Automatic metric advancing front triangulation over curved surfaces

Lee, Chi King

2000

Lee, C. (2000). Automatic metric advancing front triangulation over curved surfaces. *Engineering Computations*, 17(1), 48-74.

<https://hdl.handle.net/10356/103284>

<https://doi.org/10.1108/02644400010308080>

© 2000 Emerald Group Publishing Limited. This is the author created version of a work that has been peer reviewed and accepted for publication by *Engineering Computations*, Emerald Group Publishing Limited. 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: [\[http://dx.doi.org/10.1108/02644400010308080\]](http://dx.doi.org/10.1108/02644400010308080).

Downloaded on 26 Apr 2023 15:41:47 SGT

Automatic Metric Advancing Front Triangulation Over Curved Surfaces

C. K. Lee

School of Civil and Structural Engineering
Nanyang Technological University
Nanyang Avenue, Singapore 639798
Email: ccklee@ntu.edu.sg

Pre-printed version for the paper appeared in the journal
Engineering Computations, Vol. 17, 1999, 48-74

Summary

A 3D surface mesh generation scheme is suggested for the triangulation of general bi-variate surfaces. The target surface to be meshed is represented as a union of bi-variate sub-surfaces and hence a wide range of surfaces can be modelled. Different useful features such as repeated curves, crack lines and surface branches are included in the geometrical and topological models to increase the flexibility of the mesh generation scheme. The surface metric tensor specification is employed to define and control the element characteristics in the mesh generation procedure. A robust metric triangulation kernel is used for parametric space mesh generation. The shape qualities of the sub-surface meshes generated are then improved by using some *ad hoc* mesh quality enhancement schemes before they are combined together to form the final mesh. Numerical examples indicate that high quality surface meshes with rapid varying element size and stretching characteristics can be generated within a reasonable time limit in a few mesh adaptive iterations.

Keywords: Automatic surface mesh generation, Advancing front technique, Metric triangulation, Bi-variate mapping, Parametric surfaces

1. Introduction

An adequate discretization of a surface is one of the most difficult and important prerequisites for 3D mesh generation and finite element modelling of shell structures. An automatic mesh generator that can generate high quality, well-graded surface meshes with element size compatible with the user specification will be an indispensable tool for finite element modellers to carry out general shell and 3D finite element analyses. Under intensive research in recent years, many effective surface mesh generation procedures have been suggested [1-15]. Nearly all the mesh generation schemes suggested use one of the following two approaches:

(1) Parametric mapping approach

In this approach, the surface to be meshed is represented by a bi-variate mapping [1-4,7-9,11,12,15] such that any point on the 3D surface is mapped to a parametric space. All the mesh generation procedures are carried out in the parametric space by a 2D mesh generator. The final 3D surface mesh is then obtained by re-mapping the mesh in the parametric space back to the 3D space. One advantage of this approach is that virtually any 2D mesh generator (advancing front, Delaunay, mapping etc.) can be used in the parametric space meshing step. However, in order to obtain a finite element mesh satisfying the user specification, especially for a seriously folded surface, it is necessary to carefully control the shape and size of the elements in the parametric space. In case that the element size and shape in the parametric space are not carefully controlled, highly distorted elements in the 3D space may be formed.

(2) Direct 3D generation

In direct generation methods, elements are formed directly on the 3D surfaces without referring to their parametric representations [5,10,13,14]. This approach is particularly useful when the parametric representations of the surface are not available. New nodes and elements are first generated on the tangential planes of the surfaces and then projected back onto the 3D surfaces. The advantage of this method is that the shapes of the elements generated will not be largely affected by the parametric mapping. However, as a projection procedure is required to bring back the new nodes onto the target surfaces, the meshing algorithm is more complicated and computational intensive.

The objective of this paper is to suggest a new generation procedure for the discretization of general 3D surfaces that may be seriously folded. As an extension to the previous work [9], the geometries of the surfaces to be meshed will be again described by general bi-variate mappings.

Furthermore, extensions have been made to embrace many additional geometrical features includes multi-connected domains, crack lines and branching of surfaces. A general secondary mapping approach [15] has also been implemented to deal with surfaces that contain singular point. The shape and the stretching characteristics of the elements will be controlled by the generalized metric specification approach [16-18]. A refined metric advancing front mesh generation scheme similar to the one developed recently by the author [19] will be used as the triangulation kernel in the parametric spaces. As a result, provide that the *topological* and *geometrical* models of the surfaces are correctly defined, the generator will be able to generate meshes satisfying the user defined gradation and directional stretching effects for a wide range of surfaces.

In the next section, a brief review of all the essential elements related to the metric specification and mesh quality measurements will be given. It will then be followed by the descriptions of the topological and geometrical models used in this study. A concise summary of the surface mesh generation scheme and the *ad hoc* mesh quality enhancement steps used will be given. Finally, several mesh generation examples will be provided to demonstrate the performance, robustness and effectiveness of the new mesh generation scheme.

2. The metric specification and element quality measures

In this section, a brief summary of the metric specification and the element shape quality measures that will be used in the surface mesh generation scheme is given. Details of the generalized metric approach can be found in references [16] and [17].

(1) The 3D metric tensor

In the 3D space, $\mathcal{R}^3$, the metric tensor that defines the user specification of element size characteristics can be written as a 3×3 matrix, $\mathbf{M}_{3D}$, of the form

$$\mathbf{M}_{3D} = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} = [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} [\mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3]^T \quad (1)$$

such that $\text{Det}(\mathbf{M}_{3D}) > 0$. In Eqn. 1, $(\mathbf{e}_i, \lambda_i)$, $i=1,2,3$ are the eigenpairs of $\mathbf{M}_{3D}$ such that $\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$

and $\lambda_i = \frac{1}{(h_i)^2} > 0$ where h_i is the principal element size in the $\mathbf{e}_i$ direction. The eigenpair,

$(\mathbf{e}_i, \lambda_i)$, will determinate the stretching ratio and direction of the mesh. By using the metric tensor the distance between two points $\mathbf{P}_1$ and $\mathbf{P}_2$, $\tilde{l}(\mathbf{M}_{3D}, \mathbf{P}_1 \mathbf{P}_2)$, is defined as

$$\tilde{l}(\mathbf{M}_{3D}, \mathbf{P}_1 \mathbf{P}_2) = \int_0^1 \sqrt{\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)^T \mathbf{M}_{3D}(\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)) \mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t)} dt \quad (2a)$$

where

$$\mathbf{Q}(\mathbf{P}_1, \mathbf{P}_2, t) = \mathbf{P}_1 + t(\mathbf{P}_2 - \mathbf{P}_1) \quad 0 \leq t \leq 1 \quad (2b)$$

Thus, the length of an elementary vector, $d\xi$, in the *normalized space* can be expressed as

$$\tilde{l}^2 = d\xi^T d\xi = d\mathbf{x}^T \mathbf{M}_{3D} d\mathbf{x} \quad (3)$$

where $d\mathbf{x}$ is the elementary vector defined in $\mathcal{R}^3$.

