Finite element mesh generation and adaptive meshing

S H Lo
University of Hong Kong, China

Summary

This review paper gives a detailed account of the development of mesh generation techniques on planar regions, over curved surfaces and within volumes for the past years. Emphasis will be on the generation of the unstructured meshes for purpose of complex industrial applications and adaptive refinement finite element analysis. Over planar domains and on curved surfaces, triangular and quadrilateral elements will be used, whereas for three-dimensional structures, tetrahedral and hexahedral elements have to be generated. Recent advances indicate that mesh generation on curved surfaces is quite mature now that elements following closely to surface curvatures could be generated more or less in an automatic manner. As the boundary recovery procedure are getting more and more robust and efficient, discretization of complex solid objects into tetrahedra by means of Delaunay triangulation and other techniques becomes routine work in industrial applications. However, the decomposition of a general object into hexahedral elements in a robust and efficient manner remains as a challenge for researchers in the mesh generation community. Algorithms for the generation of anisotropic meshes on 2D and 3D domains have also been proposed for problems where elongated elements along certain directions are required. A web-site for the latest development in meshing techniques is included for the interested readers.

Key words: review; finite element mesh generation; adaptive; two-dimensional; surface; three-dimensional

Prog. Struct. Engng Mater. 2002; **4**:381–399 (DOI: 10.1002/pse.135)

1 Introduction

After a continuous research and development for more than three decades, the finite element method (FEM) has now become a general tool in solving various engineering problems. As the concept of the FEM is based on the decomposition of a continuum into a finite number of discrete elements, the solution obtained, in general, will be only an approximation to the exact solution. The error of the approximation depends on the nature of the problem, the accuracy of individual elements, the number of elements used and the position of the sampling nodes, etc. The objective of an adaptive refinement procedure is to control the discretization error by increasing the number of degrees of freedom in regions where the previous finite element model is not adequate in an optimal manner.

A number of highly effective adaptive refinement procedures have been proposed in many areas of applications of the FEM. The range of applications covers the standard two-dimensional linear elasticity problem[1–3], the plate and shell bending problems[4,5], solutions for seepage and other field problems[6], complicated flow formulations[7], visualization of shear bending in viscoplastic solids[8], application to nonlinear shell analysis[9], structural optimization problems[10] and computational fluid dynamic problems[11].

However, three-dimensional continuum mechanics problems received relatively little attention during the early development of the adaptive process, and the literature on this topic has appeared only rather recently[12]. An element-wise *a posteriori* error estimator for brick elements was suggested by Ohtsubo & Kitamura[13]. Adaptive mesh refinement scheme using a tetrahedral element octree approach was used by Baehman *et al.*[14] and more recently by Pressburger & Perucchio[15]. Remeshing scheme using closed form stiffness matrix solutions to speed up the element formulation process was reported by Schurtze *et al.*[16]. Since most of the theoretical work related to the error estimation of the FEM[17] were originally deduced from the more general three-dimensional case and then simplified and adapted for two-dimensional applications[18,19], the major difficulty hindering the development of a full three-dimensional adaptive refinement model is,

Published online 23 September 2002

obviously, not due to the lack of a sound theoretical understanding of the problem, but rather to practical computational difficulties. Two of the major difficulties are: (i) the huge amount of computational cost involved for the solution of the large systems of equations formed for a full three-dimensional adaptive analysis; and (ii) the difficulty of generating a well-graded adaptive FEM mesh for general three-dimensional objects.

Nevertheless, as mentioned before, with the rapid increase in computer speed, the expensive computational cost for three-dimensional problems will sooner or later become affordable to more and more FEM users. Moreover, with the recent advances in three-dimensional mesh generation algorithms, the discretization of a general polyhedral object into tetrahedra is now to a large extent feasible, even for the most complex domains. Work on the generation of hexahedral element meshes over general volumes has also aroused much interest, and many important results have been achieved[20–46]. Unstructured mesh generation is the main theme of discussion for this paper, as structured meshes, in general, have less flexibility to meet the requirement of rapid element size variation of an adaptive mesh.

To divide a general arbitrary domain into elements, essentially there are two ways that this could be achieved: (i) filling the interior, as yet unmeshed region with elements directly; and (ii) modifying an existing mesh that already covers the domain to be meshed. The advancing front approach represents mesh generation methods based on the first idea; the generation front is defined as the boundary between the meshed and the unmeshed parts of the domain. The key step that must be addressed for the advancing front method is the proper introduction of new elements to the unmeshed region and a consistent update of the generation front as elements are formed. Triangular and tetrahedral element meshes generated by this method are common[47,48], and the methods for generating quadrilateral and hexahedral meshes by this approach are referred to as paving or plastering techniques[49–52]. Meshing by the second idea is the well-known Delaunay triangulation method, which provides a systematic approach to modify and refine a triangular mesh by adding interior nodes. By the insertion algorithm, the key step is to introduce a new point properly to the existing mesh. The structure of the mesh has to be modified to absorb this point to form a new mesh, according to the Delaunay empty-sphere criterion. Delaunay triangulation has been used mainly for the generation of triangular and tetrahedral meshes. In fact, it provides a general rule to govern how points are connected to form a triangulation in n-dimensional space. However, it does not provide hints as to where should points be inserted to generate meshes appropriate for FEM and adaptive refinement analysis. Hence, different ways of introduction of interior points give rise to various versions of Delaunay triangulation, with different characteristics. It is remarked that the modified or finite quadtree/octree techniques[53,54] represent just one of many ways of generation of node points for a Delaunay triangulation. From a survey by Owen[55] of commercial mesh generation software, it is not surprising that most of the practical mesh generation software is based on these two fundamental concepts. It is interesting to note that these two methods are not mutually exclusive. In fact, they have been combined to produce robust and efficient mesh generation algorithms of the advancing front-Delaunay and Delaunay-advancing front types[56–59], in which generation of interior nodes and front evolution follow the advancing front approach, while element connections are modified based on the Delaunay criterion.

The problem of adaptive meshing can be defined as follows. Given a physical domain Ω and a finite element discretization M of Ω into a collection of elements, together with a node spacing function ρ defined over the entire domain Ω, the task of adaptive meshing is to discretize domain Ω into finite elements based on the existing mesh M and consistent with the given node spacing function ρ. The major problem for adaptive meshing is to generate gradation meshes over two-dimensional domains, on curved surfaces and over volumes in compliance with a specified nodal spacing function, as derived from the error estimation, in an efficient and reliable manner. The adaptive meshing problem for two-dimensional domains, on surfaces and over volumes will be presented in Sections 2, 3 and 4, respectively.

2 Adaptive meshing on two-dimensional domains

Extensive research has been done in the past on the development of robust automatic finite element mesh generators on two-dimensional domains. A scheme for the classification of mesh generation methods was proposed by Ho-Le[60]. It is noted that not every method described by Ho-Lee is suitable for mesh generation in conjunction with an adaptive refinement process. It seems that the most common and effective way to achieve a prescribed local mesh density requirement is by using the node spacing function approach proposed by Frey[61].

The node spacing function which specifies the element size of the next discretization can be defined on consideration of the current geometry of the body, *a posteriori* error estimate of the solution and some economic constraints limiting the maximum number of elements in the mesh. The node spacing function, which can take only positive values, should be defined, either explicitly or implicitly, over the problem domain. In the explicit form, the analyst supplies to the mesh generator a routine which computes the desired node spacing at every point of the domain, whereas in the implicit form, the spacing function values are provided at all nodal points. Within an element, node spacing is obtained by interpolation similar to that in the finite element method.

The following are some common mesh generation techniques that could be used to generate a finite element mesh compatible with a given node spacing function on a two-dimensional domain.

1. Delaunay triangulation.
2. Advancing front approach
3. Mesh generation using contours
4. Coring technique
5. Quadtree technique
6. Refinement by subdivision

2.1 TRIANGULAR MESH

Adaptive refinement analysis first achieved success in experiments with two-dimensional problems. Triangular meshes lend themselves very readily to the situation of adaptive analysis as triangles are less sensitive to shape distortion such that meshes with rapid changes of element size can be employed. Furthermore, many mesh generation techniques, such as Delaunay triangulation, advancing front approach, mesh generation using contours, coring technique, quadtree technique, etc. are available to generate gradation triangular meshes in a great variety of ways for different problems at hand.

2.1.1 Two-dimensional Delaunay triangulation

By far the most popular triangular mesh generation techniques are based on the concept of Delaunay triangulation[62–67]. The Delaunay criterion, also known as the empty-circle property, states that any node must not be contained within the circumscribing circle of any triangle of the mesh, as shown in Fig. 1. Although the Delaunay criterion has been known for many years, it was not until the work of Lawson[68] and Watson[69] that the criterion was exploited for developing algorithms to form a convex triangulation connecting a given set of points. With the rapid development of the FEM, the Delaunay triangulation algorithm was further extended to generate valid finite element meshes for numerical engineering analysis, by Weatherill *et al.*[70,71], Baker & Vassberg[72,73], George & Hermeline[74–76], and others.

In the incremental algorithm of Bowyer & Watson[69,77], the points are processed one at a time. In a typical step of point insertion, the triangles whose circumcircle contains the insertion point are identified and deleted. New triangles are constructed in the cavity left behind by the triangles removed. Hence, the efficiency of the triangulation algorithm depends on how fast we can identify the triangles to be removed and determine correctly the cavity for insertion, and the speed with which the circumcentres, circumradii and adjacency relationship of the new triangles are calculated.

The Delaunay criterion itself is not an algorithm for mesh generation. It merely provides a rule to connect a set of existing points in space. As a result, it is necessary to design a method to determine the number and the locations of node points to be inserted within the domain of interest. A typical approach is to first create a triangular mesh large enough to contain the entire domain. The boundary nodes are then inserted and connected according to the Delaunay criterion, and this forms a triangulation of the boundary nodes. More nodes are then inserted incrementally into the coarse boundary mesh, redefining the triangles as each new node is introduced, until a desirable number of elements are formed at appropriate positions.

In finite element applications, there is a requirement that an existing surface triangulation be maintained, i.e. the integrity of the domain boundary. In most Delaunay triangulation process, before interior nodes are inserted, a tessellation of the nodes on the domain boundary is produced. However, in this process, there is no guarantee that boundary segments will all be present in the triangulation. In many implementations, the approach is to tessellate the boundary nodes using a standard Delaunay algorithm without regard to the integrity of the domain boundary. A second step is then employed to force or recover the boundary segments. Of course, by doing so, the triangulation in general is not strictly Delaunay, hence the term 'boundary-constrained Delaunay triangulation'. In two dimensions, the edge recovery is relatively straightforward. Weatherill[78] described how edges of a triangulation may be recovered simply by swapping diagonals, as shown in Fig. 2.

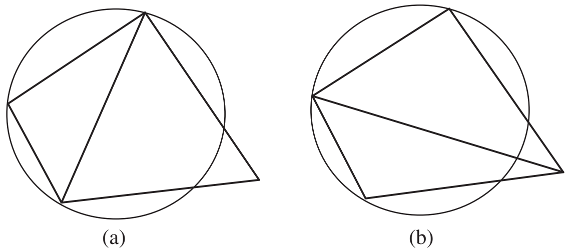


Fig. 1 (a) Delaunay triangulation; (b) non-Delaunay triangulation

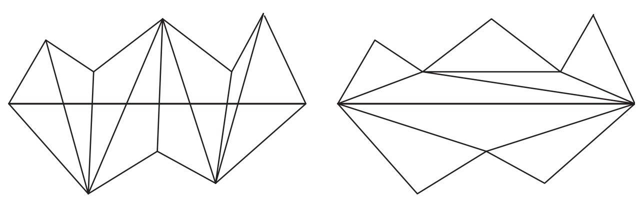


Fig. 2 The thick line is recovered by diagonal swaps

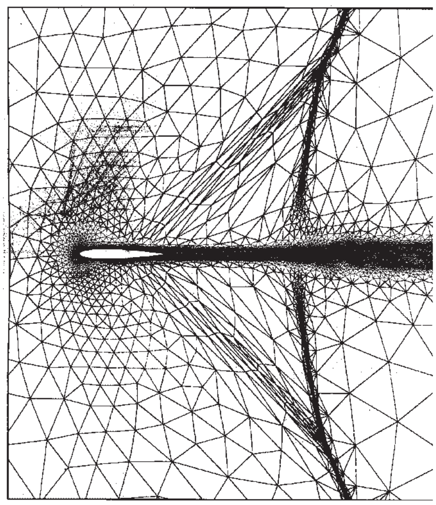


Fig. 3 Anisotropic mesh by Delaunay triangulation

There are many ways to the second phase of point insertion according to the node spacing function, which in fact would lead to meshes of different characteristics. Hermeline[79] proposed a scheme in which points are inserted at the barycentre under certain conditions. Some researchers have proposed to insert points at circumcentres of triangles,[80,81]. George[82] proposed the insertion of points along edges of triangles. Others make use of a set of points at predetermined positions with the aid of a regular grid, a quadtree network or some sort of spatial decomposition methods,[83–85]. A combined scheme with the advancing front approach was also put forward, in which points are inserted at strategic positions as determined in a frontal process and element connections are modified based on the Delaunay criterion[56–59,86–88].

