

Finite element mesh generation methods: a review and classification

K Ho-Le

An attempt is made to provide a clear overall picture of the finite element mesh generation field. A scheme for classifying mesh generation methods is proposed, whereby seven major mesh generation approaches are identified. The basis for classification is the temporal order in which nodes and elements are created. The published mesh generation methods are reviewed. Only four approaches are found to have fully automatic mesh generation methods. These approaches are compared on the criteria of element types and shapes, mesh density control, and time efficiency.

computer-aided design, finite element methods, mesh generation

The finite element method is a powerful and versatile analysis tool, but its usefulness is hampered by the need to generate a mesh, but can be very time-consuming and error-prone if done manually. In recognition of this problem, a large number of methods have been devised to automate the mesh generation task. This paper attempts to provide a clear overall picture of the finite element mesh generation field by reviewing, classifying and comparing the mesh generation methods.

There are few comprehensive reviews of mesh generation methods. Buell and Bush¹ reviewed the pre1973 literature but the field has expanded considerably since then. Thacker² published an extensive bibliography in 1980, but did not classify or compare the methods. The material in this paper is more up to date, and appears to be the first attempt at a systematic classification of the methods.

Devising a classification scheme for the mesh generation methods is difficult, and so is the placement of individual methods into the mesh generation classes. Some methods do not seem to fit into any class, while others could be put in two or more places. The final placement is necessarily somewhat arbitrary. However it is felt that the clarity afforded by the classification outweighs any problems involved in the classification.

The organization of the paper is as follows. The next section gives some useful background information. The classification scheme is then proposed, followed by the review of 2D and 3D mesh generation methods. Finally, the mesh generation classes are compared on four criteria: element types, element shapes, mesh density control, and time efficiency.

BACKGROUND INFORMATION

Delaunay triangulation

Many mesh generators produce a mesh of triangles by first creating all the nodes and then connecting the nodes to form triangles. The question arises as to what is the 'best' triangulation on a given set of points. One particular scheme, namely the Delaunay triangulation, is considered by many researchers³⁻⁵ to be the most suitable for finite element analysis. This triangulation maximizes the sum of the smallest angles of the triangles⁶; thus thin elements are avoided wherever possible.

The Voronoi diagram^{4,7-9} of a set of N points in a plane consists of N polygons $V(i)$, each centred on point i such that the locus of points on the plane nearest to node i are included in $V(i)$. The Delaunay triangulation is obtained by connecting the points associated with neighbouring Voronoi polygons (see Figure 1). The above definitions extend naturally to higher-dimensional spaces.

Conversion of element types

If a mesh generator produces only one type of element, the elements can be converted to another type as desired. Quadrilaterals and bricks are easily converted to well-shaped triangles and tetrahedra of similar sizes (see Figure 2). Triangles and tetrahedra may be subdivided into quadrilaterals and bricks (see Figure 3), but the resulting elements are not well-shaped, because the angles around the newly

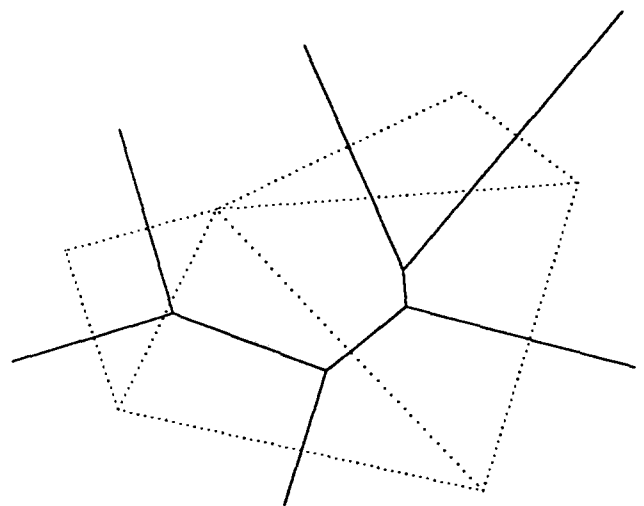


Figure 1. Dirichlet tessellation (solid lines) and the Delaunay triangulation (dashed lines) on a point set

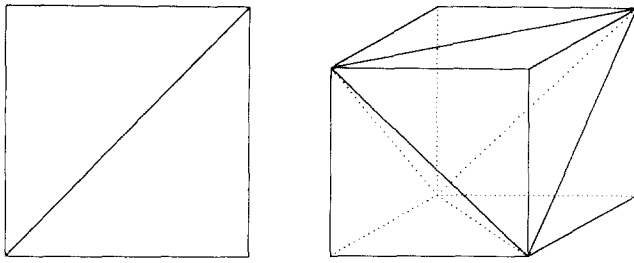


Figure 2. Conversion of the quadrilateral into triangles, and the brick into tetrahedra

introduced nodes are necessarily large. Heighway¹⁰ gives a technique to convert a mesh of triangles into a mesh of quadrilaterals by combining every two adjacent triangles into a quadrilateral.

Mesh smoothing

Often the elements produced by an automatic mesh generator are not well-shaped, in which case it is useful to apply a mesh smoothing technique to improve the mesh. The most popular technique is Laplacian smoothing which seeks to reposition the nodes such that each internal node is at the centroid of the polygon formed by its connected neighbours. This repositioning is usually done iteratively, however, a direct solution scheme has also been reported¹¹.

In some cases, the Laplacian smoothing technique does not work well; for example, the case illustrated in Figure 4(a). To redress the problem, Herrmann¹² proposes to reposition an internal node *i* as follows (cf. Figure 5)

$$P_i = \frac{1}{N(2-w)} \sum_{n=1}^N (P_{ni} + P_{ni} - wP_{nk}) \tag{1}$$

where *N* is the number of elements around the node *i*, and *w* is the weighting factor, 0 ≤ *w* ≤ 1.

If *w* = 0, Laplacian smoothing results. If *w* = 1, the smoothing is called ‘isoparametric’; the mesh quality appears to improve (see Figure 4(b)).

Mesh density and mesh conformity

A mesh must accommodate changes in element sizes from region to region. Most FEM packages require a mesh to be conforming, where adjacent elements share a whole edge or a whole face (Figure 6(a)). A mesh composed of triangles or tetrahedra can easily be made conforming, but the task is more difficult with a mesh of quadrilaterals or bricks. Two approaches for creating a transition from large quadrilaterals to small quadrilaterals are illustrated in Figure 7.

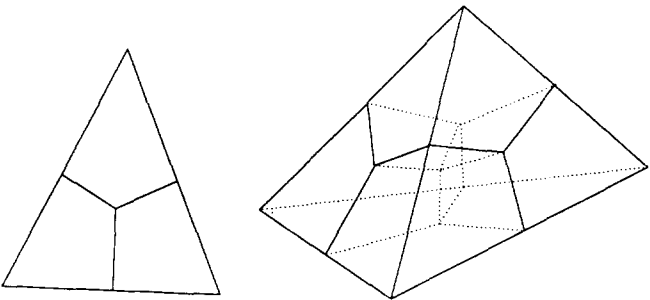


Figure 3. Conversion of the triangle into quadrilaterals, and the tetrahedron into bricks. A new node must be introduced in each case

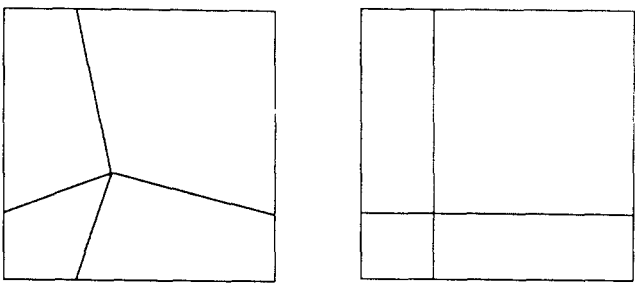


Figure 4. (a) Laplacian smoothing, and (b) isoparametric smoothing (after Herrmann¹²)

When a mesh is refined, some elements are subdivided into smaller elements while others remain unchanged. The question is how to keep the mesh conforming without the introduction of distorted elements. This is difficult with quadrilateral meshes. For triangular meshes, a triangle can be divided into two smaller ones by bisecting the longest side¹³. The diagonal transpose technique⁴, illustrated in Figure 8, can also be used during the refinement process to create a Delaunay triangulation. However, the elements are not refined in place as required by some adaptive mesh refinement methods, such as the multigrid method¹⁴.