(2) The surface metric tensor

In surface mesh generation, the geometry of the surface is defined by a bi-variate mapping of the form

$$(x, y, z)^T = \mathbf{r}(u, v) \quad (4)$$

Hence, the relationship between $d\mathbf{x}$ and $d\mathbf{u}$, the elementary vector in the parametric space, is given by

$$d\mathbf{x} = (dx, dy, dz)^T = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \\ \frac{\partial z}{\partial u} & \frac{\partial z}{\partial v} \end{pmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} = (\mathbf{r}_{,u} \quad \mathbf{r}_{,v}) d\mathbf{u} \quad (5)$$

By combing Eqn. 3 with Eqn. 5, the relationship between $d\xi$ and $d\mathbf{x}$ can be written as

$$\tilde{l}^2 = d\xi^T d\xi = d\mathbf{x}^T \mathbf{M}_{3D} d\mathbf{x} = d\mathbf{u}^T (\mathbf{r}_{,u} \quad \mathbf{r}_{,v})^T \mathbf{M}_{3D} (\mathbf{r}_{,u} \quad \mathbf{r}_{,v}) d\mathbf{u} = d\mathbf{u}^T \mathbf{M}_{sur} d\mathbf{u} \quad (6)$$

Thus, the 2×2 matrix, $\mathbf{M}_{sur}$, defined as

$$\mathbf{M}_{sur} = (\mathbf{r}_{,u} \quad \mathbf{r}_{,v})^T \mathbf{M}_{3D} (\mathbf{r}_{,u} \quad \mathbf{r}_{,v}) \quad (7)$$

is the metric tensor which combines the effects of the user specification and the surface mapping.

(2) Element shape and quality measures

With the surface metric tensor and length scale measurement defined by Eqns. 6 and 7, the shape quality for a triangle $\mathbf{P}_1\mathbf{P}_2\mathbf{P}_3$ is computed as [19]

$$\tilde{\alpha} = \min(\hat{\alpha}(\mathbf{P}_1), \hat{\alpha}(\mathbf{P}_2), \hat{\alpha}(\mathbf{P}_3)) \quad (8a)$$

where

$$\hat{\alpha}(\mathbf{P}_i) = \frac{2\sqrt{3} \cdot \text{Det}(\mathbf{M}_i) \cdot \text{Det}([\mathbf{P}_2 - \mathbf{P}_1, \mathbf{P}_3 - \mathbf{P}_1])}{\tilde{l}(\mathbf{M}_i, \mathbf{P}_1\mathbf{P}_2)^2 + \tilde{l}(\mathbf{M}_i, \mathbf{P}_2\mathbf{P}_3)^2 + \tilde{l}(\mathbf{M}_i, \mathbf{P}_1\mathbf{P}_3)^2} \quad (8b)$$

In Eqn. 8b, $\mathbf{M}_i$ is the surface metric at $\mathbf{P}_i$ while $\text{Det}(\mathbf{M})$ is the determinant of $\mathbf{M}$. Note that the shape quality $\tilde{\alpha}$ given in Eqn. 8 will only depend on the shape distortion of the triangle (with respect to the metric tensor) but not the edge length deviation. An equilateral triangle of *any* size will attain a maximum value of $\tilde{\alpha}=1$. In view of this, in the element formation step during mesh generation, a *combined* shape and size parameter, $\tilde{\mu}$, will be used to assess the overall quality of the element.

$$\tilde{\mu} = \tilde{\alpha}(\delta_1\delta_2\delta_3)^{\frac{1}{3}} \quad (9a)$$

In Eqn. 9a, δ_i is the length deviation of the i th edge from unity and is defined as

$$\delta_i = \min(\tilde{l}_i, \frac{1}{\tilde{l}_i}) \quad (9b)$$

where $\tilde{l}_i$ is the length of the i th edge of the element. $\tilde{\mu}$ will account for both the shape distortion and edge length deviation of the element. For example, the $\tilde{\mu}$ value for an equilateral triangle with edge length equal to 2.0 with respect to the metric tensor will equal to 0.5.

3. Geometrical and topological descriptions of the surfaces

Geometrical model

An unambiguous geometrical definition of the domain to be treated is a prerequisite for the generation of a valid surface mesh. In this study, it is assumed that the surface to be meshed will possess the following geometrical properties.

- (1) The problem domain to be meshed can always be presented as a union of sub-surfaces with finite surface areas.
- (2) All the sub-surfaces to be meshed are solely defined by
 - (i) The *support surfaces* on which they are rested on.

In here, it is assumed that a support surface is defined by parametric mapping of the form

$$(x, y, z)^T = \mathbf{r}((u, v)^T) \quad (10)$$

where (u, v) are the parametric coordinates of the mapping. The domain of the support surface in the parametric space is an unit square $[0,1] \times [0,1]$ (Fig. 1). It is also required that the outward normal, $\hat{\mathbf{n}}$, of the support surface is continuous and can be conveniently computed at almost any point in the parametric space.

- (ii) The *trimming curves* that enclose the domains to be meshed in the parametric spaces of the support surfaces.