Borouchaki *et al.*[89,90] made use the concepts of control space and length criterion for the insertion of points to create an adaptive mesh of variable element sizes. The element size map can be explicitly given as a continuous function over the entire domain or implicitly defined by means of a background mesh[91]. Points are inserted based on the subdivision of the edges of the elements. This idea was later extended with the introduction of a general metric tensor for the generation of anisotropic meshes in which not only element size can vary, but also elements are subjected to different size requirements along different directions, as shown in Fig. 3. As discussed in the following sections, anisotropic mesh generation on two-dimensional domains is closely related to the meshing of analytical curved surfaces.

2.1.2 Advancing front method (two-dimensional)

The concept of the frontal approach was first applied to generate unstructured triangular meshes[92], and later modified for the generation of quadrilaterals by Zhu *et al.*[93] and extended into three dimensions, surface meshing, adaptive meshing, anisotropic

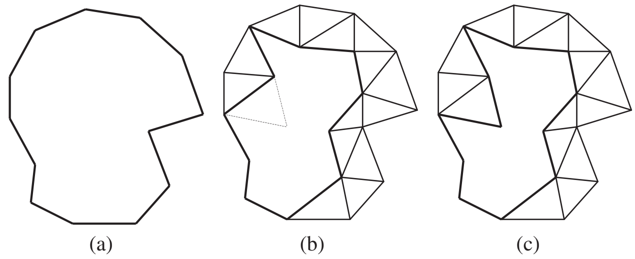


Fig. 4 (a) Initial front (domain boundary); (b) current front; (c) updated front with new element included

meshing, large scale problems and parallel meshing, etc. by other researchers[48, 94–101].

In two dimensions, the generation front consists of line segments on which triangles are to be created. It advances each time a triangle is constructed, and it is properly updated to include the new triangle, as shown in Fig. 4. At the start of the generation process, the generation front is given exactly by the collection of all the segments on the boundary. While the domain boundary always remains the same, the generation front changes continuously throughout the generation process and has to be updated whenever a new element is formed. The updating process consists of removing the faces of the generated element that belonged to the front and including the new faces with the correct orientation. The orientation of the line segments on the front is essential as it indicates the direction in which the front should move when acceptable elements are being generated. The generation process terminates when no more line segments are left in the front.

For the generation of adaptive meshes, in order to compare the quality of triangles, a criterion has to be defined combining the shape and size effects of the element. For meshes of uniform element size, the α-quality is one of the most convenient measures for judging the shape of a triangular element ABC[92]:

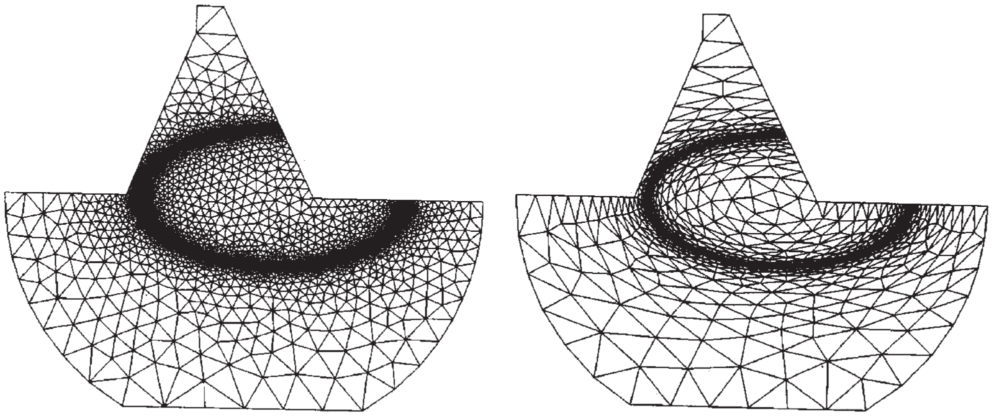
$$\alpha(ABC) = 2\sqrt{3}\frac{|AB \times AC|}{|AB|^2 = |BC|^2 = |CA|^2}$$

in which $2\sqrt{3}$ is a normalizing factor so that equilateral triangles will have a maximum α value of 1.

However, in a gradation mesh for adaptive analysis, the element size changes progressively, and the deviation of the element size from the required value should also be taken into account. In view of the element size effect, the following β parameter is proposed to judge the quality of a triangular element[96]:

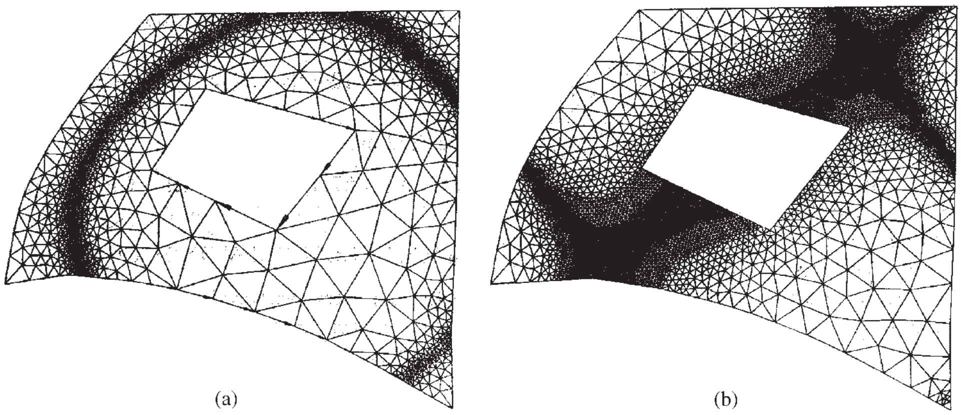
$$\beta = \lambda\alpha$$

where $\lambda = (s/\rho)(2 - s/\rho)$ is a factor accounting for the deviation from the required size ρ, which attains a maximum value of 1 when element size s equals ρ.



(a) (b)

Fig. 5 (a) Isotropic gradation mesh; (b) anisotropic gradation mesh



(a) (b)

Fig. 6 Meshes generated by the contour line method

The advancing front approach can also be employed for the generation of anisotropic meshes, which are beneficial for some fluid mechanics computations, and, with a suitable mapping technique, this can be applied to mesh generation over general curved surfaces[95]. A metric tensor is defined to calculate the length of a straight-line segment between two points, Frey & George[102] suggested the creation of internal points along the element edges of the current mesh. On the other hand, Lee[97] proposed a direct searching scheme into the open space from three points on the base edge. Fig. 5 shows some sample meshes generated by Lee's algorithms.

2.1.3 Mesh generation using contours

Mesh generation using contours is an early attempt for the generation of a triangular mesh of variable element size[103]. The contour lines of the node spacing function at suitable calculated levels provide the natural division lines of the problem domain into subregions, where finite elements of different sizes can be generated using any available general-purpose mesh generator. Fig. 6 shows two examples of a triangular mesh generated by this technique.

2.1.4 Coring technique

Mesh generation is achieved by constructing as many square elements as possible in the interior of the domain, leaving a relatively small region around the boundary which is to be discretized into triangular elements by a standard triangulation procedure[104]. As a result, only the quality of the triangular elements near the boundary will be affected by the irregularity of the domain. The main feature of this technique is to extract as much as possible from the interior of the domain to form square elements, as in an excavation process, hence the name 'coring technique'. The square elements can later be divided into triangular elements to form a pure triangular mesh. Fig. 7 shows two sample meshes with elements of different sizes generated by this method.

2.1.5 Quadtree technique

The quadtree technique, which belongs to the class of spatial decomposition methods, was originally employed as a means to approximate the object geometry[105]. The finite element mesh generation based on this technique was developed in the 1980s by Shephard and his group at Rensselaer[106]. By this method, a square grid containing the domain to be discretized is recursively subdivided until the desired resolution is reached. The subdivision can be done in

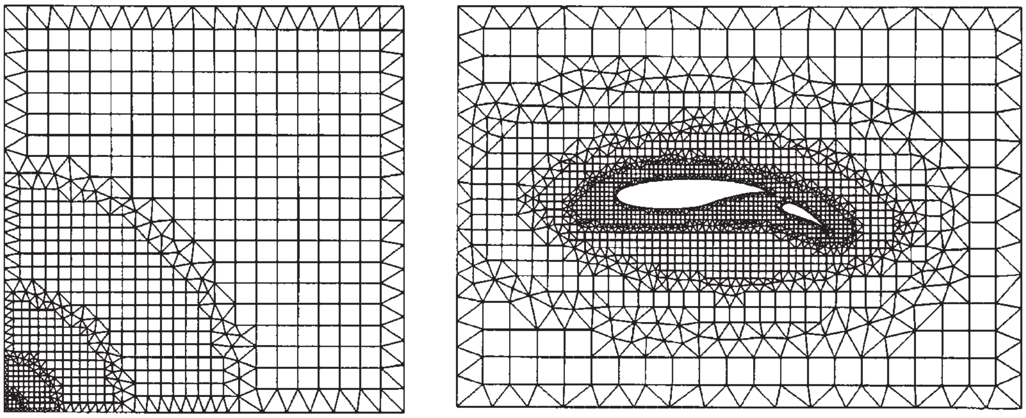


Fig. 7 Meshes generated by the coring technique

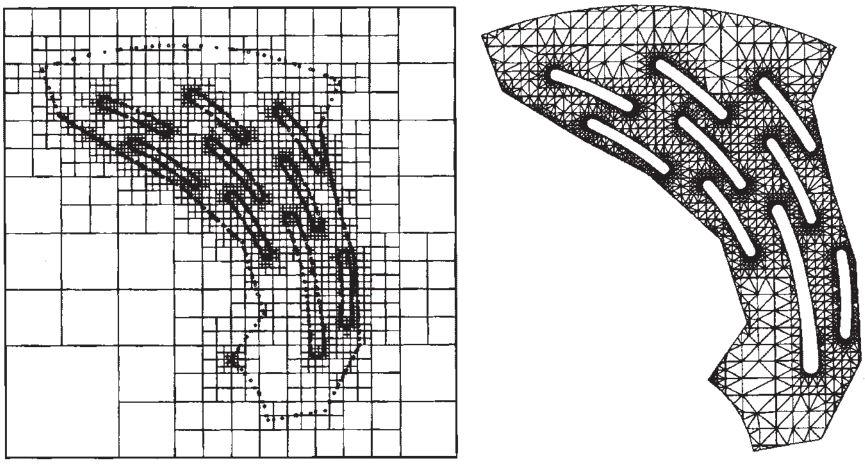


Fig. 8 Two stages of mesh generation by the quadtree technique

compliance with the complexity of the problem domain and/or in response to a given element size map for the generation of adaptive meshes. Irregular cells are created where squares intersect the object contour, often requiring a substantial amount of calculation for intersections. The modified quadtree is a technique for mesh generation, in which cuts by boundary segments at quarter points of a square are considered[107]. Two stages of mesh generation by this method are shown in Fig. 8.

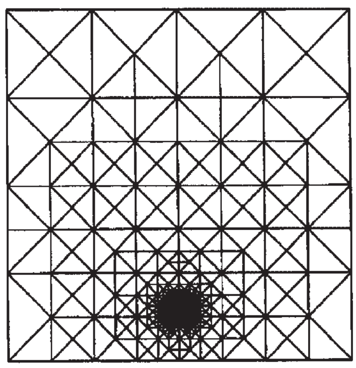


Fig. 9 Mesh generated by successive refinement

2.1.6 Refinement by subdivision (two-dimensional)

Generation of adaptive finite element meshes by refinement can be achieved by selective subdivision of triangles according to the specified nodal spacing function. Starting from a coarse mesh, elements can be subdivided until the desired refinement is reached. The most popular type of division scheme is bisection across the longest edge, for which the maximum and minimum angles of the resulting triangles are bounded[108]. Many refinement algorithms with various characteristics exist, and a comprehensive account can be found in a paper by Jones & Plassmann[109]. As subdivision of triangles is a local process which does not require too much computation, adaptive meshing by refinement in general is a very rapid process, although mesh topology and general element layout are limited by previous meshes. With some post-treatment optimization process, the quality of the elements could be drastically improved, resulting in a very attractive efficient adaptive mesh generation scheme. Fig. 9 shows a gradation mesh obtained by successive refinements.

