Adaptive mesh refinement in finite element analysis

S Rajasekaran & V Remeshan
P.S.G. College of Technology, Coimbatore 641 004, India

Received 27 April 1998; accepted 11 November 1998

Among the acceptable numerical methods, Finite Element Analysis stands as the most acceptable one for problems characterised by partial differential equations. However, inaccuracy in Finite Element Analysis is unavoidable since a continuum with infinite degrees of freedom is modelled into finite degrees of freedom. In addition to the mesh generation task being tedious and error prone, the accuracy and cost of the analysis depend directly on size, shape and number of elements in the mesh. The procedure of refining the mesh automatically based on the error estimate and distribution of the error is known as “adaptive” mesh refinement. A simplified method called “Divide and Conquer” rule based on “Fuzzy Logic” is used to refine the mesh by using h, p and hp versions. Automatic mesh generator developed in this paper based on Fuzzy logic is able to develop well shaped elements. The program for automatic mesh generation and subsequent mesh refinement is developed in “C” language and the analysis is carried out using “ANSYS” package. Automatic mesh generation is applied to problems such as dam, square plate with a hole, thick spherical pressure vessel and a corbel and error less than 5% is achieved in most of the cases.

The initial necessity for automatic mesh generation was to reduce time and money spent for creating a mesh, which could amount to half of the total time required for the Finite Element Analysis. Adaptation including both mesh optimisation and mesh refinement, was shown to be very important in case of problems for localised stress gradients and point loads.

In early 1970s, Zienkiewicz and Phillips introduced the technique of iso-parametric mapping and most of numerical mesh generation techniques still employ mapping techniques. An attempt has been made to automate the mapping technique by Blacker and Stephenson through the ultimate decomposition of the complex domain into mappable sub regions. Valliappan et al. introduced fuzzy logic for automatic mapping technique by dividing the region into sub regions by selecting optimum splitting lines. In this method “Divide and Conquer” rule by using Fuzzy Logic through expert system is used for automatic mesh generation. By this method, the whole domain is divided into a number of sub regions and quadrilaterals or triangular shapes and the mesh is generated separately in each of them satisfying prescribed conditions and finally sub regions are attached together.

Error Estimation

The term “error” is generally applied to the difference between results of computational and the actual phenomenon observed experimentally. The Finite Element Analysis yields the following errors: (a) errors due to spatial discretization, (b) round off errors, (c) approximation involved in mathematical model. With the use of today’s computers, the round off errors can be minimised assuming that the solution of the mathematical model is exact. Hence, the error due to discretization is considered in this paper. Usually, posterior approach is used to estimate the error. Finite Element Analysis is carried out for raw distribution and refined mesh. Based on the error between them, further refinement of mesh is carried out.

The error approximation technique used in ANSYS is similar to that one proposed by Zienkiewicz and Zhu. The estimation of error in stress can be done by using Eq. (1):

\[ \Delta \sigma = \sigma_n - \sigma_e \]

where \( \Delta \sigma \) is the error and \( \sigma_n \) is the nodal stress calculated by simple averaging of element at a node and \( \sigma_e \) an averaged stress at a node. Then error norm \( e \), in the \( i \)th element is given by

\[ e_i = \{ \Delta \sigma \}^T \{ C \} \{ \Delta \sigma \} \]

where \( \{ C \} \) is the constitutive matrix.

The error norm \( e \) over the entire model can be assumed to be equal to...
\[ e = \sum_{i} u_i \]  
\[ E = \frac{e}{(U + e)}^{1/2} \times 100 \]  

where \( n \) is the number of the elements in the model or part of the model.

The error energy \( E \) can be normalised against strain energy \( U \). This is done in POST1 of ANSYS as

Finally, the mesh size \( h \) is refined in combination with increase in the degrees of the polynomial in a proper manner. \( hp \)-version converges exponentially even in the presence of singularities in the exact solution. The error estimate for \( hp \)-version is given by

\[ e = e^{-\beta h} \]

where \( \beta \) is a positive constant which depends on the smoothness of the exact solution. If the mesh is not designed properly, then the \( p \)-version does not perform so well for problems with singularities although its performance is always better than the \( h \)-version with uniform mesh with respect to the number of degrees of freedom.

Refinement Methods

Mesh refinement techniques can be categorised as:

- \( h \)-version in which refinement is done by subdivision of element, where \( h \) stands for typical geometric parameter like element length or diameter. If the refinement is being done throughout the domain then it is uniform \( h \)-refinement; else it is carried out locally, and called adaptive \( h \)-refinement.

