ELSEVIER

Finite Elements in Analysis and Design 20 (1995) 47–70

FINITE ELEMENTS IN ANALYSIS AND DESIGN

Aspects of adaptive mesh generation based on domain decomposition and Delaunay triangulation

Roland W. Lewis *, Yao Zheng, Asif S. Usmani

Department of Civil Engineering, University College of Swansea, University of Wales, Swansea, SA2 8PP, UK

Abstract

The finite element method requires the generation of a mesh, based on an appropriate density distribution, so that the numerical analysis using it provides as optimal a result as possible with a reasonably low computational cost. The generation of inner points in a spatial domain of analysis may be accomplished via two types of quadtree decomposition for two-dimensional cases. The density formulations are quoted and analyses of their performance are given. Delaunay triangulation has been utilized within the mesh generator to connect the interior points. The robustness of this technique has been investigated. For real engineering applications, boundary recovery algorithms have been adopted in order to ensure the integrity of the boundary. A series of benchmark tests have been carried out on this work. Mesh quality improvement and the conversion from triangles to quadrilaterals has also been discussed.

1. Introduction

In finite element analysis, the construction of an appropriate mesh is a key problem. It is essential that a mesh be generated which is based on an appropriate density distribution such that the numerical analysis will provide as optimal a result as possible with a reasonably low computational cost [1,2]. In this paper, it has been attempted to address techniques involved in domain decomposition, and to investigate some formulations of density distribution. All the finite element meshes reported were constructed by means of Delaunay triangulation. The issue of robustness for mesh generation based on this technique has been addressed. Boundary recovery algorithms have also been investigated to ensure boundary integrity for real engineering applications. A series of benchmark tests are illustrated and mesh quality improvement and conversion from triangles to quadrilaterals has also been discussed. This paper is mainly concerned with two-dimensional geometries, however some of the discussion extends to three-dimensional geometries by analogy. An example of convection dominated heat transfer has been used to illustrate the dramatic improvement in the quality of the finite element solution when adaptive remeshing is employed.

* Corresponding author

SSDI 0168-874X(95)00010-0

2. Domain boundaries

Given a geometrical shape for finite element analysis, there are several different methods which may be used to produce the necessary nodal points and subsequent element mesh. Domain decomposition is one such method which is useful for generating the necessary nodal seeds [1–4]. All the nodal points can be directly created in this way, and then triangulated into a finite element mesh. This will include a modification process of the boundary cells at a tree level, in order to represent the domain boundary. The alternative strategy used in this paper is to create the boundary points, based on general segment subdivision, and then to generate the interior points based on domain decomposition. The analogy of this strategy in the three-dimensional case would be to produce boundary points with surface meshes without using octree decomposition separate from the generation of interior points based on domain decomposition.

The generated points on the edges of a domain should reflect the geometric features and the physical behaviour of the problem concerned. This can be achieved by using the mesh density distribution method, which will be discussed in Section 4. Also, the mesh density criterion used for the boundary point generation should correlate with that used in generating interior points.

3. Domain decomposition

For the automation of adaptive finite element analyses, one extremely important feature is the control of mesh density, which can be achieved via the use of domain decomposition. The size of the finite elements are governed by the level of the tree in which the final cells are constructed (see Figs. 1 and 2 for two-dimensional cases). In an automated environment, the sequential analysis procedures provide a density requirement for the subsequent meshes, for which the domain needs to be subdivided at different tree levels. Fig. 3 illustrates the results of using the two aforementioned quadtree schemes.

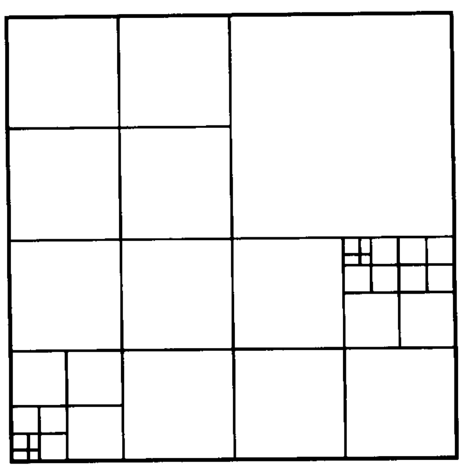


Fig. 1. Schematic of tree structure: square quadtree in two dimensions.

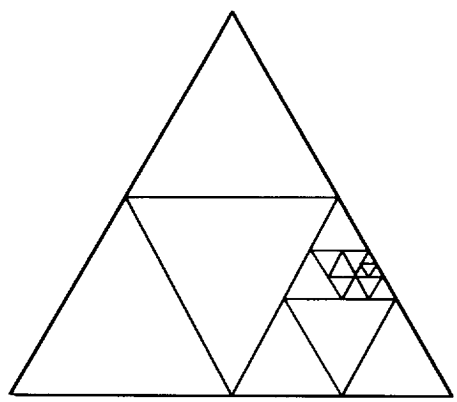


Fig. 2. Schematic of tree structure: triangle quadtree in two dimensions.

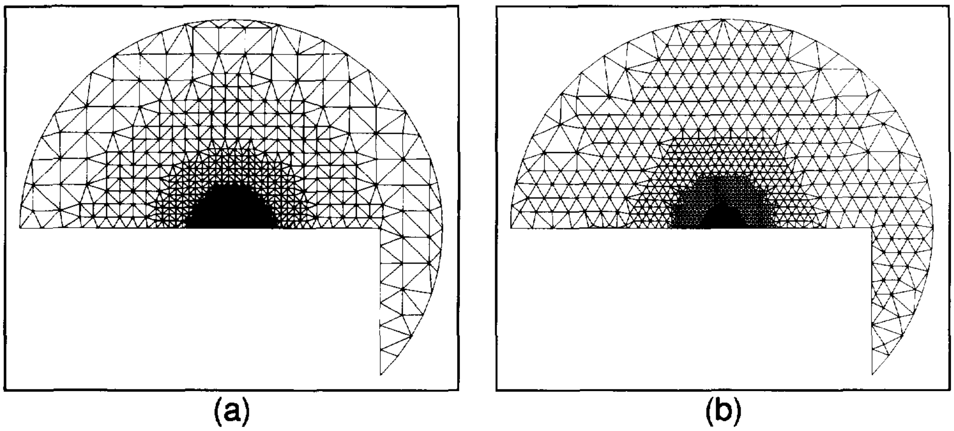


Fig. 3. Two meshes using: (a) square quadtree, (b) triangle quadtree.

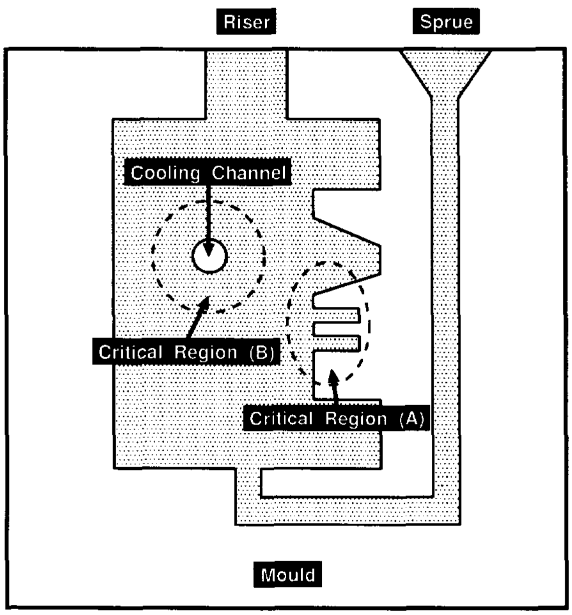


Fig. 4. The schematic diagram of a casting example.

The density can be specified according to the physical behaviour of the problem and the geometry features for the initial finite element analysis. The experience of the authors in the simulation of casting processes indicates that a mesh can be properly constructed based on information of critical regions, a mesh density distribution and a basic mesh size. A schematic illustration is given in Fig. 4. The critical regions (or points, or surfaces) can be determined by the geometrical features (Critical Region A), and boundary conditions (Critical Region B). The critical regions will be identified together with the mesh density distribution. The basic mesh size is governed by both geometrical features and the numerical accuracy desired.

3.1. *Quadtree*

In two-dimensional cases, domain decomposition takes the form of a quadtree structure, in which the corner points or the central point of each cell may be chosen as nodal points of a finite element mesh. The two kinds of cell shapes involved in quadtree, square-shaped and triangle-shaped, can be used as two alternatives. Square quadtree is the most commonly used shape of quadtree in many application areas. All four subcells in a cell are again squares of half the length of the parent cell (Fig. 1). Triangle quadtree is a recent innovation. All four subcells in a cell are triangles of half the size of the parent cell. From the locations of the points created, it can be seen that the triangle quadtree results in a mesh of better quality in the interior of the domain than does the square quadtree method.

3.2. *Octree*

For three-dimensional problems, the corresponding domain decomposition is referred to as octree decomposition, and the cells are cubes or tetrahedra, which are straightforward extensions of squares and triangles respectively.

