A new finite point generation scheme using metric specifications

Lee, Chi King

2000

Lee, C. K. (2000). A new finite point generation scheme using metric specifications.
International Journal for Numerical Methods in Engineering, 48(10), 1423-1444.

<https://hdl.handle.net/10356/103386>

[https://doi.org/10.1002/1097-0207\(20000810\)48:10<1423](https://doi.org/10.1002/1097-0207(20000810)48:10<1423)

© 2000 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:
[[http://dx.doi.org/10.1002/1097-0207\(20000810\)48:10<1423::AID-NME948>3.0.CO;2-T](http://dx.doi.org/10.1002/1097-0207(20000810)48:10<1423::AID-NME948>3.0.CO;2-T)].

Downloaded on 26 Apr 2023 19:32:24 SGT

A New Finite Point Generation Scheme Using Metric Specifications

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
International Journal for Numerical Methods in Engineering, Vol. 48, 2000, 1423-1444

Summary

A new approach to generate finite point meshes on 2D flat surface and any bi-variate parametric surfaces is suggested. It can be used to generate boundary-conforming anisotropic point meshes with node spacing compatible with the metric specifications defined in a background point mesh. In contrast to many automatic mesh generation schemes, the advancing front concept is abandoned in the present method. A few simple basic operations including boundary offsetting, node insertion and node deletion are used instead. The point mesh generation scheme is initialized by a boundary offsetting procedure. The point mesh quality is then improved by node insertion and deletion such that optimally spaced nodes will fill up the entire problem domain. In addition to the point mesh generation scheme, a new way to define the connectivity of a point mesh is also suggested. Furthermore, based on the connectivity information, a new scheme to perform smoothing for a point mesh is proposed to improve the node spacing quality of the mesh. Timing shows that due to the simple node insertion and deletion operations, the generation speed of the new scheme is nearly ten times faster than a similar advancing front mesh generator.

Keywords: Grid generation; Meshless methods; Finite point mesh generation; Metric specification

1. Introduction

In recent years, it has become clear that, in many applications involving problem domains with complicated geometry, model (mesh) generation is far more time-consuming and expensive than the finite element solution procedure. Therefore, there is an urgent need to explore new methods that require less time and computational cost in the preparation of input data. In this respect, considerable efforts have been devoted to the development of the so-called meshless/ element free/ finite point methods [1-13]. A number of different promising approaches have been reported and it is becoming more convincing that they are superior to the traditional finite element method in many application areas of computational mechanics. When comparing with the finite element method, the most remarkable difference and advantage of the meshless method is that it does not require any finite element mesh for the construction of the approximated solution [3]. When using the meshless method for solving problems, only *a finite point mesh* with good distribution of interpolation points that can capture the variation of the solution functions is required. As for any finite point mesh, it is completely not necessary to consider the connectivity and shape distortion factor of the elements in the mesh, the process of point generation is expected to be simpler when comparing with finite element mesh generation. Similar to the finite element method which needs a good finite element mesh for accurate approximation, any meshless formulation will require a good finite point mesh to achieve accurate results for the following reasons.

- (1) The discretization of the unknown function (and its derivatives) will be solely depended on the point distribution of the point mesh.
- (2) It is expected that meshless methods will be most frequently used for the solution of problems in which element generation is too difficult. For examples, problems involving complicated or rapidly changing domain geometry, highly anisotropic solution fields with strong boundary layers and rapidly changing boundary conditions [3,7,8,12]. In such situations, in order to capture the rapid variation of the solution fields, it is often required to use highly anisotropic point meshes with good distribution of point density.

One simple way to obtain a suitable point mesh is to use an existing automatic finite element mesh generator [14-19] to first discretize the problem domain with a finite

elements and then delete all the elements from the resulting mesh. Obviously, this approach is far from ideal and is not feasible at all for problems in which reasonably good meshes are too difficult to generate. Very recently, Lohner and Onate [20] re-use the advancing front technique (AFT) [14,15,18,19], which has been widely used in finite element mesh generation, to develop an isotropic point mesh generation procedure. It is found that without the need to create an explicit mesh, the speed of the finite point mesh generator is nearly one order faster than traditional AFT mesh generators.

The objective of this paper is to develop a new finite point generation procedure with can be used to generate high quality well-graded anisotropic point meshes for trimmed 2D flat and bi-variate parametric surfaces. The surface to be meshed may be seriously folded, contains crack lines and surface branching. The point density and stretching characteristics of the mesh will be controlled by the generalized metric specifications approach [14-17]. Unlike the approach used in Reference [20], as no explicit mesh is required, it is found that an active generation front is not necessary when creating a point mesh. In order to obtain a good distribution of points near the domain boundary, a boundary offsetting procedure will be used to initialize the point mesh generation scheme. Point insertion and deletion procedures are then employed to achieve the user prescribed grading requirements. Finally, a point mesh smoothing procedure will be employed to enhance the quality of the final mesh. In this paper, in addition to the point mesh generation procedure, a method to measure the quality of a point mesh will also be suggested.

In the next section, a brief review on the geometrical model used for surface patch descriptions and the metric specifications will be given. It will then be followed by the descriptions of the point mesh generation and the point mesh enhancement procedures. The point mesh quality measurement scheme will also be introduced in that section. Finally, several generation examples will be provided to demonstrate the effectiveness of the new point mesh generation scheme.

2. Geometrical model and metric specification

Geometrical Model

As similar to the previous work for surface mesh generation [15], the domain to be meshed will be represented as a union of bi-variate surface patches and NURBS surfaces will be used to defined the support surface mappings (Fig. 1).

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

Trimming curves, repeated lines, crack lines and surface branching are allowed to present and they will all be defined as NURBS curves in the parametric spaces of the support surfaces [15]. Furthermore, when singular point is present (e.g. the pole of a hemisphere), a secondary mapping will be employed to transform the domain of the parametric space from a unit square to an isosceles triangle (Fig. 2). By using the secondary mapping, boundary segment generation will not be required along the degenerated line (line AB in Fig. 2) of the parametric space. This geometrical model using the combination of NURBS curves and surfaces is highly flexible and can model most commonly encountered geometrical features in engineering applications.