- \( p \)-refinement in which refinement is done by increasing the interpolating function order. If \( p \)-version is applied throughout the domain it is called \( p \)-version, else it is called adaptive \( p \)-version.

- \( hp \)-version, which combines both of the schemes described above, i.e., simultaneous reduction in the size of the element and increase in the polynomial order. In the \( h \)-version finite element convergence study, it has been found that if the sequence of meshes is designed in such a way that the error is approximately equally distributed on each element, the meshes are called optimal meshes and the influence of singularities is eliminated giving:

\[ e \leq c N^{-p/2} \]

where

- \( e \) is positive constant depending upon the problem
- \( e \) is error in energy norm
- \( N \) is the number of degrees of freedom
- \( p \) is the polynomial degree of used shape functions

\( p \)-version on the other hand converges exponentially if the exact solution is smooth and converges always faster than the fixed polynomial degree. The error estimate is then given by:

\[ e = e^{-\beta h} \]

where \( \beta \) is a positive constant which depends on the smoothness of the exact solution. If the mesh is not designed properly, then the \( p \)-version does not perform so well for problems with singularities although its performance is always better than the \( h \)-version with uniform mesh with respect to the number of degrees of freedom.

Types of Elements Used

Adaptive mesh refinement of the finite element is carried out with or without the help of automatic mesh generation. Finite Element Model requires that the domain of interest be divided into a mesh of discrete elements, and the accuracy and expense of the calculations are strongly affected by goodness of underlying mesh. An ideal mesh is characterised by small element when stress gradients are high and larger elements when stress gradients are small, elements being regular, or undistorted in shape. When quadrilateral elements are used, these characteristics can sometimes be difficult to achieve. In this paper four and eight node iso-parametric quadrilaterals are used. In case of eight node quadrilateral elements, the triangular elements can be obtained by merging 4, 7 and 3 on one edge into one node.

Mesh generation algorithm

At the beginning of the mesh generation process, the domain boundary is divided into a number of divisions such that each division has a size less than or equal to the specified element size. The re-entrant corner must be detected as given below.

A corner is said to be re-entrant corner if the included angle is \( > 180^\circ \). If any re-entrant corner is present, this polygon is called concave polygon. This
corner is detected initially and the subdivision of the concave polygon is done along the splitting line from the re-entrant corner to make it convex. The aim is to create convex polygon (all included angles $\leq 180^\circ$) so that mesh generation in each sub domain can easily be done. Consider two lines meeting at a corner. The preceding line is denoted by "p" and the following line as "f". There are twenty possibilities existing with respect to various conditions.

After finding the re-entrant corner, the next step is to find the optimum splitting line from re-entrant corner for dividing the concave polygon into convex polygons. The selection of optimum splitting line is carried out by using Fuzzy Logic. The basic idea behind the selection of splitting line is that it should make all angles as close to $90^\circ$ as possible.

**Fuzzy inference algorithm**

Fuzzy algorithms are divided into four basic categories: Fuzzy definitional algorithm, Fuzzy generational algorithms, Fuzzy relational and behavioural algorithms and Fuzzy decisional algorithms. Fuzzy decisional algorithm is adopted here to decide the optimum splitting line.

If $\alpha, \alpha', \beta, \beta'$ are the four angles of the splitting line, then the optimum splitting line is calculated by maximising the following function.

Maximise:

$$f(\alpha, \alpha', \beta, \beta') = f(\alpha) + f(\alpha') + f(\beta) + f(\beta')$$

$$= R(\alpha) + R(\alpha') + R(\beta) + R(\beta') \quad (9)$$

If "r" is a scalar variable which takes the values of four cut angles $\alpha, \alpha', \beta, \beta'$ in the interval $[0, 2\pi]$, then the criteria for the cut angles as close as possible to $90^\circ$ can be expressed by the following IF-THEN rules. (see Fig. 1a and 1b).

