

ON AUTOMATIC MESH CONSTRUCTION AND MESH REFINEMENT IN FINITE ELEMENT ANALYSIS

SOO-WON CHAE† and KLAUS-JÜRGEN BATHE‡

†Korea Institute of Machinery and Metals, 66 Sangnam-Dong, Changwon, Kyungnam, Korea
‡Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

Abstract—A valuable approach for the automatic generation of effective finite element meshes is presented. The approach comprises, firstly, an initial mesh construction and, secondly, an h-version of adaptive refinement based on an error analysis. For the initial mesh construction, a robust triangulation scheme for 2D analysis and tetrahedrization scheme for 3D analysis are used, in which the elements are generated from the outside boundaries. For the adaptive refinement process, an error indicator is used with a relaxation factor to obtain efficient solutions. The initial mesh construction schemes have been implemented for 2D and 3D analyses whereas the self-adaptive mesh improvement procedure has only been implemented for 2D analysis. Example solutions are given to demonstrate the solution procedures.

1. INTRODUCTION

Due to the increased use of finite element analysis in CAE environments, much research effort has been focussed during the last decade on automatizing finite element analysis procedures. For linear analysis, the major problem is the automatic construction of effective finite element meshes.

Many different approaches to obtain effective finite element meshes automatically have been studied. See [1-5] for a partial review of work in this field. A desirable approach is to combine an efficient initial mesh construction and a self-adaptive refinement method based on an error analysis. If an efficient initial mesh can be constructed, the subsequent refinement will only require a few iterations.

Figure 1 shows schematically the overall procedures we use. Since triangular and tetrahedral elements are efficient to fit into any arbitrary analysis domain and quadratic elements are usually most effective in the analysis, 6-node triangular and 10-node tetrahedral elements are used in our mesh constructions.

To reach an efficient initial mesh, the automatic meshing scheme should be robust and satisfy the following conditions as much as possible.

- (i) The user should be able to control the local mesh density in any part of the analysis domain.
- (ii) The elements should be close to equilateral triangles (in 2D analysis) and equilateral (i.e. equal-faced) tetrahedra (in 3D analysis).
- (iii) The algorithm should be economical with respect to both human effort and computer time.

An extensive review of triangulation techniques has been given by Thacker [6]. The scheme developed in this paper is similar to the one by Sadek [7], in which triangular elements are generated from the outside

boundary and the local mesh density is controlled by key nodes on the boundary. This triangulation scheme is also used as the basis for the tetrahedrization in 3D analysis.

As for the adaptive refinement process, an error indicator is proposed, which is closely related to the theoretical error indicators suggested by Babuska *et al.* [3] but uses a relaxation factor. With this relaxation factor better solutions can be obtained at stress concentrations with little sacrifice on the overall accuracy. The h-version of grid enrichment scheme is employed for the refinement process, but it has only been implemented for 2D solutions. In this process an element is subdivided into four subelements.

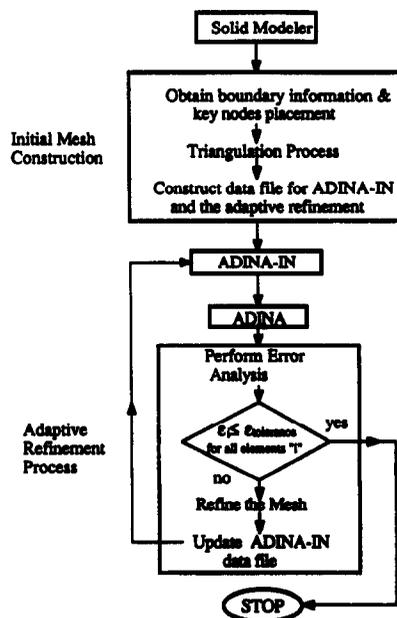


Fig. 1. The mesh generation and mesh refinement processes in 2D analysis.

In the following sections we first present the schemes used for the initial mesh constructions in 2D and 3D analyses. We then present the error indicator used and the refinement procedure for 2D solutions, and finally we demonstrate the techniques by some example solutions.

2. SURFACE TRIANGULATION

The proposed scheme for automatic triangulation is a modified version of Sadek's algorithm [7]. The basic strategy of the method is as follows. To begin, key nodes are placed by the analyst around the boundary considering the desired local mesh density. The key nodes are ordered in the counter-clockwise direction on the boundary so that the unmeshed region lies to the left as we travel along the boundary. Given a certain domain with a number of key nodes, triangular elements are generated from the boundary towards the inside of the domain by cutting the corner key nodes on the boundary. Since only the corner nodes are removed, the key nodes with concave boundary angles cannot be removed unless they become corner key nodes with convex boundary angles. Hence, if an analysis domain has a complex shape, it is recommended that the entire domain be subdivided into near convex subdomains. Each domain or subdomain to be triangulated is considered a loop and the boundary of a loop is called a loop-boundary.

Since the best form of triangular elements in finite element analysis is known to be the equilateral triangle, the resulting mesh is generated with the objective to arrive at equilateral triangular elements.

Our triangulation scheme consists of generating elements by trimming or digging from a key node on loop-boundaries in order to reduce the number of key nodes. Hence, at least two basic operations are needed to construct meshes with well-conditioned elements. One operation is to generate one element by trimming a well-conditioned key node and the other operation is to generate two elements by digging at a key node into the domain. This digging process should promote producing well-conditioned neighboring key nodes.

Consequently, two basic operations (type-1 and type-2 operations) have been designed. In addition, one more operation (type-0) is used to complete the triangulation process. This operation constructs the last two elements when the number of key nodes is four.

The type-1 operation, as shown in Fig. 2, is designed to generate one element by trimming one type-1 key node. In Sadek's algorithm, the user is supposed to input the criterion for the type-1 node decision. However, in our experience, this criterion is one of the key issues in the triangulation process. Many of the existing triangulation schemes, e.g. [8, 9], use the boundary angle at a node, as shown in Fig. 2, as a criterion for a type-1 node decision such

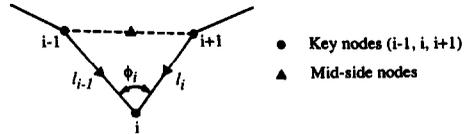


Fig. 2. Example of type-1 operation.

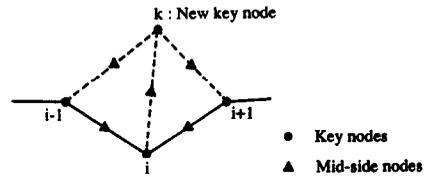


Fig. 3. Example of type-2 operation.

as $\phi_i \leq 85^\circ$ or $\phi_i \leq 90^\circ$. However, our experience is that in addition to the boundary angle, the size of adjacent edges and the effects of the type-1 operation on the loop-boundary should all be considered as criteria for a type-1 node decision. For example, consider the key node i in Fig. 2. In our scheme, an edge-length ratio, δ_i , at a key node i is defined as follows:

$$\delta_i = \text{Max}(l_i/l_{i-1}, l_{i+1}/l_i). \tag{1}$$

Then the conditions for being a type-1 key node are that the boundary angle, ϕ_i , is less than 80° ($\delta_i \geq 1.8$) or 95° ($\delta_i < 1.8$) depending on the edge-length ratio, and the neighboring edge-length ratios around node i should not be changed drastically after a type-1 operation.

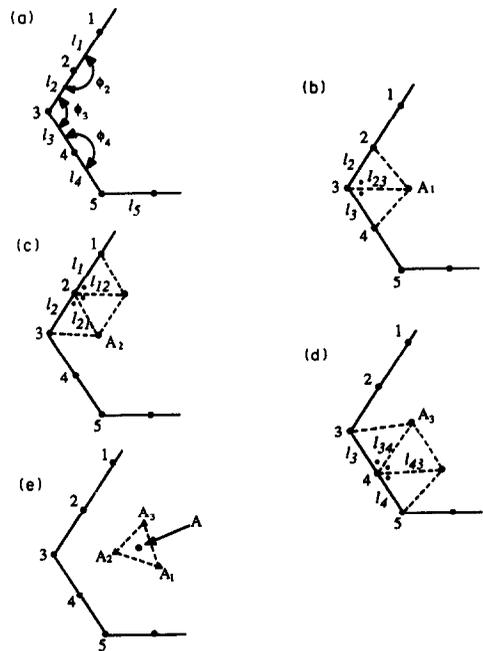


Fig. 4. Generation of a new key node.

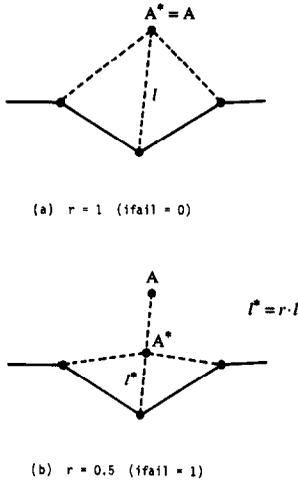


Fig. 5. Adjustment of a new key node.

The type-2 operation is designed to generate two triangular elements at a type-2 key node by introducing a new key node k as shown in Fig. 3. A type-2 key node is defined as a node for which the boundary angle is less than or equal to 150° ($\phi_i \leq 150^\circ$) and which is not a type-1 key node. We use an adaptive type-2 operation, in which the position of a new generated key node is adjusted to avoid an overlapped region or bottle-neck-like region in a loop-boundary. The adaptive type-2 operation is composed of two steps. First, a new key node is generated considering the effects of neighboring nodes and then the new key node position is adjusted in order to reduce the sizes of newly generated elements, and specifically to avoid that the new key node is too close to or falls outside the loop-boundary.

Consider the first step of a type-2 operation at the key node 3 in Fig. 4(a). In order to construct two well-conditioned elements at node 3, one candidate position A_1 is determined by the length l_{23} ($l_{23} = \sqrt{l_2 l_3}$) taken along a line bisecting the angle ϕ_i , see Fig. 4(b). The effects of neighboring nodes such as node 2 and node 4 are also included by considering the additional candidate positions A_2 and A_3 , respectively. For brevity, we only explain how A_2 and A_3 are determined when ϕ_2 and ϕ_3 are close to 180° . In this case the candidate position A_2 is determined by the length l_{21} ($l_{21} = \sqrt[3]{l_1 l_2^2}$) taken along a line trisecting the angle ϕ_2 . The candidate position A_3 is similarly generated at node 4. From these candidate positions, a final position, A , is obtained by using weighting factors,

$$A = w_1 A_1 + w_2 A_2 + w_3 A_3 \quad (2)$$

where

$$w_1 = \frac{1}{l_{23}^2 w_0}, \quad w_2 = \frac{1}{l_{21}^2 w_0}, \quad w_3 = \frac{1}{l_{34}^2 w_0},$$

and

$$w_0 = \frac{1}{l_{23}^2} + \frac{1}{l_{21}^2} + \frac{1}{l_{34}^2}.$$

In the second step the new key node position, A , is adjusted as shown in Fig. 5 to A^* by using a scaling factor

$$r = \frac{1}{1 + \text{ifail}}, \quad \text{ifail} = 0, 1, 2, \dots, \quad (3)$$

where ifail is a number that depends upon whether the basic operations could not be performed, see Fig. 5 and the explanation below. Initially, ifail is set to zero and if no further operation is possible because of the failure in the check processing, ifail is increased by one, and so on.

The type-0 operation is designed to construct the last two elements when the number of key nodes on the boundary reaches four and the remaining area is a quadrilateral. We construct the last two elements as illustrated in Fig. 6.

The nodes on the original boundary are considered to be of level one, the newly generated nodes from the original boundary are assigned to level two, and so on. In Sadek's algorithm, the level concept has been used to indicate a certain hierarchical order such that a boundary node of level two can be cut to form elements only after all the boundary nodes of level one have been removed.

This process is appropriate when the aspect ratio of an analysis domain is close to one. However, if the aspect ratio is much larger than one, the ratio generally increases due to the removal of the layer from the boundary, which may result in a bottle-neck-like region on a loop-boundary. Hence, as a modification in our scheme, the level of a node has been used only to determine the order of the basic operations among the candidate nodes on the current loop-boundary nodes. Therefore, conceptually, our scheme is to generate elements by removing corner key nodes ($\phi_i \leq 150^\circ$) from the boundary, while Sadek's algorithm in essence generates elements by removing a boundary layer from the boundary.

If the element sizes change drastically during the triangulation process, a loop-boundary may sometimes overlap itself or have a bottle-neck-like region, thus generating an ill-conditioned mesh, or even making further operations impossible. In order to

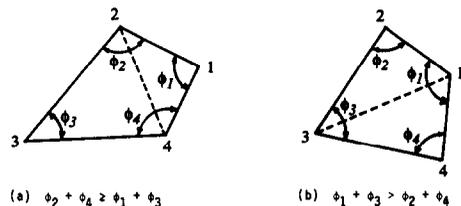
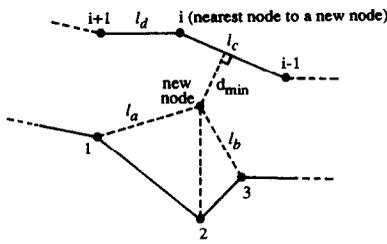


Fig. 6. Suggested type-0 operation.



$$d_{min} \geq \frac{1}{2} \cdot r \cdot \text{Average}(l_a, l_b, l_c, l_d)$$

$$\text{where } r = \frac{1}{1+ifail} \quad \text{ifail} = 0, 1, 2, \dots$$

Fig. 7. Minimum distance check.

avoid this unstable phenomenon, the following check processings have been designed. We call these tests an overlap check and a minimum distance check.

In the overlap check, the new edges of the generated triangular elements are tested to determine whether any overlapping occurs between these edges and the remaining loop-boundary edges. In the minimum distance check, a new key node of the type-2 operation is tested to discover whether the new node has enough distance to the remaining loop-boundary edges in order to avoid any bottle-neck-like region in the loop-boundary (see Fig. 7).

Because of the successive steps from the boundary toward the inside of the domain, the triangulation process is path-dependent. In addition to the definitions of a type-1 and type-2 key node, the order of the basic operations is also important. In order to decide the order of the basic operations, the following heuristic rules are employed in our scheme.

(a) The type-1 operation should be performed prior to a type-2 operation.

(b) Type-2 nodes are sorted to decide the order of applying the operation to the nodes by the following factors, in order of importance.

