INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 17, 163–176 (1993)

SURFACE GRID GENERATION WITH A LINKAGE TO GEOMETRIC GENERATION

MASAHIRO SUZUKI†

Institute of Computational Fluid Dynamics, 1-16-5 Haramachi, Meguro-ku, Tokyo 152, Japan

SUMMARY

This paper describes a new method to generate surface grids over complex configurations defined by a geometric generation system. The scheme is designed for direct utilization of the surface definition provided by a geometric modeller based on a boundary representation (the so-called B-rep modeller). Thus, the conversion of the geometric representation for the surface grid generator is not required. Consequently, this technique eliminates not only laborious tedium in the conversion of data, but also errors in the representation of the surface induced in the process of the conversion.

The proposed method is accomplished over several stages. First, the triangulation is performed on the surface of the geometry, on which the area to be grided is laid. Then linear partial differential equations are mapped and solved on these triangular elements. Finally, the surface grid is constructed by searching for the contours inside the solution domain. After the co-ordinate values of the grid points are obtained by a linear interpolation within each triangular element, these values are mapped onto the surface of the geometry through surface parametric functions provided by the B-rep modeller.

An example of generating surface grid over a car configuration is given to illustrate the capability of the method.

KEY WORDS Surface grid Grid generation Geometric generation

1. INTRODUCTION

It has been well recognized that setting grids on the surface is the most time-consuming task in the grid generation process and it is a bottle-neck of the Computational Fluid Dynamics (CFD) analysis. This is attributed to a gap between geometric generation and surface grid generation. To promote the CFD analysis to be one of useful tools in the engineering routine, it is essential to link surface grid generation to geometric generation. The purpose of this paper is to present an approach to close the gap between them.

The geometric data for complex configurations, which are often used in engineering applications, are customarily produced by use of a Computer Aided Design (CAD) system. The representation of three-dimensional geometries in CAD system uses concepts from two popular forms of solid representations: (1) Constructive Solid Geometry (CSG) and (2) Boundary representation (B-rep). CSG represents the geometry by combining (through unions, differences and intersections) many copies of a few basic primitive solids (blocks, cylinders, cones and spheres). In B-rep systems, the geometry is described by a set of patches. Each patch can be defined by a parametric function and, thus, arbitrary curved surfaces can be designed by making a choice of the function, e.g. Coon's patch and Bezier patch. Therefore, aerodynamical designs, e.g. airplanes

† Present address: Railway Technical Research Institute 2-8-38 Hikari-cho, Kokubunji-shi, Tokyo 185, JAPAN

0271–2091/93/140163–14$12.00

Received June 1992
Revised March 1993

and automobiles, are usually design by B-rep environment. The transformation from CSG form to B-rep form is possible by a post-processing operation, but generally the inverse in not. Taking account of the above situation, we consider B-rep system as the geometric generator. The geometric generator is assumed to supply the parameters of definition of the patches in this paper. The scope of this study is to present a model for generating grids over the surface defined by B-rep modller, not to link a surface grid generator to a specific CAD system.

At this time, the existing techniques generally treat generating the surface grid as a two-dimensional boundary value problem on a curved surface, which is specified by a parametric function.[1] The curved surface is not necessarily represented by a patch. Though the surface can be described by a summation of the quadrangular patches, the surface has to be defined by a global parametric function. The generation of the surface grid is first accomplised in the parametric co-ordinates $\mathbf{u}(u, v)$ by interpolations or partial differential equations' solution. Then, the grid is mapped on the Cartesian co-ordinates $\mathbf{r}(x, y, z)$ by the parametric function $\mathbf{r} = \mathbf{f}(\mathbf{u})$. This procedure requires that the parametric function $\mathbf{r} = \mathbf{f}(\mathbf{u})$, which specifies the surface of the domain to be grided, is given as the input data. In most of the cases, this requirement is very stringent. The area to be grided usually spreads over whole/part of plural original patches, which are made within the geometric generator. The area is seldom identical with any single original patch. On some patches, the area covers a part of the patch. This is because the geometric generation is, on the whole, conducted outside the CFD department, with other criteria than those of grid generation for CFD. In engineering applications, the number of original patches is much larger than that of blocks used in multiblock grids approach for CFD analysis. For instance, designers demand hundreds of patches to describe a car configuration, while at most dozens of blocks are employed in numerical simulation of the flow around an automobile.[2] Therefore the original surface definition, made within the geometric generator, is not directly available for the required parametric function as the input data of the surface grid generation. To obtain the required parametric function, a data processing with some forms of interpolation is executed. This is a cumbersome task and, in addition, it is important to note that are no guarantees that the new surface, which is defined by the obtained parametric function, fits to the original surface. After generating surface grid by the conventional method, it is possible to conform the produced surface grid to the original surface by projections.[3] However, it requires an iterative calculation for every grid point and thus it costs much. There must exit a description of the patches related to the surface grid, too.