These trimming curves will satisfy the following conditions:

- (a) They are continuous, not self-intersecting and should not intersect with other curves.
 - (b) All exterior trimming curves are arranged in an anti-clockwise manner while all interior curves are ordered in clockwise direction. Thus, the domain to be gripped is always lying on the left-hand-side of the trimming curves (Fig. 2). Furthermore, the u - v coordinates of the trimming curves can again be expressed in parametric form as

$$(u, v)^T = \mathbf{g}(t) \quad \text{and} \quad t \in [0, 1] \quad (11)$$

- (3) Two sub-surfaces are either (i) not connecting or (ii) connecting along part of their boundaries or (iii) sharing a common corner point.
 - (4) The outward normal vectors of two sub-surfaces may or may not be continuous along their common boundaries.

(5) The topology and connectivity of the sub-surfaces and their trimming curves are well defined.

Obviously, when constructing the geometrical model for the problem domain, there exist many alternatives to describe the support surfaces (Eqn. 10) and the sub-surface trimming curves (Eqn. 11). In fact, any bi-variate surface that has continuous normal vector can be used to define the support surfaces. Similarly, any parametric curve that satisfies conditions (a) and (b) can also be used to define the trimming curves in the parametric space. In this study, non-uniform rational B-spline (NURBS) surfaces are used as the geometrical tool for the definition of the support surfaces while all the trimming curves will be modelled by NURBS curves. The NURBS curves and surfaces are selected here because such a combination is highly flexible and can be applied to model most commonly encountered geometrical features in finite element analysis [20-22].

Degenerated line

In the present geometrical model, a trimming curve is a degenerated curve when the whole curve is mapped to only one point in the 3D space (Fig. 3). In such a situation, the outward normal will either be undefined (e.g. at the pole of a sphere) or not uniquely defined (e.g. the vertex of a cone). The metric tensor $\mathbf{M}_{\text{sur}}$ in Eqn. 7 will become singular along the degenerated line and this will cause difficulties during boundary segment generation. Therefore, whenever a degenerated line is present in the support surface, a secondary mapping will be employed to further transform the domain of the support surface from a unit square to an isosceles triangle (Fig. 4). From Fig. 4, the relationship between $(u,v)^T$ and the secondary mapping coordinates $(u',v')^T$ can be written as

$$u = \frac{2u' - v'}{2(1 - v')}, \quad v = v' \quad (12)$$

In order to reflect the effect of the secondary mapping, the surface metric tensor defined in Eqn. 7 have to be modified. From Eqn. 5, the relationship between the elementary vectors in the u - v space and the u' - v' space is given by

$$\mathbf{du} = (du, dv)^T = \begin{pmatrix} \frac{\partial u}{\partial u'} & \frac{\partial u}{\partial v'} \\ \frac{\partial v}{\partial u'} & \frac{\partial v}{\partial v'} \end{pmatrix} \begin{pmatrix} du' \\ dv' \end{pmatrix} = \begin{pmatrix} 1 & 2u' - 1 \\ (1 - v') & 2(1 - v')^2 \\ 0 & 1 \end{pmatrix} \mathbf{du}' \quad (13a)$$

Hence, Eqn. 6 becomes

$$\tilde{l}^2 = \mathbf{du}^\top \mathbf{M}_{\text{sur}} \mathbf{du} = (\mathbf{du}')^\top \begin{pmatrix} \frac{1}{(1-v')} & 0 \\ \frac{2u'-1}{2(1-v')^2} & 1 \end{pmatrix} \mathbf{M}_{\text{sur}} \begin{pmatrix} \frac{1}{(1-v')} & \frac{2u'-1}{2(1-v')^2} \\ 0 & 1 \end{pmatrix} \mathbf{du}' \quad (13b)$$

and the modified surface metric tensor, $\mathbf{M}'_{\text{sur}}$, is given by

$$\mathbf{M}'_{\text{sur}} = \begin{pmatrix} \frac{1}{(1-v')} & 0 \\ \frac{2u'-1}{2(1-v')^2} & 1 \end{pmatrix} \mathbf{M}_{\text{sur}} \begin{pmatrix} \frac{1}{(1-v')} & \frac{2u'-1}{2(1-v')^2} \\ 0 & 1 \end{pmatrix} \quad (14)$$

It can be seen from Eqn. 14 that the modified metric tensor will still be undefined at the singular point (0.5,1.0). However, this will not cause any problem during mesh generation. It is because whenever the surface metric tensor at the singular point is required, its value can be replaced by the corresponding value at a point close to it. Details of this procedure can be found in the Appendix.

Crack line and Surface Branch

In the current geometrical model, a crack line can be defined by using two (or more) *straight* trimming curves enclosing a region with zero area. If the crack line is an external crack (Fig. 5a), the trimming curves will be arranged in anti-clockwise direction. While for an internal crack line (Fig. 5b), the corresponding trimming curves will be arranged in a clockwise manner. In both cases, trimming curves at the opposite sides of the crack are coincident (occupy the same position in both parametric and 3D spaces) but they will not be considered as repeated lines during boundary segment generation. They will be treated as separated lines and nodes will be generated independently. Special treatment is thus required to adjust the positions of nodes lying on these curves before the parametric mesh generation procedure is started.

Another important geometrical feature that required serious consideration is surface branch (Fig 6). In this case, depends on the topology of the surfaces involved, the geometrical model will involve two or more coincident curves (Fig. 6). For a surface branch, the coincident curves involved will be considered as repeated curves in the 3D space. Nodes will first be generated on the curve with the lowest edge number and then copied to others curves according to their orientations (direction in which the parametric coordinates of the nodes with respect to the curve

are increasing). For example, in Fig. 6, nodes will be first generated on curve 6 and then copied to curves 7 and 10. When both the original curve and the target curve are located on the same support surface (e.g. curves 6 and 10 in Fig. 6), the parametric coordinates of the nodes with respect to the target curve can be easily found. However, if the target curve is located on a different support surface (e.g. curves 6 and 7), an iterative solution scheme will be employed to determine the parametric positions of the nodes in the target curve.

