USE AND IMPLEMENTATION OF THE RECOVERY BASED ERROR ESTIMATORS IN STRUCTURAL ANALYSIS

Saulius Ziukas* and Nils-Erik Wiberg†

* Department of Structural Mechanics
Chalmers University of Technology
S-412 96 Goteborg, SWEDEN

† e-mail: new@sm.chalmers.se

Key words: Finite Element Method, Error Estimates, Gradient Recovery, Object Oriented Programming, C++ programming language

Abstract. The problem of selection an effective and reliable error estimator has become the main concern in adaptive finite element analyses. One of such estimators proposed by Zienkiewicz and Zhu has recently attracted a considerable attention. Although the estimator is simple and easy to implement, it provides an accurate estimate of the discretization error for a wide class of problems. The basic concept of Zienkiewicz–Zhu error estimator (ZZ) is to replace the exact value of error of the gradients by an approximation, where \( \hat{\nabla} \) denotes an improved recovered \( C^0 \)–continuous gradient (stress) field. Obviously, performance of the estimator highly depends on the recovery procedures used to obtain the recovered gradient.

The Superconvergent Patch Recovery procedure is considered as the basic recovery tool. The original procedure is based on a discrete least square fit of the polynomial to the finite element solution at a certain number of points on local element patches. We investigate conditions necessary to ensure asymptotic exactness of the estimator and outline possible enhancements of the original procedure leading to more narrow bounds of the global effectivity index. Various modifications of the minimization functional that takes into account equilibrium, boundary conditions, differential operator, are considered aiming to improve performance of the estimator.

The paper also introduces basic concepts of object-oriented design of gradient recovery procedures using C++ programming language. Several class definitions are discussed. It is shown that object-oriented programming paradigm makes software more reliable, easier understandable, and extendible.
1 INTRODUCTION

We consider a linear two-dimensional problem in linear isotropic elasticity: find $u = (u_i)$ in $\Omega \subset \mathbb{R}^2$:

\[
\delta = \lambda \text{div}uI + 2\mu \mathbf{\varepsilon}(u), \quad -\text{div}\delta = f \quad \text{in} \ \Omega
\]

\[
u = 0 \quad \text{on} \ \Gamma_1, \quad \delta \mathbf{n} = g \quad \text{on} \ \Gamma_2.
\]

where $\partial \Omega := \Gamma_1 \cup \Gamma_2$, $\mathbf{\varepsilon}(\mathbf{v})$ is the strain tensor defined by

\[
\mathbf{\varepsilon}_{ij}(\mathbf{v}) := \frac{1}{2} \left( \frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \quad i, j = 1, 2,
\]

and $\mu$, $\lambda$ are the Lamé parameters. The variational formulation of (1) in $H = L^2(\Omega)$ reads as follows: find $u \in V$, where

\[
V = \left\{ v \in [H^1(\Omega)]^2 \quad | \quad v = 0 \quad \text{on} \ \Gamma_1 \right\}
\]

such that

\[
a(u, v) = L(v) \quad \forall v \in V.
\]

The bilinear form has the form of 'virtual work':

\[
a(u, v) := \int_{\Omega} \delta(u) : \mathbf{\varepsilon}(v) \, dx
\]

\[
L(v) := \int_{\Omega} f \cdot v \, dx + \int_{\Gamma_2} g v \, ds.
\]

Let $T_h = \{ \mathcal{E} \}$ be a standard finite element subdivision of $\Omega$ into non-overlapping disjoint triangles or quadrilaterals $K$ of diameter $h_K$. For each $K \in T_h$, $P(K)$ is a space of polynomials on $K$ containing the set of polynomials of a complete degree at most $p_k$. Defining a standard finite element space

\[
S_h^p = \left\{ v \in V : v \in C(\overline{\Omega}) \quad | \quad v |_K \in \left[ P(K) \right]^2, \forall K \in T_h \right\}
\]

we consider the standard finite element method: find $u^h \in S_h^p$ such that

\[
a(u^h, v) = L(v), \quad v \in S_h^p.
\]