The above difficulty comes from a fact which the surface grid generator does not accept the surface definition of the geometric generator. Therefore, it would be solved if a way could be found in which the surface definition of the geometric generator could be directly used for surface grid generation. Suzuki[4] proposed a scheme to convert unstructured surface grids to structured surface grids. In the method of Suzuki, a pair of partial differential equations is first mapped and solved on the unstructured surface grid using the finite element technique. Then the curvilinear co-ordinates lines are drawn by searching for the contours inside the solution domain, i.e. the unstructured surface grid, to produce the structured surface grid. The intersections of each co-ordinate line are acquired by an interpolation within each element of the unstructured surface grid. Suzuki did not mention how to obtain the unstructured grid. In this study, an algorithm for generating unstructured surface grids is introduced with the technique of co-ordinate transformation. In the method described by Suzuki, the unstructured grid can contain several types of elements, e.g. C^1-*continuous*/C^2-*continuous* triangular/quadrangular element. However, it is generally recognized that triangular elements can best adapt the boundaries of the domain, and very often only triangular elements are able to fill those domains with very irregular boundaries and openings. Therefore, in the present work, a triangulation technique is employed to construct the

unstructured grid on the original surface. By accomplishing the information of surface definition within each element during the triangulation, the intersections of each co-ordinate line are obtained by the information of surface definition. Thus, it is achieved that surface grid points are always laid not on the unstructured grid, but on the original surface.

2. THE METHOD

The present method consists of three procedures.

2.1. Unstructured grid generation

On the original surface defined by B-rep, triangular elements are constructed. In flow simulations using the finite element method, the directional refinement of elements to the flow solution is an essential ingredient. Consequently, the interior nodes are distributed according to the prescribed element size, element stretching and stretching direction during the triangulation process.[5] In the proposed method, the unstructured grid is, however, used only for constructing the surface grid. Therefore, no control of mesh refinement is conducted. We distribute the interior nodes before the triangulation. The distributing nodes is done in an automatic manner.

The adopted triangulation is based on the advancing front technique developed by Lo.[6] In the method of Lo, the global region is allowed to be divided into as many irregular subregions. Once common boundaries are defined, no connectivity information between subregions is needed. The algorithm first generates additional interior node points according to the average nodal spacing of all the boundary segments of each subregion. All the nodes are then connected to form triangular elements in such a way that no elements overlap. A triangular element mesh is generated from one subregion to another and the entire region is finally covered.

As mentioned before, the area to be grided spreads over plural patches. In this paper, each patch, on which the area to be grided is laid, is regarded as subregion. Since some patches are covered the whole of their regions by the area, the subregions are identical to the patches (the region A in Figure 1). On the while, some patches are covered part of their regions by the area, thus the subregions are bounded by the edge of the patches and the boundary of the area (the

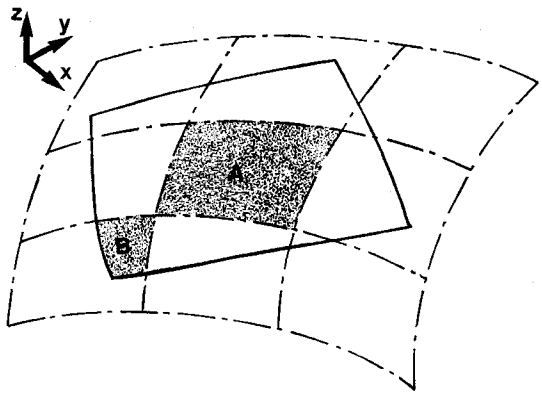

Figure 1. Subregions and original patches. Dashed lines show the edge of original patches. Bold lines show the boundary of the area to be grided. Dark region A shows a subregion, which is identical to a patch. Dark region B shows a subregion, which is bounded by the edge of a patch and the edge of the area to be grided

region B in Figure 1). Each subregion is mapped onto the parametric plane. Then interior nodes are generated and triangular elements are formed in the parametric plane. Triangular element meshes are generated in subregion by subregion. Finally the unstructured grid, which covers the whole area to be grided, is obtained.

The method proposed by Lo is changed to reduce the computational time. The modification lies in the criterion to construct the best triangulation obtained from the system of the interior nodes and the boundary nodes. Let C be a candidate opposite vertex for base $\overrightarrow{AB}$ to form a 'good-shape' triangle ABC. The consideration of the minimum value of the norm $[|\overrightarrow{AC}|^2+|\overrightarrow{BC}|^2]$ proposed by Lo is only sufficient to determine the point C for the even distribution of the interior nodes. For regions having very irregular boundaries, the resulted triangle may contain another points. Therefore, Lo's program consists of two steps to prevent the possible break down of the trianglulation. First, two nodes are chosen such that $\overrightarrow{AB}\times\overrightarrow{AC}$ is positive and which is closest to $\overrightarrow{AB}$, i.e. the minimum value of the norm $[|\overrightarrow{AC}|^2+|\overrightarrow{BC}|^2]$. Then the best node is selected by estimating the shape of the proposed triangle and its neighbouring triangles. This procedure requires much CPU time. We adopt a simple criterion:

Node C is selected if the angle ACB is a maximum. This criterion is the same as that suggested by Frederick *et al.*[7] Since they programmed to determine a node C such that the cosine value is smallest, i.e. the value of $(|\overrightarrow{BC}|^2+|\overrightarrow{AC}|^2-|\overrightarrow{AB}|^2)/(|\overrightarrow{AC}||\overrightarrow{BC}|)$ is least, it is only sufficient for angles in the range 0–180° and it may select a node with angle in the range 180–360°, i.e. a node situated in the right of $\overrightarrow{AB}$. Since C must be located in the left of $\overrightarrow{AB}$ in the advancing front technique, the above condition is not suitable. We adopt the following approach[8] (Figure 2). Make a parallel translation of co-ordinates so that the centre of $\overrightarrow{AB}$ lies at the origin:

$$u'_C=u_C-\frac{(u_A+u_B)}{2}, \qquad v'_C=v_C-\frac{(v_A+v_B)}{2}, \tag{1}$$

where (u_A, v_A), (u_B, v_B) and (u_C, v_C) are the co-ordinate values of A and B, respectively. Then make a rotation so that $\overrightarrow{AB}$ lies along the u-axis:

