On Error Estimators in Finite Element Analysis

by

Jean-François Hiller

Ingénieur Diplômé de l’Ecole Polytechnique

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2000

© Jean-François Hiller. All rights reserved.

The author hereby grants to MIT permission to reproduce and
distribute publicly paper and electronic copies of this thesis document
in whole or in part.

Author ..........................................

Department of Mechanical Engineering

December 15, 1999

Certified by ..........................................................

Klaus-Jürgen Bathe

Professor

Thesis Supervisor

Accepted by ..........................................................

Ain A. Sonin

Chairman, Department Committee on Graduate Students
On Error Estimators in Finite Element Analysis

by

Jean-François Hiller

Submitted to the Department of Mechanical Engineering
on December 15, 1999, in partial fulfillment of the
requirements for the degree of
Master of Science in Mechanical Engineering

Abstract

In this work, first a review and an analysis of various recovery-based error estimators for elliptic problems are given. We consider the Zhu Zienkiewicz estimator, the Hinton Campbell estimator, the estimator used in ADINA and a simple nodal averaging scheme, all of which are recovery-based algorithms. In the second part of this work, we discuss higher-order-accuracy points and demonstrate how their locations can be calculated exactly in a number of cases. In particular we show how these points do not always coincide with the points used by Barlow, nor with lower order Gauss points, as used by Zhu and Zienkiewicz. The use of the exact locations of the higher-order-accuracy points instead of the Barlow points or lower order Gauss points could result in improvements in the performance of recovery-based algorithms. From a more theoretical standpoint, our findings regarding the higher-order-accuracy points also allow us to show what analogies exist between implicit algorithms and recovery-based algorithms and to gain a better understanding of these two classes of error estimators. Most importantly, based on our findings, we develop a new error estimator for one-dimensional problems. The new error estimator makes use of shape functions which are perpendicular to the finite element space in a local sense in general, and in a global sense in some cases. This means that all calculations required by this new error estimator are carried out at the element level rather than at the patch level. This is a significant improvement over patch-based error estimators. The performance of the new error estimator is then evaluated in some example solutions.

Thesis Supervisor: Klaus-Jürgen Bathe
Title: Professor
Acknowledgments

First and foremost, I would like to thank Professor Bathe for his constant guidance and advice.

I am also thankful to my labmates in the Finite Element Research Group, Suvranu De, Dena Hendriana, Alexander Iosilevich, Ramon Silva, Juan Pontaza, and Sandra Rugonyi, for their help and friendly support.

While I was working on this thesis, I received financial help from the MTS Corporation Fellowship, the Rocca Fellowship and a Renault SA fellowship. To these organizations, I would like to express my heartfelt gratitude.
Contents

1 Introduction ......................................................................................................................... 7
   1.1 Motivations for error estimation, basic requirements ................................................. 7
   1.2 Overview of various approaches to error estimation for elliptic problems .............. 10
      1.2.1 Recovery-based error estimators ........................................................................ 10
      1.2.2 Explicit methods ............................................................................................... 11
      1.2.3 Implicit methods ............................................................................................... 11
   1.3 Objectives ....................................................................................................................... 12
   1.4 Notations and classical results in finite element methods ........................................... 12
      1.4.1 The isoparametric displacement-based finite element method ......................... 13
      1.4.2 Convergence results for the finite element method ............................................ 15
      1.4.3 Norms used for the assessment of the performance of error estimators .............. 17

2 Recovery-based error estimators ......................................................................................... 18
   2.1 Mathematical foundation .............................................................................................. 19
   2.2 Nodal averaging .......................................................................................................... 20
      2.2.1 Principle ............................................................................................................ 20
      2.2.2 Formal expression .............................................................................................. 20
      2.2.3 Analysis ............................................................................................................. 21
   2.3 Linear interpolation techniques: ADINA estimator, Hinton Campbell estimator .......... 22
      2.3.1 Principle ............................................................................................................ 22
      2.3.2 Formal expression for the ADINA error estimator ............................................. 23
List of Figures

2-1  Nodal averaging. .......................................... 20
2-2  Oscillatory behavior of the finite element method. .......... 22
2-3  Classical numbering of nodes and Gauss points. .......... 26
2-4  Mesh configurations excluded by the ZZ error estimator. .... 28
2-5  Node numbers for the problem illustrating recovery patches. .... 29
2-6  Recovery patches .......................................... 30
2-7  Recovery patches .......................................... 31
3-1  $S_1^M$ function is in FE space when one of the boundary conditions is natural ($M = 2$) ...................................... 44
3-2  $v_M$ function ................................................... 46
3-3  Pascal triangle .................................................. 56
3-4  The 4 element mesh employed to solve the example problems 1 and 2 of Section 3.4.2 ...................................... 59
3-5  Strains for example 1 ............................................ 61
3-6  Strains for example 2 ............................................ 62
3-7  Strains for example 2 ............................................ 63
3-8  Strains for example 3 ............................................ 64
3-9  Strains for example 3 ............................................ 65
Chapter 1

Introduction

1.1 Motivations for error estimation, basic requirements

The finite element (FE) method [4] has now been widely used in research and industry for many years. For a wide variety of real world mechanical problems, it is the most effective if not the only way of obtaining an estimate of mechanical quantities such as stresses and strains in a three-dimensional body subjected to loadings. The FE method, which essentially consists in searching in a finite-dimensional vector space for the best possible estimate of the solution to a variational problem defined on an infinite-dimensional vector space, intrinsically yields an approximation to the actual solution of the mathematical problem considered. Some theoretical results guarantee that, provided properly formulated procedures are applied, this estimate, which we will call the "finite element solution", converges to the solution of the mathematical problem, which we will call the "exact solution", as the dimension of the finite-dimensional vector space is increased. In other words, in terms of norms to be defined, as the distance between the finite-dimensional space and the exact solution tends to zero, so does the difference between the FE solution and the exact solution. In the case of elliptic problems, additional theoretical results give us indications of how fast this convergence can be expected to occur.
These results, however useful they may be, give little if any insight on where the error is located and which of the fields are most affected by the error. In practice, analysts have long used common sense and experience to evaluate the error, using indications such as the magnitude of discontinuities in discrete estimates of continuous continuum fields to evaluate the accuracy of the FE solution. Such approaches may yield good qualitative results when simple problems are dealt with but may prove unreliable for practical engineering problems. Graphical methods were also introduced to facilitate this approach [16].

Comparing the finite element solutions obtained from different meshes (regular and free form meshes, finer and coarser meshes) can also yield valuable information on the accuracy of a finite element solution. For instance, based on the knowledge that in elliptic problems the finite element strain energy converges from below to the exact strain energy, we can determine how fine a mesh we need to use to obtain an accurate finite element solution by plotting the finite element strain energy of ever finer meshes. This type of method only conveys information on how accurate the finite element solution is overall: no local information can be extracted.

These shortcomings have encouraged the development of rigorous procedures that yield estimates of the error. Also, we cannot expect to be able to determine the error exactly, because if we could, then we would be able to determine the exact solution to an infinite-dimensional optimization problem from a finite-dimensional vector space.

Different categories of such “error estimators” have been defined, depending on the type of measure used to estimate the error and on the underlying mathematical theory employed (see Section 1.2), but all useful error estimators share some characteristics:

- First of all, error estimation should only take a fraction of the solution process. Clearly, the sum of the time needed to solve the problem with a coarse mesh and the time to evaluate the error of the finite element solution should be less than the time needed to solve the same problem with a much finer mesh. An excep-
tion could be made to this rule if the error estimator gave extremely accurate information about the exact solution. To achieve this kind of computational efficiency, it is desirable that the error estimation procedures only involve local calculations, i.e. that the error estimate at a point in the domain considered be calculated from information pertaining to the neighborhood of that point only. Should error estimation require global calculations, i.e. should the error estimate at a point depend on information pertaining to the whole domain, we would need to make sure that these global calculations are few and do not yield an error estimation operation count that is comparable to the finite element solution operation count for a fine mesh. It should be noted that since in elliptic problems local changes in the loading have a global impact on the solution, the requirement that an error estimator be based on local calculations only is a strict one.

- Second, an error estimator should be rigorous and accurate. For instance, if an error estimator yields bounds on the energy norm of the error, we would expect those bounds to be unconditionally true. Obviously, it would also be desirable that they be as sharp as possible.

- Last, ideally, a good error estimator should be able to evaluate the local error density and not just the overall performance of a finite element solution. This would allow to couple the error estimator to a mesh generator in charge of remeshing those regions of the mesh where the error is the largest while leaving untouched or even derefining those regions of the mesh where an accurate solution has already been obtained. This sort of coupled error estimator and mesh generator can be used (see for instance [19, 17]) to assure that an even error density is obtained over the whole mesh, which means that the best possible use is made of a given number of elements. Also, a good error estimator should be able to tell what component of the stress field is most affected by the error.
Among the different error estimators that have been proposed in the literature, not all of them satisfy these three requirements. Specifically, some types of error estimators only give global estimates of the error and thus violate the third requirement. In this work, we concentrate on one particular category of error estimators: recovery-based error estimators.

1.2 Overview of various approaches to error estimation for elliptic problems

1.2.1 Recovery-based error estimators

The term "recovery-based error estimators" is used to designate all estimators that derive so-called recovered solution fields from the finite element fields without requiring the explicit calculation of an estimate of the error. The estimate of the error is afterwards obtained by subtracting the finite element solution fields from the recovered fields. We obtain a good error estimator if the recovered fields are a close estimate to the exact solution.

In this class of techniques, the recovered fields are calculated from the finite element solution fields by exploiting certain properties of the exact solution that are not directly imposed by the finite element procedures, such as the continuity of the stress and strain fields (see Sections 2.2 and 2.3), or properties of the finite element solution, such as its intrinsic "oscillatory" behavior (see Section 2.4).

These recovery-based methods present several advantages. First, and unlike other methods, they do not require the evaluation of the error through supplementary finite element calculations, which means that error estimation with this type of techniques is likely to require little calculations. As a result, these techniques can be expected to be relatively fast. Second, their conceptual simplicity translates into simple implementation.
On the down side, these techniques rely on little mathematical background. Their study is therefore mostly based on empirical results and the main justification of their use is that they give satisfactory results for many engineering problems. Also, recovery-based error estimators, since they smoothen out discontinuities in the solution, are not appropriate for problems where discontinuities appear, such as hyperbolic problems, interface problems (unless these interfaces are treated specifically).

The most popular recovery-based algorithms are presented in detail in Chapter 2.

1.2.2 Explicit methods

"Explicit methods" are techniques that provide bounds on the error that can be calculated directly from the finite element solution and the problem data without requiring the evaluation of additional quantities. Essentially they consist in bounding the energy norm of the error by a measure of the unbalanced forces and inter-element discontinuities.

Explicit techniques can be fast due to the fact that the bounds are evaluated directly from available information, but they yield only global bounds, and these may not be of much use due to the presence of undetermined constants in the expression of the bounds for the measure of the error.

1.2.3 Implicit methods

Whereas explicit error and recovery-based estimators do not require that an estimate to the error be calculated by an additional finite element solution, implicit error estimators necessitate such calculations. Over clusters of connected elements, the residual problem for the error is solved using finite element methods. The solution of the residual problem can require the choice of new elements and shape functions over these clusters. The unbalanced force, defined as the difference between the actual body force applied to the body and the force field that balances the finite element
stress field, is used as the volumetric load while the boundary loadings are chosen by splitting boundary fluxes to assure well-posedness (see for instance [2]).