2.2 Quadrilateral mesh

Generation of quadrilateral meshes by mapping[110] and drag method[111] are fairly common. However, only structured meshes are produced in general, which have difficulty in complying with the requirement of rapid change in element size of an adaptive mesh. As a result, adaptive meshes using quadrilateral elements are mainly unstructured, and so are more flexible in coping with the element size requirement.

Unstructured quadrilateral meshing algorithms can be grouped into two categories—direct and indirect methods. With the direct approach, quadrilaterals are generated directly on the surface without first going through the process of triangular meshing. As for the indirect approach, the physical domain is first meshed into triangles. Various algorithms are then employed to convert the triangles into quadrilaterals.

2.2.1 Direct method

The methods of directly generating quadrilateral meshes can be classified into two categories. The first are those that rely on some form of decomposition of the domain of interest into simpler regions, which allow a straightforward natural decomposition into quadrilaterals. The second category are those that make use of a moving front for direct placement of nodes and elements.

The quadtree technique proposed by Baehmann *et al.*[112] belongs to the first category, and is based on a spatial decomposition for quadrilateral meshing. After an initial decomposition of the two-dimensional space into a quadtree subdivision, taking account of the local features, quadrilaterals are fitted into the quadtree cells, and nodes are adjusted when necessary to conform to the domain boundary. Talbert[113] introduced another decomposition technique, in which the domain is recursively subdivided into simpler polygonal shapes. The resulting polygons comply with a number of templates, into which quadrilateral elements are generated. Chae & Jeong[114] and Nowottny[115] proposed improvements to Talbert's algorithm. Generation of quadrilateral meshes employing medial axis decomposition was introduced by Tam & Armstrong[116]. The medial axis can be considered as a series of lines generated from the mid-point of a maximal circle as it is rolled through the area. Having decomposed the area into simpler regions, quadrilateral elements are generated on each region. Joe[117] proposed a decomposition algorithm to divide a given two-dimensional domain into convex polygons. Using techniques formerly developed for Delaunay triangulation[118], Joe constructed a boundary-constrained quadrilateral mesh within each convex region.

Zhu *et al.*[119] were among the first to propose a quadrilateral meshing algorithm by means of an advancing front approach. Starting with an initial placement of nodes offsetting from a boundary edge, elements are formed by projecting edges towards the interior. Two triangles can be created using traditional advancing front method for triangles, which are then combined to form one single quadrilateral. The paving method introduced by Blacker & Stephenson[120] offers a way to form complete rows of quadrilaterals, starting from the boundary and working inwards. Methods of projection of nodes, handling of special geometric situations and intersection of opposing fronts are discussed. White & Kinney[121] proposed improvements to the paving algorithm, suggesting individual placement of elements rather than a complete row.

2.2.2 Indirect method

A simple means of indirect quadrilateral mesh generation is to divide each triangle in the mesh into three quadrilaterals by inserting a node at the centroid. This method guarantees an all-quadrilateral mesh, but a high number of irregular nodes are introduced into the mesh, resulting in poor element quality. Moreover, the boundary of the domain cannot be kept intact as mid-side nodes are introduced to each boundary segment. An alternative way is to combine pairs of adjacent triangles to form quadrilaterals, and the resulting mesh is a mixture of triangles and quadrilaterals. The method of combining triangles can be improved if some care is taken in the order in which the triangles are merged. In an effort to maximize the number of quadrilaterals, Lo[122] defined a shape measure for quadrilaterals in an algorithm that suggested several heuristic procedures for the sequence in which triangles are combined. The result is a quad-dominated mesh containing a minimal number of triangles. Lee & Lo[123] later proposed a frontal merging scheme which enhances Lo's strategy by including local triangle splitting. In addition, an advancing front approach is employed to divide the triangular mesh into meshed and unmeshed zones for systematic merging. An initial set of merging fronts is defined which includes all the boundary segments. Triangles are systematically combined at the front, advancing towards the interior of the triangular mesh. At an intermediate stage, the front is a collection of contour lines separating the quadrilateral already formed and the triangles yet to be combined. With this technique, it is possible to convert a given triangular mesh into an all-quadrilateral mesh, provided the number of segments on the boundary is even.

As the number and the position of the boundary nodes are not altered during the merging process, the integrity of the domain boundary can be guaranteed. Since the quadrilateral mesh is derived from a background triangular mesh, quadrilateral meshes with a nodal spacing compatible with the specified node spacing function can be expected, making it a very attractive for adaptive meshing. Since all

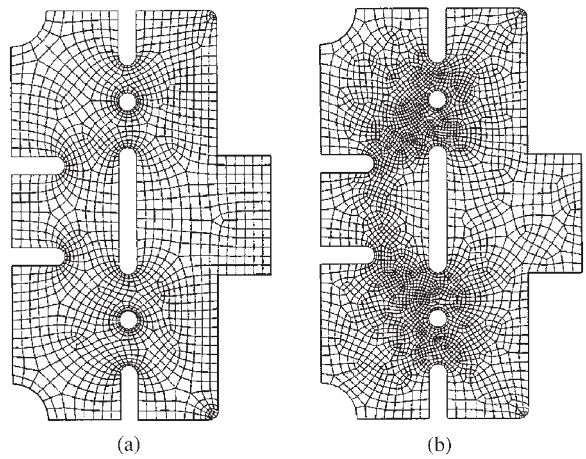


Fig. 10 Quadrilateral meshes by indirect methods: (a) Q-maps; (b) Lee's algorithm

operations are local, indirect methods enjoy the merit of being very fast. Global intersection checks are not necessary (as required by many direct methods). The drawback of the indirect method is that there are typically many irregular nodes left in the mesh, and there is no guarantee that the quadrilateral elements are in alignment with the domain boundary, a desirable feature for some applications and visualization. Irregular nodes can be reduced to a large extent by geometric and topological clean-up operations, hence improvement in the mesh quality.

A modification recently proposed by Owen *et al.*[124,125], known as quad-morphing(Q-morph) also utilizes an advancing front approach to convert triangles into quadrilaterals. This is basically a remeshing process, in which quadrilaterals are formed at the construction front while triangular elements are removed and modified. With this approach, local edge swaps are performed and additional nodes are introduced to ensure boundary alignment and orthogonality. Fig. 10 shows two quadrilateral meshes, one by Owen's algorithm and the other by Lee's algorithm.

3 Adaptive meshing on curved surfaces

3.1 Introduction

There is a growing demand for robust and efficient discretization algorithms for arbitrary curved surfaces into finite elements of variable sizes and shapes. The boundary of a physical object is generally made up of patches of curved surfaces, which can be typically represented by NURBS surfaces generated by a commercial CAD package. The surface discretization itself is a surface model of the object for rapid visualization, and it can also be used for finite element analysis or serve as input for a volumetric mesh generator. While many techniques that are used for planar mesh generation are still applicable for surface meshing, there are difficulties and different geometric characteristics over curved surfaces that require modifications/extension and special considerations. Although many surface mesh generation schemes are available, they can be broadly classified into either parametric mapping approaches or direct three-dimensional surface mesh generation.

3.2 Parametric mapping method

By this approach, the surface to be discretized is represented by a bivariate function such that any point on the three-dimensional surface is mapped to a two-dimensional parametric space[126–128]. The mesh generation process is carried out entirely on the parametric space by a two-dimensional general-purpose mesh generator. The final surface mesh is obtained by proper transformation of the mesh generated on the parametric space back to the three-dimensional space. This method gives reasonably good meshes for simple surfaces that are sufficiently smooth with respect to a planar domain. For the more complex curved surfaces, the resulting meshes are usually of poor quality, owing to element distortions induced by the mapping. An improved version of the mapping method is to first decompose the complex surfaces into a union of simple surfaces, and meshes are then generated on each of the sub-surfaces[95]. Such a modification does give much improved results; however, for general free-form surfaces with a lot of large variations of curvature, the problem of decomposing the surface into simpler smooth sub-surfaces may be just as hard as the mesh generation process. Hence, in order to obtain a finite element mesh satisfying the user specifications, it is necessary to control the shape and size of the elements in the parametric space. To generate a finite element mesh over curved surfaces in compliance with the nodal spacing requirements, we have to modify the mesh generation algorithm so that stretched or anisotropic elements generated on the two-dimensional parametric domain will map to well-shaped elements on the three-dimensional curved surface.

A method commonly employed in practice is to take advantage of the surface derivatives easily computed from NURBS surfaces. Borouchaki & George[129] proposed the use of a general anisotropic metric derived from the surface curvatures. The metric can be represented by a 2×2 matrix which can be used to transform vectors and distances in parametric space. For anisotropic meshing based on the Delaunay approach, the empty circle property effectively becomes an empty-ellipse property, and the meshing of analytical surfaces is achieved through a two-dimensional mesh generation process governed by an anisotropic metric. Without any additional effort, included also with the metric is the option to incorporate element size and stretched properties of

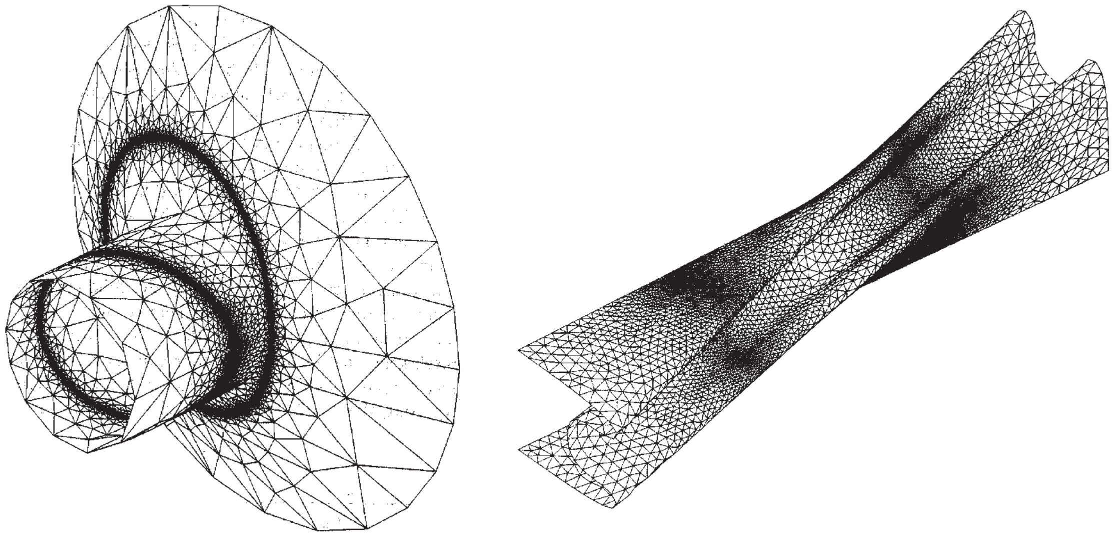


Fig. 11 Surface meshing by mapping of two-dimensional anisotropic meshes

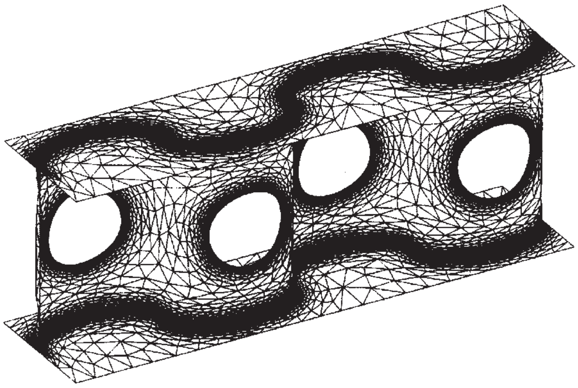


Fig. 12 Anisotropic surface mesh

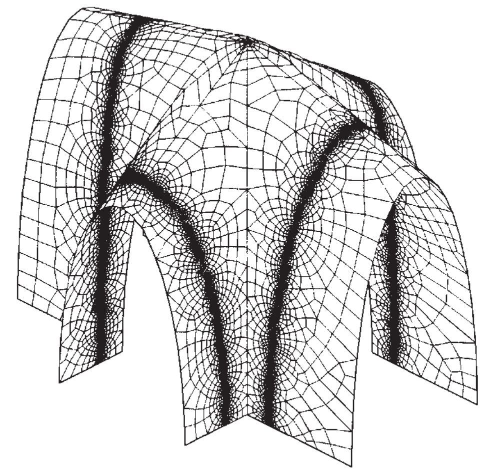


Fig. 13 Quadrilateral mesh by merging of triangles

the surface mesh. Hence, with a slight modification in the metric tensor, anisotropic meshes of variable element size can be easily constructed over curved surfaces through a mapping process. A similar approach to parametric Delaunay surface meshing is presented by Chen & Bishop[130]. Equivalent advancing front surface mesh generation algorithms which employ a metric derived from the first fundamental form of the surface are presented by Cuillere[131], Tristano *et al.*[132] and Lee[133]. Fig. 11 shows adaptive surface meshes generated by mapping techniques.