$$u''_C=cu'_C+sv'_C, \qquad v''_C=-su'_C+cv'_C, \tag{2}$$

where $c=(u_A-u_B)/r$, $s=(v_A-v_B)/r$ and $r=\sqrt{[(u_A-u_B)^2+(v_A-v_B)^2]}$. By this co-ordinates transformation, the nodes located in the right of $\overrightarrow{AB}$ are moved into the positive side of v-axis, and easily excluded. Calculate v-co-ordinate of the circumecentre of triangle ACB:

$$h=\frac{u''^2_C+v''^2_C-r^2/4}{2v''_C}. \tag{3}$$

Node C is selected if h is a minimum. Suppose O is the circumcentre of triangle ABC. The angle AOB is twice as large as the angle ACB. Since the smallest h has the largest angle AOB, the above condition is equivalent to the adopted criterion.

To transform the parametric co-ordinate values into the Cartesian co-ordinate values and to conduct the finite element analysis, the following data structure is employed:

(i) Each element has its number, the number of the original patch, in which the element is contained, and the node numbers at each corner of the element.
(ii) Each node has its parametric co-ordinate values.

To search for the contour lines quickly, the network information is constructed: Each element has three neighbouring element numbers abutted on the each side of the element. To start

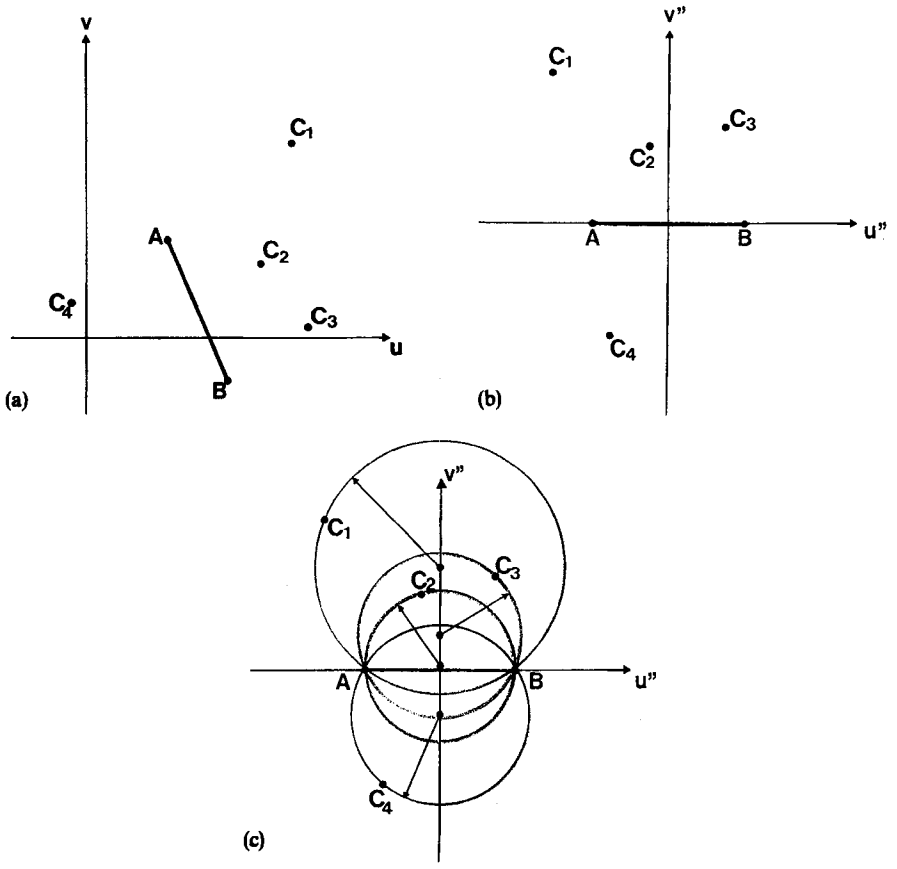

Figure 2. Selection of an opposite vertex C for a base $\overrightarrow{AB}$: (a) candidate vertexes C_i before the co-ordinate transformation; (b) candidate vertexes C_i after the co-ordiante transformation; (c) circumcircles of triangle AC_iB

searching for the contour lines, the numbers of the element, which are located on the line to be $\eta=\eta_0$ edge boundary of the surface grid, are also stored.

The following is the description of the generation procedure.

(a) The Cartesian co-ordinate values of the boundary points on the four edges of the surface grid are specified (Figure 3(a)). The number and location of these points are not necessarily coincident with those of the surface grid points.

(b) Points are distributed on the common boundaries between the original patches, which are contained within the area to be grided (Figure 3(b)). The co-ordinate values are specified in the Cartesian co-ordinates.

(c) On each original patch, these Cartesian co-ordinate values are converted into parametric co-ordinate values by the modified Newton's method[9] (Figure 3(c)). Each of these patches is considered as a subregion, respectively.

