4.1 Introduction

The Finite Element Method offers only an approximation to the exact solution of a mathematically posed problem. The error in approximate, numerical solution arise due to three primary causes:

a) The first and most important is the discretisation error, which is introduced due to incomplete satisfaction of the governing equations and the boundary conditions and is introduced by the trial function approximation.

b) The second error is known as round off error and is introduced due to the fact that only a finite amount of information may be stored at any stage of calculation process.

c) The third error is due to the approximations involved in the mathematical model to which the numerical solution is applied.

With the growing precision of today's computers, the second type of errors can be minimised, while some aspects of the errors that are involved in the mathematical model has been taken up in chapter 6. In this chapter discretisation error which arises due to approximation processes have been analysed.

Mostly we define the order of discretisation error in terms of the typical mesh size. This by itself does not determine the magnitude of the discretisation error, but if a series of
solutions on meshes of decreasing sizes is available, it allows an approximate estimate of the correct answers to be obtained. However, much more information on the possible errors is desirable, if good use is to be made of numerical approximation.

Error analysis pertains to determining the magnitude of error i.e. the difference between the exact and the approximate solution at a given finite, stage of subdivision and how best to refine the approximation to achieve results of a given desired accuracy. The most efficient method of evaluating errors is to compute them directly from the mesh structure. Such error estimates are said to be a-priori in nature. An alternative approach, and the one favoured here is to compute error estimates derived from the finite element solutions. These can only be calculated after a solution has been obtained and are therefore termed a posteriori error estimates. The error so obtained, to be meaningful, has to be processed. However, to do so many steps are required to achieve results, that are optimal. We define optimal mesh as the one that for a given number of nodal variables, gives the minimum error. Obviously this mesh is dependent on the element type being used and the way we measure the error. The premise of adaptive procedures is that, by making use of mathematics of error analysis, a finite element program can determine which region needs refining and automatically adapt the mesh to suit the problem.

This chapter is concerned with the identification of the discretization error in the finite element method and the
development of adaptive schemes which automatically enrich the finite element mesh on the basis of local error indicators to give convergence to near-optimal meshes and resolve until a prespecified accuracy is achieved.

Two of the traditional approaches of error estimation have been analysed. An attempt has been made to identify both their strengths and weaknesses. Endeavour has been made to develop two different error estimation techniques for implementation in the self-adaptive finite element solution of electro-magnetic field problems and in the processes fully automated procedures & algorithms for these approaches have been developed. The error estimator developed in Section 4.4 uses higher order polynomials for error estimation, whereas the error estimator developed in Section 4.5 is based on global gradient smoothing and is sufficiently robust to reliably indicate the error.

Once error estimates are calculated, the next stage of the adaptive process is to modify the mesh to reduce errors to an acceptable level. Adaptive mesh refinement has been taken up in Section 4.7.

4.2 Some Considerations In A-Posteriori Error Analysis

Eq. (2.19)

\[ \nabla \cdot \mu (\nabla \times A) = -\mathbf{J} \]

and eq. (2.16)

\[ \nabla \cdot \mathbf{\mu} \nabla \phi = 0 \]

can be written in the generic form
\( L u = g \) ... (4.1)

where \( L \) is a linear differential operator. In the F.E. method, \( u \)
is either \( \phi \), the scalar potential or \( A \), the vector potential.

Let \( \phi \) be approximated locally by

\[ \phi = N_i \phi_i \]

where \( N_i \) are the shape functions and \( \phi_i \) are
unknown parameters to be determined. The local error in a
computed solution can be expressed as [2]

\[ e = \phi - \hat{\phi} \] ... (4.2)

the difference between the exact solution \( \phi \) and the
approximate solution \( \hat{\phi} \), obtained by solving eq. (4.1) using
the F.E. method.

4.2.1 The Error Norm and Error Estimates

Eq (4.2) is only immediately useful when the exact solution
\( \phi \) is known. It is also occasionally misleading. For instance,
near reentrant corners, errors in potentials may be locally
infinite but overall solution may well be acceptable. In any case
the eq (4.2) contains no indication of how well the original
equations were satisfied. It is therefore, usual to introduce
norms, representing some integral scalar quantity to measure the
error or indeed the function itself.

A measure defined as the energy in the error, or the energy
norm is given by the following expression [3]

\[ ||e|| = \int_{\Omega} e^T L e \, d\Omega \] ... (4.3)

which can be related to the domain residual \( r \)
\[ r = L \hat{\phi} - g \quad \ldots (4.4) \]

which indicates how well the original equations are satisfied by the approximate solution \([1]\). Using eqs (4.2 and 4.4), eq. (4.3) can be rewritten as

\[ ||e|| = \int_{\Omega} e^T L r \, d\Omega \quad \ldots (4.5) \]

This energy norm can be evaluated over the whole domain or over sub domain or even individual elements. The global error norm then is defined by \([3]\)

\[ ||e||^2 = \sum_{i=1}^{n} ||e||^2_i \quad \ldots (4.6) \]

where \(i\) refers to individual elements such that their union is \(\Omega\). The local error \(e\) can be estimated by different techniques. In recent years several methods of error estimation have been reported \([4-7]\). They all fall in two basic categories:

a) Error estimates based on approximating the error function.

b) Error estimates based on dual or complementary variational principles.

The error estimates allows us to calculate for each element, an absolute value of the error, and in case of error indication, an element value relative to other elements in the mesh \([4]\).

4.2.2 Error Indicators and Refinement Parameter

The next step in a self-adaptive process is to decide which elements require refinement and for this we have the problem of choosing an appropriate field variable, or related quantity, as a
basis for refinement. Using the chosen refinement parameter the error indicator should be able to
a) make comparison with error indicators from elements of different materials effectively. Consider the case of a magneto static system. We may choose an energy density parameter employing the flux density like $\frac{1}{2} \mathbf{B} \cdot \mathbf{H}$, since it is energy based, it is obviously related to the functionals [5]. However such a parameter is significantly lower in high permeability iron regions than it is in air. This results in relatively little refinement in the iron, but an over-refinement in the remaining regions. The behaviour of the magnetic field in iron is of great importance to the global solution, and so parameters of this type would not help the convergence of the solution process. A flux density parameter such as $(\mathbf{B} \cdot \mathbf{B})^{1/2}$ would have opposite effect, since it gives very high values in the iron and lower values elsewhere. However, the energy in the iron provides only a minimal contribution to the total value of the functional. This can be allowed for by introducing the parameter $F$, such that

$$F = \int_{\Omega} (\mathbf{B} \cdot \mathbf{B})^{1/2} \, d\Omega \quad \ldots \quad (4.7)$$

This quantity may then be used as an alternative measure of global convergence.

b) account for the direction of the field quantity.

For example if the parameter chosen for evaluating error in element $i$, say is denoted by $P_i(\mathbf{B})$ then one may choose either

$$E_{pi} = P_i(\mathbf{B}_{\text{comp}}) - P_i(\mathbf{B}_{\text{stan}}) \quad \ldots \quad (4.8)$$

$$E_{pi} = P_i(\mathbf{B}_{\text{comp}} \cdot \mathbf{B}_{\text{stan}}) \quad \ldots \quad (4.9)$$
where the subscripts refer to the standard and complementary solutions. It is obvious that eq. (4.8) ignores errors in field direction, which may be of vital importance, therefore, eq. (4.9) is most likely to provide better error indication [6]. It may also be beneficial to weight the error indicators by the element area. For example

\[ E_{pl} = \int_{\Omega} P_i (B_{\text{comp}} - B_{\text{stan}}) \, d\Omega \quad \ldots \quad (4.10) \]

The reasoning behind the use of such a parameter is that field errors in larger elements provide a greater contribution to the global error than the same error in a small element. The problem is that, when an element is refined, its error parameter is reduced dramatically. This is more due to the reduction in element area than due to decrease in the actual element error. This is obviously wrong and means that error indicators based solely on energy difference [8] may not be too effective. Area weighting does, however, tend to limit the ratio of maximum element area to minimum element area and this can lead to refinement in regions where none is necessary.

c) indicate all or most of those elements which when refined, will reduce the error to an acceptable level. This can be achieved by operating only those elements which exceed a prescribed tolerance level, say \( e_{\text{tol}} \), which may be specified in relative terms (percentage values) or in absolute terms. The use of a percentage error parameter also causes special problems since elements containing very small fields can have a high
percentage error although their contribution to the global error is negligible. Using such a parameter reduces the rate of convergence of the method. Alternatively an average tolerance could be used, where the tolerance level is set to the average of all the error indicators. An additional specification of a global tolerance for the error estimator which can be used to terminate the programme if necessary, may be used. This can be expressed either in absolute or relative terms because it is a global quantity.