Much more complicated curved surfaces can be represented as a union of surface patches, each of which can be discretized by the mapping technique[95]. As mentioned before, it is interesting to note that adaptive anisotropic meshes over surfaces can be easily achieved by properly modifying the underlying metric used in the parametric space[133–135]. Fig. 12 shows anisotropic meshes on curved surfaces generated by a parametric mapping.

So far we have discussed generation techniques for unstructured triangular meshes of variable size and shape over general curved surfaces. Gradation quadrilateral meshes over curved surfaces can be obtained by conversion of surface triangulation with particular care taken to maintain correct surface curvatures. Quadrilateral-dominated meshes can be achieved by combining pairs of adjacent triangles[122,123]. Similar to a planar domain, all-quadrilateral meshes over curved surfaces can be generated by the systematic merging of the triangular mesh[136], as discussed in Section 2.2.2. Fig. 13 shows examples of quadrilateral meshes from the conversion of surface triangulation. The Q-morph scheme proposed by Owen can also be used to generate quadrilateral meshes on curved surfaces[137]. It can be considered as a remeshing process which converts a background triangulation into a quadrilateral mesh. This method is applicable to two-dimensional domains as well as over surfaces to generate gradation quadrilateral meshes, as shown in Fig. 14.

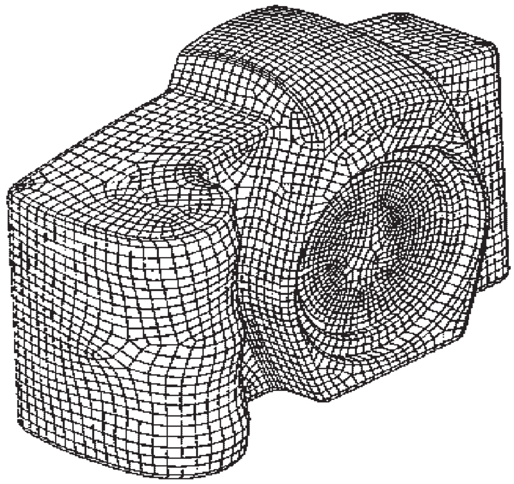


Fig. 14 Mesh generated by the Q-morph method

3.3 Direct mesh generation on surfaces

Direct three-dimensional surface mesh generation forms elements directly on the surface without the need for a parametric representation of the underlying geometry. In the case where a parametric representation is not available or where the surface parameterization is poor, direct three-dimensional surface mesh generators can be applied.

Lau & Lo[94,138] presented an advancing front scheme for the generation of triangular meshes on arbitrary three-dimensional surfaces. In this method surface normals and tangents are computed to determine the direction of the generation front. A significant number of surface projections are required to bring the new nodes generated back on the surface. Intersection checks are also required to make sure that triangles on the surface do not overlap. Direct mesh generation on surfaces will also be useful in situations of remeshing an existing surface triangulation of discrete elements for which surface parameterization is not readily available. If an adaptive refinement mesh is required for the current finite element model, defined in terms of element connections and nodal coordinates, direct three-dimensional meshing is the most suitable.

Direct three-dimensional surface mesh generation of quadrilateral elements is also possible. A direct three-dimensional implementation of the paving algorithm over curved surfaces was presented by Cass *et al.*[139]. Heuristic 'sticky space' is defined to detect intersection and overlapping quadrilaterals. Fig. 15 shows some triangular element meshes generated by a direct three-dimensional generation process.

4 Adaptive meshing for three-dimensional solids

4.1 Unstructured tetrahedral mesh generation

From the previous discussions, it appears that automatic mesh generation has reached such a mature stage that efficient algorithms are available to generate high-quality adaptive meshes on general two-dimensional domains and over arbitrary curved surfaces in a robust manner. However, when we look at the mesh generation for three-dimensional solid objects, we immediately notice that the problem gets much more complex, and many skills that work pretty well in two dimensions simply cannot be extended to a higher dimension. In two dimensions, the boundary-constrained mesh generation is almost a deterministic processes so that solutions are always guaranteed. In three dimensions, mesh generation algorithms are more iterative in nature. The fundamental difference between meshing a two-dimensional domain and a three-dimensional domain is that a two-dimensional boundary can always be meshed without the need for additional nodes. However, there are geometries in three dimensions which cannot be discretized without adding interior points[140], a twisted pentahedron with surface triangulation is a well-known example. Since there is no systematic way to decide where points should be inserted, analytical solutions are not available, leading to the development of iterative algorithms of heuristic nature for specific applications. The problem is made even more

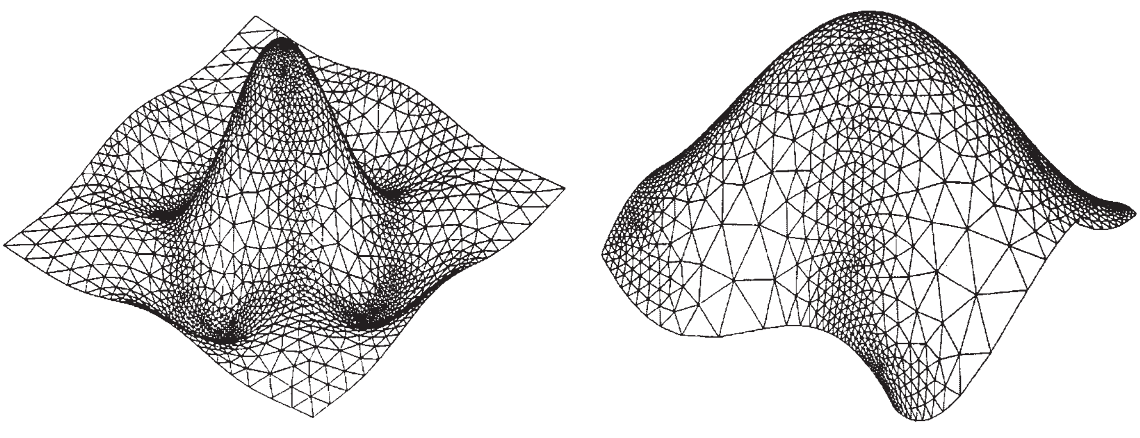


Fig. 15 Meshes by direct three-dimensional generation on surfaces

complicated when meshes of variable element size are required in an adaptive environment. Nevertheless, after years of dedicated research, many practical algorithms for meshing three-dimensional solid objects are quite reliable so that the integrity of the domain boundary can be maintained, unless extremely poor boundary conditions are encountered, characterized by the presence of many elongated surface facets with large aspect ratios between adjacent elements. The three popular techniques, namely the octree method, Delaunay triangulation and the advancing front approach again play an important role in three-dimensional mesh generation, and will be discussed in detail in the following sections. Other techniques such as the selective refinement, the medial surface method, the plastering method, the whisker weaving method and the H-morph algorithm, etc. for the generation of hexahedral meshes will be briefly described at the end of the paper.

4.1.1 Octree technique

The octree technique is a form of spatial decomposition scheme in which the object of interest is enclosed in a box of regular cubic cells which are progressively refined to capture the domain boundary or to satisfy certain element size requirements. The octree technique represents a three-dimensional object as a collection of cubes of variable sizes, more or less the same as the quadtree technique presents a planar domain in terms of square cells[54,141]. The modifications to the octree that are necessary as a mesh generation tool are similar to, but much more extensive than those used in the modified-quadtree mesh generation method.

One of the shortcomings of the quadtree or octree decomposition is the predefined orientation of the generation zone, which does not properly account for the preferential direction dictated by external and internal boundary parts. To reduce the number of elements needed to represent curved boundaries, in the modified-octree decomposition, the concept of a ‘cut octant’ is introduced. To maintain integer tree storage and to limit the number of cut octant cases to a manageable level, only the quarter and half points of an octant are used in the cutting process. This is a rather complicated process and has to be taken up on a case-by-case basis; the number of special cases required in a two-dimensional situation is 16, where as the same situation in three-dimensional increases to 4096 cases[53,142]. Other features associated with the modified-octree method are that a one-level transition rule has to be enforced to ensure a smooth change of element size and transition elements are to be used in other situations[143]. As the contact surfaces of neighbouring parts created by the modified-octree method are in general not compatible, mesh generation for complex domains through subdomain decomposition is not straightforward, as there is difficulty in matching the boundary of one subdomain with another. Variations of the modified-octree technique combined with other schemes such as Delaunay triangulation and a frontal process have also been proposed[144–147]. Nevertheless, the advantage of the octree technique is that it allows a rapid decomposition of the object into elements, and there are flexibilities in the degree of resolution in the representation of boundary geometrical features. This method may be most suitable for problems in which the physical solution is not sensitive to boundary geometrical details. Fig. 16 shows a tetrahedral mesh created by the modified-octree method.

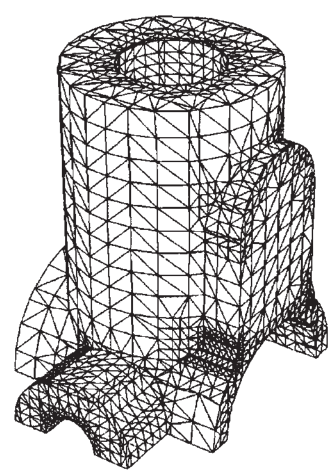


Fig. 16 Tetrahedral mesh created by the octree method

4.1.2 Three-dimensional Delaunay triangulation

In three dimensions, Watson’s algorithm[69] starts with a tetrahedron containing all points to be inserted, and new tetrahedra are formed as the points are entered one at a time. At a typical stage of the process, a new point is tested to determine which circumspheres of the existing tetrahedra contain the point. The associated tetrahedra are removed, leaving an insertion polyhedron containing the point. Edges connecting the new point to all triangular facets on the surface of the insertion polyhedron are created, defining tetrahedra which fill up the cavity. Combining these with the tetrahedra outside the insertion polyhedron produces a new Delaunay triangulation containing the newly added point. Triangulation is complete when all points are inserted and processed sequentially.

In many applications, there is a requirement that the triangular facets on the boundary surface be maintained. However, Delaunay triangulation of the boundary points does not always contain all the edges and triangular facets on the boundary surface. While in two dimensions, recovery of boundary edges will be guaranteed by swapping diagonals, there are cases in three dimensions where boundary triangulation cannot be defined without first inserting additional nodes. This phenomenon increases the complexity of the boundary recovery procedure in three

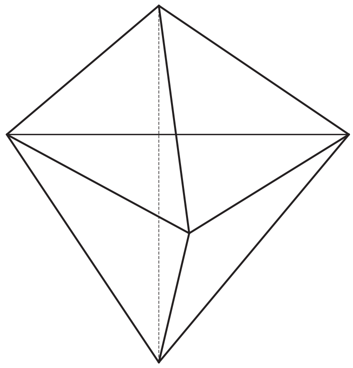


Fig. 17 2–3 swap of tetrahedral elements

dimensions. Two different methods have been proposed by George *et al.*[148] and Weatherill & Hassan[70], respectively, to deal with the surface recovery problem. In the first approach suggested by George and implemented in INRIA's GSH3D software[148], an edge joining two nodes is recovered by performing a series of tetrahedral transformations swapping two adjacent tetrahedra for three, as shown in Fig. 17. The 2–3 swap process effectively reduces the number of intersections of the line segment to be recovered with the triangular faces in the mesh by one for each swap. When there is no more intersection with the mesh, the proposed line segment is recovered. Sometimes no valid 2–3 swap can be defined to resolve an intersection, and additional nodes must be introduced to facilitate further element swaps until all intersections are removed. After all edges on the boundary are recovered, some boundary triangular facets may still be missing. In order to recover the triangular faces, element transformations are performed, mostly characterized by swapping three adjacent tetrahedra at an edge for two. More complex transformation or additional nodes may be required in the face recovery phase if simple transformations alone cannot resolve the situation.

