Knowledge-Based Free Mesh Generation of Quadrilateral Elements in Two-Dimensional Domains

Article *in* Computer-Aided Civil and Infrastructure Engineering · November 2008

DOI: 10.1111/j.1467-8667.1993.tb00211.x

2 authors:

Yong Zeng
Concordia University

G. D. Cheng
Dalian University of Technology

Microcomputers in Civil Engineering 8 (1993) 259–270

Knowledge-Based Free Mesh Generation of Quadrilateral Elements in Two-Dimensional Domains

Young Zeng

Department of Civil and Architectural Engineering, Institute of Logistics Engineering, Yuzhou Road 79, Chongqing 630041, People's Republic of China

&

Gengdong Cheng

Research Institute of Engineering Mechanics, Dalian University of Technology, Dalian 116023, People's Republic of China

Abstract: *A knowledge-based system, FREEMESH, for free mesh generation of quadrilateral elements in general two-dimensional structures, is introduced in this paper. Based on the recursive logic, a plane domain is firstly represented with boundary features, corresponding to which a group of heuristic rules is built up to generate quadrilaterals recursively. The rules include the initial design rules and the redesign rules. To facilitate the rule-based inference, the geometric reasoning is turned to acquire domain boundary features. Examples show that this strategy is effective.*

INTRODUCTION

The developments of a mesh generation system became an important research topic shortly after the appearance of the finite element method. In recent years, the problem has attracted special attention due to the following reasons:

- The rapid development of computer hardware and the widespread use of commercial finite element analysis software greatly increase the engineers' expectation to solve large-scale structure analysis problems. Excellent mesh generation system plays an essential role for this purpose.
- The finite element analysis software is becoming a constituent part of the integrated CAD system. To facilitate the mechanics analysis of the structure represented in solid modellers, geometry-based mesh generation system is indispensable.[25,26]
- In geometric non-linear problems, analysis of crack propagation and structural shape optimization, the structure domain is subjected to continuous change and the remeshing of structure is required. The fully automatic mesh generation method is decisive.[6,38]
- The developing adaptive structure analysis methods are also based on the automatic mesh generation.[38]

Up to now, most commercial mesh generation systems still use the mapping element method. As the matter of fact, the mapping element method is just a semi-automatic mesh generation one. Even if the generation of mapping elements is automated, it is not yet compatible with most of the engineering environments and it is very difficult to reach the above goals.

In the stream of fully automatic mesh generation, a good deal of commercial systems exist for domain triangulation.[13] On the contrary, the quadrilateral mesh generation in arbitrary domains remains an open question, although theoretical evidence and computing experience[29] have shown that, compared with triangle elements, quadrilateral elements are numerically more stable and have a better modelling capacity in solid mechanics. At present, most of the methods in this respect are heuristic. The widely adopted approach is proposed by Heighway[8] and developed by Lo.[16,17] The

Microcomputers in Civil Engineering 0885-9507/93/S06.00

basic idea behind the method is the fact that based on the domain triangulation two neighbouring triangles can be combined into a quadrilateral. Along the way, Liu[15] investigated the patterns of edge removal in detail, and developed a knowledge-based system XFORMQ to generate quadrilateral mesh. But it is generally difficult to get all quadrilateral elements with required quality in the domain. Recently, Zhu *et al.*[37] relied on the same fact and developed a quadrilateral mesh generation system. On the other hand, there are not many methods which can directly generate quadrilateral elements. The most robust one may be the quadtree approach,[2,14,32] but the disadvantage of this is obvious. The quality of elements near the domain boundary cannot be effectively guaranteed, which is essential for finite element analysis. Hence, the outer to inner approaches have become the hot research topic in recent years.[21,22,31]

To the authors' knowledge, the earliest study in the outer-to-inner category was performed by Sluiter *et al.*[11,23,27] They treat a domain as a loop which is heuristically divided into two new loops each time in the subdivision process. The operation stops when every loop becomes a quadrilateral element. Due to the abilities to include any shaped boundaries and three-dimensional surfaces with a good quality element, it was linked with NASTRAN and ANSYS. AMEKS developed by Blacker *et al.*[4] adopts the rule-based mesh generation method that the arbitrary domain is firstly divided into primitive regions; and then the rules are used to generate the mesh in primitive regions with the mapping element method. As the practical regions are often complex and the domain subdivision cannot always succeed,[22] recently Blacker[3] himself admitted that AMEKS is not satisfactory and he proposed a free mesh generation method named PAVING, with which a proper outer boundary is selected to generate mesh every time; although the mesh quality is good, the information needed for each operation is too much. Besides, many other researchers[5,12,24,30] have also contributed to the problem.

This paper introduces a knowledge-based free mesh generation method for quadrilateral element in any shaped domain, based on the recursive logic.[34,35] This method focuses on the boundary features of domain by which the quadrilateral elements with good quality are generated.

FORMULATION OF PROBLEM

This paper addresses the following problem: for a single-connected plane domain with the even prescribed boundary segment number, such as that shown in Fig. 1, the quadrilateral mesh is generated to satisfy the following requirements:[18-20,29]

- each inner corner of the quadrilateral should range from 45 degree to 135 degree;
- the aspect ratio and taper ratio of each quadrilateral should be between 0·1 and 10;
- in the region away from the boundary, the mesh should be coarse; the transformation from a dense mesh to a coarse mesh should be smooth;
- in the region of mesh refinement, the quality of quadrilateral should be as good as possible; the best one is square.

Indeed, any two-dimensional engineering problem can be transformed into the above one. In finite element modelling, extra consideration should be given to the following two cases:

- the mesh generation of a domain with inner physics attributes, as shown in Fig. 2(a);

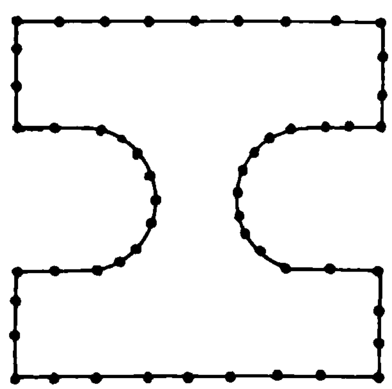


Fig. 1. Problem formulation.