Fig. 5 illustrates an encoding scheme for tetrahedral octree. Note that the sides drawn in thick lines are $\sqrt{2}$ times longer than those of the corresponding thin lines. A tetrahedron is defined as

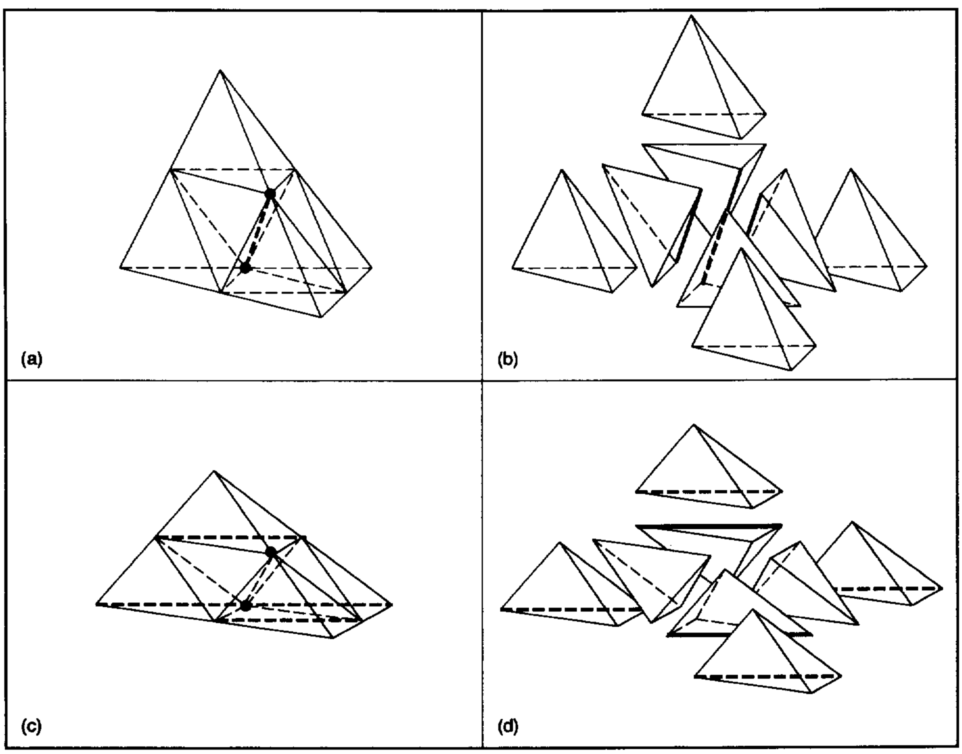


Fig. 5. Encoding scheme of tetrahedral octree: (a) a regular tetrahedron; (b) splitting of regular tetrahedron; (c) an irregular tetrahedron; (d) splitting of irregular tetrahedron.

being "regular" if its six sides are of equal length; otherwise, it is referred to as being "irregular". A regular tetrahedron octant can be split into four regular and four irregular tetrahedral sons, while an irregular tetrahedron octant can be split into two regular and six irregular tetrahedral sons [5]. The tetrahedral-octree encoding is more difficult than hexahedral-octree encoding. However, it is expected that tetrahedral octree will lead to a better mesh than achieved via cubic octree regardless of the elements near the boundaries.

3.3. *Removal of unnecessary points*

For the spatial domain of the problem concerned, the tree structure starts with an initial cell which covers all the subdomains. The subsequent decomposition will depend upon the density distribution with the resulting cells being either inside, outside or crossing the boundary. Some adjustment schemes can be used to move the points of certain cells to match the boundary. However, in the present scheme, the cells are only used to locate the interior nodal points for the mesh, in which case no adjustment is necessary to move the points of cells straddling the subdomain boundary.

Once the cells representing the density requirement are constructed, points can be located to define the possible mesh nodes. However, some of these may be outside or very near to the boundary. Therefore, a strategy has been designed to remove these points with reference to the geometrical information (in/out) and the rest of the points are assigned as the interior nodal points of the subdomain being processed. This is repeated for all the subdomains.

3.4. *Comments*

Since a "quadtree" or "octree" structure represents a collection of hierarchically structured cells which are tied together through the use of a tree structure, they have the advantage of being easily created and accessed, and also provide a useful data structure for the mesh generation procedure.

However, the distribution of generated points is such that the real density of the points changes step by step, therefore the spacing of the resulting meshes do not transit smoothly from one location to the other even if the desired density varies continuously. This is an essential drawback of the point generation scheme based on this domain decomposition approach.

Alternative methods can be designed to create the points which are generated during the Delaunay triangulation process, where every point is located in the centroid or circumcircle center of a previously formed triangle. This scheme will improve the speed of the mesh generation process significantly. However, in this approach, the density specification should be treated with caution, as may be observed in the next section.

4. Density specification

Recursive subdivision is controlled by the mesh density requirement as specified by the finite element result and the geometric features of the object concerned. Generally for the finite element method, a more accurate result is obtained with a finer mesh.

In order to reduce computing costs, an "adaptive" finite element scheme can be used, in which each mesh is constructed to match the density distribution as calculated from discretization error estimates

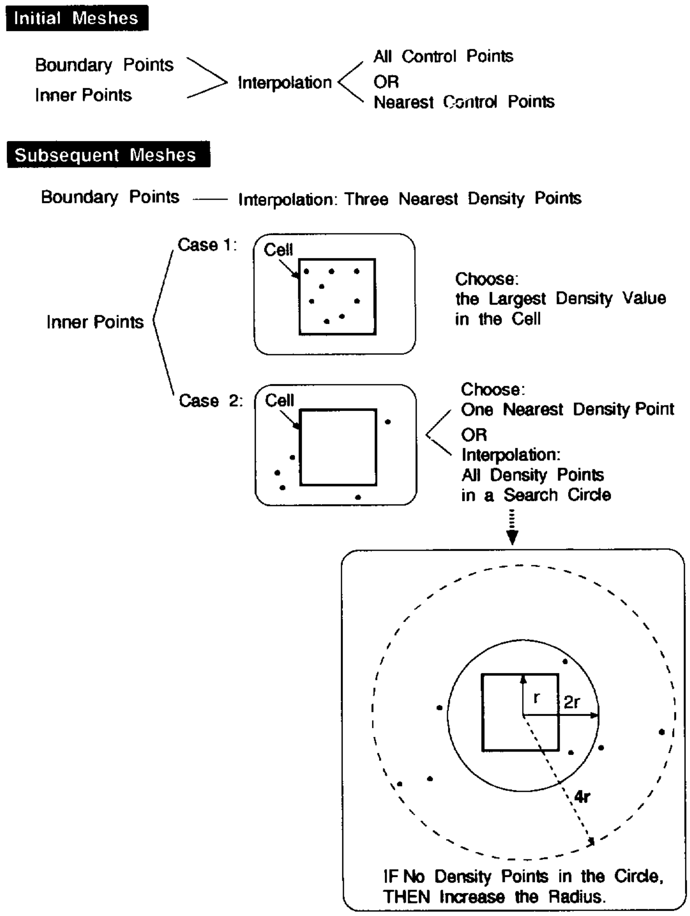


Fig. 6. Density specification.

based on previous analyses. The numerical analysis proceeds in an adaptive mode with a sequence of meshes based on the results of the error analysis. Fig. 6 illustrates the density specification scheme used in the present work for adaptive analyses, which is to be outlined as follows.

4.1. Initial meshes

The sequence of an adaptive analysis requires an initial specification of a basic mesh size and control points assigning local mesh densities. These control points and the corresponding densities should be appropriately located to best satisfy the requirement for an accurate analysis.

In order to allow the mesh generator to make a suitable judgment regarding further cell subdivision, the density at a chosen point of the cell must be calculated. Many interpolation methods can be used to determine the density over the domain [6]. The following one is of particular interest, which can be written as

$$d_p = \sum_{i=1}^{n} \frac{d_i}{r_i^{\alpha}} \Big/ \sum_{i=1}^{n} \frac{1}{r_i^{\alpha}} \quad (1.0 \leqslant \alpha \leqslant 2.0), \tag{1}$$

where n control points are involved and r_i is the distance between the current point and the control point i (see Fig. 7). Generally, only the nearest control points should be involved instead of all the control points in the domain, in order to reflect the local features.

In an analogous way to the method of using point sources as control functions with elliptic partial differential equations, it is possible to define line and point sources to provide density control for unstructured meshes (see [7]).

4.2. Subsequent meshes

The mesh density required is an alternative indicator of estimated error. The mesh density values are given at the nodal points on the previous mesh (i.e. background mesh). A reasonable scheme should be designed to establish a mapping from the nodal values to the value at any concerned point within the domain. Using the same shape functions as those embedded in the corresponding finite element implementation, a density value can be obtained (cf. Fig. 6). Alternatively, direct interpolation can be used from the density values of the points within the neighbourhood of the point concerned.

The judgement whether a cell should be further subdivided is a sensitive decision in the point generation stage. Some researchers use density values at the corner points of cells, whilst others employ those at central points of cells.

In the present work, it was observed that these schemes have drawbacks which sometimes resulted in undesirable meshes. Fig. 8 illustrates some cases where the above schemes will fail to perform further subdivisions. Fig. 9 shows an example exhibiting the behaviour of these schemes with reference to a basic mesh size. For this example, the mesh density distribution and location of the high mesh density points are given in Fig. 10. Therefore, a relatively better scheme was adopted in which all the density points within the current cell are considered, and the maximal density value within the cell was used as a criterion for subdivision. In the case where there was no density point within the cell, the nearest density point was used.

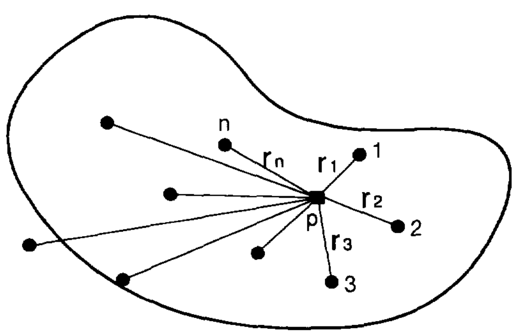


Fig. 7. A general interpolation scheme.

