ENS6139 Master of Engineering 2

Final Report

A h-adaptive scaled boundary finite element method based on maximisation of the error decrease rate

Cong Thanh BUI
Student ID 10396077

Master of Engineering (Mechanical)
Edith Cowan University
School of Engineering

Date of Submission: 29 May 2017

Supervisor: Dr Thu Hang Vu

Copyright and Access

Author Declaration

I certify that this thesis does not, to the best of my knowledge and belief:

(i) incorporate without acknowledgement any material previously submitted for a degree or diploma in any institution of higher education;
(ii) contain any material previously published or written by another person except where due reference is made in the text; or
(iii) contain any defamatory material.

Name: _____

Signature: _____ Date: _____

Abstract

This report implements the projection-based interpolation into the h-refinement method for scaled boundary finite element method. Despite many research about projection-based methods, most of them uniformly increase the polynomials order by one to achieve the fine mesh and then project it to the coarse mesh to study the error decrease (Vu & Deeks, 2007). There is not yet a specific work for projecting an h-refined mesh to the coarse mesh. This report will study the theory of projecting method and derive the formula to implement projection-based technique with h-refinement. To compare the performance, this study also works on superconvergent patch method and reimplement it (O. C. Zienkiewicz & Zhu, 1992) to drive the refinement process.

This work selects p-hierarchical Lobatto shape functions to deploy the solution. As the research of superconvergent patch recovery used traditional Lagrange shape function, this implementation with Lobatto shape functions come with an assumption that the stress of the hierarchical element at the optimal point is also accurate and confirmed their statement and the end of the report (Oh & Batra, 1999).

From the result, it shows that the superconvergent patch recovery implementing with p-hierarchical shape functions provides a smooth stress field and can be used to drive the h-adaptivity process. The implementation of the projection-based method also shows an apparent converged rate which confirms it is possible to use the h-refined mesh with projecting technique. It also indicates that the projecting method has the faster-converged rate comparing to the traditional patch recovery. The resulting mesh in each iteration is recorded to illustrate the adaptivity procedure. It is clear that the adaptivity process is successful to locate and refine elements that contribute the most error calculating by the error estimations.

Finally, the report attempts to convert the formula from loop form to vector form to utilise the advantages of vector operations. It also discusses numerical problems happen when the size of an element become small. This calculation problem causes the instability in the computation, and effect the result when the mesh become finer.

Acknowledgement

I would like to express my genuine gratitude to my project supervisor Dr. Hang VU for her dedication, and patience in helping me with my project and report. I am extremely lucky to be supervised by Dr. Hang VU. The standard and amount of previous work she did in scaled boundary finite element method inspired me to study about this subject. By sharing her expert knowledge in this subject, she helps me gain the insights and become more confident in dealing with the problem.

Table of Contents

1 Introduction

1.1 Introduction

The scaled boundary method which is a novel semi-analytical method was introduced firstly by Wolf and Song (Song & Wolf, 1996). In the beginning, as the derivation of this technique is significantly more complicated than the origin of finite element method, Deeks and Wolf in their work has formed a new virtual work formulation for elastostatics to amplify the invasion of the scaled boundary finite method into generic domains and therefore be more accessible (Deeks & Wolf, 2002c). The scaled boundary method has proved its versatility, accuracy, and potential in certain problems (Song & Wolf, 1999). For instance, when addressing the stress singularities and unbounded domain problem, the performance of this method is outstanding in comparison with the standard finite element method (Deeks & Cheng, 2003). However, a drawback of the method is the computation burden of solving quadratic eigenproblems to obtain the solution. The computation expense has a tendency to grows rapidly as the number of degrees of freedom increases. Therefore, a common problem in deploying scaled boundary method is to minimise the number of degrees of freedom while still reserves the desired accuracy.

As an approach to address this issue, Song reformulates the problem to reduce a set of deformation modes which results in the smoothest deformation of the boundary. This procedure is used to model the far field while using the finite element method to model the near field where the deformation is not smooth (Song, 2004). Another approach is introduced by Deeks and Wolf as they describe in their previous work a h-hierarchical adaptive procedure to attain the optimised solution while retaining the lowest number degrees of freedom. The refinement process was driven by stress recovery and error estimation techniques developed in their previous work (Deeks & Wolf, 2002a). They reported using superconvergent patch recovery method to recover the stress field and assume that a "smoother" version is closer to the exact solution. Then, the error estimators can compute the energy norm of the stress error and subdivide the element that failed to meet the criteria. Deeks and Wolf have done various research about superconvergent patch recovery in scaled boundary finite element method (Deeks & Wolf, 2002b), and most of their work used traditional Lagrange shape functions.