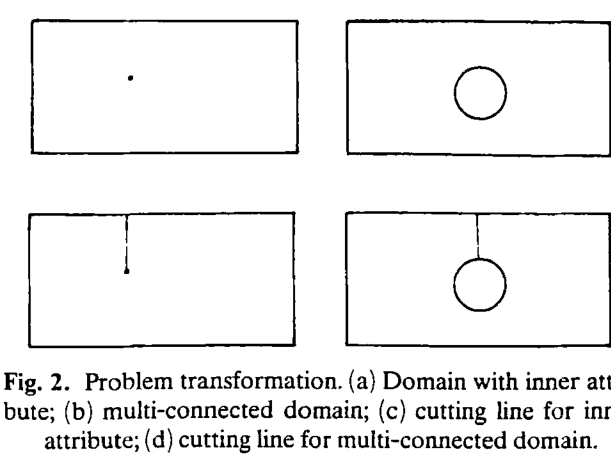


Fig. 2. Problem transformation. (a) Domain with inner attribute; (b) multi-connected domain; (c) cutting line for inner attribute; (d) cutting line for multi-connected domain.

— the mesh generation of a multi-connected domains, as shown in Fig. 2(b).

The above two cases can be solved by adding some cutting lines in the domain, such as in Fig. 2(c) and Fig. 2(d), respectively. By doing this, the problem becomes the mesh generation of several single-connected domains, and also the mesh control parameters such as mesh density can be naturally defined in the boundary edges of domain.

STRATEGY FOR MESH GENERATION

The mesh generation is actually a design problem subjected to modelling constraints behind which there must exist a design mechanism, no matter what kinds of method are adopted.[7] It is suggested that the logic of design be the recursive logic[34] and accordingly a recursion model of designing is proposed.[35] The model is especially suitable to the layout problem such as space planning, mesh generation and so on. It firstly resolves the design problem recursively into an atomic design problem and a subdesign problem. The solution of the atomic design problem can be directly stemmed from the existing deductive design rules and design knowledge, which brings about a set of new constraints imposed on the subdesign problem. Then the same procedure recursively works on the new subdesign problem until the subdesign becomes an atomic design.

In the case of mesh generation, the first step is to determine the atomic designs. One can take, for examples, one element or a group of elements in a specified region, as the atomic designs. Thereby the domain can be defined recursively Ω_n is used to denote a plane domain while Q_i, called atom, is turned to represent the quadrilateral element or quadrilateral elements in a specified region with prescribed features; then the domain Ω_n can be expressed as

$$\Omega_n = \langle Q_i, \Omega_{n-i} \rangle \tag{1}$$

Therefore, the quadrilateral mesh generation of domain Ω_n can be resolved into generating valid atom Q_i and generating the mesh for the new domain Ω_{n-i} until Ω_{n-i} becomes a valid atom (see Fig. 3). In the present case, the atom is assumed to be a quadrilateral, and the element is generated one by one. This consideration has the following advantages over the existing approaches:[36] (1) the information required for element extraction each time is very simple; (2) the mesh flexibly and naturally conforms to the boundary features of the domain; (3) the heuristic knowledge generating quadrilateral is easy to evolve which makes the system behave better and better; (4) the strategy itself is complete and perfect.

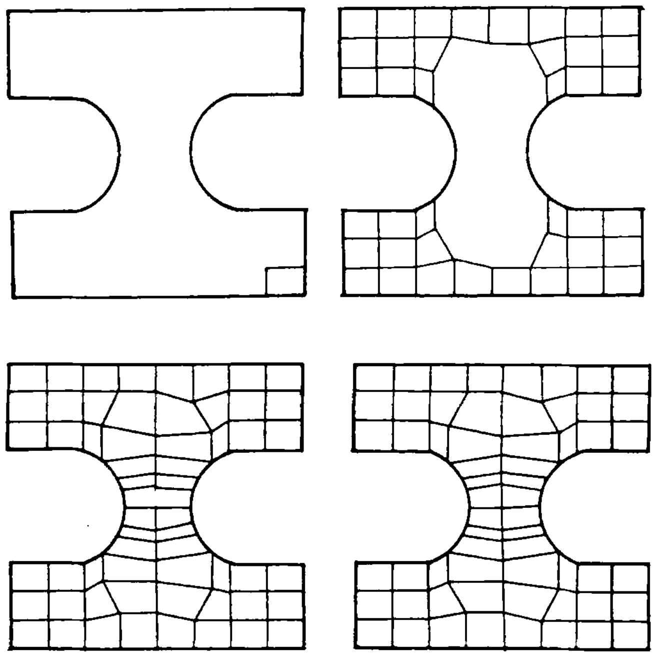


Fig. 3. Recursive generation of quadrilaterals.

INTRODUCTION OF SYSTEM FREEMESH

According to the above strategy, we developed a system, FREEMESH, to generate all quadrilateral elements in the plane domain. The architecture of the system is shown in Fig. 4. Three subsystems exist in FREEMESH — the feature recognition subsystem, the initial design subsystem and the redesign subsystem. The recursion strategy is obliged to control the whole design process. From the feature recognition subsystem, the semantic information of the geometric data of the structure is acquired, which facilitates the formal inference in the initial design subsystem. The redesign subsystem is used to correct the bad effects from the initial design which keep the execution going.

Feature recognition subsystem

Since the mesh generation starts from the domain boundary, the way of generating the element must depend upon the boundary features; and at the same time, the representation of engineering structure in a CAD environment does not support knowledge-based inference. To facilitate these, the geometry model in the system should assume the boundary representation (bRep) and the technique of geometric reasoning is required.[1,28,33] The structure of the feature recognition subsystem is shown in Fig. 5.

In tackling geometric information, the geometric reasoning has the following advantages:

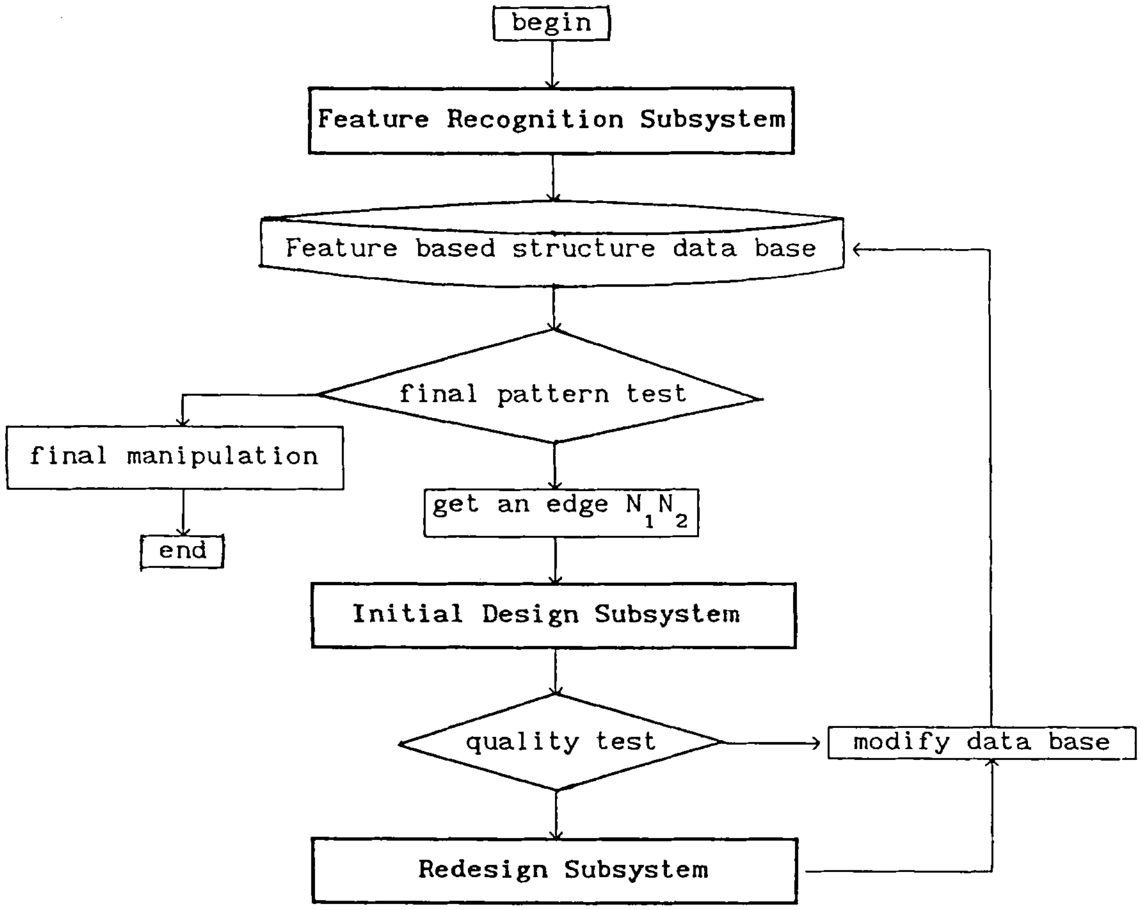


Fig. 4. Architecture of FREEMESH.

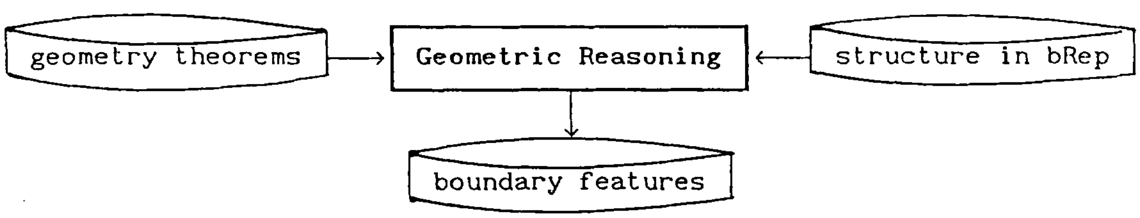


Fig. 5. Feature recognition subsystem.

- Geometric reasoning is declarative. The attribute of an object is either obtained from the axiom of the formal system of geometric reasoning or the theorem deducible from the formal system. The explicit as well as the implicit geometric information can be easily retrieved;
- Geometric reasoning supports the multiviews of a geometry model.

For example, to find the relation of two straight lines passing through points point(X_1, Y_1), point(X_2, Y_2) and point(X_3, Y_3), point(X_4, Y_4), respectively, the following theorem in predictive logic can be used:

parallel (point(X_1, Y_1), point(X_2, Y_2), point(X_3, Y_3), point(X_4, Y_4)):-

find_line_equation (point(X_1, Y_1), point(X_2, Y_2), A1, B1, C1),

find_line_equation (point(X_3, Y_3), point(X_4, Y_4), A2, B2, C2),

A1*B2 = A2*B1.

The boundary features of the domain, such as concave point, point line and point domain relations, can be acquired in this way.

Initial design subsystem

The aim of the initial design subsystem is to construct the topology of the quadrilateral element by a set of design rules according to boundary features. The design rules assume the following form:

$$\text{if } b_j \text{ then } a_k^j \qquad (2)$$

where b_j and a_k^j represent a boundary feature and the counterpart design action, respectively. Design action determines element topology and measure. Figure 6 is the flow chart of the initial design subsystem. Obviously, the definition of design rules is decisive to the subsystem.

Basic design actions
To define the design rule, the basic design actions should first be considered. There are only three possibilities or patterns to generate a quadrilateral over an edge N_1N_2 of a domain:

(a_1) Adding three lines: starting from the two end points N_1, N_2 of the edge N_1N_2, draw two supplementary lines N_1N_4 and N_2N_3, then link N_3N_4. The required parameters are φ_1, φ_2, l_2 and l_4, respectively. See Fig. 7(a).
(a_2) Adding two lines: two lines can be generated from the points N_2 and N_4 inclined to line N_1N_2 and N_4N_1 with the angle φ_2 and φ_4, respectively. See Fig. 7(b).
(a_3) Adding one line: link two points N_3 and N_4. See Fig. 7(c).

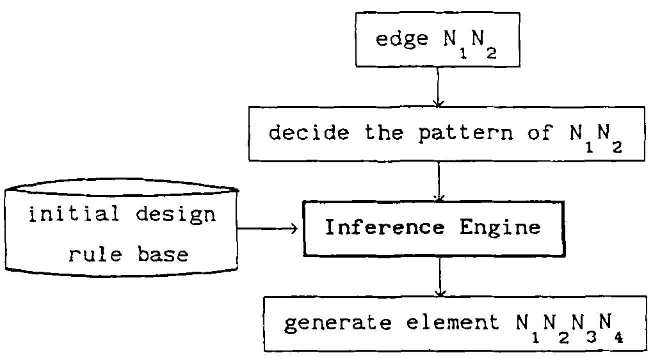


Fig. 6. Structure of initial design subsystem.

Definition of boundary features and design rules
It can be seen from Exp. (2) that it is only necessary to define boundary features in terms of the forward three basic design actions; such definition should facilitate the generation of good element with ease.

Obviously, the following basic three features cannot be absent in the system:

(b_1): $\theta_1 \geq U_c$, $\theta_2 \geq U_c$, as in Fig. 8(a);
(b_2): $\theta_1 \leq U_c$, $\theta_2 \geq 180°$, $\theta_4 \geq 180°$, as in Fig. 8(b);
(b_3): $\theta_1 \leq U_c$, $\theta_2 \leq U_c$, as in Fig. 8(c);

where U_c represents the upper bounds of the element inner corner. Accordingly, the basic design actions are:

(a_1^1): $\varphi_1 = \theta_1/2$, $\varphi_2 = \theta_2/2$, $l_2 = (l_1 + l_6)/(2^*\sin\varphi_2)$, $l_4 = (l_5 + l_1)/(2^*\sin\varphi_1)$;
(a_2^2): $\varphi_2 = \pi - \theta_1$, $\varphi_4 = \pi - \theta_1$;
(a_3^3): link N_3 and N_4.