(d) The interior nodes are distributed within subregion in the parametric plane (Figure 3(d)). First, points are set equidistantly within the whole area of the original patch. If the subregion is identical to the original patch, all of these points are adopted as the interior

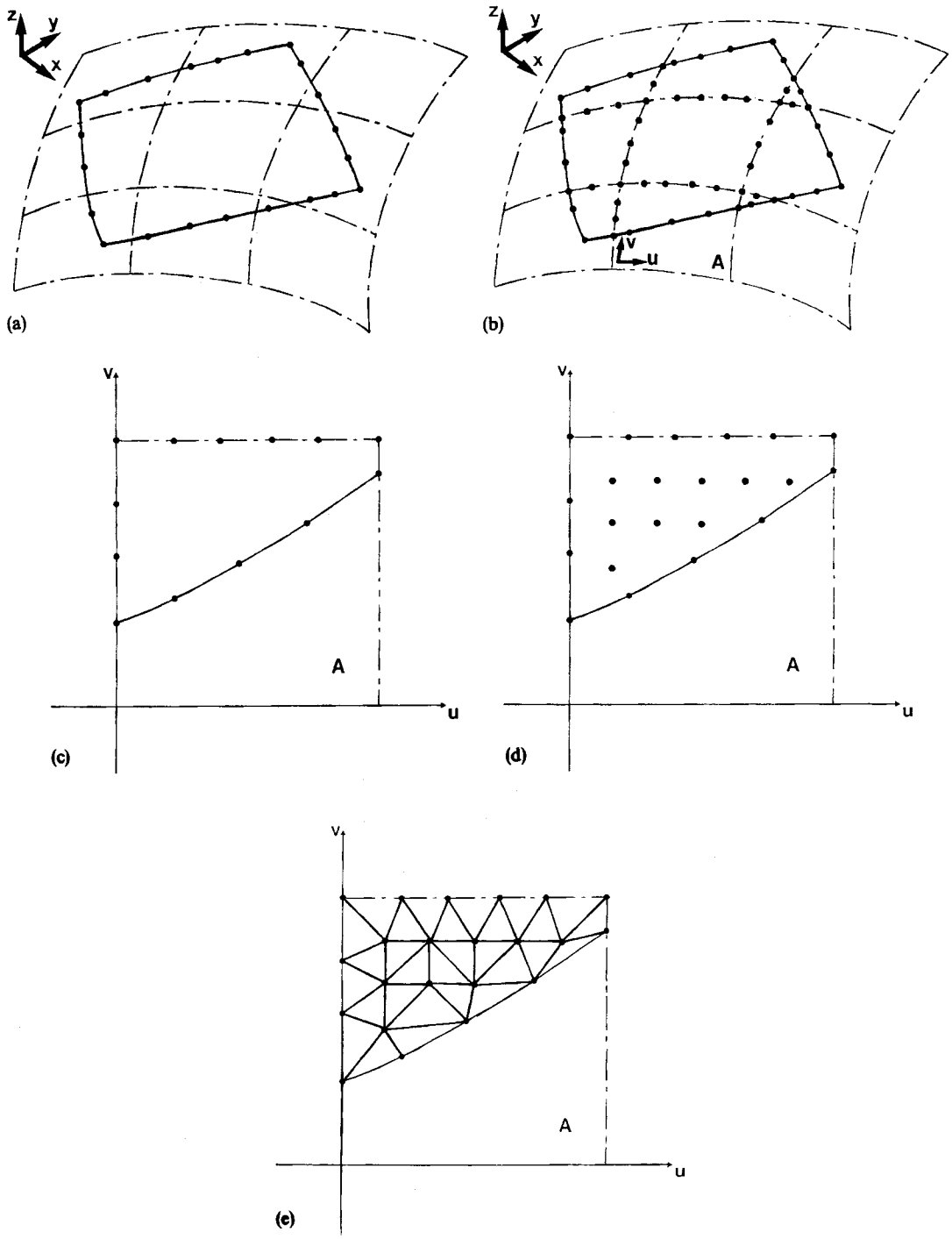

Figure 3. Construction of unstructured grid. Dashed lines show the original patches. Bold lines show the boundary of the area to be grided: (a) points on the boundary of the area to be grided are specified; (b) points are distributed on the common boundaries between the subregions; (c) Cartesian co-ordinate values are converted into parametric values; (d) interior nodes are set; (e) triangulation is performed

nodes. If the boundaries of the area to be grided cut cross the original patch, points outside the subregion and points too close to the boundaries are excluded.[10]*

(e) Triangular element meshes are generated subregion by subregion in the parametric plane until the whole global domain is covered (Figure 3(e))

2.2. *Finite element solution procedure*

According to the methodology of Suzuki,[4] following partial differential equations are mapped and solved by a finite element method on the unstructured grid:

$$\nabla(\lambda\nabla\xi)=0, \tag{4}$$

$$\nabla(\lambda\nabla\eta)=0, \tag{5}$$

where λ is weight function. The Galerkin's weighted residual approach is employed. The linear interpolation function of three-node triangular element is adopted to avoid the numerical integration and to get a high computational efficiency. The co-ordinate values of nodes are converted from the parametric co-ordinates (u, v), which is set in Section 2.1, into the Cartesian co-ordinates (x, y, z) by using the parametric form of each original patch. Then these values are transformed onto a two-dimensional plane (X, Y) (Figure 4),

$$\begin{aligned}(X_1, Y_1)&=(0, 0),\\ (X_2, Y_2)&=(l_3, 0),\\ (X_3, Y_3)&=(l_2\cos\theta_1, \sqrt{(l_2^2-X_3^2)}),\end{aligned} \tag{6}$$

