Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part 1 : construction of underlying geometrical model

Lee, Chi King

2003

Lee, C. K. (2003). Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part 1: Construction of underlying geometrical model. International journal for numerical methods in engineering, 56(11), 1593-1614.

<https://hdl.handle.net/10356/103368>

<https://doi.org/10.1002/nme.630>

© 2003 John Wiley & Sons, Ltd. This is the author created version of a work that has been peer reviewed and accepted for publication by International Journal for Numerical Methods in Engineering, John Wiley & Sons, Ltd. It incorporates referee's comments but changes resulting from the publishing process, such as copyediting, structural formatting, may not be reflected in this document. The published version is available at:
[DOI:<http://dx.doi.org/10.1002/nme.630>].

Downloaded on 26 Apr 2023 22:00:48 SGT

Automatic metric 3D surface mesh generation using subdivision surface geometrical model

Part I: Construction of underlying geometrical model

*C. K. Lee

*School of Civil and Environmental Engineering
Nanyang Technological University
Nanyang Avenue
Singapore 639798*

*E-mail address: ccklee@ntu.edu.sg

(Pre-printed version of the paper in IJNME Vo. 56, 1593-1614, 2003)

Summary

This paper proposes a new automatic mesh generation algorithm for 3D surface mesh generation. The algorithm is based on the metric specification approach and can generate anisotropic meshes on 3D surfaces. It is based on a new geometrical model using the interpolating subdivision surface concept. By using the subdivision surface concept, the new mesh generator can generate finite element meshes to model a wide range of surfaces which may contain sharp features such as cusp and crease lines. When comparing with other algorithms which use analytical surface patches as the underlying geometrical model, the new mesh generation scheme can be used in applications such as large deformation or crack analyses in which the domains to be gridded are not well defined or involve changing boundary. The presentation of the work is divided into two parts. In Part I, i.e. the present paper, a detailed description of the underlying geometrical model used will be given while in Part II, attentions will be focused on the mesh generation algorithms and the performance of the mesh generator.

KEYWORDS: *Metric 3D anisotropic adaptive surface mesh generations; Geometrical model; Interpolating subdivision surfaces; Sharp features on 3D surface meshes; Cusp points and crease lines*

1. Introduction

Since surface mesh generation is the fundamental element for carrying out shell analyses and 3D mesh generation, intensive research efforts had been spent on the development of a robust surface mesh generation algorithm for either triangular or quadrilateral mesh generation [1-7]. Among these mesh generation algorithms developed, nearly all of them employ some analytical functions for the definition of the underlying geometrical model. In particular, as most engineering data is originated from some CAD-CAM packages, the use of analytical surface patches such as non-uniformly rational B-spline (NURBS) surface patches are very popular. The use of analytical surface patches provides a natural parameterization of the surface geometrical model for the mesh generation process. Note that the existence of such a parameterization plays a critical role in many mesh generation algorithms which heavily depended on the parametric mapping for the control of element shape and size and other anisotropic properties [4,5,8,9]. While using analytical surfaces can provide a neat mathematical description of the geometry of the surface, it is not optimal in many applications when the boundaries and the domains to be gridded are not well defined. Some of the situations where analytical parameterizations are not optimal include

(1) *Manipulation of complex analytical functions defined by superposition or convolution.*

One commonly encountered situation of this kind is the large deformation analysis of shell structures. Initially, the shell surfaces may be presented by analytical surface patches. However, during the analysis, the shell surface will undergo finite deformation which is implicitly described by the incremental displacements at the nodal points of the mesh. Thus, it is rather difficult to employ analytical surface patches to describe the deformed configurations since discontinuous features such as *cusp points* and *crease lines* may be created during the deformation of the surface.

(2) *Numerical modelling of discrete geometric input data*

In many engineering applications such as land surveying, astronomical measurements and many medical problems, the main input data is obtained from field measurements and no continuous description of any form will be available. One characteristic of such applications is that while the resolution of the input data can be improved by carrying out finer field measurements or using higher resolution sampling equipment, the data obtain will still remain discrete and no intrinsic continuous form will be available.

(3) *Remeshing with changing surface or domain boundary topology.*

In many engineering applications, remeshing is often required to obtain a more accurate and optimal solution. For example, in a crack prorogation analysis, remeshing is carried

out around the growing crack tip in which the domain boundary topology is changing constantly.

(4) *Intersection or other Boolean operations among surface patches.*

When it is required to carry out surface intersection or other Boolean operations of surfaces, it will be far more easy to develop algorithms for the discretized surfaces rather than their corresponding analytical forms [10,11].