The second method proposed by Weatherill[149] also involves an edge recovery phase and a face recovery phase. The main difference from George's approach is that, instead of attempting element transformations to recover edges and faces, nodes are inserted directly into the triangulation at positions where the boundary surface is intersected. This process temporarily introduces additional nodes to the boundary surface. Once the surface facets are formed, nodes that were inserted to facilitate the boundary recovery are deleted.

A strong point of Delaunay triangulation is that mesh generation is based on a sound theoretical background from which efficient and reliable algorithms can be formulated. The development in the boundary recovery technique further enhances the scope of applications of this method. However, while the Delaunay criterion provides a rule for connecting a given set of points, it does not suggest strategic positions for node insertion, except of course those nodes essentially on the boundary. While this seems to be a deficiency of the method, it allows a lot of freedom in defining node insertion strategies to generate meshes of different characteristics, including those used in adaptive analysis. The minimum triangulation with boundary integrity provides the basis for node insertion to create all kinds of meshes for various purposes.

In the mesh generator GSH3D developed by INRIA[150], interior nodes are inserted on positions along element edges. A list of candidate nodes is generated by marching along the existing internal edges of the mesh at a specified spacing. Nodes are then considered one by one, discarding nodes that would be too close to an existing node. This process can be done in a recursive manner until a background size function is satisfied. Another node insertion scheme suggested by Marcum & Weathers[59] and Frey *et al.*[87,88] is based on the advancing front approach. Each triangular face on the generation front is examined to determine the ideal location for a new fourth node at the interior of the existing Delaunay mesh. If a node is accepted, connection of this node to the existing elements is done by a standard point insertion kernel[151], and the front is updated accordingly. This method tends to generate elements in good alignment with the domain boundary, and they are usually of better quality compared with other point insertion schemes. Other point insertion strategies are available; a comprehensive account on these schemes and their characteristics is available[102]. As Delaunay triangulation can give rise to very thin degenerated elements, known as slivers, which may not be suitable for numerical analysis, optimization based on an appropriate element shape measure has to be applied to improve the overall quality of the mesh[152]. Fig. 18 shows adaptive meshes generated by the Delaunay triangulation method.

Extension to mesh generation governed by a general anisotropic metric is also possible[153,154]. Here, we have to pay attention to the two major aspects in a Delaunay triangulation process. First, in the creation of interior points, their strategic positions have to be determined by a length calculation based on the general metric so as to produce elements with anisotropic characteristics. Second, the empty-sphere criterion used in the Delaunay triangulation has to be revised as well. For each tetrahedron, we have to find a central point inside the tetrahedron which is of equal distance from the four vertices as measured by the given metric. The Delaunay criterion will hold if any point is at a distance from the centre greater than the distance between the centre and any one of the four vertices.

4.1.3 Advancing front method (three-dimensional)

First applied to planar domains and curved surfaces, the advancing front approach is equally effective for

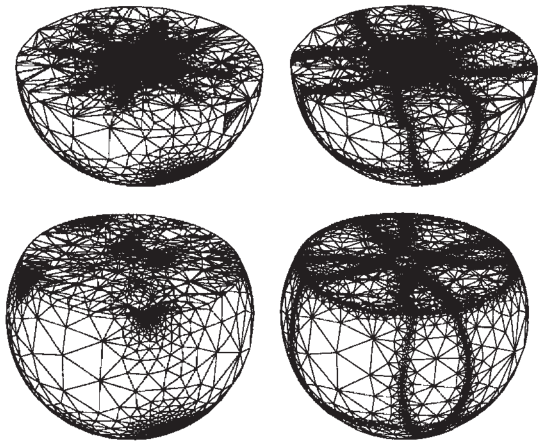


Fig. 18 Adaptive meshes by Delaunay triangulation

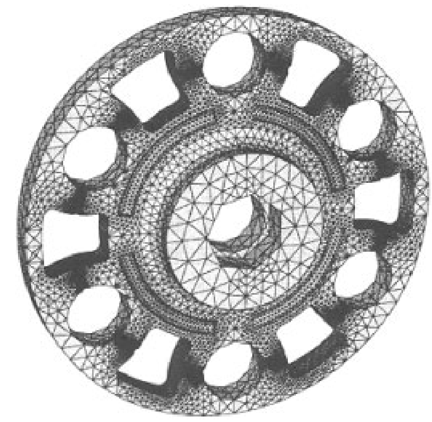


Fig. 19 Tetrahedral mesh generated by the advancing front approach

generating tetrahedral meshes in three dimensions. In three dimensions, the generation front is one or more closed surfaces of triangular facets, and the entire boundary surface can be taken as the initial front when meshing starts. Given a triangular facet on the front, an ideal location for a new fourth node is determined. The nodes on the generation front that form valid tetrahedra with the boundary facet under consideration are also noted. Similar to the two-dimensional case, a convenient shape measure for tetrahedra, the γ-coefficient, can be defined as follows[48]:

$$\gamma = \frac{72\sqrt{3} \times \text{volume of tetrahedron}}{(\text{sum of squares of edges})^{3/2}}$$

The algorithm selects the new fourth node or an existing node to form the best tetrahedron with a maximum γ value, and the front advances with the formation of the new element. Intersection checks are required to make sure no tetrahedron penetrates into the generation front during element construction. Meshing is completed when the generation front shrinks to zero, i.e. no more triangular facets remain. Adaptive meshes can also be generated with this method by defining a node spacing function to control element sizes as elements are created. In this case, both the placement of nodes and the element shape measure have to be modified to take into account the continuous change of element size[155]. Lohner & Onate[156,157] proposed using a coarse Delaunay mesh found on the selected boundary nodes, over which the size function can be easily interpolated. The major problem with the advancing front approach is that convergence is not always guaranteed for general complex three-dimensional domains. Nevertheless, convergence for three-dimensional domains of arbitrary geometry can usually be achieved with a sound node placing strategy. If there is no convergence, the volume has to be broken down into simpler parts before mesh generation, and the advancing front method is just such a suitable tool to handle meshing of subvolumes sharing a common boundary, as boundary integrity is observed in the meshing process. The stability and the efficiency of the method can be enhanced when it is coupled with Delaunay triangulation[56–59], especially when it is used for the generation of adaptive meshes. Fig. 19 shows a tetrahedral mesh generated by the advancing front approach.

4.1.4 Mesh generation by selective refinement

Instead of direct mesh generation in compliance with a node spacing function, an element refinement procedure in conjunction with a subsequent shape optimization process may be a better alternative to obtain an adaptive mesh as there are in general a lot of difficulties in generating gradation meshes over complex three-dimensional domains[140]. Two noticeable advantages offered by a selective refinement scheme as a means of mesh generation are the speed with which meshes are generated and the robustness in coping with the specified node spacing requirement. A selective refinement procedure for three-dimensional tetrahedral meshes based on successive bisection of edges controlled by a suitable element shape measure is proposed by Lo[158].
A typical subdivision process involves the division of all tetrahedra connected to an edge by introducing a node at the midpoint of the edge. The quality of the elements generated can be guaranteed if the subdivision is performed in a sequence according to the length of the line segments. The order of priority can be determined by a simple sorting process on all the line segments for which refinement is needed. The list of ordered line segments has to be updated from time to time to take account of the new line segments generated in the subdivision process. The element shape during and after the refinement process can be controlled by a convenient shape measure to guarantee mesh quality in each refinement[152]. Fig. 20 shows some typical adaptive meshes generated by the method of selective refinement.

The selective refinement can be easily extended to cover the case of generating anisotropic meshes. Refinement is done by dividing lines of the mesh based on the lengths calculated using the specified anisotropic metric at three points along the line under

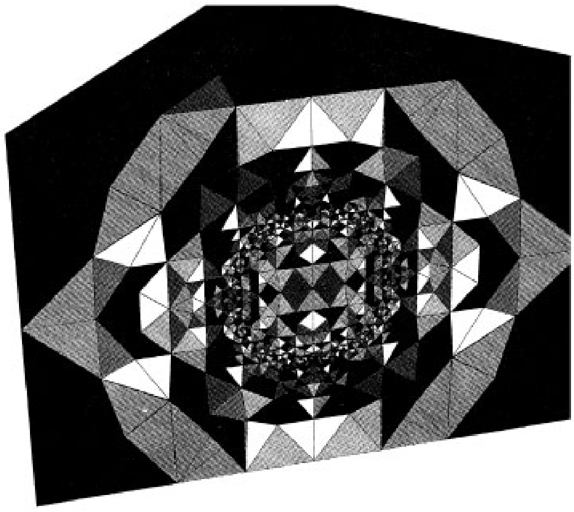


Fig. 20 Mesh generation by element subdivision

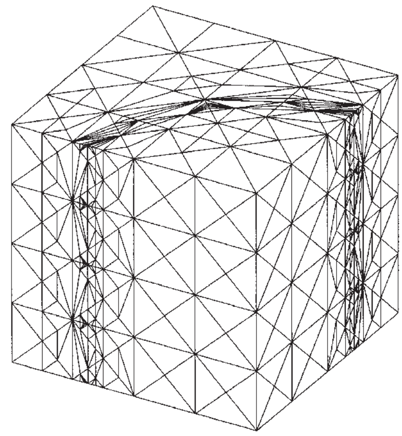


Fig. 21 Anisotropic tetrahedral mesh by refinement

consideration. The element size ρ along a particular direction specified by unit vector $\boldsymbol{v}$ is given by[159]

$$\rho = \lambda_1(\boldsymbol{v} \cdot \boldsymbol{u}_1)^2 = \lambda_2(\boldsymbol{v} \cdot \boldsymbol{u}_2)^2 = \lambda_3(\boldsymbol{v} \cdot \boldsymbol{u}_3)^2$$

where unit vectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$ and $\boldsymbol{u}_3$ are principal stretch directions (eigenvectors) of the given metric tensor $\boldsymbol{M}$ and λ_1, λ_2 and λ_3, respectively, are the required element size along these directions. The lines of the mesh are recursively divided until all edges are in compliance with the size requirement governed by the anisotropic metric $\boldsymbol{M}$. Fig. 21 shows a tetrahedral mesh generated by selective refinement governed by an anisotropic metric.

4.2 UNSTRUCTURED HEXAHEDRAL MESH GENERATION

Automatic unstructured mesh generation algorithms usually refer to the generation of tetrahedral meshes, as mapping techniques based on a regular grid will, in general, give rise to structured meshes. While most of the literature and software on unstructured meshes are about triangulation methods, there have been continuous research efforts on unstructured hexahedral mesh generation in recent years[20–46]. Nevertheless, these meshing techniques are only at an early stage of development; further improvements are expected and verifications are required before any conclusions can be drawn on the merits and drawbacks of the various schemes put forward. The situation is compounded by the fact that answers to many fundamental questions are still outstanding. For instance, what are the generally acceptable boundary conditions for a hexahedral meshing problem? Given a closed surface meshed into quadrilaterals, what can you say about the chance of having an all-hexahedral mesh? What criteria are there to judge the quality of a hexahedral mesh? What would hexahedral meshes of variable element sizes look like? Anyway, hexahedral meshing will remain as one of the most interesting topics in the world of mesh generation, and answers to some of these questions along with better meshing algorithms will emerge as time goes by. Unlike triangular and tetrahedral meshes, extension from a two-dimensional quadrilateral mesh to a three-dimensional hexahedral mesh is by no means straightforward, and very often a completely novel approach has to be adopted. Similar to quadrilateral mesh generation, there are both direct and indirect ways for the construction of unstructured hexahedral meshes.

4.2.1 Direct methods

There are currently four popular, yet distinct, approaches for the generation of unstructured hexahedral meshes, namely the grid based method, the medial surface method, the plastering method and the whisker weaving method.