Metric Specifications

In this study, the metric specifications approach [16,17] will be used to control the node density of the point mesh. In the 3D space, the metric tensor can be conveniently expressed as a symmetric positive definite matrix, $\mathbf{M}_{3D}$, such that $\text{Det}(\mathbf{M}_{3D}) > 0$ and the eignpairs of $\mathbf{M}_{3D}$, $(\mathbf{e}_i, \lambda_i)$, $i=1,2,3$, will define the principal stretching directions and node spacing of the point mesh. By using $\mathbf{M}_{3D}$, the length scale transformation between the 3D space and the normalize space is given by

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

where in Eqn. 2, $d\boldsymbol{\xi}$ and $d\mathbf{x}$ are the elementary vector in the normalize space and the 3D space respectively.

For surface and 2D point mesh generation, by using Eqn. 1, the surface metric tensor, $\mathbf{M}$, corresponding to $\mathbf{M}_{3D}$ can be defined as

$$\mathbf{M} = \begin{pmatrix} \frac{\partial \mathbf{r}}{\partial u} & \frac{\partial \mathbf{r}}{\partial v} \end{pmatrix}^T \mathbf{M}_{3D} \begin{pmatrix} \frac{\partial \mathbf{r}}{\partial u} & \frac{\partial \mathbf{r}}{\partial v} \end{pmatrix} \quad (3)$$

The 2×2 matrix $\mathbf{M}$ will combine the effects of the user specifications of node density and the surface mapping. It can be directly used to control the node spacing during point formation in the parametric space. Let $\mathbf{p}_1(u_1, v_1)$ and $\mathbf{p}_2(u_2, v_2)$ be two points in the parametric space, then the distance between $\mathbf{p}_1$ and $\mathbf{p}_2$ with respect to $\mathbf{M}$, $\tilde{l}(\mathbf{M}, \mathbf{p}_1, \mathbf{p}_2)$, will equal to

$$\tilde{l}(\mathbf{M}, \mathbf{p}_1, \mathbf{p}_2) = \sqrt{\begin{pmatrix} u_2 - u_1 \\ v_2 - v_1 \end{pmatrix}^T \mathbf{M} \begin{pmatrix} u_2 - u_1 \\ v_2 - v_1 \end{pmatrix}} \quad (4)$$

The angle between two infinitesimal vectors $\mathbf{du}$ and $\mathbf{dv}$ with respect to $\mathbf{M}$, $\tilde{\theta}(\mathbf{M}, \mathbf{du}, \mathbf{dv})$, is defined as

$$\tilde{\theta}(\mathbf{M}, \mathbf{du}, \mathbf{dv}) = \begin{cases} \cos^{-1} \left(\frac{(\mathbf{du})^T \mathbf{M} \mathbf{dv}}{\tilde{l}(\mathbf{M}, \mathbf{du}) \cdot \tilde{l}(\mathbf{M}, \mathbf{dv})} \right) & \text{if } \mathbf{du} \times \mathbf{dv} > 0 \\ 2\pi - \cos^{-1} \left(\frac{(\mathbf{du})^T \mathbf{M} \mathbf{dv}}{\tilde{l}(\mathbf{M}, \mathbf{du}) \cdot \tilde{l}(\mathbf{M}, \mathbf{dv})} \right) & \text{otherwise} \end{cases} \quad (5)$$

Metric interpolation from background point mesh

In order to generate a well-graded anisotropic point mesh, the metric tensor will vary over the problem domain. A background point mesh will be used to define the metric tensor implicitly at its nodal points in the parametric space. As a point mesh contains no element, the metric tensor at a given point of the current mesh cannot be obtained by the usual method of element interpolation [14-19]. In the current complementation, a simple approach based on the nearest point criterion is used to retrieve the metric tensor from the background mesh. The metric tensor, $\mathbf{M}_p$, for a given point $\mathbf{p}$ in the current mesh is obtained from the background point mesh by using the condition

$$\mathbf{M}_p = \mathbf{M}_{p_i} \quad (6a)$$

where $\mathbf{M}_{p_i}$ is the metric tensor associated with the background point $\mathbf{p}_i$ such that $\mathbf{p}_i$ is the nearest point to $\mathbf{p}$. That is

$$|\mathbf{p}, \mathbf{p}_i| = \min_{j=1, \text{NNB}} |\mathbf{p}, \mathbf{p}_j| \quad (6b)$$

In Eqn. 6b, $\mathbf{p}_j$ and NNB are the j th node and the number of nodes in the background point mesh respectively. Note that in Eqn. 6b, instead of the metric distance measure, $\tilde{l}(\mathbf{M}, \mathbf{p}, \mathbf{p}_j)$, the usual Euclidean distance measure $|\mathbf{p}, \mathbf{p}_j|$ is used. In practice, the point $\mathbf{p}_i$ can be efficiently located by using an appropriate searching algorithm such as the alternating digital tree technique [21], the quadtree approach [22] or the bin sort algorithm [23] and the operation complexity of the searching operation will not exceed $O(\text{NNB}^{1/2})$ in general.

Obviously, Eqn. 6 is equivalent to interpolating the metric tensor from the Voronoi diagram of the background point mesh and the interpolation is discontinuous along the inter-cellular boundary. It should be noted that it is possible to obtain a continuous piecewise linear interpolation by using the corresponding Delaunay triangulation of the background mesh. However, the continuous approach is not adopted here since it is more complicated and time consuming while it is found that the simpler discontinuous approach is already good enough to converge rapidly in most practical applications. In addition, in finite point applications, the formation of the background Delaunay triangulation itself could be a difficult task to complete [3].

3. Metric finite point mesh generation

Overview

The finite point mesh generation scheme developed will follow a highly hierarchical generation sequence consists of the following three stages.