4.2.3 Error Processing

The next stage of the self-adaptive process is to take the error indicators which are computed for every element in the mesh and use them to determine a suitable refinement index 'k' so as to achieve an optimum mesh with a minimum of self-adaptive iterations, and hence a minimum of finite-element solutions. In the present work, depending upon the refinement index the given triangle has been divided into two or three or four.

4.3 Error Estimation Based on Insufficient Approximations of Potentials

4.3.1 Introduction

Methods using Ampere's law, for verification of boundary continuity between elements in vector potential have been applied with success [9-12]. Referring to Fig. 4.1

\[ \int_{\Gamma_1} \mathbf{H} \cdot d\mathbf{l} = \int_{\Gamma_2} \mathbf{H} \cdot d\mathbf{l} + \int_{\Gamma_3} \mathbf{H} \cdot d\mathbf{l} = \int_{\partial\Omega} \mathbf{J} \cdot d\mathbf{s} \]

... (4.11)
But in some problems, the approximation may be insufficient and may present several difficulties during implementation in problems. For example, using eq. (2.10)
\[ \nabla \times \mathbf{A} = \mathbf{B} \]
eq (2.4) is immediately verified.
\[ \operatorname{V} \cdot \nabla \times \mathbf{A} = 0 \quad \ldots \quad (4.12) \]
But if
\[ \mathbf{H} = \frac{1}{\mu} \mathbf{B} = \frac{1}{\mu} \nabla \times \mathbf{A} \quad \ldots \quad (4.13) \]
is used, eq. (2.2) is not verified; \( \nabla \times \mathbf{A} \) is not continuous at the interface between elements when the F.E. method is used.
* The normal component of \( \mathbf{B} \) is continuous, not the tangential component of \( \mathbf{B} \).
* The tangential component of \( \mathbf{H} \) is continuous, not the normal component of \( \mathbf{H} \).

4.3.2 Error Formulation

The error estimator developed takes into account insufficient approximation of vector potential or scalar potential due to discontinuities of boundary conditions between elements \([13,14]\). For magneto static vector potential problems
\[ \nabla \times (\mathbf{H} + \Delta \mathbf{H}) = \mathbf{J} \quad \ldots \quad (4.14) \]
\[ \nabla \cdot (\mathbf{B} + \Delta \mathbf{B}) = 0 \quad \ldots \quad (4.15) \]
with the constitutive relations
\[ \mathbf{B} = \mu \mathbf{H} \quad \ldots \quad (2.6) \]
\[ \Delta \mathbf{B} = \mu \nabla \mathbf{H} \quad \ldots \quad (4.17) \]
with

105
\[ B = \nabla \times A \]
\[ \Delta B = \nabla \times e \quad \ldots \quad (4.18) \]

where \( e \) is the unknown error given by eq. (4.2).

From eq. (4.14)
\[ \nabla \times \Delta H = J - \nabla \times H \quad \ldots \quad (4.19) \]

Eq. (4.15, 2.6, 4.16, 2.8 and 4.17), lead to a differential equation of the form.
\[ \nabla \times \frac{1}{\mu} \nabla \times e = J - \nabla \times \frac{1}{\mu} \nabla \times A \quad \ldots \quad (4.20) \]

Applying Galerkin method
\[ \int_{\Omega} N (\nabla \times \frac{1}{\mu} \nabla \times e) \, d\Omega = \int_{\Omega} N (J - \nabla \times \frac{1}{\mu} \nabla \times A) \, d\Omega \ldots \quad (4.21) \]

Using the Green-Gauss theorem the vector identities becomes
\[ -\int_{\Omega} N_{1} \frac{1}{\mu} \nabla \times e \, d\Omega + \int_{\Omega} A (N_{j} - \frac{1}{\mu} \nabla \times e) \, d\tau = \int_{\Omega} N_{j} J \, d\Omega + \int_{\Omega} N_{1} \frac{1}{\mu} \nabla \times A \, d\Omega - \int_{\tau} A (N_{j} \frac{1}{\mu} \nabla \times A) \, d\tau \quad \ldots \quad (4.22) \]

where \( n \) is the normal vector, \( N_{i} \) are the shape functions and \( \tau \) denotes the interface between the elements.

Setting the residual of the system to zero
\[ r = \int_{\Omega} N_{1} J \, d\Omega + \int_{\Omega} \nabla \times N_{1} \frac{1}{\mu} \nabla \times A \, d\Omega = 0 \quad \ldots \quad (4.23) \]

and
\[ \int_{\tau} n A (N_{j} \frac{1}{\mu} \nabla \times e) \, d\tau = 0 \quad \ldots \quad (4.24) \]

For each element, the following system of linear algebraic equations can be expressed as
For magneto static scalar potential problems, the system is of the same kind, and for each element the system of equations is written as follows:

\[
\int_{\Omega} \nabla \times N_1 \frac{1}{\mu} \nabla \times e \, d\Omega = \int_{\tau} n \cdot (N_j - \frac{1}{\mu} \nabla \times A) \, d\tau \quad \ldots (4.25)
\]

Boundary Continuity Between Elements

In section 4.3.2, the error which may occur in the F.E. method was defined. The errors in eqs. (4.24) and (4.25) are directly derived from the boundary continuity between elements.

In vector potential problems the error is derived from \( B_t \) discontinuity where \( B_t \) is calculated with the three nodal points as shown in Fig. 4.2(a).

In scalar potential problems the error is derived from \( H_n \) discontinuity where \( H_n \) is calculated with the three nodal points as shown in Fig. 4.2(b).

4.3.3 The Error System Equations

The usual interpolation is used. In two dimensional vector potential problems, for each element the equations written as follows are used.

\[
[S_{ij}] [A_j] = [D_1] \quad \ldots (4.26)
\]

where

\[
S_{ij} = \int_{\Omega} \frac{1}{\mu} \nabla \cdot N_1 \nabla \cdot N_j \, d\Omega \quad \ldots (4.27)
\]

107
The solution of the system eqs. gives nodal values of the error which allow the mesh refinement to be adapted automatically.

In two dimensional scalar potential problems, for each element the equations written as follows are used.

\[ [K_{ij}] [\phi_j] = [0] \]  \hspace{1cm} \text{(4.31)}

where

\[ K_{ij} = \int_{\Omega} \mu \nabla N_i \cdot \nabla N_j \, d\Omega \]  \hspace{1cm} \text{(4.32)}

after

\[ [K_{ij}] [e_j] = [C_1] \]  \hspace{1cm} \text{(4.33)}

where

\[ C_1 = \int_{\tau} \mu N_i (\partial N_j / \partial n) \, d\tau. \phi_j \]

The solution of the system equations gives nodal values of the error which allow the mesh refinement to be adapted automatically.