- (i) Low level.
- (ii) Large edge-length ratio, δ_i .
- (iii) Small adjacent edge-length, $l^* : l^* = \text{Min}\{l_i, l_{i-1}\}$.
- (iv) Small boundary angle, ϕ_j .

After the triangulation process has been completed for a given subregion domain using the above-mentioned basic operations, the resulting mesh is improved by the application of a smoothing process. In this process, the interior nodes are, by iterations, placed at the average of the centroid of the neighboring nodes and the current location, while the boundary node locations remain unchanged. An example of the smoothing process is shown in Fig. 8.

The triangulation scheme developed in this research has a certain directionality in the mesh generation. There are two reasons for this phenomenon. One reason is that the basic operations are performed in sequence, which changes the loop-boundary conditions after every operation. Hence, the loop-boundary may not be symmetric during the triangulation process, although it is symmetric initially. The other reason is due to the counter-clockwise direction used for ordering the key nodes.

3. VOLUME TRIANGULATION

The mesh generation in 3D objects presents some considerable additional difficulties when compared to 2D analysis due to the topological and finite element analysis requirements [10–14].

The basic process of our volume triangulation can be considered to be an extension of the surface triangulation discussed above.

Assume that the surfaces of the 3D object under consideration have been triangulated. The volume triangulation then starts from the outside surfaces toward the inside by cutting the sharp corner edges of a component loop-boundary using basic operations. Initially, all the edges on a loop-boundary are assigned to be of level one and as the volume triangulation proceeds, the levels of new generated edges are increased by one for each operation.

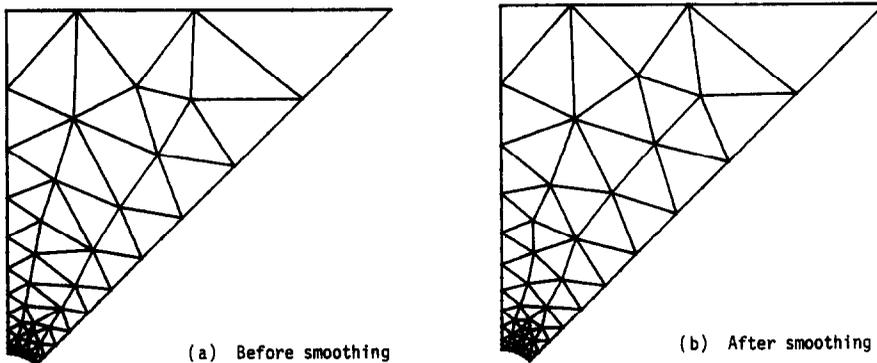


Fig. 8. Effect of smoothing process; square plate with a hole; 1/8th of plate is analyzed.

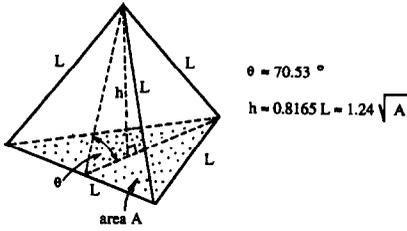


Fig. 9. Desired tetrahedral element.

Since the best form of tetrahedral elements in finite element analysis can be considered to be equilateral tetrahedra, the resulting elements are generated with the objective to obtain equilateral tetrahedra (see Fig. 9).

Due to the topological requirements described below, at least three basic operations are needed to reduce the number of edges and faces in a loop-boundary and, in our scheme, one more basic operation is used to construct the last two or three tetrahedral elements.

In order to describe the basic operations, the following definitions are introduced. Consider the edge KE in Fig. 10. The edge KE has two adjacent faces, a left face and a right face, and four surrounding edges, el_1, el_2, er_1 and er_2 . The adjacent faces of these surrounding edges are F_1, F_2, F_3 and F_4 respectively, and are called surrounding faces to the edge KE. The edge angle is the dihedral angle formed by the two adjacent faces, the left face and the right face. A face area ratio, δ_i ($\delta_i \geq 1.0$), at an edge i is an area ratio between two adjacent faces such as

$$\delta_i = \text{Max}(A_l/A_r, A_r/A_l) \quad (4)$$

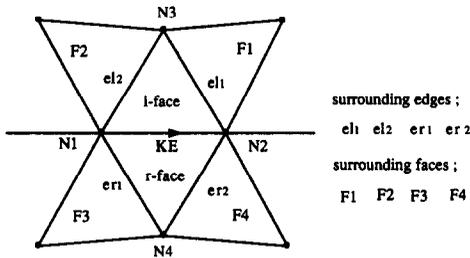


Fig. 10. Unfolded view around the edge KE.

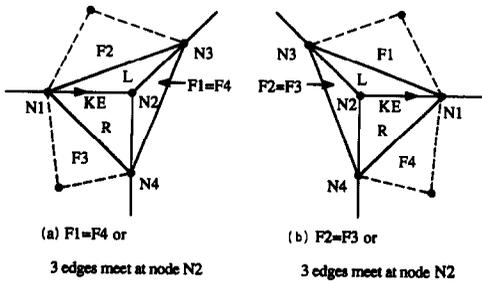


Fig. 11. Type-1A edge.

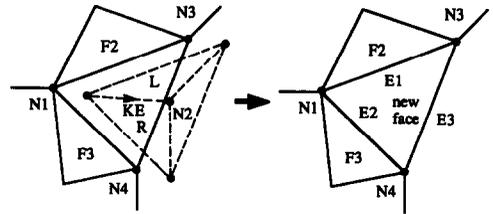


Fig. 12. Type-1A operation—trimming.

where A_l is the area of the left face and A_r is the area of the right face.

Four basic operations (type-1A, type-1B, type-2, type-0) are designed to perform the volume triangulation. The type-1A and type-1B operations are designed to generate one element at a time from a loop-boundary and the type-2 operation is designed to generate two elements. The type-0 operation is designed to construct the last two or three elements in order to complete the volume triangulation process.

Topologically, a type-1A corner edge is an edge for which two surrounding faces are identical as shown in Fig. 11. This condition is equivalent to the condition that only three edges meet the node N_2 . However, in order to generate well-conditioned elements, geometric restrictions are imposed, i.e. an edge angle should be less than or equal to 120° .

A type-1B corner edge is an edge for which all surrounding faces are different from each other, i.e. four or more edges meet at both end nodes. Such an edge is suitable for the construction of one well-conditioned tetrahedral element. For this operation an edge angle should be less than about 85° ($\delta_i \geq 1.8$) or 100° ($\delta_i < 1.8$) depending on the face area ratio, δ_i , and the face area ratios at the surrounding edges

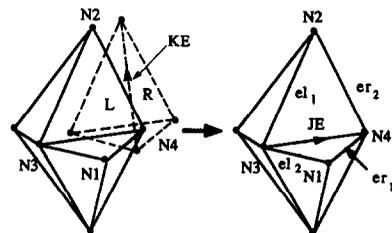


Fig. 13. Type-1B operation—wedging.

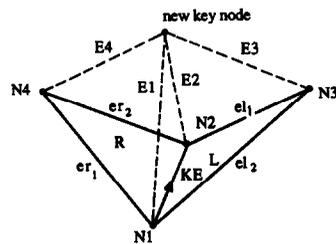


Fig. 14. Type-2 operation—digging.

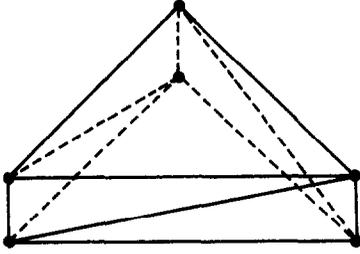


Fig. 15. Example of a pathological case.

should not change drastically due to the type-1B operation.

Since the desired tetrahedral element has an edge angle of about 70° , a type-2 edge for which two well-conditioned elements are constructed can have a maximum edge angle of 175° . Hence, a type-2 edge is defined as an edge for the edge angle less than 175° , and the edge is not a type-1A or a type-1B edge.

In order to use topologically valid operations, the basic operations (type-1A, type-1B and type-2) must produce results in which all the vertices, edges and faces added or deleted to an object satisfy both the Euler's formula and the following four conditions [15].

- (i) All faces are simply connected with no holes. They are topological disks.
- (ii) The solid object is simply connected and has no holes.
- (iii) Each edge adjoins exactly two faces and is terminated by a vertex at each end.
- (iv) At least three edges meet at each vertex.

Our basic operations generically satisfy the above conditions and Euler's formula is tested below for each basic operation.

Type-1A operation

A type-1A operation on a type-1A edge generates one tetrahedron by removing three faces, three edges and one key node from a loop-boundary and generates one new face as shown in Fig. 12. This operation can be considered to be a trimming process.

The type-1A operation reduces the number of loop-boundary edges and faces and key nodes ($\Delta E = -3$, $\Delta F = -2$, $\Delta V = -1$), which satisfies the derivative of Euler's formula,

$$\Delta V - \Delta E + \Delta F = 0. \quad (5)$$

After a type-1A operation, the edge angles of the remaining edges E_1 , E_2 , E_3 have also been reduced. In addition, the topological conditions on nodes N_1 , N_3 , N_4 have been changed so that the number of edges connected to these nodes have been reduced by one. Therefore, this operation can generate new type-1A edges around nodes N_1 , N_3 , N_4 when the number of edges connected to these nodes becomes three due to this operation.

Type-1B operation

This operation is designed to generate one tetrahedral element at a type-1B corner edge as shown in Fig. 13. The operation can be considered to be a wedging process. In this operation, one edge and two faces are removed and one new edge and two new faces are introduced ($\Delta V = 0$, $\Delta E = 0$, $\Delta F = 0$). Therefore, the derivative of Euler's formula is also satisfied,

$$\Delta V - \Delta E + \Delta F = 0. \quad (6)$$

Due to this operation, the edge angles of surrounding edges el_1 , el_2 , er_1 , er_2 are decreased so that it promotes to generate type-1A, type-1B or type-2 edges. Topologically, for nodes N_1 and N_2 , the number of edges connected to these nodes has decreased by one due to the removal of the edge KE , while for nodes N_3 and N_4 , the number of edges has increased by one due to the introduction of the new edge JE . Hence, this operation can generate new type-1A edges around nodes N_3 and N_4 .

Type-2 operation

The type-2 operation is designed to generate two tetrahedral elements at a type-2 corner edge by introducing a new key node as shown in Fig. 14. This operation can be considered to be a digging process.

In this operation, one edge and two faces are removed and four new edges, four new faces, and one new key node are introduced ($\Delta E = +3$, $\Delta F = +2$, $\Delta V = +1$). Therefore, this operation also satisfies the derivative of Euler's formula,

$$\Delta V - \Delta E + \Delta F = 1 - 3 + 2 = 0. \quad (7)$$

The operation is necessary not only for constructing well-conditioned elements but also for some pathological cases as shown in Fig. 15. In Fig. 15, no more operations are possible if only the type-1A and the type-1B operations are considered. Therefore, a type-2 operation which generates a new key node inside the loop-boundary is necessary.

The geometrical contribution of this operation to the entire volume triangulation process is to reduce the surrounding edge angles at el_1 , el_2 , er_1 , er_2 so that it promotes to generate type-1A, type-1B and type-2 edges among the edges el_1 , el_2 , er_1 and er_2 . Topologically, the number of edges connected to nodes N_3 , N_4 is increased by one, while the number of edges connected to nodes N_1 , N_2 remains unchanged. Thus, the primary goal of this operation is to generate type-1B edges and type-2 edges by reducing the edge angles. However, sometimes this operation can generate type-1A edges by reducing the edge angle when an edge has one common surrounding face and the edge angle is greater than 120° .

Since the number of edges, faces and nodes of the loop-boundary is increased by this operation, this operation has an adverse effect on the volume trian-

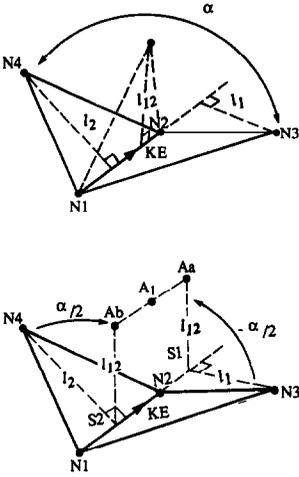


Fig. 16. Generation of a new key node at an edge KE.

gulation process, in which the number of edges, faces and nodes is to be reduced. However, as already mentioned, the operation is sometimes necessary to generate type-1A or type-1B edges.

The type-2 operation, in which a new key node is generated, is more complicated than the type-1A and type-1B operations, in which only the connectivity information of a loop-boundary is changed. A type-2 operation is designed to be adaptive and is composed of two steps. First a new key node is generated considering the effects of surrounding faces and edges and then, the new key node position is adjusted according to a scaling factor.

Consider the first step of a type-2 operation (see Fig. 16). In order to construct two well-conditioned tetrahedral elements, the height of a new face l_{12} is taken as $\sqrt{l_1 l_2}$ with the following conditions:

$$\frac{l_1}{l_{12}} = \frac{l_{12}}{l_2} \quad \text{or} \quad l_{12} = \sqrt{l_1 l_2}. \quad (8)$$

The position A_a is determined by rotating a vector of length l_{12} in the direction $S1-N3$ with respect to the axis $N1-N2$ by the angle of $-\alpha/2$,

$$\begin{aligned} \vec{OA}_a &= \vec{OS}_1 + \vec{S}_1 A_a \\ &= \vec{OS}_1 + l_{12} \left[\cos\left(-\frac{\alpha}{2}\right) \vec{e}_2 + \sin\left(-\frac{\alpha}{2}\right) \vec{e}_1 \times \vec{e}_2 \right] \end{aligned} \quad (9)$$

where

$$\vec{e}_1 = \frac{\vec{N}_1 \vec{N}_2}{|N_1 N_2|}, \quad \vec{e}_2 = \frac{\vec{S}_1 \vec{N}_3}{|S_1 N_3|}.$$

Similarly, the position A_b is determined by rotating a vector of length l_{12} in the direction $S2-N4$ with

respect to the axis $N1-N2$ by the angle of $\alpha/2$,

$$\begin{aligned} \vec{OA}_b &= \vec{OS}_2 + \vec{S}_2 A_b \\ &= \vec{OS}_2 + l_{12} \left[\cos\left(\frac{\alpha}{2}\right) \vec{e}_3 + \sin\left(\frac{\alpha}{2}\right) \vec{e}_1 \times \vec{e}_3 \right] \end{aligned} \quad (10)$$

where

$$\vec{e}_3 = \frac{\vec{S}_2 \vec{N}_4}{|S_2 N_4|}.$$

Using the height of a new face, the new candidate key node position A_1 is determined as the average of the positions, A_a and A_b ,

$$\vec{OA}_1 = \frac{1}{2} (\vec{OA}_a + \vec{OA}_b). \quad (11)$$

In order to include the effects of surrounding edges and faces, the procedure is repeated at the surrounding edges, $el1$, $el2$, $er1$ and $er2$, to obtain the candidate positions, A_2 , A_3 , A_4 and A_5 , respectively.