If mesh conformity is allowed to be violated, mesh refinement as well as transitioning between coarse- and fine-mesh regions become easier. In a nonconforming mesh, several small elements may abut a larger one (see Figure 6(b)). Multipoint constraints^{15,16} can be applied to the midside nodes to make the mesh analysable. Some novel elements have been introduced to model a large element having many smaller ones abutting its sides^{17–19}. See also Kela *et al.*²⁰, Rheinboldt and Mesztenyi²¹ and Simpson²².

CLASSIFICATION OF MESH GENERATION METHODS

In this section, the mesh generation methods, automatic or not, are classified. The methods are still evolving; the ones that are not completely automatic at the moment may become so in the future, so it is worthwhile including them here. For ease of exposition, the proposed classification scheme is explained in terms of 2D mesh generation, but the scheme is applicable to 3D as well.

The output of a mesh generator consists of a set of

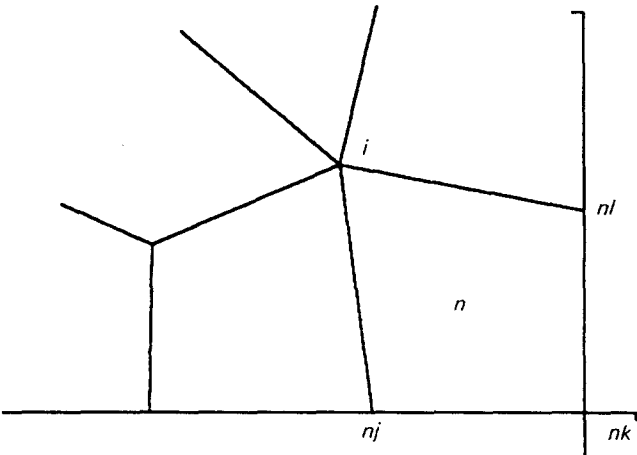


Figure 5. Neighbourhood of an internal node *i* (after Herrmann¹²)

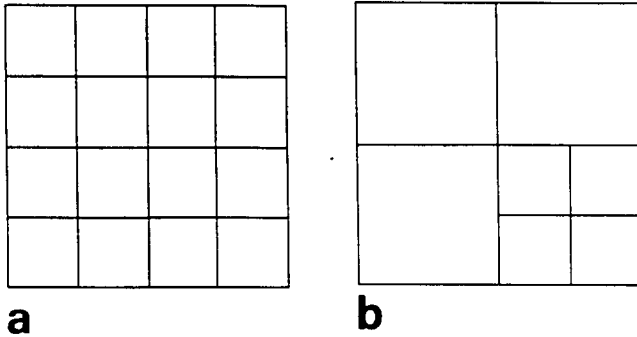


Figure 6. (a) Conforming mesh, and (b) nonconforming mesh, where several small elements may abut a larger one

nodes, and a set of elements (which represents the mesh topology). It is proposed to classify the mesh generation methods according to the temporal order of the creation of these sets. There are four main cases as shown in Figure 9.

Mesh topology first

The mesh topology is determined first, followed by the nodal positions. Once the mesh topology has been determined, the mesh smoothing techniques described previously may be used to calculate the position of the nodal points. The problem with this approach is that there is no known algorithm for creating the mesh topology, except by using other automatic mesh generation methods! Since mesh smoothing has already been explained, this case will be dropped from further discussions.

Nodes first

The nodes are created first, and they are then connected to form triangular or quadrilateral elements. This can be divided into two subcases. If the vertexes of the object are assumed to be the only nodes in the mesh, a triangulation algorithm can be applied to create a minimum number of nonoverlapping triangles that cover the object (see Figure 10). This minimal set of triangles is determined mainly by the topology of the object. The complex topology of the object has been decomposed into the simple topology of the triangles, so this approach will be called the topology decomposition approach. The mesh thus created has distorted and mostly coarse elements, and cannot be used for analysis purposes. Mesh refinement, possibly with element rearrangements by the diagonal transpose technique, must be applied to produce a reasonable mesh.

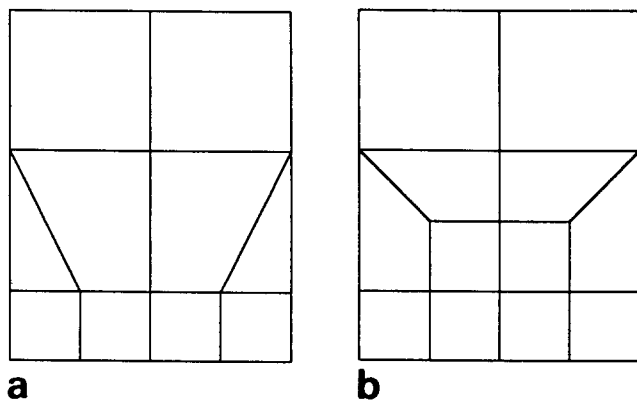


Figure 7. Transition from large to small quadrilaterals: (a) with triangles, and (b) with quadrilaterals only

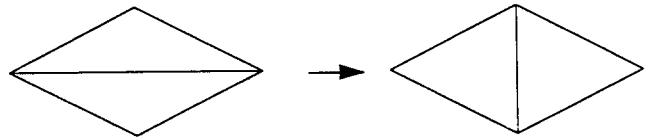


Figure 8. Diagonal transpose to maximize the smallest angle in the triangles

The mesh refinement step above can be avoided if all the nodes in the final mesh are already generated such that their distribution conforms to the mesh density specification. Then, in the element generation phase, one only has to consider how to connect the nodes to form the best possible elements. This approach (see Figure 11), which covers the bulk of the mesh generation literature, will be called the node connection approach.

Adapted mesh template

A mesh template is pregenerated elsewhere, and then adapted to the object being meshed. Three approaches within this case can be identified.

Grid-based approach

The mesh template is an infinite rectangular or triangular grid. The grid is superimposed on the object. The grid cells that fall outside the object are removed. The grid cells that intersect the object boundary are adjusted or trimmed so that they fit into the object. The result is a mesh with excellent interior elements (see Figure 12). This approach is becoming popular, especially in 3D mesh generation.

Mapped element approach

The mesh template is a rectangular mesh in the unit square (or a triangular mesh in a unit triangle) in the parametric space. It is mapped onto a four-sided (or three-sided) region to induce a mesh in the region via a blending function²³. An arbitrary object has to be subdivided manually into three- or four-sided regions, which are in effect macro elements (see Figure 13). This approach is the mainstay of existing commercial mesh generators.

Conformal mapping approach

A polygon Q that has the same number of vertexes as the simply-connected region to be meshed (represented by a polygon P) is constructed such that it can be readily meshed. A conformal mapping F from Q to P is found, based on the correspondence between their vertexes. The mesh in the polygon Q is then mapped onto the object P by this conformal mapping (see Figure 14). This approach is relatively undeveloped compared to other approaches.

Nodes and elements created simultaneously

The nodes and the elements are created simultaneously, with no distinctive phases that can be labeled 'nodes only' or 'elements only'. In this approach, an attempt is made to generate good elements by considering the object geometry, in contrast to the topology decomposition approach which mostly ignores the geometry (except for interference checking purposes). This approach will be called the geometry decomposition approach. Figure 15 illustrates a representative method of this approach.