4.4 Error Estimation Using Hierarchic Formulation

4.4.1 Introduction

It is usual in conventional finite element analysis to refine the mesh where the field gradient is highest. A local measure of this gradient in element \( i \) of area \( \Omega_i \) is given by
\[ \eta_1 = \int_{\Omega_1} |\nabla A|^2 \, d\Omega \quad \ldots \quad (4.34) \]

where \( A \) is the vector potential. If one were to refine the grid wherever \( \eta_1 \) is large, then most elements would be placed where the flux is concentrated. Alternatively one could refine the mesh according to the jump in the normal derivative at inter element boundaries

\[ \lambda_1 = \int \left[ (\nabla A_i - \nabla A_j) \cdot \mathbf{n} \right]^2 \, dS \quad \ldots \quad (4.35) \]

or by using a combination of \( \eta_1 \) and \( \lambda_1 \). This procedure was proposed and analysed by Babuska and Rheinboldt [15]. Unfortunately, methods based on the parameters \( \eta_1 \) & \( \lambda_1 \) to refine the mesh in the finite element process do not always work. The problem is that \( \eta_1 \) & \( \lambda_1 \) do not measure the errors in the solution, only its steepness and curvature. If the field is steep but linear, as it is for instance in the center of a limb of a transformer core, then the error in the computed field is zero, but \( \eta_1 \) will have a large value.

### 4.4.2 Error Formulation

A better approach is to solve for the error function directly. The governing poisson equation of magnetostatics is

\[ \nabla \cdot \frac{1}{\mu} \nabla \times A = - J \quad \ldots \quad (2.11) \]

Solution of this equation by F.E. method gives the approximate solution \( \hat{A} \). Then error \( e \) equals the difference between the exact solution and approximate solution \( \hat{A} \).
\[ e = \mathbf{A} - \hat{\mathbf{A}} \]...

Substituting eq. (4.36) into eq. (2.11) gives
\[ \nabla \cdot \left( \frac{1}{\mu} \nabla e \right) = -r \]...

where \( r \) is the residual and is given by
\[ r = \nabla \cdot \left( \frac{1}{\mu} \nabla \mathbf{A} + \mathbf{J} \right) \]...

Introducing the approximations
\[ \hat{\mathbf{A}} = \sum_i N_i(x,y) \hat{A}_i \]...
\[ e = \sum_i P_i(x,y) e_i \]...

where \( N_i \) are the \( n \)th order finite elements shape functions used to find \( \hat{\mathbf{A}} \). Substituting eqs. (4.39) and (4.40) into eq. (4.36) and taking the inner product with \( P_i \) gives
\[ \begin{bmatrix} S \end{bmatrix} \begin{bmatrix} e \end{bmatrix} = \begin{bmatrix} R \end{bmatrix} \]...

where the elements of the matrix \( S \) and the vector \( R \) are.
\[ S_{ij} = \int_{\Omega} \frac{1}{\mu} \nabla P_i \nabla N_j \, d\Omega \]...
\[ R_i = \sum_j (S_{ij} - B_{ij}) \hat{A}_j - \int_{\Omega} P_j J \, d\Omega \]...

here
\[ B_{ij} = \int_{\Omega} \frac{1}{\mu} P_i \frac{\partial N_j}{\partial n} \, d\Omega \]...

since \( \frac{\partial N}{\partial n} \) is discontinuous we need to rewrite it as
\[ B_{ij} = \frac{1}{2} \int_{\Omega} P_i \left( \frac{1}{\mu} \frac{\partial N_1}{\partial n^+} + \frac{1}{\mu} \frac{\partial N_1}{\partial n^-} \right) \]...

110
While the above analysis is straightforward, its application to adaptive mesh refinement is not clear. If one were to use the same mesh and the same approximation functions to compute the error function ‘e’ as had been used to find \( \hat{A} \), then we would obtain \( e = 0 \). If we introduce an additional, complete polynomial to the approximation, then the difference between two such consecutive solutions will give an estimate to the local error albeit at a very considerable cost. On the other hand, refining the mesh everywhere and solving resulting large global matrix for \( e \) is again prohibitively expensive. However if a hierarchic form of approximation is used this difference can be obtained at more reasonable cost, utilizing the original approximation.

4.3.3 Higher Order Approximation using Hierarchic Formulation

In such a hierarchic form the discretized equations are \([1]\) for \( \hat{A} = N \hat{A} \)

\[
\begin{align*}
S \hat{A} &= R \\
\text{and, with hierarchic added variable} \quad \hat{A}^h,
\end{align*}
\]

\[
\begin{align*}
\hat{A} &= N \hat{A}^n + N^h \hat{A}^h \\
\begin{bmatrix} S & S^{nh} \\ S^{nh} & S^{hh} \end{bmatrix} \begin{bmatrix} \hat{A}^n \\ \hat{A}^h \end{bmatrix} &= \begin{bmatrix} R_1 \\ R_2 \end{bmatrix}
\end{align*}
\]

(4.46a)

Solution of eqs. (4.46) will, of course, yield the error approximation as

\[
e = N(\hat{A}^n - \bar{A}) + N^h \hat{A}^h
\]

(4.47)

A, reasonable approximation, however can be made by putting

\[
\hat{A}^n = \bar{A}
\]

(4.48)
into the second equation, i.e., using the original approximation.
Now $A^h = (S^{hh})^{-1} (R_2 - S^{hn} A)$ ... (4.49)
and to get this only the inversion of the matrix $S^{hh}$ is needed.
Even a cruder approximation will suffice by introducing hierarchical refinement one by one. Now for each degree of freedom introduced at a 'node' $i$, $A^h_i$ is a scalar and

$$(S^{hh})^{-1} = \frac{1}{S^{hh}}$$

therefore

$$e = \frac{N^h}{S^{hh}} (R_2 - S^{hn} A)$$

... (4.51)

4.4.4 Implementation of the Adaptive Procedure

This error estimate can be translated into error measure in the following way.

The solution using original approximations, which in our case are first order shape functions gives us the value of potential at the nodes. For each triangle side, value of $A$ at the mid-point $A_{av}$ is calculated, and which is nothing but the average value of potential $A$ at the two end nodes of the side in consideration.

For approximating the error function, second order hierarchical shape functions have been used. Its solution gives us the values at the three nodes of each side of the triangles. Designating the value at the mid node of the side of the triangle...
in consideration as $A_{\text{mid}}$, the ratio $\frac{A_{\text{mid}}}{A_{\text{avg}}} = \xi$ is found. In all triangles, $\xi$ is calculated on each side and compared with the given error tolerance $\tilde{\xi}$.

• If in a given triangle, $\xi > \tilde{\xi}$ at one of the three sides, then that side is bisected resulting in division of the given triangle into two.

• If in a given triangle, $\xi > \tilde{\xi}$ at two of the three or all sides, then that triangle is trisected.

• If in a given triangle, $\xi > \tilde{\xi}$ at all sides, then that triangle is divided into four.

4.4.5 Illustrative Example

For testing the accuracy of the program, we have solved the transformer core problem for the potential gradients at various points inside the core. The dimensions of the problem under consideration has been shown in Fig. 4.3. The values of magnetic vector potential at boundary AB and BC have been taken as zero and at DE and EF as 10. The total domain has been divided in four parts. These four parts are identical to each other for their initial values and boundary conditions. So, to save the execution time and computer resources we have restricted our analysis to only one part shown shaded in Fig. 4.4. For other parts potential distribution will be exact replica of this portion.

Initially the domain under analysis has been discretized into first order quadrilateral elements (Fig. 4.5).
This initial mesh served as input data to the computer program for doing the analysis for finding out the potential distribution inside the core. After solving the problem on this mesh an estimate of the error has been made using the technique developed in this section and implemented through the subroutine ERRESTH. Elements having energy error more than the prespecified are refined by automatic activation of the procedures developed for adaptive refinements in Section 4.7. The mesh status at each iteration is shown in [Fig.4.4(a) to (c)]. The analysis is stopped when energy difference of various elements is less than specified, i.e., 1%. The final mesh which converges to the solution corresponds to Fig.4.5 (d). Equipotentials corresponding to each iteration have been plotted in Fig.4.8, 4.8(a), 4.8(b). From the analysis it has been observed that
i) the elements are actually concentrated in regions where the potential gradients were high
ii) when mesh was graded with smaller elements in regions of high gradients, the calculated values of potential gradients were in agreement with the experimental results (Fig. 4.8).