As the generation of the element $N_1N_2N_3N_4$ changes the boundary description of the remained domain, the design actions should forecast their own effects to such an extent that the succeeding action will not be too difficult. This can be reached through defining the relatively complex boundary features, which relies on the influence region of the element $N_1N_2N_3N_4$ and can be perfected by graph tests. For example, the following extended features had better be placed into the system:

(b_4): $\theta_1 \leq U_c$, $U_C \leq \theta_2 \leq 240°$, $\theta_5 \leq U_c$, $U_C \leq \theta_4 \leq 240°$, $\theta_7 \leq U_c$, see Fig. 9(a);
(b_5): $\theta_1 \leq U_c$, $U_C \leq \theta_2 \leq 240°$, $\theta_5 \leq U_c$; see Fig. 9(b).

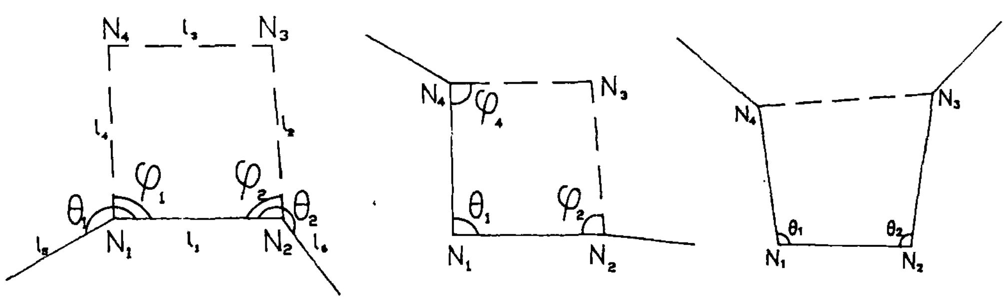


Fig. 7. Basic design actions. (a) Adding three lines. (b) Adding two lines. (c) Adding one line.

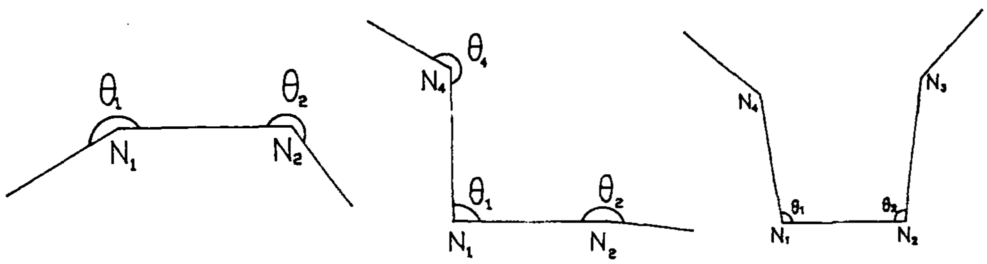


Fig. 8. Basic boundary features.

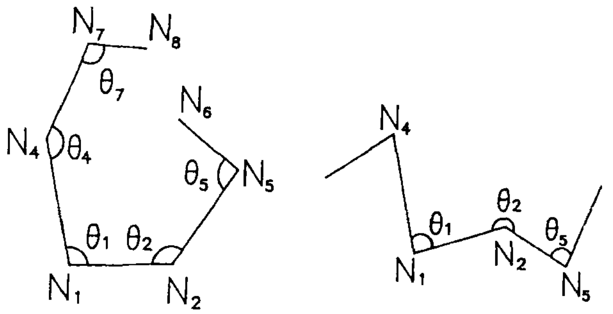


Fig. 9. Extended boundary features.

The counterpart design actions are:

(a_4^1): $X_3=(X_2+X_4+X_6)/3$, $Y_3=(Y_2+Y_4+Y_6)/3$, (X_3, Y_3) is the coordinate of the point N_3;
(a_2^5): $\varphi_4=\pi-\theta_1$, $\varphi_2=\theta_5/(\theta_1+\theta_5)^*\theta_2$.

It should be noted that the definition of extended boundary features and the determination of parameters in design actions are subject to evolution in the execution of the system. The present strategy is acceptable for this purpose.[10]

Redesign subsystem

What has been done in the stage of initial design is just the determination of element topology; in which the element was generated to be as good as possible, satisfying the element quality requirements. As is required by the recursion strategy, however, the generated element should be less good so that the remaining domain is easily tackled. This means that the generated element must satisfy some other requirements to guarantee the quality of the remaining domain. They are:

- every outer corner of the generated element should be greater than the lower bounds of the inner corner;
- every edge of the generated element should not be too near to the concave point of the remaining domain;
- every corner point of the generated element should not approach the boundary of the remaining domain.

Here, the distance between point and segment refers to both the vertical and the horizontal one. Only when both distances are small, does the point approach the segment (see Fig. 10).

To realize the idea, the generated element must be evaluated and modified when necessary. This is the task of the redesign subsystem which is shown in Fig. 11.

Denote I_k^i as the ith invoked constraint by element $N_1N_2N_3N_4$ in the kth design action, then the corresponding redesign rule will assume the following form:

$$R_d\text{: if } I_k^i \text{ then } r_k^i \tag{3}$$

In cases of design actions a_1 and a_2, as the outer corner of element can be controlled in the stage of initial design by decreasing the corresponding inner corner, the constraints subjected to practical consideration are distance between element nodes and domain boundary, element edges and domain concave corner points. For different design actions, there exist different redesign rules.

Redesign rules when adding three lines

(R_d^1): if the edge N_2N_3 is too near to a boundary concave point, then decrease the inner corner φ_2, see Fig. 12(a);
(R_d^2): if the edge N_4N_1 is too near to a boundary concave point, then decrease the inner corner φ_1, see Fig. 12(b);
(R_d^3): if the edge N_3N_4 is too near to a boundary concave point, then decrease the inner corner φ_2 or φ_1, or decrease the length of edge N_2N_3 or N_4N_1, see Fig. 12(c);
(R_d^4): if the point N_3 is too near to a boundary segment, then decrease the inner corner φ_2, see Fig. 12(d);

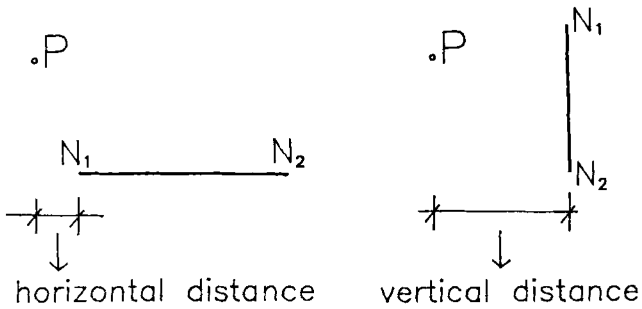


Fig. 10. Point–segment relation.