There are altogether seven classes of mesh generation methods as shown in Figure 9, each of which represents an approach to mesh generation. They will be explained in

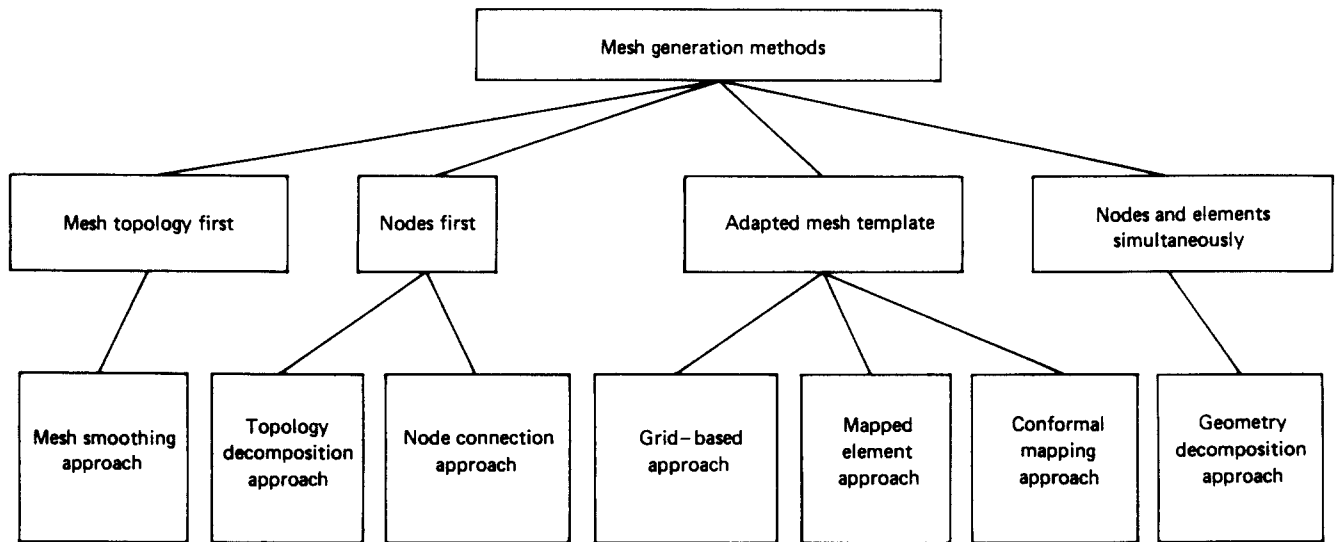


Figure 9. Classification scheme for mesh generation methods

detail in the next two sections, except for the mesh smoothing approach.

2D MESH GENERATION METHODS

This section presents 2D mesh generation methods in as similar a way as possible to their original publication; in some cases the wording and the illustrations are reproduced without change.

Topology decomposition approach

The 2D topology decomposition approach was developed by Wordenweber^{24,25} relatively recently compared to other approaches. In this approach, an object is a (possibly curved) polygon represented by two types of entities: vertexes and edges. Each entity has a geometry which specifies its position or shape, and a topology which specifies its relations to other entities. The geometry of a vertex is its coordinates, and its topology consists of a list of the incident edges. The geometry of an edge is the curve it lies on, and the topology consists of a list of the two end points.

This approach decomposes an object into a set of gross elements by connecting its vertexes to form triangles. Element sizes and shapes cannot really be considered because it is the object topology that determines how the object is decomposed. These gross elements must be refined later to satisfy the required mesh density distribution.

In Wordenweber's method, any hole in the object is first cut open using an operator OP_j to produce singly connected regions. An operator OP_1 is then applied repeatedly to cut off triangles whose vertexes are the object vertexes, until only 3 vertexes are left. An operator OP_0 is

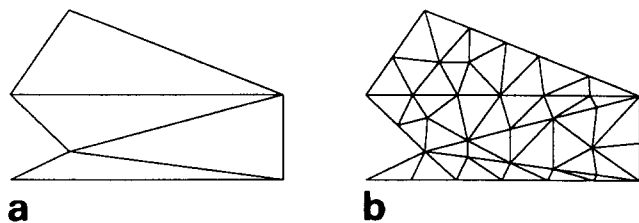


Figure 10. The topology decomposition approach: (a) vertexes are connected to form gross elements, and (b) mesh is refined

finally applied to remove the last triangle. When OP_j and OP_1 are applied, geometric checks must be made to ensure that cuts do not interfere with the object boundary, and that triangles are valid.

An important contribution of this approach to mesh generation is the concept of operators, which was perhaps borrowed from the concept of Euler operators pioneered by Baumgart²⁶. Operators take care of the details of data structure manipulations, and make it possible to think in terms of higher-level commands.

Node connection approach

This mesh generation approach is very popular, perhaps because it is conceptually simple. There are two main phases in the approach: node generation, and element generation.

Node generation

Some mesh generators require the nodes to be created manually. However, there are automatic techniques for node generation. The random node generation techniques²⁷⁻³⁰ have been well-investigated; here the technique by Cavendish²⁸ is chosen as representative.

Cavendish technique: nodes are first added to the object boundary at regular intervals. Then interior nodes are generated to satisfy mesh density requirements as follows. The object is divided into a number of zones of different desired element sizes. In zone i a square grid of gauge $r(i)$ is superimposed. For each sub-square of the grid, one interior node is randomly generated. If this node falls inside the object, and is of a distance greater than $r(i)$ from the boundary and from previously generated nodes, it is accepted. If not, another node is generated and

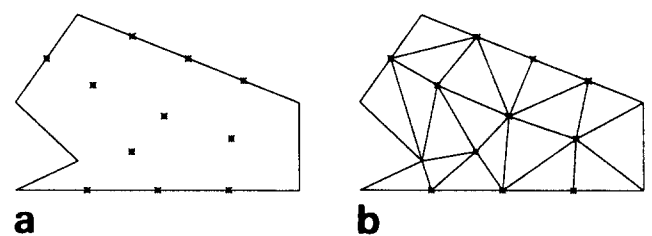


Figure 11. Node connection approach: (a) all nodes are generated, and then (b) connected to form elements

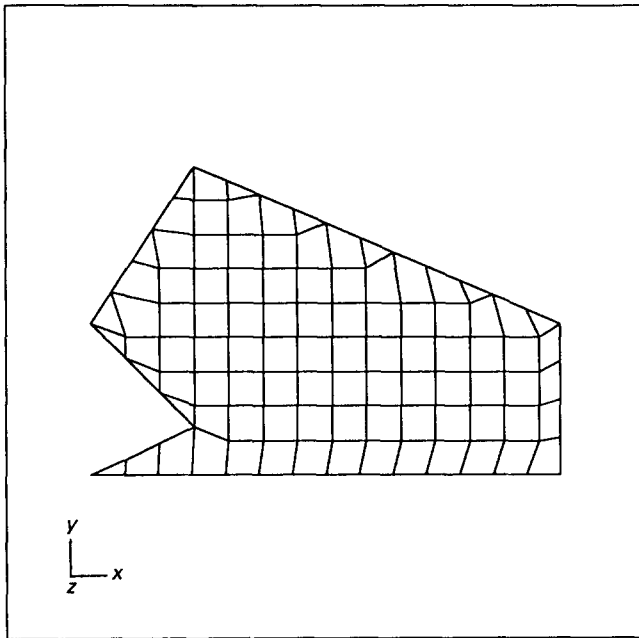


Figure 12. Grid-based approach: interior grid cells give good elements; grid cells along the boundary must be adjusted or trimmed

checked. A fixed number of attempts, say 5, are made to generate an acceptable node for each subsquare, if none is successful then that subsquare is not considered further.

Nonrandom techniques: two such techniques for node generation are known. In Lo's method³¹, horizontal lines with spacing l , where l is the desired element length, are intersected with the object. Nodes at approximately distance l apart are placed on the portions of the lines inside the object to produce a uniformly distributed point set. Lee³² uses the CSG representation for a 2D object that is created by the solid Boolean operations, such as intersection, union, and difference, on the primitives (e.g. circles and rectangles). In each primitive a node pattern of uniform distribution can be easily generated. These nodes are then combined to produce well-distributed nodes depending on the Boolean operations specified. Some nodes in the overlapping regions may need to be deleted, others may need to be merged or moved.

Element generation

In this phase, nodes are connected to form elements such that no elements overlap and the entire object is covered. Most methods produce triangular elements; an exception is Lee's method which attempts to generate quadrilaterals³².

In Lee's method, a square grid whose spacing is the same as the desired element size is superimposed on the object. The nodal points are 'bucketed' in the grid cells. The cells are visited in a columnar manner from left to right, and within the same column, from bottom to top. Within a cell, the points are sorted in ascending x -coordinates. Points having the same x -coordinate are sorted in ascending y -coordinates. The points are visited in turn. For each point, neighbouring points are found so as to form the nodes of a good quadrilateral, failing which, a triangle is formed.

Most of the references on node generation above also give triangulation methods. Some other methods are given below.