- (1) Control point formation
- (2) Boundary segments generation along sub-surface trimming curves
- (3) Point mesh generation and mesh quality enhancement

Similar to the previous work for automatic finite element mesh generation [14,15], in order to ensure the compatibility between meshes of different sub-surfaces, nodes and boundary segments generated in any one stage will not be modified in all subsequent stages. In addition, all generation steps will be carried out in the parametric spaces of the support surfaces and the 3D coordinates of the nodes generated will be computed by using Eqn. 1.

This mapping approach can ensure that all the points generated will strictly locate on the target surface patches.

Control point and boundary segment generation

Control points are the end-points of the trimming curves which directly define the problem domain to be meshed. Control points are the only set of nodes with both positions and numbering that are unchanged with respect to any target mesh density and number of refinement. Their positions are trivially obtained from the geometrical model of the problem domain. Following the hierarchical mesh generation principle, the node numbers of all the control points will be smaller than the node numbers for all boundary and interior points.

After all the controls points are defined, boundary segments with lengths compatible with the user specifications will be generated. In this stage, in addition to the metric specifications from the background point mesh, both the *geometrical* and the *topological* information of the surface patches are used to form a topologically correct and well-defined discretization of all boundary trimming curves. Furthermore, in order to facilitate the boundary segment generation procedure, all the boundary trimming curves will be classified either as *free* curves or as *dependent* curves according to their relationships with other boundary curves. More details for the boundary curve classification and the discretization procedure can be founded in Reference [15] and will not be repeated here.

When crack lines and surface branching are present in the problem domain, it is necessary first to slightly modify the positions of the boundary nodes (Fig. 3). This is to ensure that during subsequent interior node formation steps, the mesh generator will be able to distinguish two overlapping boundary segments and every addition point created will locate inside the problem domain. Details of the perturbation procedure can again be found in Reference [15].

Parametric space interior nodes generation

In this study, unlike many advancing front generation schemes, an active generation front will not be maintained during the point mesh generation. The advancing front approach is abandoned for the reason that it can neither speed up the point generation procedure nor it

can significantly improve the quality of the point mesh. Since during point mesh generation no element will be formed, the generation front will only consist of unlinked discrete points rather than continuous loops of front segments. Hence, unlike in the element formation step of an automatic finite element generation scheme, it cannot be used as a convenient tool to test whether a newly created point is inside the problem domain or not. Furthermore, since no fixed connectivity relationship will be defined among the generated nodes, it will also be difficult to use the point generation front to optimize the quality of the newly generated node during each generation step.

Instead of using the advancing front concept for node creation, the point mesh generation procedure proposed in this paper employs three simple basic operations, namely, metric boundary node offsetting, node insertion and node deletion, for point creation. More details for these three procedures are given in the following sections.

Metric boundary node offsetting

The point mesh generation procedure will be started by performing metric boundary node offsetting. The main objective of this step is to fill up the problem domain with an initial cloud of points which is boundary sensitive and with node spacing compatible with the metric specifications. This will help to reduce the number of iterations needed for the subsequent node insertion and deletion procedures. The procedure used is somewhat similar to the normal offsetting procedures used by Johnston and Sullivan [24] and Zhu et al [25] but in a more generalized form using metric measure to determine the directions of offset lines and the distance between successive offsetting nodes.

In the metric boundary node offsetting procedure, offset lines will be generated from all the boundary nodes enclosing the problem domain. For a typical boundary node A with neighbouring node B and C and surface metric $\mathbf{M}_A$ (Fig. 4), the internal angle $\tilde{\theta}_A = \tilde{\theta}(\mathbf{M}_A, AB, AC)$, can be computed by using Eqn. 5. Depends of the value of $\tilde{\theta}_A$, offset lines will be generated from point A according to Table 1 and Fig. 5.

Range of $\tilde{\theta}_A$	Number of offset lines	Offset angles from segment AB	Figure
$\tilde{\theta}_A \leq 240^\circ$	1	$\frac{\tilde{\theta}_A}{2}$	5a
$240^\circ < \tilde{\theta}_A \leq 315^\circ$	3	$90^\circ, \frac{\tilde{\theta}_A}{2}, \tilde{\theta}_A - 90^\circ$	5b
$310^\circ < \tilde{\theta}_A \leq 360^\circ$	5	$90^\circ, \frac{\tilde{\theta}_A}{4} + 45^\circ, \frac{\tilde{\theta}_A}{2}, \frac{3\tilde{\theta}_A}{4} - 45^\circ, \tilde{\theta}_A - 90^\circ$	5c

Table 1. Offset lines and offset directions from point A

The direction of the offset line, $\hat{\mathbf{n}}$, can be easily computed by using the vector rotation scheme described in Reference [14]. Let $\mathbf{M}_A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and $\hat{\mathbf{n}}_{AB}$ be the surface metric tensor at the point A and the direction of the segment AB respectively (Fig. 6). Then the direction of the offset line with an offset angle φ from AB is given by

$$\hat{\mathbf{n}} = \frac{\mathbf{n}}{|\mathbf{n}|} \quad \mathbf{n} = \mathbf{M}_A^{-\frac{1}{2}} \mathbf{R}(\varphi) \mathbf{M}_A^{\frac{1}{2}} \hat{\mathbf{n}}_{AB} = \mathbf{G}(a, b, c, d, \varphi) \hat{\mathbf{n}}_{AB} \quad (7)$$

In Eqn. 7, the rotation matrix $\mathbf{R}(\varphi)$ and the square root matrices $\mathbf{M}_A^{\frac{1}{2}}$ and $\mathbf{M}_A^{-\frac{1}{2}}$ are given by

$$\mathbf{R}(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \quad (8a)$$

$$\begin{aligned} \mathbf{M}_A^{-\frac{1}{2}} &= \frac{1}{d\sqrt{a+c+2d}} \begin{bmatrix} c+d & -b \\ -b & a+d \end{bmatrix} \\ \mathbf{M}_A^{\frac{1}{2}} &= \frac{1}{\sqrt{a+c+2d}} \begin{bmatrix} a+d & b \\ b & c+d \end{bmatrix} \\ d &= \sqrt{ac-b} \end{aligned} \quad (8b)$$