Another choice is to implement the reference solution rather than the superconvergent patch recovery method in approximating the exact hidden result. The reference solution resulted from the uniform adding one degree of freedom into the coarse mesh, and the new refined mesh will be substituted for the exact unknown solution in error estimates functions. As this reference solution is more accurate than the solution measured from the coarse mesh, the stress recovery method is not necessary to compute the error estimator and can be obsoleted. There are various research using this technique for finite element problems and scaled boundary finite element problems (Vu & Deeks, 2007; Gui & Babuska, 1985; Demkowicz, 2006). In this technique, it is popular to use a uniform order refined mesh as a reference solution. In this manner, a coarse mesh is refined by increase the order of the shape polynomials by one degree of freedom. This approach was also introduced to scaled boundary finite element method (Vu & Deeks, 2008). On

another hand, one can choose a h-refined mesh as a reference solution. This approach requires adding the new nodes to every element of the coarse mesh to obtain the fine mesh. Note that, the total number degree of freedom still increases only one unit in each element as the same with the p-refine method. Although, in his book (Demkowicz, 2006), Demkowicz mentioned about this h-refinement. There is not yet an explicit work derived for this problem.

Vu and Deeks introduced a new set of refinement criteria to drive the refinement process (Vu & Deeks, 2008). This approach follows the concept of steepest descent method to achieve the optimal mesh. The steepest gradient technique locates the stationary point (or a saddle-point) of a function by moving step-by-step in the direction of its negative gradient. The function used to guide the procedure is termed the cost function. In the adaptive method, the cost function can be defined as a difference between the exact solution and the current solution. This cost function can be calculated by employing the projection-based interpolation technique. During the adaptive process, approximating the correct solution is conduct by using the reference solution and compared by projecting it on the rough mesh. Elements occupy for a large participation to the global error will be refined or subdivided.

1.2 Objectives

As mentioned in the previous section, scaled boundary finite element method or SBFEM has proved its advantages in capturing the singularity and reducing the complexity of a problem to one degree of freedom. But, this method also represents a quadratic eigenproblem that is needed to obtain the solutions. This problem requires heavy computation and surge with the increment in the number degree of freedom. Therefore, it is critical to seek for an optimal mesh that has high accuracy but low number degree of freedom.

Inspiring from Vu and Lezes Demkowics (Demkowicz, Rachowicz, & Devloo, 2002), this project attempts to investigate the performance of the h-refinement method using projected result of uniform refined mesh as a reference solution. In this manner, a fine mesh obtained from h-uniform refinement will be used to model the exact solution. It is required to apply the method of steepest descent in interpolating the projection of the reference solution and use it to drive the process. It also reviews and implements superconvergent patch recovery method to guide the h-refinement process. Finally, the result of both methods can be compared to justify the converged rate. The algorithm will be implemented in Matlab environments.

1.3 Significance

Implementing a projection-based method to h-refinement will enable the further research to develop the hp-refinement for scaled boundary finite element problem. It helps to complete three popular adaptivity tools in finite element method which are h, p, and hp methods. Moreover, this report can determine the converged rate of the h-adaptivity method using projecting recovery and compare it with the traditional superconvergent patch recovery.

1.4 Report Organisation

This report contains five chapters which are arranged as below:

Chapter 2 presents a background of scaled-boundary finite element method, reference solution, refinement procedure and an introduction of example problems used in the project. The introduction of SBFEM includes the coordinate system definition of the method, equations to calculate displacements and stress field, and the vectorised form of those equations. It also discusses two types of shape function and their expressions. The final section of chapter represents two example problems which are used throughout the project.

Chapter 3 addresses the problem by a quick introduce of superconvergent patch recovery method. In this section, it will discuss least square fitting and error estimation for patch recovery method. Next, it discusses projecting theory to derive the formula to project the fine solution to the coarse mesh. Eventually, a complete formula to implement the projecting recovery and error estimation is provided.

Chapter 4 describes the results and discussion. In this chapter, there are results from two main example problems. The result will reflect the effect of the projection-based method in h-refinement. It also discusses any remained problems and suggestion to remedy the problems.

Chapter 5 provides conclusion and future work.