Lewis and Robinson's method³³: this recursive algorithm

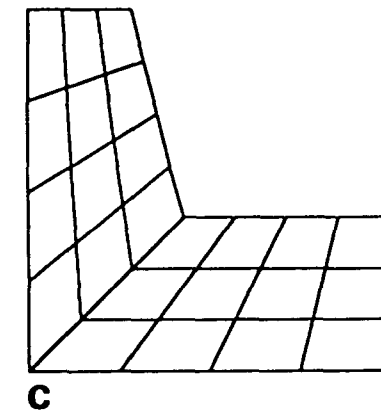
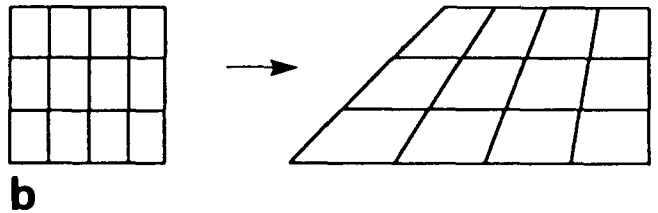
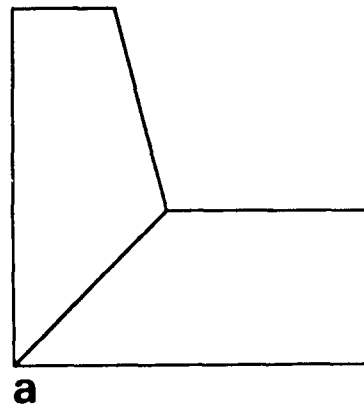


Figure 13. Mapped element approach: (a) object is divided into macro elements, and (b) mesh template in unit square is mapped into each element to produce (c) the final mesh

first divides the object into two halves and then keeps on dividing the halves until only triangles remain. An attempt is made to find the split line that best divides the region into two 'equally sized halves' while keeping the halves as 'circular' as possible. For example, for elliptical shapes a split along the minor axis is preferable. Any interior nodes

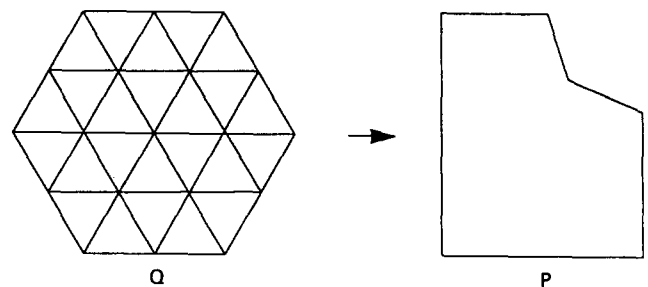


Figure 14. Conformal mapping approach: polygon Q and its mesh is constructed in the parametric space, and then mapped onto the object P by a conformal mapping (after Brown⁵⁶)

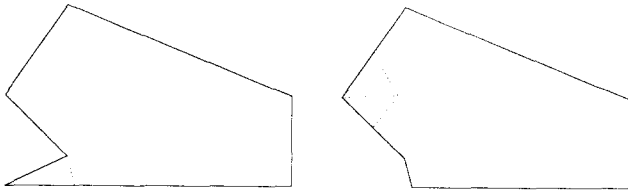


Figure 15. Geometry decomposition approach: one element after another is removed from the object

which lie close to the proposed split line are included as part of the new boundaries, thus giving a zigzag effect.

Frederick's method³⁴: each node is to be surrounded with triangular elements. The method begins by selecting a node i , and a node j nearest to i . The side ij is then used to find a node k such that the angle ikj is a maximum and ijk is in a counterclockwise sequence. The new point k then becomes j , and the process is repeated until the node i is fully surrounded with triangles. A new i node is selected and the process is repeated.

Delaunay triangulation methods: these are perhaps the most important triangulation methods because, as mentioned previously, a Delaunay triangulation maximizes the sum of the smallest angles in all of the triangles in the mesh, thus creating the best mesh for a given set of points. Most of the references on the Delaunay triangulation given previously also contain algorithms for triangulation. Here, another algorithm, by Nelson³⁵, is given.

Nelson's algorithm: start with any boundary segment as a baseline. Find a node which can be connected to form a triangular element without the resulting element crossing any region boundary, and such that all other nodes fall outside the circle circumscribing the element. Of the two new baselines created by this connection, stack one baseline and let the other be the new current baseline. Whenever the current baseline is a boundary-segment or already occurs in two existing elements, unstack a saved baseline. When the stack is empty, the routine is finished.

Instead of finding the Delaunay triangulation, the method of McGirr *et al.*³⁶ starts by finding the Voronoi diagram. The point associated with each Voronoi polygon can then be connected to the polygon vertexes to form triangles. Many of the resulting triangles, however, appear to be too thin, and have to be searched for and removed.

Grid-based approach

The grid-based approach arises out of the observation that a grid looks like a mesh, and it can be made into a mesh provided that the grid cells along the object boundary can be turned into elements. The finer the grid, the better the mesh, since the internal elements which are all well-shaped will become more numerous. The actual methods for mesh construction differ greatly, but the outputs are always similar, since the internal elements are all the same; the only difference being the boundary elements.

The method by Thacker, Gonzalez and Putland³⁷ is probably the first published one using the grid-based approach. An object is first enclosed in an equilateral triangle fitted with a triangular grid. The grid points that fall outside the object are eliminated, leaving a zigzag boundary. The grid points on the zigzag boundary are then moved onto the object boundary to create the finished mesh. Kikuchi³⁸ extends this method to create meshes of predominantly quadrilaterals, but with some triangles, by

using a rectangular grid. One problem with both these methods is that small features, having edges too short compared to the grid spacings, are lost.

Heighway and Biddlecombe³⁹ start with a zigzag boundary that lies entirely within the object, and then mesh the thin strip left between the zigzag boundary and the actual boundary with a triangulation algorithm. The interior elements are initially rectangles but now must be diagonalized into triangles.

Kela, Perucchio and Voelcker²⁰ similarly mesh the interior first with rectangles, but attempt to mesh the boundary strip with quadrilaterals where possible, and triangles otherwise. Only uniform meshes can be generated initially; they may be refined afterwards. They use a CSG representation of the object.

TIPS-1⁴⁰, a geometric modelling system, stores a coarse grid as a secondary geometric representation, in addition to the CSG model. This grid could be turned into a mesh by moving the grid points near the boundary onto the boundary; however in the implementation by Akiyama and Wang⁴¹, this movement has to be done manually.

Yerry and Shephard⁴² start with a quadtree encoding of an object⁴³. This quadtree is modified to be more suitable to mesh generation as follows:

- The object interior is subdivided into quadrants whose sizes satisfy the mesh density distribution.
- Neighbouring quadrants may differ by at most one level of subdivision.
- The quadrants on the boundary may have cut corners.

Each quadrant is next broken up into triangles such that the resulting mesh is conforming. The triangles, especially the boundary ones, may not be well-shaped.

Mapped element approach

This approach requires an object be subdivided manually into simple regions, each of which consists of three or four sides. As such, it is not completely automatic and could have been ignored, were it not for the fact that almost all commercial mesh generators rely on it. Yeung and Hsu⁴⁴ made an attempt to automate the division of the object into gross quadrilaterals, but their algorithm is not robust enough. Indeed, if the algorithm had been robust, it could have been extended to generate elements of the right sizes in the first place!

Given a four-sided region, a mesh can be induced in it by mapping a mesh template of the unit square in the parametric space to the region. Consider the quadrilateral in Figure 16(a) whose sides are f_1 , f_2 , g_1 , and g_2 . Let u be a normalized parametric coordinate along f_1 and f_2 , and v along g_1 and g_2 . The Coons patch methodology⁴⁵ 'blends'

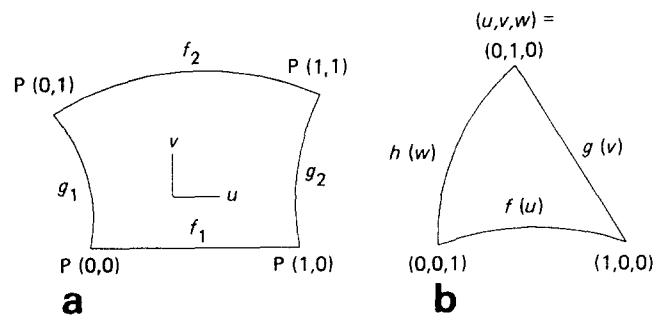


Figure 16. (a) Four-sided region, and (b) three-sided region

these sides to give the Cartesian coordinates of a point $P(u, v)$ in the quadrilateral corresponding to a point (u, v) in the unit square in the parametric space. In its simplest form

$$P(u, v) = (1 - v) f_1(u) + v f_2(u) + (1 - u) g_1(v) + u g_2(v) - (1 - u)(1 - v) P(0, 0) - (1 - u) v P(0, 1) - u v P(1, 1) - u(1 - v) P(1, 0) \quad (2)$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1$$

The mesh template in the parametric space can be a rectangular grid occupying the unit square; the number of subdivisions in the grid depends on the mesh density distribution. The mesh can also be graded by using unequal u - or v -spacings for the grid lines.