The number of tetrahedral elements that can be generated at an edge i with an edge angle ϕ_i is taken as the nearest integer of $\phi_i/70^\circ$ so that the tetrahedra are as equilateral as possible. However, if a surrounding edge angle is greater than 245° for which more than three elements should be generated, the effect of this edge is neglected for convenience. If the surrounding edge angle is less than or equal to 175° for which two elements are to be generated, the new key node is generated using the above procedure.

If the surrounding edge angle is between 175° and 245° , the new key node is generated as shown in Fig. 17. In order to construct three well-conditioned elements, the height of a new face l_i is taken as

$$l_i = 3\sqrt{l_a^2 l_b}. \quad (12)$$

The position of A_a is determined as

$$\begin{aligned} \vec{OA}_a &= \vec{OC}' + \vec{C}' A_a \\ &= \vec{OC}' + l_i \left[\cos\left(-\frac{\alpha}{3}\right) \vec{e}_2 + \sin\left(-\frac{\alpha}{3}\right) \vec{e}_1 \times \vec{e}_2 \right] \end{aligned} \quad (13)$$

where

$$\vec{e}_1 = \frac{\vec{AB}}{|AB|}, \quad \vec{e}_2 = \frac{\vec{C}'\vec{C}}{|C'\vec{C}|}.$$

Similarly, the position of A_b is determined as

$$\begin{aligned} \vec{OA}_b &= \vec{OD}' + \vec{D}' A_b \\ &= \vec{OD}' + l_i \left[\cos\left(\frac{2\alpha}{3}\right) \vec{e}_3 + \sin\left(\frac{2\alpha}{3}\right) \vec{e}_1 \times \vec{e}_3 \right] \end{aligned} \quad (14)$$

where

$$\vec{e}_3 = \frac{\vec{D}'\vec{D}}{|D'\vec{D}|}.$$

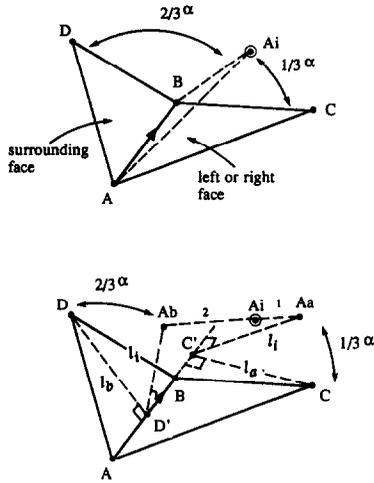


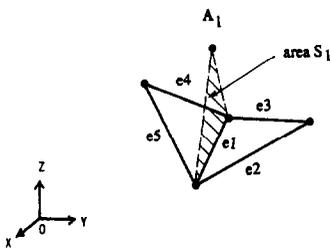
Fig. 17. New key node candidate from a surrounding edge ($175^\circ \leq \phi_i < 245^\circ$).

The new key node position A_i is obtained as the weighted average of the positions A_a and A_b ,

$$\vec{OA}_i = \frac{1}{3}(\vec{OA}_a + \vec{OA}_b). \quad (15)$$

After obtaining five candidate positions for a new key node from an edge of interest and four surrounding edges, weighting factors are used to determine a unique position of a new key node. Since the shapes of small elements are more distorted by shifting a new key node than the shapes of large elements, the weighting factors are taken in inverse proportions to the area of each new face, as shown in Fig. 18.

After obtaining the new key node position A , the next step of an adaptive type-2 operation is to adjust the key node position from A to A^* using a scaling factor r , as shown in Fig. 19. The scaling factor r is



$$w_i \propto \frac{1}{S_i} \quad i = 1, 2, \dots, 5$$

$$\text{Let } w_i = \frac{1}{S_i} \cdot \frac{1}{w_0}$$

$$\text{where } w_0 = \frac{1}{\sum_{i=1}^5 \frac{1}{S_i}} = \left(\frac{1}{S_1} + \frac{1}{S_2} + \dots + \frac{1}{S_5} \right)^{-1}$$

$$\vec{OA} = w_1 \cdot \vec{OA}_1 + w_2 \cdot \vec{OA}_2 + w_3 \cdot \vec{OA}_3 + w_4 \cdot \vec{OA}_4 + w_5 \cdot \vec{OA}_5$$

Fig. 18. Key node position using the weighting factors.

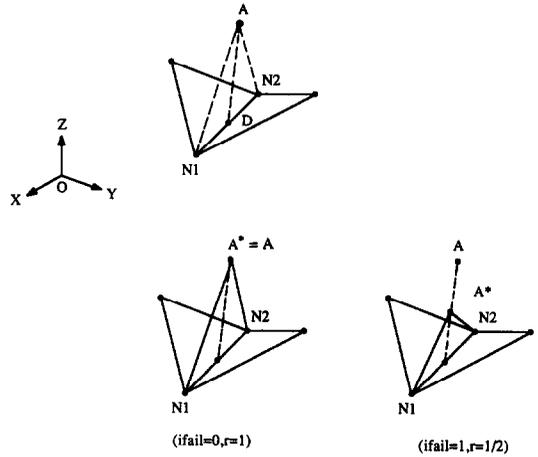


Fig. 19. Adjustment of a new key node position.

defined as

$$r = \frac{1}{1 + \text{ifail}}, \quad \text{ifail} = 0, 1, 2, \dots \quad (16)$$

where ifail is the number of failures to perform any type-1A, type-1B and type-2 operation. Initially, ifail is set to zero and if no more basic operation is possible because of failure in the overall check or minimum distance check (see below), ifail is increased by one, and so on.

Type-0 operation

The type-0 operation is designed to construct the last two or three elements when the number of edges in a loop-boundary reaches nine. With these nine

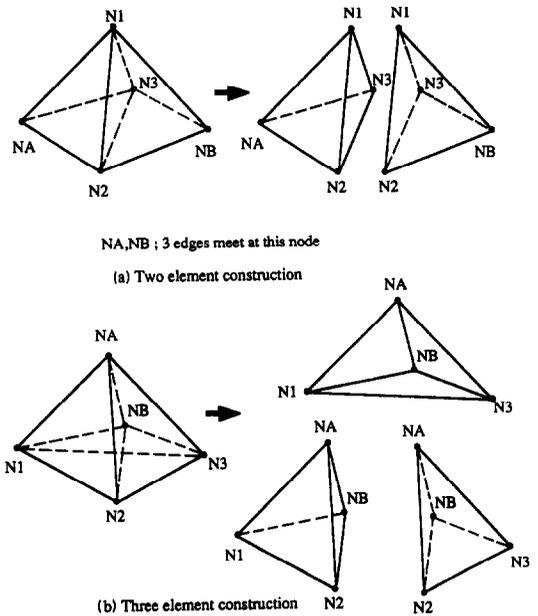


Fig. 20. Type-0 operation.

edges, two kinds of polyhedra (loop-boundaries) are possible and the corresponding mesh constructions are shown in Fig. 20.

As shown in Fig. 20, the type-0 operation is nothing else but a combination of the type-1A and type-1B operations. Therefore, it also satisfies the derivative of Euler's formula. However, in order to avoid any overlapping between elements, a special operation is needed to complete the tetrahedronization. This is achieved by the type-0 operation.

Remark

The surface triangulation on a component satisfies the following Euler's formula with V key nodes, E edges and F faces,

$$V - E + F = 2. \quad (17)$$

The resulting polyhedron has a large number of faces, edges and nodes, and the volume triangulation is to reduce the number of faces, edges and nodes by using basic operations and finally to obtain one tetrahedron ($V = 4, E = 6, F = 4$). Since the basic operations satisfy the derivative of Euler's formula, the resulting polyhedron after each operation will also satisfy Euler's formula. Among the basic operations, only the type-1A operation contributes to reducing the number of faces, edges and nodes ($\Delta V = -1, \Delta E = -3, \Delta F = -2$), hence enough type-1A operations must be performed to complete the volume triangulation process. A type-1B operation ($\Delta V = 0, \Delta E = 0, \Delta F = 0$) is performed in order to generate type-1A edges when no more type-1A edges are available. The type-2 operation ($\Delta V = 1, \Delta E = 3, \Delta F = 2$) is performed to generate a type-1B operation when both type-1B and type-1A operations are not possible. Since this operation increases the number of faces, edges and nodes of a loop-boundary, one additional type-1A operation is necessary to compensate for this adverse effect if a type-2 operation is performed.

Therefore at least three basic operations (type-1A, type-1B, type-2) are required for the volume triangulation process. Starting from an initial polyhedron, the volume triangulation process can always be completed by using these basic operations, provided there is no overlapping of loop-boundary edges and faces during the process. In order to avoid any overlapping or bottle-neck-like region, check processes are designed for the process.

Check processing

A check processing is necessary during the volume triangulation process for an overlap check and minimum distance check. In the overlap check the new faces of the generated tetrahedral elements are tested to determine whether any overlapping occurs between these faces and the remaining loop-boundary faces. In the actual computer implementation it is easier to check the overlapping between new faces and the remaining loop-boundary edges rather than

to check between new faces and the remaining loop-boundary faces. In the check, one new face is tested in the type-1A operation, two new faces are tested in the type-1B operation and four new faces are tested in the type-2 operation.

In the minimum distance check, a new key node is tested to discover whether it has enough distance to the remaining loop-boundary faces. The minimum distance requirement to a loop-boundary face also changes according to the scaling factor r in eqn (16).

As in the surface triangulation process, important issues in the proposed volume triangulation process are (i) how to define type-1B and type-2 corner edges among the loop-boundary edges and (ii) in which order the basic operations should be performed. The proposed scheme has some heuristic rules which have been identified in an effort to imitate human-like reasoning during the tetrahedronization process.

Namely, in order to decide the order of the basic operations, the following heuristic rules are employed in our scheme.

(i) The type-1A operation should be performed prior to a type-1B operation, and the type-1B operation should be performed prior to a type-2 operation.

(ii) Among the same types of operations, the order of operation is also an important factor especially in the case of the type-2 operation. Type-2 edges are sorted to decide the order of applying the operation to the edges by the following factors, in order of importance.

- Low level.
- Large adjacent face area ratio, δ_i .
- Small adjacent face area,
 $A_i^*: A_i^* = \text{Min}\{A_i, A_{i-1}\}$.
- Small edge angle, ϕ_i .

Smoothing process

After the volume triangulation process has been completed for a given component, the resulting mesh is improved by a smoothing process. In this procedure, the internal nodes are placed at the average of the centroid of neighboring nodes and the current location, while the boundary nodes remain unchanged,

$$X_{\text{new}} = \frac{1}{2}(X_{\text{centroid}} + X_{\text{old}}) \quad (18)$$

where

$$X_{\text{centroid}} = \frac{1}{N} \sum_i^N X_i |_{\text{nodes connected}}.$$

If $|X_{\text{new}} - X_{\text{old}}| \leq e$ for all the internal nodes, the smoothing process has converged, where $e = L_{\text{system}}/500$ and L_{system} is the characteristic length of the system. We have observed that 5–10 iterations are usually enough to converge to a given tolerance

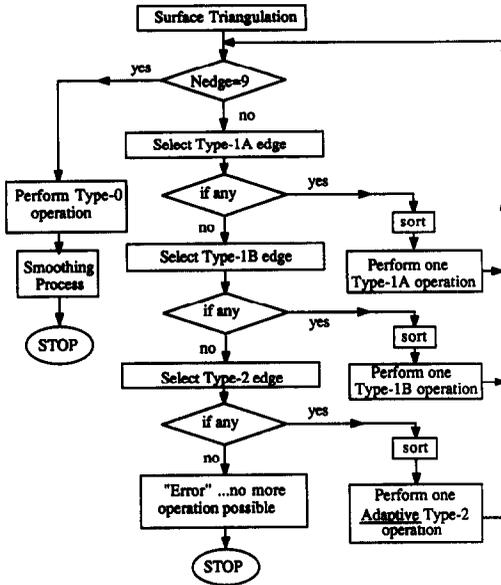


Fig. 21. Flow chart of the volume triangulation process.

of $L_{system}/500$. The flow chart of the entire volume triangulation procedure is shown in Fig. 21.

4. ERROR ANALYSIS

Error analysis has been recognized as an essential task both for assessing the quality of finite element solutions and for an adaptive mesh refinement process [3, 16].

The error in finite element analysis can be taken as any measure between the exact solution and the finite element solution. However, since the exact solution is not generally known, certain features of the exact solution must be used to indicate the error, or the exact solution needs to be estimated to compute (or indicate) the error involved.

The approach used in this paper considers the equilibrium equations to estimate the error. In this case, two sources of error should be considered: the body force residual error in each element domain and the traction jumps between adjacent elements.

Babuska and Yu [17] have shown that, measured in the energy norm, in the case of linear displacement-based elements the traction jumps along the element boundaries are dominant in the total error while for

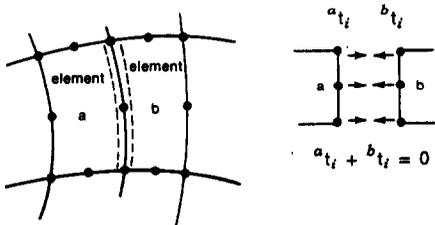


Fig. 22. Traction jumps between adjacent elements.

quadratic displacement-based elements the body force residual error provides the dominant part. As described below, our experiences with the quadratic elements lead to similar observations.

Consider the general elasticity problem in the domain Ω for which the equilibrium equation is

$$\tau_{ij,j} + f_i^B = 0 \quad \text{in } \Omega \tag{19}$$

with the boundary conditions

$$\tau_{ij}n_j - t_i^S = 0 \quad \text{on } \Gamma_t \tag{20}$$

$$u = u_0 \quad \text{on } \Gamma_u \tag{21}$$

and

$$\Gamma = \Gamma_t \cup \Gamma_u.$$

For each finite element m two sources of error are measured

$$R_i^{FE} = \tau_{ij,j} + f_i^B \neq 0 \quad \text{in } \Omega_m \tag{22}$$

$$T_i^{FE} = \tau_{ij}^F n_j - t_i \neq 0 \quad \text{on } \Gamma_m \tag{23}$$

where τ_{ij}^{FE} represents the finite element solution and t_i is the exact unknown traction transmitted through the boundary of an element.