When compared with the use of analytical surfaces approach, relatively few alternative approaches had been suggested for mesh generations for the above applications. Lohner [12] suggested a method for regriding surface triangulation and also developed methods to recover cusp and crease features. Fery et al. [13] developed an adaptive finite element surface remeshing algorithm in which Walton's approach [14] to recreate G^1 patches from the boundary curves of the problem domain was employed. In addition, Kobbelt [15] suggested an iterative mesh generation algorithm based on surface subdivision technique. Recently, Rassineux et al. [16] developed a procedure based on the hermite diffuse interpolation in which the surface geometry is defined in a patch-by-patch manner while least squares fitting procedures are employed for the computations of surface tangents and normal vectors.

The objective of current study is to suggest a new mesh generation procedure for the discretization of 3D surfaces without using analytical surface patches. The theory and implementation details of the mesh generator will be reported in two parts. In the Part I of the study, attentions will be focused on the theoretical works for the setting up the geometrical model of the mesh generator. In addition, the properties of the underlying geometrical model employed will be described. In Part II of the study [17], concentration will be focused on the implementation details of the new mesh generation scheme using the suggested geometrical model. Furthermore, mesh generation examples will be given in Part II to demonstrate the performance and robustness of the new mesh generation scheme.

In the next section, a summary of the basic requirements needed when a geometrical model is used in conjunction with a parametric surface mesh generator will be given. It will then be followed by a detailed description of the interpolation subdivision algorithm. A brief outline for the computations of some essential geometrical properties of the interpolation subdivision model will then be provided. These procedures are the fundamental operations required to control the element size and shape properties during adaptive mesh generation. After this, the procedures used for adding various sharp surface features to the underlying geometrical model will be presented. Finally, a number of examples will be given to demonstrate the

flexibility of the geometrical model when constructing surfaces with different geometrical properties.

2. Basic requirements for the geometrical model

In any parametric mesh generator the geometry of the surface should be expressed as

$$(x,y,z)^T = \mathbf{r}(u,v) \quad (1)$$

where (u,v) are the parametric coordinates of a point on the surface and $\mathbf{r}$ is the *mapping* function of the parameterization. Note that $\mathbf{r}$ can either be expressed explicitly (e.g. in many analytical surface patches) or implicitly (e.g. via various superpositions, deformations or subdivision rules). With the surface definition given by Eqn. 1, the basic requirements for the geometrical model can be summarized below.

(1) Uniqueness of mapping

During surface mesh generation, the only requirement for 3D coordinates computation using Eqn. 1 is that, for a given parametric coordinates (u,v) , the corresponding 3D coordinates in the physical space should be uniquely defined. However, in practice, this requirement is often relaxed in such away that the mapping defined by Eqn. 1 may not be unique at a finite number of singular points [4].

(2) Existence and continuity of first fundamental form

The well-known *first fundamental form* of Eqn. 1 is given by

$$\mathbf{dx} = (dx, dy, dz)^T = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} = (\mathbf{r}_u \ \mathbf{r}_v) \mathbf{du} \quad (2)$$

The tangents $\mathbf{r}_u$ and $\mathbf{r}_v$ defined in Eqn. 2 are essential for the description of length scale transformation due to the mapping effect. In particular, if the metric specifications [4,8,9] are employed for the computation of the surface metric tensor for element shape and size controls, it is required that $\mathbf{r}_u$ and $\mathbf{r}_v$ should be bounded and continuous. However, as in most surface mesh generation algorithms, these two vectors will only be evaluated at a finite number of points (in spacing compatible with the element size requirement) in the problem domains, the continuous requirement of Eqn. 2 can again be relaxed and excluded at a small number of points in the problem domain. Note that such a relaxation is essential for the present study as one can expect that in many applications mentions in

Section 1, the problem domain will contain cusp points and crease lines at which the surface tangents are not continuous or even not defined.

(3) *Curvatures and second derivatives*

In 3D surface mesh generation, the *fidelity* (i.e. the good approximation of the surface by the mesh) of the surface mesh is also an important concern. Towards this end, it is found that [18,19] the maximum element size needed to obtain a good approximation of the target surface will depend on the principal curvatures and directions at the point under consideration. However, in order to obtain the principal curvatures and the principal directions, it is required to evaluate the second order derivatives $\mathbf{r}_{uu}$, $\mathbf{r}_{vu}$ and $\mathbf{r}_{vv}$ of the surface. Thus, if one would like to carry out curvature element-size control, it is required that the parametric mapping in Eqn. 1 must be at least C^2 continuous. Again, in practice, it will be too stringent to impose such a requirement at every point on the target surface. In fact, it will be shown later in this study that good curvature element-size control can be achieved even the higher derivatives $\mathbf{r}_{uu}$, $\mathbf{r}_{vu}$ and $\mathbf{r}_{vv}$ are defined in a discontinuous manner over the target surface.