**Rule 1**

If $r$ is very close to $90^\circ$ then $R(r)$ is very large

**Rule 2**

If $r$ is close to $90^\circ$ then $R(r)$ is large

**Rule 3**

If $r$ is medium close to $90^\circ$ then $R(r)$ is medium

**Rule 4**

If $r$ is far from $90^\circ$ then $R(r)$ is small

**Rule 5**

If $r$ is very far from $90^\circ$ then $R(r)$ is very small

**Rule 6**

If $r$ is very very far from $90^\circ$ then $R(r)$ is very very small

**Rule 7**

If $r$ is too far from $90^\circ$ then $R(r)$ is too small

**Computation of fuzzy control output or de-fuzzification**

The possible splitting lines are the imaginary lines joining the re-entrant corner and the division points along the domain boundary. Let PQ be one of the splitting lines as shown in Fig.2. Now to calculate the value of $(\beta_1, \beta_2)$, the angles at "Q" consider the adjoining division of d1, d2 along the boundary and solve the two triangles thus formed by applying cosine rule to get $(\beta_1, \beta_2)$. Similarly following the same procedure one gets the value of angles $(\alpha, \alpha')$ by considering two adjacent division points at the re-entrant corner. Any splitting line which is having any of its angles as zeroes, more than zero and less than $90^\circ$. There are twenty possibilities existing with respect to various conditions.

**Diagram**

![Membership function for angles](image1a.png)

![Membership function for rating variable](image1b.png)
or equal to 180°, will be a candidate for competition for selection of optimum splitting line. After getting the four angles, we are normalising it by subtracting it from 180°, if the angle obtained is >90°. In general, the input (angles) for getting Fuzzy rating value should be ≤ 90°.

Procedure
Consider the angle α_i which is between 70° and 90°. Now, produce a vertical line to strike two Fuzzy sets very close and closes as shown in Fig.3. From these intersecting points, produce a horizontal line to meet the corresponding Fuzzy sets of membership grade and draw triangles with its base equal to the base of Fuzzy sets in membership function and height equal to the height of the horizontal line with vertex lying at the midpoint of the Fuzzy set. Rating value is calculated by taking moment about origin to find rating value. Similarly the rating of other three angles is calculated and added to get the rating value of the splitting line. The procedure is repeated for all other possible splitting lines and that having maximum value is selected. If there are more than one re-entrant corners then the procedure is repeated for the concave polygons till sub polygons are convex in nature as shown in Fig. 4.

Division of convex polygons into 4 sided quadrilaterals
The generated convex polygon (from previous step) may contain more than 4 sides. Our aim is to get convex polygons of 4 sides or 3 sides so that mesh generation can be done in these domains easily. The Fuzzy logic based optimum splitting line is again calculated to divide convex polygons of more than 4 sides and the procedure is continued till we get all the convex polygons of sides three or four. A quadrilateral is accepted for generation if the ratio between opposite sides is between 0.5 and 2.0 as:
\[ 0.5 \leq AB/CD \leq 2.0 \] (10)

Smoothening along the splitting line
Mesh generation is carried out individually in each convex sub polygon obtained by following procedure mentioned in previous paragraph. But continuity of the elements along the boundary should be maintained. In order to achieve this, smoothening of the elements along the splitting line is carried out.

Algorithm
Consider a splitting line “AB” as shown in Fig. 5a. The smoothening technique works as follows.
1. Calculate the length of “AB” and divide the length by size of elements specified in the input such that length of each division is equal to or less than that specified size, and these points are called pivot points.
2. Consider first pivot point “p” and let the node N1, N2 and N3 lying within the circle of influence of radius equal to half of (AP). The smoothening is carried out by connecting all nodes to the mid point of the extreme points N1 and N3, i.e. co-ordinates of point N2 after smoothening are \((X(N1)+X(N3))/2\) and \((Y(N1)+Y(N3))/2\).
This method avoids the large shifting of any node coming within the circle of influence if we are connective to pivot points. (see Fig. 5b). Fig. 6 shows the splitting line Q1 and Q2 consisting of uncommon nodes and the smoothened mesh.

The extra nodes which are formed during smoothening operation is removed by eliminating the larger values of nodes and renumbering the following nodes. Nodes having equal “x” and “y” co-ordinates are selected and then node renumbering is done by following the continuation of the previous highest corrected node.

Adaptive mesh refinement

The data obtained from the automatic mesh generation is given for ANSYS solution and the result of this solution is analysed to find out whether refinement is required or not. The output from the ANSYS solution is obtained in the form of error in strain energy in each element (ENRM) and actual strain energy of the element (SENE). The total error percentage is given by the equation

\[
\text{ERROR} = 100 \times \frac{\sum \text{ENRM} / \sum \text{SENE}}{1} 
\]

and the element to be refined is found out by calculating

\[
e = k \sqrt{\frac{\sum \text{ENRM} / \sum \text{SENE}}{M}} 
\]

where \( M \) is the total number of elements, \( k \) is a constant (acceptable percentage of error)

ENRM = error in strain energy in element

SENE = actual strain energy in element which gives the permissible percentage or error in element. If the error in strain energy in an element exceeds this value then that element is to be refined.