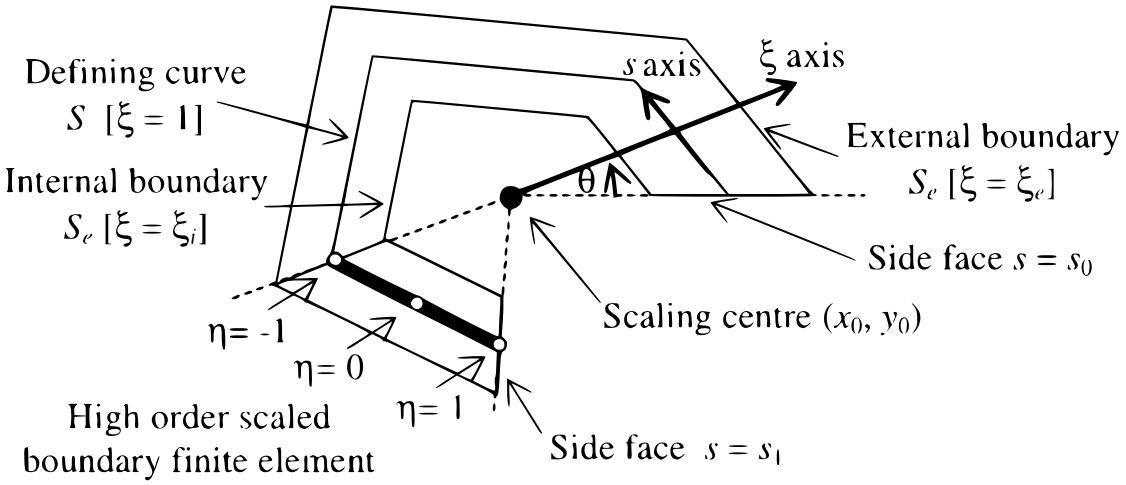

Figure 1: Definition of the scaled boundary coordinate system

2 Background

2.1 The scaled boundary finite-element method

Scaled-boundary finite element method adopted a scaling coordinate system around a scaling center to exploit the ability to solve analytically ordinary differential equations. As illustrating in figure 1, a scaling center (x_0, y_0) is selected within a 2D domain problem. Unlike traditional finite element method which discretises directly the whole domain, scaled-boundary finite element method only discretises the boundary regarding to the scaling center. The problem domain can be introduced by a circumferential coordinate s presenting the length along the boundary and a radial coordinate ξ presenting scaled boundary from the center to the boundary. The radial coordinate ξ is zero at the caling center (x_0, y_0) and is one at the boundary. A transformation equation between Cartesian coordinate systam and Scaled boundary coordinate system can be derived by the following equations. For convenience, this report will adopt the same symbol system from Deeks's work (Deeks & Wolf, 2002c).

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \xi \begin{bmatrix} x_s \\ y_s \end{bmatrix} \tag{1}$$

The scaling centre is indicated by (x_0, y_0), while the $x_s(s)$ and $y_s(s)$ present the coordinates of the intersection between the radial line and the defining curve. Using the transformation formula (1), the unknown displacement of elements u(x, y) in Cartesian coordinate will become u(ξ, s) in Scaled boundary coordinate. The process to achieve the solution of displacement u(ξ, s) required establishing and solving a the virtual work equation, the details solution can be found in Deeks and Wolf papers (Deeks & Wolf, 2002c). As a result, a vector $\{c\}$ of size $n \times 1$, a vector $\{\lambda\}$ of size $n \times 1$ and a matrix $[\Phi]$ of size $n \times n$ is obtained, with n is the total number degree of freedoms. Then, the

stress field $u(\xi, s)$ is determined by interpolating between the radial nodal nodes $u(\xi)$ and shape functions $N(s)$.

$$u(\xi, s) = [N(s)]u(\xi) = [N(s)] \sum_{i=1}^{n} c_i \xi^{-\lambda_i} \{\phi_i\} \tag{2}$$