The above blending technique can be formulated in terms of the elegant projector theory⁴⁶, and is called the transfinite mapping technique⁴⁷⁻⁴⁹. The projector concept allows the generalization of the technique to the use of higher order interpolants to force the coordinate curves to pass through specified curves on the interior of the region⁵⁰.

A three-sided region can be similarly divided into a mesh of triangular elements by the use of a trilinearly blended interpolant as described by Barnhill *et al.*⁵¹. Referring to Figure 16(b), the simplest formulation of the technique is

$$P(u, v, w) = \frac{1}{2} \left[\frac{ug(v)}{1-v} + \frac{wh(1-v)}{1-v} + \frac{vh(w)}{1-w} + \frac{uf(1-w)}{1-w} + \frac{wf(u)}{1-u} + \frac{vg(1-u)}{1-u} - wf(0) - ug(0) - vh(0) \right] \quad (3)$$

$$u + v + w = 1$$

$$0 \leq u \leq 1, \quad 0 \leq v \leq 1, \quad 0 \leq w \leq 1$$

The isoparametric mapping is a special case of the trans-

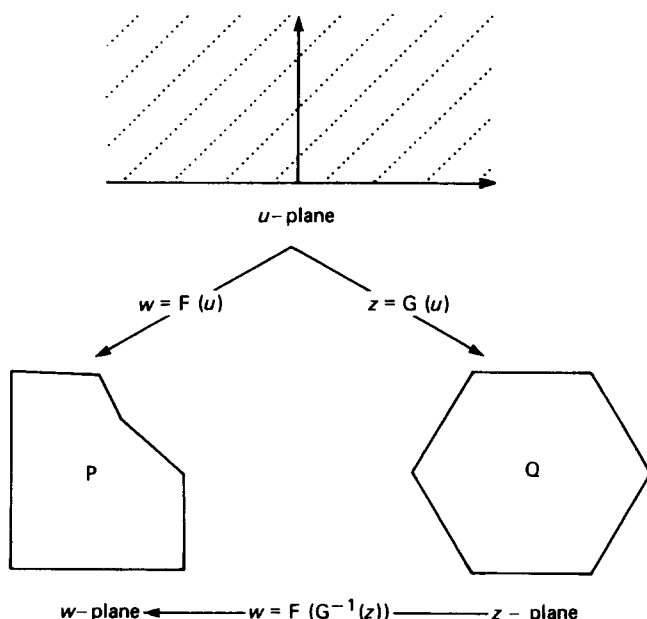


Figure 17. Conformal mapping from the polygon Q to the polygon P (after Brown⁵⁶)

finite mapping where the boundary curves of a quadrilateral are described by Lagrange polynomials. The technique, first described by Zienkiewicz and Phillips⁵², is widely used for mesh generation. In their formulation, two opposite sides of the quadrilateral must have the same number of subdivisions. In this case, the bandwidth of the object stiffness matrix can be minimized by an appropriate node numbering scheme. Cohen⁵³ presents a method that allows opposite sides to have different numbers of subdivisions.

The discrete transfinite mapping⁴⁷ is another special case of the transfinite mapping where a boundary curve is represented as a finite list of points located on the curve, with a unique coordinate associated with each point in the list. The coordinate of the i th point in a boundary curve of n points can be taken as $(i-1)/(n-1)$. The boundary curves can be very general. For example, several simpler curves can be concatenated to form a single complex curve. Boundaries that have no exact mathematical description are easily handled.

Yildir and Wexler⁵⁴ models the boundaries of a simple region with splines. By using cubic splines or higher order splines, the curvature from one region to another can be made continuous.

Conformal mapping approach

While this approach can deal with simply-connected regions having more than four sides, and hence is more general than the mapped element approach, it is not popular because element shape and mesh density are difficult to control. Three methods based on this approach are known⁵⁵⁻⁵⁷. The method by Brown⁵⁶ will be described here, as being representative of this approach. This method consists of four steps:

- Step 1: model the 2D simply-connected region to be meshed with a straight-sided polygon P in the complex w -plane.
- Step 2: define a polygon Q in the complex z -plane such that Q has the same number of vertexes as P, and a mesh for Q composed entirely of ideally-shaped elements may be generated easily. The mesh topology for Q will be the mesh topology for P.
- Step 3: find the Schwarz-Christoffel transformation G that maps from the upper half u -plane onto the interior of Q, and the transformation F from the upper half u -plane onto the interior of P (see Figure 17).
- Step 4: map the mesh from Q onto P using the composite mapping

$$w = F(G^{-1}(z)) \quad (4)$$

One problem with this method is that the inverse mapping G^{-1} cannot always be found.

Geometry decomposition approach

This approach gives some consideration to element shapes and sizes while decomposing an object. Some methods are based on recursion, others on iteration. Representative methods are described below.

In Bykat's method⁵⁸, the original object has to be divided into convex parts first. This can be done automatically⁵⁹. Nodes are next inserted into the boundary of a convex object to satisfy the mesh density distribution. Then the object is divided roughly in the middle of its

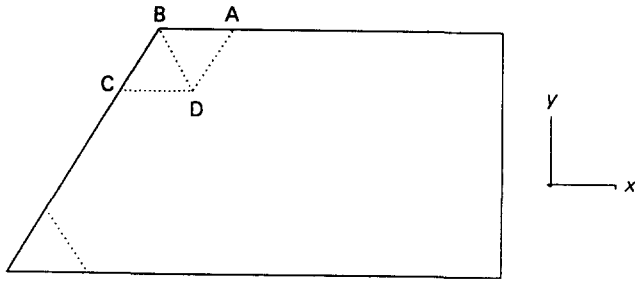


Figure 18. Tracy's algorithm: angles less than 90° are cut off to form one triangle, and angles between 90° and 180° are cut off to form two triangles

'longest axis'. More nodes are inserted into this split line according to the mesh density distribution. The two halves are recursively divided until they become triangles.

Triquamesh⁶⁰, a commercial mesh generator, is based on a recursive algorithm similar to the above, but uses a different criterion for choosing the split line. It is capable of generating an all-quadrilateral or all-triangle mesh.

An iterative algorithm that removes one or two triangles at a time from the boundary inwards is given by Tracy⁶¹. It assumes the boundary of a singly-connected region has been subdivided into straight line segments whose lengths correspond to the desired element sizes. The algorithm has four steps:

- Step 1: assemble the vertexes of the region in an array, in the counterclockwise order, to form a polygon.
- Step 2: remove all vertex angles of the polygon which are less than 90° by forming triangular elements (Figure 18).
- Step 3: start at some vertex B with an angle of less than 180° and form two triangles with the adjacent points to the vertex, A and C, and a new point D (see Figure 18). For (X_D, Y_D) use

$$\begin{aligned} X_D &= \frac{X_A + X_C}{2} + \frac{Y_A - Y_C}{5} \\ Y_D &= \frac{Y_A + Y_C}{2} + \frac{X_C - X_A}{5} \end{aligned} \quad (5)$$

where the factor 5 is arbitrary. Update the polygon array by replacing B with D.

- Step 4: check if step 3 creates new angles of less than 90°. If so, repeat step 2 until it is possible to do step 3 again or the region has only three points left. From the last three points form a triangular element, and the process is complete.

Bykat⁵⁹ and Sadek⁶² also report similar iterative algorithms. Lindholm's iterative algorithm⁶³ is slightly different: boundary layers, rather than elements, are removed from the object one at a time and then triangulated (see Figure 19).

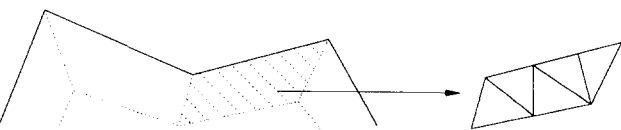


Figure 19. Lindholm's algorithm: boundary layer is removed from object and triangulated

3D MESH GENERATION METHODS

There are very few published 3D mesh generation methods compared to 2D methods due to the greatly increased complexity. While several mesh generation approaches offer the quadrilateral element as an output in 2D, none has been extended to offer the brick element for the 3D case. The following description of the 3D methods is brief because they are based on the same ideas as the 2D methods.