Topological model

In order to generate a valid surface mesh, an unambiguous topological model that clearly defines the connectivity of all the sub-surfaces and their trimming curves is required. Such topological model is essential during both boundary segment generation and sub-surface mesh generation. In order to determine the topological relationships among different sub-surfaces and their boundary curves, the following information is needed:

- (1) A list of trimming curves for each sub-surface

As mentioned in the last section, all external trimming curves will be arranged in anti-clockwise direction while all internal trimming curves will be arranged in clockwise direction. Note that trimming curves in the parametric spaces and the boundary curves in the 3D space are one-to-one correspondence (Fig. 3).

- (2) The connectivity information among all trimming curves.

In order to ensure the compatibility of nodal positions along the common boundary curves between two (or more) sub-surfaces, the connectivity information among all trimming curves and their orientations must be defined. Such information will be used in the boundary segment generation step to determine whether new nodes are required to generate along a particular curve or their positions should be retrieved from a coincident curve.

4. The mesh generation procedure

The finite element mesh will be generated in a bottom up (point-curve-surface) sequence based on the hierarchical generation principle. Three generation stages, namely, (1) control point formation, (2) boundary segment generation and (3) sub-surface mesh generation, can be identified. In order to ensure the compatibility of meshes among different sub-surfaces, nodes and elements generated in any one stage will not be modified in subsequent stages.

Control point formation

Control points are the endpoints of the trimming curves enclosing the sub-surfaces. They are the only set of nodes with positions and numbering (in both the parametric and the 3D spaces) that are invariant with respect to the target mesh density. Control points in the parametric spaces are not one-to-one correspondence to the control points in the 3D space. In general, depends on the topology of the surface patch, the number of control points in the parametric spaces will be equal to or greater than the number of control points in the 3D space (Fig. 7). One control point in the 3D space can associate with several points in the parametric spaces if it is a common point of two sub-surfaces. In addition, a control point can associate with a singular point in one sub-surface while it is the endpoint of a normal trimming curve in another sub-surface (e.g. node 4 in Fig. 7). The numbering of control points will always be smaller than the numbering of all other nodes that are generated in subsequent mesh generation stages.

Boundary segment generation

After all control points are established, new boundary segments will be generated along the trimming curves of all sub-surfaces. It should be noted that according to the connectivity of the surface patch, a trimming curve can always be classified into one of the following three categories (Fig. 8):

- (1) Simple curves that do not coincide with any other curves.
- (2) Repeated curves that only coincide with other trimming curves which are lying on the same sub-surface.
- (3) Repeated curves that coincide with other trimming curves which are lying on the same or on a different sub-surface.

When discretizing a simple curve (category 1), the parametric values of the new nodes will be determined by considering the surface metric tensor (Eqn. 7) along that curve. For repeated curves (categories 2 and 3), the coincident curve with the lowest curve number will be discretized first. The parametric values of the nodes lying all other coincident curves will then be deduced from that curve. A trimming curve is a *free curve* if the parametric values of the nodes on it are to be determined by using the surface metric tensor. Otherwise, the curve is a *dependent curve*. Obviously, all simple curves and repeated curves with the lowest curve number among their

coincident neighbors are free curves. With these definitions, the boundary segment generation procedure can be summarized by the pseudo-coding shown in Box 1.

```

(1) Examine the topological information of all trimming curves and classify them according to their
    connectivity.
(2) LOOP over all the trimming curves
    (a) Determine all the free curves and dependent curves.
    END LOOP
(3) LOOP over all the trimming curves
    Let  $C$  be the current curve under consideration and  $S$  be the support surface on which it is lying.
    IF  $C$  is a free curve THEN
        (b) Discretize  $C$  (in the parametric space of  $S$ ) by considering the surface metric tensor along
            the curve. Compute the corresponding 3D coordinates of the all boundary nodes
            generated.
    ELSE
        (c) Identify the all the curves that have already been discretized and coincide with  $C$ . Collect
            all these curves to a set denoted as  $C$ .
        (d) Divide  $C$  into two subsets  $C_{\text{same}}$  and  $C_{\text{diff}}$  such that  $C_{\text{same}}$  contains all the curves in  $C$  that
            are lying on  $S$  and  $C_{\text{diff}} = C \setminus C_{\text{same}}$ .
        IF  $C_{\text{same}} \neq \emptyset$  THEN
            (e) Select the curve in  $C_{\text{same}}$  with the lowest curve number as the reference curve  $C_{\text{ref}}$ .
                Use the parametric coordinates of nodes on  $C_{\text{ref}}$  to determine the corresponding
                coordinates of nodes on  $C$ .
        ELSE
            (f) Select the curve in  $C_{\text{diff}}$  with the lowest curve number as the reference curve  $C_{\text{ref}}$ . Use
                the parametric coordinates of nodes on the old discretization of  $C$  to determine the
                corresponding coordinates of nodes on  $C$ .
        ENDIF
    ENDIF
END-LOOP

```

Box 1. Boundary segment generation

In step (b), boundary segments will be generated according to the metric tensor specification by the scheme described in reference [19]. The total length of the curve (in the normalized space) will first be determined. Segments with length as close to unity as possible will then be formed. The corresponding parametric values of the new nodes with respect to the boundary curves are interpolated from the old boundary segments. Finally, the 3D coordinates of the nodes are computed by using Eqn. 11 and then Eqn. 10.

In step (e), when C_{ref} and C are from the same sub-surface, the orientation of C_{ref} will be opposite to C (Fig. 8). Suppose that t_{ref} is the coordinate of a given node on C_{ref} , the corresponding

parametric value, t , of the same node on $\mathbf{C}$ can be obtained by the reversibility property of the boundary curve and is given by

$$t = 1.0 - t_{\text{ref}} \quad (15)$$

In step (f), when the reference and dependent curves are from different support surfaces, the corresponding parametric coordinates of the nodes on $\mathbf{C}$ can no longer be expressed explicitly in the simple form shown in Eqn. 15. Now assume that the definitions of the curves $\mathbf{C}_{\text{ref}}$ and $\mathbf{C}$ and their corresponding support surfaces are given by (c.f. Eqns. 4 and 11)

$$\text{For } \mathbf{C}_{\text{ref}} \quad (u, v)^T = \mathbf{g}_{\text{ref}}(t_{\text{ref}}) \quad \text{and} \quad (x, y, z)^T = \mathbf{r}_{\text{ref}}((u, v)^T) \quad (16a)$$

$$\text{For } \mathbf{C} \quad (u, v)^T = \mathbf{g}(t) \quad \text{and} \quad (x, y, z)^T = \mathbf{r}((u, v)^T) \quad (16b)$$