3. Subdivision surfaces in geometrical modelling

3.1 *Overviews of subdivision surfaces*

In this study the *subdivision surface* is selected as the geometrical tool for the definition of the underlying geometrical model for mesh generation. Subdivision surface is a popular modelling primitive in computer graphics and has been used extensively in the computer graphic community for computer picture/image generation and computer animation [20-26]. It has only recently used by some researchers in the area of computational mechanics for the solution of thin-shell problems [27] and simple finite element mesh generation [15]. The paradigm for subdivision surface is, in fact, quite simple. By starting with an initial (coarse) mesh, a sequence of subdivision meshes will be constructed in which new nodes are inserted according to some simple rules. In most cases, the positions of the new nodes are computed by using some *local, affine combinations* of neighbouring nodes. With carefully chosen rules to define the positions of the inserted points and how to split the elements of the mesh, the resulting subdivision mesh will be gradually smoothed so that the angles between adjacent elements will be flattened. Eventually, after an infinite number of refinements, a smooth surface in differential geometric sense can be obtained (Fig. 1).

The advantages of subdivision algorithms are that the schemes are local and can provide definitions of surfaces over arbitrary topology with many interpolation/approximation

capabilities. In addition, subdivision schemes are simple to implement and various robust algorithms had also been developed for fast evaluation. When comparing with traditional modelling tools such as NURBS, subdivision schemes can model sharp features concisely. Even in theory the limiting surface will only be obtained after infinite number of iterations, the surface will be good enough for most applications after a small number of refinement steps. More importantly, due to the local property of the subdivision scheme, the quality of the surface at any point can be improved arbitrarily by applying more local refinements.

3.2 *Interpolating and approximating subdivision schemes*

The first subdivision scheme used for surface construction was proposed by Catmull and Clark more than two decades ago [20]. Since then, many variants have been proposed for various applications [20-26]. Almost all the schemes proposed can be separated into two classes according to the nature of the limiting surfaces:

(1) *Interpolating subdivision schemes (ISS)*

In this type of subdivision scheme [15,22,23,26], the nodal positions of the coarser mesh will be fixed and will *never* be changed. The positions of the inserted nodes will be computed according to the refinement rule and, again, once they are created their positions will not be altered. Hence, the nodes of the initial mesh together with the nodes inserted during all subdivisions will *interpolate* the limiting surface. An 1D example of interpolation subdivision is shown in Fig. 2a.

(2) *Approximating subdivision schemes*

In these schemes [20,21,24,25], the nodal positions of *both* the coarser mesh and the newly inserted nodes will be altered during successive subdivisions. Thus, the nodal positions of the initial mesh will not interpolate in the limiting surface. An 1D example of approximating subdivision is shown in Fig. 2b.

The geometrical properties of the limiting surfaces produced by using the above two types of subdivisions schemes have been studied thoroughly. In addition, quite of number of evaluation algorithms for rapid computation of various geometrical qualities of the limiting surfaces have also been suggested. Details of the geometrical analyses are quite lengthy and can be found in references [20,21,24-27]. In here, only a concise summary is given below.

- (1) For ISS, in general, the limiting surfaces are C^1 continuous and the surface tangent vectors can be computed at any point on the surface, except at a number of isolated points at where special subdivision templates are used to obtain sharp features like cusp points. The curvatures of the surface may not exist at certain points and is not square

integrable. However, given any subdivision level, the curvatures of the surface can be estimated locally within an element [17]. For ISS, at a given a point on the surface, neither the position nor the tangents and normal vectors can be expressed in an explicit form as functions of the initial configuration of the neighbourhood nodes and elements.

- (2) For approximating subdivision schemes, the limiting surfaces are globally C^2 continuous except at a number of isolated points where the surfaces are only C^1 . In addition, surface curvatures exist and are square integrable. In contrast to the ISS, except at some extraordinary vertices, the tangent and normal vectors can be obtained in close form as functions of the initial neighbourhood nodes and elements configuration.

Since the approximating subdivision schemes will only approximate the initial surface and may lead to significant loss in the fidelity of the surface model, they are not suitable to be used as the underlying geometrical model for surface mesh generations. Hence, in this study, the ISS will be employed as the geometrical tool for surface mesh generation. Note that since ISS will always interpolate the input data, the fidelity of the model can be maintained and *improved* via successive refinements. Furthermore, since many simple and effect recursive evaluation algorithms exist and the subdivision can be performed *adaptively*, the absence of explicit forms of tangents and normal vectors will only lead to a small amount of additional computational cost. Finally, as the curvatures of the surface can be estimated locally, the ISS will allow one to impose curvatures element-size control during mesh generation.

In this study, the ISS used is the one suggested by Zorin et al. [26] and is designed for triangular meshes. Therefore, it is assumed that the input initial mesh will only consist of triangular elements. Note that another ISS based on pure quadrilateral mesh has also been suggested [22]. However, as surface quadrilateral generation is still not a trivial task [28,29], the use of an ISS based on pure quadrilateral meshes would probably limit the range of application of the mesh generator developed and therefore, it is not used in the current study.