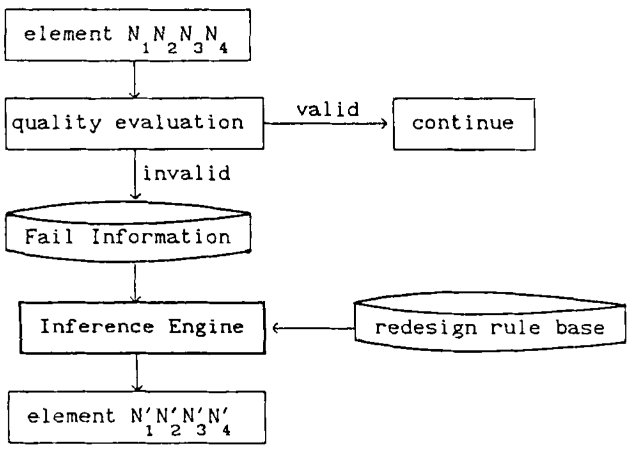


Fig. 11. Structure of redesign subsystem.

(R_d^5): if the point N_4 is too near to a boundary segment, then decrease the inner corner φ_1, see Fig. 12(e).

Redesign rules when adding two lines

(R_d^6): if the point N_3 is too near to a boundary segment, then φ_2 or φ_4 is suggested to be decreased, see Fig. 13(a);

(R_d^7): if the edge N_2N_3 is too near to a boundary concave point, then φ_2 may be decreased, see Fig. 13(b);

(R_d^8): if the edge N_3N_4 is too near to a boundary concave point, then φ_4 may be decreased, see Fig. 13(c).

Redesign rules when adding one line

(R_d^9): if φ_3 is too small or φ_4 is too large, then the redesign may be performed by taking design action a_2 on the edge N_1N_2, as shown in Fig. 14(a);

(R_d^{10}): if φ_4 is too small or φ_3 is too large, then the redesign may be performed by taking design action a_2 on the edge N_2N_3, as shown in Fig. 14(b);

(R_d^{11}): if θ'_3 is too small, then the redesign may be performed by taking design action a_2 on the edge N_1N_2, as shown in Fig. 14(c);

(R_d^{12}): if θ'_4 is too small, then the redesign may be performed by taking design action a_2 on the edge N_2N_3, as shown in Fig. 14(d);

(R_d^{13}): otherwise, the edge N_1N_2 should be redesigned by design action a_1, as shown in Fig. 14(e).

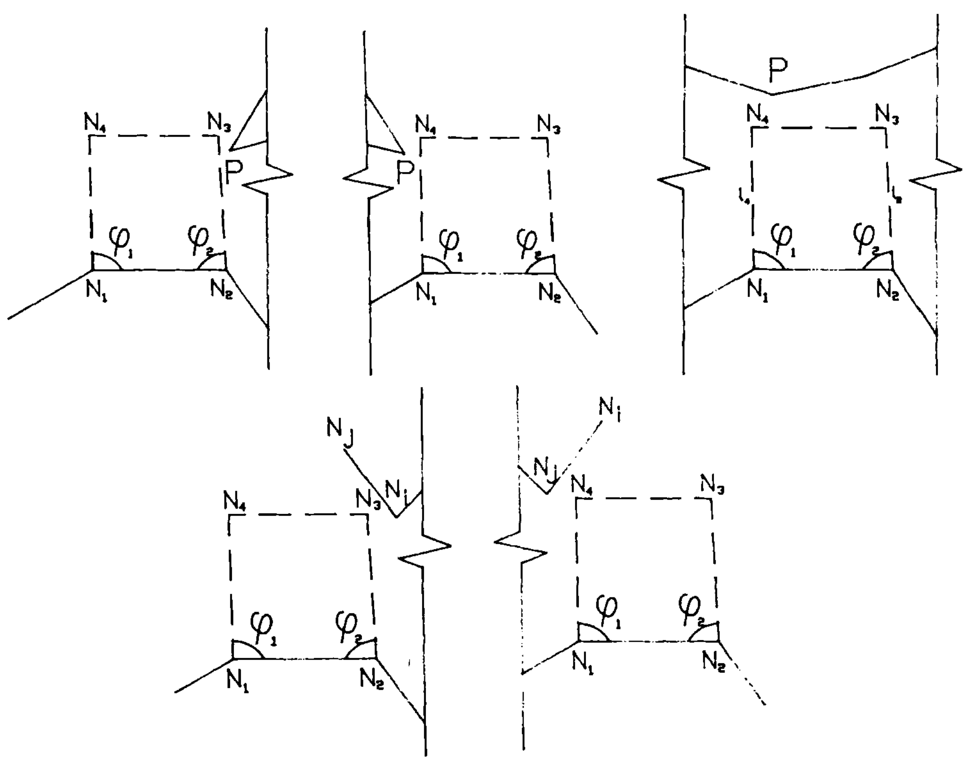


Fig. 12. Redesign when adding three lines.

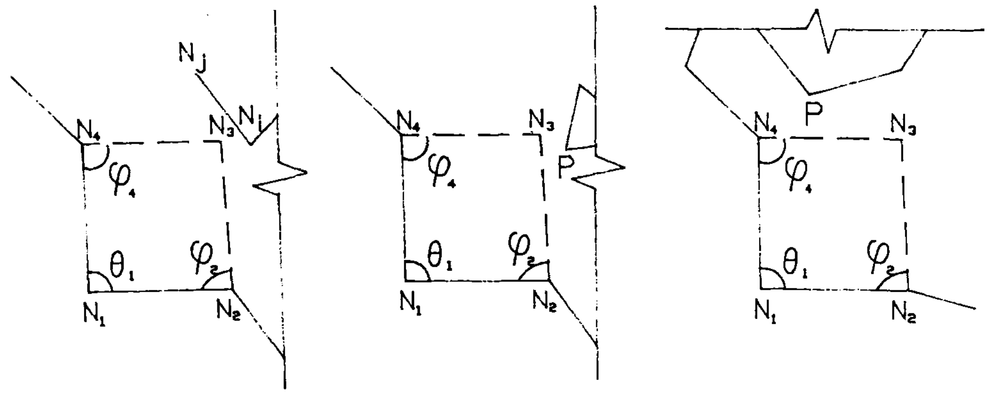


Fig. 13. Redesign when adding two lines.