Then in order to compute the parametric coordinates of the nodes on $\mathbf{C}$, it is required to solve the following nonlinear equation

$$\mathbf{r}(\mathbf{g}(t)) = \mathbf{r}_{\text{ref}}(\mathbf{g}_{\text{ref}}(t_{\text{ref}})) \quad t \in [0, 1] \quad (17)$$

In general, the solution of Eqn. 17 can only be obtained numerically by a nonlinear equation solver using an iterative solution procedure. Obviously, a good initial guess close to the exact solution will be able to ensure the convergence of the solution procedure and reduces the number of iterations needed. In practice, such a guess can be obtained from the background discretization of the dependent curve $\mathbf{C}$. Let the total number of segments of $\mathbf{C}_{\text{ref}}$ (and hence of $\mathbf{C}$) is equal m . Suppose that it is now required to determine the parametric coordinate, t_i , of a node N_i on $\mathbf{C}$ and $L(N_i)$ is the total length (in the 3D space) of all the segments from the initial node, N_0 , up to N_i (Fig. 9) ($L(N_i)$ can be easily computed from the discretization of $\mathbf{C}_{\text{ref}}$). Then an initial guess for t_i can be found by considering the background discretization of $\mathbf{C}$ from the previous mesh by the following procedure.

- (1) Locate the j th segment in the background discretization of $\mathbf{C}$ (Fig. 9) such that the two endpoints of it, N'_{j-1} and N'_j satisfy the condition

$$L(N'_{j-1}) \leq L(N_i) \leq L(N'_j) \quad (18)$$

- (2) Retrieve the parametric coordinates of N'_{j-1} and N'_j with respect to $\mathbf{C}$ in the background discretization and denoted them as t'_{j-1} and t'_j respectively.

(3) Compute the initial guess, t_{guess} , for t_i by linear interpolation

$$t_{\text{guess}} = (1 - \xi)t'_{j-1} + \xi t'_j \quad \text{and} \quad \xi = \frac{L(N_i) - L(N'_{j-1})}{L(N'_{j-1}) - L(N'_j)} \quad (19)$$

The above procedure can be used to obtain an initial guess for N_i , $i=1, \dots, m-1$. At the two ends of C , one will have the trivial cases that $t_0=0$ and $t_m=1.0$. After t_{guess} is determined, it is then fed into a nonlinear equation solver as the initial guess for the solution of Eqn. 17. Eqn. 17 is considered to be solved if the solver can find a value of t satisfying the condition

$$|\mathbf{r}(\mathbf{g}(t)) - \mathbf{r}_{\text{ref}}(\mathbf{g}_{\text{ref}}(t_{\text{ref}}))| \leq \varepsilon \quad (20)$$

In the current implementation where the position of a point is stored using the DOUBLE PRECISION data type in FORTRAN, a value of $\varepsilon=1 \times 10^{-8}$ is employed.

It should be mentioned that as the definitions of the parametric curves and the support surfaces can assume any form, it is difficult to prove that the initial guess obtained from the above procedure can always lead to a converged solution. However, numerical experience indicates that when the background mesh is reasonably refined, the above solution scheme can usually find a solution satisfying Eqn. 17 within a few iterations.

A final remark about Steps (e) and (f) in Box 1 is that after the parametric coordinates of nodes on C are determined, it is not necessary to re-compute their 3D positions since they are already established during the discretization of the reference curve C_{ref} .

After the boundary segment generation procedure is completed, *two* sets of boundary discretization data will be created.

(1) 3D Boundary discretization

This refers to the boundary discretization of the problem domain in the 3D space and will include the boundary segments information and boundary nodes coordinates in the 3D space. Note that in this data set, the coordinates of nodes on repeated curves (e.g. due to surface branches) will only be stored once.

(2) Sub-surface boundary discretization

This refers to the boundary discretization of all the sub-surfaces with respect to their parametric spaces and will include all sub-surfaces boundary segments information and

coordinates of their boundary nodes. In this data set, the coordinates of nodes on repeated curves with respect to different sub-surfaces will be stored separately.

Support surface triangulation

Once all the boundary curves are discretized, sub-surfaces will be triangulated one by one using the metric advancing triangulation scheme developed recently [19]. The triangulation procedure for a sub-surface will consist of the following steps:

- (1) Retrieval of boundary segments from the boundary discretization and initialization of the generation front.
- (2) If the sub-surface contains crack lines or surface branches, adjust the positions of the nodes lying on all the coincident curves of the initial generation front.
- (3) Generate the mesh in the parametric space of based on the surface metric tensor specification.
- (4) Enhance the quality of the output mesh.
- (5) Combine the sub-surface mesh with the global mesh.

These five steps are considered in detail below.

Initialization of the generation front

The boundary segments enclosing a given sub-surface can be retrieved by considering the topological information of the sub-surface and its boundary discretization information. The trimming curves for the sub-surface to be meshed will be identified. Boundary segments and nodes on these curves will be used as the initial generation front for the parametric mesh generation procedure. External and internal segments will be arranged in anti-clockwise and clockwise directions respectively. Nodes lying on coincident curves of the sub-surface will be labelled *separately*.

Modification of initial generation front

If coincident curves (either due to crack lines or surface branches) are present in a sub-surface, the positions of the nodes on these curves will be adjusted. This is to ensure that during subsequent mesh generation steps, the mesh generator will be able to distinguish two overlapping

boundary segments (Fig. 10a). The positions of all the nodes lying on these curves will be slightly perturbed according to the following scheme.