After some straightforward matrix operations, the matrix $\mathbf{G}(a, b, c, d, \varphi)$ can be expressed as

$$\mathbf{G}(a, b, c, d, \varphi) = \frac{1}{d} \begin{bmatrix} d\cos\varphi - b\sin\varphi & -c\sin\varphi \\ a\sin\varphi & d\cos\varphi + b\sin\varphi \end{bmatrix} \quad (8c)$$

From the boundary node A, interior nodes will be generated along the offset lines one by one such that the distances between successive points will equal to unity with respect to the surface metric tensor. Let $\mathbf{p}_L$ be the position of the last generated point on an offset line with direction $\hat{\mathbf{n}}$ and $\mathbf{M}_{\mathbf{p}_L}$ be the surface metric tensor at $\mathbf{p}$ (Fig. 7). The offset distance, α , can then be obtained from the conditions that

$$\mathbf{p}_N = \mathbf{p}_L + \alpha \hat{\mathbf{n}} \quad (9a)$$

$$1 = \tilde{l}(\mathbf{M}_{\mathbf{p}_L}, \mathbf{p}_L, \mathbf{p}_N) = \sqrt{(\alpha \hat{\mathbf{n}})^T \mathbf{M}_{\mathbf{p}_L} (\alpha \hat{\mathbf{n}})} = \alpha \sqrt{\hat{\mathbf{n}}^T \mathbf{M}_{\mathbf{p}_L} \hat{\mathbf{n}}} \quad (9b)$$

Hence,

$$\alpha = \frac{1}{\sqrt{\hat{\mathbf{n}}^T \mathbf{M}_{\mathbf{p}_L} \hat{\mathbf{n}}}} \quad (9c)$$

By first setting $\mathbf{p}_A = \mathbf{p}_L$ and using Eqn. 9, offset points will be generated one by one from all the offset lines radiating from point A. An offset point $\mathbf{p}_N$ will be accepted as a new interior point if and only if the following two conditions are satisfied.

- (1) The line segment $\mathbf{p}_L \mathbf{p}_N$ does not intersect with any domain boundary segment.
- (2) The shortest distance between $\mathbf{p}_N$ and the boundary segments, $\tilde{l}_d(\mathbf{M}_{\mathbf{p}_N}, \mathbf{p}_N, \mathbf{Q})$, is greater than or equal to 0.5 (Fig. 8).

Condition (1) will ensure that all the offsetting nodes is inside the problem domain while condition (2) will prevent the formation of an interior node which is too close to the domain boundary. Offsetting nodes satisfying conditions (1) and (2) will be generated along the offset lines until the last segment $\mathbf{p}_L \mathbf{p}_N$ intersects with the domain boundary (or when the point $\mathbf{p}_N$ is lying exactly on a boundary segment). Since the main function of boundary node offsetting is to fill up the problem domain with nodes with spacing compatible with the background metric, no distance checking between two nodes generated from two different offset lines will be performed. By using an appropriate data structures [21-23] to store the boundary segments information, both the intersection and the shortest distance problems encountered (conditions 1 and 2) can be solved within $O(NB^{1/2})$ operations where NB is the number of boundary segments. Hence, the overall complexity of the offsetting procedure will not exceed the bound

$$O(NNO \times NB^{1/2}) = O(NN \times NB^{1/2}) \quad (10)$$

In Eqn. 10, NNO is the total number of nodes generated in the offsetting procedure and will be of the same order as NN, the total number of nodes in the final mesh. In actual time measurement, it is found that the speed of the boundary offsetting procedure is very fast and only contributes a small portion to the total generation time needed.

Node Deletion

As no distance checking is carried out among nodes from different offset lines, closely spaced nodes may be present after the initial offsetting. Hence, a node elimination procedure will be used to remove all those nodes that are lying within a short distance from their neighbourhoods. All the nodes in the mesh (including all boundary nodes) will be considered one by one. Suppose that $\mathbf{p}$ is the current node under consideration, then based on the surface metric at $\mathbf{p}$, $\mathbf{M}_p$, a searching ellipse E with centre at $\mathbf{p}$ will be formed (Fig. 9). All the *interior* nodes that are inside E will be examined. An interior node $\mathbf{q}$ that is inside E will be deleted if the following conditions are satisfied.

- (1) The distance between $\mathbf{p}$ and $\mathbf{q}$, $\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q})$, is less than 0.5.
- (2) If the point $\mathbf{p}$ is an interior node, the line segment $\mathbf{pq}$ does not intersect with any other boundary segment (Fig. 9a). In case that $\mathbf{p}$ is a boundary node, the line segment $\mathbf{pq}$ intersects with the domain boundary at the point $\mathbf{p}$ only (Fig. 9b).

Similar to the boundary offsetting procedure, special data structures for all the interior nodes can be set up to speed up the searching of all the potential points inside the ellipse E . Usually, the operation complexity for locating all the nodes close to a given point $\mathbf{p}$ will equal to $O(NN^{1/2})$. Thus, the overall complexity for the whole deletion procedure will not exceed

$$O(NN \times NN^{1/2}) = O(NN^{3/2}) \quad (11)$$

Another remark is that only one pass of the deletion procedure will be sufficient to remove all the closely spaced nodes in the mesh and no iteration is required. Hence, the node deletion procedure will again only contribute a small portion to the total generation time needed.

Node Insertion

In some situations, the boundary offsetting procedure may not be able to completely fill up the whole problem domain with optimally spaced points. The main function of the node insertion procedure is to improve the node density of the point mesh by inserting points at where the point density is lower than the user specifications. All the interior nodes of the point mesh will be considered one by one during node insertion. Similar to the boundary offsetting and the node deletion procedure, the node insertion procedure for a typical interior node $\mathbf{p}$ will only involve a few steps as follow.