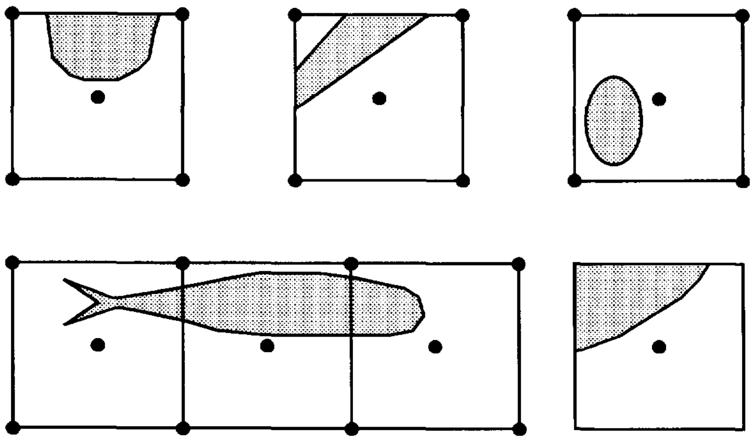


Fig. 8. Some failure cases of point generation schemes based on the densities calculated at the corner points or the central point of a cell.

5. Delaunay triangulation

5.1. The algorithm

Delaunay triangulation is one form of triangulation in which each element of the mesh satisfies the Delaunay criterion, i.e. the circumcircle (circumsphere in three dimension) of the element contains no other points. This form of triangulation of a set of points will result in a convex hull, which covers all these points. There are many algorithms available which will construct this kind of triangulation [8–11].

All the boundary and interior points are sorted to determine their range and a bounding convex polygon (convex polyhedron in three dimension) is created to initialize the triangulation. Subsequently, boundary recovery and the removal of unwanted elements are executed to obtain the desired mesh.

5.2. Robustness issue

When the Delaunay triangulation scheme is implemented, one is faced with issues pertaining to the robustness of the mesh generation as degeneracy may occur in the triangulation stage. One of the most common problems is the case where four points lie on a circle (five points lie on the surface of a sphere in the three dimensional case). The points generated from the aforementioned tree structure result in a regular distribution. Therefore, some degree of perturbation is required for shifting the interior points slightly in order to reduce the possibility of degeneracy. The perturbation can be formulated in either a deterministic or random form. In the present work, the perturbation takes the following form for two dimensions (see Fig. 11), while a similar analogy can also be formulated for the three-dimensional case:

$$x_i' = x_i + \lambda C_i \cos(i\alpha), \qquad y_i' = y_i + \lambda C_i \sin(i\alpha), \qquad \alpha = \pi/\beta, \tag{2}$$

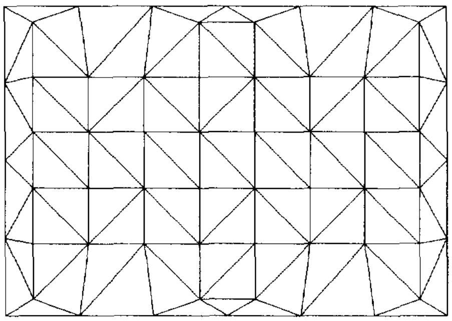
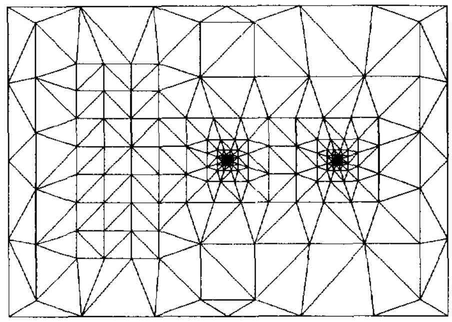
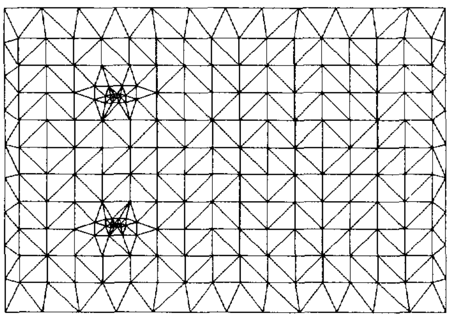
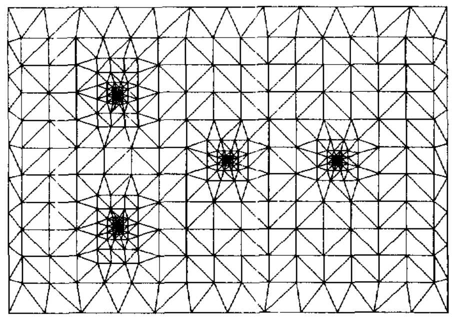
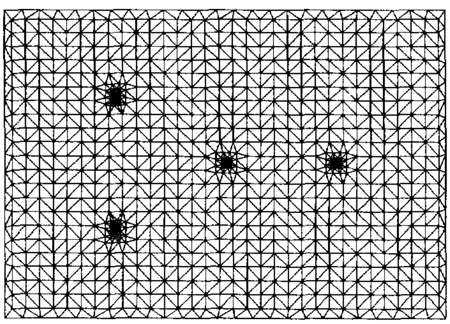
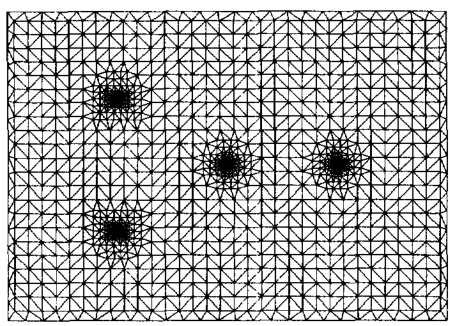
Basic mesh size ($\times 10^{-2}$)	According to the density at the central point of a cell	According to the densities at the corner points of a cell
4.0		
2.0		
1.0		

Fig. 9. Behaviour of subdivision schemes associated with reference mesh sizes.

where β is an irrational coefficient which prevents duplicate movement of these interior points; λ is a control coefficient of value much less than unity; and C_i is a local spacing scale related to point i. If the aforementioned domain decomposition is used to create the interior points, C_i can be chosen to be equal to the size of the cell containing point i.

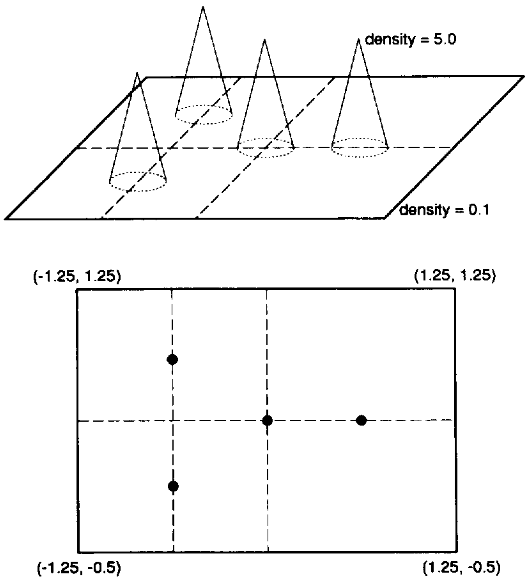


Fig. 10. Mesh density distribution and location of the high mesh density points.

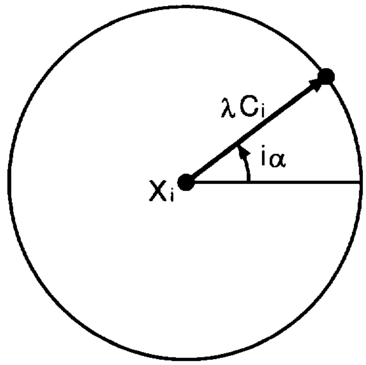


Fig. 11. Perturbation scheme for all interior points.

The perturbation procedure may be used in two ways. One is to perturb all the points before triangulation. The other is to perturb those points which lead to degeneracy during triangulation in a recovery stage. In the authors' experience this technique was found useful to recover from degeneracy in many cases.

There is a possibility that the boundary points are distributed regularly, which again leads to degeneracy. An existing triangle is rejected during triangulation if a point is formed to lie in its circumcircle. A point X_i is contained within the circumcircle of the triangle, centered at X_o with radius R_o, if

$$||X_i - X_o|| < R_o. \tag{3}$$

However, due to round-off error, this can be redefined by introducing a small value ε. If there exists

$$||X_i - X_o|| < (1 - \varepsilon)R_o, \tag{4}$$

then the point X_i is contained within the circumcircle. Likewise, if

$$||X_i - X_o|| > (1 + \varepsilon)R_o, \tag{5}$$

then the point X_i is outside the circumcircle. From Fig. 12, it can be seen that there is a shaded annular region. No decision can be made about points if they are located in this region.

Although a compatible mesh can be produced in the degenerate case by control of the parameter ε, the resulting mesh is not unique and will depend on the sequence of inserting the points. During the triangulation process, a point can be perturbed or dealt with at a later stage, when it is located within the annular region with respect to a triangle.

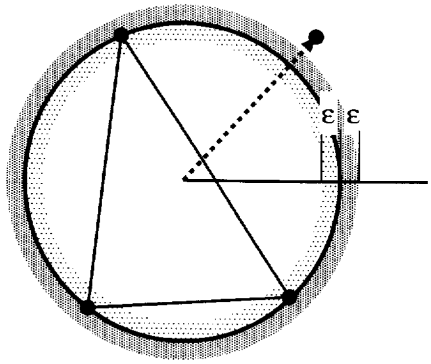


Fig. 12. Tolerance ε.

It should be mentioned at this point that degeneracy occurs when three (four) points are collinear in determining the centre of their circumcircle (the corresponding circumsphere in three dimension). In this case, the centre will be at infinity, however, this case can be removed by temporarily rejecting the current point and recalling it for insertion at a later stage.

6. Boundary integrity

In realistic engineering problem, mesh generation often involves complex shapes, and boundary integrity become an important aspect [10,12–14]. Two procedures which may be employed are outlined as follows.