The element thus stored is then subjected to \( h/p/hp \) type of refinement. In this paper, the uniform \( h \) version is adopted till one gets an error percentage around 15%, after which the uniform \( p \)-refinement is applied after the adaptive \( p \), adaptive \( h \) or adaptive \( hp \) type refinements are applied.

Uniform \( h \)-refinement

The coarse mesh generated initially is subjected to uniform \( h \)-refinement and the procedure is continued till one gets the error percentage around 15%.

The insertion of nodes in each element to be refined is done in counter clockwise direction. The co-ordinates of these nodes are calculated by averaging the co-ordinates of the vertices in case of triangular element.

Centre node location is calculated by averaging the mid-point co-ordinates of the line joining mid nodes in case of quadrilateral and by averaging the co-ordinates of the vertices in case of triangular element.

During the insertion of the new node, node numbering is done continuously and there exist common nodes. The existence of the common node is found out by searching the nodes having equal \( x \) and \( y \) co-ordinates. These nodes are renumbered in continuation with the highest corrected node.

Uniform \( p \)-refinement

In case of uniform \( p \)-refinement, the middle side nodes are inserted in each element by following the procedure described as before and finally node renumbering is done.

Adaptive \( h \)-refinement

Result the previous analysis is verified and if the error percentage is greater, adaptive \( h \)-refinement is followed. Refinement procedure is same as that mentioned before.

Adaptive \( p \)-refinement

If the error percentage is still above acceptable limit after adaptive \( h \)-refinement then adaptive \( p \)-refinement is followed.
Adaptive $hp$-refinement

The combination of $h$ and $p$ type refinement can be carried out if found necessary and the program works as follows. First it searches for the element surrounding the elements actually to be refined. Adaptive $h$-refinement of these elements is carried out and then adaptive $h$-refinement of the actual elements to be refined is carried out. Subsequently, the new elements generated from the second step are subjected to uniform $p$-version. Finally, nodes renumbering is carried out.

Numerical Examples

Various problems have been solved and the results are found to be well below the acceptable percentage. As an engineering measure, acceptable percentage of error is taken as 10%. In each problem, only two types of refinements (uniform $h$ and uniform $p$) are carried out and with this itself acceptable results are obtained.

Example 1—L-shaped domain

This is a plane stress problem. Geometry and loading conditions are given in Fig. 7a. Initial coarse mesh consists of 4 noded 48 quadrilateral elements having 120° of freedom in total. Then, uniform $h$-refinement is performed to get 192 elements having total D.O.F equal to 428. Further, uniform $p$-refinement is done to increase the D.O.F to 1240 and with this analysis an error percentage of 8.928% is obtained. The results are given in Table 1 and adaptive $h$-refinement is shown in Fig. 7b.

Example 2—Dam

This is a plane strain problem. Geometry and loading conditions are shown in Fig. 8a and the initial coarse mesh consists of 112 elements with small number of triangular elements and mesh totally having 228 D.O.F. Then adaptive $h$-refinement is carried out and the error percentage of 9.765% is obtained. Then adaptive $p$-refinement is carried out and error percentage of 3.365% is obtained with 2654 D.O.F. The results are given in Table 1 and the uniform $h$-refinement is shown in Fig. 8b.

Example 3—Square plate with circular hole

Geometry and loading conditions are shown in Fig. 9a and initial coarse coarse mesh consists of 92 elements and 212 nodes is subjected to uniform $h$-refinement and then uniform $p$-refinement is applied. With this, an error of 1.94% is obtained. The results are given in Table 1 and the adaptive $h$-refinement is shown in Fig. 9b.

Example 4—Thick spherical pressure vessel

Geometry and loading conditions are as shown in Fig. 10a. Initial coarse mesh consists of 72 elements and 178° of freedom and after uniform $h$-refinement and subsequent uniform $p$-refinement, error percentage of 1.788% has been reached as shown in
Table 1—Error in adaptive mesh refinement