Using these two error sources, the body force residual error, $F_i^{\Omega m}$, and the traction residual error, $F_i^{\Gamma m}$, can be defined

$$F_i^{\Omega m} = \int_{\Omega_m} R_i^{FE} d\Omega_m \tag{24}$$

$$F_i^{\Gamma m} = \int_{\Gamma_m} T_i^{FE} d\Gamma_m \tag{25}$$

where clearly $F_i^{\Omega m} = F_i^{\Gamma m}$.

Hence, the integral of the body force residual error is equal to that of the traction residual error.

Generally, we cannot compute the traction residual error directly, because the exact force transmitted through the element boundary is not known. Instead, the traction jumps between adjacent elements can be used to estimate the traction residual error if a proper error allocation algorithm is available.

Consider 2D analysis and the two adjacent elements a, b in Fig. 22. The traction jump, TM_i , between the two elements a and b is composed of two contributions, aT_i and bT_i ,

$$\begin{aligned} TM_i &= ({}^a\tau_{ij}^{FE} - {}^b\tau_{ij}^{FE})n_j^a = {}^a\tau_{ij}^{FE}n_j^a + {}^b\tau_{ij}^{FE}n_j^b \\ &= ({}^a\tau_{ij}^{FE}n_j^a - {}^a t_i) + ({}^b\tau_{ij}^{FE}n_j^b - {}^b t_i) \\ &= {}^aT_i + {}^bT_i. \end{aligned} \tag{26}$$

As a simple error allocation algorithm, the traction jump between adjacent elements is equally distributed

to each element. For our numerical use, the two error indicators are defined as

$$F_{\Omega_m} = \int_{\Omega_m} (R_y^2 + R_z^2)^{1/2} d\Omega_m \quad (27)$$

$$F_{\Gamma_m} = \frac{1}{2} \int_{\Gamma_m} (TM_y^2 + TM_z^2)^{1/2} d\Gamma_m \quad (28)$$

Another common measure is the error in the energy norm

$$\|e\|^2 = \sum_{m=1}^N \int_{\Omega_m} e^T (B^T C B e) d\Omega_m \quad (29)$$

where the error, e , is defined as the difference between the exact displacement u and the approximate solution \bar{u}

$$e = u - \bar{u} \quad (30)$$

and N is the number of finite elements in the mesh.

In eqn (29), B is the strain–displacement and C is the material stress–strain matrix.

Babuska and Rheinboldt [18] have proved that there exist constants $k_2 \geq k_1 > 0$ such that

$$k_1 \epsilon^2 \leq \|e\|^2 \leq k_2 \epsilon^2 \quad (31)$$

where

$$\epsilon^2 = \sum_{m=1}^N \left(C_1 h_m^2 \int_{\Omega_m} R^2 d\Omega_m + C_1 h_m \int_{\Gamma_m} J^2 d\Gamma_m \right) \quad (32)$$

where R is the domain residual, J is the inter-element traction jump and C_1 is a constant. In order to study which term is dominant in eqn (32) in the case of quadratic elements, we will consider the following terms,

$$E_{\Omega_m} = \frac{h_m^2}{E} \int_{\Omega_m} (R_y^2 + R_z^2) d\Omega_m \quad (33)$$

$$E_{\Gamma_m} = \frac{h_m}{2E} \int_{\Gamma_m} (TM_y^2 + TM_z^2) d\Gamma_m \quad (34)$$

where h_m is the characteristic length of element (m) and E is the material Young's modulus.

The above four error indicators in eqns (27), (28), (33) and (34) are non-dimensionalized with respect to the total external force and the total strain energy respectively as follows,

$$f_{\Omega_m} = \frac{F_{\Omega_m}}{F_{\text{external force}}} \quad (35)$$

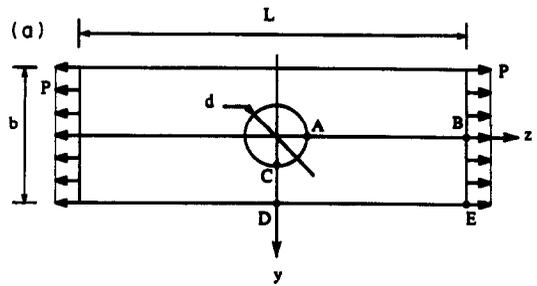
$$f_{\Gamma_m} = \frac{F_{\Gamma_m}}{F_{\text{external force}}} \quad (36)$$

$$e_{\Omega_m} = \frac{E_{\Omega_m}}{E_{\text{total strain energy}}} \quad (37)$$

$$e_{\Gamma_m} = \frac{E_{\Gamma_m}}{E_{\text{total strain energy}}} \quad (38)$$

The error indicators in eqns (35)–(38) were computed for the examples in Figs 23–25 for plane stress, plane strain, and axisymmetric conditions. In this case, 8-node quadrilateral elements were used for the error analysis. The results for the calculation of the error indicators are summarized in Table 1.

One interesting point in Table 1 is that the traction jump error and the body force residual error in the force norm are almost the same in their magnitudes, while in the energy norm, the body force residual



- $L = 56 \text{ mm}$
- $b = 20 \text{ mm}$
- $d = 10 \text{ mm}$
- $h = 1 \text{ mm (thickness)}$
- $E = 7.0 \times 10^4 \text{ N/mm}^2$
- $\nu = 0.25$
- $P = 25.0 \text{ N/mm}^2$

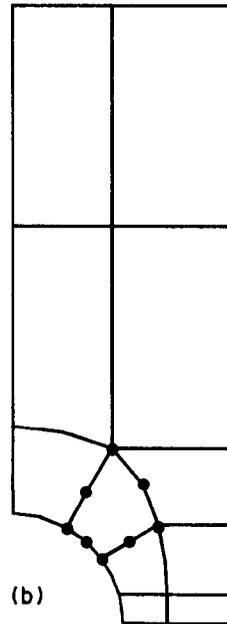
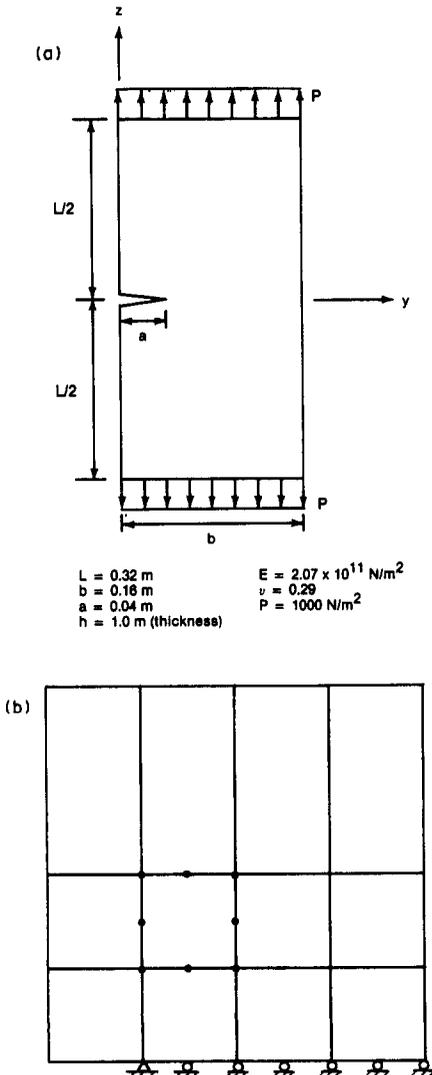


Fig. 23. (a) Plate with a hole in tension. (b) Eleven element model.

Table 1. Comparison of the error indicators

	Non-dimensionalized (w.r.t. force)		Non-dimensionalized (w.r.t. energy)	
	Traction $\sum_m f_{r_m}$	Residual $\sum_m f_{\Omega_m}$	Traction $\sum_m e_{r_m}$	Residual $\sum_m e_{\Omega_m}$
A. Structure with a hole (Fig. 23)				
(1) Plane stress ($t = 1$)	1.03	1.21	0.0692	0.14
(2) Plane strain	1.10	1.27	0.0829	0.169
(3) Axisymmetric	0.49	0.533	0.0202	0.0424
B. Structure with a crack (Fig. 24)				
(1) Plane strain	0.959	0.949	0.00777	0.593
(2) Plane stress ($t = 0.01$)	0.950	0.902	0.00747	0.506
(3) Axisymmetric	0.179	0.172	0.00133	0.13
C. Structure of a vessel form (Fig. 25)				
(1) Axisymmetric	3.50	4.71	0.00779	0.424
(2) Plane strain	4.32	5.86	0.00288	0.148
(3) Plane stress ($t = 0.1$)	2.99	4.20	0.00118	0.053



error is the dominant one. The residual error in the energy norm is about twice the traction jump error for the problem in Fig. 23, while for the problems in Figs 24 and 25, the residual term is about 50–100 times as large as the traction jump term. Hence, the body force residual in the energy norm is much larger than the traction jump residual for the problems with high stress concentrations.

In the numerical evaluations an error indicator using the body force residual is more efficiently computed than an indicator involving traction jumps. Also, in our tests, an error indicator in the energy norm was found to be more efficient than in the force norm for the adaptive refinement process. Therefore, the best choice of error indicator for an adaptive refinement process using quadratic elements appears to be the body force residual in the energy norm as given in eqns (33) and (37).

Using the error indicator in eqn (33), we observe the following in terms of efficiency. In most cases, we are interested in the solution of a stress concentration and the element size in this region should become more rapidly smaller than in the other regions during the refinement process. This aim in the refinement is enhanced by introducing the relaxation factor $1/h_m$; i.e. by multiplying the above error indicator by $1/h_m$. Namely, this way we scale up the error indicator of the small sized elements compared with the larger elements, consequently refinement is more concentrated in the region of the small sized elements. The result is that better solutions can be obtained at stress concentrations with little sacrifice on the overall accuracy. Therefore, we use the following error indicator in 2D and 3D solutions

$$\eta_m = \frac{h_m}{E} \int_{\Omega_m} (R_x^2 + R_y^2 + R_z^2) d\Omega_m \quad (39)$$

Fig. 24. (a) Crack tip problem. (b) Twelve element model. where h_m represents the characteristic length of ele-

ment (m), E is the Young's modulus, R_x , R_y and R_z are the body force residuals, and Ω_m is the volume of element m . Note that for quadratic displacement-based triangular 2D and tetrahedral 3D elements with straight sides, the values of R_x , R_y and R_z are constant within each element. As the reference strain energy

value, we use

$$U_{\text{reference}} = \frac{E_{\text{total strain energy}}}{L_{\text{system}}} \tag{40}$$

where $U_{\text{reference}}$ is the reference strain energy per unit length, $E_{\text{total strain energy}}$ is the total strain energy of the system, and L_{system} is the characteristic length of the system.

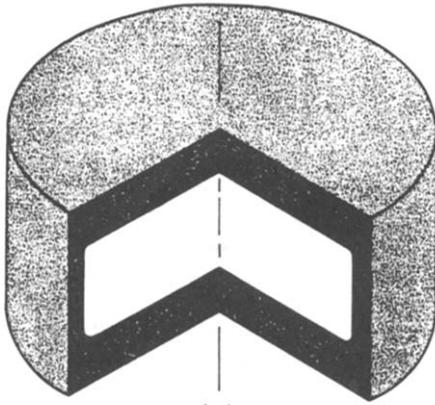
As a criterion for further refinement, a non-dimensionalized error indicator for each element m defined as follows is used,

$$\epsilon_m = \frac{\eta_m}{U_{\text{reference}}} \tag{41}$$

The total error indicator of the system is found to be

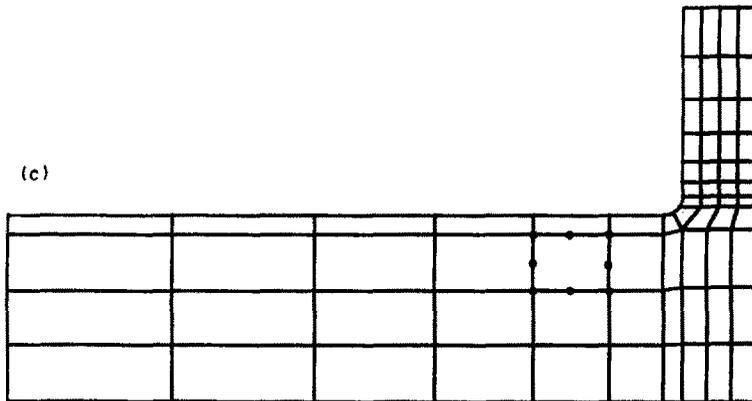
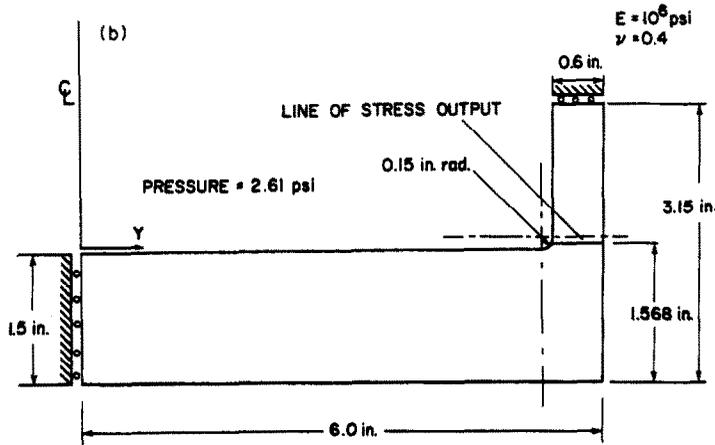
$$\epsilon_{\text{total}} = \sum_{m=1}^N \epsilon_m \tag{42}$$

Our numerical experiences with this error indicator show that the magnitude of the total error indicator in eqn (42) indicates reasonably well the amount of error involved in the finite element solution, although



(a)

a) Schematic drawing



(c)

Fig. 25. Pressure vessel problem; (a) schematic drawing; (b) dimensions and model analyzed; (c) 69 element model.