- (1) The principal directions $(\mathbf{e}_1, \mathbf{e}_2)$ and lengths (h_1, h_2) at the point $\mathbf{p}$ are computed from the metric tensor $\mathbf{M}_{\mathbf{p}}$.
- (2) Four potential new stencil points, $\mathbf{q}_i$, $i=1,2,3,4$ are introduced by using $(\mathbf{e}_1, \mathbf{e}_2)$ and (h_1, h_2) . The coordinates of these four stencil points are given by (Fig. 10)

$$\mathbf{q}_1 = \mathbf{p} - h_2 \mathbf{e}_2, \quad \mathbf{q}_2 = \mathbf{p} + h_1 \mathbf{e}_1, \quad \mathbf{q}_3 = \mathbf{p} + h_2 \mathbf{e}_2, \quad \mathbf{q}_4 = \mathbf{p} - h_1 \mathbf{e}_1 \quad (12)$$

- (3) Retrieve the metric tensor, $\mathbf{M}_{\mathbf{q}_i}$, for the four stencil points $\mathbf{q}_i$ from the background point mesh. The searching ellipse $\mathbf{E}_i$ centered at $\mathbf{q}_i$ will be constructed based on the principal directions and lengths of $\mathbf{M}_{\mathbf{q}_i}$ (Fig. 10).
- (4) All the existing points inside $\mathbf{E}_i$ are collected into a point set denoted as S_i .
- (5) If $S_i \neq \emptyset$, find the node $\mathbf{m}_i$ in S_i which is closest to $\mathbf{q}_i$. The shortest length corresponding to $\mathbf{q}_i$, $\tilde{l}_i$, are then defined as

$$\tilde{l}_i = \tilde{l}(\mathbf{M}_{\mathbf{q}_i}, \mathbf{q}_i, \mathbf{m}_i) \quad (13)$$

If $S_i = \emptyset$, set $\tilde{l}_i = 1.0$.

- (5) The potential point $\mathbf{q}_i$ will be accepted as a new point if the following conditions are satisfied.
 - (i) The line segment $\mathbf{q}_i \mathbf{p}$ does not intersect with any boundary segment. I.e. the point $\mathbf{q}_i$ is inside the problem domain.
 - (ii) The shortest length associated with $\mathbf{q}_i$, $\tilde{l}_i$, is greater than 0.5 so that the newly inserted node will not be too close to an existing node in the mesh.

Unlike node deletion, usually a few cycles of node insertion are needed to ensure that a sufficient number of optimally placed points will be generated over the whole problem domain. In the first iteration, all the existing interior nodes will be considered for node insertion while in subsequent cycles only those nodes that are created during the previous cycle will be considered. Usually, after a few iterations, the number of nodes inserted will be reduced to a small proportion of the number of existing nodes. In the current implementation, the node insertion procedure will be stopped whenever the number of new nodes added is less than 1% of the number of existing nodes in the mesh. The operation complexity of the node insertion procedure will equal to

$$O(NI \times NN^{1/2}) \quad (14)$$

where NI is the number of node inserted. After some computational experiments, it is found that NI is proportional to NN . Hence the order of operation complexity of the node insertion procedure will again equal to $O(NN^{3/2})$.

4. Point mesh quality measurement and enhancement

After the point mesh generation steps are completed, the domain will be covered by points with node spacing compatible with the user specifications. Just similar to the automatic mesh generation, it is found that the grading of the final the mesh can be improved by applying a Laplacian smoothing [26] procedure to the point mesh. However, for a point mesh, no fix connectivity relationship exists among the interior nodes. Hence, before the point smoothing procedure is carried out, it is first necessary to define the nodal connectivity of a point mesh. In this study, the distance with respect to the metric space is used as the criterion for defining the connectivity relationship of a point mesh. Based the metric distance measure between two points in the parametric space (Eqn. 4), the *r-connectivity* or the *r-neighbourhoods* of a given node will be defined and used in the point mesh smoothing procedure.

r-connectivity (neighbourhoods) for an interior point $\mathbf{p}$

For a given interior point $\mathbf{p}$ in a point mesh, let $\mathbf{M}_p$ be the metric tensor at $\mathbf{p}$ with principal directions $(\mathbf{e}_1, \mathbf{e}_2)$ and principal lengths (h_1, h_2) . For any real number $r > 0$, it is always

possible to construct an ellipse, E_r , centered at $\mathbf{p}$ with major and minor axes in the $\mathbf{e}_1$ and $\mathbf{e}_2$ directions and with lengths equal to rh_1 and rh_2 respectively (Fig. 11). The *r-neighbourhood number*, NC_r , for the node $\mathbf{p}$ is then defined as the number of nodes (excluding $\mathbf{p}$) that are lying inside or on E_r . The real number r is called the *connectivity factor* of the mesh. Furthermore, the *r-connectivity* of the mesh is defined as the collection of all the *r-neighbourhood numbers* information for all the interior nodes. Obviously, for all the nodes in any point mesh, the following conditions will always be true.

$$NC_0=0, NC_d=1 \text{ and } NC_\infty=NN \quad (15)$$

In Eqn. 15, d is the distance between the point and its nearest neighbourhood and NN is the total number of nodes in the mesh. For a uniform mesh, one will have $NC_d=4$ for all the nodes in the mesh. For a graded point mesh, one can expect that the *r-neighbourhood number* for some nodes will deviate from the ideal value of 4 and the *average neighbourhood value* of the mesh, $\overline{NC}_r$, defined as

$$\overline{NC}_r = \frac{1}{NNI} \sum_{i=1}^{NNI} NC_r^i \quad (16)$$

can be higher or less than 4. In Eqn. 16, NNI and NC_r^i are the number of interior nodes and the *r-neighbourhood number* for the i th interior node respectively.

