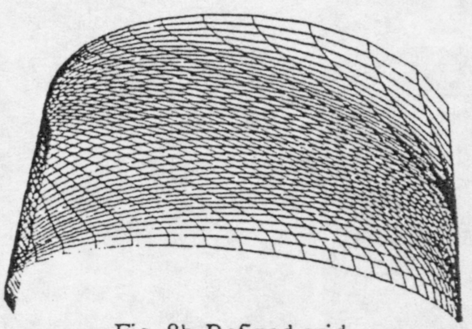

Fig. 8b. Refined grid

An example demonstrating the influence of the refinement algorithm is presented in Figure 8a.b.

REFERENCES

1. Bartels, R.H., Beatty, J.C., and Barsky, B.A. (1987), *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*, Morgan Kaufmann Publishers, Inc.
2. Farin, G. (1992), *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 3rd ed.
3. Franke, R. (1982), "Scattered Data Interpolation: Tests of Some Methods," *Math. Comp.*, Vol. 38, 181-200.
4. Piegl, L.A. (1991), "Rational B-spline Curves and Surfaces for CAD and Graphics," in: Rogers, D.F., and Earnshaw, R.A., eds., *State of the Art in Computer Graphics*, Springer-Verlag, 225-269.
5. Soni, B., (1992) "Grid Generation for Internal Flow Configurations," *Journal of Computers & Mathematics with Applications*, Vol. 25, No. 5/6, September 1992.
6. Soni, B.K., (1991) "Grid Optimization: A Mixed Approach," *Proceedings of the 3rd International Conference of Numerical Grid Generation in Computational Fluid Dynamics*, Barcelona, Spain, June, edited by A.S. Arcilla, J. Hauser, P.R. Eiseman and J.F. Thompson; North Holland.
7. Soni, B.K., (1985) "Two and Three Dimensional Grid Generation for Internal Flow Applications of Computational Fluid Dynamics," AIAA-1526-85.
8. Thompson, J.F., "A General Three-Dimensional Elliptic Grid Generation System on A Composite Block Structure," *Computer Methods in Applied Mechanics and Engineering*, p. 377, 1987.
9. Thompson, J.F., Warsi, Z.U.A. and Mastin, C.W., *Numerical Grid Generation: Foundations and Applications*, North Holland Publisher.