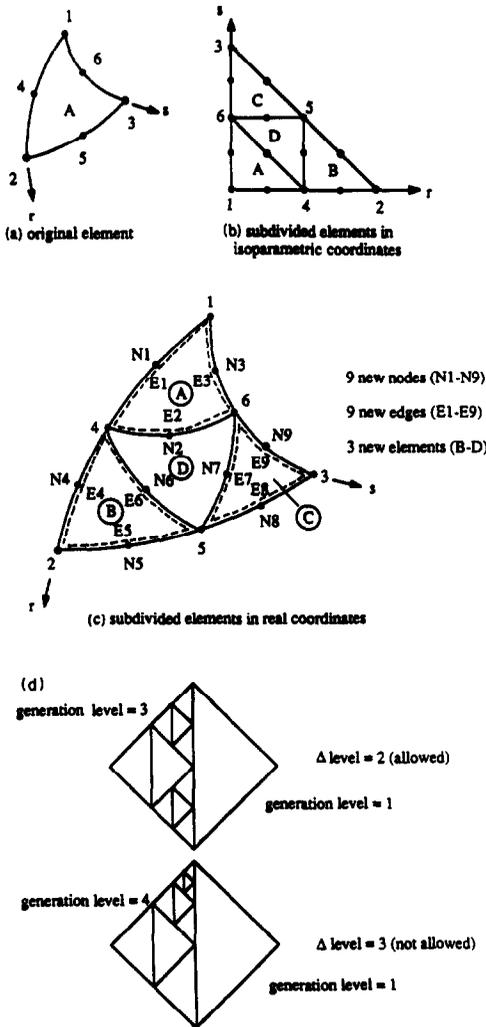


Fig. 26. (a)–(c) Refinement for a 6-node triangular element. (d) Limit of refinement.

the actual value of ϵ_{total} cannot directly be interpreted as a percentage error on the predicted stresses and displacements.

5. ADAPTIVE REFINEMENT IN 2D ANALYSIS

In our algorithm, the h-version of refinement has been adopted as a grid enrichment method, in which a 2D element is subdivided into four subelements. The refinement process is performed using the basic refinement unit shown in Fig. 26. During the refinement process, the compatibility conditions between adjacent elements should also be satisfied by using constraint equations. The constraint equation management scheme is designed to deal with a maximum difference of 2 in the generation level. Thus, up to 4 elements on one side with respect to one element on the other side are allowed, as shown in Fig. 26. If further refinement is necessary and the difference of generation level is already 2, the adjacent element

is refined first in order to keep the maximum difference 2. In the case of a boundary edge refinement, boundary conditions of new nodes are also generated appropriately and if there is any pressure load on the refined boundary edge, it is updated appropriately. In addition to the above process, all the edge connectivity data is updated.

6. SAMPLE SOLUTIONS

We present two examples of 2D analysis that demonstrate the use of our algorithms for the mesh generation and mesh refinement. There is no unique way in constructing effective meshes. If we have enough information about the problem already, we might be able to construct an effective mesh within a given accuracy of solution in one step by controlling the key node placements carefully on the loop-boundaries. Otherwise, we may need to construct a coarse mesh and perform the adaptive refinement process several times. However, in general, the desirable way would be to construct an initial mesh close to an effective mesh and limit the number of refinements to about two iterations.

In the first example, two different error indicators are employed for the refinement process in order to compare their efficiencies: first we use the energy norm without a relaxation factor as in eqn (37), and then we use the energy norm with a relaxation factor as in eqn (39). In the second example, the error indicator with the relaxation factor is used for the error analysis.

Finally, we present an example of a 3D solution in which a cylinder attached to a block is considered. In this example we only demonstrate the initial mesh generation for the 3D geometry.

Analysis of an axisymmetric pressure vessel

The first example concerns the analysis of the axisymmetric pressure vessel shown in Fig. 25. The goal of the analysis is to determine the stress distribution due to the internal pressure loading on the 'line of stress output' shown in Fig. 25(b).

This problem was first considered in a paper by Floyd [19] and more rigorous analyses have been performed by Sussman and Bathe [20]. In their paper, Sussman and Bathe give an effective mesh with 181 eight-node elements to obtain a reasonable accuracy of solution. This mesh is shown in Fig. 27 and the corresponding graphs of the principal in-plane stresses along the line of interest are shown in Fig. 28.

In order to be able to study the efficiency of the refinement process, a relatively coarse initial mesh was constructed. To obtain an initial mesh with well-conditioned elements, the analysis domain was subdivided into the four near-convex subdomains shown in Fig. 29.

The triangulation algorithm automatically constructed the 104 triangular element mesh shown in

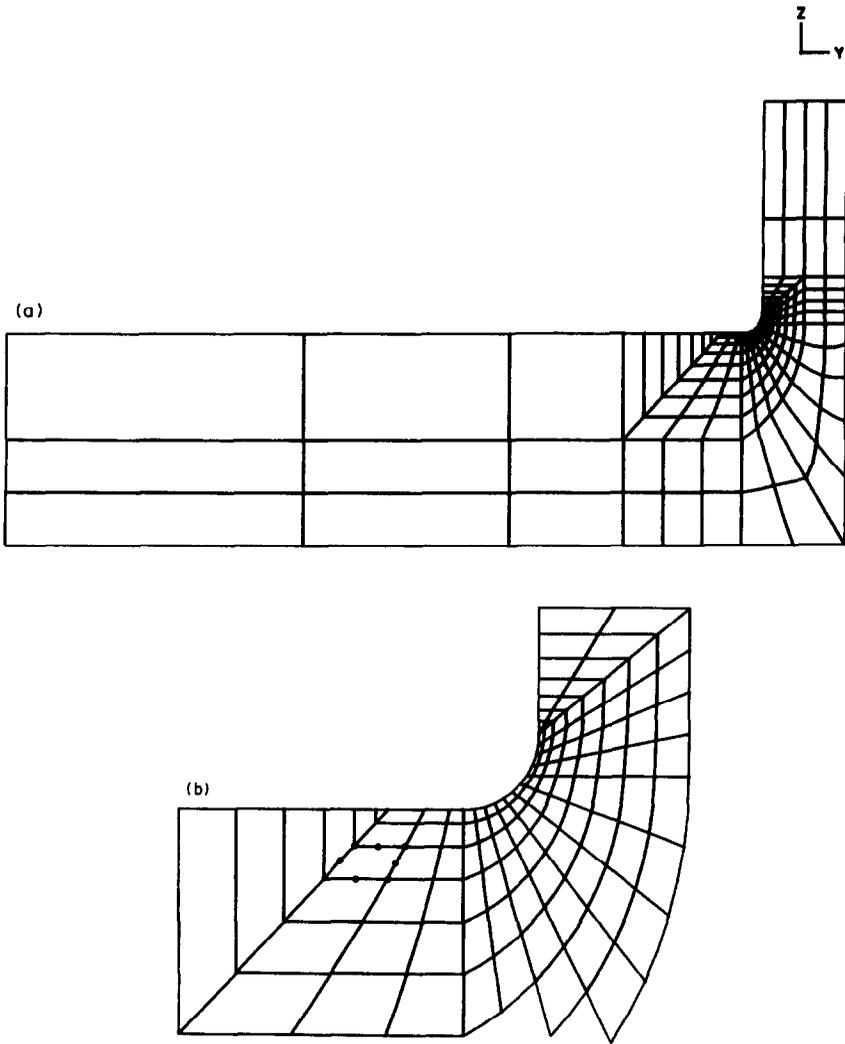


Fig. 27. 181 eight-node element mesh for axisymmetric pressure vessel; (a) complete mesh; (b) detail of mesh.

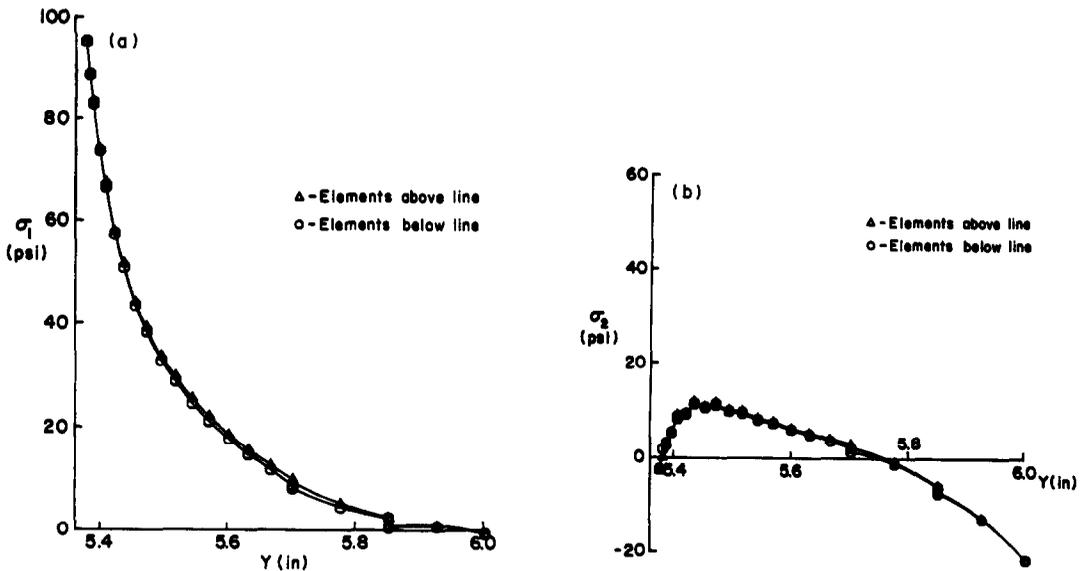


Fig. 28. (a) Maximum in-plane principal stress for 181 element mesh [20]. (b) Minimum in-plane principal stress for 181 element mesh [20].

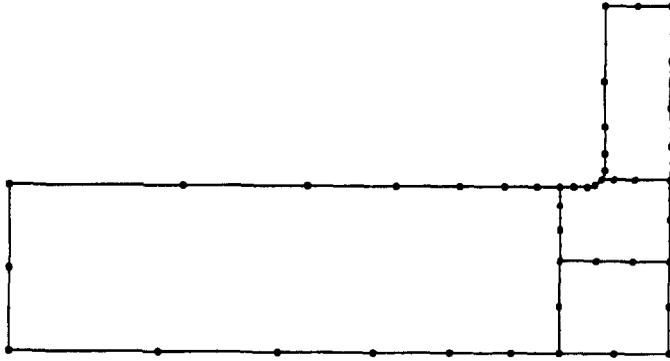


Fig. 29. Key nodes for a coarse mesh construction in analysis of the pressure vessel.

Fig. 30. For this mesh, the stresses are computed at the nodal points and the principal stress distributions on the 'line of stress output' are shown in Fig. 31. The figure shows the stresses calculated from the elements above the line of interest and from the element below that line. The results in the figure show large stress discontinuities between adjacent elements, which means that the mesh is not satisfactory.

If we use the error indicator with the relaxation factor [see eqn (41)] for the adaptive refinement process, a 281 element mesh is obtained in 2 steps of refinement (104 element \rightarrow 236 element \rightarrow 281 element) within a given error indicator tolerance $e_{\text{tolerance}} = 0.02$ (see Fig. 32), where $e_m \leq e_{\text{tolerance}}$ for all elements m . The stress output results for the 281 element mesh are shown in Fig. 33, and compare well with the earlier reported solutions [20]. Note that the tolerance $\epsilon_{\text{tolerance}}$ does not give the actual errors in the stresses.

If we use the error indicator without a relaxation factor as in eqn (37) with $e_{\text{tolerance}} = 0.02$ for the adaptive refinement process, a 278 element mesh is obtained in 2 steps of refinement (104 elements \rightarrow 266 elements \rightarrow 278 elements) (see Fig. 34). In this case, the indicator tolerance is chosen ($e_{\text{tolerance}} = 0.02$) to generate a similar number of elements as when using the relaxation factor after 2 steps of refinement in

order to compare the efficiencies of the two different error indicators.

The stress output results for this analysis are shown in Fig. 35. These stress predictions are not as satisfactory as those obtained for the 281 element mesh.

By comparing the 281 element mesh in Fig. 32 and the 278 element mesh in Fig. 34, we can see that in the 281 element mesh, the refinement is more concentrated at the stress concentration. However, the total strain energy in the 281 element mesh is less than the strain energy of the 278 element mesh.

The reason for the improved results is due to the use of the relaxation factor $1/h_m$ employed in the error indicator in eqn (41). This relaxation factor is employed so that the refinement is more concentrated in the region of the small sized elements, which results in better solutions at the stress concentration with little sacrifice on the overall accuracy. Therefore, we recommend the use of the error indicator with the relaxation factor in eqn (41) for the adaptive refinement process.

If we have enough information about an analysis problem already, such as the strengths of stress concentrations, we are usually able to reduce the number of refinement iterations by starting with an efficient initial mesh.

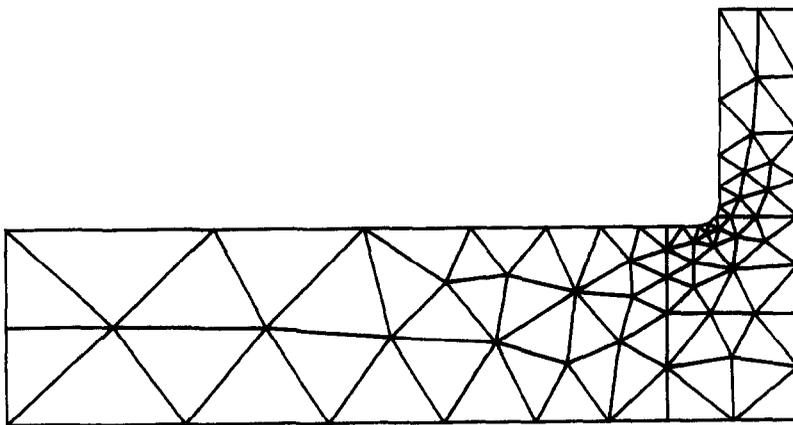


Fig. 30. Coarse mesh with 104 elements; total strain energy: 0.297365×10^{-2} lbf-in., $\epsilon_{\text{total}} = 7.78$.