4.5 Error Estimation Based on Global Smoothing of Gradients

4.5.1 Introduction

The governing field equation for steady state magneto static field given by eq.(2.14)

\[
\frac{\partial}{\partial x} \left( \mu_x \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left( \mu_y \frac{\partial \phi}{\partial y} \right) = 0 \quad \ldots \quad (2.14)
\]

1.14
with the boundary conditions of
\[ \phi = \phi_0 \quad \text{on boundary } \Gamma_1 \]
\[ -\mu \frac{\partial \phi}{\partial n} = B_n \quad \text{on boundary } \Gamma_2 \]

Discretizing in space by F.E. method yields a system of algebraic equations. Scalar potential field is then obtained by solving this set of equations. Errors of interest in the field problem are given by the following difference expressions viz:
\[ e_\phi = \phi - \hat{\phi} \quad \ldots (4.52) \]
\[ e_B = B - \hat{B} \quad \ldots (4.53) \]
\[ -\mu_x \frac{\partial \phi}{\partial x} = B_x \quad \ldots (4.54) \]
\[ -\mu_y \frac{\partial \phi}{\partial y} = B_y \quad \ldots (4.54) \]

where \( \phi \) and \( B \) are actual potential & potential gradients, while \( \hat{\phi} \) and \( \hat{B} \) are the solution obtained from the F.E. analysis.

### 4.5.2 Error Formulation

The energy norm in the magneto static problem can be defined as follows
\[ ||e|| = \left( \int_\Omega \mu_x \left( \frac{\partial \phi}{\partial x} - \frac{\partial \hat{\phi}}{\partial x} \right)^2 + \mu_y \left( \frac{\partial \phi}{\partial y} - \frac{\partial \hat{\phi}}{\partial y} \right)^2 \, dx \, dy \right)^{1/2} \quad \ldots (4.55) \]

or in the vector form as
\[ ||e|| = \left( \int_\Omega (\nabla \phi - \nabla \hat{\phi})^T \mu (\nabla \phi - \nabla \hat{\phi}) \, d\Omega \right)^{1/2} \quad \ldots (4.56) \]

where

115
\[ \mu = \begin{pmatrix} \mu_x & 0 \\ 0 & \mu_y \end{pmatrix} \] ... (4.57)

It can be shown that \[ f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega = f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega = f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega \] ... (4.58)

Thus we have

\[ ||e||^2 = f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega - f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega \] ... (4.59)

if we define

\[ ||B||^2 = f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega \]

\[ ||\hat{B}||^2 = f_{\Omega}(\nabla \phi)^T \mu \nabla \phi \, d\Omega \] ... (4.60)

Eq. (4.59) can be rewritten as

\[ ||e||^2 = ||B||^2 - ||\hat{B}||^2 \] ... (4.61)

Expressing the error norm in terms of a percentage error, that is

\[ \eta = \frac{||e||}{||B||} \times 100 \% \] ... (4.62)

The significance of the term \[ ||e|| / ||B|| \], can be realized from the following analysis. We consider a special boundary condition where \( \phi = 0 \) and \( \partial \phi / \partial n = 0 \) on domain boundary \( \Gamma \). Using Green's lemma we can write

\[ ||B||^2 = \int_{\Omega} (\nabla \phi)^T \mu \nabla \phi \, d\Omega = - \int_{\Omega} \phi \nabla (\mu \nabla \phi) \, d\Omega \] ... (4.63)

and similarly for \[ ||\hat{B}||^2 \]
\[ ||\mathbf{B}||^2 = \int_{\Omega} (\nabla \phi)^T \mu \nabla \phi \, d\Omega = \int_{\Omega} (\nabla \phi)^T \mu \nabla \phi \, d\Omega = -\int_{\Omega} \phi \nabla (\mu \nabla \phi) \, d\Omega \]

Thus from eqs. (4.64) & (4.65) we have

\[ ||e||^2 = ||B||^2 - ||\hat{B}||^2 = -\int_{\Omega} (\phi - \hat{\phi}) \nabla (\mu \nabla \phi) \, d\Omega \]

\[ \eta^2 = \frac{||e||^2}{||B||^2} = \left| \frac{\phi - \hat{\phi}}{\phi} \right| \]

where \((\phi - \hat{\phi})^*\) and \(\phi^*\) are the potentials at the appropriate points. Since a uniform distribution of errors is expected, therefore \(\eta^2\) can be thought of as representing an average error for the field potentials.

4.5.3 Gradient Smoothing

Before the implementation of the adaptive procedure is discussed the problem of calculating the error norm \(||e||\) must be addressed as it contains so-called ‘exact’ values of the potential gradients, which are obviously not available. In this study the smoothed continuous values of potential gradients [16] are used. In the smoothing process it is assumed that the approximation quantities are interpolated by the same basis functions as the variable \(\phi\). The smoothed continuous values of potential gradients are represented in the following manner as given by Hinton and Campell [17]. However, basis functions of the same order have been used here for smoothing as for the main variable \(\phi\), unlike lower order basis function used in [17].
smoothing procedure involves representing the smoothed gradients \( \tilde{G} \) in the element in terms of their unknown nodal values \( \tilde{G}_1 \), i.e.

\[
\tilde{G} = \sum_{i=1}^{n} N_i \tilde{G}_1
\]

... (4.67)

where \( n \) is the number of nodes per element and \( N_i \) are smoothing shape functions which are usually defined on the original mesh of elements. If \( \hat{G} \) represents the non-smoothed values obtained from the original finite element analysis, then the problem becomes one of finding the nodal values \( \tilde{G}_1 \) which minimize the functional

\[
I = \int_{\Omega} (\tilde{G} - \hat{G})^2 \, d\Omega
\]

... (4.68)

Hence for \( I \) to be a minimum

\[
\frac{\partial I}{\partial \tilde{G}} = 0
\]

... (4.69)

This leads to a set of linear simultaneous equations in \( \tilde{G} \) as follows

\[
\left( \int_{\Omega} N_i N_j \, d\Omega \right) \tilde{G}_1 = \int_{\Omega} N_i \hat{G} \, d\Omega
\]

... (4.70)

It may be re-written as

\[
\sum M_{ij} \tilde{G}_1 = f_i
\]

... (4.71)

The 'smoothing matrix' \( M_{ij} \) in each element is given as:

\[
M_{ij} = \int_{-1}^{1} \int_{-1}^{1} N_i N_j \det J \, d\xi \, d\eta
\]

... (4.72)

and the 'smoothing forces' \( f_i \) in each element are given as:

\[
f_i = \int_{-1}^{1} \int_{-1}^{1} \hat{G} \det J \, d\xi \, d\eta
\]

... (4.73)

When Gauss-Legendre integration rule is applied to the 'smoothing
forces' we obtain

$$f_i^e = \sum_i N_i (\xi_k, \eta_k) \det J (\xi_k, \eta_k) \hat{G} (\xi_k, \eta_k)$$  \quad (4.7)$$

where \( \hat{G} (\xi_k, \eta_k) \) are the potential gradients at Gauss point \((\xi_k, \eta_k)\) which are available from the finite element solutions. It has been proved that the potential obtained by the finite element method is most accurate at nodal points [18], whereas the potential gradient is most accurate at Gauss points [19]. This characteristic is often referred to as superconvergence phenomena. After solving this equation system, globally smoothed values of potential gradients are obtained with the higher approximation to the exact solution. With a higher approximate available, it is now possible to carry on with adaptive analysis using error estimation [20].