Let $\mathbf{A}$ be a node lying on a repeated curve and $\mathbf{A}'$ be the corresponding node lying on the opposite side of the curve (Fig. 10b). Furthermore, assume that nodes $\mathbf{B}$ and $\mathbf{C}$ are the two neighboring nodes connected to $\mathbf{A}$ by the segments ISEG1 and ISEG2 respectively. From the positions of nodes $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, two unit vectors $\hat{\mathbf{n}}_1$ and $\hat{\mathbf{n}}_2$ which are normal to ISEG1 and ISEG2 respectively will be established. The perturbation direction of $\mathbf{A}$, $\hat{\mathbf{n}}$, is then taken as the direction of the vector $\hat{\mathbf{n}}_1 + \hat{\mathbf{n}}_2$. That is,

$$\hat{\mathbf{n}} = \frac{\hat{\mathbf{n}}_1 + \hat{\mathbf{n}}_2}{|\hat{\mathbf{n}}_1 + \hat{\mathbf{n}}_2|} \quad (21a)$$

While the new position of $\mathbf{A}$, denoted as $\mathbf{A}_{\text{new}}$ will be computed as

$$\mathbf{A}_{\text{new}} = \mathbf{A} + \phi \hat{\mathbf{n}} \quad (21b)$$

The perturbation factor ϕ in Eqn. 21b is given by

$$\phi = \frac{\varepsilon}{\min(\hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_1, \hat{\mathbf{n}} \cdot \hat{\mathbf{n}}_2)} \quad (22b)$$

where ε is a small constant and a value of $\varepsilon=1 \times 10^{-8}$ is employed. Note that by selecting ϕ according to Eqn. 22b, it will ensure that the shortest distances between $\mathbf{A}_{\text{new}}$ and the segments ISEG1 and ISEG2 is greater than or equal to ε . Note that from Eqn. 21a, the perturbation procedure will fail if

$$\hat{\mathbf{n}}_1 + \hat{\mathbf{n}}_2 = 0 \quad (23)$$

This condition will only occur when node $\mathbf{A}$ is the common endpoint of two coincident segments (Fig. 10c). In such a case, the position of node $\mathbf{A}$ will not be (and need not to be) perturbed. Similarly, the node $\mathbf{A}'$ at the other side of the curve will also be perturbed in a similar manner. After the above perturbation procedure all the coincident curves will be separated and the mesh generator will be able to distinguish all the boundary segments in the problem domain.

Parametric space mesh generation

The metric advancing front mesh generated developed earlier [19] is employed for the mesh generation in the parametric space. Element size and stretching characteristics will be controlled by the input surface metric tensor which is defined in the input parametric background mesh. The surface metric tensor will combine both the user node spacing specification with the effects of the parametric mapping. It is always aimed to generate unit equilateral triangles with respect to the normalized space. In the element formation step, the shortest front segment in the 3D space will be selected as the base segment. A new element will be formed by either creating a new interior node or by connecting the base segment with the best front node. New interior nodes will be established by an edge rotation scheme [19] so that the resulting elements will be as close to an equilateral as possible. Whether the interior node or the best front node is selected will depend on the $\tilde{\mu}$ quality (Eqn. 9) of the resulting element. After a new element is formed, the generation front will be updated and the generation process is terminated when no front segment is remained. Details of the element formation step can be found in reference [19] and will not be repeated here.

Mesh quality enhancement

After the mesh generation step is completed, the following mesh quality enhancement procedures will be employed to improve the shape and size qualities of the mesh.

(1) Diagonal swapping

In diagonal swapping, all diagonals shared by two triangles will be examined and swapping will be preformed if it can increase the $\tilde{\alpha}$ values of the resulting elements.

(2) Edge length smoothing

In edge length smoothing, the positions of a given node will be adjusted in such a way that the length of all the edges connected to it will be as close to unity as possible. This is done by progressively moving the node to the centroid (with respect to the metric tensor) of all the nodes connected to it [19].

(3) Element quality smoothing

In element quality smoothing, the position of a given node will be adjusted by considering the $\tilde{\alpha}$ qualities of all the triangles surrounding it. An edge rotation procedure similar to the one used in the element formation step will be used to determine all the possible new positions of

the node [19]. The centroid of these positions will then be set as the new position of the node if the product of all $\tilde{\alpha}$ values of the surrounding elements is increased.

Usually, a few cycles of diagonal swapping and smoothing procedures will be sufficient to produce a good quality mesh.

Combine the sub-surface mesh with the global mesh

After the mesh for a given sub-surface is generated and enhanced, it will be pasted back to the global mesh. The positions of the nodes generated in the 3D space will be computed. The old background mesh in the parametric space is then discarded and the newly generated parametric mesh will be set as the background mesh for the next refinement. The mesh generation procedure is finished when all the sub-surface meshes are defined.

5. Surface mesh generation examples

In this section, totally six mesh generation examples will be presented to demonstrate the performance of the surface mesh generation procedure developed. In all the examples, the element size distributions are directed by some fictitious metric tensors defined over the 3D space. The corresponding surface metric tensors are computed by using Eqns. 7 and 14. In order to imitate the situation during an adaptive analysis [23,24], the surface metric tensors used were defined implicitly over the nodes of the background meshes. In all the examples, very coarse initial meshes were used to start the mesh generation procedure. Both isotropic and anisotropic cases will be considered. Surfaces with different degrees of complexity (simply and multi-connected, with one or more than one sub-surfaces) and features (degenerated lines, crack lines and surface branches) were used. However, details about the exact definitions of the surfaces and the metric tensors used will be omitted here since they are deemed lengthy and tedious but not very informative. It is more convenient to examine the performance of the mesh generation scheme by studying the final meshes produced. Table 1 summarizes all the essential characteristics of the surfaces and the metric tensors used in the examples. The initial meshes used are shown in Figs. 11a to 16a. In each example, four refinements were carried out to generate the final meshes shown in Fig. 11b to 16b. A summary of the characteristics for the final meshes is listed in Table 2. From Table 2, it can be concluded that the qualities of all the meshes generated are good. The geometrical mean $\tilde{\alpha}$ values of all the meshes are greater than or equal to

0.9 while the minimum values are greater than 0.35. For the grading of the final meshes, it can be seen that more than 95% of all the edges generated with lengths within the range [0.7,1.3]. Hence, it can be concluded that the current mesh generation scheme can adequately capture the variations of the input surface metric tensors.

For the convergence rate of the mesh generation process, the convergence history for the Example 5 (Fig. 15) is shown in Fig. 17 which shows that both the edge length and element quality distributions of the meshes in fact converged to their final forms within two to three iterations.