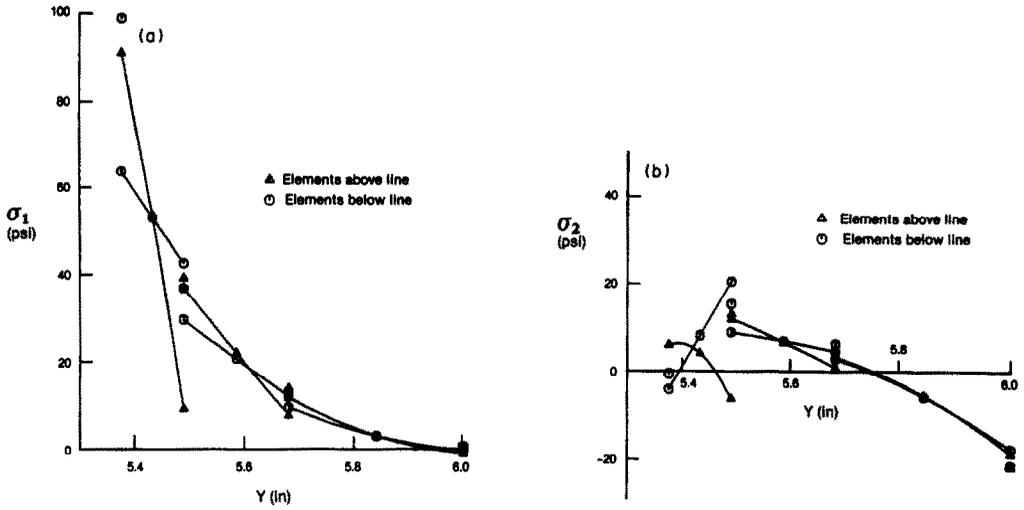


Fig. 31. (a) Maximum in-plane principal stress for 104 element mesh. (b) Minimum in-plane principal stress for 104 element mesh.

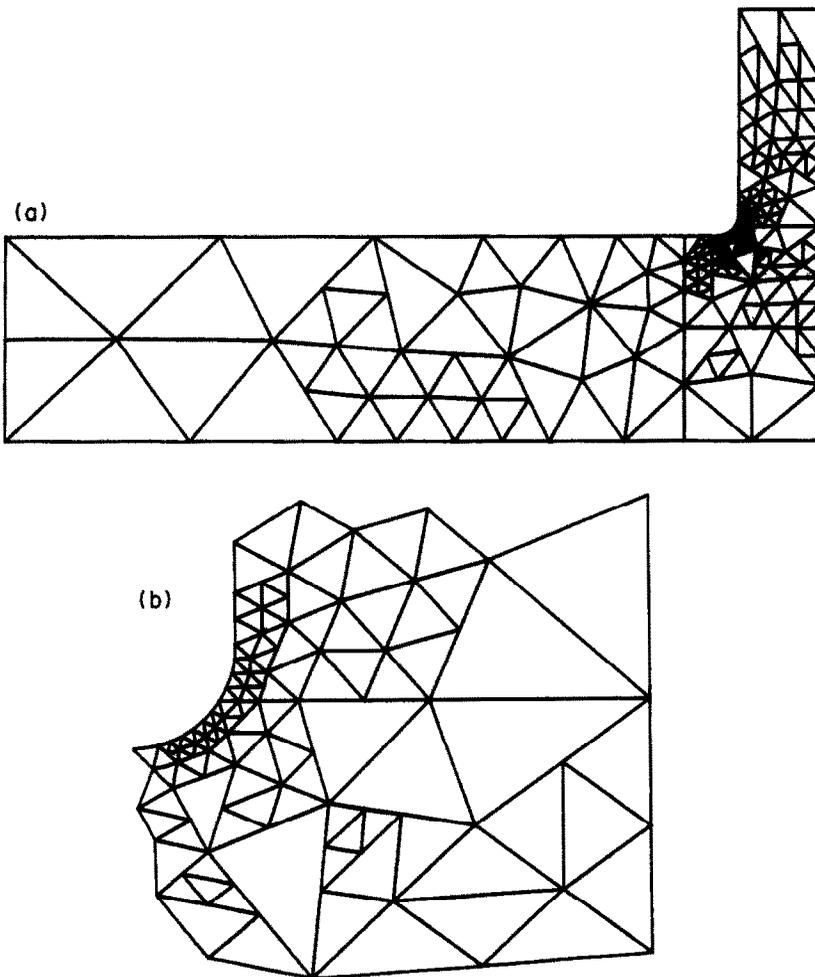


Fig. 32. (a) Final mesh with 281 elements using the error indicator with the relaxation factor; total strain energy: 0.298093×10^{-2} lbf-in., $\epsilon_{total} = 1.25$, $\epsilon_{tolerance} = 0.02$ (104 el \rightarrow 236 el \rightarrow 281 el). (b) Detail of 281 element mesh.

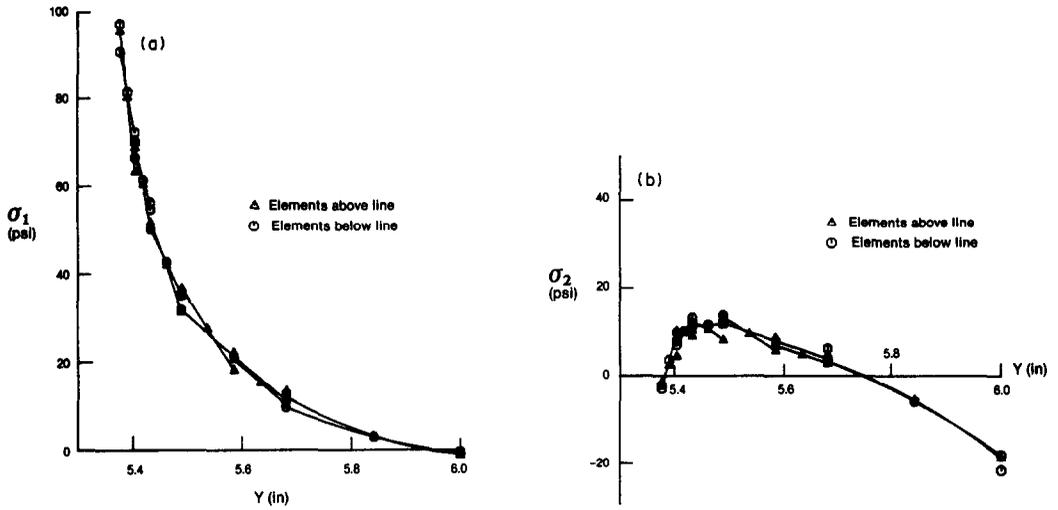


Fig. 33. (a) Maximum in-plane principal stress for 281 element mesh. (b) Minimum in-plane principal stress for 281 element mesh.

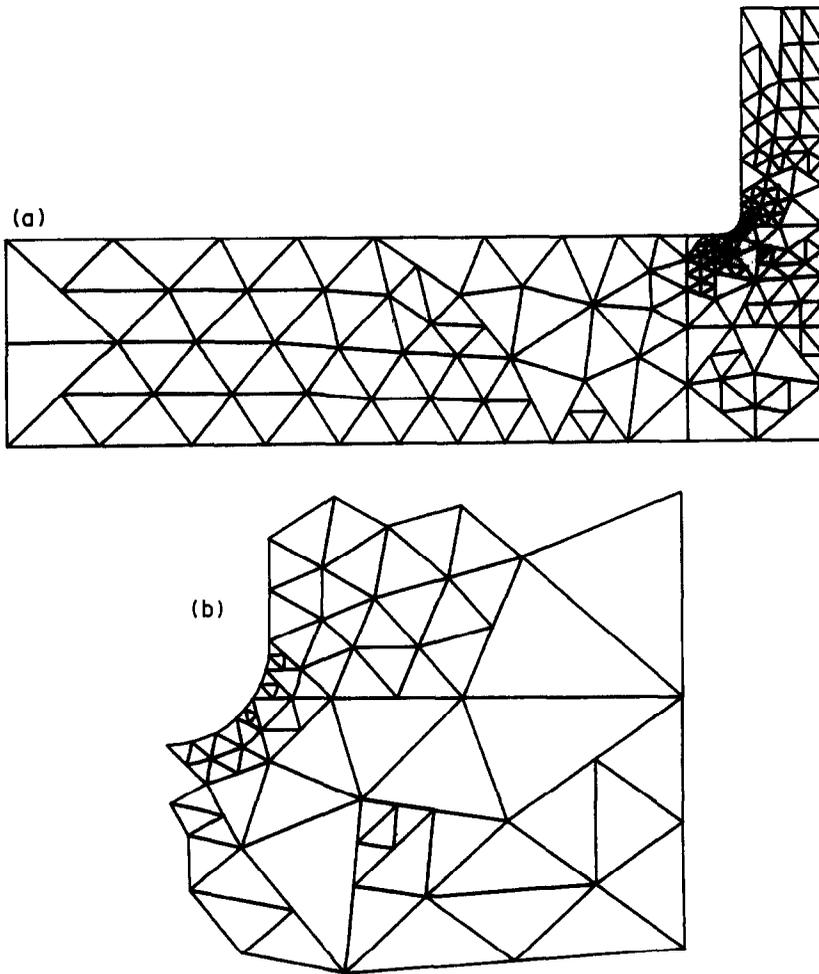


Fig. 34. (a) Final mesh with 278 elements using the error indicator without a relaxation factor; total strain energy: 0.298111×10^{-2} lbf-in., $e_{\text{tolerance}} = 0.02$ (104 el \rightarrow 266 el \rightarrow 278 el). (b) Detail of 278 element mesh.

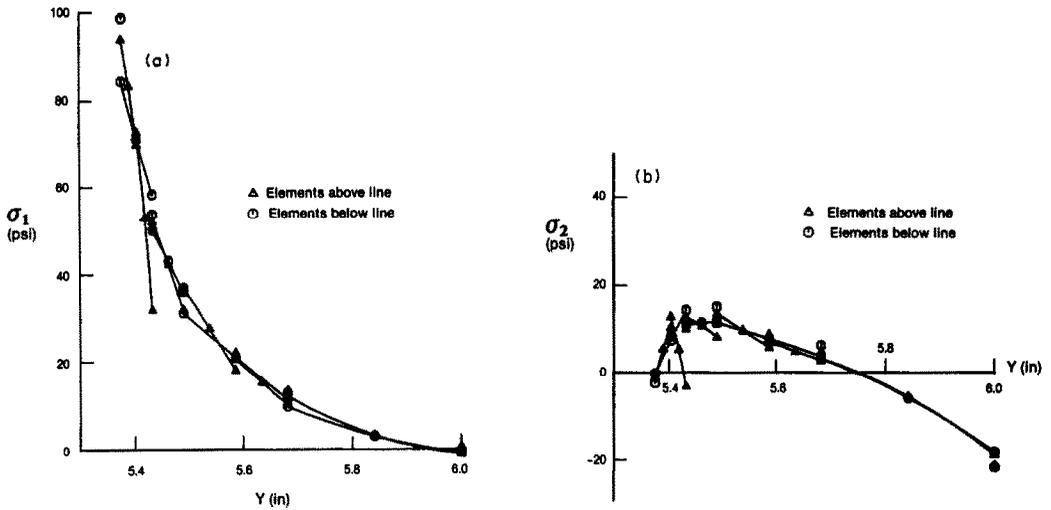


Fig. 35. (a) Maximum in-plane principal stress for 278 element mesh. (b) Minimum in-plane principal stress for 278 element mesh.

Based on the experience obtained in the above analysis, an efficient initial mesh was constructed using 221 elements as shown in Fig. 36. For this 221 element mesh, the adaptive refinement was performed

with the indicator tolerance $\epsilon_{\text{tolerance}} = 0.02$ and a 281 element mesh was obtained in one step (see Fig. 37). The stress output results for this mesh given in Fig. 38 show that better results can be obtained in only one

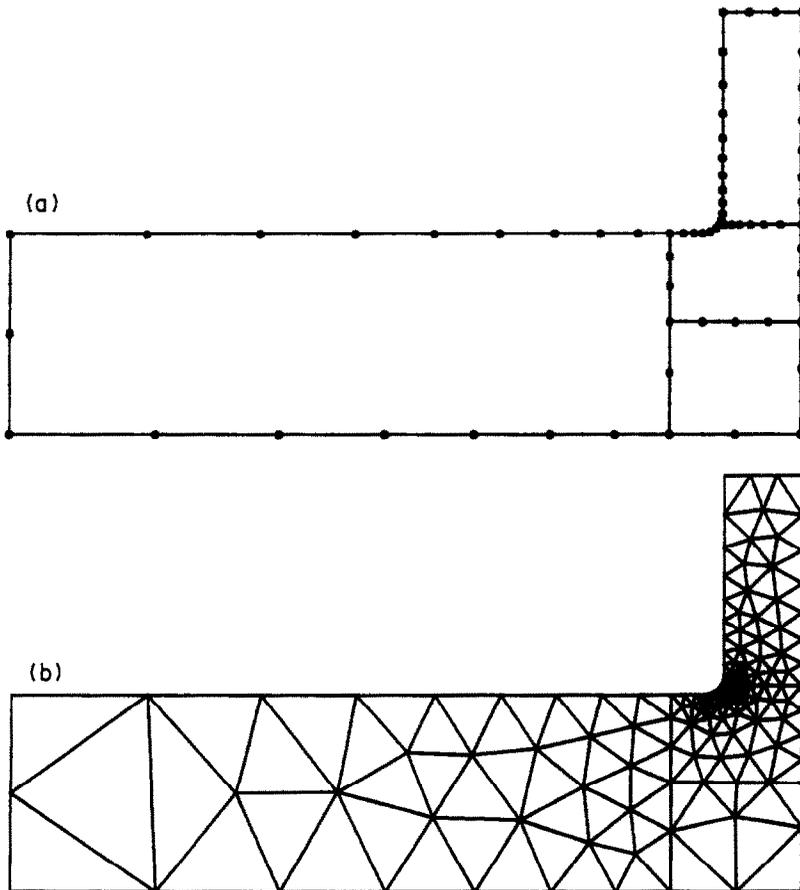


Fig. 36. (a) Key nodes distribution for a fine mesh construction. (b) A fine starting mesh with 221 elements; total strain energy: 0.298060×10^{-2} lbf-in.