Like the recovery-based techniques, implicit methods yield an approximation of the error at every point in the mesh and can therefore provide information on the local accuracy of a finite element solution. When appropriate formulations are employed, they are based on a firm mathematical foundation.

As a pay back for the extra computational cost and more intricate implementation, implicit estimators present several advantages. In particular, they have a more rigorous foundation than recovery-based algorithms and unlike explicit algorithms they yield bounds that do not contain unknown constants and they provide local estimates.

1.3 Objectives

Each category (recovery-based, explicit, implicit) of error estimation techniques has advantages and disadvantages. Through a mathematical analysis of the sampling points used in recovery-based techniques, we intend to show analogies between implicit and recovery-based algorithms. Then we can use this better understanding of existing techniques to develop a new category of error estimators that are rigorous yet relatively inexpensive.

1.4 Notations and classical results in finite element methods

Here, to introduce the notations used in the rest of this work, we recall the basic principles underlying the displacement-based finite element method and some important mathematical results (see also [4, 15, 8]). We restrict our work to the solution of the linearized equations of mechanics by displacement-based methods.
1.4.1 The isoparametric displacement-based finite element method

Let us denote by $\Omega$ a three-dimensional volume of finite measure and $\partial \Omega$ its boundary. This boundary is subdivided into two parts: $\partial \Omega_u$ where Dirichlet boundary conditions ($u = u_d$) are imposed and $\partial \Omega_f$ where Neumann conditions are imposed. Let us define the following sets:

$$
V = \{ f / f_{|\partial \Omega_u} = u_d, f \in H^1(\Omega) \} \quad (1.1)
$$

and

$$
\hat{V} = \{ f / f_{|\partial \Omega_u} = 0, f \in H^1(\Omega) \} \quad (1.2)
$$

The volume itself is subdivided into a collection of $N_h$ finite elements $T_i$, where $i \in \{1, ..., N_h\}$. Let us assume that admissible (i.e., $H^1(\Omega)$ functions) shape functions have been chosen corresponding to the nodes of the finite elements such that in $\Omega$ the finite element displacements can be written in the form

$$
u_h = HU \quad (1.3)
$$

with $H$ the finite element interpolation functions matrix and $U$ the finite element nodal displacement vector. Let us denote by $B$ the engineering strain interpolation functions matrix which results from the differentiation of the shape functions in $H$:

$$
\epsilon_h = BU \quad (1.4)
$$

where $\epsilon$ denotes the engineering strain vector defined as

$$
\epsilon = \begin{pmatrix}
\epsilon_{11} & \epsilon_{22} & \epsilon_{33} & 2\epsilon_{12} & 2\epsilon_{23} & 2\epsilon_{31}
\end{pmatrix}^T = \begin{pmatrix}
\epsilon_{11} & \epsilon_{22} & \epsilon_{33} & \gamma_{12} & \gamma_{23} & \gamma_{31}
\end{pmatrix}^T \quad (1.5)
$$
with the linearized strain tensor defined by
\[
\varepsilon = \begin{pmatrix}
\varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\
\varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\
\varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33}
\end{pmatrix} = \frac{1}{2}(\nabla u + \nabla u^T)
\] (1.6)

In the Galerkin method, in which the same interpolation functions are used for the real and virtual displacements, the discretization of the continuum problem

\[
\text{Find } u \in V / \forall v \in \bar{V} \quad a(u, v) = (f^B, v)
\] (1.7)

\((f^B \text{ includes forces per unit volume and forces per unit surface on } S_f)\) is discretized to

\[
\text{Find } u_h \in \bar{V}_h / \forall v_h \in \bar{V}_h \quad a(u_h + u_0, v_h) = (f^B, v_h)
\] (1.8)

where \(\bar{V}_h\) is the FE subspace of \(\bar{V}\), and \(u_0\) is an arbitrary function in \(V\). Putting the effect of any non-zero imposed displacement boundary conditions on the right side of equations 1.7 and 1.8, we obtain for the continuum:

\[
\text{Find } u \in \bar{V} / \forall v \in \bar{V} \quad a(u, v) = (f, v)
\] (1.9)

and for the discretized problem:

\[
\text{Find } u_h \in \bar{V}_h / \forall v_h \in \bar{V}_h \quad a(u_h, v_h) = (f, v_h)
\] (1.10)

where
\[
\forall b \in \bar{V} \quad (f, b) = (f^B, b) - a(u_0, b)
\] (1.11)

Equations 1.9 and 1.10 yield the orthogonality property of the error:
\[
\forall v_h \in V_h, a(u - u_h, v_h) = 0
\] (1.12)
In matrix form, equation 1.10 results in the usual finite element equation

\[ KU = R \] (1.13)

where the stiffness matrix \( K \) is defined by

\[ K = \int_\Omega B^T C B d\Omega \] (1.14)

and the load vector \( R \) by

\[ R = \int_\Omega H^T f d\Omega \] (1.15)

Once equation 1.13 has been solved for \( U \), the FE engineering strains \( \epsilon = BU \) and stresses \( \tau = CBU \) can be evaluated at any point in \( \Omega \).

### 1.4.2 Convergence results for the finite element method

A priori error estimation is based on the convergence results of the displacement-based finite element method.

Let us denote by \( Q_p \) the finite element vector space on \( \mathbb{R} \) defined by:

\[ Q_p = \text{span}\{x^i y^j z^k, (i, j, k) \in \{0, 1, \ldots, p\}^3\} \] (1.16)

Then the strain energy of the finite element solution converges to the exact strain energy from below, and the convergence rate is given by:

\[ |a(u_h, u_h) - a(u, u)| \leq C(u, E, \nu)h^{2p} \] (1.17)

where \( C(u, E, \nu) \) depends on the actual solution of the problem, on the material properties (Young's modulus \( E \) and Poisson ratio \( \nu \)), but \textit{not} on the typical element size \( h \). This essentially assures us that the finite element stresses and strains converge to the actual stresses and strains with a convergence rate of \( p \).
In practice, we will not observe a convergence rate for the energy larger than $2p$, so that we often simply write

$$|a(u_h, u_h) - a(u, u)| = C(u, E, \nu)h^{2p}$$  \hspace{1cm} (1.18)

Using equation 1.18, we can extrapolate from two finite element solutions obtained from different meshes an estimate of the actual strain energy and therefore we can estimate $a(u_h, u_h) - a(u, u)$ without having to solve the finite element problem on a very fine mesh. From two solutions $u_{h1}$ and $u_{h2}$ obtained on meshes of typical size $h_1$ and $h_2$, we can evaluate $a(u_{h1}, u_{h1})$ and $a(u_{h2}, u_{h2})$. We then know that

$$A = -a(u_{h1}, u_{h1}) + a(u, u) = C(u, E, \nu)h_1^{2p}$$
$$B = -a(u_{h2}, u_{h2}) + a(u, u) = C(u, E, \nu)h_2^{2p}$$  \hspace{1cm} (1.19)

where $A$ and $B$ are two unknowns we want to evaluate.

The quotient $A/B$ can be easily calculated using the right hand side of equation 1.19:

$$A/B = (h_1/h_2)^{2p}$$  \hspace{1cm} (1.20)

Also the difference $A - B$ can be evaluated, because, from the left hand side of equation 1.19, we have

$$A - B = a(u_{h1}, u_{h1}) - a(u_{h2}, u_{h2})$$  \hspace{1cm} (1.21)

Now, noting that $A = \frac{(A-B)}{1-(A/B)}$ and that $B = \frac{(A-B)}{-1+(A/B)}$, we can evaluate $A$ and $B$ from known quantities. $A$ and $B$ indicate the difference between the exact strain energy and the finite element strain energy. We can then evaluate the constant $C$ on the right hand side of equation 1.19 and determine how fine a mesh we need to have the difference between the actual and FE strain energies smaller than a chosen value.

It is well known that the constant $C(u, E, \nu)$ goes to infinity when $\nu$ approaches $1/2$, which accounts for the poor performance of the displacement-based method.
in the analysis of almost incompressible media. In practice, we would use mixed formulations to analyze incompressible media.

1.4.3 Norms used for the assessment of the performance of error estimators

In Section 1.4.2, we have encountered one possible way of measuring the error: the difference between the strain energy of the exact solution and the finite element solution. This is but one of different indicators of how large the error is.

The overall error can also be measured by its global strain energy or equivalently by $a(e_h, e_h)$ which is arguably the “natural” norm for this type of problems.

$$a(e_h, e_h) = \int_{\Omega} (\epsilon^T - U^T B^T) C(\epsilon - BU) d\Omega \quad (1.22)$$

The error at one particular point can be measured by the strain energy density $(\epsilon^T - U^T B^T) C(\epsilon - BU)$ at that point.

The Sobolev norms $L^2(\Omega)$ and $H^1(\Omega)$ can also be used to measure the error. Different error estimation techniques are best studied using different norms.
Chapter 2

Recovery-based error estimators

The idea behind recovery-based error estimators is to use only the finite element derivative field (which is in general discontinuous) to build a so-called "recovered derivative field". This recovered field is obtained by smoothing in one way or another the finite element derivative field. The error in the derivative field is then approximated by the difference between the recovered derivative field and the finite element derivative field.

Different recovery-based error estimators use different smoothing techniques. Even though global smoothing was indeed envisaged by some authors (see [6]), the most common recovery-based error estimators rely on local smoothing techniques because of their much lower cost. The main methods are known as nodal averaging, linear interpolation techniques and least squares fit techniques. These methods are presented in Sections 2.2, 2.3 and 2.4, respectively.

We present these techniques for the 9-node isoparametric displacement-based finite element.
2.1 Mathematical foundation

There is essentially one result that guarantees convergence of the estimated error obtained from a recovery-based error estimator to the actual error under certain circumstances.

In a cartesian coordinate system, let us denote by $\tau$, $\tau_h$, and $\tau_r$ respectively the matrix corresponding to the exact stress tensor, the finite element stress tensor and the recovered stress tensor, whose components are all functions of class $L^2(\Omega)$. Depending on the particular recovery method employed, the recovered stress field $\tau_r$ might be discontinuous along inter-element boundaries, but this does not matter here.

The error in the stress is $\Delta \tau_h = \tau - \tau_h$, its estimate by the recovery-based algorithm is $\Delta \tau_* = \tau - \tau_r$, and $\Delta \tau_r = \tau - \tau_r$ is the error in the recovered field. Let us also denote by $\|\cdot\|$ a norm for $\tau$. For instance, the modified $L^2(\Omega)$ norm defined by

$$
\|\Delta \tau_h\| = \sqrt{\int_\Omega \Delta \tau_h^T C^{-1} \Delta \tau_h d\Omega} = \sqrt{\int_\Omega (\tau^T - U^T B^T C^T) C^{-1} (\tau - C B U) d\Omega} \quad (2.1)
$$

could be used here to measure the actual error, as stated in Section 1.4.3. Then, using the triangle inequality, we can write:

$$
\|\Delta \tau_h\| - \|\Delta \tau_r\| \leq \|\Delta \tau_*\| \leq \|\Delta \tau_h\| + \|\Delta \tau_r\| \quad (2.2)
$$

It immediately follows that

$$
1 - \frac{\|\Delta \tau_r\|}{\|\Delta \tau_h\|} \leq \frac{\|\Delta \tau_*\|}{\|\Delta \tau_h\|} \leq 1 + \frac{\|\Delta \tau_r\|}{\|\Delta \tau_h\|} \quad (2.3)
$$

Hence, the so-called “effectivity ratio” $\frac{\|\Delta \tau_r\|}{\|\Delta \tau_h\|}$, which is a measure of the accuracy of the error estimation, tends to one if the ratio $\frac{\|\Delta \tau_r\|}{\|\Delta \tau_h\|}$ tends to zero as the mesh is refined. Hence, if $\|\Delta \tau_r\| = o(\|\Delta \tau_h\|)$ the effectivity index of the error estimator goes to unity asymptotically. In particular, this result is valid when $\|\Delta \tau_r\|$ converges to zero at a faster rate than $\|\Delta \tau_h\|$. 
A stress component before nodal averaging... ... and after nodal averaging

Figure 2-1: Nodal averaging.