The α quality for an interior node and the overall $\overline{\alpha}_r$ quality for a point mesh

With the *r-connectivity* defined in the last section, it is then possible to define the grading quality of a point mesh. For a given interior node $\mathbf{p}$ with *r-neighbourhood number* NC_r . The $\alpha(r, \mathbf{p})$ quality of the point $\mathbf{p}$ can be defined as

$$\alpha(r, \mathbf{p}) = \left(\prod_{i=1}^{NC_r} \min(\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i), \frac{1}{\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i)}) \right)^{\frac{1}{NC_r}} \text{ if } NC_r > 0 \quad (17a)$$

otherwise $\alpha(r, \mathbf{p}) = \varepsilon$

In Eqn. 17a, $\mathbf{q}_i$ is the i th neighbourhood of $\mathbf{p}$ and ε is a small positive constant taken as 0.01 in the current implementation. Note that the term

$$\min(\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i), \frac{1}{\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i)}) \quad (17b)$$

is a measure which reflects the amount of distance deviation between the two points $\mathbf{p}$ and $\mathbf{q}_i$. If point $\mathbf{q}_i$ is located at an ideal distance from $\mathbf{p}$, the expression in Eqn. 17b will attend a maximum value of unity and its value will be reduced as the distance deviation between $\mathbf{p}$ and $\mathbf{q}_i$ increases. $\alpha(r, \mathbf{p})$ is thus the geometrical mean of all the distance measures of the all the neighbourhoods of $\mathbf{p}$. Obviously, $\alpha(r, \mathbf{p})$ will attend a maximum value of one when all the neighbourhoods of $\mathbf{p}$ are lying exactly on the ellipse corresponding to $\mathbf{M}_p$.

Once the $\alpha(r, \mathbf{p})$ values of all the interior points are known, the overall $\bar{\alpha}_r$ quality of a point mesh is defined as the geometrical mean of all the $\alpha(r, \mathbf{p})$ values. That is

$$\bar{\alpha}_r = \left(\prod_{i=1}^{NNI} \alpha(r, \mathbf{p}_i) \right)^{\frac{1}{NNI}} \quad (18)$$

In general, only using the $\bar{\alpha}_r$ and the $\overline{NC}_r$ values may not be sufficient to assess the grading quality of a point mesh accurately. The reason is that a mesh with a high $\bar{\alpha}_r$ value may not have a good NC_r distribution. Hence, a more complete way to assess the quality of the point mesh is to consider both the $\bar{\alpha}_r$ and $\overline{NC}_r$ values together with the $\alpha(r, \mathbf{p})$ and the NC_r distributions of the nodes in the mesh.

Another remark about point mesh quality measurement is that in order to reflect the point density distribution of a point mesh correctly, the value of the measurement factor r should be carefully selected. In fact, r is the most important parameter that will affect the mesh quality measurement results. It is found in this study that in order to measure the quality of a point mesh accurately, the value of r should within the range [1.10, 1.25].

Point mesh smoothing

The main objective of point mesh smoothing is to increase the grading quality of a point mesh. The operation is somehow similar to the edge length smoothing procedure used in the metric finite element mesh generator described in Reference [14]. The basic idea is to adjust the positions of nodes in such a way that the distance between two connected nodes

will be as close to unity as possible. In order to carry out the point mesh smoothing procedure, a suitable value of r will be selected and the r -connectivity information of the mesh will be created. All the interior nodes will be considered one by one for smoothing. Again, let $\mathbf{p}$ be a typical node under consideration with r -neighbourhood number NC_r . Furthermore, let $\mathbf{q}_i, i=1, \dots, NC_r$ be the nodes connected to $\mathbf{p}$ (Fig. 12). For the node $\mathbf{q}_i$ the new ideal position of $\mathbf{p}$, denoted as $\mathbf{p}'_i$, such that the length $\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i) = 1$ can be obtained by

$$\mathbf{p}' = \mathbf{q}_i + \frac{1}{\tilde{l}(\mathbf{M}_p, \mathbf{p}, \mathbf{q}_i)}(\mathbf{q}_i - \mathbf{p}) \quad (19)$$

Eqn. 19 can be used to compute a set of possible new positions of $\mathbf{p}$ with respect to its neighbourhoods. The final new position of $\mathbf{p}$ is then defined as the centroid of all $\mathbf{p}'_i$. This new position will be accepted if and only if the new position of the node remains inside the problem domain and the $\alpha(r, \mathbf{p})$ value of $\mathbf{p}$ is increased after the reposition. In practice, a value of $r=1.25$ is used in the point smoothing procedure and usually three to five cycles of point smoothing will be sufficient to achieve the convergence and improve the overall quality of the mesh.

After some simple deductions, it can be proved that the operation complexity of the smoothing procedure (included both setting up the r -connectivity and the smoothing operation) is of the order of $O(NN^{3/2})$. However, when comparing with node insertion and deletion, more computationally intensive distance measurements are necessary in the smoothing procedure. Consequently, it is found that the point smoothing procedure is the most time consuming procedure in the whole finite point mesh generation program.

5. Point meshes generation examples

In this section, several mesh generation examples will be presented to test and demonstrate the performance of the point mesh generator. The point density of these examples will be controlled by some fictitious 3D metric tensors. Both 2D flat surfaces and 3D bi-variate NURBS patches will be used in the examples. Isotropic and anisotropic point meshes will be generated by specifying the 3D metric tensors implicitly over the nodal points of some initial meshes. The corresponding surface metric tensors are then computed by using Eqn.

3. Surfaces with different complexities (simply and multi-connected, with one or more than one sub-surfaces and the presence of crack and repeated lines) were used. However, since the exact detail definitions of the surfaces and the 3D metric tensors are too lengthy and tedious to write down explicitly, their definitions will not be given here. Instead, it is more convenient to examine the performance of the point mesh generation scheme by studying the quality of the point meshes generated. All the important characteristics and features of the surfaces and the metric tensors used are summarized in Table 1.