where

$$\cos\theta_1=\frac{l_2^2+l_3^2-l_1^2}{2l_2l_3},$$

$$l_1=\sqrt{[(x_2-x_3)^2+(y_2-y_3)^2+(z_2-z_3)^2]},$$

$$l_2=\sqrt{[(x_3-x_1)^2+(y_3-y_1)^2+(z_3-z_1)^2]},$$

$$l_3=\sqrt{[(x_1-x_2)^2+(y_1-y_2)^2+(z_1-z_2)^2]}.$$

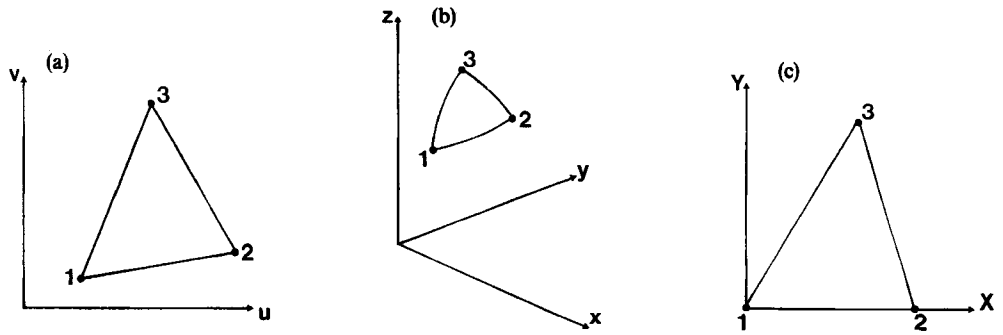

Figure 4. Mapping of a triangular element: (a) parametric co-ordinates (u, v); (b) Cartesian co-ordinates (x, y, z); (c) two-dimensional plane (X, Y).

* This routine is described in the appendix.

Here the subscripts of x, y, z, X and Y indicate the node number of the element. The integrations are done by using these values. In contrast to the well-known elliptic grid generation equations solved in the computational domain, these equations are solved in the physical domain. This avoids the need to solve non-linear equations. The incomplete Cholesky decomposition conjugate gradient (ICCG) method[9] is used to solve the linear system of equations. In conjugate with solving the linear equations, the ICCG method drastically reduces the computational time in comparison with the common direct method. The solution contours of each equation define the curvilinear co-ordinate lines of ξ and η, respectively. Therefore, the boundary conditions specify the topology of the surface grid. $\xi=1$ and $\xi=i_{\max}$ for equation (4), and $\eta=1$ and $\eta=j_{\max}$ for equation (5) are set at two sides of the area to be grided as the Dirichlet boundary (Figure 5). On the others boundary, Dirichlet or Neumann boundary condition can be set. The control of the grid space can be implemented by use of the weight functions λ. As λ increases, the gradient of the solution decreases, i.e. the grid spacing becomes large.

2.3. *Surface grid construction*

To obtain the surface grid, the curvilinear co-ordinate lines are constructed by searching for the contours inside the solution domain. It does not take much computational time to search for the contours; the network information of the unstructured grid and a hierarchical data structure is executed in advance (Section 2.1). The parametric co-ordinate values of the grid point are first obtained by a linear interpolation function within the triangular element, then they are converted onto the Cartesian co-ordinates by using the information of the original surface definition. This guarantees that the resulted surface grid points always fit to the original surface. The searching for the contours is proceeded as follows:

(a) Find an element, in which a line of $\xi=i$ crosses, among the elements, which are located on the $\eta=1$ edge boundary and are listed up in advance (Section 2.1).
(b) Check whether a line of $\eta=j$ crosses the element. If not, go to (g).
(c) Obtain the co-ordinate value of an intersection of $\xi=i$ and $\eta=j$. As the linear interpolation function of three-node triangular elements is adopted, the values of ξ and η within the element are as follows:

$$\xi=\alpha_1+\alpha_2 u+\alpha_3 v, \qquad \eta=\beta_1+\beta_2 u+\beta_3 v, \tag{7}$$

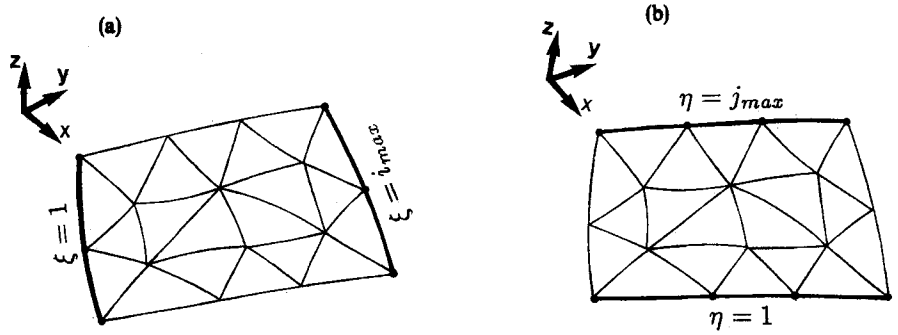

Figure 5. Boundary conditions: (a) $\xi=1$ and $\xi=i_{\max}$ are set at two side of the area to be grided for equation (4); (b) $\eta=1$ and $\eta=j_{\max}$ are set at the other two side of the area to be grided for equation (5)

where α and β are constant and specified by

$$\begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix}=\begin{pmatrix}1 & u_1 & v_1\\ 1 & u_2 & v_2\\ 1 & u_3 & v_3\end{pmatrix}^{-1}\begin{pmatrix}\xi_1\\ \xi_2\\ \xi_3\end{pmatrix},$$

$$\begin{pmatrix}\beta_1\\ \beta_2\\ \beta_3\end{pmatrix}=\begin{pmatrix}1 & u_1 & v_1\\ 1 & u_2 & v_2\\ 1 & u_3 & v_3\end{pmatrix}^{-1}\begin{pmatrix}\eta_1\\ \eta_2\\ \eta_3\end{pmatrix}.$$

Here subscripts of u, v, ξ and η indicate the node number of the element. Therefore, the co-ordinate value of an intersection of $\xi=i$ and $\eta=j$ are obtained by solving the linear system of equations (7).