The author is not aware of any 3D mesh generation method based on the conformal mapping approach. The mapped element approach is simple to extend to 3D, so that a mesh can be generated in a four-, five- or six-faced region^{52,64,65}. However, as in the 2D case, this approach is not completely automatic, and will not be discussed further. The other four approaches have been extended to 3D, and are reviewed below.

Topology decomposition approach

Wordenweber's 2D mesh generation method²⁴ has been extended to 3D^{24,25}. Another method based on the topology decomposition approach, by Woo and Thomas⁶⁶, is intended solely for 3D, and is given here.

A polyhedral object having holes is first reduced to an object without holes by cutting open the holes with an operator T_3 as shown in Figure 20. Three tetrahedra are produced as by-products. An object having no hole is decomposed into tetrahedra by two operators. The operator T_1 operates on a convex trivalent vertex (that is, a vertex having three edges) to construct a tetrahedron by cutting out the corner. The operator T_2 digs out a tetrahedron from a convex edge. In either case the tetrahedron must not interfere with any part of the object. Interference is defined by the following two rules:

- VT: No vertex of the object lies on any of the four faces of the tetrahedron.
- ET: No edge of the object intersects any of the four faces of the tetrahedron.

The algorithm proceeds as follows. If there exists a convex trivalent vertex in the object, then apply the operator T_1 to construct a tetrahedron T . If T passes interference tests VT and ET, then remove it from the object. Continue with the operator T_1 on the remaining object. If all the remaining vertexes are not convex trivalent, or if all the constructed tetrahedra fail the interference tests, then apply the operator T_2 to dig out a tetrahedron. If this tetrahedron passes the interference tests, go back to the operator T_1 . Continue this process until the object is reduced to a single tetrahedron.

Node connection approach

The two following mesh generation methods, based on the node connection approach, require the user to create the nodes manually. The author is not aware of any published algorithm for the automatic generation of nodes in 3D.

Nguyen-Van-Phai⁶⁷ extends Frederick's 2D method to 3D by fully surrounding a line connecting two nearest nodes with tetrahedra. Cavendish³ uses a 3D Delaunay triangulation algorithm to create tetrahedra. These tetrahedra form a convex set, with many of them falling outside the object, so each one has to be tested to see if it is inside or outside. The ones outside the object are discarded. It is surprising that while the 2D Delaunay triangulation is, in

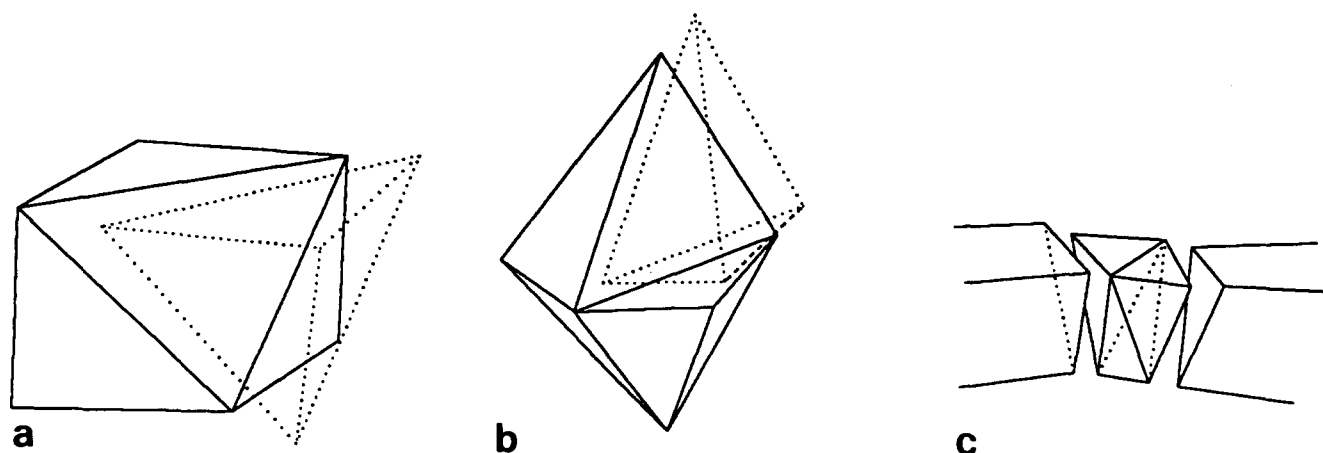


Figure 20. Woo and Thomasma's method: (a) operator T_1 : cutting a convex corner, (b) operator T_2 : digging out a tetrahedron, and (c) operator T_3 : cutting open a hole (after Woo and Thomasma⁶⁶)

some sense, the optimal triangulation on a given point set, the 3D Delaunay triangulation may contain very thin tetrahedra, which must be identified and removed.

Grid-based approach

Of all the six grid-based methods for 2D mesh generation discussed previously, only the one by Yerry and Shephard⁶⁸ has been extended to 3D. The object is octree-encoded⁴³, again with some modifications. The octants are broken up into tetrahedra by using a complicated algorithm.

Geometry decomposition approach

Triquimesh⁶⁹ appears to be the only 3D mesh generator based on this approach. It recursively divides an object into two subvolumes, by choosing a 'best splitting plane', until all subvolumes have been reduced to tetrahedra. Bricks cannot be generated directly, in contrast to the 2D case where quadrilaterals can be produced by the method. Each tetrahedron may be subdivided into four bricks if desired.

COMPARISON OF APPROACHES

Comparing the mesh generation approaches is difficult because access to the actual mesh generators is not available. Therefore, it is necessary to make do with the literature and try to infer the capabilities of the approaches.

Only fully automatic mesh generation approaches are compared. This restriction eliminates the two semi-automatic approaches, namely the conformal mapping approach and the mapped element approach. The current implementations of the node connection approach for 3D are not really automatic because nodes must be created manually, but this problem can be easily solved. One possible solution is to superimpose a rectangular grid on the object and choose the grid points that fall inside the object. Therefore this approach should be admitted as an automatic one. Altogether, there are four approaches to be compared: the topology decomposition approach, the node connection approach, the grid-based approach, and the geometry decomposition approach.

Some mesh generation methods, notably the ones based on the topology decomposition approach, do not generate an initial mesh that is good enough for analysis. They rely on postprocessing to improve the mesh. Three postprocessing steps can be used:

- Step 1: if the elements generated are not of the desired type, then subdivide them into elements of the desired type.
- Step 2: if the elements do not have sizes compatible with the desired mesh density distribution, then refine them.
- Step 3: if the elements are not well-shaped, then apply a mesh smoothing technique.

The methods for performing these steps have already been discussed. If the postprocessing steps work perfectly, then any initial mesh will do. However, they do not, and the quality of the final mesh depends strongly on the quality of the initial one. Thus the approaches will be compared based on their unprocessed output mesh. Four criteria for comparison are used:

- element type
- element shape
- mesh density control
- time efficiency

The criterion of space efficiency is relatively unimportant and hence will not be discussed. Bandwidth or frontwidth optimization, while very desirable, unfortunately cannot be achieved by any automatic mesh generation approaches (but can be done in the semi-automatic approaches^{52,57}). Postprocessing must be done to optimize the bandwidth or the frontwidth, using algorithms such as those in Cuthill and McKee⁷⁰ and Pina⁷¹.

The mesh generation approaches are now compared. The results of the best implementations of the approaches on each criterion are summarized in Table 1.

Table 1. Comparison of the mesh generation approaches

Approach	Quadri-lateral	Brick	Element shape	Mesh density control	Time efficiency
Topological decomposition	No	No	Poor	No	$O(N^2)$
Node connection	Yes	No	2D good, 3D fair	Yes	$O(N)$
Grid-based	Yes	Relatively easy	Interior elements excellent	Yes	$O(N)$
Geometric decomposition	Yes	No	2D good, 3D unknown	Yes	unknown

(N = number of nodes in the mesh)

Element type

The triangle and the tetrahedron are relatively easy to generate, and all approaches can produce them as the primary output. We are interested in the approaches that can also produce the quadrilateral and the brick directly.

In 2D, most methods produce triangles; the exceptions being Lee's method³² (of the node connection approach), the method of Kela *et al.*²⁰ (of the grid-based approach), and Triquamesh⁶⁰ (of the geometry decomposition approach). The former two methods produce a mixture of triangles and quadrilaterals; the latter, all quadrilaterals. In 3D, no existing method can produce bricks. The grid-based approach, however, already offers bricks in the object interior, and should therefore be easier than other approaches to coax into generating bricks.