2.2 Nodal averaging

2.2.1 Principle

Nodal averaging is probably the simplest of smoothing methods. In this method, the recovered derivative field at each node of the mesh is obtained by simply averaging the derivative field at the node from all elements connecting to that node. Then the recovered derivative field is obtained by fitting shape functions through these recovered nodal values. Nodal averaging is illustrated in Figure 2-1.

2.2.2 Formal expression

Formally, we can therefore define the recovered value of the derivative field at a node \( n \) in the following way:

\[
\tau_r(n) = \frac{\sum_{N \in K(n)} \tau(n, N)}{\text{Card}(K(n))}
\] (2.4)
where $K(n)$ is the set of all the elements to which node $n$ belongs, $\text{Card}(K(n))$ is the cardinality of $K(n)$, and $\tau(n, N)$ is the value of the finite element stresses at node $n$ as seen from element $N$.

The values of the finite element stresses at a node $n$, as seen from element number $N$, are obtained from the simple formula

$$\tau(n, N) = C(N)B(n, N)U(N)$$

(2.5)

where $\tau(n, N)$ is the stress vector at point $n$ as seen from element $N$, $C(N)$ is the material law for element $N$, $B(n, N)$ is the $B$ matrix for element $N$ evaluated at node $n$, and $U(N)$ is the vector of nodal displacements pertaining to element $N$.

The recovered stresses within the elements are then interpolated from the nodal recovered values by using the displacement shape functions: at any point $x$ in element $N$, we use

$$\tau_r(x) = \sum_{n \in \text{Nodes}(N)} \tau_r(n)h(N, n, x)$$

(2.6)

where $\text{Nodes}(N)$ denotes the set of nodes belonging to element $N$, and $h(N, n, x)$ denotes the shape function for node $n$ in element $N$ evaluated at point $x$.

2.2.3 Analysis

Even though this smoothing technique could be used in general with any derivative field (i.e. not necessarily a field yielded by a finite element code), it makes sense to use it with an initial derivative field generated by a displacement-based finite element method. As a matter of fact, the derivative field yielded by such a method is discontinuous, unlike the actual solution to the mathematical problem. It is easily seen that by construction the recovered derivative field is of class $C^0$, owing to the fact that the field is the finite linear combination of shape functions which are themselves of class $C^0$. Besides, the displacement-based finite element method is well known to yield an "oscillatory" solution about the actual solution, as illustrated in Figure 2-2.
It therefore makes sense to use the average of the stresses in the elements connected at a node as an improved value for the stresses at that node.

2.3 Linear interpolation techniques: ADINA estimator, Hinton Campbell estimator

2.3.1 Principle

Unlike the nodal averaging method which requires the evaluation of the initial (finite element) stress field at the nodes, linear interpolation techniques such as the one implemented in the ADINA code [1] and the Hinton Campbell method [6] derive a recovered stress field from Gauss points using linear interpolations. The algorithm in the ADINA code uses a bilinear interpolation from the closest four 3x3 Gauss points...
in each element and then averages the values obtained from different elements to calculate the nodal values of the recovered stress field. In the local Hinton Campbell algorithm, bilinear polynomials in the local (r,s) coordinates are passed through the stresses at the 2x2 Gauss points of every element, yielding nodal values which are averaged to obtain the recovered nodal stresses.

2.3.2 Formal expression for the ADINA error estimator

Like in the nodal averaging method, the recovered derivative field at any point is calculated from recovered nodal values, using the same shape functions that are used to interpolate displacements and coordinates:

\[
\tau_r(x) = \sum_{n \in \text{Nodes}(N)} \tau_r(n) h(N, n, x) \tag{2.7}
\]

The recovery method used in the ADINA code however differs from the nodal averaging technique in that the recovered nodal values of the derivative field are obtained by a bilinear extrapolation from the closest four 3x3 Gauss points in each element, followed by averaging. The averaging is carried out over the patch of elements to which node n belongs, just like in the nodal averaging method:

\[
\tau_r(n) = \frac{\sum_{N \in K(n)} \tau(n, N)}{\text{Card}(K(n))} \tag{2.8}
\]

Let us show how one component of the recovered stress tensor is evaluated in the method described above: the \(i^{th}\) component of the recovered stress field for node n as seen from element N is

\[
\tau_{ri}(n, N) = P(r(n, N), s(n, N)) a_i \tag{2.9}
\]

where \(P(r(n, N), s(n, N))\) is the row vector

\[
P(r, s) = \begin{pmatrix} 1 & r & s & rs \end{pmatrix} \tag{2.10}
\]
evaluated at the local coordinates of point \( n \) in element \( N \), and where \( a_i \) is the column vector
\[
a_i = (a_{i1} \ a_{i2} \ a_{i3} \ a_{i4})^T
\] (2.11)
which is obtained by inverting the bilinear interpolation equation
\[
\begin{pmatrix}
\tau_{i1} \\
\tau_{i2} \\
\tau_{i3} \\
\tau_{i4}
\end{pmatrix}
= P \ a_i
\] (2.12)
in which \( P \) is a square matrix built from the evaluation of vector \( P \) at the four 3x3 Gauss points that are closest to node \( n \) in element \( N \):
\[
P = \begin{pmatrix}
\leftarrow P(r_1, s_1) \rightarrow \\
\leftarrow P(r_2, s_2) \rightarrow \\
\leftarrow P(r_3, s_3) \rightarrow \\
\leftarrow P(r_4, s_4) \rightarrow
\end{pmatrix}
\] (2.13)
\((r_i, s_i)\) are the local coordinates of the four Gauss points closest to node \( n \) in element \( N \).

It should be noted that while a different vector \( a_i \) has to be evaluated for each component \( \tau_i \), the \( P \) matrix is the same for all components of the stress field, for any given pair \((n, N)\). This is an advantage of using interpolation in the local coordinate system rather than in the global coordinate system. One advantage of using the 3x3 Gauss points is that the \( B \) matrices are already evaluated at those points in the finite element method if full integration was used to evaluate the element stiffness matrices.

Once the \( a_i \) vector has been evaluated by solving equation 2.12, we can immediately obtain the nodal recovered value of the stress field at node \( n \) as seen from element \( N \) by using equation 2.9, and finally the recovered stress field at any point in the mesh is obtained through equations 2.8 and 2.7.
2.3.3 Formal expressions for the Hinton Campbell method

In their original paper, Hinton and Campbell [6] proposed two recovery methods. The first one consists of the least squares fitting of a continuous, piecewise bilinear in the local coordinates function to all the 2x2 Gauss points of the structure (global least squares fit), while the second one, even though presented in [6] as a least squares fit method is in fact really a bilinear extrapolation method in each element followed by averaging. It is the latter we are focusing on here, the former being too computationally expensive for systems of practical interest (it can indeed be shown that the cost of that error estimator grows with the square of the number of elements in the mesh employed, which is comparable to the cost of running the same problem with a finer mesh).

Similar to the nodal averaging and ADINA methods, the recovered stress field at any point in the domain will be determined from nodal recovered values using equation 2.6 and equation 2.8, and all we need to do is show how the \( \tau_r(n, N) \) are evaluated. First, let us determine the nodal recovered stresses at node \( n \) as seen from element \( N \). From the fact that the recovered field is a bilinear (in the \((r, s)\) coordinates) extrapolation from the 2x2 Gauss points, we immediately get that

\[
\tau_h(r_k, s_k, N) = \sum_{i=1}^{4} h_i(r_k, s_k) \tau_r(i, N) \quad \text{for} \quad k \in \{1, 2, 3, 4\} \tag{2.14}
\]

where \((r_k, s_k)\) (with \(k \in \{1, 2, 3, 4\}\)) are the 2x2 local coordinates of the Gauss points of element \( N \) (the classical numbering pattern for Gauss points and nodes is recalled on Figure 2-3).

Replacing the \( h_i(r_k, s_k) \)'s by their values and solving for the right hand side vector elements, we obtain for the \( i^{\text{th}} \) direction:
Figure 2-3: Classical numbering of nodes and Gauss points.

\[
\begin{pmatrix}
\tau_{ri}(1, N) \\
\tau_{ri}(2, N) \\
\tau_{ri}(3, N) \\
\tau_{ri}(4, N)
\end{pmatrix} =
\begin{pmatrix}
1 + \sqrt{3}/2 & -1/2 & 1 - \sqrt{3}/2 & -1/2 \\
-1/2 & 1 + \sqrt{3}/2 & -1/2 & 1 - \sqrt{3}/2 \\
1 - \sqrt{3}/2 & -1/2 & 1 + \sqrt{3}/2 & -1/2 \\
-1/2 & 1 - \sqrt{3}/2 & -1/2 & 1 + \sqrt{3}/2
\end{pmatrix}
\begin{pmatrix}
\tau_{hi}(r_1, s_1, N) \\
\tau_{hi}(r_2, s_2, N) \\
\tau_{hi}(r_3, s_3, N) \\
\tau_{hi}(r_4, s_4, N)
\end{pmatrix}
\]  

(2.15)

To evaluate the nodal recovered stresses as seen from element \(N\) to nodes 5 to 9, we use the following equalities which result immediately from the fact that we are using an extrapolation by bilinear functions:

\[
\begin{align*}
\tau_{ri}(5, N) &= 1/2(\tau_{ri}(1, N) + \tau_{ri}(2, N)) \\
\tau_{ri}(6, N) &= 1/2(\tau_{ri}(2, N) + \tau_{ri}(3, N)) \\
\tau_{ri}(7, N) &= 1/2(\tau_{ri}(3, N) + \tau_{ri}(4, N)) \\
\tau_{ri}(8, N) &= 1/2(\tau_{ri}(1, N) + \tau_{ri}(2, N)) \\
\tau_{ri}(9, N) &= 1/4(\tau_{ri}(1, N) + \tau_{ri}(2, N) + \tau_{ri}(3, N) + \tau_{ri}(4, N))
\end{align*}
\]  

(2.16)
These relations along with equation 2.15 determine the nodal recovered stresses at all nodes as seen from all elements, and averaging is used to obtain the recovered nodal values (equation 2.8) used by the Hinton Campbell error estimator.

2.3.4 Analysis of the Hinton Campbell method: higher-order-accuracy points

The Hinton Campbell method is based on the existence of higher-order-accuracy points which are studied in depth in Chapter 3: it is hoped that if higher order convergence occurs at the points from which the stresses are extrapolated then the stresses recovered by the Hinton Campbell method will also show higher order convergence.