6.1. Element rejection

There are essentially two methods of dealing with the missing boundaries. The first is to insert a temporary point for every missing edge as an aid for the procedure, while the second is to directly check and recover [13,14]. Both these procedures may result in rejection of some elements and addition of new ones.

6.2. Boundary recovery

The procedure outlined results in a triangulation which covers the geometry. It is necessary to delete all triangles (tetrahedra) which are located outside the subdomains concerned. The directionality of the boundary edges (faces), and therefore the directionality of each triangle (tetrahedron) is related to the corresponding area (volume). If the area (volume) is negative, then the triangle (tetrahedron) lies outside the subdomain and should be deleted. However, there are other procedures which can be used based on the geometrical data structure of the triangulation and the boundary edge (face) information.

7. Benchmark tests

7.1. Basic meshes

A series of benchmark tests have been carried out as initial meshes for some real problems. Fig. 13 illustrates the corresponding meshes for the typical problem shown in Fig. 4, while Figs. 14, 15 and 16 demonstrate the basic meshes for some real casting problems.

7.2. Adaptive meshes

A typical benchmark problem of convection dominated heat transfer is illustrated in Fig. 17 [15]. The differential equation governing steady convective-diffusive heat transfer can be written as

$$u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \nabla \cdot k \nabla T, \tag{6}$$

in which k is thermal conductivity; u and v are the Cartesian components of the velocity vector; and T is temperature. In this problem, an assumed flow field is defined, which is of unit magnitude and constant with time. The inflow boundaries are specified as shown in the figure. The outflow boundaries are considered homogeneous (i.e. diffusive flux is zero, which of course implies no temperature gradient). The value of θ in Fig. 17 for this example has been chosen as 30°. The value of k is set to 10^{-6}, and this low value of conductivity ensures a convection dominated problem with a Peclet number of 10^6.

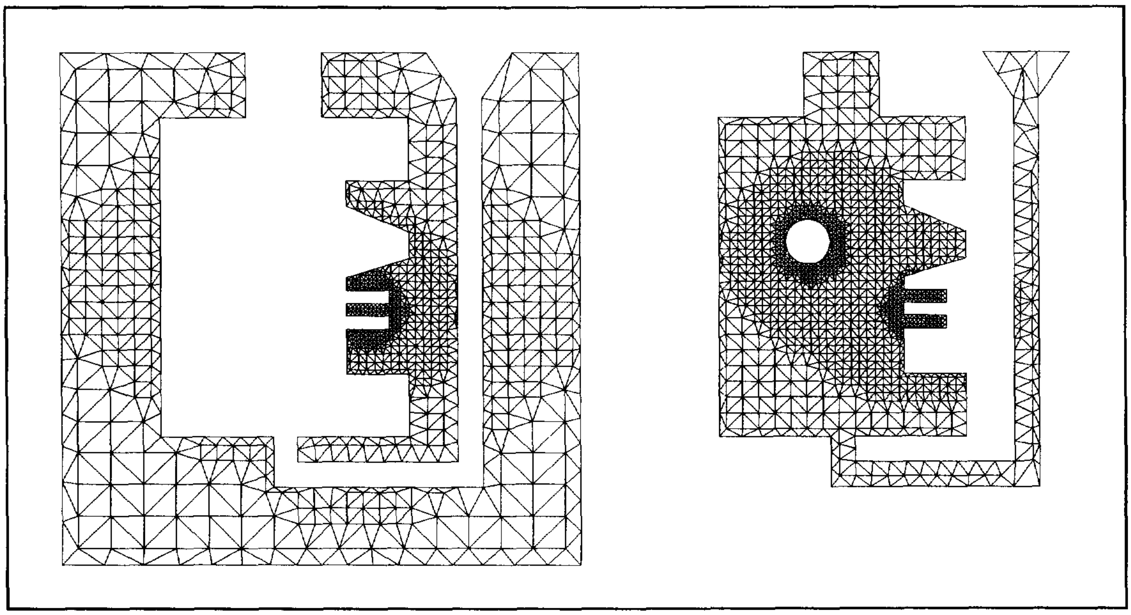
This problem is solved by combining the adaptive mesh refinement technique with the streamline upwind Petrov-Galerkin (SUPG) method [15]. Fig. 18 shows a sequence of meshes which were automatically generated based on a specified target error of 20%. The last mesh at which the target error was achieved clearly shows a band of very fine elements marking the region of the highest gradient. All these meshes are smoothed after triangulation, in a manner which will be discussed in the next section. The solutions from the first and last meshes are shown in Fig. 19 in the form of line contours. The quality of the adaptive solution is obviously far superior. The converged adaptive solution is again shown in Fig. 20(b) in comparison with the exact solution of Fig. 20(a). The viewing direction is as shown in Fig. 17. It may be observed that the adaptive solution is very close to the exact one.

8. Quality improvement

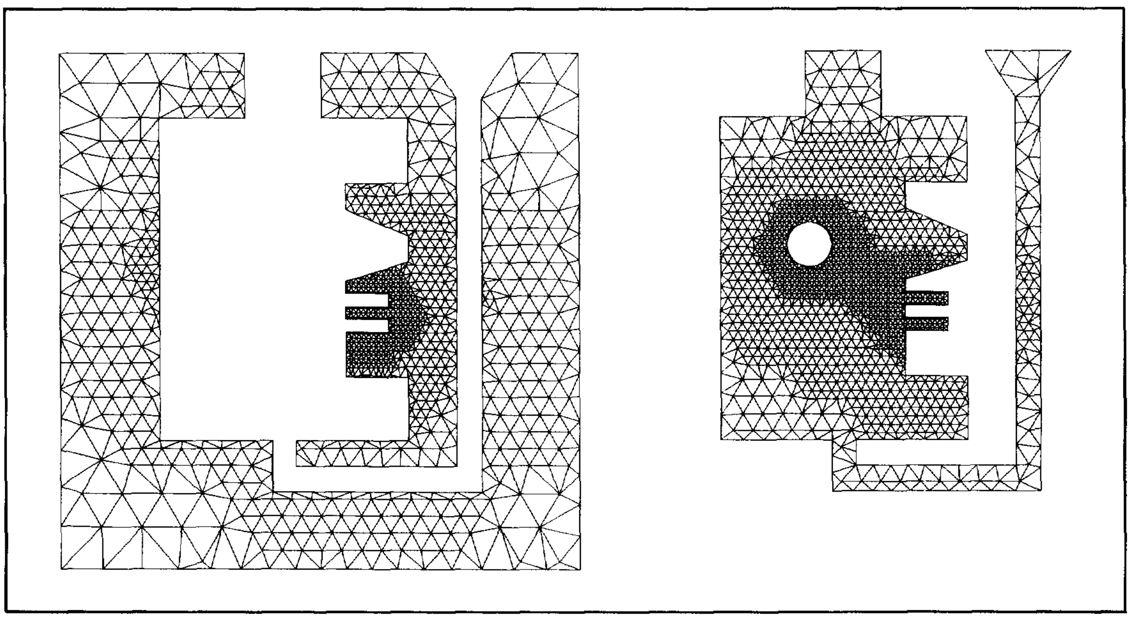
It is well known that the meshes generated by square quadtree include mainly right-angled triangles which perform worse than elements with nearly equal angles. However, the mesh quality can be improved by means of the Laplace smoothing technique [16], and is demonstrated in Figs. 21–23. In Fig. 23, the meshes are associated with density distribution $d(x, y)$ in the following forms

$$d(x,y) = 1/[0.05f(x,y) + 0.15], \qquad f(x,y) = 100(y - x^2)^2 + (1 - x)^2, \tag{7}$$

where $x \in [-1.25, 1.25]$, $y \in [-0.5, 1.25]$.

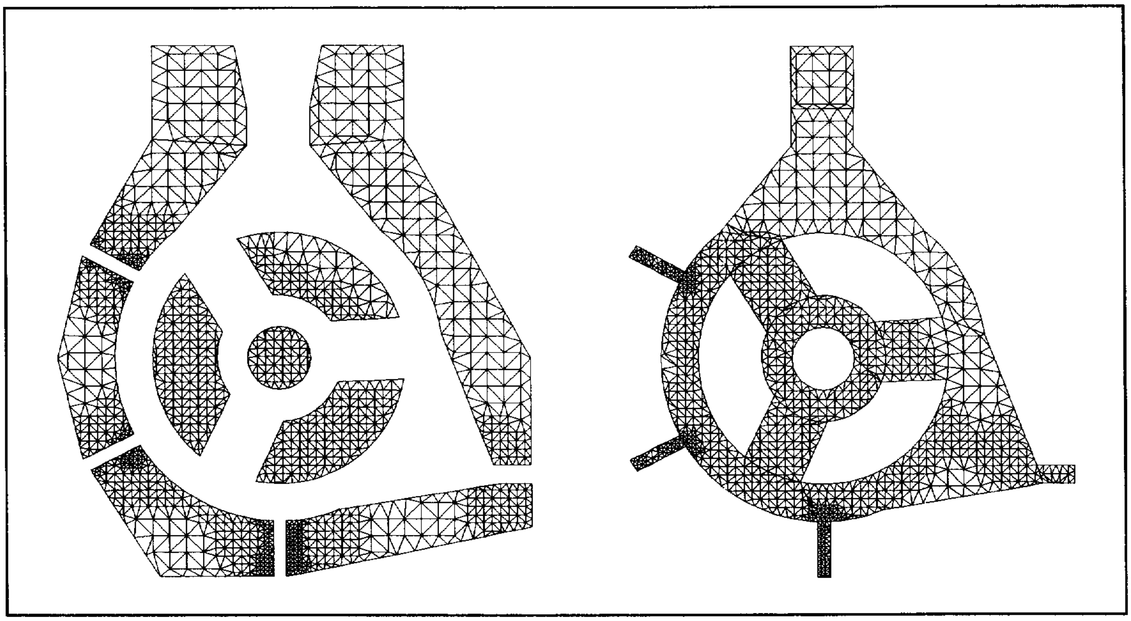


(a)