4.5.4 Implementation of the Adaptive Procedure

The first step for implementing adaptive analysis is to specify a maximum permissible error \( \eta \) that is to be achieved at the end of the analysis. The requirement for a near optimum analysis is that all the elements of the final mesh must contain an approximately equal error. After the first analysis, from a preliminary coarse mesh, the square of the total error is calculated, which is the sum of all individual element contributions, that is
\[ ||B||^2 = ||\hat{B}||^2 = \sum_{i=1}^{m} \int_{\Omega_i} (\nabla \hat{\phi})^T \mu \nabla \hat{\phi} \, d\Omega_i \]  \hspace{1cm} (4.75)

\[ ||B||^2 = ||\hat{B}||^2 = \sum_{i=1}^{m} \int_{\Omega_i} \left( \mu_x \left( \frac{\partial \hat{\phi}}{\partial x} \right)^2 + \mu_y \left( \frac{\partial \hat{\phi}}{\partial y} \right)^2 \right) \, dx \, dy \]  \hspace{1cm} (4.76)

where \( m \) is the total number of elements. Now, the maximum permissible error for each element can be calculated by distributing \( ||B||^2 \) equally over all the elements, i.e.

\[ ||\tilde{e}||^2 \leq \tilde{\eta} \left( \frac{||B||^2}{m} \right) \]  \hspace{1cm} (4.77)

where \( \tilde{\eta} \) is some specified maximum value. The approximate error in each element after the first analysis can be calculated according to eq. (4.49) using, of course, the smoothed values of potential gradients instead of the exact values. This can be written for a two-dimensional element \( i \) as

\[ ||\tilde{e}||_{1} = \left( \int_{\Omega_i} \mu_x \left( \frac{\partial \hat{\phi}}{\partial x} - \frac{\partial \phi}{\partial x} \right)^2 + \mu_y \left( \frac{\partial \hat{\phi}}{\partial y} - \frac{\partial \phi}{\partial y} \right)^2 \, dx \, dy \right)^{1/2} \]  \hspace{1cm} (4.78)

where \( \frac{\partial \hat{\phi}}{\partial x}, \frac{\partial \hat{\phi}}{\partial y} \) are the smoothed values of potential gradients and are the finite element solutions. Now this error is compared for all elements to the maximum permissible error in an element as calculated above and used to modify the mesh for a second analysis. If we define a variable \( \xi_1 \), where

\[ \xi_1 = \frac{||\tilde{e}||_{1}}{||\tilde{e}||_{1}} \]  \hspace{1cm} (4.79)

If \( \xi_1 > 1 \), the size of element \( e \) must be reduced and the mesh
will require refinement.

4.5.5 Illustrative Example

For testing the accuracy of the program, transformer core problem of finding the potential gradients at various points inside the core has been solved. This is the same problem as has been considered in Section 4.4 and shown in Fig.4.3. To check the efficacy of the error estimator developed the problem has been solved using triangular and quadrilateral elements.

Initially the domain under analysis is divided into second order triangular elements having 65 nodes and 24 elements. The values of potential at boundary nodes 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 19, 28, 37, 46, 51, 56, 61 is 0 volts & at nodes 45, 44, 43, 42, 41, 50, 55, 60, 65 is 10 volts. This initial mesh served as input data to the computer program for doing the analysis to find out the potential distribution inside the core as is shown in Fig.4.5. After solving the problem on this mesh an estimate of the error has been made automatically using the technique developed in this section, by the subroutine ERRESTG. Elements having energy error more than the prespecified have been adaptively refined by procedures developed for adaptive refinements in Section 4.7. The mesh obtained at each iteration is shown in Fig.4.5(a) to (d). The analysis is stopped when energy difference of various elements is less than 1%. The final mesh which converges to the solution is shown in Fig.4.5(e).

The problem was also solved using first order quadrilateral elements (Fig.4.6) and first order triangular
elements (Fig. 4.7). The respective meshes generated by the programme on subsequent runs are reproduced in Figs. 4.6(a) to (c) and Figs. 4.7(a) to (c). The final mesh when the problem converges to the solution is shown in Fig. 4.6 (c) and Fig. 4.7 (c) respectively. From these meshes it can be seen that the elements are actually concentrated in regions where the potential gradients were high. The equipotentials have been plotted in Fig. (4.8). The results were also compared with Maxwell package and it has been found that the results are matching up to 2 decimal places, which confirms the accuracy of our program.

4.6 Error Estimation Based on Dual and Complementary Variational Techniques

4.6.1 Introduction

An entirely different approach to error analysis uses the concept of upper and lower energy bounds [21]. In electromagnetics, duality is manifest in at least two different respects. First, the field equations themselves are symmetric in dual variables; second energy expressions are available as both lower and upper bounds.

There are at least four different approaches to solving a given magnetic field problem; the difference between the solution provide an excellent measure of accuracy obtained. These techniques have been reported by Penman et al. [22], and they are used to provide error bounds to the global energy of a given problem. Also, Pinchuk et al. [6] have proposed a technique based on complementary energy methods that provides local error bounds.
4.6.2 Error Formulation

The basic equations for computing the magnetostatic field in source free region are given by eqs. (2.2, 2.4 and 2.6).

\[ \nabla \times \mathbf{H} = \mathbf{J} \quad \ldots \text{(2.2)} \]
\[ \nabla \cdot \mathbf{B} = 0 \quad \ldots \text{(2.4)} \]

with the constitutive relation

\[ \mathbf{B} = \mu \mathbf{H} \quad \ldots \text{(2.6)} \]

By using approximate methods it is not possible to solve eqs. (2.2 and 2.4) exactly, rather a whole range of solutions is possible depending whether one approximate eq. (2.2) or (2.4) or both in varying degrees. There are two known methods of solving this problem.

**Method 1** Solve eq. (2.2) exactly by introducing the magnetic scalar potential

\[ \mathbf{H} = \nabla \phi \quad \ldots \text{(2.3)} \]

and substituting eqs. (2.6 & 2.3) into (2.3) we get

\[ \nabla \cdot \mu \nabla \phi = 0 \quad \ldots \text{(2.14)} \]

This equation is then solved by F. E. method after introducing boundary conditions.

**Method 2** Solve eq. (2.4) exactly by introducing the magnetic vector potential

\[ \mathbf{B} = \nabla \times \mathbf{A} \quad \ldots \text{(2.8)} \]

and substituting eqs. (2.6 & 2.8) into (2.2) we get

\[ \nabla \times \frac{1}{\mu} \times \mathbf{A} = 0 \quad \ldots \text{(4.80)} \]
This equation is then solved by F. E. method after introducing boundary conditions.

A duality exists between these two methods as can be seen from the following discussion.

The energy stored in magnetic field is given by

$$W(H, B) = \frac{1}{2} \int H \cdot B \, d\Omega \quad \ldots (4.81)$$

The error in $H$ and $B$ equals the difference between the approximate and the exact solutions. Thus the cross compute error energy is

$$W(H - H', B - B') = \frac{1}{2} \int (H - H') \cdot (B - B') \, d\Omega \quad \ldots (4.82)$$

Using eqs. (2.8 and 2.13) gives

$$W = \frac{1}{2} \int \nabla (\phi - \phi') \cdot \nabla \times (\mathbf{A} - \mathbf{A}) \, d\Omega \quad \ldots (4.83)$$

Using a standard vector identity, this may be written as a surface integral

$$W = \frac{1}{2} \int (\mathbf{A} - \mathbf{A'}) \times \nabla (\phi - \phi') \cdot \mathbf{n} \, ds \quad \ldots (4.84)$$

The boundary conditions used in method (1) and method (2) are complementary. By comparing $x$ and $y$ components, the following relationships can be established for the exact solution.

$$\frac{\partial \phi}{\partial x} = -\frac{1}{\mu} \frac{\partial \mathbf{A}}{\partial y} \quad \ldots (4.85a)$$

$$\frac{\partial \phi}{\partial y} = -\frac{1}{\mu} \frac{\partial \mathbf{A}}{\partial x} \quad \ldots (4.85b)$$
Thus a Dirchlet boundary condition on $\phi$ will result in a Neumann boundary condition on $A$ and vice versa. It follows that

$$W_{(H - H, B - B)} = \frac{1}{2} \int (H - H) \cdot (B - B) \, d\Omega = 0 \quad \ldots \quad (4.86)$$

i.e. the error in the solution of eqs. (2.2, 2.4 and 2.6) by method 1 is orthogonal in an energy sense to the error obtained by method 2.