2.4 Least squares fit technique: the Zienkiewicz Zhu estimator

The Zienkiewicz Zhu error estimator [18, 19], or ZZ estimator, is one of the most researched error estimators available. Similarly to the Hinton Campbell and ADINA error estimators, the ZZ estimator is a “patch-based” error estimator, in the sense that the stresses at a node are recovered from patches of elements surrounding the node in question. But whereas the implementation of the Hinton Campbell and ADINA estimators involves element level computations only, the ZZ estimator requires patch level computations. This makes the “book keeping” more complicated as well as more expensive in terms of the data structure used.

2.4.1 Principle

The idea behind the ZZ error estimator is to obtain recovered nodal stresses from a least squares fit to the lower order Gauss points of patches of elements (e.g. for the 9 node element of interest, the 2 by 2 Gauss points). Boundary nodes require special treatment.
2.4.2 Formal expressions for the Zienkiewicz Zhu method

The ZZ algorithm assumes that the mesh employed is reasonably fine, so that all elements in the mesh either

- have no side on the boundary of the computational domain, or
- have one side on the boundary of the computational domain, or
- have two consecutive sides on the boundary of the computational domain

This means that a number of mesh patterns are not allowed, as illustrated on Figure 2-4.

Once again, the recovered stress field at any point in the domain will be determined from nodal recovered values using equation 2.6, so that we only need to show how these nodal recovered values are computed in the ZZ method.
Figure 2-5: Node numbers for the problem illustrating recovery patches

Corner nodes are indicated by larger numbers

At this point, we need to introduce some notation to handle patches. So, to any node \( n \) which is located at the corner of an element \( N \), we associate the patch of elements \( \Omega_n \) defined as the set of all the elements of the mesh which share node \( n \).

To each patch \( \Omega_n \) is associated a set \( P_{\Omega_n} \) of nodes determined by the following algorithm: we deal with each corner node \( n \) in turn.

1. If the corner node \( n \) is not on the boundary of the computational domain, we include in \( P_{\Omega_n} \) all nodes that are in \( \Omega_n \) but not on its boundary.
Figure 2-6: Recovery patches

$P_{0_{12}}$ is used to recover stresses to nodes 9 through 12, 41 through 46 and 59 through 64.
Figure 2-7: Recovery patches

$P_{\Omega}$ is used to recover stresses to nodes 1, 2, 7, 8, 23, 24, 29 through 34 and 39 through 44. The recovered stresses at nodes 41 through 44 receive contributions from $P_{\Omega_1}$ and $P_{\Omega_{12}}$. 
2. If the patch $\Omega_n$ associated with the corner node $n$ consists of only one element, node $n$ and the two non corner nodes of $\Omega_n$ that are on the boundary of the computational domain are included in the set $P_{\Omega_{n'}}$ where $n'$ is the corner node opposite node $n$ in the element.

3. If the patch $\Omega_n$ associated with the corner node $n$ consists of two elements, node $n$ and the two other nodes of $\Omega_n$ that are on the boundary of the computational domain are included in the set $P_{\Omega_{n'}}$ where $n'$ is the node such that $nn'$ is the element side shared by the two elements in $\Omega_n$.

This intricate definition for the sets $P_{\Omega_n}$ is needed because patches that consist of only one or two elements cannot be used to recover stresses. This is because they have less than 9 lower order Gauss points and therefore a least squares fit with 9 degrees of freedom would be undetermined.

Patches $\Omega_n$ such that $P_{\Omega_n}$ is empty are not used in the ZZ algorithm. We call recovery patches those patches $\Omega_n$ for which $P_{\Omega_n}$ is not empty.

Finally, to any recovery patch $\Omega_n$, we also associate the set $R_{\Omega_n}$ of all the lower order Gauss points of all elements in $\Omega_n$. These definitions assure that all $R_{\Omega_n}$ sets contain more than 9 points (see Figures 2-5, 2-6 and 2-7).

At the recovery patch level, each component of the stress tensor is dealt with in turn. So, say we are interested in the $i^{th}$ component of the stress tensor. Over a recovery patch $\Omega_n$, we denote by $\tau_{*i}(x,y,z)$ the biquadratic polynomial obtained by a least squares fit to the finite element stresses at the points in $R_{\Omega_n}$. Writing

$$\tau_{*i}(x,y) = a_{\Omega_n}X^T$$

$$= (a_{0,\Omega_n} \ldots a_{8,\Omega_n}) (1 \ x \ y \ x^2 \ xy \ y^2 \ x^2y \ xy^2 \ x^2y^2)^T$$

we obtain the unknown coefficients in $a_{\Omega_n}$ from the least squares fit procedure.
\[ \forall j \in \{0, 1, \ldots, 8\}, \quad \frac{d}{da_j, \Omega_n} \left( \sum_{k \in R_{\Omega_n}} (\tau_{i}(r_k, \xi_k) - \tau_{hi}(r_k, \xi_k))^2 \right) = 0 \]  

(2.19)

which in matrix form gives:

\[ a_{\Omega_n} = \left( \sum_{k \in R_{\Omega_n}} X(x_k, y_k)X(x_k, y_k)^T \right)^{-1} \sum_{k \in R_{\Omega_n}} X(x_k, y_k)\tau_{hi}(x_k, y_k) \]  

(2.20)

where \( \tau_{hi}(x_k, y_k) \) is the \( i \)th component of the finite element stress tensor at point \( k \).

The nodal stress recovered to a node \( j \in \Omega_n \) is then

\[ \tau_{*i}(j, \Omega_n) = a_{\Omega_n} \left( \begin{array}{cccc} x_j & y_j & x_j^2 & x_jy_j & y_j^2 & x_j^2y_j & x_jy_j^2 & x_j^2y_j^2 \end{array} \right)^T \]  

(2.21)

where \( x_j \) and \( y_j \) are the global coordinates of point \( j \).

Unlike in the ADINA and Hinton Campbell methods, in the ZZ estimator nodal recovered values are not obtained at the element level, but at the patch level, so that equation 2.8 transforms into

\[ \tau_r(j) = \frac{\sum_{n/j \in P_{\Omega_n}} \tau_{*i}(j, \Omega_n)}{\text{Card}(n/j \in P_{\Omega_n})} \]  

(2.22)

Using equation 2.6, we can obtain the stresses at any point in the computational domain. The ZZ method can be extended in concept straightforwardly to three-dimensional problems and other types of elements but the algorithm gets complicated.

### 2.4.3 Analysis of the Zienkiewicz Zhu method

The ZZ estimator carries a number of disadvantages:

1. Like the Hinton Campbell method, the ZZ estimator uses lower order Gauss point values of the stresses as a starting point. If full integration is used, stresses at these points are not immediately available and they need to be evaluated from the displacements. This is particularly a disadvantage in inelastic analysis.
2. As was mentioned earlier, the ZZ estimator requires the implementation of data structures to relate nodes to the elements they belong to.

3. The specific treatment of elements on the boundary is only justified by practical, numerical reasons: If we had defined patches associated with nodes located on the boundary of the computational domain, such patches could have had only 4 or 8 lower order Gauss points in them, respectively when the patch in question consists of 1 or 2 elements. If a patch has less than 9 lower order Gauss points in it, the least squares fit procedure is undetermined because the matrix \( \sum_{k \in R_0} X(x_k, y_k)X(x_k, y_k)^T \) is then singular. In this case, there exists an infinite number of biquadratic polynomials that exactly interpolate 8 or less values. Indeed, Zienkiewicz and Zhu themselves in [18] argue that boundary elements can be dealt with in different ways.

4. In three dimensional problems, patches can consist of dozens of elements, making the least squares fit a rather expensive method. As a matter of fact, the ZZ algorithm is all the more expensive as patches consisting of many elements are used, as shown by equation 2.20.

5. Finally, compared to explicit and implicit error estimators, recovery based methods are not "physically" motivated. In fact, they are more smoothing techniques than actual error estimation methods. It is only the choice of the lower order Gauss points as the recovery points that justifies the fact that higher order convergence can be expected from these recovery based methods.

On the other hand, it has been shown in [18, 19] that the ZZ estimator performs much better than the Hinton Campbell estimator, and in particular fourth order convergence has been observed for certain problems solved with the 9-node quadrilateral element. The finite element stresses show only second order convergence, and the third order convergence observed at the lower order Gauss points has been coined "superconvergence", prompting the use of the term "ultraconvergence" to denote the fourth order convergence observed in [18].

34
Chapter 3

Higher-order-accuracy points and error assessment

3.1 Introduction

Much attention has been given lately to the Zienkiewicz Zhu algorithm (see [18, 19]) presented in Section 2.4. The method has been proven to be effective for various applications (see [19]). However, the Zienkiewicz Zhu algorithm is not ideal with respect to some issues. Firstly, recovery-based techniques, because they are patch-based, smoothen out the effects of material discontinuities on the solution unless special procedures are used. Secondly, the program data structures required by patch-based algorithm when employed in engineering practice are complicated. Thirdly, patches can include a relatively large number of elements especially if a three-dimensional analysis is being carried. Fourthly, its mathematical basis is not strong. These four disadvantages would not be present if an element-based error estimator were available. Finally, recovery-based algorithms require the calculation of stresses at certain sampling points that are not the points used for the evaluation of the stiffness matrix (assuming that "full" numerical interpolation is employed [4]).

Understandably, the choice of appropriate sampling points is a major issue in the development of a recovery-based error estimator. For the smoothing algorithm to
yield a recovered stress field that is significantly closer to the exact solution than the finite element stress field, sampling points need be chosen that provide higher order convergence [11]. Frequently, the algorithm developed by Barlow [3] is used to select these points, but these points have been the subject of a debate (see [13, 10, 14]) mainly due to the fact that Barlow's algorithm only yields approximate locations for the higher-order-accuracy points. MacKinnon and Carey have proposed an expression for the location of the higher order accuracy points when solving the Laplace equation (see [9]). However, the results given are rather restrictive, since even for the one-dimensional scalar problem of a bar in tension with varying material properties and/or cross-sectional area, an expression for the location of the higher-order-accuracy points is not yet available.

The first objective in this chapter is to show how the existence and location of higher-order-accuracy points can be systematically established. Further, we set out to show under what conditions the higher-order-accuracy points coincide with the lower-order Gauss points (MacKinnon and Carey proved this result for the Laplace equation) and thereby clear up the controversy on these points. The second objective is to show how these results on higher-order-accuracy points can be used to develop an error estimator that is element-based and that does not require the evaluation of the finite element stresses at sampling points. The performance of this new error estimator is then tested by applying it to some sample problems. We give results for a one-dimensional scalar problem solved with isoparametric elements and discuss how these results can be extended to more challenging problems, such as the Poisson equation solved over two- or three-dimensional domains. In our analyses, we allow for element distortions and variable material properties.

### 3.2 Higher-order-accuracy points

The objective in this section is to derive the higher-order-accuracy points. We consider one-dimensional problems.
3.2.1 Earlier work

The notion that the finite element method yields strains that are more accurate at certain points than others is the foundation for recovery-based error estimators, as was shown in Section 2.1 (see also [2]).

We shall from here on use the following definition: in an $N$ node element, the higher-order-accuracy points are defined as the points where the finite element strains are equal to the exact strains whenever the exact displacements are polynomials of order $N$ in the local coordinates. Note that such points may not always exist.

Since the finite element strain energy converges to the exact strain energy at a rate $2(N - 1)$ the finite element strains converge to the exact strains at a rate $N - 1$ on average; hence it is said that a higher order of convergence is observed at the higher-order-accuracy points.