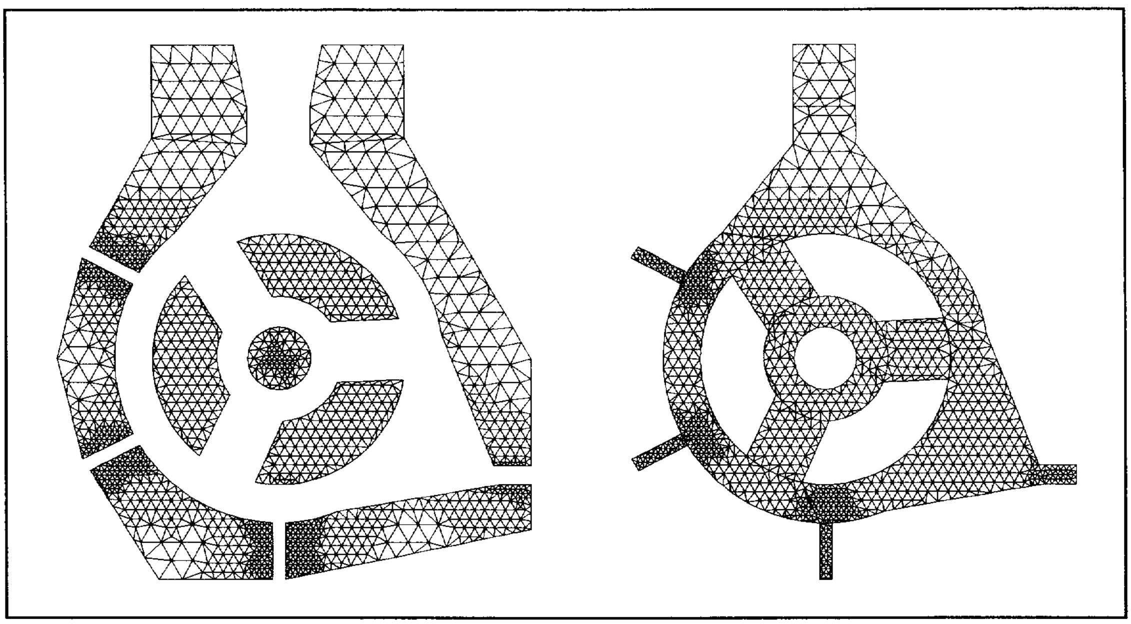


(b)

Fig. 13. Benchmark test 1: (a) inner points generated using square quadtree; (b) inner points generated using triangle quadtree.



(a)



(b)

Fig. 14. Benchmark test 2: (a) inner points generated using square quadtree; (b) inner points generated using triangle quadtree.

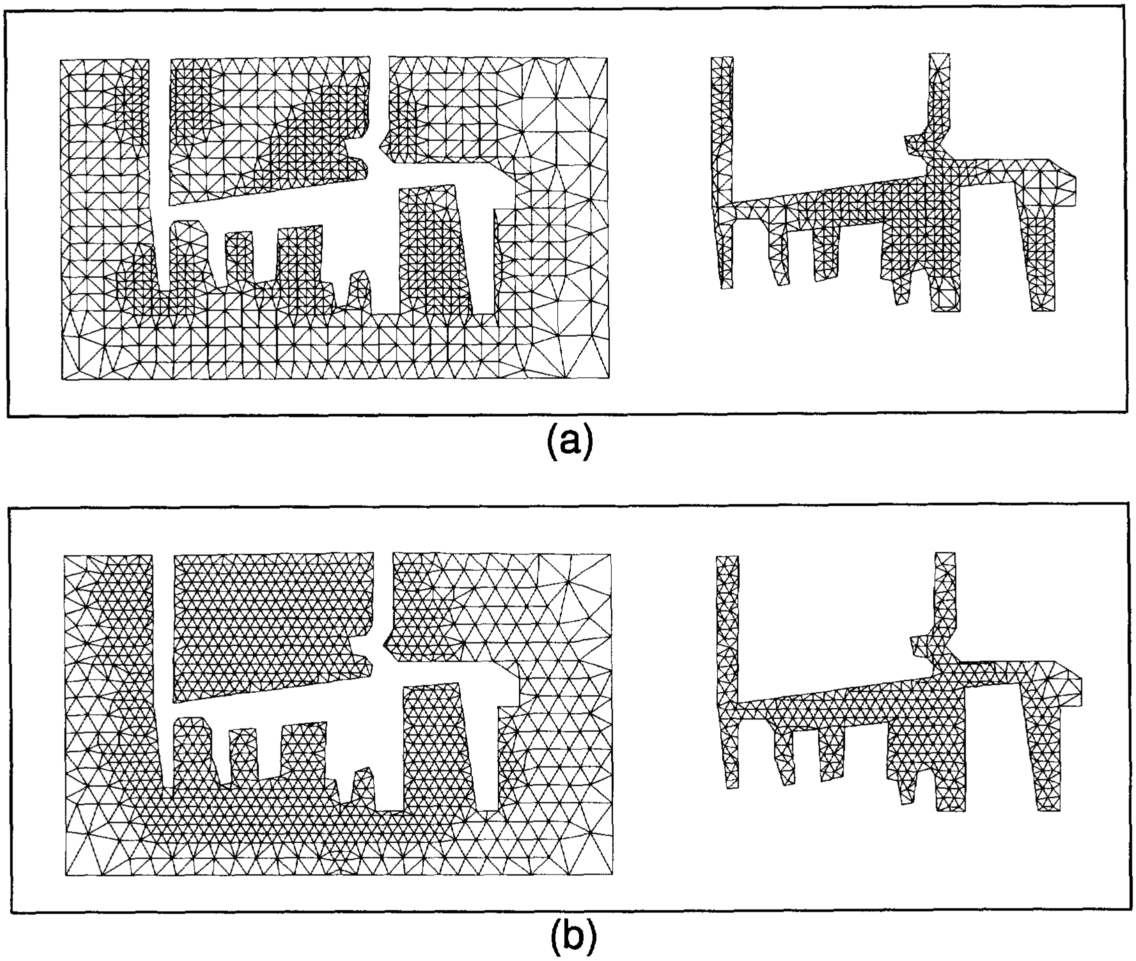


Fig. 15. Benchmark test 3: (a) inner points generated using square quadtree; (b) inner points generated using triangle quadtree.

In the Laplace smoothing procedure, the position X_i of an interior node i is positioned to satisfy the expression

$$X_i = \frac{1}{n_i}\sum_{j=1}^{n_i} X_j \quad (i = 1, \cdots, m), \tag{8}$$

where n_i is the number of nodes connected to node i, X_j is the position of the connected nodes, and m is the total number of interior nodal points. This equation places the interior nodes at the centroid of the connected nodes. This smoothing process can be performed many times if required.

Sometimes, the smoothing can be problematic as an interior point may be moved away from the domain (see Fig. 24), or result in an element having a negative area. In this case, an alternative method may be used to provide greater artificial control on the smoothing process, in

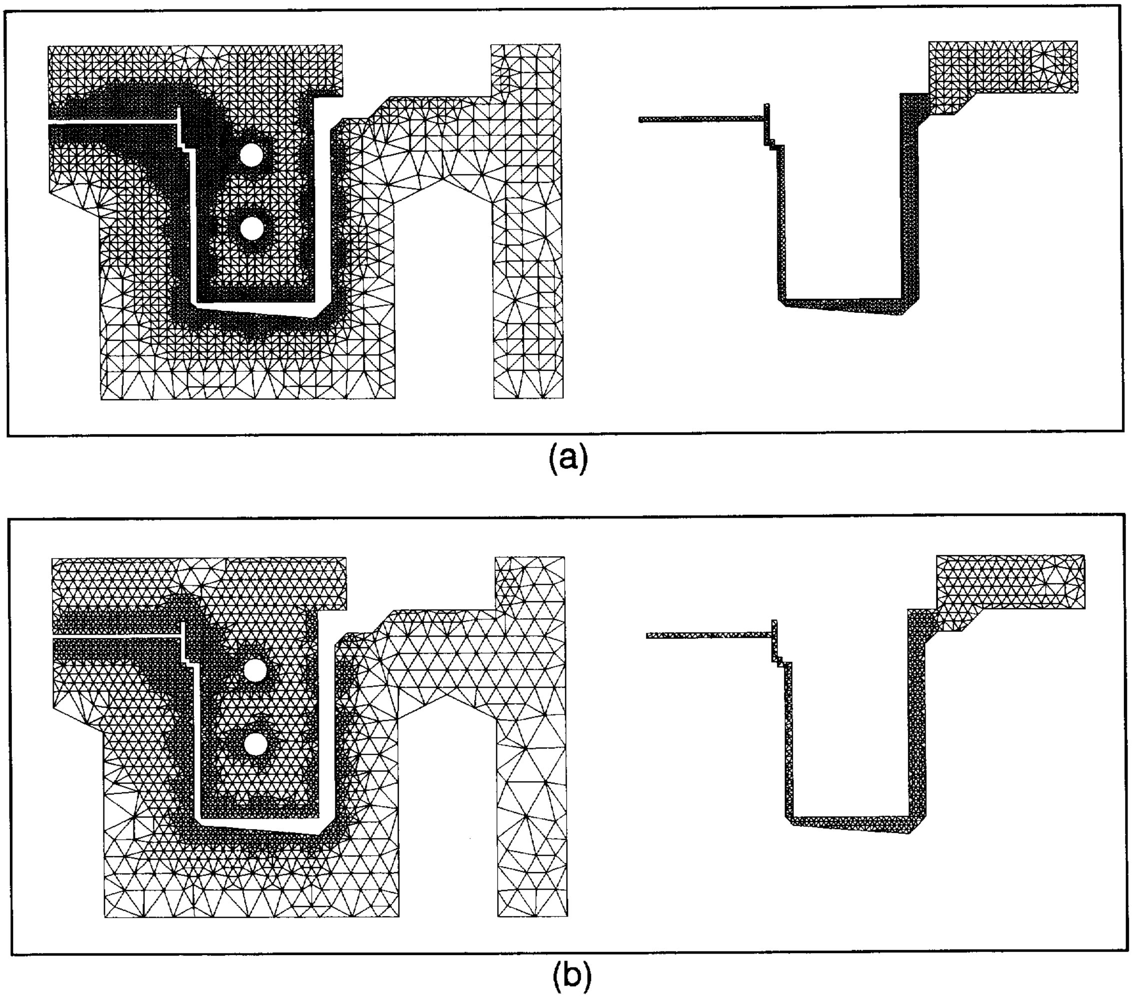


Fig. 16. Benchmark test 4: (a) inner points generated using square quadtree; (b) inner points generated using triangle quadtree.