$$\begin{pmatrix}u_{i,j}\\ v_{i,j}\end{pmatrix}=\begin{pmatrix}\alpha_2 & \alpha_3\\ \beta_2 & \beta_3\end{pmatrix}^{-1}\begin{pmatrix}\xi_i-\alpha_1\\ \eta_j-\beta_1\end{pmatrix}. \tag{8}$$

(d) Check whether the point $(u_{i,j}, v_{i,j})$ is located inside the element. If not, go to (g).
(e) Converting $(u_{i,j}, v_{i,j})$ into the Cartesian co-ordinate values $(x_{i,j}, y_{i,j}, z_{i,j})$ though using the parametric form of the original patch, which contains the element. Then register the co-ordinate values as those of the surface grid point. This procedure guarantees the surface grid point always fits to the original surface.
(f) $j=j+1$. Then go to (b).
(g) Find the next element, in which the line of $\xi=i$ crosses, through utilizing the information of the neighbouring element number (Section 2.1). Then go to (b).

The above procedures are performed recursively for searching all i, j points.

3. EXAMPLES

Though the presented method can deal with any kinds of parametric surfaces defined by B-rep environment, the bilinearly blended Coons patches are treated here as an example. The bilinearly blended Coons patch can be expressed not only by four corner points but also by four edge lines. The parametric form of the Coons patch is expressed by[11]

$$\mathbf{f}(u,v)=(1-u\ u)\begin{pmatrix}\mathbf{f}(0,\ v)\\ \mathbf{f}(1,\ v)\end{pmatrix}+(\mathbf{f}(u,\ 0)\,\mathbf{f}(u,\ 1))\begin{pmatrix}1-v\\ v\end{pmatrix}$$
$$-(1-u\ u)\begin{pmatrix}\mathbf{f}(0,\ 0) & \mathbf{f}(0,\ 1)\\ \mathbf{f}(1,\ 0) & \mathbf{f}(1,\ 1)\end{pmatrix}\begin{pmatrix}1-v\\ v\end{pmatrix}, \tag{9}$$

where $\mathbf{f}(u, 0)$, $\mathbf{f}(u, 1)$, $\mathbf{f}(0, v)$ and $\mathbf{f}(1, v)$ are the parametric functions, which specifies the four edge lines.

The first example is generating grid on the surface defined by two patches. Each of the edge lines of the surfaces is given by the following functions.

For surface no. 1:

$$\mathbf{f}_1(u,0)=\begin{pmatrix}u-1\\ -1\\ (\cos(-1)-1)(1-u)\end{pmatrix},$$

$$\mathbf{f}_1(u,1)=\begin{pmatrix} u-1 \\ 1 \\ (\cos 1-1)(1-u) \end{pmatrix},$$

$$\mathbf{f}_1(0,v)=\begin{pmatrix} -1 \\ 2v-1 \\ \cos(2v-1)-1 \end{pmatrix},$$

$$\mathbf{f}_1(1,v)=\begin{pmatrix} 0 \\ 2v-1 \\ 0 \end{pmatrix},$$

where $0\leq u\leq 1$ and $0\leq v\leq 1$.

For surface no. 2:

$$\mathbf{f}_2(u,0)=\begin{pmatrix} u-1 \\ -1 \\ 0 \end{pmatrix},$$

$$\mathbf{f}_2(u,1)=\begin{pmatrix} u-1 \\ 1 \\ \sin 1(1-u) \end{pmatrix},$$

$$\mathbf{f}_2(0,v)=\begin{pmatrix} 0 \\ 2v-1 \\ 0 \end{pmatrix},$$

$$\mathbf{f}_2(1,v)=\begin{pmatrix} 1 \\ 2v-1 \\ \sin v \end{pmatrix},$$

where $0\leq u\leq 1$ and $0\leq v\leq 1$.

The surface of 10×10 is shown in Figure 6. First, part of both Coons patches are triangulated in Figure 6(a) and 6(b). The equations (4) and (5) are solved on the unstructured grid and then surface grid is constructed (Figure 6(c) and 6(d)). The points of the surface grid precisely fits to the Coons patches.

In the bilinear blended Coons patch, an approximating or interpolating curve can be used as the edge line. Here, B-spline curves are employed as the second example. In the B-spline curves representation, the equation $\mathrm{BS}_m(s)$, of the curve between input points $\mathrm{V}_0, \mathrm{V}_1, \cdots, \mathrm{V}_n$ is written in vector forms,[11]

$$\mathrm{BS}_m(s)=\mathrm{BS}_m(\mathrm{V}_0,\mathrm{V}_1,\ldots,\mathrm{V}_n;\, s)=\sum_{k=0}^{n} N_{k,m}(s)\mathrm{V}_k, \tag{10}$$