The discretization error of the finite element solution is defined as the difference between the exact $u$ and the approximate solution $u^h$: $e = u - u^h$. In the standard 'displacement' formulation for minimization problems the difference between $u$ and $u^h$ is most naturally measured in the so called 'energy' norm $\| \cdot \|_{E, \Omega}$ which is constructed using the bilinear form.
\( a(\cdot, \cdot) \) as \( \|v\|_{E, \Omega} = a(v, v)^{1/2} \). For elastostatic problems discretization error in the energy norm reads as

\[
\|e\|_{E, \Omega} = \left( \int_{\Omega} (\sigma - \sigma^h)^T \mathbf{D}^{-1} (\sigma - \sigma^h) \, d\mathbf{x} \right)^{1/2}
\]

where \( \mathbf{D} = (D_{ij}) \) denotes the constitutive matrix. The global rate of convergence of the stresses \( \sigma^h \) calculated directly differentiating the finite element solution is one order lower than that of the finite element solution \( u^h \) itself. Usually, some “smoothing” procedure is always employed in order to achieve more acceptable approximations to the stresses. Using such recovered gradients, the Zienkiewicz–Zhu (ZZ) error estimator can be calculated at a fraction of the total cost of computation.

### 2 ZIENKIEWICZ-ZHU ERROR ESTIMATOR

The basic concept of the ZZ error estimator consists of replacing the exact value of error of the stresses \( e_{\sigma} = \sigma - \sigma^h \) in the analytical expression of the discretization error in energy norm (8) by an approximation defined as \( e_{\sigma}^* = e_{\sigma}^* = \sigma^* - \sigma^h \) and using it as a measure of the local error, i.e. accomplishing directly a local auxiliary solution to determine the error. \( \sigma^* \) denotes in a some way recovered stress field. The approximation of the error in energy norm over an element is

\[
\eta_K := \left( \int_K (\sigma^* - \sigma^h)^T \mathbf{D}^{-1} (\sigma^* - \sigma^h) \, d\mathbf{x} \right)^{1/2}
\]

and this is known as error indicator. Error estimator in the domain \( \Omega \) is the square root of the sum of squared elemental error indicators as an approximation to true error \( \|e\|_{E, \Omega} \):

\[
E_{\Omega}(u)^h := \left( \sum_{K \in T_h} \eta_K \right)^{1/2} = \left( \sum_{K \in T_h} \left( \int_K (\sigma^* - \sigma^h)^T \mathbf{D}^{-1} (\sigma^* - \sigma^h) \, d\mathbf{x} \right)^{1/2} \right)^{1/2}
\]

Obviously, reliability of the estimator is dependent on the properties of the recovered gradient \( \sigma^* \). The reliability of an error estimator is usually judged by an effectivity index \( \theta \) defined as a ratio of the estimated error to the true error:

\[
\theta := \frac{E_{\Omega}(u)^h}{\|e\|_{E, \Omega}}
\]

Two conditions are often desirable to be satisfied by an error estimator: asymptotic exactness and prediction of reliable error bounds. Estimator is said to be asymptotically exact if

...
\[ \| e \|_{E,\Omega} \sim \left\{ 1 + O(h^\gamma) \right\} \| e_h \|_{E,\Omega} \quad \text{as } h \to 0 \]  

(12)

where \( \gamma > 0 \) is independent of \( h \). The condition for ZZ error estimator is achieved if \( \| e^* \| = \| \delta - \delta_h \|_{0,\Omega} \) converges at a higher rate than \( \| e \| = \| \delta - \delta_h \|_{0,\Omega} \). It follows that if \( \| e^* \| \) is superconvergent then asymptotic exactness of the error estimator is assured. While a higher rate of convergence of \( \| e^* \| \) is desirable the error estimator will always be practically applicable provided recovered values are more accurate though not necessarily superconvergent than those obtained by the finite element computations.

An error estimator should yield reliable upper and lower bounds for the true error

\[ C_L E_\Omega \left( u^h \right) \leq \| e_h \|_{E,\Omega} \leq C_U E_\Omega \left( u^h \right) \]  

(13)

where \( C_L \) and \( C_U \) are constants depending only on qualitative properties of the mesh but are independent of \( h \) and the particular solution. Babuska et al.\(^4\) developed a computer based approach to measure the robustness index of the estimator which is defined as

\[ \mathcal{R} := \max \left\{ |1 - C_L| + |1 + C_U|, \left| 1 - \frac{1}{C_L} \right| + \left| 1 + \frac{1}{C_U} \right| \right\} \]  

(14)