which a relaxation parameter ω is introduced (e.g. [10]). The corresponding equation can be written as

$$X_i^{k+1} = X_i^k + \frac{\omega}{n_i}\sum_{j=1}^{n_i}(X_j^k - X_i^k) \quad (i = 1,\cdots,m), \tag{9}$$

where the superscript k denotes the iteration number. This algorithm keeps X_i^{k+1} different from X_i^k before a smoothing iteration completes, and it differs from that of Eq. (8), where X_i will be overwritten during a smoothing iteration. Unfortunately, these expressions cannot guarantee that an interior node will not be moved which will violate the above geometry requirement. For a mesh of better quality there is a reduced probability of such violations. A checking procedure of the mesh structure must be conducted subsequent to smoothing in order to avoid this problem.

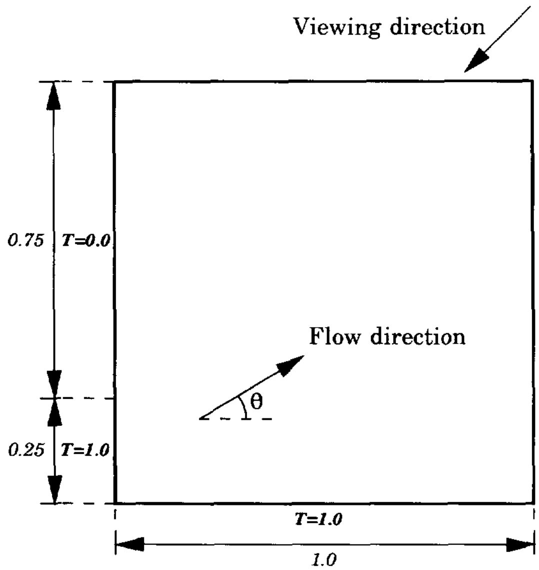


Fig. 17. Benchmark test 5: a problem of convection dominated heat transfer.

9. Conversion from triangles to quadrilaterals

All the meshes constructed in the previous examples consist of triangles. However, quadrilateral elements normally give better numerical performance for finite element analyses. Therefore, several techniques which convert triangular meshes to quadrilateral meshes have been proposed (e.g. [17,18]).

Basically, there are two simple conversion schemes as shown in Fig. 25. Fig. 25(a) shows that every triangle can be split into three quadrilaterals by connecting the middle points on the sides with the centre point of the triangle. The other scheme is illustrated in Fig. 25(b) where two neighbouring triangles have been combined to form one quadrilateral. If the resulting quadrilateral is convex, then it can be further split into four quadrilaterals. However, this strategy cannot create a complete quadrilateral mesh from one which originally consisted of an odd number of triangles. Also for the case of a mesh with an even number of triangles, islands of isolated triangles may be produced leading to a mixed mesh consisting of quadrilaterals and triangles.

In the present work, a facility, based on combining triangles, splitting quadrilaterals and quality sorting has been established, as illustrated in Fig. 26. Every pair of triangles of the mesh concerned can be given an indicator which relates to the quality of the corresponding quadrilateral in terms of the angles. For every triangle inside a domain, there exist three triangle pairs which are related to it. In the case of a triangle adjacent to a boundary, there are one or two triangle pairs related it. A sorting process is then completed resulting in an ordering of these pairs with respect to the quality indicators. At each step, the pair of triangles with the largest quality indicator is combined and converted into four quadrilaterals. Many triangles will be left unpaired due to the pairing of all neighbouring triangles with other triangles. This combining and splitting process will finally leave

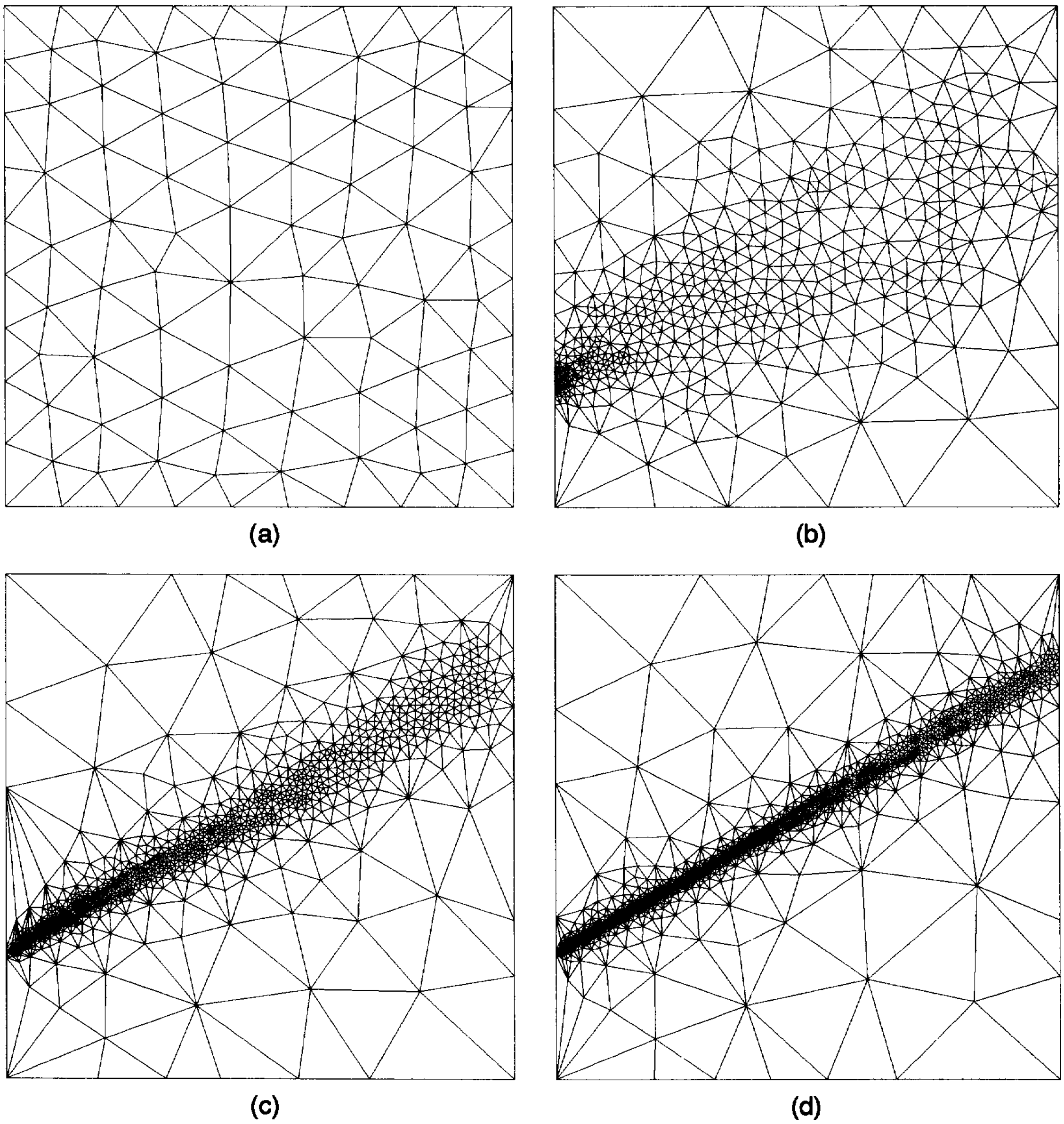
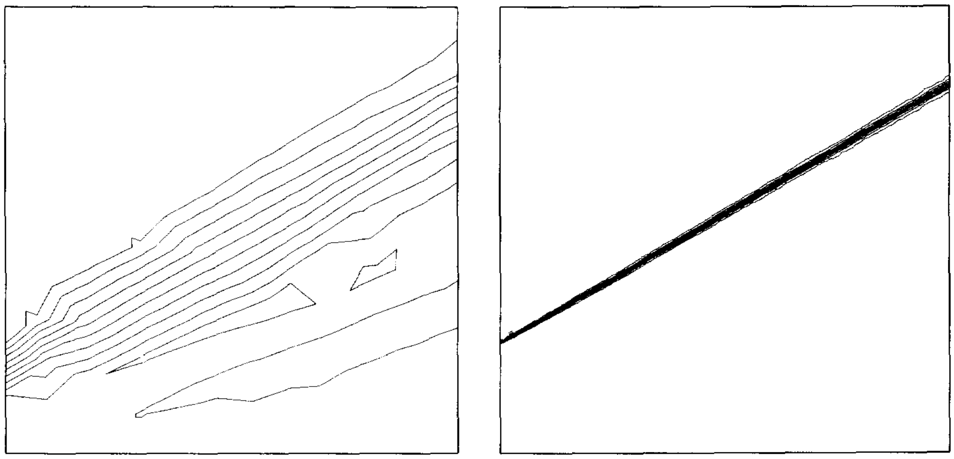


Fig. 18. (a) The first uniform mesh, (b), (c) two subsequent adaptive meshes, and (d) the last mesh at which the target error was achieved.