The expression "Barlow points" (as well as "superconvergent points") has been used since Barlow’s first paper on the subject in 1976 [3] to denote two different items. On the one hand, the term was used to denote points of higher order accuracy in general, and on the other hand the term was used to denote the location of the points predicted by a method proposed by Barlow in his original paper.

In this work, we call "Barlow points" the points determined by the method proposed by Barlow, whereas we call "higher-order-accuracy points" the higher-order-accuracy points defined above.

This distinction is necessary because the method proposed by Barlow is in general only approximate in giving the higher-order-accuracy points because Barlow makes the assumption that at certain points higher-order displacement patterns can be exactly represented by lower order shape functions [3]. A notable fact is that the local coordinates of the Barlow points are independent of the material properties and element distortions.
MacKinnon and Carey showed that in an undistorted element the higher-order-accuracy points for the Laplace equation are located at the lower order Gauss points [9]. They also showed that this is true for the Poisson equation in two and three dimensions. However, the Barlow points are located at the lower order Gauss points only when certain elements are used. For instance, for the Laplace equation in 1D, the Barlow points are not located at the Gauss points for elements with more than 4 nodes.

Using numerical studies, some authors [19] have concentrated on lower order elements and confirmed that higher-order accuracy is obtained at the lower order Gauss points. But because the term “Barlow points” was used to denote both the higher-order-accuracy points and the points calculated by Barlow’s method, there has been some confusion as to where the higher-order-accuracy points are located in the general case (varying material properties, etc). A technique developed by MacNeal to locate the higher-order-accuracy points and which confirms Barlow’s results has fueled the controversy over higher-order-accuracy points. Even in the case when constant material properties are used, it is still a debated question [13, 14, 10] whether the higher-order-accuracy points coincide for elements of any order with the lower order Gauss points despite the proof by MacKinnon and Carey [9] of this result for this specific simple case. In the more complex case of non-constant material properties and cross-sectional area, even less is known, and some authors [14] have even questioned the very existence of predictable higher-order-accuracy points in this case.

We present in this section a proof of the existence of higher-order-accuracy points in the general (varying material property and varying cross-sectional area) one-dimensional case based on element orthogonal displacement patterns, and show that the higher-order-accuracy points in 1D for undistorted meshes coincide with lower order Gauss points when the material property and cross-sectional area are constant.

In the past, element orthogonal displacement patterns were used in the development of finite elements by Bergan (see [5]).
3.2.2 Bar with distributed load

Let us consider a one-dimensional problem involving a bar of varying Young's modulus $E(x)$ and cross-sectional area $A(x)$ subjected to a loading force per unit length $f(x)$, and modeled using (possibly) distorted $N$-node elements ($N > 1$). The exact solution to the problem $u$ satisfies the equation

$$\frac{d}{dx}(E(x)A(x)\frac{du}{dx}) = -f(x) \quad (3.1)$$

in the domain $\Omega = ]L, R[$, plus boundary conditions at the two ends of the domain (see [15]). Possible boundary conditions are:

1. Essential boundary condition: the boundary displacement is known, the boundary force is unknown.
2. Natural boundary condition: the boundary displacement is unknown, the boundary force is known,

For well-posedness, at least one of the two boundary conditions has to be essential. We will always assume here that the boundary condition at $x = L$ is essential. The other boundary condition (at $x = R$) can be either natural or essential.

For any element $M$, we can define the element energy scalar product for the derivatives of the displacement field [4]:

$$a^M_h(\frac{du}{dr}, \frac{dv}{dr}) = \int_M E A \frac{du}{dr} \frac{dv}{dr} dr \quad (3.2)$$

where isoparametric elements are used, with $r$ being the natural coordinate for element $M$, and the global scalar product for the problem:

$$a(\frac{du}{dx}, \frac{dv}{dx}) = \int_\Omega E A \frac{du}{dx} \frac{dv}{dx} dx \quad (3.3)$$

We assume that in every element the usual polynomial shape functions (denoted by $h_1$ to $h_N$) for isoparametric displacement-based finite elements are employed so
that
\[ \text{Span}(h_1(r), \ldots, h_N(r)) = \text{Span}(1, r, \ldots, r^{N-1}) \] (3.4)

Let us define \( T_0^M(r) = 1 \) and \( T_1^M(r) = r \). Then we can show the following result:

For any element \( M \), there exists a unique set \( T^M \) of polynomials \( T_k^M \) \((T_k^M = a_{kk}^M r^k + \ldots + a_{k1}^M r + a_{k0}^M)\) of order \( k \) \((k \geq 2)\) in the local coordinates, such that these polynomials satisfy:

1. For all \( k > 1 \), the leading coefficient \( a_{kk}^M \) of \( T_k^M \) is 1.
2. For all \( k > 1 \), the trailing coefficient \( a_{k0}^M \) of \( T_k^M \) is 0.
3. 
   \[ \text{if } k > 1 \quad a_{hk}^M \left( \frac{dT_k^M}{dr}, \frac{dT_k^M}{dr} \right) = 0 \quad \text{for } j=0, \ldots, k-1 \] (3.5)

Proof of the existence of the set \( T^M \): The polynomials \( T_k^M \) can be constructed by recurrence using the Gram-Schmidt procedure:

\[ T_{m+1}^M(r) = r^{m+1} - \sum_{i=1}^{m} a_{hi}^M \left( \frac{dT_i^M}{dr}, \frac{dT_i^M}{dr} \right) T_i^M(r), \quad m \geq 1 \] (3.6)

We obtain a set \( T^M \) that satisfies the three conditions stated above.

Proof of the uniqueness of the set \( T^M \): Suppose there exists a second set \( \tilde{T}^M \) of polynomials \( \tilde{T}_k^M \) that satisfy the conditions above. If \( \tilde{T}^M \) differs from \( T^M \), there exists at least one polynomial \( \tilde{T}_k^M \) that is not equal to \( T_k^M \). Let \( w^M \) \((w^M > 1)\) be the smallest value of \( k \) such that \( \tilde{T}_k^M \) is not equal to \( T_k^M \).

Since both \( \tilde{T}_w^M \) and \( T_w^M \) have 1 as their leading coefficient, \( T_w^M - \tilde{T}_w^M \) is a polynomial of order at most \((w^M - 1)\), and its derivative \( f_w^M \) is a polynomial of order at most \((w^M - 2)\) which is orthogonal to all polynomials of order \((w^M - 2)\), including itself. Since \( a_h^M(.,,.) \) is a scalar product, it is a definite form, i.e. the only element with a zero norm is 0. Hence \( f_w^M = 0 \) and \( T_w^M - \tilde{T}_w^M \) is a constant polynomial \( c_w^M \).
In turn, this implies that $T_{wM}^M$ and $T_{wM}^M$ only differ in their trailing coefficient. But we have assumed that both polynomials have the same trailing coefficient. This implies that the constant $c_{wM}$ is zero, and hence $T_{wM}^M = T_{wM}^M$.

We can now determine the location of higher-order-accuracy points in a specific element $M$ for this problem.

**Undistorted elements, constant Young’s modulus and cross-sectional area**

First, let us consider the case for which $E A_r^d$ is constant over every element $M$ in the mesh. In this case, the polynomials $T_k^M$ (where $k \geq 2$) in the set $T^M$ take on a particular form:

$$T_k^M(r) = \frac{k2^k((k - 1)!)^2}{(2k - 2)!} \int_0^r P_{k-1}(\xi)d\xi$$  \hspace{1cm} (3.7)

where $P_k$ is the $k^{th}$ Legendre polynomial.

Proof: This result is obtained from the following two standard properties of the Legendre polynomials [7]:

1. $$\int_{-1}^1 P_i(r)P_j(r)dr = \frac{2\delta_{ij}}{2i + 1}$$  \hspace{1cm} (3.8)

2. $$P_i(r) = \frac{(2i)!}{2^i(i!)^2}r^i + c(r^i)$$  \hspace{1cm} (3.9)

where $\delta_{ij}$ denotes the Kronecker delta and $c(r^i)$ is a polynomial of order lower than $i$. Uniqueness of the set $T^M$ allows us to conclude that $T_k^M$ is given by equation 3.7. In
this case the polynomials in the set $T^M$ are independent of the element $M$. We have:

$$
\begin{align*}
T_0^M(r) &= 1 \\
T_1^M(r) &= r \\
T_2^M(r) &= r^2 \\
T_3^M(r) &= r^3 - r \\
T_4^M(r) &= r^4 - 6/5r^2 \\
T_5^M(r) &= r^5 - 10/7r^3 + 3/7r \\
T_6^M(r) &= r^6 - 5/3r^4 + 5/7r^2 \\
T_7^M(r) &= r^7 - 63/33r^5 + 35/33r^3 - 5/33r
\end{align*}
$$

For $j \geq 2$, $T_j^M(r)$ is orthogonal to $r$ in the sense of the element scalar product defined above, which immediately implies that $T_j^M(-1) = T_j^M(1)$. Let us define the set $S_j^M$ of polynomials $S_j^M$ defined over the whole computational domain $\Omega$ such that $S_j^M(r) = T_j^M(r) - T_j^M(1) = T_j^M(r) - T_j^M(-1)$, $j \geq 2$, over element $M$ and $S_j^M$ equals zero everywhere else. We define $S_1^M$ to be linear with a slope of 1 over element $M$, constant elsewhere, zero at $x = L$ and continuous.

It is immediately seen that if $1 < j < N$ (where $N$ still denotes the number of nodes per element), $S_j^M$ is an element of the finite element space, whereas if $j \geq N$, $S_j^M$ is orthogonal to the finite element space in the sense of the global scalar product $a(\cdot, \cdot)$ (cf equation 3.3). If the boundary condition at $x = R$ is natural, then $S_1^M$ is also in the finite element space.

We define $S_0^M$ to be equal to 1 over element $M$ and 0 elsewhere.