<table>
<thead>
<tr>
<th>Problem</th>
<th>refinement</th>
<th>Elements</th>
<th>nodes</th>
<th>D.O.F</th>
<th>SENE</th>
<th>ENRM</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>L-domain</td>
<td>cm</td>
<td>48</td>
<td>65</td>
<td>120</td>
<td>0.364574</td>
<td>0.025248</td>
<td>24.45</td>
</tr>
<tr>
<td></td>
<td>uh</td>
<td>192</td>
<td>225</td>
<td>432</td>
<td>0.374306</td>
<td>0.009229</td>
<td>15.57</td>
</tr>
<tr>
<td></td>
<td>up</td>
<td>192</td>
<td>641</td>
<td>1248</td>
<td>0.378571</td>
<td>0.003035</td>
<td>8.928</td>
</tr>
<tr>
<td>Corbel</td>
<td>cm</td>
<td>112</td>
<td>128</td>
<td>224</td>
<td>67.7849</td>
<td>1.58185</td>
<td>15.32</td>
</tr>
<tr>
<td></td>
<td>uh</td>
<td>431</td>
<td>437</td>
<td>812</td>
<td>66.266</td>
<td>0.638026</td>
<td>9.765</td>
</tr>
<tr>
<td></td>
<td>up</td>
<td>431</td>
<td>1388</td>
<td>2654</td>
<td>66.66</td>
<td>0.07555</td>
<td>3.365</td>
</tr>
<tr>
<td>Square plate</td>
<td>cm</td>
<td>92</td>
<td>113</td>
<td>212</td>
<td>0.115784</td>
<td>0.002702</td>
<td>15.11</td>
</tr>
<tr>
<td></td>
<td>uh</td>
<td>367</td>
<td>409</td>
<td>792</td>
<td>0.116968</td>
<td>0.000998</td>
<td>9.14</td>
</tr>
<tr>
<td></td>
<td>up</td>
<td>367</td>
<td>1184</td>
<td>2318</td>
<td>0.117457</td>
<td>0.000042</td>
<td>1.94</td>
</tr>
<tr>
<td>Thick vessel</td>
<td>cm</td>
<td>72</td>
<td>93</td>
<td>178</td>
<td>14.811</td>
<td>0.24785</td>
<td>12.83</td>
</tr>
<tr>
<td></td>
<td>uh</td>
<td>284</td>
<td>329</td>
<td>644</td>
<td>14.91</td>
<td>0.085154</td>
<td>7.534</td>
</tr>
<tr>
<td></td>
<td>up</td>
<td>284</td>
<td>941</td>
<td>1854</td>
<td>14.961</td>
<td>0.004787</td>
<td>1.789</td>
</tr>
<tr>
<td>Corbel</td>
<td>cm</td>
<td>133</td>
<td>162</td>
<td>312</td>
<td>0.02987</td>
<td>0.001986</td>
<td>24.85</td>
</tr>
<tr>
<td></td>
<td>uh</td>
<td>524</td>
<td>589</td>
<td>1138</td>
<td>0.03083</td>
<td>0.000774</td>
<td>15.65</td>
</tr>
<tr>
<td></td>
<td>up</td>
<td>524</td>
<td>1701</td>
<td>3250</td>
<td>0.031246</td>
<td>0.000075</td>
<td>4.89</td>
</tr>
</tbody>
</table>

cm—coarse mesh; uh—uniform h; up—uniform p

Table 1 and the meshes at different stages are shown in Fig. 10b.

Example 5—Corbel

Geometry and loading conditions are shown in Fig. 11a and the initial coarse mesh consists of 133 elements having 304 D.O.F. An error percentage of 4.89% is obtained after the uniform p-refinement, which is done in continuation with uniform h-refinement (see Table 1) with 3250 D.O.F. The meshes generated using adaptive h-refinement are shown in Fig. 11b.

Conclusions

Automatic mesh generator developed based on the Fuzzy logic is able to develop well shaped elements. For various examples solved, an error percentage of less than 5% is achieved except in case of L-shaped domain where singularity affects further refinement. The developed automatic mesh generator has proved
Fig. 9a — Square plate with a circular hole

Fig. 9b — Adaptive h-refinement for plate with a hole

Fig. 10a — Thick spherical pressure vessel

Fig. 10b — Uniform h-refinement for vessel
suitable for multi-connected domain provided there are no re-entrant corner at the outer boundary of the given domain. Refinement strategy followed proves to be very effective in achieving the error percentage less than the acceptable limit.

Acknowledgements

The authors thank the management and the Principal, Dr. P. Radhakrishnan, P.S.G College of Technology for providing necessary facilities to carry out the work.

References

3 Valliappan S, Pham, T D, Murthy V & Franklin J, Micro
4 Valliappan S & Pham, T D, Micro Comput Civil Eng, 8 (1993) 75-82.