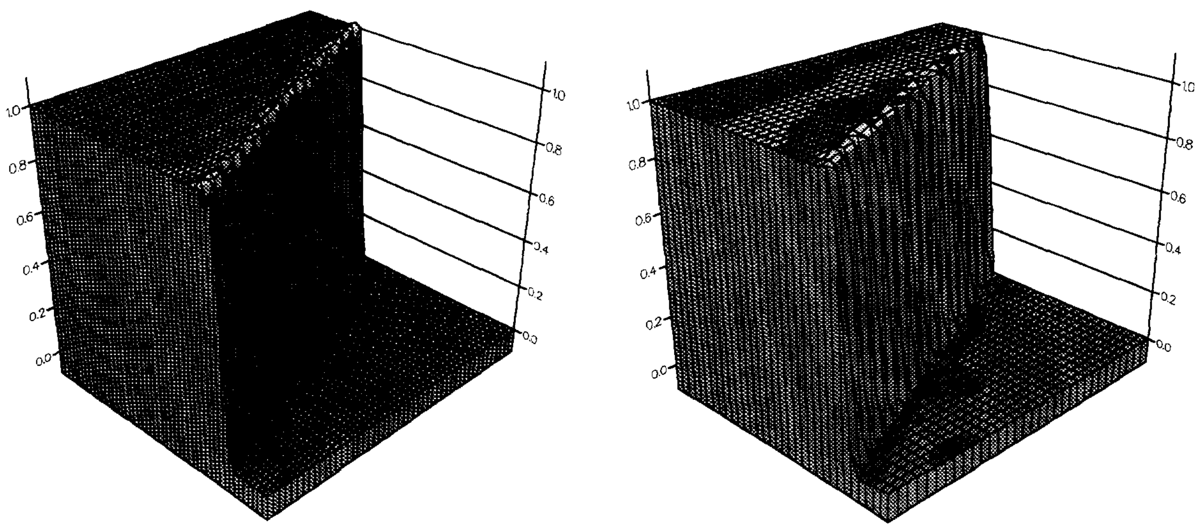
some islands of isolated triangles, which can be split into three quadrilaterals and locally smoothed. Fig. 27 demonstrates the conversion scheme proposed in this paper.

It should be mentioned that the total element number is at least doubled on using this conversion scheme. Also, the analogy of this strategy for tetrahedral meshes in three dimensions is not obvious.



(a) (b)

Fig. 19. (a) Solutions corresponding to the first uniform mesh and (b) the final adaptive mesh.

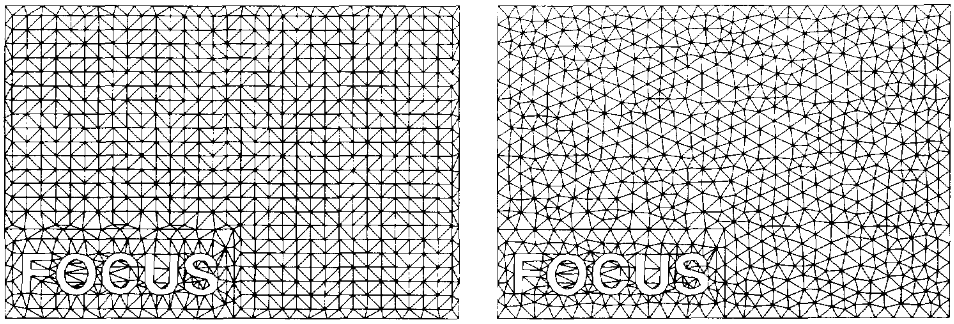


(a) (b)

Fig. 20. Three-dimentional illustration for comparison of the exact solution (a) with the final adaptive solution (b).

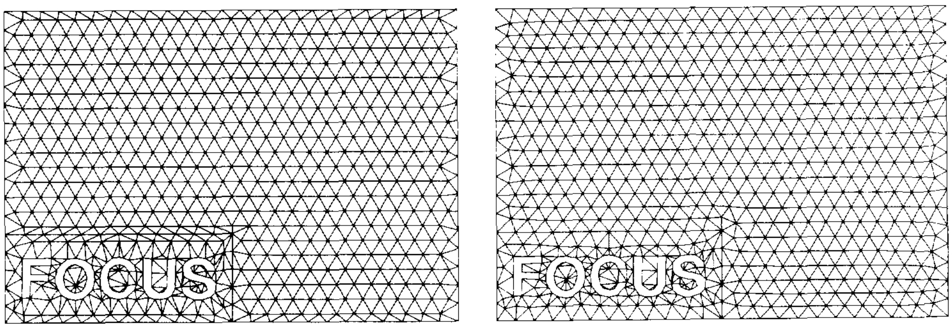
10. Discussion

Recently there have been some applications of artificial intelligence techniques for mesh generation which appear to assist the user to some extent. Some rules for deciding control densities for finite



(a) (b)

Fig. 21. The smoothing process: (a) a mesh generated using square quadtree; (b) the corresponding smoothed mesh (five smoothing iterations).



(a) (b)

Fig. 22. The smoothing process: (a) a mesh generated using triangle quadtree; (b) the corresponding smoothed mesh (five smoothing iterations).

element meshes can be inductively constructed from examples provided by experts [19]. The expert system generated in [20] was able to intelligently identify critical regions and choose a proper mesh size. Using a neural network technique, a system was developed to predetermine the mesh density. This system can be trained by incorporating examples of ideal meshes [21,22].

11. Conclusions

The present paper has addressed our experience on mesh generation based on domain decomposition and Delaunay triangulation techniques. The domain decomposition technique has the advantage of being easily created and accessed due to the use of a tree structure, however, the spacing of the resulting meshes do not transit smoothly from one location to the other, which is the essential drawback. In the two dimensional case, the triangle quadtree results in a mesh of better quality in the interior of the domain than does the square quadtree method. By analogy in the three dimensional case, tetrahedral octree is expected to lead to a better mesh than achieved via cubic octree regardless of the elements near the boundaries.

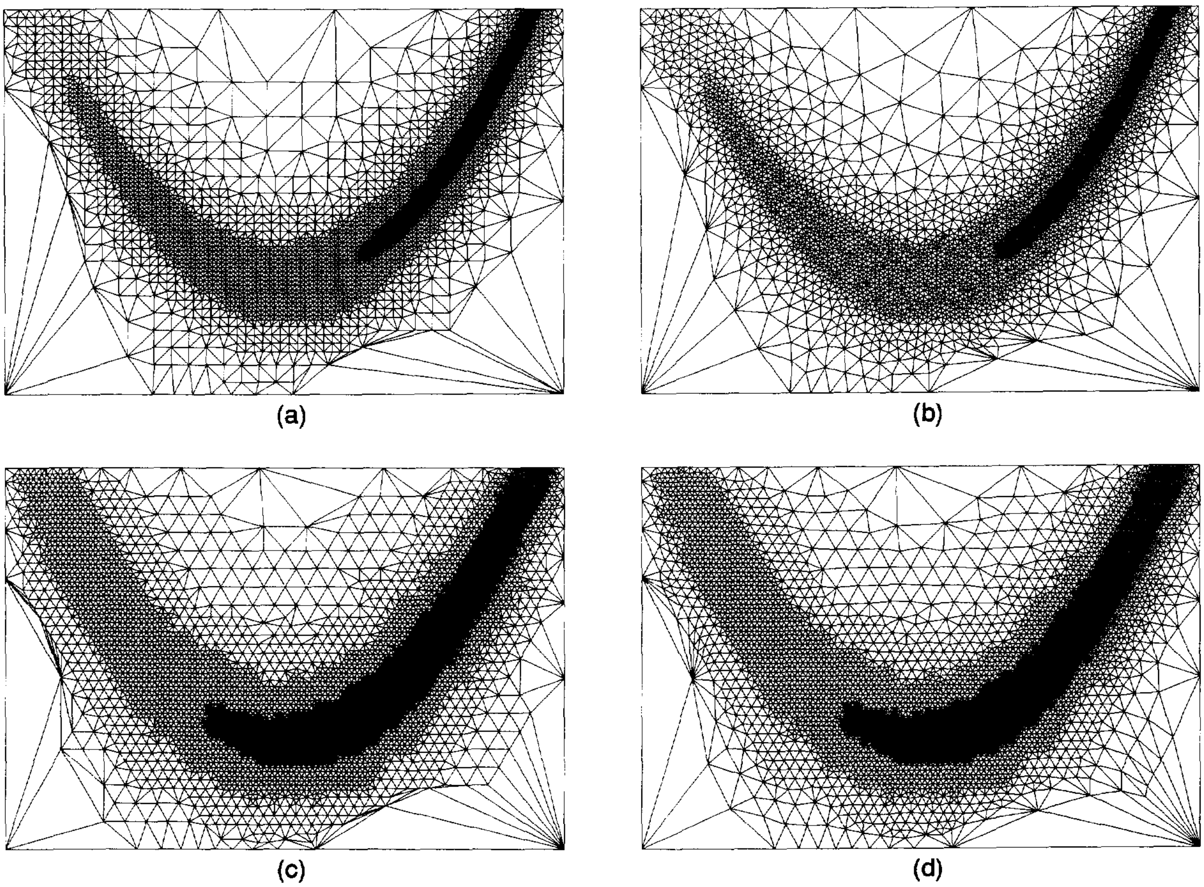


Fig. 23. Meshes with a density distribution in the form of the banana function: (a) and (b) are a mesh generated using square quadtree and its corresponding smoothed mesh (three smoothing iterations); whilst (c) and (d) are a mesh generated by means of triangle quadtree decomposition and its corresponding smoothed mesh (three smoothing iterations).