As for the speed of the present mesh generation procedure as concern, since all geometrical measurements are computed using the surface metric tensor and mappings are required to compute the 3D positions of the nodes formed, the surface mesh generator will incur a greater computational effort than the corresponding 2D metric mesh generator. However, as all the generation algorithms used are similar to those used in 2D mesh generation, the operation complexities of the current scheme are equal to that of a 2D advancing front mesh generator [9] and are equal to $O(NE^{1.5})$ and $O(NE)$ for the triangulation and the mesh quality enhancement steps respectively. It is found that when running on a low-end PC (266MHz Pentium II CPU and 64MB RAM), the generator can generate 10000 elements (the background contains 5000 elements) in about 80 seconds and thus it should be fast enough for most practical applications.

Example	No. of sub-surfaces	Metric tensor	Singular point	Internal opening	Crack line	Surface Branch/ Repeated line
1 (Fig.11)	1	isotropic	No	Yes	No	No
2 (Fig. 12)	1	anisotropic	Yes	Yes	Yes	No
3 (Fig. 13)	4	isotropic	Yes	Yes	No	No
4 (Fig. 14)	6	isotropic	Yes	No	No	Yes
5 (Fig. 15)	3	anisotropic	No	Yes	No	Yes
6 (Fig. 16)	10	isotropic	No	Yes	No	Yes

Table 1. Characteristics of surfaces and metric tensors used

Example	NN	NE	ND	$\tilde{\alpha}_{\text{mean}}$	$\tilde{\alpha}_{\text{min}}$	ND _{0.7-1.3} (%)
1 (Fig. 11b)	2933	5176	8110	0.94	0.36	96
2 (Fig. 12b)	3192	6019	9211	0.93	0.41	95
3 (Fig. 13b)	10708	21274	32075	0.95	0.36	97
4 (Fig. 14b)	13293	26561	40151	0.96	0.38	97
5 (Fig. 15b)	13763	27388	42026	0.90	0.37	95
6 (Fig. 16b)	20796	41402	62630	0.95	0.39	97

Table 2. Characteristics of final meshes generated

Legends for Table 2

NN = Number of nodes in the mesh

NE = Number of elements in the mesh

ND = Number of edges in the mesh

$\tilde{\alpha}_{\text{mean}}$ = Geometrical mean $\tilde{\alpha}$ value of all triangles

$\tilde{\alpha}_{\text{min}}$ = Minimum $\tilde{\alpha}$ value of all triangles

ND_{0.7-1.3}(%) = Percentage of edges with length (with respect to the metric tensor) inside the range [0.7,1.3]

6. Conclusions and future investigations

In this paper, the metric advancing front technique has been extended from 2D mesh generation to surface mesh generation. The geometrical model used regards the target surface to be meshed as a general bi-variate mapping so that the operation complexity of the generation procedure is equal to that of a 2D advancing front mesh generation scheme. The element size distribution of the mesh is controlled by the metric tensor specification. From the numerical examples given in the last section, it is demonstrated that such an approach can generate well-graded meshes with element size distributions compatible with the input metric specifications. The mesh generation speed is fast enough for practical applications. The convergence rates of the quality and grading of the meshes are fast and two to three adaptive meshing iterations are adequate. The surface meshes generated can be ready used as the input boundary meshes for 3D anisotropic mesh generation [12,25-27].

One immediate extension of the present research is to combine the mesh generation scheme with the mesh gradation control methods recently suggested by H. Borouchaki et al [28] to control the element size variation in a mesh refinement scheme. By using the mesh gradation control methods described in reference [28] to modify the metric specification in an *a priori* manner, both the convergence rate of the mesh refinement procedure and the quality of the final adaptive mesh generated could be improved.

Another potential future research area is the generation of complete anisotropic quadrilateral meshes over general parametric surfaces. However, the task to generate a highly anisotropic quadrilateral mesh on a seriously folded surface will not be a trivial task. As extremely stretched elements will be formed in the parametric space, traditional methods that work well in isotropic case such as the systemic merging approach [29-31] and the paving technique [14,32] may not be

directly applicable in surface mesh generation. Hence, it appears that more research effort is required to develop a new method for such situations.

Appendix Evaluation of surface metric tensor at singular point

During boundary segment and sub-surface mesh generations, it is sometime required to evaluate the surface metric tensor at the singular point in the parametric space. One example is the interpolation of the metric tensor at a point $\mathbf{P}$ inside an element E while the singular point $\mathbf{P}_s$ is one of the vertexes of E (Fig. A1). In such a case, the interpolation procedure will involve two steps:

- (1) Interpolation along the line $\mathbf{P}_1\mathbf{P}_2$ to obtain the metric tensor at point $\mathbf{Q}$.
- (2) Interpolation along the line $\mathbf{P}_s\mathbf{Q}$ to obtain the metric tensor at point $\mathbf{P}$.

Obviously, in step (2) it is required to evaluate the surface metric tensor at $\mathbf{P}_s$. However, from Eqn. 14, the metric tensor at $\mathbf{P}_s$ will be undefined even after the secondary mapping. This problem can be solved by evaluating the metric tensor at a point $\mathbf{P}_r$ which is lying on the line $\mathbf{P}_s\mathbf{Q}$ and locates at a small distance ε_s from $\mathbf{P}_s$. In practice, the value of ε_s can be conveniently set as 1×10^{-8} without impairing the quality and grading of the final mesh generated.