The grid-based approach proposed by Schneiders[160] involves the fitting of a three-dimensional grid of hexahedral elements into the interior of a volume. More hexahedral elements have to be added to fill gaps between the regular grid and the boundary surface of the solid. Very much limited by the geometry of the voids, poor elements of irregular shapes are almost inevitable in the boundary fitting process. Hexahedral elements are in general not in good alignment with the domain boundary, and the resulting mesh is rather sensitive to the orientation of the interior grid. Owing to the use of a regular grid, element size at the interior of the volume will be approximately the same. Weiler and Schneiders[161,162] have made modifications that allow for significant changes in element size, based on an octree decomposition of the domain.

The medial surface method[163–165] involves an initial decomposition of the volume into subregions. As a direct extension of the medial axis method for quadrilateral meshing, the domain is subdivided by a set of medial surfaces, which can be thought of as surfaces generated from the midpoint of a maximal sphere allowed to roll through the volume. The decomposition of the volume by medial surfaces will define regions meshable by means of a mapping procedure. A set of templates for the expected topology of the regions formed by the medial surfaces are employed to fill the volume with hexahedral mesh. Linear programming is used to make sure element divisions match from one region to another. This method, while proving useful for some geometries has been less than reliable for meshing general three-dimensional objects. Robustness issues in generating medial surfaces, as well as providing for all cases regions readily defined for simple hexahedral meshing, seem to be rather difficult problems.

Plastering[166] is an attempt to extend the paving method to generate hexahedral meshes in three dimensions. By this method, elements are formed from the boundary quadrilateral facets and the construction advances towards the volume interior. A set of heuristic procedures are defined to determine the order in which elements should be created. Similar to the advancing front approach for tetrahedral mesh generation, the generation front for plastering is composed of quadrilaterals. Hexahedral elements are defined by projecting quadrilaterals on the generation front towards the interior of the volume. Of a more complicated nature compared with the classical frontal technique, intersecting faces have to be checked, and the problems of when and how to connect existing nodes or to seam faces have to be resolved. As the front advances, complex irregular internal voids may occur which in some cases cannot be filled with hexahedral elements. Existing elements already placed have to be removed or modified from time to time to cater for the formation of new elements towards the interior. The plastering algorithm has not yet been proved to be reliable for general applications.

The whisker weaving method [167,168] is based on the concept of a spatial twisting continuum (STC). Tautges *et al.*[167] described the STC as the dual of the hexahedral mesh represented by intersecting surfaces which bisect hexahedral elements in each direction. The principle behind the whisker weaving method is to first construct the STC from the boundary quadrilateral faces. By means of the STC, the hexahedral elements can be defined within the volume, using the STC as a guide. The intersection of the twisted planes with the volume will form closed loops on the surface which can be deduced from the boundary quadrilateral facets. The objective of the whisker weaving algorithm is to determine where the intersection of the twisted planes will occur within the volume. This is entirely based on topological consideration, and there are no actual geometrical calculations involved. Once a valid topological representation of the twist plane model has been established, hexahedra are formed inside the volume at places where three twist planes meet. The whisker weaving algorithm faces problems of unresolved topological situations and the formation of degenerated hexahedral elements of zero volume.

4.2.2 Indirect methods

Indirect methods involve the generation of hexahedral meshes by first triangulating the three-dimensional object and then converting the tetrahedra into hexahedra. A simple way to generate a hexahedral mesh from a tetrahedral mesh is to divide each tetrahedron into four hexahedra. The elements generated by this method are of very poor quality in general and not suitable for finite element analysis. The boundary integrity is also violated as nodes are introduced at the centre of each triangular facet. Another possibility of converting tetrahedra into hexahedral is to combine several tetrahedra to form hexahedra. As at least five tetrahedra are needed to form a hexahedron, a method that combines tetrahedra to form hexahedra would therefore need to look for combination of five or more tetrahedra to form a single hexahedral. This idea to date has not yet proved to be a viable option for hexahedral meshing.

Since most methods for all-hexahedral meshing seem to be less than robust, some researchers have proposed the generation of a mixed mesh of hexahedral and tetrahedral elements, along with any other elements such as pyramids and wedges to serve as transition elements as necessary. One approach introduced by Owen *et al.*[169] is to manually subdivide the volume into regions that could be meshed into hexahedra through a mapping process, or meshed into tetrahedral elements by a standard triangulation process. Pyramidal elements can be used to link up hexahedral and tetrahedral elements at the interface. Tuchinsky & Clark[170] have presented an algorithm which combines plastering and three-dimensional triangulation. Applying the plastering method, hexahedral elements are generated as far as possible into the volume. The remaining voids within the volume are meshed into tetrahedra by a general triangulation procedure. On the other hand, Min[171] presents a hexahedral-dominated meshing technique by making offsets from the boundary to form layers of hexahedra. After shells of hexahedral elements are peeled off, the shrunken volume is filled with tetrahedra.

The *H*-morph approach proposed by Owen & Saigal[172,173] is an indirect method operating on a triangulated volume in such a way that hexahedra are created while tetrahedra are removed and modified. It is essentially a remeshing process, generating hexahedra by techniques common to the plastering method; H-morph can be considered as a three-dimensional version of the Q-morph method. However, unlike the Q-morph procedure which produces all-quadrilateral meshes, H-morph does not always guarantee an all-hexahedral mesh for arbitrary geometry. It will give meshes ranging from hexagedral-dominated to all-hexahedral meshes, depending on the complexity of the domain and the quality of hexahedral elements required. Starting from a tetrahedral mesh of the volume, the algorithm

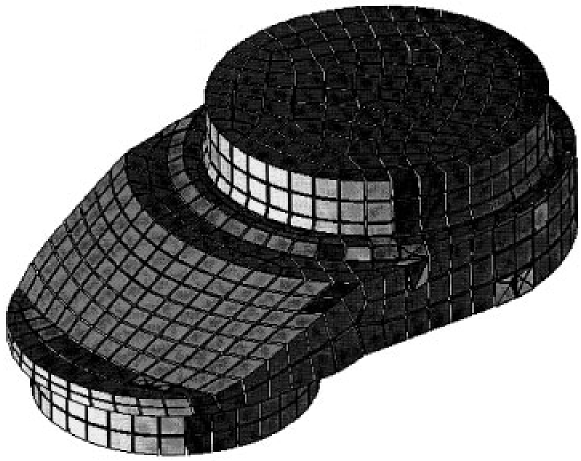


Fig. 22 Hexahedral-dominated mesh created by the H-morph technique

systematically transforms tetrahedra into hexahedra. At any instant in the transformation, a valid mesh of hexahedral and tetrahedral elements exists. At some stage when no reasonably shaped hexahedra can be formed within the volume, the procedure stops resulting in a hexahedral-dominated mesh. Employing the same concept of an advancing front approach, the H-morph algorithm operates more or less the same way as the plastering method. Similar to plastering, H-morph defines an initial front composed of quadrilaterals on the surface and systematically projects hexahedral elements towards the interior in an attempt to fill the volume completely with hexahedra. Operating within a meshed space, H-morph method avoids to a large extent the error-prone process of checking intersection and front closure inherent in plastering. Both stability and flexibility are enhanced in the H-morph procedure, as a valid finite element mesh is maintained throughout the generation process. Fig. 22 shows some meshes produced by the Q-morph technique. Before closing, I would like to mention the web-site http://www.andrew.cmu.edu/user/sowen/survey/index.html, in which a comprehensive list of commercial software for finite element mesh generation is presented.

References and recommended reading

*Papers of particular interest have been marked:
*Special interest
**Exceptional interest

[1] **Li XD & Wiberg NE.** A posteriori error estimate by element by element patch post-processing, adaptive analysis and L_2 norms. *Computers and Structures* 1994: **53**: 907–919.

* [2] **Lo SH & Lee CK.** On using meshes of mixed element types in adaptive finite element analysis. *Finite Elements in Analysis and Design* 1992: **11**: 307–336.

[3] **Coorevits P, Ladeveze P & Pelle JP.** Mesh optimization for problems with steep gradient. *Engineering with Computers* 1994: **11**: 129–144.

[4] **Choi CK & Park YM.** An adaptive h-refinement using transition element for plate bending problems. *International Journal for Numerical Methods in Engineering* 1992: **35**: 145–163.

** [5] **Zienkiewicz OC & Zhu JZ.** Error estimates and adaptive refinement for plate bending problems. *International Journal for Numerical Methods in Engineering* 1992: **35**: 2839–2853.

[6] **Burkley VJ & Bruch Jr. JC.** Adaptive error analysis in seepage problem. *International Journal for Numerical Methods in Engineering* 1991: **31**: 1333–1356.

* [7] **Zienkiewicz OC, Liu YC & Huang GC.** Error estimation and adaptivity in flow formulation for forming problems. *International Journal for Numerical Methods in Engineering* 1988: **25**: 23–42.

[8] **Fish J.** The s-version of the finite element method. *Computers and Structures* 1992: **43**: 539–547.

[9] **Okstad KM & Mathisen KM.** Towards automatic adaptive geometrically non-linear shell analysis. Part I: Implementation of an h-adaptive mesh refinement procedure. *International Journal for Numerical Methods in Engineering* 1994: **37**: 2657–2678.

[10] **Kodiyalam S & Parthasarathy VN.** Optimized/adapted finite element for structural shape optimization. *Finite Elements in Analysis and Design* 1992: **12**: 1–11.

* [11] **Peraire J, Peiro J, Formaggia L, Morgan K & Zienkiewicz OC.** Finite element Euler computations in three dimensions. *International Journal for Numerical Methods in Engineering* 1988: **26**: 2135–2159.

* [12] **Lee CK & Lo SH.** Automatic adaptive refinement finite element procedure for 3D stress analysis. *Finite Elements in Analysis and Design* 1997: **25**: 135–166.

[13] **Ohtsudo H & Kitamura M.** Numerical investigation of element-wise a posteriori error estimation in two and three dimensional elastic problems. *International Journal for Numerical Methods in Engineering* 1992: **34**: 969–977.

[14] **Baehmann PL, Shephard MS & Flaherty JE.** A posteriori error estimation for triangular and tetrahedral quadratic elements using interior residuals. *International Journal for Numerical Methods in Engineering* 1992: **34**: 979–996.

[15] **Pressburger Y & Perucchio R.** A self-adaptive FE system based on recursive spatial decomposition and multigrid analysis. *International Journal for Numerical Methods in Engineering* 1995: **38**: 1399–1421.

[16] **Schietze KT, Shiakolas PS, Muthukrishnan SN, Nambiar RV & Lawrence KL.** A study of adaptively remeshed finite element problems using higher order tetrahedra. *Computers and Structures* 1995: **54**: 270–288.

* [17] **Babuska I & Szabo S.** On the rates of convergence of the finite element method. *International Journal for Numerical Methods in Engineering* 1982: **18**: 323–341.

[18] **Babuska I & Li L.** The problem of plate modeling: Theoretical and computational results. *Computer Methods in Applied Mechanics and Engineering* 1992: **100**: 249–273.

* [19] **Lee CK & Lo SH.** An automatic adaptive refinement finite element procedure for 2D elastostatic analysis. *International Journal for Numerical Methods in Engineering* 1992: **35**: 1967–1989.

[20] **Liu SS & Rajit G.** Automatic hexahedral mesh generation by recursive convex and sweep volume decomposition. *Proceedings of the 6th International Meshing Roundtable*, Sandia National Laboratories, 1997: 217–231.

[21] **Schneiders R, Schindler R & Weiler F.** Octree-based generation of hexahedral element meshes. *Proceedings of the 5th International Meshing Roundtable*, Sandia National Laboratories, 205–216, October 1996.

[22] **Melander DJ, Tautges JT & Benzley SE.** Generation of multi-million element meshes for solid model-based geometries: The Dicer Algorithm. AMD-Vol. 220, Trends in Unstructured Mesh Generation, *ASME*, 131–135, July 1997.

[23] **Pavlakos CJ, Jones JS & Mitchell SA.** An immersive environment for exploration of CUBIT meshes. *Proceedings of the 6th International Meshing Roundtable*, Sandia National Laboratories, 47–48, October 1997.

[24] **Tautges TJ, Liu SS, Liu YL, Kraftcheck J & Rajit G.** Feature recognition applications in mesh generation. AMD-Vol. 220, Trends in Unstructured Mesh Generation, *ASME*, 117–121, July 1997.

[25] **Dhondt, Guido D.** Unstructured 20-node brick element meshing. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 369–376, October 1999.