Also the energy of the difference between solutions obtained from method 1 is an upper bound on the energies of the errors in either solution. This can be proved as follows.

Let $H_1$ be the magnetic field obtained in method 1 and $H_2$ be the magnetic field obtained in method 2. Obviously

$$\frac{1}{2} \int \mu (H_2 - H)^2 \, d\Omega \geq 0 \quad \ldots \quad (4.87)$$

and from eq. (4.81)

$$\int \mu (H_1 - H) \cdot (H - H_2) = 0 \quad \ldots \quad (4.88)$$

Adding the terms in eqs. (4.82) & (4.83) and rearranging gives

$$\frac{1}{2} \int \mu (H_2 \cdot H_2) \, d\Omega - \frac{1}{2} \int \mu (H_1 \cdot H_2) \, d\Omega \geq$$

$$\frac{1}{2} \int \mu (H \cdot H_2) \, d\Omega - \frac{1}{2} \int \mu (H_1 \cdot H) \, d\Omega \quad \ldots \quad (4.89)$$

Adding $\int \mu H_1^2 \, d\Omega$ to both sides and factorizing

$$\frac{1}{2} \int \mu (H_1 - H_2)^2 \, d\Omega \geq \frac{1}{2} \int \mu (H_1 - H)^2 \, d\Omega \quad \ldots \quad (4.90)$$
Similarly it can be shown that
\[ \frac{1}{2} \int_{\Omega} (\mathbf{H}_1 - \mathbf{H}_2)^2 \, d\Omega \leq \frac{1}{2} \int_{\Omega} (\mathbf{H} - \mathbf{H}_2)^2 \, d\Omega \quad \ldots \quad (4.91) \]
or
\[ \|e\|_{H_1,H_2} \geq \|e\|_{H,H_2} \quad \ldots \quad (4.92) \]

4.6.3 Adaptive Procedures

An estimate of the error is obtained by using the vector difference between the two complementary solutions. Eqs. (4.90 and 4.91) provide a global error on any two approximate solutions obtained by method (1) and (2). Provided that the same mesh is used for both solutions, in the F.E. method this bound is given by the sum
\[ \|e\|_{H_1,H_2} = \sum_{i=1}^{M} \|e\|_{1,H_1,H_2} \quad \ldots \quad (4.92) \]
where \(M\) represents the number of elements and \(\|e\|_1\) the error bound in element \(i\)
\[ \|e\|_{1,H_1,H_2} = \int_{\Omega} \mu (\mathbf{H}_1 - \mathbf{H}_2)^2 \, d\Omega \quad \ldots \quad (4.93) \]
The convergence of the F.E. method is accelerated by selective refinement. Let \(\|e\|_{\text{max}}\) be the largest error bound in any element
\[ \|e\|_{\text{max}} = \max_{i} \|e\|_{1,H_1,H_2} \quad \ldots \quad (4.94) \]
and refine the \(i^{\text{th}}\) element into \(m_i\) sub-elements such that
where $f(x)$ is an increasing integer function of $x$.

Applied iteratively, this procedure has the effect of concentrating elements in the region with the largest errors while leaving elements with small errors alone. An example of a finite elements mesh produced by this procedure is shown in Fig. 4.7.1 Introduction

The error estimators developed in the preceding sections allow the global energy (or similar) norm of the error to be determined and the errors occurring locally (at the element level) are also represented. If these errors are within the limits, for instance, if the relative energy norm percentage error $\eta$ is less than some specified value $\bar{\eta}$, then, clearly the work is completed. More frequently these limits are exceeded and refinement is necessary. The problem which this section addresses is, how best to effect the refinement. Many strategies are possible and much depends on the objectives to be achieved.

One of the most drastic modification schemes is the one developed by Girdino [23], where the mesh is reconstructed completely after every error analysis. A more straightforward and
efficient approach is to simply enhance the most recently available mesh structure by increasing the order of the shape functions (p-method) or by subdivision of the already existing finite elements into elements of the same type (h-method) or by using a mixture of both.

Babuska and Szabo [24] demonstrated that, in general, the p-method of refinement has a better rate of convergence than the h-method, and that a combined h and p method is even better. Nevertheless, in this contribution the h-method has been adopted, primarily for reasons of algorithmic efficiency.

4.7.2 The h refinement process—adaptivity

In an 'optimal mesh' it is desirable that the distribution of energy norm error (i.e., $||e||_1$) should be equal between all elements. Thus if the total permissible error is determined (assuming that is given by the result of the approximate analysis) as

$$\eta(||\phi||^2 + ||e||^2)^{1/2}$$

we could pose a requirement that error in any element $i$ should be

$$||e||_1 < \eta \left(\frac{||\phi||^2 + ||e||^2}{m}\right)^{1/2} = e_m$$

where $m$ is the number of elements involved.

Elements in which the above is not satisfied are obvious candidates for refinement. Thus if we define the ratio
\[
\frac{||e||_1}{e_m} = \xi_1 \quad \ldots (4.98)
\]

we shall refine wherever

\[
\xi_1 > 1 \quad \ldots (4.99)
\]

The refinement is carried out progressively by refining only a certain number of elements in which \( \xi \) is higher than a certain limit. This process is called enrichment of mesh.

The mesh refinement should meet three conditions:

i) All previous meshes should be contained in the current (refined) mesh.

ii) Every point in the body can be included within an arbitrarily small element at any stage of the mesh refinement.

iii) The same order of approximation for the solution should be retained through all stages of the refinement process.

Also with the h-method, the mesh refinement can be either conforming or non-conforming [26]. If a quadrilateral element is chosen for the basic building block then it is usual to use a non-conforming scheme to refine the mesh. Conforming meshes, which have less computational difficulties, are normally constructed using triangular elements.

In the present work we have developed two different conforming refinement schemes applied to linear (3-noded) triangular simplex elements. These refinement schemes have been adopted because they are simple and could easily be incorporated.
as 'add-ons' to the already developed programs.

First order triangular elements are straightforward to implement and it has been proved by Babuska and Szabo [24] that they give the same rate of convergence as higher order elements. The two refinement schemes developed are described in the following sections. The scheme presented in section 4.6.3 trisects the given triangular element whereas the procedure developed in section 4.6.4 bisects the given triangle.

4.7.3 Delaunay Method

Triangular elements are optimum for use as finite elements when they are equilateral in shape. In order to obtain an accurate solution with an efficient use of nodes it is therefore necessary that the adapted refinement procedure avoids the generation of thin finite elements, especially those close to 180 degrees. In the past the Delaunay’s algorithm has been the approach used to regenerate the mesh after each refinement step. The Delaunay algorithm as discussed in section 3.2.4 provides a means of achieving this end in an efficient and reliable manner. Basically, it is a technique for improving an initial triangulation by swapping a number of the edges in the mesh [26,27].