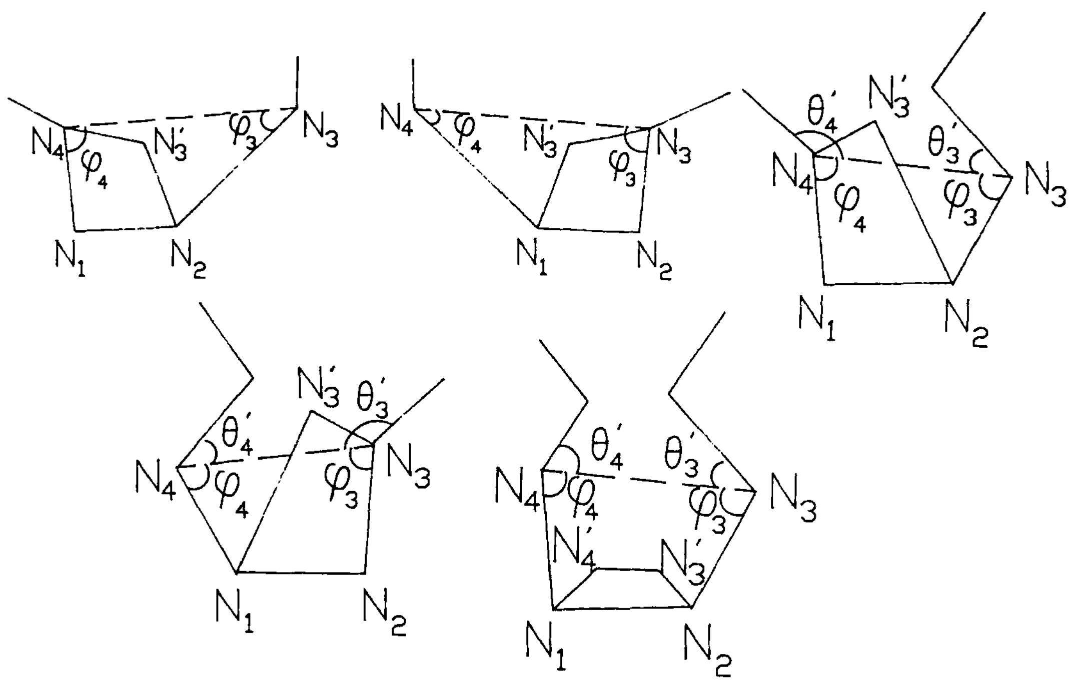


Fig. 14. Redesign when adding one line.

Final patterns and mesh smoothing

When there are only six segments in domain boundary, some special considerations should be given. The Laplace method[9] plays an essential part in this point. As in Fig. 15, the coordinate of the ith point is:

$$\begin{cases} X_i = \dfrac{1}{2N_i} \sum\limits_{n=1}^{N_i} (X_{nj} + X_{ni}) \\ Y_i = \dfrac{1}{2N_i} \sum\limits_{n=1}^{N_i} (Y_{nj} + Y_{ni}) \end{cases} \quad (4)$$

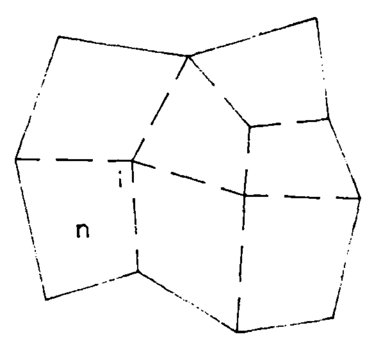


Fig. 15. Laplace method.

where N_i represents the total element number connected with the ith node, n_i and n_j denote the node number of the nth element, connected with the ith node.

Hence, the final patterns and the corresponding design rules are as follows:

Pattern 1: as in Fig. 16(a), the coordinate of ith node is

$$\begin{cases} X_i = \frac{1}{3}(X_2 + X_4 + X_6) \\ Y_i = \frac{1}{3}(Y_2 + Y_4 + Y_6) \end{cases} \tag{5}$$

Pattern 2: as in Fig. 16(b), the coordinate of ith node is

$$\begin{cases} X_i = \frac{1}{8}(3*X_3 + 3*X_5 + X_2 + X_6) \\ Y_i = \frac{1}{8}(3*Y_3 + 3*Y_5 + Y_2 + Y_6) \end{cases} \tag{6}$$

Pattern 3: as in Fig. 16(c), link points N_3 and N_6.

Although the presented strategy can generate a good mesh in general, the smoothing technique is still necessary after the mesh topology is determined. And also the Laplace method is adopted.

Recursion strategy

As there are so many features available for consideration for each time of initial design, and the strategy is proven to be sensitive to the selection order of the features,[34] a suitable recursion strategy is very important to generate a good mesh. Through a deal of graph tests, the following sequential order is accepted for the recursion resolution strategy in the system FREE-MESH:

$$b_4, b_5, b_3, b_2, b_1$$

which is an important difference of the present strategy from other free mesh generation methods.[35]

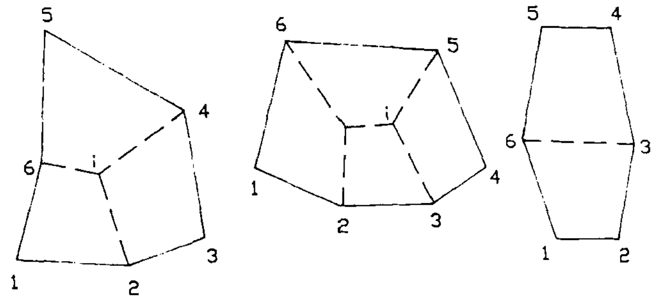


Fig. 16. Final patterns. (a) Pattern 1. (b) Pattern 2. (c) Pattern 3.

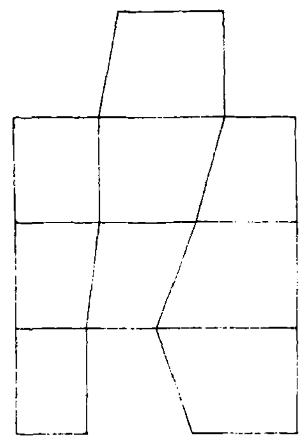


Fig. 17. Automatic generation of mapping element.