[26] **Kraft P.** Automatic remeshing with hexahedral elements: Problems, solutions and applications. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 357–367, October 1999.

[27] Labbe P, Dompierre J, Guibault F & Camarero R. On element shape measures for mesh optimation. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[28] Lu Y, Gadh R & Tautges TJ. Volume decomposition and feature recognition for hexahedral mesh generation. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 269–280, October 1999.

[29] Meshkat, Sia & Dafna Talmor. Generating a mixed mesh of hexahedra, pentahedra and tetrahedra from an underlying tetrahedral mesh. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

* **[30] Owen SJ.** Constrained triangulation: application to hex-dominant mesh generation. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 31–41, October 1999.

* **[31] Owen SJ & Saigal S.** H-morph: An indirect approach to advancing front hex meshing. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[32] Benzley SE, Perry E, Merkley K, Clakr B & Sjaardema G. A comparison of all-hexahedral and all-tetrahedral finite element meshes for elasto-plastic analysis. *Proceedings of the 4th International Meshing Roundtable*, Sandia National Laboratories, 179–191, October 1995.

[33] Sheffer A, Etzion M, Rappoport A & Bercovier M. Hexahedral meshing of non-linear volumes using Voronoi faces and edges. *2nd Symposium on Trends in Unstructured Mesh generation*, University of Colorado, Boulder, August 1999.

[34] Shephard J, Benzley S & Mitchell S. Interval assignment for volume and holes. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[35] Tautges TJ. On automating the generation of hex meshes for assembly geometries. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[36] Thompson, David S & Soni BK. Generation of quad and hex dominant semistructured meshes using an advancing layer scheme. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 171–178, October 1999.

[37] Trease HE & Barrett GL. Three-dimensional, parallel, hybrid grid generation for ASCI program. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[38] Vassberg JC. Multi-block mesh extrusion driven by a globally elliptic system. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[39] Wada Y, Yoshimura S & Yagawa G. Intelligent local approach for automatic hexahedral mesh generation. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[40] White DR & Tautges TJ. Automatic scheme selection for Toolkit hex meshing. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[41] Folwell NT & Mitchell SA. Reliable Whisker Weaving via curve contraction. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laborities, 365–378, October 1998.

[42] Leland RW, Melander DJ, Meyers RW, Mitchell SA & Tautges TJ. The Geode algorithm: Combining hex/tet plastering, dicing and transition elements for automatic, all-hex mesh generation. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, 515–521, October 1998.

[43] Li H & Cheng G. New method for graded mesh generation of all hexahedral finite elements. Numerical Grid Generation in Computational Field Simulations. *Proceedings of the 6th International Conference*, University of Greenwich, 1031–1038, July 1998.

[44] Meyers RJ, Tautges TJ & Tuchinsky PM. The hex-tet dominant meshing algorithm as implemented in CUBIT. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, 151–158, October 1998.

[45] Mitchell SA. The all-hex geode—template for conforming a diced tetrahedral mesh to any diced hexahedral mesh. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, 295–305, October 1998.

[46] Muller-Hannemann M. Hexehedral mesh generation by successive dual cycle elimination. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, 365–378, October 1998.

* **[47] Lo SH.** Delaunay triangulation of non-convex planar domains. *International Journal for Numerical Methods in Engineering* 1989: **28**: 2695–2707.

* **[48] Lo SH.** Volume discretization into tetrahedra—II. 3D triangulation by advancing front approach. *Computers and Structures* 1991: **39**(5): 501–511.

[49] Blacker TD, Jung J & Witkowski WR. An adaptive finite element technique using element equilibrium and paving. *The American Society of Mechanical Engineers*, November 1990.

[50] Stephenson MB, Canann SA & Blacker TD. Plastering: A new approach to automated, 3D hexahedral mesh generation—Progress Report I, *Sandia Report No. 89–2192*, Sandia National Laboratories, Albuquerque, N.M., 1990.

[51] Blacker TD & Stephenson MB. Paving: a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1992: **32**: 811–847.

* **[52] Blacker TD & Meyers RJ.** Seams and wedges in plastering: a 3D hexahedral mesh generation algorithm. *Engineering with Computers* 1993: **9**: 83–93.

* **[53] Yerry MA & Shephard MS.** Automatic three-dimensional mesh generation by the modified-octree technique. *International Journal for Numerical Methods in Engineering* 1984: **20**: 1965–1990.

* **[54] Shephard MS & Georges MK.** Automatic three-dimensional mesh generation by the finite octree technique. *International Journal for Numerical Methods in Engineering* 1991: **32**: 709–749.

* **[55] Owen SJ.** A survey of unstructured mesh generation technology. *Proceedings of the 7th International Meshing Roundtable*, Sandia National Laboratories, 239–267, October 1998.

[56] Merriam ML. An efficient advancing front algorithm for Delaunay triangulation. *AIAA*, **91**–0792, 1991.

[57] Mavriplis DJ. An advancing front Delaunay triangulation algorithm designed for robustness. *ICASE report*, **92**–49, 1992.

[58] Fleischmann P & Siegfried S. Three-dimensional Delaunay mesh generation using a modified advancing front approach. *Proceedings of the 6th International Meshing Roundtable*, Sandia National Laboratories, 267–278, 1997.

[59] Marcum DL & Weatherill NP. Unstructured grid generation using iterative point insertion and local reconnection. *AIAA Journal* 1995: **33**(9): 1619–1625.

* **[60] Ho-Le K.** Finite element mesh generation methods: A review and classification. *Computer Aided Design* 1988: **20**: 27–38.

[61] Frey WH. Selective refinement: A new strategy for automatic node placement in graded triangular meshes. *International Journal for Numerical Methods in Engineering* 1987: **24**: 2183–2200.

[62] Delaunay B. Sur la sphere vide. *Bulletin, Académie des Sciences URSS*. 1934: 793–800.

[63] Cavendish JC, Field DA & Frey WH. An approach to automatic three-dimensional finite element mesh generation. *International Journal for Numerical Methods in Engineering* 1985: **21**: 329–347.

[64] George PL, Hecht F & Saltel E. Fully automatic mesh generator for 3D domains of any shape. *Impact of Computer Science in Engineering* 1990: **2**: 187–218.

[65] Wright JP & Jack AG. Aspect of three-dimensional constrained Delaunay meshing. *International Journal for Numerical Methods in Engineering* 1994: **37**: 1841–1861.

* **[66] Lo SH.** Delaunay triangulation of non-convex planar domains. *International Journal for Numerical Methods in Engineering* 1989: **28**: 2695–2707.

[67] George PL & Hermeline F. Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra. *International Journal for Numerical Methods in Engineering* 1992: **33**: 975–995.

[68] Lawson CL. Software for C1 surface interpolation. *Mathematical Software III* 1977: 161–194.

[69] Watson DF. Computing the Delaunay Tessellation with application to Voronoi polytopes. *The Computer Journal* 1981: **24**(2): 167–172.

* **[70] Weatherill NP & Hassan O.** Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints. *International Journal for Numerical Methods in Engineering* 1994: **37**: 2005–2039.

[71] Weatherill NP, Said R & Morgan K. The construction of Large unstructured grid by parallel Delaunay grid generation. Numerical Grid Generation in Computational Field Simulations. *Proceedings of the 6th International Conference*, University of Greenwich, 53–73, July 1998.

[72] Baker TJ. Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation. *Engineering with Computers* 1989: **5**: 161–175.

[73] Baker TJ & Vassberg JC. Tetrahedral mesh generation and optimization. Numerical Grid Generation in Computational Field simulations. *Proceedings of the 6th International Conference*, University of Greenwich, 337–349, 1998.

[74] **George PL & Hermeline F.** Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra. *International Journal for Numerical Methods in Engineering* 1992: **33**: 975–995.

[75] **George PL.** Automatic 3D mesh generation conforming a prescribed size map, Numerical Grid Generation in Computational Field Simulations. *Proceedings of the 6th International Conference*, University of Greenwich, 21–32, July 1998.

[76] **George PL.** Tet meshing: construction, optimization and adaptation. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 133–141, October 1999.

[77] **Bowyer A.** Computing Dirichlet tessellation. *The Computer Journal* 1981: **24**: 162–166.

[78] **Weatherill NP.** The integrity of geometrical boundaries in the 2-dimensional Delaunay triangulation. *Communication in Numerical Methods in Engineering* 1990: **6**: 101–109.

[79] **Hermeline F.** Triangulation automatique d'un polyhedre en dimension N. *Rairo, Analyse Numerique* 1982: **16**(3): 211–242.

[80] **Holmes DG & Snyder DD.** The generation of unstructured triangular meshes using Delaunay triangulation. Numerical grid generation in computational fluid mechanics '88, *Miami* 1988: 643–652.

[81] **Chew LP.** Constrained Delaunay triangulation. *Algorithmica* 1989: **4**: 97–108.

[82] **George PL.** Improvement on Delaunay based 3D automatic mesh generator. *Finite Elements in Analysis and Design* 1997: **25**: 297–317.

[83] **Weatherill NP.** The generation of unstructured grids using Dirichlet tessellation, *MAE report no. 1715*, Princeton University, 1985.

[84] **Baker TJ.** Three dimensional mesh generation by triangulation of arbitrary point sets. *Proceedings of the AIAA 8th Computational Fluid Dynamics conference*, Honolulu, HI, USA, **87**–1124, 1986.

[85] **Mavriplis DJ.** Adaptive mesh generation for viscous flows using Delaunay triangulation. *Journal of Computational Physics* 1990: **90**(2): 272–291.

[86] **Rebay S.** Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer/Watson algorithm. *Journal of Computational Physics* 1993: **106**: 125–138.

[87] **Frey PJ, Bourochaki H & George PL.** Delaunay tetrahedralization using an advancing front approach. *Proceedings of the 5th International Meshing Roundtable*, Pittsburg, PA, 31–43, 1996.

* [88] **Frey PJ, Borouchaki H & George PL.** 3D Delaunay mesh generation coupled with an advancing front approach. *Computer Methods in Applied Mechanics and Engineering* 1998: **157**: 115–131.

* [89] **Borouchaki H, George PL, Hecht F, Laug P & Saltel E.** Delaunay mesh generation governed by metric specifications. Part I: Algorithms. *Finite Elements in Analysis and Design* 1997: **25(**1–2**)**: 61–83.

[90] **Borouchaki H, George PL & Mohammadi B.** Delaunay mesh generation governed by metric specifications. Part II: Application examples. *Finite Elements in Analysis and Design* 1997: **25**: 85–109.

* [91] **Lo SH & Lee CK.** Generation of gradation meshes by the background grid technique. *Computers and Structures* 1994: **50**(1): 21–32.

* *[92] **Lo SH.** A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering* 1985: **21**: 1403–1426.

* [93] **Zhu JZ, Zienkiewicz OC, Hinton E & Wu J.** A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1991: **32**: 849–866.

* [94] **Lau TS & Lo SH.** Finite element mesh generation over analytical curved surfaces. *Computers and Structures* 1996: **59**(2): 301–309.

* [95] **Lee CK & Hobbs RE.** Automatic adaptive finite element mesh generation over rational B-spline surfaces. *Computers and Structures* 1998: **69**: 577–608.

[96] **Lo SH.** Generation of high-quality gradation finite element mesh. *Engineering Fracture Mechanics* 1992: **41**(2): 191–202.

[97] **Lee CK.** Automatic adaptive mesh generation using metric advancing front approach. *Engineering Computations* 1999: **16**(2): 230–263.

[98] **Lohner R & Parikh P.** Three-dimensional grid generation by the advancing front method. *International Journal for Numerical Methods in Fluids* 1988: **8**: 1135–1149.

[99] **Lohner R, Camberos J & Merriam M.** Parallel unstructured grid generation. *Computer Methods in Applied Mechanics and Engineering* 1992: **95**: 343–357.

[100] **Lohner R.** Extension and improvements of the advancing-front grid generation technique. *Communications in Numerical Methods in Engineering* 1996: **12**: 683–702.

[101] **Lohner R.** Automatic unstructured grid generators. *Finite Elements in Analysis and Design* 1997: **25**: 111–134.

[102] **Frey PJ & George PL.** *Maillage, Application aux Elements Finis*. Hermes Paris: 1999.

[103] **Lo SH.** Automatic mesh generation and adaptation by using contours. *International Journal for Numerical Methods in Engineering* 1991: **31**: 689–707.