The initial point meshes used are shown in Figs. 13a to 19a. Note that all the initial meshes are very coarse and consist of boundary nodes only. In each example, four refinements were carried out to generate the final meshes shown in Figs. 13b to 19b. In all the examples, the connectivity factor r used in point mesh smoothing was set as 1.25. A summary of the characteristics for the final meshes is listed in Table 2. Note that when carrying out the mesh quality measurement, a more stringent value of $r=1.1$ was used. From Table 2, it can be concluded that the quality of all the meshes generated is good and the geometrical mean $\bar{\alpha}_{1,1}$ values of the final meshes are all higher than 0.8. Furthermore, more than 90% of all the nodes in the final meshes have $\alpha_{1,1}$ values within the range [0.75-1.0]. Table 2 also shows that the connectivity of the final mesh is acceptable in the sense that more than 80% of the nodes in the mesh are connected to 2 to 6 nodes.

Example	Description	NINT	No. of sub-surface	Metric tensor	Singular point	Internal openings	Crack line	Branching/ Repeated line
1 (Fig. 13)	Jet fighter	20	1	A	No	Yes	No	No
2 (Fig. 14)	Logo	36	1	I	No	Yes	No	No
3 (Fig. 15)	Hemisphere	7	1	I	Yes	No	No	Yes
4 (Fig. 16)	Corner	12	4	I	Yes	Yes	No	No
5 (Fig. 17)	Octant	11	1	A	Yes	Yes	Yes	No
6 (Fig. 18)	Beam	42	3	A	No	Yes	No	Yes
7 (Fig. 19)	Engine	18	8	I	Yes	No	No	Yes

Table 1. Characteristics of surfaces and metric tensors
(A=anisotropic, I=isotropic, NINT= No. of nodes in the initial mesh)

Example	NN	$\bar{\alpha}_{1,1}$	$\overline{NC}_{1,1}$	$\alpha_{1,1,[0.75-1.0]}$ (%)	$NC_{1,1,[2-6]}$ (%)
1 (Fig. 13b)	8496	0.84	4.8	99.4	88.3
2 (Fig. 14b)	3269	0.82	5.2	96.0	81.2
3 (Fig. 15b)	15489	0.87	4.6	98.0	87.1
4 (Fig. 16b)	10992	0.87	4.6	98.7	90.0
5 (Fig. 17b)	3375	0.84	4.6	95.9	84.9
6 (Fig. 18b)	17321	0.83	4.7	94.6	87.0
7 (Fig. 19b)	13519	0.84	4.6	97.7	90.3

Table 2. Characteristics of the final point meshes generated

Legend for Table 2

NN = Number of nodes in the mesh

$\bar{\alpha}_{1,1}$ = Geometrical mean $\alpha_{1,1}$ value of all interior points

$\overline{NC}_{1,1}$ = Mean $NC_{1,1}$ value of all interior points

$\alpha_{1,1,[0.75-1.0]}$ = Percentage of nodes with $\alpha_{1,1}$ value inside the range [0.75-1.0]

$NC_{1,1,[2-6]}$ = Percentage of node with $NC_{1,1}$ value inside the range [2-6]

For the convergence rate of the point mesh generation process, the convergence history for Example 3 (Fig. 15) is shown in Fig. 20. From Fig. 20, it can be seen that both the $\alpha_{1,1}$ and the $NC_{1,1}$ distributions of the mesh in fact converged to their final forms after two to three iterations. This indicates that the background interpolation scheme based on the distance measure described in section 2 can effectively allow the mesh generation scheme to capture the variation of the surface metric tensor. Fig. 20 also shows that in the final mesh, most of the nodes in the mesh have high $\alpha_{1,1}$ values close to the ideal value of unity.

For the speed of the finite point mesh generation procedure, detail timing confirms that the order of complexity of the procedure is proportional to $O(NN^{3/2})$. Since no element will be formed and fewer computational intensive steps are required, for a mesh with similar number of nodes, the surface point mesh generator will incur much less computational effort when comparing with metric finite element mesh generation. For the current implementation, it is found that when running on a low-end PC computer equipped with a 266MHz Pentium II CPU and 64MB of RAM, the generator can generate 15000 points (the background mesh used contains 11000 points) in about 13 seconds and is about ten times faster than surface finite element mesh generation [15].

6. Conclusions and future investigations

In this paper, a new metric finite point generator has been developed for the finite point mesh generation on 2D flat surfaces and bi-variate parametric surfaces. The point density of the mesh is controlled by the metric measurement approach so that anisotropic point meshes can be generated. A metric boundary offsetting procedure is used to initialize the point mesh generation procedure and to fill up the problem domain with points compatible with the user specifications. Node insertion and deletion procedures are then employed to improve the point density of the mesh. A new method based on the concept of metric distance measurement is suggested to establish the connectivity relationships of the interior nodes of the mesh. Based on the connectivity information, two parameters, namely, the point quality value, α_r , and the node connectivity number, NC_r , are defined to measure the quality of a point mesh. Finally, a new point mesh smoothing procedure is proposed to enhance the quality of the point mesh. From the numerical examples given in the last section, it is demonstrated that the proposed mesh generator can generate well-graded point meshes with node density distribution compatible with the input metric specifications. It is also found that since relatively few computational intensive steps are involved in the current point mesh generation scheme, the point mesh generation speed is about ten times faster than metric finite element mesh generation. The same set of examples also indicated that the convergence rate of the suggested scheme is satisfactory. Hence, in theory, with a reliable error estimation procedure and a suitable refinement scheme, the present point mesh generation scheme can be used for the development of an automatic adaptive meshless refinement procedure.

As all the generation principles used in the mesh generator can be readily extended to 3D situation, one immediate extension of the present research is to develop a similar point mesh generator for 3D anisotropic point mesh generation. Regarding this, a metric surface mesh generator (e.g. the one described in Reference [15]) is required to first generate a surface mesh completely enclosing the 3D problem domain. The finite point generation scheme will then be used to fill up the computational domain with a point mesh.