The problem of the reliability of the estimator has been addressed by Ainsworth and Craig\(^5\). They considered an auxiliary error problem: find \( e^* \in H^1_0(\Omega) \) such that

\[ a(e, v) = (f, v) - a(u^h, v) + (g, v)_{\Gamma}, \quad \forall v \in H^1_0(\Omega) \]  

(15)

and showed that for \( h \) sufficiently small

\[ \| u - u^h \|_{E,\Omega} \leq 2(1 + C_L h^\alpha) \hat{k}(\delta^*) \]  

(16)

if \( \delta \cdot n - g = 0 \) on \( \Gamma_2 \). \( C_L, C_2, \alpha \in (0,2) \) are some constants and

\[ \hat{k}(\sigma^*) = - \frac{1}{2} \int_\Omega (\delta^* - \delta_h) D^{-1}(\delta^* - \delta_h) dx - \frac{1}{2} \int_\Omega (f + \text{div} \hat{\delta}^*) dx \]  

(17)

The theorem states conditions when the Zienkiewicz–Zhu error estimator become the upper one. i.e. under what conditions the effectivity index \( \theta \) is always larger than 1. The difficulty is to satisfy equality constraints on \( \delta^* \)

\[ \text{div} \delta^* + f = 0 \quad \text{in } \Omega \]  

\[ \delta n - g = 0 \quad \text{on } \Gamma_2 \]  

(18)

As ZZ error estimator usually involves only local computations, so generally the condition
cannot be satisfied exactly. It should be noted that the theorem suggests ways of improving the estimator by trying to minimize the residuals (18). This observation triggered intensive research in the field of postprocessing procedures to develop more adequate postprocessing procedures that enable a better estimate of the discretization error, see Wiberg et al.\textsuperscript{6}, Lee et. al\textsuperscript{7}.

3 SUPERCONVERGENT PATCH RECOVERY

Let us consider a linear continuous gradient recovery operator \( G : u^h \in S^p_h \rightarrow \delta^* \in S^p_h \). We notice that \( \delta^* \) might even belong to a wider subspace of \( H^1(\Omega) \) than \( S^p_h \). The gradient recovery operator is said to be superconvergent if

\[
\| \delta - Gu^h \| = O(h^\alpha) \quad \text{as } h \rightarrow 0
\]

where \( \| \cdot \| \) has to be close to \( \| \cdot \|_{W^1(\Omega)} \) in some sense, and \( \alpha \) is a larger value than the convergence rate of the gradient in the finite element method. Several schemes for a recovery of a superconvergent gradient field from the finite element solution \( G \) have been an object of an intensive research in the last decade. They share a common feature to use superconvergent properties of the finite element solution, see survey of Krizek and Neittaanmäki\textsuperscript{8}. The reason of using the superconvergent properties is to get a sufficiently good approximation to the exact gradient \( \delta \), i.e. the recovered gradient must be a much better approximation to \( \delta \) than \( \delta^h \)

\[
\| \delta - \delta^* \|_{L^2(\Omega)} < \| \delta - \delta^h \|_{L^2(\Omega)} \quad \text{(20)}
\]

in order to ensure narrow bounds of the effectivity index.

We distinguish two approaches to the problem of constructing a higher order gradient field. The first is based on the observation that the rate of convergence at certain points exceeds the possible global rate. The gradient recovery operator \( G \) in that case is merely interpolation operator of a degree at least \( p \). However, a necessary condition for the superconvergence includes translation invariant meshes and a rather high regularity of the exact solution. Generalization of the result for locally symmetric meshes was made by Wahlbin\textsuperscript{9}.

Another approach is based on superconvergence in a much broader sense than before. One can recover the finite element solution and/or its derivatives by means of various post-processing techniques and get acceleration of convergence. After such a post-processing, one can often get an increase of accuracy not only at some isolated points, but also in a subdomain (local superconvergence) or even in the whole domain (global superconvergence). In this case the gradient recovery operator \( G \) takes the form like

\[
G(u^h) = \sum w_i \nabla u_i^h
\]

provided the exact solution is sufficiently regular. Different weights \( w_i \) are used for different configurations of element patches.
Saulius Ziukas and Nils-Erik Wiberg.