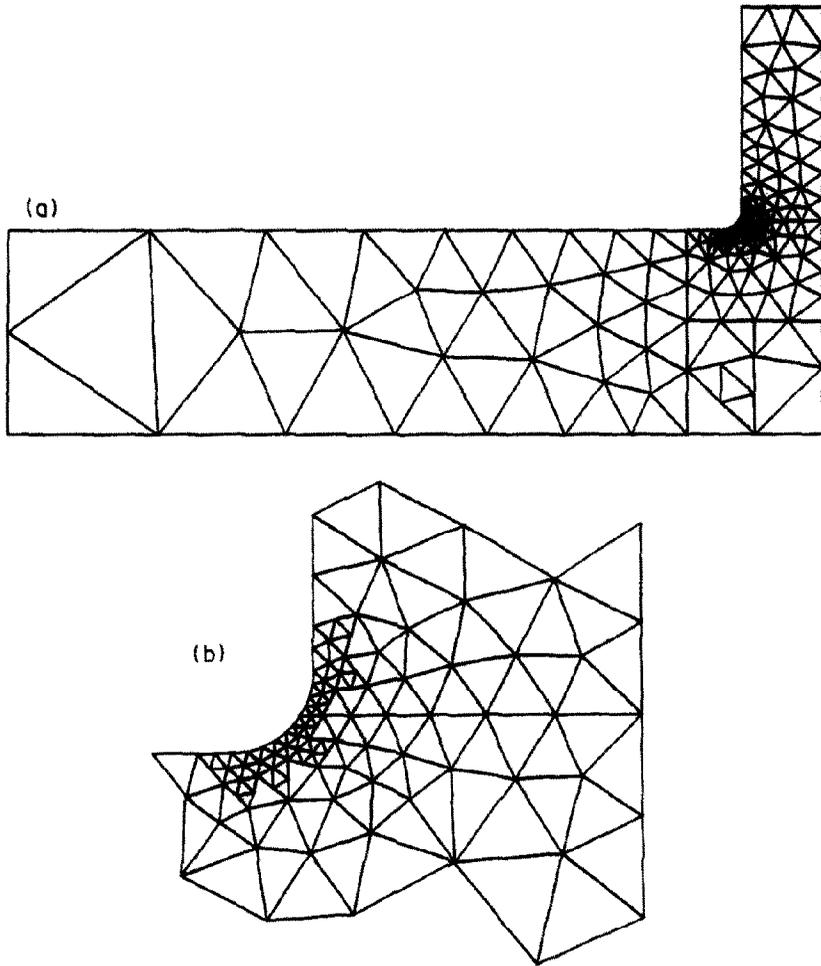


Fig. 37. (a) Final 281 element mesh; total strain energy: 0.298093×10^{-2} lbf-in., $\epsilon_{total} = 1.23$, $\epsilon_{tolerance} = 0.02$ (221 el \rightarrow 281 el). (b) Detail of 281 element mesh.

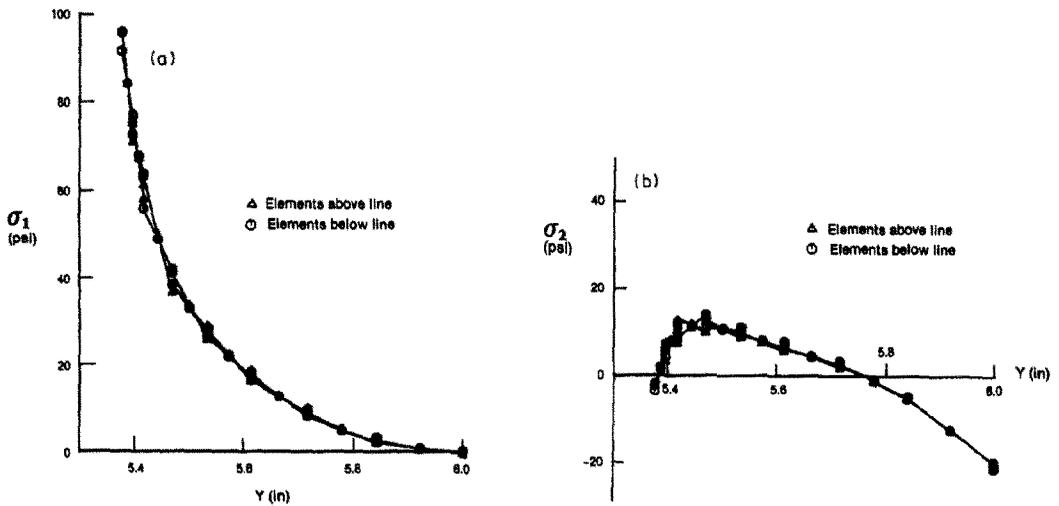


Fig. 38. (a) Maximum in-plane principal stress for 281 element mesh. (b) Minimum in-plane principal stress for 281 element mesh.

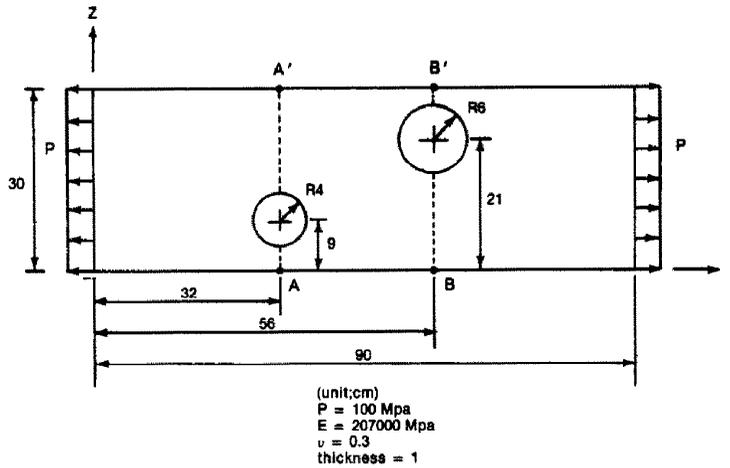


Fig. 39. Plate with two holes.

step of refinement, if available experience is used in constructing an efficient initial mesh.

Analysis of a plate with two holes in tension

Next, we consider the plane stress analysis of a plate with two holes, in tension, as shown in Fig. 39. The goal of the analysis is to determine the stress distributions on the lines $A-A'$ and $B-B'$.

The analysis domain was subdivided into three subdomains, see Fig. 40(a), and a coarse initial mesh with 198 elements was constructed automatically, see Fig. 40(b). The τ_{yy} stress distributions along the lines $A-A'$ and $B-B'$ are shown in Fig. 41. The results show that large discontinuities occur at the region of stress concentration; hence, the stress predictions are not satisfactory.

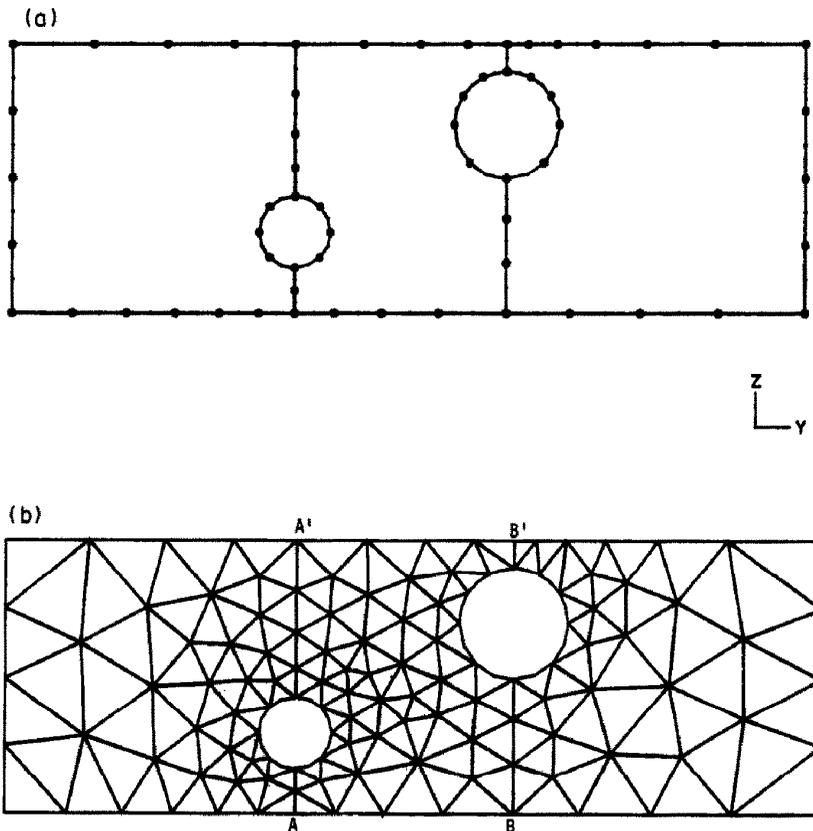


Fig. 40. (a) Key nodes distribution for a coarse mesh generation. (b) A coarse starting mesh with 198 elements; total strain energy: 0.814892×10^2 N-m, $\epsilon_{total} = 5.35$.

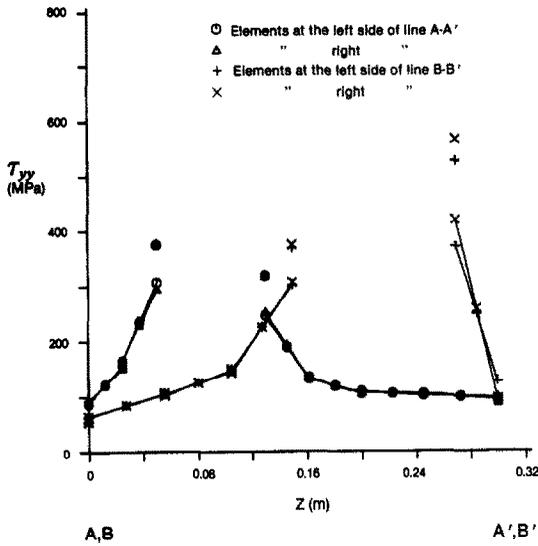


Fig. 41. τ_{yy} distribution along the lines $A-A'$ and $B-B'$.

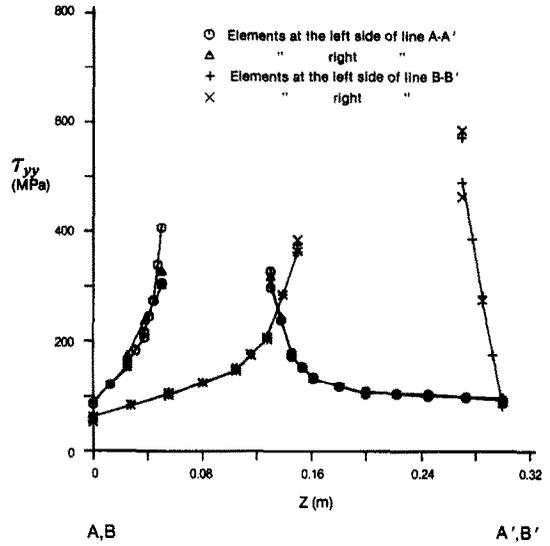


Fig. 43. τ_{yy} distribution along the lines $A-A'$ and $B-B'$.

In order to obtain better results, the adaptive refinement was performed with an indicator tolerance $\epsilon_{tolerance} = 0.02$ and a 381 element model was obtained in 3 steps of refinement (198 elements \rightarrow 303 elements \rightarrow 375 elements \rightarrow 381 elements) as shown in Fig. 42. The stress output results for the 381 element mesh are shown in Fig. 43, which shows that the accuracy of solution has been improved compared with the results of the 198 element mesh, but there are still some large stress jumps at the holes.

To improve the analysis results, the 429 element mesh shown in Fig. 44 was constructed as the initial mesh. By using the same error tolerance, $\epsilon_{tolerance} = 0.02$, a 495 element mesh was obtained in one step of refinement (429 elements \rightarrow 495 elements), see Fig. 45. The stress output results in Fig. 46 show now good results with only small stress discontinuities, and in some regions the mesh is perhaps now even over-refined.

Analysis of a cylinder attached to a block

As a general three-dimensional problem, we consider a cylinder in bending attached to a block (see Fig. 47). For the analysis of this problem, a coarse and a fine mesh were constructed, and to obtain well-conditioned elements, the structure was subdivided for each case into convex subobjects.

For the coarse mesh construction, the structure was subdivided into three convex subobjects as shown in Fig. 48, while for the fine mesh construction it was subdivided into six subobjects as shown in Fig. 49. Using our volume triangulation algorithm, the meshes of the 514 elements and 1568 elements shown in Figs 50 and 51 were obtained. The steps in this volume triangulation process were as described in Sec. 4: first the faces of the objects were triangulated as prescribed by key nodes and then the volume was filled with tetrahedra. For the stress solution, it is

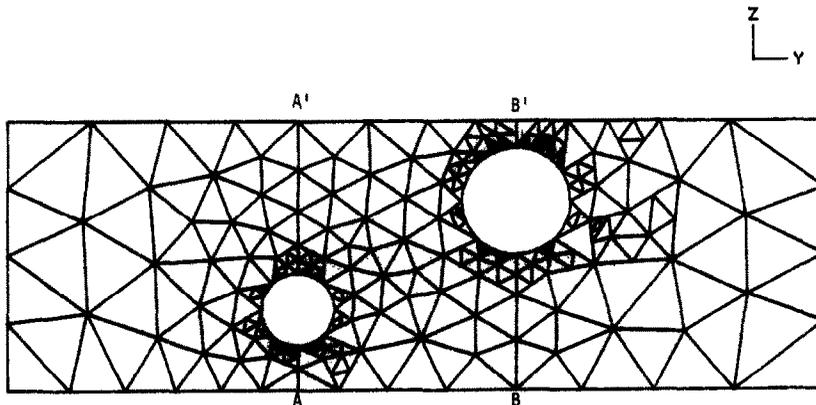


Fig. 42. Final 381 element mesh after 3 steps of refinement; total strain energy: 0.817691×10^2 N-m, $\epsilon_{total} = 1.35$, $\epsilon_{tolerance} = 0.02$ (198 el \rightarrow 303 el \rightarrow 375 el \rightarrow 381 el).