Let us assume that the exact solution (i.e. the mathematical solution to equation 3.1) over element $M$ is analytic in the local coordinate (i.e. it can be developed in the form of a Taylor expansion of $r$ over element $M$). Then, because the set $S^M$ is a basis of the space of polynomials over element $M$, there is a unique set $a^M$ of
coefficients $a^M_k (k \geq 0)$ such that, over element $M$,

$$u(r) = \sum_{i=0}^{\infty} a^M_i S^M_i(r) \quad (3.11)$$

Let us develop the finite element solution obtained from the solution of this problem with $N$-nodded elements on the same basis:

$$u_h(r) = \sum_{i=0}^{N-1} \alpha^M_i S^M_i(r) \quad (3.12)$$

We also have that, for any virtual displacement $v$ in the finite element space,

$$a(\frac{du_h}{dr}, \frac{dv}{dr}) = \int f v = a(\frac{du}{dr}, \frac{dv}{dr}) \quad (3.13)$$

where $a(.,.)$ is the global scalar product. With $v$ equal to, in turn $S^M_2, S^M_3, ..., S^M_{N-1}$, we obtain that, for any $i > 1$, $\alpha^M_i = a^M_i$. If one of the boundary conditions is natural (see 3-1), then $S^M_1$ is in the finite element space and with the same arguments as above, we obtain that $a^M_1 = \alpha^M_1$. Hence

$$u_h(r) = \alpha^M_0 + \sum_{i=1}^{N-1} a^M_i S^M_i(r) \quad (3.14)$$

We can also show that $\alpha^M_0$ is equal to $a^M_0$: since the exact solution and the finite element solutions are continuous, the error is continuous, hence we must have, for any $1 \leq M < P$,

$$(a^M_0 - \alpha^M_0) + \sum_{i=N}^{\infty} (a^M_i - \alpha^M_i) S^M_i(1) = (a^{M+1}_0 - \alpha^{M+1}_0) + \sum_{i=N}^{\infty} (a^{M+1}_i - \alpha^{M+1}_i) S^{M+1}_i(-1) \quad (3.15)$$

which implies

$$(a^M_0 - \alpha^M_0) = (a^{M+1}_0 - \alpha^{M+1}_0) \quad (3.16)$$

also we know that the error is zero at the essential boundary condition, hence

$$(a^M_0 - \alpha^M_0) + \sum_{i=N}^{\infty} (a^1_i - \alpha^1_i) S^1_i(-1) = 0 \quad (3.17)$$
Figure 3-1: $S_1^M$ function is in FE space when one of the boundary conditions is natural ($M = 2$)
which implies that \( a_0^1 = a_0^1 \). Using equation 3.16, we conclude that in any element \( M \) the error can be decomposed in the form

\[
e_h(r) = \sum_{i=N}^{\infty} (a_i^M - \alpha_i^M) S_i^M(r)
\] (3.18)

Over element \( M \), the error in the derivative \( \frac{de_h}{dr} \) then satisfies:

\[
\frac{d(e_h)}{dr} = \sum_{i=N}^{\infty} (a_i^M - \alpha_i^M) \frac{dS_i^M}{dr} = \sum_{i=N}^{\infty} (a_i^M - \alpha_i^M) \frac{dT_i^M}{dr}
\] (3.19)

In element \( M \), the accuracy of the derivative of the finite element solution is therefore one higher order at the zeros of \( \frac{dT_i^M}{dr} \). \textbf{Conclusion:} For an undistorted \( N \)-node element used to model a bar of constant Young’s modulus and constant cross-sectional area, Higher-Order-Accuracy Points are located at the zeros of \( \frac{dT_i^M}{dr} \), i.e. at the lower order Gauss points of the element, because of equation 3.7. Since it was shown that \( T_N \) is unique up to a multiplicative and an additive constant, at no point other than the zeros of \( \frac{dT_i^M}{dr} \) is higher order accuracy achieved.

As a side result, we note that equation 3.18 implies that the error in the displacements is zero at the end points of all elements, a classic result [4].

\textbf{Physical interpretation:} Of all functions that are supported by element \( M \), \( S_N^M \) is the polynomial of the lowest order that is orthogonal to the finite element space. \( S_N^M \) is therefore the first “hidden” displacement pattern: it is the lowest order displacement pattern that cannot be picked up by the \( N \) node element. However, at the points where \( \frac{dS_N^M}{dr} = 0 \), this displacement pattern does not create any strain, and therefore we obtain the same accuracy in the strain at these points whether we include this hidden displacement pattern or not.

It can therefore be seen that whereas implicit algorithms explore the functional space orthogonal to the finite element space, recovery-based algorithms are based on the use of the zeros of the derivative of the lower order polynomial in this orthogonal
Elements $M$ and $M+1$ are not distorted ($M=2$).

Let us now consider the case when both boundary conditions are essential. In this case, $S^M_1$ is not an element of the finite element space, so that we cannot use it as our test function in equation 3.13. However, we still have, over element $M$

$$
e_h = (a_0^M - \alpha_0^M)S^M_0 + (a_1^M - \alpha_1^M)S^M_1 + \sum_{i=N}^{\infty}(a_i^M - \alpha_i^M)S^M_i \quad (3.20)$$

For $1 \leq M < P$ (where $P$ is the number of elements), we can construct a test function $v_M$ that has a slope of 1 in the local coordinate over element $M$, a slope of $-1$ in the local coordinate over element $M+1$, is continuous, and satisfies the boundary conditions (see Figure 3-2). Using the orthogonality equation 3.13 with $v_M$ as our
test function \( v \),
\[
\alpha \left( \frac{d e_h}{d r} , \frac{d v_M}{d r} \right) = 0
\]
we obtain:
\[
\int_M \varepsilon A \frac{d e_h}{d r} (1) \frac{d r}{d x} \, d r + \int_{M+1} \varepsilon A \frac{d e_h}{d r} (-1) \frac{d r}{d x} \, d r = 0
\]
and then
\[
(a_1^M - \alpha_1^M) \int_M \varepsilon A \frac{d r}{d x} \, d r = (a_1^{M+1} - \alpha_1^{M+1}) \int_{M+1} \varepsilon A \frac{d r}{d x} \, d r
\]
(3.23)
We define \( d_M = \frac{\int_{M+1} \varepsilon A \frac{d r}{d x} \, d r}{\int_M \varepsilon A \frac{d r}{d x} \, d r} \) and \( b_j^M = a_j^M - \alpha_j^M \) for any \( 1 \leq M < P \) and for any \( j \)
and rewrite equation 3.23 in the form:
\[
b_1^M - d_M b_1^{M+1} = 0
\]
(3.24)
Besides, since the exact solution and finite element solution are continuous at the
element junctions, the error itself is continuous at these junctions. At the junction of
elements \( M \) and \( M + 1 \) (with \( M < P \)), we have:
\[
b_0^M S_0^M (1) + b_1^M S_1^M (1) + \sum_{i=N}^{\infty} b_i^M S_i^M (1) = b_0^{M+1} S_0^{M+1} (1) + b_1^{M+1} S_1^{M+1} (1) + \sum_{i=N}^{\infty} b_i^{M+1} S_i^{M+1} (1)
\]
(3.25)
or,
\[
b_0^M + 2b_1^M - b_0^{M+1} = 0
\]
(3.26)
Finally, the error satisfies the boundary conditions, hence,
\[
e_h(L) = b_0^1 = 0
\]
(3.27)
and
\[
e_h(R) = 2b_1^P + b_0^P = 0
\]
(3.28)
We recognize that equations 3.24, 3.26, 3.27 and 3.28 are a system of \( 2P \) linear
equations in the \( 2P \) variables \( b_0^i \) and \( b_1^i \) (with \( i = 1, \ldots, P \)). In matrix form, we have:
or \( Ab = 0 \). We show in Appendix A that

\[ \text{det}(A) = 2(1 + \sum_{i=1}^{P-1} \prod_{j=i}^{P-1} d_j) \]  

(3.30)

From their definition, all the \( d_i \)'s are positive and therefore this determinant is positive and \( b = 0 \) is the only solution of equation 3.29. We conclude that, in the case where both boundary conditions are natural and \( EA^4 \) is a constant, over element \( M \)

\[ e_h = \sum_{i=N}^{\infty} (a_i^M - \alpha_i^M) S_i^M \]  

(3.31)

As a result, higher order accuracy in the strain field is obtained at the points where the derivative of \( S_N \) is zero. Again, equation 3.31 shows that the error in the displacements is zero at the end points of all elements, a classic result.

**Distorted elements, varying Young’s modulus and varying cross-sectional area, with a natural boundary condition**

In the general case (i.e. when we no longer make the assumption that the product of the Young’s modulus times the cross-sectional area and the determinant of the mapping function is a constant), we need to modify the above proof.
In the case of a non-constant product $EAdr/dx$, the element orthogonal polynomials $T_j^M$ in the set $T^M$ no longer satisfy $T_j^M(-1) = T_j^M(1)$ and it is not possible to construct a set $S^M$ of polynomials $S_j^M$ that are globally orthogonal to the finite element space by the above method. As a result, it is not immediately obvious that the higher-order-accuracy points are located at the zeros of the derivative of $T_N^M$, like in the case of a constant product $EAdr/dx$, and we need extra arguments to obtain the same conclusion when one of the boundary conditions is natural.

Again, we assume that the exact solution (i.e. the mathematical solution to equation 3.1) over element $M$ is analytic in the local coordinates. Thus we can find a set $b^M$ of coefficients $b_i^M$ such that over element $M$

$$u(r) = u_h(r) + \sum_{i=0}^{\infty} b_i^M T_i^M(r)$$

In the case of constant product $EAdr/dx$, we could argue that for $0 < i < N$ we had $b_i^M = 0$ because of the global orthogonality of the polynomials $S_i^M$. We can no longer use this argument. However, we can still write for any test function $v$

$$\alpha_h^M (\frac{du}{dr}, \frac{dv}{dr}) = \int_M f v + [EAdr/dx v]^M_+ - [EAdr/dx v]^M_-$$

where $M^+$ and $M^-$ are the coordinates of the right and left ends of element $M$.

For every polynomial $T_k^M (k > 2)$, we define as $g_k^M$ the function of the local coordinate $r$ that is linear and that equals $T_k^M$ at 1 and at −1:

$$g_k^M(r) = \text{def} \frac{T_k^M(1) + T_k^M(-1)}{2} + \frac{T_k^M(1) - T_k^M(-1)}{2} r$$

Using $T_j^M - g_j^M$ (with $j > 1$) as our test function $v$ in equation 3.33, and replacing $u$ by its expression 3.11, we obtain

$$\alpha_h^M \left( \frac{d}{dr} (u_h(r) + \sum_{i=0}^{\infty} b_i^M T_i^M(r)) \right) = \int_M f (T_j^M - g_j^M)$$
because the boundary terms vanish due to our choice of \( v \). Using the bilinearity of the element scalar product, we obtain

\[
\sum_{i=0}^{\infty} \beta_i^M (a_h^M (\frac{dT_i^M}{dr}, \frac{dT_i^M}{dr}) - a_h^M (\frac{dT_i^M}{dr}, \frac{dg_i^M}{dr})) = \int_M f(T_j^M - g_j^M) - a_h^M (\frac{d u_h}{dr}, \frac{dT_j^M - g_j^M}{dr})
\]

(3.36)

Finally, exploiting the orthogonality properties of the set \( T^M \) we are left with

\[
b_j^M a_h^M (\frac{dT_j^M}{dr}, \frac{dT_j^M}{dr}) - b_0^M a_h^M (\frac{dT_0^M}{dr}, \frac{dg_j^M}{dr}) - b_1 a_h^M (\frac{dT_1}{dr}, \frac{dg_j^M}{dr}) = \\
\int_M f(T_j^M - g_j^M) - a_h^M (\frac{d u_h}{dr}, \frac{dT_j^M - g_j^M}{dr})
\]

(3.37)

If we choose \( 1 < j < N \), the right hand side of equation 3.37 vanishes, hence

\[
b_j^M a_h^M (\frac{dT_j^M}{dr}, \frac{dT_j^M}{dr}) = b_0^M a_h^M (\frac{dT_0^M}{dr}, \frac{dg_j^M}{dr}) + b_1 a_h^M (\frac{dT_1}{dr}, \frac{dg_j^M}{dr}) \text{ for } 1 < j < N
\]

(3.38)

or, using the explicit form of the element scalar product \( a_h^M, \) with \( T_0^M \),

\[
b_j^M a_h^M (\frac{dT_j^M}{dr}, \frac{dT_j^M}{dr}) = b_1 a_h^M (\frac{dT_1}{dr}, \frac{dg_j^M}{dr}) \text{ for } 1 < j < N
\]

(3.39)

Now, we make use of the orthogonality property of the error, equation 1.12:

\[
\forall v \in V_h, a(u - u_h, v) = 0
\]

(3.40)

with \( v \) chosen to be the \( S_1^M \) function of Figure 3-1. It should be noted that this choice of \( v \) is a valid one for as long as one of the boundary conditions is natural. This gives

\[
0 = \int_{\Omega} EA \frac{de_h}{dx} \frac{dv}{dx} \, dx = \int_M EA \frac{de_h}{dx} \, dx = \int_M EA \sum_{i=0}^{\infty} \beta_i^M \frac{dT_i^M}{dr} \, dx \, dr = b_1^M \int_M EA \frac{dT_1^M}{dr} \, dx \, dr
\]

50
\[ b_i^M \int_M E A \frac{dr}{dx} dr \]  
(3.41)

which implies immediately

\[ b_i^M = 0 \]  
(3.42)

From equations 3.42 and 3.39, we conclude that

\[ \text{if } 1 \leq j < N, b_j^M = 0 \]  
(3.43)

We can therefore write

\[ \frac{du}{dr} = \frac{du_h}{dr} + \sum_{i=N}^{\infty} i_i^M T_i^M(r) \]  
(3.44)

**Conclusion:** For a distorted \( N \)-node element used to model a bar of varying Young's modulus and varying cross-sectional area, higher-order-accuracy points are located at the zeros of \( \frac{dT_i^M}{dr} \) if one of the boundary conditions is natural.