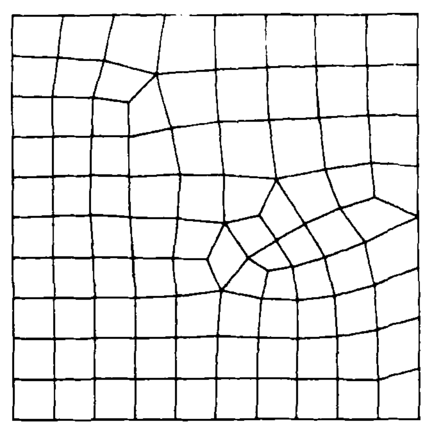


Fig. 18. Mesh generation in mapping element.

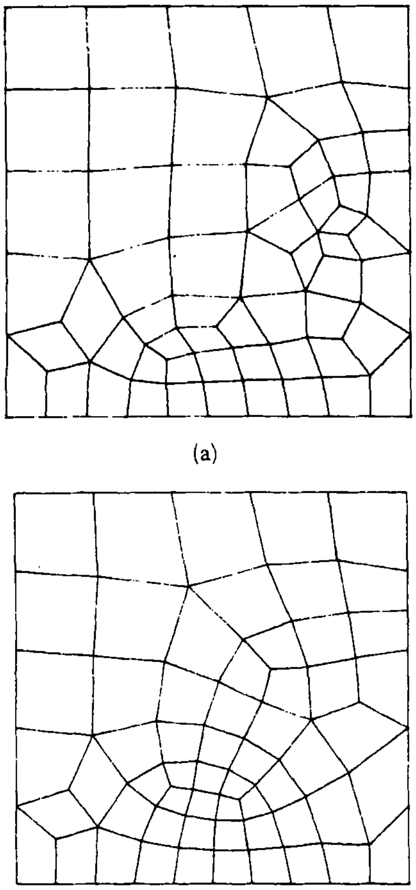
CASE STUDIES

Besides directly generating finite element mesh, the strategy can serve other purposes. For example, in the mapping element method, there are two open questions, i.e. the automatic generation of mapping element and the mesh generation in mapping element when two opposite sides do not share a subdivision number. Figures 17 and 18 are examples in two aspects.

Figure 19(a) and (b) are the mesh before and after smoothing. Figures 20–24 are some more examples.

CONCLUDING REMARKS

A new mesh generation scheme for all quadrilateral elements is proposed in this paper, based on the know-



(a)

(b)

Fig. 19. Mesh before and after smoothing.

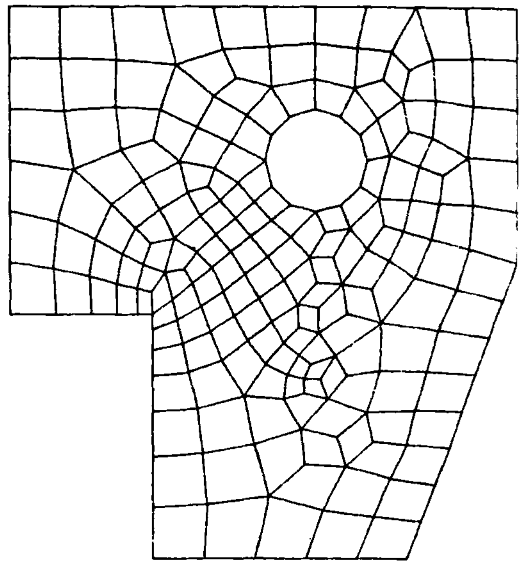


Fig. 22. Multi-connected domain.

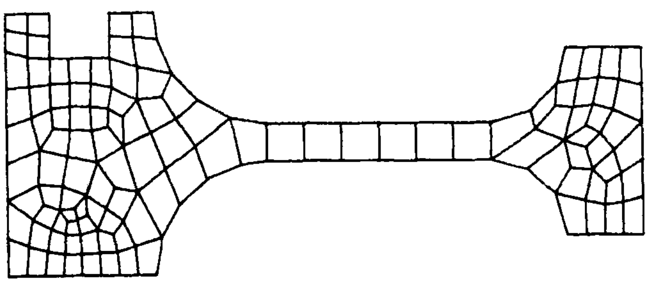


Fig. 20. Turbine disc.

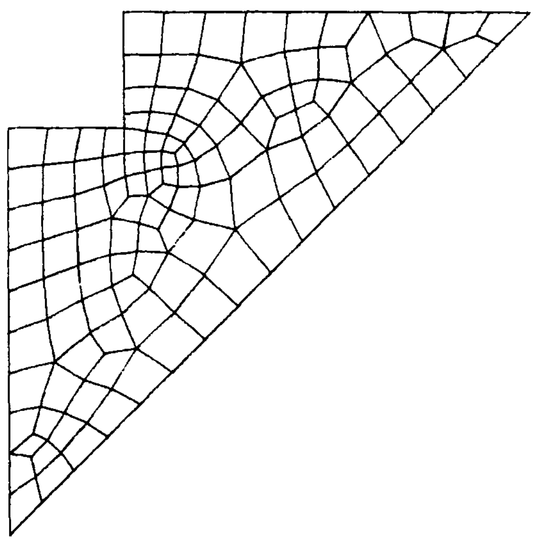


Fig. 23. Plugger.

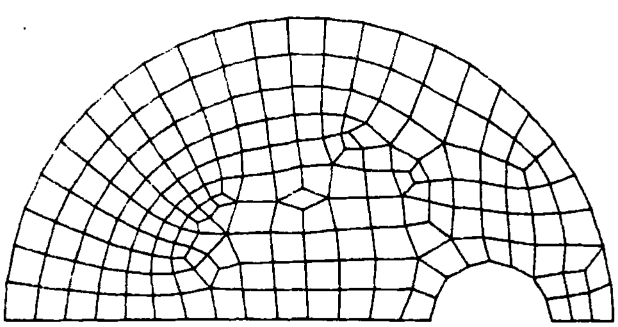


Fig. 21. Revolving disc.

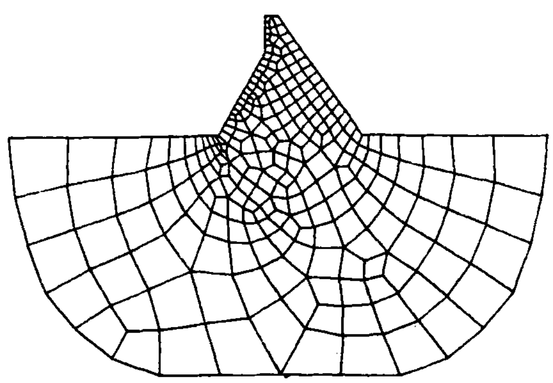


Fig. 24. Meshing of a dam.

ledge-based design theory. The scheme is flexible and robust. The key points of the presented strategy are the design and redesign rules, the latter of which are relatively fixed and easily acquired. The design rules can be improved by graph tests. The problem lies in the fact that the evolution of the system behaviour depends upon better design rules; this relies on the augmentation of more complex features which causes the inference efficiency to become lower. To overcome this, we are developing a neural network based design rule representation system.[10] Another problem is how to extend the present strategy to 3D structures.