A modified form of this algorithm can be applied to triangular finite elements, whereby the refinement process begins by inserting an extra node at the centroid of every element, where the refinement level is greater than zero. This creates three new elements. Continuing the refinement in this manner
produces unsatisfactory element after only two or three steps. This is avoided by using a side-swapping algorithm similar to that devised by Delaunay. It first selects the 'outside' edges of the original element, then the two common elements to each selected edge which form a quadrilateral. They are then checked to find out the most satisfactory diagonal. The original Delaunay algorithm chooses the diagonalisation that maximises the minimum angle occurring within the two elements. However the diagonal which minimises the maximum angle is chosen here. For well formed elements, there is little difference between these conditions, but, when some elements have poor aspect ratios, some thought shows that the approach adopted is superior, and can produce high-quality graded meshes from very basic initial meshes.

However, difficulties do arise whenever boundaries or material interfaces are encountered. Edge swapping is impossible in these situations and a slightly different approach is required. An element adjacent to a boundary or an interface can be refined in one of two ways, depending on the angle subtended by the interface side. Those elements with a small subtended angle should be trisected, and those with larger angles bisected. Or the nodes be selectively added to the mesh after the error refinement stage, in conjunction with the Delaunay's algorithm, so as to maintain the smallest angle above a specified value. This can be achieved by selecting a node referred to as seed node, which corresponds to the mid point of the longest side. The mesh structure is then improved by subjecting the new element
found, to a further edge check. Finally, nodal relaxation is employed to ensure that all elements are as close to equilateral triangles as possible. This procedure adjusts the position of every internal node in the mesh, in an attempt to make all triangles as equilateral as possible.

### 4.7.4 Bisection Method

This technique was first proposed by Rivara [28] and has two variants termed the local and the global procedures. The method commences by bisection of the longest side in each element, in which the refinement level is greater than zero. This, by itself, produces a nonconforming mesh structure. There are two different ways to make the grid conform. The local method searches for all the nonconforming elements and bisects them to eliminate the nonconforming nodes. A more satisfactory method is to repeatedly bisect the longest side of all nonconforming elements until a conforming mesh is eventually produced. This is the basis of the global method. Both the local and global methods produced a smooth transition between large and small elements, and all angles in subsequently refined triangulations are greater than one half of the smallest angle in the original triangulation [36]. We note that the Delaunay method will refine a mesh more rapidly than the bisection method, this is because bisection refines an element by dividing it in two, while Delaunay divides it in three. However mesh modification is generally a trade off between time and quality.
4.7.5 Computer Implementation

Option for incorporating either of the schemes developed in the preceding sections have been provided in the self adaptive Finite Element package. Delaunay Algorithm has been discussed in Section 3.2.4. Algorithm for Bisection method has been developed in this section and is based on Longest Side Propagation Path for the Refinement.

DEFINITION 1: For any triangle $t_0$ of any conforming triangulation $T$, the LONGEST SIDE PROPAGATION PATH of $t_0$ will be the ordered list of all the triangles $\{ t_0, t_1, t_2, t_3, \ldots, t_n \}$ such that $t_i$ is the neighbour triangle of $t_{i-1}$ by the longest side of $t_{i-1}$ for $i = 1, 2, 3, \ldots, n$.

Proposition 1: For any unstructured, non degenerated triangulation of any bounded two dimensional geometry $G$, the following properties hold

a) The LONGEST SIDE PROPAGATION PATH $\{ t_0, t_1, t_2, \ldots, t_n \}$ of any triangle is always finite.

b) The triangles $t_0, t_1, t_2, t_{n-1}$ strictly increasing longest side (if $n > 1$).

c) For the triangle $t_n$ of the LONGEST SIDE PROPAGATION PATH of any triangle $t_n$, it holds that either

i) $t_n$ has its longest side along the boundary, and this is greater than the longest side of $t_{n-1}$ or

ii) $t_n$ and $t_{n-1}$ share the same common longest side.

DEFINITION 2: We shall say that two adjacent triangles $t_n, t_{n-1}$ are a pair of terminal triangles if they share their respective (common) longest side.
It should be pointed out here that the LONGEST SIDE PROPAGATION PATH of any triangle t corresponds to an associated polygon, which in certain cases measures the local quality of the current point distribution induced by t.

ALGORITHM:
1) Generate a triangular mesh.
2) The triangular mesh generated has to be refined according to the error element in the mesh using LONGEST SIDE PROPAGATION PATH procedure, so we select an error element by inputting the number of the triangular element.
3) Length of sides of error element is calculated after the nodes of the triangle are taken from the file "out.dat". The longest side is considered and its nodes are noted as $b_1$, $b_2$. Another triangle with sides whose nodes are common to those of the longest side of the error triangle is searched from the file "out.dat" by scanning from starting till end.
   * If the nodes of such triangle except itself are found then it proceeds to CHECKT().
   * If it encounters its own nodes while scanning the file then it proceeds to BISECTO().
4) If the longest side of any two triangles found while traversing the path from the error triangle onwards are same between the similar nodes then it proceeds to BISECT(). If the longest sides are same but nodes are different (as might be the case in an equilateral or an isosceles triangle) then it proceeds to SEARCH().
5) If it finds nodes common to the longest side of the current triangle and the previous triangle then it proceeds to BISECT(). If the current triangle has the longest side as the boundary element then it proceeds to BISECTO().

6) In the above case longest side of the triangle is bisected from the vertex of the triangle. The element node list is then updated and it proceeds to SWAP().

7) The two triangles with common longest side has its longest side bisected from the respective vertices. The four triangles so formed are proceeded to the SWAP().

8) The four or two triangles so formed according to common longest side bisection or boundary bisection respectively are used. The bisected triangles and their material numbers are taken in an array and circle check and diagonal check are applied. For this a circumcircle is drawn using the nodes of that triangle. The adjacent triangle whose nodes are not in the array is considered for each triangle. The following checks are applied in the same order.

- If the material of the two triangle is different then swapping is not done otherwise it is.
- If the distance from the vertex from the adjacent triangle to the circumcentre is less than the radius of the circumcircle swapping is done.
- If the common side called diag0 of the two triangle is greater than the other diagonal diag1 of the quadrilateral so formed, then swapping is done. The file is updated according
to the new nodes formed and it goes back to the error triangle and procedure is repeated till the error triangle \( t_0 \) is bisected or swapped.

In the program used the following functions are made use of:

**ERROR ():** We define the variables to be used, graphics device is initialized and file "out.dat" is opened in the write mode which contains the tabulated information about elements and its node numbers. The error triangle number \( t_0 \) is inputted.

According to the error triangle number the corresponding nodes from the file are taken out and side lengths are calculated between two nodes at a time. If \( (x_1, y_1) \) and \( (x_2, y_2) \) be the coordinates of two nodes then side length is given by

\[
LEN = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}
\]

The longest length is stored in variable LENO. Another triangle \( t_{02} \) with same longest side as the common side with the error triangle is searched by scanning the file from top to bottom. If it finds itself then it continues in the loop till the end of the file is reached and then it proceeds to procedure BISECTO() else if another triangle is encountered it goes to procedure CHECKT().

**CHECKT ():** The triangle \( t_{02} \) found in the ERROR() procedure has its length of the sides calculated and the longest is stored in variable LEN1. If LENO = LEN1 but the nodes differ (as may be the case in an equilateral or an isosceles triangle) then it searches another triangle by following the procedure SEARCHT() and changing the value of LENO by the that of LEN1. Else if LENO
LEN1 and the nodes are same then it goes to procedure BISECTO().

SEARCHT(): This function finds the nodes of the longest side common to the current triangle \( t_{02} \) and the previous triangle \( t_{01} \). If the current triangle \( t_{02} \) has its longest side as the boundary side then it proceeds to procedure BISECTO() otherwise it goes to CHECKT().

BISECTO(): The longest side of the triangle is bisected from the vertex of the triangle. This midpoint is calculated as

\[
\begin{align*}
\text{xmid} &= \left( x_1 + x_2 \right) / 2 \\
\text{ymid} &= \left( y_1 + y_2 \right) / 2
\end{align*}
\]

The element-node list is updated and then it proceeds to procedure SWAP().