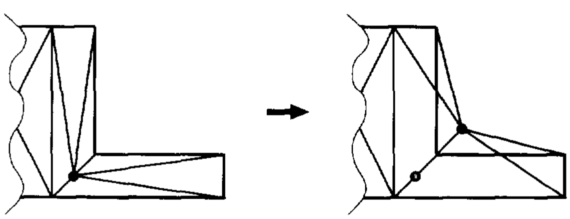


Fig. 24. A failure case of the Laplace smoothing technique.

The recursive subdivision is controlled by the mesh density requirement as specified by the finite element analysis result and the geometric features of the object concerned. The judgement whether a cell should be further subdivided is a sensitive decision in the point generation stage. The maximal density value within the cell should be used as a criterion for subdivision, and in the case where there is no density point within the cell, the nearest density point should be used.

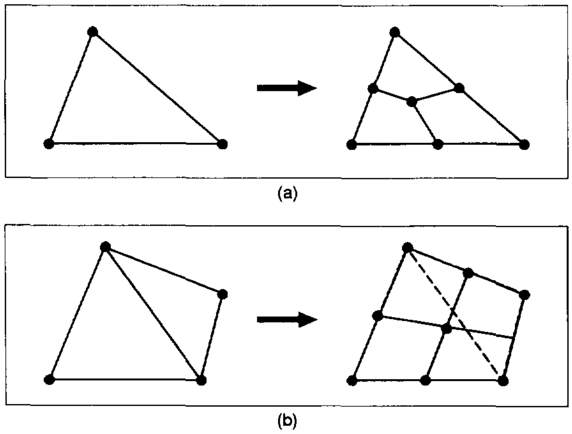
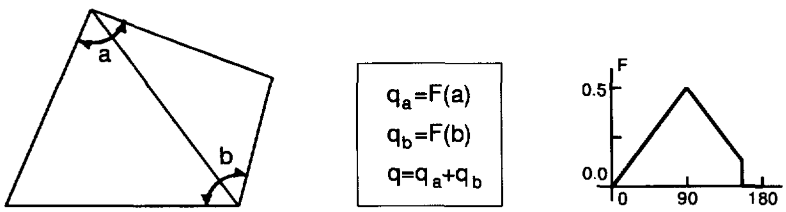


Fig. 25. Two basic schemes of the conversion.



Scheme

(1) Each Step, Convert the Triangle Pair with the Largest q, into Four Quadrilaterals;
(2) Convert Each Triangle Island into Three Quadrilaterals;
(3) Smooth on Quadrilaterals Created from Triange Islands.

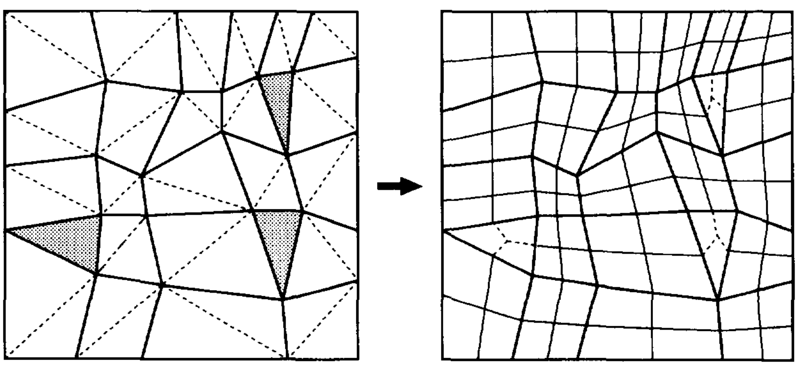
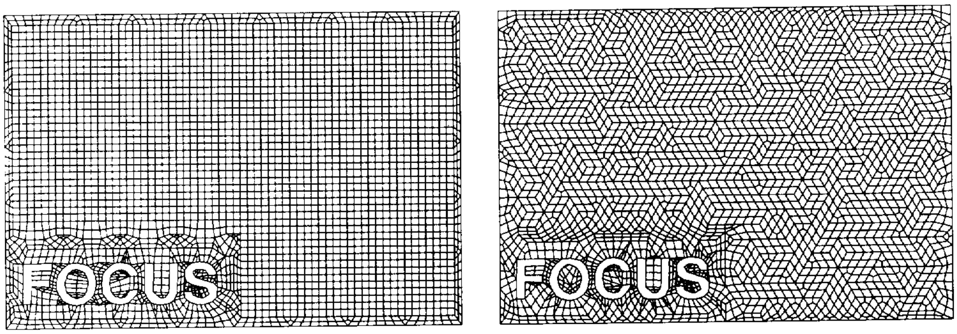


Fig. 26. Conversion from triangles to quadrilaterals.



(a) (b)

Fig. 27. Meshes converted from those shown in Figs. 21(a) and 22(a).

When Delaunay triangulation is used within a mesh generator, the question of robustness is important for the mesh generator to be used in real engineering applications. Boundary recovery algorithms provide the possibility of ensuring the integrity of the boundary of real geometries. Benchmark tests such as the ones illustrated, may be used to demonstrate the validity and the efficiency of a mesh generator.

It has been pointed out that a checking procedure for the mesh should be conducted when the smoothing process is used to improve the mesh quality.

Acknowledgement

This work has been carried out under partial financial support by the Science and Engineering Research Council of the United Kingdom (Grant Number GR/G 35107), which is gratefully acknowledged.

References

[1] M.S. Shephard, "Approaches to the automatic generation and control of finite element meshes", *Appl. Mech. Rev.* **41**(4), pp. 169–185, 1988.

[2] M.S. Shephard, "Automatic generation of finite element models", in: M. Papadrakakis (ed.), *Solving Large-Scale Problems in Mechanics*, Wiley, New York, pp. 431–460, 1993.

[3] M.A. Yerry and M.S. Shephard, "A modified quadtree approach to finite element mesh generation", *IEEE Comput. Graphics Appl.* **3**(1), pp. 39–46, 1983.

[4] P.L. Baehmann and M.S. Shephard, "Adaptive multiple-level *h*-refinement in automated finite element analyses", *Eng. with Comput.* **5**, pp. 235–247, 1989.

[5] Y.H. Jung and K. Lee, "Tetrahedron-based octree encoding for automatic mesh generation", *Comput. Aided Design* **25**, pp. 141–153, 1993.

[6] S. Cescotto and Z.D. Wu, "A variable-density mesh generation for planar domains", *Comm. Appl. Numer. Methods* **5**, pp. 473–481, 1989.

[7] N.P. Weatherill and O. Hassan, "Efficient three-dimensional Delaunay triangulation with automatic point creation and imposed boundary constraints", *Int. J. Numer. Methods Eng.* **37**, pp. 2005–2039, 1994.

[8] A. Bowyer, "Computing Dirichlet tessellations", *Comput. J.* **24**, pp. 162–166, 1981.

[9] D.F. Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes", *Comput. J.* **24**, pp. 167–172, 1981.
[10] N.P. Weatherill, "Delaunay triangulation in computational fluid dynamics", *Comput. Math. Appl.* **24**, pp. 129–150, 1992.
[11] P.L. George and F. Hermeline, "Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra", *Int. J. Numer. Methods Eng.* **33**, pp. 975–995, 1992.
[12] N.P. Weatherill, "The integrity of geometrical boundaries in the two-dimensional Delaunay triangulation", *Comm. Appl. Numer. Methods* **6**, pp. 101–109, 1990.
[13] P.L. George, F. Hecht, and E. Saltel, "Automatic mesh generator with specified boundary". *Comput. Methods Appl. Mech. Eng.* **92**, pp. 269–288, 1991.
[14] M.K. Loze and R. Saunders, " Two simple algorithms for constructing a two-dimensional constrained Delaunay triangulation", *Appl. Numer. Math.* **11**, pp. 403–418, 1993.
[15] H.C. Huang and A.S. Usmani, "*Finite Element Analysis for Heat Transfer – Theory and Software*, Springer, Berlin, 1994.
[16] W.R. Buell and B.A. Bush, "Mesh generation – a survey", *J. Eng. Industry, ASME, Ser. B* **95**(1), pp. 332–338, 1973.
[17] B.P. Johnston, J.M. Sullivan, Jr. and A. Kwasnik, "Automatic conversion of triangular finite element meshes to quadrilateral elements", *Int. J. Numer. Methods Eng.* **31**, pp. 67–84, 1991.
[18] E. Rank, M. Schweingruber and M. Sommer, "Adaptive mesh generation and transformation of triangular to quadrilateral meshes", *Commn. Appl. Numer. Methods* **9**, pp. 121–129, 1993.
[19] B. Dolšak and A. Jezernik, "Mesh generation expert system for engineering analysis with FEM", *Comput. Ind.* **17**, pp. 309–315, 1991.
[20] E. Kang and K. Haghighi, "A knowledge-based a priori approach to mesh generation in thermal problems", *Int. J. Numer. Methods Eng.* **35**, pp. 915–937, 1992.
[21] D.N. Dyck, D.A. Lowther and S. McFee, "Determining an approximate finite element mesh density using neural network techniques", *IEEE Trans. Magnetics* **28**, pp. 1767–1770, 1992.
[22] D.A. Lowther and D.N. Dyck, "A density driven mesh generator guided by neural network", *IEEE Trans. Magnetics* **29**, pp. 1927–1930, 1993.