ACKNOWLEDGEMENT

This project is supported by the National Natural Science Foundation of China. The authors are indebted to Dr D. X. Zhang for his discussion.

REFERENCES

1. Arab, F. & Wing, J. M., Geometric reasoning: new paradigm for processing geometric information. In *Design Theory for CAD*, ed. H. Yoshikawa & E. A. Waterman. Elsevier Science Publishers, 1987.
2. Baehmann, P. L. *et al.*, Robust, geometrical based, automatic two dimensional mesh generation. *International Journal for Numerical methods in Engineering*, **24** (1987) 1043–78.
3. Blacker, T. D. & Stephenson, M. B., PAVING: a new approach to automated quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, **32** (1991) 811–47.
4. Blacker, T. D. *et al.*, Knowledge system approach to automated two dimensional quadrilateral mesh generation. In *Proc. ASME Int. Comp. Eng. Conf.*, San Francisco, 1988, Vol. 3, p. 153.
5. Bova, S. W. & Carey, G. F., Mesh generation/refinement using fractal concepts and iterated function systems. *International Journal for Numerical Methods in Engineering*, **33** (1992) 287–305.
6. Braibant, V. & Morelle, P., Shape optimal design and free mesh generation. *Structural Optimization*, **2** (1990) 223–31.
7. Cheng, G. D. & Zeng, Y., Automated strategies for finite element modeling. *Computer & Structures*, **44** (4) (1992) 905–9.
8. Heighway, E. A., A mesh generator for automatically subdividing irregular polygons into quadrilaterals. *IEEE Trans. on Magnetics*, **MAG-19** (1983) 2535–8.
9. Herrmann, L. R., Laplacin-isoparametric grid generation scheme. *Journal of Engineering Mechanics Division, Proceedings of ASCE*, **102** (1976) 749–56.
10. Huang, J. W. & Zeng, Y., A neural network approach to mesh generation of quadrilateral elements, (in prep.).
11. Jain, A., Modern methods for automatic FE generation. In *Modern Methods for Automating Finite Element Mesh Generation*. ASME, New York, 1986, pp. 19–28.
12. Johnston, B. P. & Sullivan, J. M., Fully atomatic two dimensional mesh generation using normal offsetting. *International Journal for Numerical Methods in Engineering*, **33** (1992) 425–42.
13. Ho-Le, K., Finite element mesh generation methods: a review and classification. *Computer Aided Design*, **20** (1988) 27–38.
14. Kela, A. *et al.*, Toward automatic finite element analysis. *Computers in Mechanical Engineering*, **8** (1986) 57–71.
15. Liu, Y. C. *et al.*, An expert system for forming quadrilateral finite elements. *Engineering Computation*, **7** (1990) 249–57.
16. Lo, S. H., A new mesh generation scheme for arbitrary planar domains. *International Journal for Numerical Methods in Engineering*, **21** (1985) 1403–26.
17. Lo, S. H., Generating quadrilateral elements on plane and over curved surfaces. *Computers & Structures*, **3** (1989) 421–6.
18. Robinson, J., Some new distortion measures for quadrilaterals. *Finite Elements in Analysis and Design*, **3** (1987) 183–97.
19. Robinson, J., CRE method of element testing and Jacobian shape parameters. *Engineering Computation*, **4** (1987) 113–18.
20. Robinson, J., Validity of aspect ratio sensitivity testing — an analytical investigation. *Finite Elements in Analysis and Design*, **9** (1991) 125–32.
21. Rudd, B. W., Impacting the design process using solid modelling and automated finite element mesh generation. *Computer Aided Design*, **20** (1988) 212–16.
22. Sapidis, N. & Perucchio, R., Advanced techniques for automatic finite element meshing from solid models. *Computer Aided Design*, **21** (1989) 248–53.
23. Schoofs, A. J. G. *et al.*, A general purpose two-dimensional mesh generator. *Advanced Engineering Software*, **1** (1979) 131–6.
24. Sezier, L. & Zeid, I., Automatic quadrilateral/triangular free form mesh generation for planar regions. *International Journal for Numerical Methods in Engineering*, **32** (1991) 1441–83.
25. Shephard, M. S. & Finnigan, P. M., Integration of geometric modeling and advanced finite element preprocessing. *Finite Elements in Analysis and Design*, **4** (1988) 147–62.
26. Shephard, M. S. & Yerry, M. A., Toward automated finite element modeling for the unification of engineering design and analysis. *Finite Elements in Analysis and Design*, **2** (1986) 143–60.
27. Sluiter, M. L. C. & Hansen, D. C., A general purpose automatic mesh generator for shell and solid finite elements. In *Computers in Engineering*, Vol. 3, ed. L. E. Hulbert. ASME, New York, 1982, pp. 29–34.
28. Smithers, T., AI-based design versus geometry-based design or why design cannot be supported by geometry alone. *Computer Aided Design*, **21** (1989) 141–50.

29. Steele, J. M., *Applied Finite Element Modeling.* Marcel Dekker, New York, 1989.
30. Talbert, J. A. & Parkinson, A. R., Development of an automatic two dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definition. *International Journal for Numerical Methods in Engineering*, **29** (1990) 1551–67.
31. Wördenweber, B., Finite element mesh generation. *Computer Aided Design*, **16** (1984) 285–91.
32. Yerry, M. A. & Shephard, M. S., A modified quadtree approach to finite element mesh generation. *IEEE Computer Graphics & Application*, **4** (1983) 39–46.
33. Zeng, Y., Knowledge based design of finite element model: theories and implementation. PhD dissertation, Dalian University of Technology, 1992.
34. Zeng, Y. & Cheng, G. D., On the logic of design. *Design Studies*, **12** (1991) 137–41.
35. Zeng, Y. & Cheng, G. D., A recursion model of design reasoning, *Artificial Intelligence in Engineering* (submitted).
36. Zeng, Y. & Yu, J., Automatic mesh generation: strategies and algorithms, *Computer Aided Design* (submitted).
37. Zhu, J. Z. *et al.*, A new approach to the development of automatic quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering*, **32** (1991) 849–66.
38. Zienkiewicz, O. C., Computational mechanics today. *International Journal for Numerical Methods in Engineering*, **34** (1992) 9–33.