Another potential future research area is to explore the possibility of using this finite point generation scheme as the first step for automatic finite element mesh generation. Since the spacing characteristics of the point mesh generated by the present scheme are already

compatible with the user specification, it can be used as a good starting point for the formation of finite element mesh and to increase the speed of the element mesh generation procedure.

Reference

1. T. Belytschko, Y. Y. Lu and L. Gu, Element-free Galerkin Methods method, *Int. J. Numer. Methods Engrg.*, **37**, 229-256 (1984)
2. Y. Y. Lu, T. Belytschko and L. Gu, A new implementation of the element free Galerkin method, *Comput. Methods Appl. Mech. Engrg.*, **113**, 397-414 (1994)
3. E. Onate, S. Idesohn, O. C. Zienkiewicz and R. L. Talyor, A finite point method in computational mechanics. Application to convective transport and fluid flow, *Int. J. Numer. Methods Engrg.*, **39**, 3839-3866 (1996)
4. Wing Kam Liu, Sukky Jun, Shaofan Li, Jonathan Adee and Ted Belytschko, Reproducing kernel particle methods for structural dynamics, *Int. J. Numer. Methods Engrg.*, **38**, 1655-1679 (1995)
5. H. J. Chung, T. Belytschko, An error estimate in the EFG method, *Computational Mechanics*, **21**, 91-100 (1998)
6. Krongauz, Y. Belytschko, T. Consistent pseudo-derivatives of meshless methods *Comput. Methods Appl. Mech. Engrg.*, **146**, 371-386 (1997)
7. Liszka, T J. Duarte, C A M. Tworzydlo, W W. hp-meshless cloud method, *Comput. Methods Appl. Mech. Engrg.*, **139**, 263-288 (1996)
8. Belytschko, T. Krongauz, Y. Organ, D. Fleming, M. Krysl, P. Meshless methods: an overview and recent developments, *Comput. Methods Appl. Mech. Engrg.*, **139**, 3-47 (1996)
9. Krysl, P. Belytschko, T. Element-free Galerkin method: convergence of the continuous and discontinuous shape functions, *Comput. Methods Appl. Mech. Engrg.*, **148**, 257-277 (1997)
11. Belytschko, T. Krysl, P. Krongauz, Y. Three-dimensional explicit element-free Galerkin method, *Int. J. Numer. Methods Fluids*, **24**, 1253-1270 (1997)
12. Fleming, M. Chu, Y A. Moran, B. Belytschko, T. Enriched element-free Galerkin methods for crack tip fields, *Int. J. Numer. Methods Engrg.*, **40**, 1483-1504 (1997)
13. N. Sukumar, B Maran and T. Belytschko, The natural element method in solid mechanics, *Int. J. Numer. Methods Engrg.*, **43**, 839-887 (1998)

14. C. K. Lee, Automatic adaptive mesh generation using metric advancing front approach, *Engrg. Computations*, **16**, 230-263 (1999)
15. C. K. Lee, Automatic metric advancing front triangulation over curved surfaces, To appear in *Engrg. Computations*.
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. P. L. George and E. Seveno, The advancing-front mesh generation method revisited, *Int. J. Numer. Methods Engrg.*, **37**, 3605-3619 (1994)
19. P. Moller and P. Hansbo, On advancing front mesh generation in three dimensions, *Int. J. Numer. Methods Engrg.*, **38**, 3551-3569 (1995)
20. R. Lohner and E. Onate, An advancing front point generation technique, *Commun. Numer. Meth. Engng*, **14**, 1097-1108 (1998)
21. J. Bonet and J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems, *Int. J. Numer. Methods Engrg.*, **31**, 1-17 (1991)
22. R. Lohner, Some useful data structures for the generation of unstructured grids, *Commun. Numer. Methods Engrg.*, **4**, 123-135 (1988)
23. S. W. Sloan, A fast algorithm for generating constrained Delaunay triangulations, *Comput. Struct.*, **47**, 441-450 (1993)
24. B. P. Johnson and J. M. Sullivan, Jr., Fully automatic two-dimensional mesh generation using normal offsetting, *Int. J. Numer. Methods Engrg.*, **33**, 425-442 (1992)
25. 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)
26. L. R. Herrmann, Laplacian-Isoparametric grid generation scheme, *J. Engrg. Mech. Division ASCE.*, **12**, 749-759 (1976)

Fig. 1 Support surface mapping and trimming curves enclosing the problem domain

Fig. 2 Secondary mapping for singular point

Fig. 3 Perturbation of overlapping boundary segments

Fig. 4 Typical boundary node A

(a) $\tilde{\theta}_A \leq 240^\circ$

(b) $240^\circ < \tilde{\theta}_A \leq 315^\circ$

(c) $\tilde{\theta}_A > 315^\circ$

Fig. 5 Boundary offsetting

Fig. 6 Direction of offset line with an angle φ from segment AB

Fig. 7 Generation of offset point p_N

Fig. 8 Shortest distance between p_N and the domain boundary

(a) p is an interior node

(b) p is a boundary node

Fig. 9 Node deletion, node q will be deleted while node q' will be kept

Fig. 10 Stencil point surrounding point p and the searching ellipse E_1

Fig. 11 r -connectivity for an interior point p

Fig. 12 Point smoothing for point p , $NC_r=4$

(a) Initial Mesh

(b) Final Mesh

Fig. 13 Example 1, Jet fighter

(b) Initial Mesh

(b) Final Mesh

Fig. 14 Example 2, Logo

(a) Initial Mesh

(b) Final Mesh

Fig. 15 Example 3, Hemisphere

(a) Initial Mesh

(b) Final Mesh

Fig. 16 Example 4, Corner

(a) Initial Mesh

(b) Final Mesh

Fig. 17 Example 5, Octant

(a) Initial Mesh

(b) Final Mesh

Fig. 18 Example 6, Beam

(a) Initial Mesh

(b) Final Mesh

Fig. 19 Example 7, Engine

(a) Convergence history for Example 3, $\alpha_{1,1}$

(b) Convergence history for Example 3, $NC_{1,1}$

Fig. 20 Convergence history for Example 3