3.3 Interpolating subdivision scheme used

The basic operations of ISS are very simple. As shown in Fig. 3, for a given mesh, during each subdivision step, it will be uniformly refined by the following steps.

- (i) The position of the nodes in the original mesh will not be altered.
- (ii) New edge nodes will be inserted to the edges of the original mesh.
- (iii) The positions of the edge nodes inserted will be computed as a weighed sum of all its 1-neighbourhood nodes. For a given edge node E, the 1-neighbourhood nodes are the union

of the two end nodes of the edge on which the edge node located (nodes A and B in Fig. 4) and all the nodes connected to them (Fig. 4).

(iv) Every element in the original mesh will then be subdivided to four elements.

In step (iii), the weights applied for the calculations of the new positions of the inserted edge nodes will depend on the *valence* of the end nodes the edges. For a given node A in the mesh, the *valence* of the node is defined as the number of nodes it connects to and will be denoted as N_A . For example, in Fig. 4, $N_A=N_B=6$ while $N_C=4$ and $N_D=3$. A node is *regular* if its valence is equal to 6. Hence, in Fig. 4, nodes A and B are regular nodes while nodes C and D are *irregular* nodes. Note that all the new edge nodes inserted during the subdivision process will have valences equal to 6.

In order to obtain a smooth limiting surface, the weights applied for the computation of the new positions of the edges nodes must be carefully selected. In this study, a subdivision scheme based on the modified butterfly subdivision scheme suggested by Zorin et al. [26] is used. In this subdivision scheme, the weights applied will be depended on (i) whether the edge is an *interior edge* or a *boundary edge* (Fig. 5) and (ii) the valences of the two end points of the edge. An edge in the finite element mesh is called a boundary edge if it is adjacent to only one element in the mesh. Hence, for the mesh shown in Fig. 5, edges AB, CD and GF are interior edges while edge HJ is a boundary edge.

3.3.1 Subdivision scheme for interior edges

For an interior edge, if *both* the end points of the edge are of valence 6, the position of the inserted edge node will be computed by using the 8point *template* as shown in Fig. 6. The new position of the inserted point E, $\mathbf{x}_E$, is given by

$$\mathbf{x}_E = \frac{1}{2}(\mathbf{x}_1 + \mathbf{x}_2) - \frac{1}{16}(\mathbf{x}_3 + \mathbf{x}_4 + \mathbf{x}_5 + \mathbf{x}_6) + \frac{1}{8}(\mathbf{x}_7 + \mathbf{x}_8) \quad (3)$$

where in Eqn. 3, $\mathbf{x}_i$ is the coordinates of node i.

If one end point A, of the edge is irregular with valence not equal to six while the other end point B is regular, the position of the inserted node will be calculated based on an irregular template applied to the irregular end point A only (Fig. 7). In case that $k \geq 5$, $\mathbf{x}_E$ is given by

$$\mathbf{x}_E = \frac{3}{4}\mathbf{x}_A + \frac{1}{k} \sum_{i=0}^{k-1} \left(\frac{1}{4} + \cos\left(\frac{2\pi i}{k}\right) + \frac{1}{2} \cos\left(\frac{4\pi i}{k}\right) \right) \mathbf{x}_i \quad (4)$$

where $\mathbf{x}_A$ is the position of the irregular node A. $\mathbf{x}_i$, $i=0, \dots, k-1$ are the coordinates of the nodes adjacent to the irregular end point B. Note that the regular end point B is always labeled as $\mathbf{x}_0$

and the points $\mathbf{x}_i$ are labelled in an anti-clockwise manner with respect to the edge AB. For $k < 5$, (Fig. 8), the coordinates of node E are computed as

For $k=4$ (Fig. 8a)
$$\mathbf{x}_E = \frac{3}{4}\mathbf{x}_A + \frac{3}{8}\mathbf{x}_0 - \frac{1}{8}\mathbf{x}_2 \quad (5a)$$

For $k=3$ (Fig. 8b)
$$\mathbf{x}_E = \frac{3}{4}\mathbf{x}_A + \frac{5}{12}\mathbf{x}_0 - \frac{1}{12}(\mathbf{x}_1 + \mathbf{x}_2) \quad (5b)$$

In case that both end points A and B are irregular nodes, Eqns. 4 and 5 will first be employed to compute the two positions corresponding to the two irregular templates at nodes A and B. The final position of point E will then be taken as the average of these two positions.