We now consider the case when both boundary conditions are essential: In this case, we show by a counterexample that higher order accuracy points do not exist.

**Counterexample:** We choose \( L = 0, R = 4, A = 1, E = 1 + x \), and two identical three-node non-distorted elements, so that we have a three degree of freedom problem, both ends being constrained. We solve this finite element problem exactly using a symbolic mathematics program.

First, we impose forces such that the exact solution is

\[ u_1(r) = r^3 - \frac{12}{55}r^2 - \frac{1}{2}r + \frac{79}{110} \]  
(3.45)

over the first element (\( r = x - 1 \) being the local coordinate pertaining to that element) and

\[ u_1(r) = r^3 - \frac{24}{235}r^2 - \frac{3}{2}r + \frac{283}{470} \]  
(3.46)
over the second element \((r = x - 3\) being the local coordinate pertaining to that element). The error corresponding to this loading is, over the first element,

\[
e_{h1}(r) = r^3 - \frac{14}{69}r^2 - \frac{341}{345}r + \frac{74}{345}
\]  

(3.47)

and over the second element:

\[
e_{h1}(r) = r^3 - \frac{34}{345}r^2 - \frac{349}{345}r + \frac{38}{345}
\]  

(3.48)

Then, we impose forces such that the exact solution is

\[
u_2(r) = 2r^3 - \frac{122}{55}r^2 - \frac{3}{2}r + \frac{299}{110}
\]  

(3.49)

over the first element \((r = x - 1\) being the local coordinate pertaining to that element) and

\[
u_2(r) = r^3 - \frac{24}{235}r^2 - \frac{3}{2}r + \frac{283}{470}
\]  

(3.50)

over the second element \((r = x - 3\) being the local coordinate pertaining to that element). The error corresponding to this loading is, over the first element,

\[
e_{h2}(r) = 2r^3 - \frac{47}{115}r^2 - \frac{226}{115}r + \frac{51}{115}
\]  

(3.51)

and over the second element:

\[
e_{h2}(r) = r^3 - \frac{11}{115}r^2 - \frac{119}{115}r + \frac{3}{23}
\]  

(3.52)

In the second element, the error in the strain \(\frac{de_{h1}}{dr}\) corresponding to the first loading goes to zero at the points with coordinates \(r = -0.548765...\ and \ r = 0.614466...\), whereas the error in the strain \(\frac{de_{h2}}{dr}\) corresponding to the second loading goes to zero at the points with coordinates \(r = -0.556286...\ and \ r = 0.620054...\). Similarly, in the first element, the zeros of the error in the strain depend on the exact solution. This
proves that in this case there does not exist higher-order accuracy points.

3.2.3 Other one-dimensional scalar products

The approach used in Section 3.2.2 can be more generally employed for other one-dimensional scalar problems that can be written in the form:

\[
\text{Find } u \in V \text{ / } \forall v \in V \quad a(u, v) = (f, v) \tag{3.53}
\]

where \(a(., .)\) is a bilinear form (scalar product), \((f, .)\) is a linear form, and \(V\) includes the space of polynomials. In this more general setting, we can still derive a unique set \(T^M\) of polynomials \(T^M_k\) of order \(k\) orthogonal to each other for the element scalar product \(a^M(., .)\). Again the set \(T^M\) can be constructed by recurrence using the Gram-Schmidt procedure. In general, only the polynomial \(T^M_0\) can be chosen arbitrarily. In the previous section we could also choose \(T^M_1\) arbitrarily because the displacements only appeared through their derivatives in the scalar product. Generally, if the lowest order derivative of the displacements that appears in the scalar product is the \(m^{th}\) derivative, we can choose the first \((m + 1)\) functions arbitrarily. For instance, in the case of a beam in flexure, for which the bilinear form is (see [4])

\[
a(u, v) = \int EI \frac{d^2u}{dx^2} \frac{d^2v}{dx^2} dx \tag{3.54}
\]

we can impose \(T^M_0(x) = 1\), \(T^M_1(x) = x\), \(T^M_2(x) = x^2 - 1\), whereas for the problem \(-\Delta u + cu = f\) on \((0, 1)\) (this problem was studied in [18] with \(c=1\)), we can only impose \(T^M_0(x) = 1\) (it turns out that \(T^M_1(x) = x - \frac{1}{2}\)).

**Beam in flexure** In the case of the beam in flexure, for \(j > 2\), the polynomials \(T^M_j\) are determined by the Gram-Schmidt procedure up to a linear part which is arbitrary (because this part does not enter the bilinear form 3.54). We can therefore choose this linear part such that the polynomials \(T^M_j\) are zero at \(r = 1\) and \(r = -1\). A proof similar to the one used in the case of constant \(EA \frac{d^2r}{dx^2}\) for problem of the bar in tension still applies here and we obtain that for the beam in flexure problem
\[
\frac{d^2(u-u_h)}{dr^2} = \sum_{i=N}^{\infty} b_i^M \frac{d^2S_i^M}{dr^2} = \sum_{i=N}^{\infty} b_i^M \frac{d^2T_i^M}{dr^2} \tag{3.55}
\]

Hence the points where curvature is assessed with higher order accuracy in an N-degree of freedom finite element are located at the zeros of the second derivative of the polynomial \(T_N^M\) (which are two order lower Gauss points in the case of the constant product \(EI\frac{dr}{dx}\), because in this case the polynomials \(T_j^M\) are obtained by integrating the Legendre polynomials twice and then fixing their linear part as indicated above).

**Dependance of \(T^M\) on the problem** It should be realized that the set \(T^M\) is dependent on the problem nature (as shown by the beam in flexure problem above) and also on the constants involved in the problem, as shown by the following example:

Let us consider the problem \(-\Delta u + cu = f\) over \((-1, 1)\). For this problem, we can show that, for any non-distorted element \(M\), the set \(T^M\) starts with the following polynomials:

\[
\begin{align*}
T_0^M(x) &= 1 \\
T_1^M(x) &= x \\
T_2^M(x) &= x^2 - \frac{1}{3} \\
T_3^M(x) &= x^3 - \frac{2+c}{2+\frac{c}{3}}x \\
T_4^M(x) &= x^4 - \frac{6(c+21)}{7(c+15)}x^2 + \frac{3(35+c)}{35(15+c)}
\end{align*}
\tag{3.56}
\]

We note that, for this new problem, only when \(j\) is even (and therefore \(T_j^M\) is even) do we have \(T_j^M(1) = T_j^M(-1)\), unlike in the bar in tension problem. Clearly, the polynomials in the set \(T^M\) for this problem differ from those we obtain for the bar in tension and depend on the parameter \(c\).

### 3.3 The Poisson problem

Let us now show how the same ideas can be extended to two-dimensional problems, such as the Poisson problem (see [8])
The bilinear form for this problem is

$$ \int_J \frac{\partial u}{\partial x} \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial v}{\partial y} d\Omega $$

(3.58)

For clarity, let us assume that this problem is solved using non-distorted 9-node quadrilateral isoparametric elements, and that the usual polynomial shape functions \( h_1 \) to \( h_9 \) are used so that

$$ \text{Span}(h_1(x), ..., h_9(x)) = \text{Span}(1, x, y, x^2, y, x^2y, xy^2, x^2y^2) $$

(3.59)

Since the order of convergence of the finite element strain energy is dictated by the degree of completeness of the polynomial approximation, we can see from Figure 3.3 that all that is needed to increase the order of convergence of this element is to include one polynomial that includes the \( x^3 \) term and one that includes the \( y^3 \) term.

We can show that the polynomial which includes \( x^3 \) and is orthogonal to the finite element space is \( x^3 - x \) and the polynomial that includes \( y^3 \) and is orthogonal to the finite element space is \( y^3 - y \). The points where the derivatives of these two polynomials are zero are the lower order Gauss points for the element of interest, \( x = \pm \frac{1}{\sqrt{3}} \) and \( y = \pm \frac{1}{\sqrt{3}} \).

Obviously, the same approach is valid for other elements than the 9-node element. For elements used to solve the Poisson problem, the zeros of the derivatives of the lowest order orthogonal polynomials are always the lower order Gauss points.

### 3.4 Application to error assessment

In this section we show how the results of the previous section can be used to develop a class of error estimators.
Including the $x^3$ and the $y^3$ polynomials would raise the convergence order of the 9 node quadrilateral isoparametric finite element

Figure 3-3: Pascal triangle
3.4.1 Error estimator for the bar problem

Again, let us consider the deformation of the bar subjected to the loading \( f \). For the problem solution we use 3-node finite elements.

We have shown in the previous section that in any element, distorted or not, it is possible to develop in any element a set of polynomials \( T_k^M \) \((k \geq 0)\) that are orthogonal to each other in the sense of the energy for this problem.

The error in element \( M \) can then be written as

\[
e_h = u - u_h = \sum_{i=0}^{\infty} b_i^M T_i^M(r)
\]

For every polynomial \( T_k^M \) \((k \geq 2)\), we define as before \( g_k^M(r) = \frac{T_k^M(1)+T_k^M(-1)}{2} + \frac{T_k^M(1)-T_k^M(-1)}{2} r \).

Starting from the differential equation for the problem, applying the Galerkin procedure to the test function \( T_l^M - g_l^M \) \((l > 2)\), and using the orthogonality properties of the set \( T^M \), we obtain:

\[
b_l^M = \frac{\int (T_l^M - g_l^M) dx - \int EA \frac{d(T_l^M - g_l^M)}{dx} du dx + b_l^M a_h^M \left( \frac{dT_l^M}{dx}, \frac{dg_l^M}{dx} \right)}{\int EA \left( \frac{dT_l^M}{dx} \right)^2 dx} \quad (3.61)
\]

We have shown in Section 3.2.2 that under certain circumstances the constant \( b_1^M = 0 \), so that all the terms on the right hand side are known or can be evaluated, and therefore for any \( i \), \( b_i^M \) can be calculated. In case we do not have \( b_1^M = 0 \), we are still able to use equation 3.61 to obtain an estimate for \( b_i^M \) by neglecting the \( b_i^M a_h^M \left( \frac{dT_l^M}{dx}, \frac{dg_l^M}{dx} \right) \) term in the numerator.