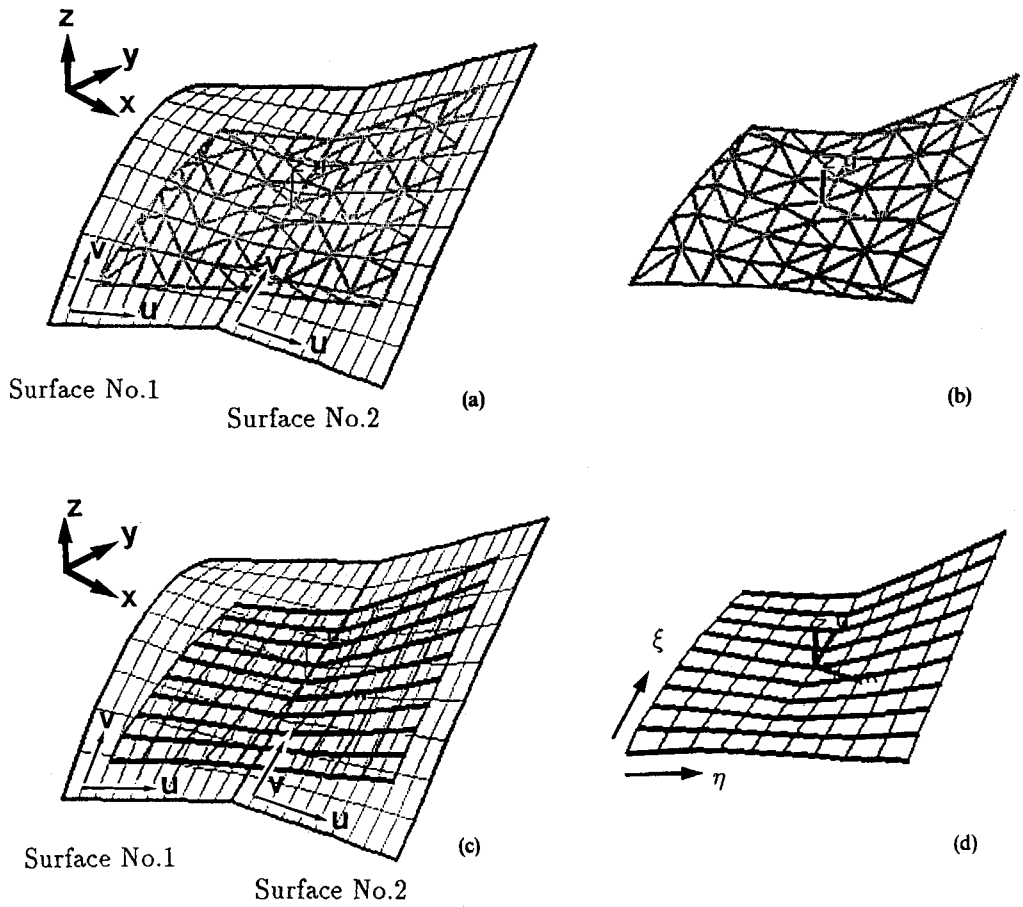

Figure 6. Surface grid generation on two bilinear blended Coons patches: (a) unstructured grid on the Coons patches; (b) unstructured grid; (c) surface grid on the Coons patches; (d) surface grid (10 × 10 grid points)

where the s co-ordinate is defined in a line segment. $N_{k,m}$ is the B-spline basis of degree m, and is defined recursively,

$$N_{k,1}(s)=\begin{cases}1 & \text{if } x_k \le s < x_{k+1},\\ 0 & \text{else,}\end{cases}$$

$$N_{k,m}(s)=\frac{(s-x_k)}{(x_{k+m-1}-x_k)}N_{k,m-1}(s)+\frac{(x_{k+m}-s)}{(x_{k+m}-x_{k+1})}N_{k+1,m-1}(s); \quad m>1, \tag{11}$$

here x is the knot and $x_0 \le x_1 \le \cdots \le x_{m+n}$.

Figure 7 shows generating the surface grid of 80×40 on a car configuration. The wire frame rending of the original patches is presented in Figure 7(a). The body consists of 14 patches. The edge lines of the patches are described by B-spline curves of third degree ($m=3$) using 84 input points. The surface grid on the upper half of the body, which is a part of a single block H–H-type grid, is generated. This type of grid has been used in the analysis of flow around an automobile.[12] The surface grid spreads over 12 patches: six patches are completely, covered by the surface grid,

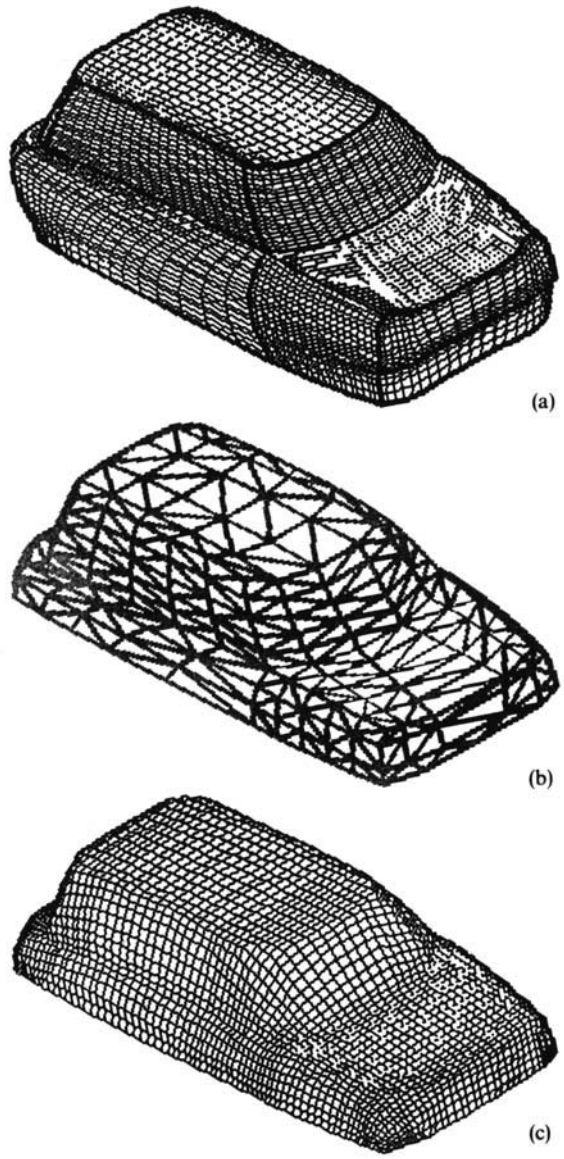

Figure 7. Surface grid on the upper half of car configuration: (a) wire frame rendering of the Coons patches; (b) unstructured grid; (c) surface grid (80×40 grid points)