Reference

1. D. Rypl and P. Krysl, Triangulation of 3D surfaces, *Engineering with Computers*, **13**, 87-98 (1997)
2. H. N. GURSOY and N. M. Patrikalakis, An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part I Algorithms, *Engineering with Computers*, **8**, 121-137 (1992)
3. H. N. GURSOY and N. M. Patrikalakis, An automatic coarse and fine surface mesh generation scheme based on medial axis transform: Part II Implementation, *Engineering with Computers*, **8**, 179-196 (1992)
4. M. C. Lee and M. S. Joun, General approach to automatic generation of quadrilaterals on three-dimensional surfaces, *Comm. Numer. Methods. Engrg.*, **14**, 609-620 (1998)
5. R. Lohner, Regridding surface triangulations, *J. Comput. Physics*, **126**, 1-10, 1996
6. L. Kobbelt, Iterative mesh generation for FE-computations on free form surfaces, *Engineering, Comput.*, **14**, 806-820 (1997)

7. M. Suzuki, Surface grid generation with a linkage to geometric generation, *Int. J. Numer. Methods Engrg.*, **17**, 163-176 (1993)
8. A. Khamayseh and C. W. Mastin, Computational conformal mapping for surface grid generation, *J. Comput. Physics*, **123**, 394-401 (1996)
9. C. K. Lee and R. E. Hobbs, Automatic adaptive finite element mesh generation over rational B-spline surfaces, *Comput. Struct.*, **69**, 577-608 (1998)
10. T. S. Lau and S. H. Lo, Finite element mesh generation over analytical curved surfaces, *Comput. Struct.*, **59**, 301-309 (1996)
11. J. Peraire, J. Peiro, K. Morgan and O. C. Zienkiewicz, Finite element mesh generation and adaptive procedure for CFD, *Proceedings of GAMNI/SMAI Conference on Automatic and adaptive mesh generation*, Grenoble, France, October 1-2, 1987.
12. 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)
13. T. S. Lau, S. H. Lo and C. K. Lee, Generation of quadrilateral mesh over analytical curved surfaces, *Fin. Ele. Anal. Design*, **27**, 251-272 (1997)
14. 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)
15. E. Hinton, N. V. R. Rao and M. Ozakca, Mesh generation with adaptive finite element analysis, *Adv. Engrg. Software*, **13**, 238-262 (1991)
16. 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)
17. 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)
18. H. Borouchaki and P. J. Frey, Adaptive triangular-quadrilateral mesh generation, *Int. J. Numer. Methods Engrg.*, **41**, 915-934 (1998)
19. C. K. Lee, Automatic adaptive mesh generation using metric advancing front approach, *Engineering Computations*, **16**, 230-263 (1999)
20. L. Piegl and A. M. Richard, Tessellating Trimmed NURBS surfaces, *Comput. Aided Design*, **27**, 16-28 (1995)
21. L. Piegl and W. Tiller, Curve and Surface constructions using rational B-Splines, *Comput. Aided. Design*, **19**, 485-498 (1987)

22. L. Piegl and W. Tiller, Algorithm for approximate NURBS skinning, *Comput. Aided Des.*, **28**, 699-706 (1996)
23. E. Hinton, M. Ozakca and N. V. R. Rao, Adaptive analysis of thin shells using facet elements, *Int. J. Numer. Methods Engrg.*, **32**, 1283-1301 (1991)
24. C. K. Lee and R. E. Hobbs, Automatic adaptive refinement for shell analysis using nine-node assumed strain element, *Int. J. Numer. Methods Engrg.*, **40**, 3601-3638 (1997)
25. Y. Zheng, R. W. Lewis and D. T. Gethin, Three dimensional unstructured mesh generation: Part 1. Fundamental aspects of triangulation and point creation, *Comput. Methods Appl. Mech. Engrg.*, **134**, 249-268 (1996)
26. Y. Zheng, R. W. Lewis and D. T. Gethin, Three dimensional unstructured mesh generation: Part 3. Volume meshes, *Comput. Methods Appl. Mech. Engrg.*, **134**, 285-310 (1996)
27. P. J. Frey, H. Borouchaki and P. L. George, 3D Delaunay mesh generation coupled with an advancing-front approach, *Comput. Methods Appl. Mech. Engrg.*, **157**, 115-131 (1998)
28. H. Borouchaki, R. Hecht and P. J. Frey, Mesh gradation control, *Int. J. Numer. Methods. Engrg.*, **43**, 1143-1165 (1998)
29. J. Z. Zhu, O. C. Zienkiewicz, E. Hinton and J. Wu, A new approach to the development of automatic quadrilateral mesh generation, *Int. J. Numer. Methods Engrg.*, **32**, 849-866 (1991)
30. C. K. Lee and S. H. Lo, A new scheme for the generation of a graded quadrilateral mesh, *Comput. and Struct.*, **52**, 847-857 (1994).
31. S. J. Owen, M. L. Staten, S. A. Canann and S. Saigal, Q-Morphing: An indirect approach to advancing front quad meshing, *Int. J. Numer. Methods. Engrg.*, **44**, 1317-1340 (1999)
32. T. D. Blacker and M. B. Stephenson, Paving: A new approach to automated quadrilateral mesh generation, *Int. J. Numer. Methods Engrg.*, **32**, 811-847 (1991)

Fig. 1. Parametric mapping of support surface

Fig. 2. Trimming curves enclosing the problem domain

Fig. 3. Mapping of degenerated line

Fig. 4. Secondary mapping for degenerated line

Fig. 5. Definition of cracked lines in parametric space

Fig. 6. Surface branching

Fig. 7. Control points for different sub-surfaces

Simple curves: 1, 3, 7, 8, 9

Repeated curves coincide with curves from the same sub-surface: 2, 4

Repeated curves coincide with curves from the same and different sub-surfaces: 5, 6, 10

Fig. 8. Simple and repeated curves

Fig. 9. Determination of parametric coordinates of nodes on C .

(a) Overlapping boundary segments (b) Adjustment of point A (c) Common endpoint of segments

Fig. 10. Perturbation of node A

(a) Initial Mesh

(b) Final Mesh

Fig. 11. Example 1

(Logo of the *Computational Mechanics Research Group*
School of Civil and Structural Engineering, NTU)

(a) Initial Mesh

(b) Final Mesh

Fig. 12. Example 2

(a) Initial Mesh

(b) Final Mesh

Fig. 13. Example 3

(a) Initial Mesh

(b) Final Mesh

Fig. 14. Example 4

(a) Initial Mesh

(b) Final Mesh

Fig. 15. Example 5

(a) Initial Mesh

(b) Final Mesh

Fig. 16. Example 6

(a) Convergence of element quality, $\tilde{\alpha}$

(b) Convergence of edge length

Fig. 17. Convergence history for Example 5

Fig. A1. Evaluation of surface metric tensor near singular point