BISECT(): The two triangles with common longest side has its longest side bisected from the respective vertices shown by green lines. The midpoint is calculated as

\[
\begin{align*}
\text{xmid} &= \left( x_1 + x_2 \right) / 2 \\
\text{ymid} &= \left( y_1 + y_2 \right) / 2
\end{align*}
\]

The element-node list is updated and the four triangles so formed are then proceeded to procedure SWAP().

SWAP(): The four of the two triangles so formed according to whether common longest side bisection or boundary side bisection takes place are considered and the nodes of the bisected triangles and their material numbers are taken in an array and then circle-check and diagonal-check is applied.

The length of the common side of one of the triangle of the
array and its adjacent triangle not forming the part of the array is taken as DIAGO and DIAG1 is taken as other diagonal of the quadrilateral. Then another triangle is searched having $t_1$ and $t_2$ as its two nodes and this third node $t_3$ and its material number $M$ are stored and the following checks are applied

a) MATERIAL CHECK : If the material numbers of the two triangles is same then they are considered for circle check else ignored and the control continues in the loop.

b) CIRCLE CHECK : The distance between the nodes $t_3$ and the circumcentre (DIS) is calculated.

If DIS $\leq$ RADIUS then diagonal check is applied else swapping is rejected and control continues in the loop till it checks all the triangles in the way.

c) DIAGNOL CHECK : The other diagonal DIAG1 of the quadrilateral so formed is calculated using the same distance formula as above.

If DIAG1 $< DIAGO$ then swapping is done and control continues in the loop where upgradation of the file "out.dat" takes place according to the new nodes formed and modification of the triangular elements, else swapping is rejected and control is transferred back to the ERROR() procedure and the whole procedure is repeated till the error triangular element $t_0$ is either swapped or bisected as the case may be. The final figure is then drawn.

The above developed procedures have been sucessfully used to refine the given mesh [Fig.4.9(a)to(d)].
4.8 Termination of the Adaptive Procedure

The final feature of the self-adaptive algorithm to be considered is the stopping conditions. These are the checks that have to be made to see if the mesh adaptation algorithm should proceed further. The termination of the adaptive algorithms has to take into consideration the following:

a) The achieved accuracy. Here we impose that the error norm has to be below the specified tolerance.

b) Storage limitation and time. It is obvious that we cannot achieve an accuracy which is too high.

c) A natural restriction is that the mesh size should exceed the maximum permitted level specified by system. The mesh modification process should, therefore, be carefully controlled to prevent this limit exceeded.

After a self-adaptive process had ended, it should be available for the user to inspect, so that he can determine the locality and extent of any further refinement which should be performed if the process were to continue further. This highlights an essential component of a well-constructed self-adaptive process, i.e., solutions should be resynthesisable. If a solution has been terminated prematurely, or if a more accurate solution is required, it should be possible to re-enter the current solution in the self-adaptive scheme with a revised set of tolerance level and to continue the process where it left off.

4.9 Conclusions

The methods of estimating errors and adaptive refinement
which are the subject matter of this Chapter constitute a very important tool for practical application of finite element methods. The a-posteriori analysis presented has three objectives. Firstly, to study some of the traditional error estimators used. Secondly, to estimate the error in the F.E. solution and thirdly, to indicate as to how to improve the F.E. mesh so that the error in all elements is less than the prespecified.

The error estimator presented in Section 4.3 is based on the analysis of continuity of $H$ and $B$ at the boundaries between elements. The advantage of this approach are the easy formulation and easy application in 2-D problems. The error estimator presented in Section 4.6 is based on the complementary energy methods, wherein, one must solve both the primary problem and its dual and compute the norm of the difference between the two solutions. An additional problem with the complementary approach is the difficulty of finding the dual variational principle for all problems. In this work, two new procedures for computing errors in finite element solution for electromagnetic field problems have been developed. In Section 4.4 a self-adaptive strategy combining higher order hierarchic finite elements has been proposed. This program is adaptive since each step depends upon the information provided by the previous ones. The program is called self adaptive since no user interaction is necessary to activate the adaptive process. The solution is first evaluated on a small coarse grid where solution times are very small. As the
mesh size and matrix size grows, one interpolates the coarse grid solution onto the finer grid so that convergence is very rapid. However, if the degree of polynomial used is of very high order, it may lead to local oscillations. In Section 4.5 error estimation strategy based on global gradient smoothing has been developed. Examples using triangular or quadrilateral have been worked out, wherein the proposed schemes have refined the elements where the gradients were high. The proposed schemes are therefore highly efficient from the adaptive point of view. From amongst different error estimators, the error estimation strategy based on global gradient smoothing has been incorporated in the self-adaptive finite element developed, primarily for reasons of algorithmic simplicity. During the last 15 years the triangular and quadrilateral mesh generation problem has evolved into an important and interdisciplinary research field. In this general context, the following three related problems have been consistently considered:

1) Automatic generation of good quality of surface triangles and quadrilaterals.

2) Triangular and quadrilateral mesh refinement.

3) Triangular and quadrilateral mesh improvement.

Two approaches based on h-refinement have been mainly used to deal with some aspects of these questions. The approach based on the Delaunay algorithm assures at least in 2-dimensions, the construction of the most equilateral triangulation for a given set of N vertices. The second approach based on RIVARA
refinement algorithms guarantees the robust construction of good 
quality triangulations. For elements close to material boundaries 
these methods alone will not guarantee the removal of thin 
elements. It has therefore, has been suggested that nodes be 
selectively added to the mesh after the error refinement stage in 
conjunction with the Delaunay's algorithm, so as to maintain the 
smallest angle above a specified value. This can be achieved by 
selecting a node referred to as seed node, which corresponds to 
the mid point of the longest side.

The system developed combines the concept of Delaunay 
triangulation along with variational principles, error-
estimators proposed and adaptive refinement schemes to provide a 
grid which adapts to the characteristics of the solution.

*****
Fig. 4.1 AMPERE'S LAW

(a) $B_t$ discontinuity

(b) $H_n$ discontinuity

Fig. 4.2 BOUNDARY DISCONTINUITY
Fig. 4.3 RECTANGULAR TRANSFORMER CORE
Fig. 4.4 INITIAL MESH

14°
Fig. 4.40 THE MESH AFTER 1st RUN
Fig. 4.4 b THE MESH AFTER IInd RUN
Fig. 4.4c THE FINAL MESH AFTER CONVERGENCE
Fig. 4.5 INITIAL MESH
Fig. 4.5a MESH AFTER 1st. RUN (BEFORE CIRCLE CHECK)
Fig. 4.5 b MESH AFTER V RUN (AFTER CIRCLE CHECK)
Fig. 4.5c MESH AFTER II RUN
Fig. 4.5 d MESH AFTER III RUN
Fig. 4.5e FINAL MESH AFTER CONVERGENCE (AFTER IV RUN)
Fig. 4-6 INITIAL MESH
Fig. 4.6b THE MESH AFTER 2nd RUN
Fig 4.6c THE FINAL MESH AFTER CONVERGENCE
Fig. 4.7 INITIAL MESH
Fig. 4.7b MESH AFTER IInd RUN
Fig 4.7c MESH AFTER CONVERGENCE
Fig. 4.8: EQUIPOTENTIALS
Fig 4.9 THE GIVEN DOMAIN FOR TRIANGULATION ILLUSTRATING THE REFINEMENT SCHEME DEVELOPED
Fig 4.9 a INITIAL MESH FOR THE GIVEN DOMAIN
* ERROR TRIANGLE
Fig 4.9b END OF THE FIRST PROPAGATION PATH AND SWAPPING OF THE TRIANGLES
Fig. 4.9c NEXT PROPAGATION PATH AND SWAPPING OF THE ERROR TRIANGLE
Fig. 4.10 d FINAL TRIANGULATION AFTER SWAPPING OF ERROR TRIANGLE
REFERENCES


*****