3.3.2 Subdivision scheme for boundary edges

In this study, it is assumed that the domain under consideration is enclosed by a number of *limiting boundary curves* which at any time is approximated by a number of boundary segments. The boundary curves are separated by a number of *corner nodes* (Fig. 9). For an edge node located along the boundary, its position will be computed by applying the 4-point/3-point subdivision schemes suggested by Dyn et al. [23,30] and will only depend on the positions of its surrounding boundary nodes. Note that for any boundary segment on the limiting curve, it can either be bounded by (i) two other boundary segments (e.g. segment BD in Fig. 9) or (ii) by one boundary segment and one corner node (segment AB of Fig. 9). For every boundary segment bounded by two other boundary segments (Fig. 10a), the position of the inserted edge node E will be computed by the following 4-point scheme

$$\mathbf{x}_E = \frac{9}{16}(\mathbf{x}_2 + \mathbf{x}_3) - \frac{1}{16}(\mathbf{x}_1 + \mathbf{x}_4) \quad (6a)$$

For boundary segment bounded by one corner node C and one boundary segment (Fig. 10b), the position of the inserted node G will be computed by the following 3-point scheme

$$\mathbf{x}_G = \frac{3}{8}\mathbf{x}_C + \frac{6}{8}\mathbf{x}_2 - \frac{1}{8}\mathbf{x}_3 \quad (6b)$$

It can be proved that the resulting limiting surface obtained by using the subdivision scheme described above will be smooth (C^1) *globally* [26]. Hence, with the construction of a suitable parametric space [17], the first fundamental form of the surface can be established and this will allow one to carry out metric surface mesh generation.

3.3.3 Adaptive evaluation of subdivision scheme

In addition to the smoothness of the surface, another important characteristic of the above subdivision scheme is the locality property of its evaluation procedure. Suppose it is now required to compute certain geometrical properties (e.g. position, tangents or normal) of the limiting surface at a position corresponding to a point D within an element ABC of the initial mesh (Fig. 11a). Since the subdivision scheme involves 1-neighbourhood nodes of an edge and it is required to obtain the positions of edge nodes on the boundary of the first layer elements (e.g. node H in Fig. 11b) for subsequent subdivision steps, the data needed for the evaluation process is the sub-mesh obtained by collecting *two* layers of elements surrounding element ABC (Fig. 11b). After the first subdivision step is finished, the newly subdivided element which encloses point D (element AFG in Fig. 11b) will be found. The sub-mesh contains two layers of elements surrounding that element will then be extracted for further subdivision (Fig. 11c). Finally, the accuracy of the evaluation process can then be improved arbitrarily by repeating the above adaptive subdivision process around the target point D (Fig. 11d). For the complexity of the above process, the operational complexity to locate the element which contains the interested point is proportional to $O(NE^{1/2})$ where NE is the number of element in the mesh [31]. The subdivision process can easily be implemented in such a way that the complexity is only proportional to the number of refinement needed. Hence, the operation complexity to carry out one uniform subdivision of the whole mesh is will be equal to $O(NE^{3/2})$.

4. Modelling of sharp features

In many applications, it is essential that the geometrical model used can represent sharp (C^0) features. When analytical surface patches such as NURBS are employed as geometrical model for mesh generation, sharp features can usually be introduced by using the technique of multiple knots [32]. However, such a procedure will increase the order of the surface and will complicate the construction and evaluation process. In the case of subdivision surfaces, sharp features can be introduced by a simple tagging process in which the statuses of some nodal points of the input mesh are modified. The concept of tagging was suggested by Hoppe et al. [25] for the construction of piecewise smooth surfaces based on the *approximating subdivision scheme*. In this study, a similar tagging procedure is developed for the ISS.

4.1 Sharp features used in the geometrical model

(1) Corner

As mentioned in Section 3.3.2, a corner feature is defined by using the 3-point interpolating scheme (Eqn. 6b) along the boundary of the domain. The result of using the 3-point scheme is that the tangent vectors of the surface at the two sides of the boundary curves will not be continuous (Fig. 12).

(2) Cusp

A cusp is point feature at where the tangent vector of the surface is discontinuous at *any direction* around the point. Three types of cusp points are used in the current study. They are, namely, *interior cusp*, *boundary cusp* and *corner cusp*. An interior cusp point is a point located inside the problem domain (Fig. 13a). In order to obtain the discontinuity properties, the subdivision template around it will be modified locally. For any edge connected with the interior cusp point $\mathbf{x}_u$, independent of its valence, the position of the edge node E, $\mathbf{x}_E$ (Fig. 13a) will be computed by the *mid-point rule*

$$\mathbf{x}_E = \frac{1}{2}(\mathbf{x}_u + \mathbf{x}_i) \quad (7)$$

Similarly, boundary and corner cusps are formed by applying the same mid-point rule to all the edge nodes connected to them as shown in Figs. 13b and 13c respectively.