Element shape

In the topology decomposition approach, element shapes are determined by the object topology and hence cannot be easily controlled. With careful implementation, all other approaches appear to be capable of generating well-shaped 2D elements; see for example the methods of Lee³², Kela *et al.*²⁰ and Bykat⁵⁹. In 3D, the Delaunay triangulation³ of the node connection approach cannot guarantee well-shaped elements as it does in 2D. The grid-based approach⁶⁸ gives excellent interior elements and the mesh should be fine overall, though in Yerry and Shephard's implementation, some boundary-elements appear to have poor shapes. The geometry decomposition approach in 3D, as represented by Triquamesh, cannot be assessed, because details of the method are not available.

Mesh density control

The topology decomposition approach cannot control mesh densities directly and has to rely on mesh refinement. The other approaches are all capable of mesh density control, as exemplified by Cavendish *et al.*³, Yerry and Shephard⁴² and Bykat⁵⁸.

Time efficiency

Let N be the number of nodes in the output mesh. Wordenweber's method (of the topology decomposition approach) requires $O(N^2)$ time²⁴. The 2D Delaunay triangulation algorithm by Green and Sibson⁹ (of the node connection approach) requires $O(N^{3/2})$ time, though one can do better by using the linear expected time algorithm of Bentley *et al.*⁷² as cited in Devroye⁷³. By using a bucket algorithm, Lee's method (also of the node connection approach) requires only $O(N)$ time. The 3D Delaunay triangulation algorithm by Cavendish³ requires $O(N^2)$ time. The grid-based approach should require only $O(N)$ by virtue of the spatial addressing property of the grid. The efficiency of the geometry decomposition approach has not been analysed by any author, though Bykat's recursive algorithm in 2D⁵⁸ has an observed $O(N)$ running time.

CONCLUDING REMARKS

In this paper, the published mesh generation methods are reviewed, classified and compared. Seven mesh generation approaches are identified, but only four of them are found to be automatic. The classification scheme presented here is new and may need further refinements. However, it is already useful, not only in showing a clear overall picture of the mesh generation field, but also in giving insights into

the relationship between the approaches. For example, the topology decomposition approach and the node connection approach, two subbranches from a common branch (see Figure 9), both involve a phase where nodal connectivity is generated from a given point set. Many techniques in the node connection approach, e.g. the Delaunay triangulation, would also be applicable to the topology decomposition approach, as far as this phase is concerned.

Accurate comparison of the approaches is difficult, since access to the actual mesh generators is not available. Another reason is that the approaches are still evolving; present limitations in their capability may be overcome later. However, the comparison does give some idea of their potential for 3D mesh generation.

ACKNOWLEDGEMENTS

The author wishes to thank Mr R J Maxwell, Dr T F Berreen, Dr T H Siau, Dr W R Thorpe and Dr M C Good for discussions and helpful comments. This work arises out of a collaborative project between CSIRO and Monash University. The support of CSIRO for this work is greatly appreciated. The author also wishes to thank the referees for suggestions to improve the paper.

REFERENCES

- 1 Buell, W R and Bush, B A 'Mesh generation — a survey' *Trans. ASME, J. Eng. Ind.* (February 1973) pp 332–338
- 2 Thacker, W C 'A brief review of techniques for generating irregular computational grids' *Int. J. Numer. Meth. Eng.* Vol 15 (1980) pp 1335–1341
- 3 Cavendish, J C, Field, D A and Frey, W H 'An approach to automatic three-dimensional finite element mesh generation' *Int. J. Numer. Meth. Eng.* Vol 21 (1985) pp 329–347
- 4 Cendes, Z J, Shenton, D and Shahnasser, H 'Magnetic field computation using Delaunay triangulation and complementary finite element methods' *IEEE Trans. Mag.* Vol MAG-19 No 6 (November 1983)
- 5 Boubez, T I, Funnell, W R J, Lowther, D A, Pinchuk, A R and Silvester, P P 'Mesh generation for computational analysis' *Comput.-Aided Eng. J.* (October 1986) pp 190–201
- 6 Lawson, C L 'Software for C1 surface interpolation' in Rice, J (ed) *Mathematical Software III* New York Academic (1977) *Math. Software Symp.* Madison, USA pp 161–193
- 7 Brostow, W and Dussault, J P 'Construction of Voronoi polyhedra' *J. of Computational Physics* Vol 29 (1978) pp 81–92
- 8 Shamos, M I 'Geometric complexity' *Proc. 7th Annual ACM SIGACT Conf.* (May 1975) pp 224–233
- 9 Green, P J and Sibson, R 'Computing dirichlet tessellations in the plane' *The Computer J.* Vol 21 No 2 (1977) pp 168–173
- 10 Heighway, E A 'A mesh generator for automatically subdividing irregular polygons into quadrilaterals' *IEEE Trans. on Mag.* Vol MAG-19 No 6 (November 1983) pp 2535–2538
- 11 Denayer, A 'Automatic generation of finite element meshes' *Comput. & Struc.* Vol 9 (1978) pp 359–364

- 12 Herrmann, L R 'Laplacian-isoparametric grid generation scheme' *J. of Eng. Mech. Div. Proceedings of Amer. Soc. Civil Eng.* Vol 102(EM5) (October 1976)
- 13 Rivara, M C 'Algorithms for refining triangular grids suitable for adaptive and multi-grid techniques' *Int. J. Numer. Meth. Eng.* Vol 20 (1984) pp 745–756
- 14 Brandt, A 'Multi-level adaptive solutions to boundary-value problems' *Math. of Computation* Vol 31 No 138 (April 1977) pp 333–390
- 15 *Abaqus user's manual* Hibbitt, Karlsson, and Sorensen, Inc. (May 1984) Version 4.5
- 16 *Supertab user's manual* Structural Dynamics Research Corporation (1986)
- 17 Gupta, A K 'A finite element for transition from a fine to a coarse grid' *Int. J. Numer. Meth. Eng.* Vol 12 (1978) pp 35–45
- 18 Gregory, J A, Fishelov, D, Schut, B and Whiteman, J R 'Local mesh refinement with finite elements for elliptic problems' *J. Comput. Physics* Vol 29 (1978) pp 133–140
- 19 Cavendish, J C, Gordon, W J and Hall, C A 'Substructured macro elements based on locally blended interpolation' *Int. J. Numer. Meth. Eng.* Vol 11 (1977) pp 1405–1421
- 20 Kela, A, Perucchio, R and Voelcker, H B 'Toward automatic finite element analysis' *Comput. Mech. Eng.* Vol 5 No 1 (July 1986)
- 21 Rheinboldt, W C and Mesztenyi, C K 'On a data structure for adaptive finite element mesh refinements' *ACM Trans. on Math. Software* Vol 6 No 2 (June 1980) pp 166–187
- 22 Simpson, R B 'Automatic local refinement for irregular rectangular meshes' *Int. J. Numer. Meth. Eng.* Vol 14 (1979) pp 1665–1678
- 23 Gordon, W J 'Blending-function methods of bivariate and multivariate interpolation and approximation' *SIAM J. Numer. Anal.* Vol 8 No 1 (1971) pp 158–177
- 24 Wordenweber, B 'Volume triangulation' *Technical Report* University of Cambridge, UK, CAD Group Document No. 110 (1980)
- 25 Wordenweber, B 'Finite element mesh generation' *Comput.-Aided Des.* Vol 16 No 5 (September 1984) pp 285–291
- 26 Baumgart, B G 'Geometric modeling for computer vision' *Report No. CS-463* Stanford Artificial Intelligence Laboratory, Computer Science Dept, Stanford, USA (October 1974)
- 27 Suhara, J and Fukuda, J 'Automatic mesh generation for finite element analysis' in Oden, J T, Clough, R W and Yamamoto, Y (eds) *Advances in computational methods in structural mechanics and design* UAH Press, Huntsville, Alabama, USA (1972)
- 28 Cavendish, J C 'Automatic triangulation of arbitrary planar domains for the finite element method' *Int. J. Numer. Meth. Eng.* Vol 8 (1974) pp 679–696
- 29 Moscardini, A O, Lewis, B A and Cross, M 'AGTHOM — automatic generation of triangular and higher order meshes' *Int. J. Numer. Meth. Eng.* Vol 19 (1983) pp 1331–1353
- 30 Shaw, R D and Pitchen, R G 'Modifications to the Suhara-Fukuda method of network generation' *Int. J. Numer. Meth. Eng.* Vol 12 (1978) pp 93–99
- 31 Lo, S H 'A new mesh generation scheme for arbitrary planar domains' *Int. J. Numer. Meth. Eng.* Vol 21 (1985) pp 1403–1426
- 32 Lee, Y T 'Automatic finite element generation based on constructive solid geometry' *PhD thesis* Mechanical Engineering Dept, University of Leeds, Leeds, UK (1983)
- 33 Lewis, B A and Robinson, J S 'Triangulation of planar regions with applications' *Comput. J.* Vol 21 No 4 (1977) pp 324–332
- 34 Frederick, C O, Wong, Y C and Edge, F W 'Two-dimensional automatic mesh generation for structural analysis' *Int. J. Numer. Meth. Eng.* Vol 2 (1970) pp 133–144
- 35 Nelson, J M 'A triangulation algorithm for arbitrary planar domains' *Appl. Math. Modeling*, Vol 2 (1978) pp 151–159
- 36 McGirr, M B, Corderoy, D, Easterbrook, P and Hellier, A 'A new approach to automatic mesh generation in the continuum' *Proc. 4th Int. Conf. Australia Finite Element Method* Melbourne, Australia (18–20 August 1982) pp 36–40
- 37 Thacker, W C, Gonzalez, A and Putland, G E 'A method for automating the construction of irregular computational grids for storm surge forecast models' *J. of Computational Physics* Vol 37 (1980) pp 371–387
- 38 Kikuchi, Noboru 'Adaptive grid-design methods for finite element analysis' *Comput. Methods in Applied Mech. Eng.* Vol 55 (1986) pp 129–160
- 39 Heighway, E A and Biddlecombe, C S 'Two-dimensional automatic triangular mesh generation for the finite element electromagnetics package PE2D' *IEEE Trans. on Mag.* Vol MAG-18 No 2 (March 1982) pp 594–598
- 40 Wang, K K and Hashimoto, N 'Test and evaluation of TIPS-1 system' *Technical Report MME-04* Cornell University, Ithaca, NY, USA (1981)
- 41 Akiyama, T and Wang, K K 'A TIPS-1 based CAD program for mold design' *Proc. 9th North American Manufacturing Research Conf.* (19–22 May 1981)
- 42 Yerry, M A and Shephard, M S 'A modified quadtree approach to finite element mesh generation' *IEEE Comput. Graph. & Appl.* (February 1983) pp 39–46
- 43 Samet, H 'The quadtree and related hierarchical data structures' *ACM Computing Surveys* Vol 16 No 2 (1984) pp 187–260
- 44 Yeung, S F and Hsu, M B 'A mesh generation method based on set theory' *Comput. & Struct.* Vol 3 (1973) pp 1063–1077
- 45 Coons, S A 'Surfaces for computer-aided design of space forms' *Technical Report MAC-TR 44* MIT, Cambridge, MA, USA (1967)
- 46 Gordon, W J 'An operator calculus for surface and volume modeling' *IEEE Comput. Graph. & Appl.* (October 1983) pp 18–22