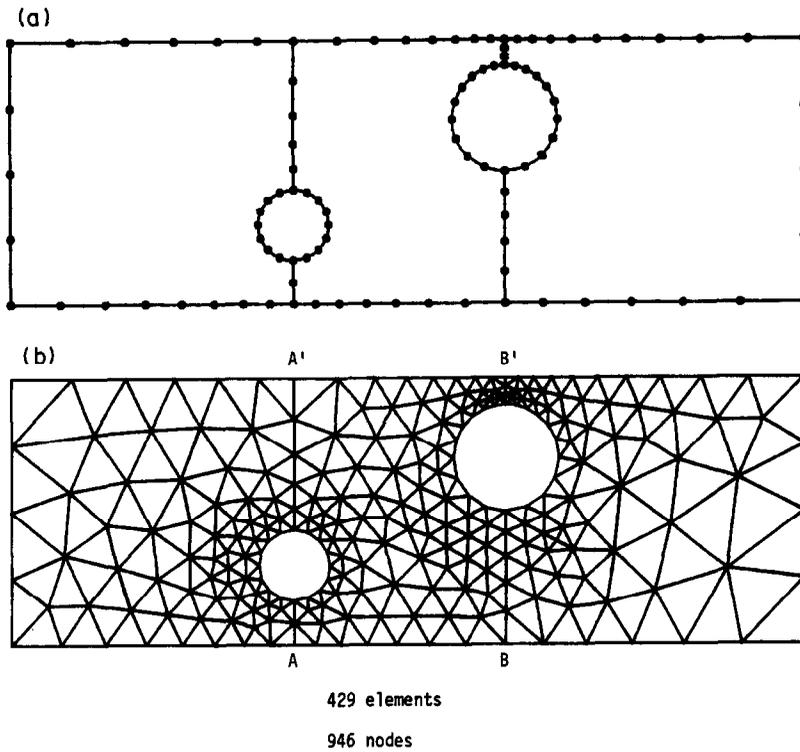


Fig. 44. (a) Key nodes distribution for a fine mesh construction. (b) Results of triangulation.

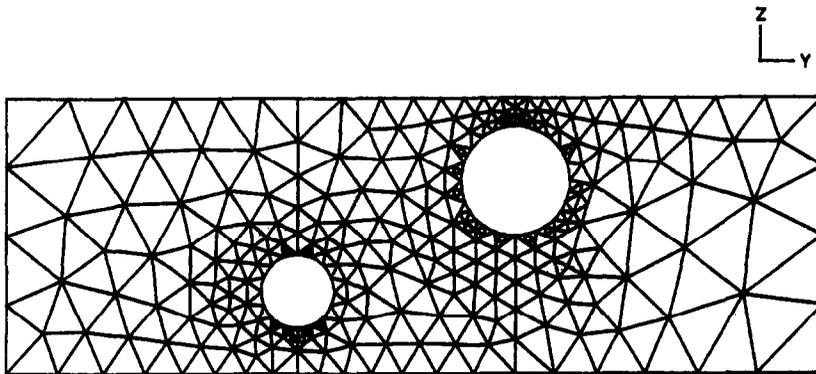


Fig. 45. Final 495 element mesh after one step of refinement; total energy: 0.818737×10^2 N-m, $\epsilon_{\text{total}} = 1.14$, $\epsilon_{\text{tolerance}} = 0.02$ (429 el \rightarrow 495 el).

Table 2. Results of the cylinder bending analysis

Total No. of elements (No. of elements in cylinder)	Total strain energy (lbf-in)	Displacement δ under load (in.)	Distance l from load (in.)	Average of normal stress τ_{yy} at distance l from load (% deviation (beam theory) (psi)
514 elements (194 elements)	23.3	-0.0233	20	429.0 5.3% (407.5)
1568 elements (322 elements)	24.6	-0.0246	22.5	480.4 4.8% (458.4)
Beam bending theory (cylinder only)		-0.0175		

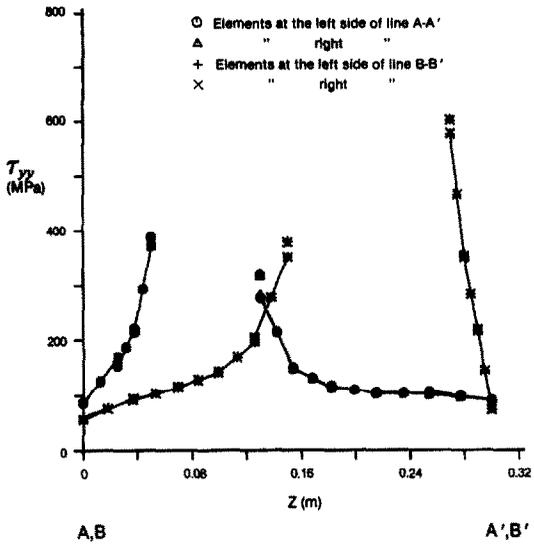


Fig. 46. τ_{yy} distribution along the lines A-A' and B-B'.

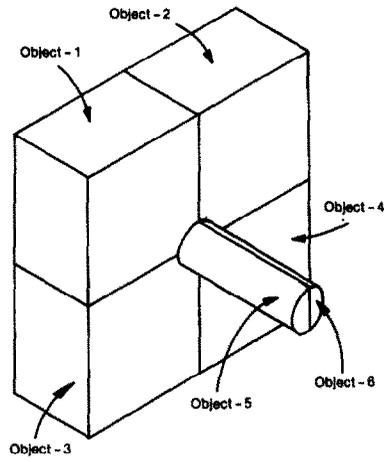


Fig. 49. Subdivisions for a fine mesh construction.

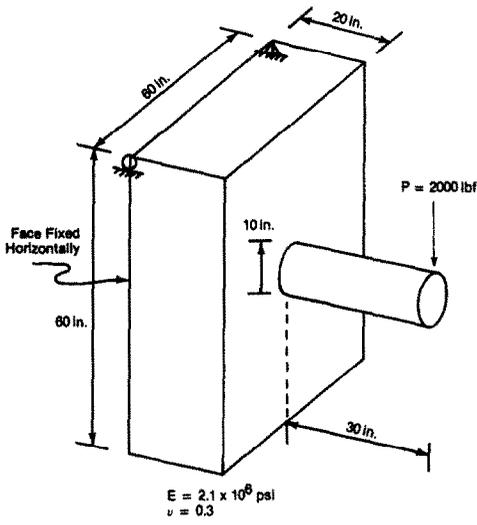


Fig. 47. Cylinder attached to a block.

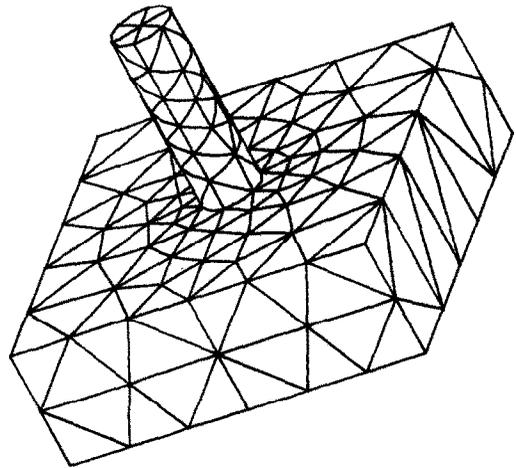


Fig. 50. 514 element mesh with curved-sided elements.

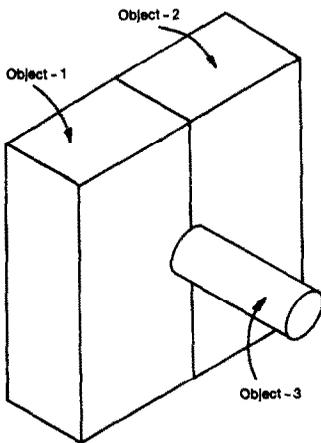


Fig. 48. Subdivisions for a coarse mesh construction.

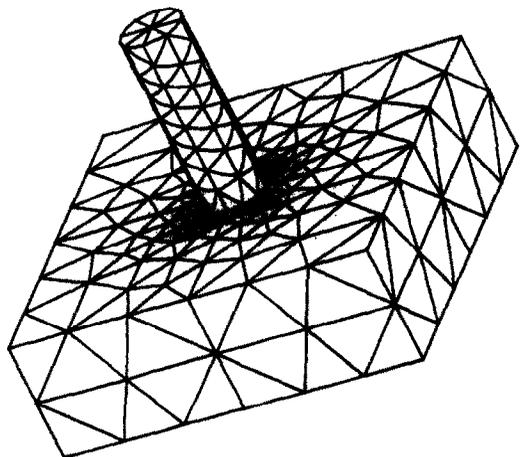


Fig. 51. 1568 element mesh with curved-sided elements.

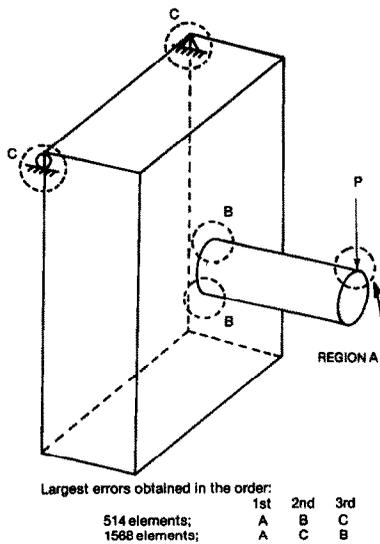


Fig. 52. Error distribution in the bending analysis.

important that the curved surface of the cylinder is represented accurately; hence, the element faces on the cylindrical surface are placed curved with the isoparametric parabolic interpolation assumption.

In this particular mesh generation each subobject was individually volume triangulated, but in practice symmetry conditions should of course be used to minimize the effort for mesh construction. Table 2 lists the results obtained in the analyses. The beam theory solution gives here only a very approximate comparison, of course.

The calculation of the error indicators (which assumed that all elements are flat-faced) gave the results indicated in Fig. 52. The error indicators were large where expected because of the stress singularities, but in the fine mesh solution the indicators were larger in regions C than in regions B. Note that the stress peaks in regions C are 'artificial' because of the supports placed in these regions. Some further results on this problem solution are given in [5].

7. CONCLUDING REMARKS

The objective of this paper was to present some research results on the automatic mesh construction and mesh refinement in finite element analysis. The following conclusions can be reached.

(i) In 2D analysis, the automatic mesh construction and refinement procedures presented in this paper are quite robust and efficient, although various improvements can be envisaged (see below).

(ii) In 3D analysis, the efficient automatic construction and refinement of finite element meshes is much more difficult due to the topological requirements. In this paper we only dealt with the initial mesh construction and the development of an error indicator. Further research and development is neces-

sary to establish an efficient scheme for the mesh refinement, but the 2D developments presented herein provide a basis for such work.

A major shortcoming of the refinement process used in this paper is that the final mesh is reached in steps from the original mesh by simple subdivision of the elements. It may be more efficient to construct automatically a completely new mesh (i.e. with a new topology) depending on the value of the error indicator. Such remeshing could also be used only in certain regions. Furthermore, the mesh topology could be kept but the nodal points be shifted or the order of the element displacement expansions be increased. Hence, there are various possibilities to reach improved automatic meshing procedures.

In addition, of course, there is the desire to have actual error estimates for the calculated solution. In this work we used an error indicator (with the over-relaxation factor $1/h_m$) to force the mesh refinement. The accuracy of the solution can then be judged by stress graphs or isoband plots of stresses [20]. This visual identification of whether the stress solution is acceptable is quite practical and general but exact error bounds would of course be very useful.

In concept, the above further developments may appear simple but the major difficulty lies in reaching a method that shows high efficiency and is a reliable technique—that 'always' works—for general applications.

REFERENCES

1. M. S. Shephard, *Finite Element Grid Optimization—A Review*. ASME Special Publications PVP, Vol. 38, pp. 1–14 (1979).
2. M. S. Shephard, Finite element modeling within an integrated geometric modeling environment: part I—mesh generation. *J. Engng Comput.* 1, 61–71 (1985).
3. I. Babuska, O. C. Zienkiewicz, J. P. de S. R. Gago and E. R. de A. Oliveira, *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*. John Wiley, New York (1986).
4. D. W. Kelly, J. P. de S. R. Gago, O. C. Zienkiewicz and I. Babuska, *A posteriori* error analysis and adaptive processes in the finite element method: part I—error analysis and part II—adaptive mesh refinement. *Int. J. Numer. Meth. Engng* 19, 1593–1656 (1983).
5. S. W. Chae, On the automatic generation of near-optimal meshes for three-dimensional linear elastic finite element analysis. Ph.D. thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology (1988).
6. W. C. Thacker, A brief review of techniques for generating irregular computational grids. *Int. J. Numer. Meth. Engng* 15(9), 1335–1341 (1980).
7. E. A. Sadek, A scheme for the automatic generation of triangular finite elements. *Int. J. Numer. Meth. Engng* 15(12), 1813–1822 (1980).
8. M. Yokoyama, Automated computer simulation of two-dimensional elastostatic problems by the finite element method. *Int. J. Numer. Meth. Engng* 21, 2273–2289 (1985).
9. F. T. Tracy, Graphics pre- and post-processor for two-dimensional finite element programs. *Comput. Graph. (Proc. Siggraph '77)* 11(2), 8–12 (1977).

10. V. Ph. Nguyen, Automatic mesh generation with tetrahedron elements. *Int. J. Numer. Meth. Engng* **18**, 273–280 (1982).
11. J. C. Cavendish, D. A. Field and W. H. Frey, An approach to automatic three-dimensional finite element mesh generator. *Int. J. Numer. Meth. Engng* **21**, 329–347 (1985).
12. B. Wördenweber, Finite element mesh generation. *Comput.-Aided Des.* **16**(5), 285–291 (1984).
13. M. A. Yerry and M. S. Shephard, Automatic three-dimensional mesh generation by the modified-octree technique. *Int. J. Numer. Meth. Engng* **20**, 1965–1990 (1984).
14. M. S. Shephard, M. A. Yerry and P. L. Baehmann, Automatic mesh generation allowing for efficient *a priori* and *a posteriori* mesh refinement. *Comput. Meth. appl. Mech. Engng* **55**, 161–180 (1986).
15. M. E. Mortenson, *Geometric Modeling*. John Wiley, New York (1985).
16. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*. Prentice-Hall, Englewood Cliffs, NJ (1982).
17. I. Babuska and D. Yu, Asymptotically exact *a posteriori* error estimator for biquadratic elements. Technical Report BN-1050, Institute for Physical Science and Technology, University of Maryland (1986).
18. I. Babuska and W. C. Rheinboldt, Adaptive approaches and reliability estimations in finite element analysis. *Comput. Meth. appl. Mech. Engng* **17/18**, 519–540 (1979).
19. C. G. Floyd, The determination of stresses using a combined theoretical and experimental analysis approach. *2nd International Conference on Computational Methods and Experimental Measurements* (Edited by C. A. Brebbia) (1984).
20. T. Sussman and K. J. Bathe, Studies of finite element procedures—stress band plots and the evaluation of finite element meshes. *Engng Comput.* **3**, 178–191 (1986).