(3) Creases

A crease is a line feature inside the problem domain. In fact, it is equivalent to an interior boundary curve (or cracked line) which enclosing an opening with zero area. The tangent vectors will be discontinuous along the opposite sides of the crease line (Fig. 14a). In practice, a crease feature can be introduced by topologically disconnecting the elements and nodes at the opposite sides of the crease (Fig. 14b) in the underlying mesh. During the surface subdivision step, edges on opposite sides the crease line will be subdivided separately. For the edges lying on the crease line, the boundary subdivision schemes described in Section 3.3.2 will be applied to obtain a smooth crease line. In addition, if needed, boundary and corner cusps points can also be added to the crease line.

4.2 Tagging of nodes in the geometrical model

In order to include the sharp features mentioned in Section 4.1, a tagging procedure is developed for the efficient implementation of the geometrical model. Under this tagging procedure, every node in the mesh is associated with an integer status number which indicates

the nature of the limiting surface at that node. During any subdivision step, the refinement template used for a particular edge will be selected based on the statuses of its end nodes. The status numbers for various features described in Section 4.1 are listed in Table 1. From Table 1, it can be seen that the status numbers of all boundary nodes are higher than that of all interior nodes. In addition, sharp features such as corner and cusp nodes will have a higher status number. Note that no special status number is reserved for crease line. As mentioned in Section 4.1, a crease line is defined by separating the nodes and elements on the opposite sides of the crease (Figs. 15a and 15b). All the nodes on the crease line will be considered as boundary nodes and then assigned with node statuses equal to 2-5 (Fig. 15c). Finally, during the subdivision step, all the new interior edge nodes inserted will be assigned a status equal to 0 while all new boundary edge nodes inserted will be assigned a status equal to 2 (Fig. 16).

Table 1. Definition of status of nodes

Node status	Status Number
Interior smooth node	0
Interior cusp node	1
Boundary smooth node	2
Boundary cusp node	3
Corner node	4
Corner cusp node	5

5. Surface generation examples

In this section, a number of numerical examples will be given to demonstrate the use of the subdivision scheme and the tagging procedure for the construction of underlying geometrical models. Surface models containing different sharp features will be created to indicate the effects of initial mesh connectivity and tagging on the final limiting surfaces.

Example 1: Model of a hemisphere

In the first example, the target surface is a hemisphere in which no sharp feature is present. Two slightly different initial meshes with all their nodal points located on the hemisphere were used to create the geometrical models (Figs. 17a and 17b). The meshes obtained after applying four subdivision steps are shown in Figs. 17c and 17d. Note that by slightly modifying the connectivity of the elements near the base of the hemisphere, the shape and the curvature of the limiting surfaces (Figs. 17e and 17f) near the base are locally modified.

Example 2: Model of a cone

In the second example, the input mesh is a simple cone as shown in Fig. 18a. Two different models are generated in this example. In the first model, all the interior nodes in the initial mesh are smooth nodes. In the second model, the apex node of the initial mesh was tagged as

an interior cusp node. The limiting surfaces obtained are shown in Figs. 18b and 18c for the untagged and the tagged models respectively. Zoom views of the limiting surface near the apex of the cone are shown in Figs. 18d and 18e. It can be seen that the sharp apex of the cone was reproduced in the tagged model. This example demonstrated that the subdivision surface model can be used for the generation of sharp features by changing the statuses of a small number of nodes in the initial mesh.

Example 3: Modelling of crease lines

In this example, the effect of crease lines will be shown. The initial input mesh, which writes out the three letters "CoM" is shown in Fig. 19a. Again, two different models were generated. In the first model, no crease line was included in the model and all the interior nodes of the input mesh were marked as normal smooth nodes (Fig. 19b). In the second model, the nodes corresponding to the letter "CoM" were separated and crease lines were created (Fig. 19c). Zoom views near the letter "C" for the smooth and crease models are shown in Fig. 19d and Fig. 19e respectively. From these figures, the differences between a smooth surface and crease lines can be seen.

Example 4: Model without boundary

In the last example, the input mesh, which is shown in Fig. 20a, assumes the shape of a cube. In this example, the domain of the geometrical model contains no boundary segment and all the nodes are tagged as interior smooth nodes. The meshes generated after two, four and five subdivisions are shown in Figs. 20b to 20d respectively. Note that even the initial input mesh contains eight sharp corners, they are gradually smoothed out by the subdivision process.

For the speed of the subdivision process, the CPU time needed to carry out uniform subdivision for Example 1 (Mesh I) is plotted against the term $NE^{3/2}$ in Fig. 21. From Fig. 21, it can be seen that, as predicted in Section 3, the CPU time needed is proportional to $NE^{3/2}$. It is also found that the speed of subdivision is quite fast. The total CPU time needed for the generation of a subdivision mesh which contains nearly two hundred thousands elements is less than 30 seconds on a low-end PC equipped with an Intel 450MHz Pentium II CPU.