Zienkiewicz and Zhu\(^2\) proposed a superconvergent patch recovery procedure which became very popular in engineering community. The original SPR assumed existence of the superconvergent points, however, the procedure seems to be a simplified version of that mentioned above. It is based on a discrete least squares fit of a polynomial \( P \) to the gradient values at superconvergent or superior accuracy points. In order to maintain efficiency of the procedure a number of local subproblems are solved instead of the global one: find \( \hat{\sigma}^* = Pa \) such that

\[
\| \hat{\sigma}^* - \sigma^h \|_{L^2, \Omega} = \inf_{\sigma} \| \hat{\sigma}^* - \sigma^h \|_{L^2, \Omega} \tag{22}
\]

In order to accomplish this, the domain \( \Omega \) is subdivided into overlapping subdomains \( w_j \) (patches of elements), \( w_j = \bigcup_{i=1}^m K_j, m \) is the number of elements constituting the patch. The minimization condition results in the solution of an equation system

\[
\sum_{i=1}^{n_{sp}} P'(x_i) P(x_i) a = \sum_{i=1}^{n_{sp}} P'(x_i) \hat{\sigma}^h(x_i) \tag{23}
\]

constructed at points \( \{x_i\}_{i=1}^{n_{sp}} \). \( a \) denotes unknown coefficients of polynomial \( P \).

4 OBJECT-ORIENTED PROGRAMMING

Many older FEM codes written in FORTRAN are carefully designed using features of traditional structured programming. However, the limitations of FORTRAN are especially noticeable expanding large programs: those programs require a significant amount of work. During the last 10-15 years significant achievements have been made in the development of large-scale software systems using object-oriented programming. Object-oriented paradigm (OOP) offers better organized and more open way of programming.

The fundamental concept in OOP is a class. An essential characteristic of a class is encapsulation: data of the class are directly accessible only by this object. Another important feature of OOP is inheritance when classes that have common methods or data are grouped together. Polymorphism allows having methods with different implementations to share the same name. Inheritance and polymorphism are essential for the extensibility of the code. Generally, object-oriented programs are built on a kernel to which new classes can be appended. The additions are made deriving a class from the base class.

We have chosen C++ programming language for the development of our code. C++ is most popular programming language that supports data abstraction and OOP. C++ was designed with the purpose of combining advantages of object-oriented programming with the computational efficiency of the C language\(^6\). C++ type-checking and data-hiding features rely on compile-time analysis of programs. They are used freely without incurring run-time or space overheads. The template facility is primarily used to support statically typed containers and to support elegant and efficient use of such containers (generic programming). Features like templates and inline functions helps to achieve computational efficiency that is
comparable with FORTRAN codes.

The main building blocks (classes) of the SPR gradient recovery procedure are shown in Figure 1.

As the recovery procedure represents local projections over overlapping element patches, we create `Patch` and `Projection` classes. Responsibility of the `Patch` class is to collect topological information stored in `FEMesh` class about neighboring elements that form element patch and are connected by a vertex node. `Patch` class has access to the finite element solution through the base class pointer of the `FiniteElement` class. It can also provide additional data as location and number of superconvergent points in a particular finite element of the patch, total number of superconvergent points, minimum and maximum coordinates for computing jacobian of the path of elements. This information is used to update `Projection` class object iterating over all elements surrounding the vertex.

5 NUMERICAL EXAMPLES

We present here two numerical examples to demonstrate performance of SPR for problems when the finite element method produces significant discretization errors. However, even for complex problems SPR gives fairly good results.

The first problem is L–shaped domain subjected to a symmetric (Mode 1) and
Saulius Ziukas and Nils-Erik Wiberg.

antisymmetric loading (Mode 2), see Figure 2. The solutions possess singularities $\lambda_1 = 0.544$ and $\lambda_2 = 0.90852$ for Mode 1 and Mode 2 respectively. The problem was solved with bilinear quadrilaterals using uniform discretization. We noticed obvious similarity in the behavior of the recovered and finite element solution: singularity deteriorates accuracy of the finite element solution as well as the recovered solution. Note the strong pollution effect for the first mode. However, the accuracy of the recovered field still is better than of the finite element solution itself. It is interesting that the lowest elemental effectivity indices are far away from the singular point.

![Mode 1](image1.png)  ![Mode 2](image2.png)

Figure 2: Elemental effectivity indices using SPR for L-shaped domain problem

REFERENCES