[104] **Lo SH & Lau TS.** Generation of hybrid finite element mesh. *Microcomputers in Civil Engineering* 1992: **7**: 235–241.

[105] **Knuth DE.** *The Art of Computer Programming*. 2nd edn. Reading, MA: Addison-Wesley. 1975.

[106] **Shephard MS.** Approaches to the automatic generation and control of finite element meshes. *Applied Mechanics Reviews* 1988: **41**: 169–185.

[107] **Yerry MA & Shephard MS.** A modified quadtree approach to finite element mesh generation. *IEEE Journal* 39–46, Jan./Feb. 1983.

[108] **Rivara MC.** New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International Journal for Numerical Methods in Engineering* 1997: **40**: 3313–3324.

[109] **Jones MT & Plassmann PE.** Adaptive refinement of unstructured finite-element meshes. *Finite Elements in Analysis and Design* 1997: **25**: 41–60.

[110] **Zienkiewicz OC & Phillips DV.** An automatic mesh generation scheme for plane and curved surfaces isoparametric coordinates. *International Journal for Numerical Methods in Engineering* 1971: **3**: 519–528.

[111] **Park S & Washam CJ.** Drag method as a finite element mesh generation scheme. *Computers and Structures* 1979: **10**: 343–346.

[112] **Baehmann PL, Wittchen SL, Shephard MS, Grice KR & Yerry MA.** Robust geometrically-based automatic two-dimensional mesh generation. *International Journal for Numerical Methods in Engineering* 1987: **24**: 1043–1078.

[113] **Talbert JT & Parkinson AR.** Development of an automatic two-dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definitions. *International Journal for Numerical Methods in Engineering* 1991: **29**: 1551–1567.

[114] **Chae SW & Jeong JH.** Unstructured surface meshing using operators. *Proceedings of the 6th International Meshing Roundtable*, 281–291, 1997.

[115] **Dietrich N.** Quadrilateral mesh generation via geometrically optimized domain decomposition. *Proceedings of the 6th International Meshing Roundtable*, 309–320, 1997.

[116] **Tam TKH & Armstrong CG.** 2D Finite element mesh generation by medial axis subdivision. *Advances in Engineering Software* 1991: **13**: 313–324.

* [117] **Joe B.** Quadrilateral mesh generation in polygonal regions. *Computer Aided Design* 1995: **27**: 209–222.

[118] **Joe B.** Delaunay triangulation meshes in convex polygons. *SIAM Journal of Scientific and Statistical Computation* 1986: **7**: 514–539.

[119] **Zhu JZ, Zienkiewicz OC, Hinton E & Wu J.** A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 1991: **32**: 849–866.

[120] **Blacker TD & Stephenson MB.** Paving: A new approach to automatic quadrilateral mesh generation. International *Journal for Numerical Methods in Engineering* 1991: **32**: 811–847.

[121] **White DR & Kinney P.** Redesign of the Paving algorithm: Robustness enhancements through element by element meshing. *Proceedings of the 6th International Meshing Roundtable*, Sandia National Laboratories, 323–335, 1997.

[122] **Lo SH.** Generating quadrilateral elements on plane and over curved surfaces. *Computers and Structures* 1989: **31**: 421–426.

[123] **Lee CK & Lo SH.** A new scheme for the generation of a graded quadrilateral mesh. *Computers and Structures* 1994: **52**(5): 847–857.

[124] **Owen SJ, Staten ML, Canann SA & Saigal S.** Advancing front quad meshing using local triangle transformations. *Proceedings of the 7th International Meshing Roundtable*, 1998.

[125] **Owen SJ, Staten ML, Canann SA & Saigal S.** Q-Morph: An indirect approach to advancing front quad meshing. *International Journal for Numerical Methods in Engineering* 1999: **44**: 1317–1340.

[126] **Borouchaki H & Frey PJ.** Optimization tools for adaptive surface meshing, AMD-Vol. 220 Trends in Unstructured Mesh Generation, *ASME 81–88*, July 1997.

[127] **Borouchaki H, Laug P & George PL.** About parametric surface meshing. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

[128] **Borouchaki H, Frey PJ & George PL.** Unstructured triangular-quadrilateral mesh generation. Application to surface meshing. *Proceedings of the 5th*

International Meshing Roundtable, Sandia National Laboratories, 229–242, October 1996.

[129] Borouchaki H & George PL. Maillage des surfaces parametrique. Partie I: Aspects theorique, *Computer Rendus, Académie des Sciences*, Paris, t324, Series I, 833–837, 1997.

[130] Hao C & Jonathan B. Delaunay triangulation for curved surfaces. *Proceedings of the 6th International Meshing Roundtable*, 115–127, 1997.

[131] Cuilliere JC. An adaptive method for automatic triangulation of 3D parametric surfaces. *Computer-aided Design* 1998: **30**(2): 139–149.

[132] Tristano JR, Owen SJ & Canann SA. Advancing front mesh generation in parametric space using a Riemannian surface definition. *Proceedings of the 7th International Meshing Roundtable*.

* **[133] Lee CK.** Automatic metric advancing front triangulation over curved surfaces. *Engineering Computations* 2000: **17**(1): 48–74.

[134] Frey PJ & Borouchaki H. Triangulation des surfaces implicites, Computer Rendus, Académie des Sciences, Paris, t325, Series I, 101–106, 1996.

[135] Frey PJ & Borouchaki H. Geometric evaluation of finite element surface meshes. *Finite Elements in Analysis and Design* 1998: **31**: 33–53.

* **[136] Lau TS, Lo SH & Lee CK.** Generation of quadrilateral mesh over analytical curved surfaces. *Finite Elements in Analysis and Design* 1997: **27**: 251–272.

* **[137] Owen SJ.** Non-simplical unstructured mesh generation, *Ph.D. thesis*, Carnegie Mellon University, Pittsburg, PA, USA, 1999.

[138] Lo SH & Lau TS. Mesh generation over curved surfaces with explicit control on discretization error. *Engineering Computations* 1998: **15**(3): 357–373.

[139] Cass RJ, Benzley SE, Meyers RJ & Blacker TD. Generalized 3D Paving: An automated quadrilateral surface mesh generation algorithm. *International Journal for Numerical Methods in Engineering* 1996: **39**: 1475–1489.

[140] Ruppert J & Seidel R. On the difficulty of triangulating three-dimensional nonconvex polyhedra. *Discrete Computational Geometry*1992: **7**: 227–253.

[141] Shephard MS, Guerinoni F, Flaherty JE, Ludwig RA & Baehmann PL. Adaptive solution of the Euler equations using finite quadtree and octree grids. *Computers and Structures* 1988: **30**: 327–336.

[142] Schneiders R. An algorithm for the generation of hexahedral elements meshes based on an octree technique. *Proceedings of the 6th International Meshing Roundtable*, Sandia National Laboratories, 183–194, 1997.

[143] Carey GF, Sharma M & Wang KC. A class data structures for 2D and 3D adaptive mesh refinement. *International Journal for Numerical Methods in Engineering* 1984: **26**: 1965–1990.

[144] Sapidis N & Perucchio R. Combining recursive spatial decomposition and domain Delaunay tetrahedralization for meshing arbitrary shaped curved solid models. *Computer Methods in Applied Mechanics and Engineering* 1993: **108**: 282–302.

[145] Schreoder WJ & Shephard MS. Geometry-based fully automatic mesh generation and the Delaunay triangulation. *International Journal for Numerical Methods in Engineering* 1988: **26**: 2503–2525.

[146] Schroeder WJ & Shephard MS. A combined octree/Delaunay method for fully automatic 3D mesh generation. *International Journal for Numerical Methods in Engineering* 1990: **29**: 37–55.

[147] MacMorris H & Kallideris Y. Octree-advancing front method for generation of unstructured surface and volume meshes. *AIAA Journal* 1997: **35**(6): 976–984.

** **[148] George PL, Hecht F & Saltel E.** Automatic mesh generator with specified boundary. *Computer Methods in Applied Mechanics and Engineering* 1991: **92**: 269–288.

[149] Weatherill NP. The reconstruction of boundary contours and surfaces in arbitrary unstructured triangular and tetrahedral grids. *Engineering Computations* 1996: **13**(8): 66–81.

[150] TetMesh. GSH3D web site. http://www.simulog.fr/tetmesh/

[151] Borouchaki H, George PL & Lo SH. Optimal Delaunay point insertion. *International Journal for Numerical Methods in Engineering* 1996: **39**: 3407–3437.

[152] Lo SH. Optimization of tetrahedral meshes based on element shape measures. *Computers and Structures*1997: **63**(5): 951–961.

[153] Borouchaki H, Hecht F & Frey PJ. Mesh gradation control. *International Journal for Numerical Methods in Engineering* 1998: **43**: 1143–1165.

[154] George PL & Borouchaki H. Generation automatique de maillages tridimensionnels respectant une carte de taille. *Revue Europeenne des Elements Finis* 1998: **7**(4): 339–363.

[155] Lohner R. Progreee in grid generation via the advancing front technique. *Engineering with Computers* 1996: **12**: 186–210.

[156] Lohner R, Regridding surface triangulations. *Journal of Computational Physics* 1996: **126**: 1–10.

[157] Lohner R & Onate E. An advancing front point generation technique. *Communications in Numerical Methods in Engineering*, Wiley, 1998: **14**: 1097–1108.

[158] Lo SH. 3D mesh refinement in compliance with a specified node spacing function. *Computational Mechanics* 1998: **21**: 11–19.

[159] Lo SH. 3D anisotropic mesh refinement in compliance with a general metric specification. *Finite Elements in Analysis and Design* 2001: **38**: 3–19.

[160] Schneiders R. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with Computers* 1996: **12**: 168–177.

[161] Weiler F, Schindler R & Schneiders R. Automatic geometry-adaptive generation of quadrilateral and hexahedral element meshes for the FEM. *Proceedings of the 5th International Conference on Numerical Grid Generation in Computational Field Simulations*, Mississippi State University, 689–697, 1996.

[162] Schneiders R. An algorithm for the generation of hexahedral element meshes on an Octree technique. *Proceedings of the 6th International Meshing Roundtable*, 183–194, 1997.

[163] Li TS, McKeag RM & Armstrong CG. Hexahedral meshing using midpoint subdivision and integer programming. *Computer Methods in Applied Mechanics and Engineering* 1995: **124**: 171–193.

[164] Price MA & Armstrong CG. Hexahedral mesh generation by medial surface subdivision: Part I. *International Journal for Numerical Methods in Engineering* 1995: **38**: 3335–3359.

[165] Price MA & Armstrong CG. Hexahedral mesh generation by medial surface subdivision: Part II. *International Journal for Numerical Methods in Engineering* 1997: **40**: 111–136.

[166] Canann SA. Plastering and optismoothing: New approach to automated, 3D hexahedral mesh generation and mesh smoothing. *PhD Dissertation*. Brigham Young University, Provo, UT, USA, 1991.

[167] Tautges TJ, Blacker T & Mitchell S. The Whisker-Weaving algorithm: A connectivity based method for constructing all-hexahedral finite element meshes. *International Journal for Numerical Methods in Engineering* 1996: **39**: 3327–3349.

[168] Murdoch P & Benzley SE. The spatial twist continuum. *Proceedings of the 4th International Meshing Roundtable*, Sandia National Laboratories, 243–251, 1995.

[169] Owen SJ, Canann SA & Saigal S. Pyramid elements for maintaining tetrahedra and hexahedra conformability, AMD-Vol. 220, Trends in Unstructured Mesh Generation, *ASME*, 123–129, 1997.

[170] Tuchinsky P & Clark BW. The hex-tet, hex-dominant automesher: An interim progress report. *Proceedings of the 6th International Meshing Roundtable*, 183–193, 1997.

[171] Min W. Generating hexahedron-dominant mesh based on shrinking-mapping method. *Proceedings of the 6th International Meshing Roundtable*, 171–182, 1997.

[172] Owen SJ. Constrained triangulation: Application to Hex-dominated mesh generation. *Proceedings of the 8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, 31–41, October 1999.

[173] Owen SJ & Saigal S. H-Morph: An indirect approach to advancing front Hex meshing. *2nd Symposium on Trends in Unstructured Mesh Generation*, University of Colorado, Boulder, August 1999.

S H Lo
Department of Civil Engineering,
The University of Hong Kong,
Pokfulam Road, Hong Kong
E-mail: hreelsh@hkucc.hku.hk