6. Conclusions

In this paper, an essential description has been given for the construction of a novel geometrical model for metric 3D surface mesh generation. The geometrical model used in this study is based on the subdivision surface concept which is originated in the area of computer graphics and animations. Detailed explanations for the efficient construction and

implementation of the modified butterfly interpolating subdivision scheme used in this study are given. In addition, the potentials and advantages of using the subdivision scheme as the geometrical tool for the construction of the underlying geometrical model for surface mesh generation are explored and discussed. Furthermore, in order to allow the geometrical model to include sharp features like cusp points and crease lines, modification procedures based on a new tagging procedure for the interpolating subdivision scheme are suggested. Finally, several surface generation examples are given to demonstrate the use of the suggested model for the creation of 3D surfaces with different shapes, curvatures properties and sharp features.

References

1. B. Hamann and P. Y. Tsai, A tessellation algorithm for the representation of trimmed NURBS surfaces with arbitrary trimming curves, *Comput. Aided Design*, **28**, 461-472 (1996)
2. Y. Zheng, R. W. Lewis and D. T. Gethin, Three dimensional unstructured mesh generation: Part 2. Surface meshes, *Comput. Methods Appl. Mech. Engrg.*, **134**, 269-284 (1996)
3. L. A. Piegl and A. M. Richard, Tessellating trimmed NURBS surfaces, *Comput. Aided. Des.*, **27**, 16-28 (1995)
4. C. K. Lee, Automatic metric advancing front triangulation over curved surfaces, *Engrg. Computations*, **17**, 48-74 (2000)
5. H. Borouchaki, P. Laug and P. L. George, Parametric surface meshing using a combined advancing-front generalized Delaunay approach, *Int. J. Numer. Methods. Engrg.*, **49**, 233-259 (2000)
6. S. J. Owen, M. L. Staten, S. A. Canann and S. Saigal, Q-Morph: An indirect approach to advancing front quad meshing, *Int. J. Numer. Methods. Engrg.*, **44**, 1317-1340 (1999)
7. D. Rypl and P. Krysl, Triangulation of 3D surfaces, *Engineering with Computers*, **13**, 87-98 (1997)
8. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part I. Algorithms, *Fin. Ele. Anal. Des.*, **25**, 61-83 (1997)
9. H. Borouchaki, P. L. George, F. Hecht, P. Laug and E. Saltel, Delaunay mesh generation governed by metric specifications. Part II. Applications, *Fin. Ele. Anal. Des.*, **25**, 85-109 (1997)
10. J. L. Graysmith and C. T. Shaw, Automated procedures for Boolean operations on finite element meshes, *Engrg. Computations*, **14**, 702-717 (1997)

11. A. A. Shostko, R. Lohner and W. C. Sandberg, Surface triangulation over intersecting geometries, *Int. J. Numer. Methods. Engrg.*, 44, 1359-1376 (1999)
12. R. Lohner, Regridding surface triangulations, *J. Comput. Physics*, **126**, 1-10, (1996)
13. P. J. Frey, H. Borouchaki, F. Hecht and E. Saltel, Adaptive Finite Element Surface Remeshing, *Proceedings of the 6th International Conference*, held at the University of Greenwich, pp.371-380, July 1998, Editors: M. Cross., B. K. Soni, J. F. Thompson, J. Hauser, P. R. Eiseman.
14. D. J. Walton and D. S. Meek, A triangular G^1 patch from boundary curves, *Comput. Aided. Design*, 28, 113-123, (1996)
15. L. Kobbelt, Iterative mesh generation for FE-computations on free form surfaces, *Engineering, Comput.*, 14, 806-820 (1997)
16. A. Rassineux, P. Villon, J. M. Savignat and O. Stab, Surface meshing by local hermite diffuse interpolation, *Int. J. Numer. Methods Engrg.*, **49**, 31-49 (2000)
17. C. K. Lee, Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part II. Mesh generation algorithm and examples. To appear.
18. R. J. Cass and S. E. Benzley, R. J. Meyers and T. D. Blacker, Generalized 3D paving: An automated quadrilateral surface mesh generation algorithm, *Int. J. Numer. Methods Engrg.*, **39**, 1475-1489 (1996)
19. C. K. Lee, On curvature element-size control in metric surface mesh generation, *Int. J. Numer. Methods. Engrg.*, **50**, 787-807 (2001)
20. E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes, *Comput. Aided Design*, 10, 350-355 (1978)
21. C. Loop, Smooth subdivision surface based on triangles. Master's Thesis, University of Utah, Department of Mathematics, 1987
22. L. Kobbelt, Interpolatory subdivision on open quadrilateral nets with arbitrary topology, *Eurographics'96* 15. 409-420 (1996)
23. N. Dyn, D. Levin and J. A. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, *Proceedings of Computer Graphic, ACM SIGGRAPH'90*, 160-169 (1990)
24. J. Stam, Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, *Proceedings of Computer Graphic, ACM SIGGRAPH'98*, 385-404 (1998)
25. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer and W. Stuetzle, Piecewise Smooth surface reconstruction, *Proceedings of Computer Graphic, ACM SIGGRAPH'94*, 295-302 (1994)