- 47 Haber, R, Shephard, M S, Abel, J F, Gallagher, R H and Greenberg, D P 'A general two-dimensional graphical finite element preprocessor utilizing discrete transfinite mappings' *Int. J. Numer. Meth. Eng.* Vol 17 (1981) pp 1015–1044
- 48 Hall, C A 'Transfinite interpolation and applications to engineering problems' in Law and Sahney (eds) *Theory of approximation* Academic Press (1976) pp 308–331
- 49 Cook, W A 'Body oriented (natural) co-ordinates for generating three-dimensional meshes' *Int. J. Numer. Meth. Eng.* Vol 8 (1974) pp 27–43
- 50 Gordon, W J and Hall, C A 'Construction of curvilinear co-ordinate systems and applications to mesh generation' *Int. J. Numer. Meth. Eng.* Vol 7 (1973) pp 461–477
- 51 Barnhill, R E, Birkhoff, G and Gordon, W J 'Smooth interpolation in triangles' *J. Approx. Theory* Vol 8 (1973) pp 114–128
- 52 Zienkiewicz, O C and Phillips, D V 'An automatic mesh generation scheme for plane and curved surfaces by 'isoparametric' co-ordinates' *Int. J. Numer. Meth. Eng.* Vol 3 (1971) pp 519–528
- 53 Cohen, H D 'A method for the automatic generation of triangular elements on a surface' *Int. J. Numer. Meth. Eng.* Vol 15 (1980) pp 470–476
- 54 Yildir, Y B and Wexler, A 'MANDAP – A FEM/BEM data preparation package' *IEEE Trans. Mag.* Vol MAG-19 No 6 (November 1983) pp 2562–2565
- 55 Barfield, W D 'Numerical method for generating orthogonal curvilinear meshes' *J. Computational Physics* Vol 5 (1970) pp 23–33
- 56 Brown, P R 'A non-interactive method for the automatic generation of finite element meshes using the Schwarz-Christoffel transformation' *Comp. Methods in Appl. Mech. Eng.* Vol 25 (1981) pp 101–126
- 57 Baldwin, K H and Schreyer, H L 'Automatic generation of quadrilateral elements by a conformal mapping' *Eng. Comput.* Vol 2 (1985) pp 187–194
- 58 Bykat, A 'Design of a recursive, shape controlling mesh generator' *Int. J. Numer. Meth. Eng.* Vol 19 (1983) pp 1375–1390
- 59 Bykat, A 'Automatic generation of triangular grid: I – subdivision of a general polygon into convex sub-regions. II – triangulation of convex polygons' *Int. J. Numer. Meth. Eng.* Vol 10 (1976) pp 1329–1342
- 60 Sluiter, M L C and Hansen, D C 'A general purpose automatic mesh generator for shell and solid finite elements' in Hulbert, L E (ed) *Computers in engineering*, Vol 3, Book No. G00217, ASME (1982) pp 29–34
- 61 Tracy, F T 'Graphical pre- and post-processor for two-dimensional finite element method programs' *Proc. SIGGRAPH '77* Vol 11 No 2 (1977) pp 8–12
- 62 Sadek, E A 'A scheme for the automatic generation of triangular finite elements' *Int. J. Numer. Meth. Eng.* Vol 15 (1980) pp 1813–1822
- 63 Lindholm, D A 'Automatic triangular mesh generation on surfaces of polyhedra' *IEEE Trans. Mag.* Vol MAG-19 No 6 (November 1983) pp 1539–1542
- 64 Miki, K and Takagi, T 'A domain decomposition and overlapping method for the generation of three-dimensional boundary-fitted coordinate systems' *J. Computational Physics*, Vol 53 (1984) pp 319–330
- 65 Lamont, J H and Butlin, G 'The FECS pre-processor FEMGEN' in Adey, R A (ed) *Engineering software III* Springer-Verlag (1983) pp 1005–1039
- 66 Woo, T C and Thomasma, T 'An algorithm for generating solid elements in objects with holes' *Comp. & Struct.* Vol 18 No 2 (1984) pp 333–342
- 67 Nguyen-Van-Phai 'Automatic mesh generation with tetrahedron elements' *Int. J. Numer. Meth. Eng.* Vol 18 (1982) pp 273–289
- 68 Yerry, M A and Shephard, M S 'Automatic three-dimensional mesh generation by the modified-octree technique' *Int. J. Numer. Meth. Eng.* Vol 20 No 11 (November 1984) pp 1965–1990
- 69 Jain, Amresh 'Modern methods for automatic FE mesh generation' in Baldwin, K (ed) *Modern methods for automating finite element mesh generation* The American Society of Civil Engineers (1986) pp 19–28
- 70 Cuthill, E and McKee, T 'Reducing the bandwidth of sparse symmetric matrices' *Proc. 24th National Conf. Assoc. Comput. Mach.* Publication P-69 ACM (1969) pp 151–172
- 71 Pina, H L G 'An algorithm for frontwidth reduction' *Int. J. Numer. Meth. Eng.* Vol 17 (1981) pp 1539–1546
- 72 Bentley, J L, Weide, B W and Yao, A C 'Optimal expected-time algorithm for closest point problems' *ACM Trans. on Math. Software* Vol 6 (1980) pp 563–580
- 73 Devroye, Luc *Lecture notes on bucket algorithms* Birkhauser (1986)