and the other six are only partially covered. It is difficult and cumbersome to describe the surface by a surface function; consequently, it is very hard to obtain a surface grid without any errors by the conventional technique. The present method, which can deal with the plural patches directly, treates this problem easily. First, part/whole of patches, on which the area to be grided is laid, are triangulated in Figure 7(b). Equations (4) and (5) are solved on the unstructured grid and then surface grid is constructed (Figure 7(c)). The weight function λ is set to be related to the distance from the edge of the patches. Smooth distributed surface grid is obtained. Points precisely lay

over the Coons patches. Once the points on the edge boundary lines of the surface grid are specified, the whole computation is done within a few seconds on a workstation (Personal IRIS 4D/35).

4. CONCLUDING REMARKS

A method linking the surface grid generation to the geometric is proposed. The surface definition of the geometric generator is directly used. There is no need of converting the surface definition of the geometric generator and it results in not only eliminating the error of the geometry representation but also in reducing the human labour. Since the adopted simple but proficient method takes less CPU time it may be suitable for the interactive environment. The model presented in this paper may be efficacious when one tries to connect the existing CAD system with the CFD analysis system.

APPENDIX: ALGORITHM FOR EXCLUDING POINTS OUTSIDE THE REGION[10]

Let V be a node on the boundary of the region. The counterclockwise order is assigned to the nodes on the exterior boundary, and the clockwise order is assigned to the nodes on the inner boundary. G is a point to be checked. First, the shortest distance l_i between G and each bondary segment $\overrightarrow{V_iV_{i+1}}$ is calculated. Draw a line a perpendicular to $\overrightarrow{V_iV_{i+1}}$ and let the intersection be X_i. X_i is represented by

$$X_i = t\overrightarrow{V_iV_{i+1}}, \tag{12}$$

where

$$t = \frac{\overrightarrow{V_iG} \cdot \overrightarrow{V_iV_{i+1}}}{|\overrightarrow{V_iV_{i+1}}|}.$$

The shortest distance between G and boundary segment $\overrightarrow{V_iV_{i+1}}$ is

$$l_i = \begin{cases} |\overrightarrow{V_iG}| & \text{for } t \leq 0, \\ |\overrightarrow{X_iG}| & \text{for } 0 < t < 1, \\ |\overrightarrow{V_{i+1}G}| & \text{for } 1 \leq t. \end{cases} \tag{13}$$

Then find a segment such that l_i is minimum. In case of $0 < t < 1$ for this segment, G is located outside the region if $\overrightarrow{V_iV_{i+1}} \times \overrightarrow{V_iG} < 0$. When $t \leq 0$ or $1 \leq t$, the distinction is depended on whether the V_i is a vertex of the convex or of the concave; if $\overrightarrow{V_{i-1}V_i} \times \overrightarrow{V_iV_{i+1}} > 0$, the point is out of the region. By introducing the parameter ε, the point too close to the boundary is excluded by checking that $l_i < \varepsilon$.

REFERENCES

1. J. F. Thompson, 'Some current trends in numerical grid generation', in K. W. Morton and M. J. Baines (eds), *Numerical Methods for Fluid Dynamics III*, Clarendon Press, Oxford, 1988, pp. 87–100.
2. R. Himeno, M. Takagi, K. Fujitani and H. Tanaka, 'Numerical analysis of the airflow around automobiles using multi-block structured grids', *SAE Paper 90-0319*, 1990.
3. J. Miguel de la Viuda, J. Diet and G. Ranoux, 'Patch-independent structured multiblock grids for CFD computations', in A. S.-Arcilla, J. Häuser, P. R. Eiseman and J. F. Thompson (eds), *Numerical Grid Generation in Computational Fluid Dynamics and Related Fields*, Elsevier Science Publishers B. V., Netherlands, 1991, pp. 703–715.

4. M. Suzuki, 'Surface grid generation based on unstructured grid', *AIAA J.*, **29,** 2262–2264 (1991).
5. J. Peraire, M. Vahdati, K. Morgan and O. C. Zienkiewicz, 'Adaptive remeshing for compresible flow computations', *J. Comput. Phys.*, **72,** 449–446 (1987).
6. S. H. Lo, 'A new mesh generation scheme for arbitrary planar domains', *Int. j. numer. methods eng.*, **21,** 1403–1426 (1985).
7. C. O. Frederick, Y. C. Wong and F. W. Edge, 'Two-dimensional automatic mesh generation for structural analysis, *Int. j. numer. methods eng.*, **2,** 133–144 (1970).
8. Y. Takasawa, 'C and contour line', *Bit*, **19,** 148–157 (1987) (in Japanese).
9. T. Watanabe, M. Natori and T. Oguni, *Numerical Analysis Programs by Fortran 77*, Maruzen, Tokyo, 1989 (in Japanese).
10. S. Nagashima, 'Drawing for CG', *Pixcel*, **71,** 173–177 (1988) (in Japanese).
11. G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*, 2nd edn, Academic press, San Diego, CA, 1990.
12. K. Kawaguchi, M. Hashiguchi, R. Yamasaki and K. Kuwahara, 'Computational study of the aerodynamic behavior of a three-dimensional car configuration', *SAE Paper 89-0598*, 1989.