26. D. Zorin, P. Schroder and W. Sweldens, Interpolating subdivision with arbitrary topology, Proceedings of Computer Graphic, ACM SIGGRAPH'96, 189-192 (1996)
27. F. Cirak, M. Ortiz and P. Schroder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis, Int. J. Numer. Methods Engrg., **47**, 2039-2072 (2000)
28. S. J. Owen, M. L. Staten, S. A. Canann and S. Saigal, Q-Morph: An indirect approach to advancing front quad meshing, Int. J. Numer. Methods. Engrg., **44**, 1317-1340 (1999)
29. Y. K. Lee and C. K. Lee, Automatic Generation of Anisotropic Quadrilateral Meshes on Three Dimensional Surfaces using Metric Specifications, Int. J. Numer. Methods Engrg., **53**, 2673-2700 (2002)
30. N. Dyn, D. Levin and J. A. Gregory, A 4-point interpolatory subdivision scheme for curve design, Comput. Aided Design, **4**, 257-268 (1987)
31. R. Lohner, Some useful data structures for the generation of unstructured grids, Commun. Numer. Methods Engrg., **4**, 123-135 (1988)
32. L. Piegl and W. Tiller, Curve and surface constructions using rational B-splines, Comput. Aided Des., **19**, 485-498 (1987)

Fig. 1 Smooth surface generated by subdivision process: (a) input mesh; (b) mesh generated after one subdivision; (c) mesh generated after second subdivision; (d) limiting surface.

Fig. 2 Interpolating and approximating subdivisions: (a) interpolating subdivision;
(b) approximating subdivision.

Fig. 3 The subdivision operation: (a) initial mesh; (b) mesh after subdivision.

Fig. 4 1-neighbourhood of an edge node.

Fig. 5 Definition of interior and boundary edges.

Fig. 6 8-point template for interior edge.

Fig. 7 An irregular template surround node A.

(a) $k=4$

(b) $k=3$

Fig. 8 Template for the cases of $k=4$ and $k=3$.

Fig. 9 Definitions of boundary curves, boundary edges and corner vertices.

(a)

(b)

Fig. 10 Templates for boundary edge node insertion: (a) 4-point subdivision template;
(b) 3-point subdivision template for corner edge.

Fig. 11. Adaptive refinement around a point D in element ABC: (a) initial mesh;
(b) first level subdivision; (c) extraction of sub-mesh; (d) second level subdivision.

Fig. 12 A corner feature.

Fig. 13 Cusp node features: (a) interior cusp; (b) boundary cusp; (c) corner cusp.

(a)

(b)

Fig. 14 Crease line feature: (a) a crease line; (b) creation of crease line by separation of nodes and element in the underlying mesh.

Fig. 15 Definition of crease line: (a) location of crease line; (b) separation into two repeated curves enclosing zero area; (c) tagging of nodes.

Fig. 16 Tagging of inserted nodes: (a) initial statuses of nodes; (b) statuses after subdivision.

(a)

(b)

(c)

(d)

(e)

(f)

Fig. 17 Example 1, modelling of a hemisphere: (a) initial mesh I; (b) initial mesh II; (c) mesh obtained after four subdivisions, mesh I; (d) mesh obtained after four subdivisions, mesh II; (e) limiting surface, mesh I; (f) limiting surface, mesh II.

Fig. 18 Example 2, reproduction of the apex of a cone by using a cusp node: (a) initial mesh model; (b) limiting mesh obtained by not tagging the apex node; (c) limiting mesh obtained by tagging the apex node as cusp point; (d) zoom view near the apex node obtained by the untagged model; (e) zoom view near the apex node obtained by the tagged model.

Fig. 19 Example 3, effects of crease lines: (a) initial mesh; (b) limiting surface without crease details; (c) limiting surface with crease details; (d) zoom view near the smooth letter "C"; (e) zoom view near the crease letter "C".

(d)

(e)

Fig. 19 (Continued.) Example 3, effects of crease lines: (a) initial mesh; (b) limiting surface without crease details; (c) limiting surface with crease details; (d) zoom view near the smooth letter "C"; (e) zoom view near the crease letter "C".

(a)

(b)

(c)

(d)

Fig. 20 Example 4, model without boundary edge: (a) input mesh; (b) mesh after two subdivisions; (c) mesh after four subdivisions; (d) mesh after five subdivisions.

Fig. 21 CPU time used for subdivision of Example 1, Mesh I.