In the particular case of \( EA \) constant and undistorted elements, the second integral in the numerator equals zero. Since the set \( T^M \) is known, the \( b_j^M \)'s can be calculated before the finite element solution is obtained. We have in this case an a-priori error estimator, in essence by applying p-refinement. When the product \( EA \frac{dr}{dx} \) is not
constant, the polynomials in the set $T^M$ are not known a priori and we have an a posteriori error estimator.

In the general case, we use

$$b_i^{M*} = \frac{\int f(T_i^M - g_i^M)dx - \int EA \frac{d(T_i^M - g_i^M)}{dx} \frac{du_h}{dx} dx}{\int EA \frac{d(T_i^M)}{dx}^2 dx} \tag{3.62}$$

as an estimate for $b_i^M$ and

$$e_h^*(r) = \sum_{i=N}^{i=N'} b_i^{M*} T_i^M(r) \tag{3.63}$$

as an estimate of the error. $N'$ is an integer which determines the desired accuracy of our error estimator. We show in Section 3.4.2 that $N' = N + 1$ might be a good choice.

### 3.4.2 Examples

In this section, we test the error estimator presented above on some sample problems to assess its efficiency. We choose to include only two terms in the error estimate, i.e. we take $N' = N + 1$. All problems are solved on the non-uniform mesh consisting of 3-node elements given in Figure 3.4.2. The Young's modulus varies as $E(x) = 1 + x$.

In Examples 1 and 2 one end ($x = 0$) is clamped and one end ($x = 10$) is free. In Example 3, both ends are clamped.

**Example 1:** We impose a loading per unit length such that the exact displacement is a polynomial of order 4:

$$u(x) = (x + 1)^2 (x - 10)^2 - 100 \tag{3.64}$$

Because $N' = N + 1 = 4$ and the exact solution is a polynomial of order 4, we expect our error estimator to give the exact error for this problem. The recovered strain $\frac{d(u_h + e^*)}{dx}$, the finite element strain $\frac{du_h}{dx}$ and the exact strain are plotted in Fig
Figure 3-4: The 4 element mesh employed to solve the example problems 1 and 2 of Section 3.4.2
3-5. We see that the recovered strain equals the exact strain, as expected.

**Example 2:** We next impose a loading per unit length such that the exact displacement cannot be developed as a finite series:

\[ u(x) = \tan\left(\frac{x - 5}{5}\right) - \frac{x - 5}{5}\tan(1) \] (3.65)

Figures 3-6 and 3-7 shows that the recovered strain is much closer to the exact strain than the finite element strain. The recovered strain is not equal to the exact strain because only two terms have been included in equation 3.63.

**Example 3:** The exact displacement is

\[ u(x) = \tan\left(\frac{x - 5}{5}\right) - \frac{x - 5}{5}\tan(1) \] (3.66)

The difference with example 2 is that now both ends are clamped. Hence in this case, both boundary conditions are essential and, like before, the solution is not a polynomial, which implies that the calculated coefficients given by equation 3.62 are not equal to the exact coefficients in equation 3.61.

The strains presented in Figures 3-8 and 3-9 show that despite the approximation, the recovered strain is very close to the exact strain. Comparing the results of example 3 to those of example 2, we see that the fact that the accuracy of our error estimator is hardly affected by the fact that the boundary condition at \( x = 10 \) is now essential.
Figure 3-5: Strains for example 1

curve 1: $\frac{du}{dx}$, curve 2: $\frac{d(u_h + e_h)}{dx}$; curve 3: $\frac{du_h}{dx}$
Figure 3-6: Strains for example 2

curve 1: \( \frac{du}{dx} \); curve 2: \( \frac{d(\bar{u} + e_3)}{dx} \); curve 3: \( \frac{d(\bar{u} + e_3)}{dx} \)
Figure 3-7: Strains for example 2

curve 1: $\frac{du}{dx}$, curve 2: $\frac{d(u_1 + e^1)}{dx} - \frac{du}{dx}$
Figure 3-8: Strains for example 3

curve 1: $\frac{du}{dx}$, curve 2: $\frac{du_h}{dx}$, curve 3: $\frac{d(u_h + e_h^*)}{dx}$
Figure 3-9: Strains for example 3

curve 1: \( \frac{d(u_h + \varepsilon_h^* x)}{dx} = \frac{du}{dx} \), curve 2: \( \frac{du_h}{dx} = \frac{du}{dx} \)
Chapter 4

Conclusions

The accurate assessment of the error is a crucial issue in finite element analysis (see [4]). In particular, error estimates are an indispensable part of any mesh refinement process. Numerous contributions have been made to the development of error estimates, for example to establish implicit and explicit estimators, the truth mesh approach and recovery-based procedures (see [2, 12]).

Implicit algorithms are a very important class of error estimators. Until now, recovery-based algorithms and implicit algorithms were considered two separate and unrelated approaches to the same problem. In this thesis, we have shown how these two categories of algorithms are in fact closely related.

In engineering, recovery-based algorithms are the most popular. However, they show a significant number of deficiencies. Specifically, their patch-based nature is the source of a number of drawbacks: their implementation requires extra data structures, they smoothen out discontinuities of the stress and strain fields when material properties are constant piecewise only (unless interfaces are dealt differently from the rest of the computational domain), and the mathematical foundation is weak. In particular, the choice of the sampling points used in recovery-based algorithm is a critical issue. The most popular recovery-based algorithms use either the lower order Gauss points or the Barlow points as sampling points. We showed in this thesis that these
are not the best points to sample the strain fields in general. We showed how the location of the best sampling points (the points in the elements where a higher rate of convergence is observed, or higher-order-accuracy points) can be exactly evaluated.

The knowledge of the exact location of the higher-order-accuracy points could be used to design a more efficient recovery-based error estimator. As it turns out, the proof of the existence of the higher-order-accuracy points also suggests a different approach to error estimation, one that does not require the use of patches and thus does not show the above-mentioned deficiencies. This approach is based on the projection of the error on a set of polynomials that is orthogonal to the finite element space employed for the analysis. This projection is carried out either exactly or approximately on a limited number of polynomials. The performance of this type of error estimator was assessed by applying it to a number of sample one-dimensional problems. Our sample problems included non-uniform meshes, point forces and varying material properties. These are typical problems that are not handled well by existing recovery-based algorithms. We showed that the algorithm performs very well in all these cases, even when the projection is performed only approximately.
Appendix A

Determinant of the matrix A

In this appendix, we show that the determinant of the following matrix can be calculated exactly:

\[
A = \begin{pmatrix}
1 & 2 & -1 & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_i & & & & \\
1 & 2 & -1 & & & & \\
& & & \ddots & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_i & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_{P-1} & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_{P-1} & & & & \\
\end{pmatrix}
\] (A.1)

The determinant of this matrix can be calculated in the following way: first, we define for \(0 < N < P\)

\[
B_{P-N} = \text{det} \begin{pmatrix}
2 & -1 & & & & & \\
1 & 0 & -d_N & & & & \\
1 & 2 & -1 & & & & \\
& & & \ddots & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_i & & & & \\
& & & & \ddots & & & & \\
1 & 2 & -1 & & & & \\
1 & 0 & -d_{P-1} & & & & \\
1 & 2 & -1 & & & & \\
\end{pmatrix}
\] (A.2)
and for $0 < N < P - 1$ we define

$$C_{P-N} = \det \begin{pmatrix} 0 & -d_N & 1 & 2 & -1 \\ 1 & 0 & -d_{N+1} & 1 & 2 & -1 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_1 & 1 & 2 \\ 1 & 0 & -d_{P-1} & 1 & 2 \end{pmatrix}$$

(A.3)

We develop $\det(A)$ with respect to the first row, to obtain

$$\det(A) = \det \begin{pmatrix} 2 & -1 & 1 & 0 & -d_1 & 1 & 2 & -1 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_i & 1 & 2 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_{P-1} & 1 & 2 \end{pmatrix}$$

(A.4)

We then develop the determinant on the right side of equation A.4 with respect to the first row:

$$\det(A) = 2\det \begin{pmatrix} 1 & -d_1 & 0 & -d_2 \\ 1 & 0 & -d_2 & 1 & 2 & -1 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_i & 1 & 2 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_{P-1} & 1 & 2 \end{pmatrix} + \det \begin{pmatrix} 1 & -d_1 & 0 & -d_2 \\ 1 & 0 & -d_2 & 1 & 2 & -1 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_i & 1 & 2 \\ \vdots & 1 & 2 & -1 \\ 1 & 0 & -d_{P-1} & 1 & 2 \end{pmatrix}$$

(A.5)
Then we develop the second determinant on the right side of equation A.5 with respect to the first column:

\[
\begin{align*}
det(A) &= 2 \det \begin{pmatrix}
0 & 1 & -d_1 \\
1 & 2 & -1 \\
1 & 0 & -d_2
\end{pmatrix} + \det \begin{pmatrix}
2 & 1 & -1 \\
1 & 0 & -d_2 \\
\ldots
\end{pmatrix} \\
&= 2 \det \begin{pmatrix}
0 & 1 & -d_1 \\
1 & 2 & -1 \\
1 & 0 & -d_2
\end{pmatrix} + \det \begin{pmatrix}
2 & 1 & -1 \\
1 & 0 & -d_2 \\
\ldots
\end{pmatrix}
\end{align*}
\]

Hence, we have

\[
det(A) = B_{p-1} = 2C_{p-1} + B_{p-2}
\]

and more generally, the recurrence formula

\[
B_{p-j} = 2C_{p-j} + B_{p-j-1}
\]

holds. Hence we have,

\[
det(A) = B_{p-1} = 2 \sum_{j=1}^{j=P-2} C_{p-j} + B_1
\]

Also, we can obtain a recurrence formula for the \(C_{p-j}\)'s: developing \(C_{p-j}\) with respect to its first row, we have

\[
C_{p-j} = d_j \det \begin{pmatrix}
1 & 2 & -1 \\
0 & -d_{j+1} \\
1 & 2 & -1
\end{pmatrix} \\
&= d_j \det \begin{pmatrix}
1 & 2 & -1 \\
0 & -d_{j+1} \\
1 & 2 & -1
\end{pmatrix}
\]

(A.10)
and then developing the determinant on the right side with respect to the first column, we obtain

\[ C_{P-j} = d_j C_{P-j-1} \]  \hspace{1cm} (A.11)

Finally, using the fact that

\[
\begin{pmatrix}
0 & -d_{P-2} & 0 & 0 \\
1 & 2 & -1 & 0 \\
0 & 1 & 0 & -d_{P-1} \\
0 & 0 & 1 & 2
\end{pmatrix} = -det
\begin{pmatrix}
-d_{P-2} & 0 & 0 \\
1 & 0 & -d_{P-1} \\
0 & 1 & 2
\end{pmatrix} = d_{P-1} d_{P-2}
\]  \hspace{1cm} (A.12)

and that

\[
B_1 = det
\begin{pmatrix}
2 & -1 & 0 \\
1 & 0 & -d_{P-1} \\
0 & 1 & 2
\end{pmatrix} = 2d_{P-1} + 2
\]  \hspace{1cm} (A.13)

along with equations A.9 and A.11, we conclude that

\[ det(A) = 2(1 + \sum_{i=1}^{P-1} \prod_{j=i}^{P-1} d_j) \]  \hspace{1cm} (A.14)
Bibliography