Formula (2) can be re-organised in vector form as follow

$$u(\xi, s) = [N(s)][\Phi]\{K_{c\lambda}(\xi)\} \tag{3}$$

where

$$\{K_{c\lambda}(\xi)\} = [c_1 \xi^{-\lambda_1} \ c_2 \xi^{-\lambda_2} \ ... \ c_n \xi^{-\lambda_n}]^T \tag{4}$$

$$[\Phi] = [\phi_1 \ \phi_2 \ ... \ \phi_n] \tag{5}$$

The central distinction has been illustrated through equation (2). The scaled boundary finite element method is different than the finite element method in obtaining the nodal displacements regarding functions changing in the radial orientation ($u(\xi)$), instead of individual nodal values, and retrieves the answer as a summation of modal participation ($\sum_{i=1}^{n} c_i \xi^{-\lambda_i} \{\phi_i\}$). As mentioned above, the number of modes n identified by the answer method is indistinguishable to the number of degree of freedoms in the mesh practised. Each term $c_i \xi^{-\lambda_i} \{\phi_i\}$ describes an self-supporting mode of deformation. For each mode, λ_i is the modal scaling portion in the radial orientation, while $\{\phi_i\}$ is the modal displacement vector which incorporates the boundary nodal displacements regarding that mode, and the integration constant c_i expresses the participation of that mode to the resolution.

The solution procedure comprises constructing and solving a quadratic eigenproblem for the subset of n modal displacement vectors $\{\phi_i\}$, which build the columns of the global matrix $[\Phi]$. The solution manner can be converted to a linear eigenproblem at the cost of doubling the number of degree of freedoms. As the number of degree of freedoms in the mesh increases, the computational cost of solving the quadratic eigenproblem grows rapidly.

A linear operator $[L]$ is used to express the relationship between the strains $\epsilon(x, y)$ and the displacement $u(x, y)$. This linear operator $[L]$ can be mapped from Cartesian coordinate into the scaled boundary coordinate system using conventional techniques.

$$[L] = [L^1]\frac{\partial}{\partial x} + [L^2]\frac{\partial}{\partial y} = [b^1(s)]\frac{\partial}{\partial \xi} + \frac{1}{\xi}[b^2]\frac{\partial}{\partial s} \tag{6}$$

The expression of $[L]$, $[L^1]$, $[L^2]$, $[b_1(s)]$ and $[b_2(s)]$ can be found in the Appendix A of Deeks and Wolf papers (Deeks & Wolf, 2002b). This mapping turns the stress field to

$$\sigma(\xi, s) = [D] \sum_{i=1}^{n} c_i \xi^{-\lambda_i - 1} [-\lambda_i [B^1(s)] + [B^2(s)]] \{\phi_i\} \tag{7}$$

where

$$[B^1(s)] = [b^1(s)][N(s)] \tag{8}$$
$$[B^2(s)] = [b^2(s)][N(s)]_{,s} \tag{9}$$

Formula (7) can be re-organised in vector form as follow

$$\sigma(\xi, s) = [D]\Big([B^1(s)][\Phi]\{K_{c\lambda 1}(\xi)\} + [B^2(s)][\Phi]\{K_{c\lambda 2}(\xi)\}\Big) \tag{10}$$

where

$$\{K_{c\lambda 1}(\xi)\} = -[c_1\lambda_1\xi^{-\lambda_1-1}\ c_2\lambda_2\xi^{-\lambda_2-1}\ ...\ c_n\lambda_n\xi^{-\lambda_n-1}]^T \tag{11}$$
$$\{K_{c\lambda 2}(\xi)\} = -[c_1\xi^{-\lambda_1-1}\ c_2\xi^{-\lambda_2-1}\ ...\ c_n\xi^{-\lambda_n-1}]^T \tag{12}$$

Notice that equations (3) and (10) express the vector form of the equations (2) and (7). As in programming environments, a loop implementation is notoriously slow and uses more memory than a vector operations (Tveito, Langtangen, Nielsen, & Cai, 2010). Vectorising those equations above not only improves the speed and effiency of the code, but also make it more like the mathematical expressions which is easier to read, reducing the programming errors, and is often shorter. For example, the equation (2) can be modified to calculate the whole stress field of an local element s_i with $\eta \in [-1, 1]$ and $\xi \in (0, 1]$ by assemblying the shape matrix $[N]$ and coefficience matrix $[K_{c\lambda}]$

$$[N] = \Big[N(\eta_1)\ N(\eta_2)\ ...\ N(\eta_i)\Big]^T \qquad with\ \eta_i \in [-1, 1] \tag{13}$$

$$[K_{c\lambda}] = \begin{bmatrix} c_1\xi_1^{-\lambda_1} & c_2\xi_1^{-\lambda_2} & ... & c_n\xi_1^{-\lambda_n} \\ c_1\xi_2^{-\lambda_1} & c_2\xi_2^{-\lambda_2} & ... & c_n\xi_2^{-\lambda_n} \\ ... & & & \\ c_1\xi_n^{-\lambda_1} & c_2\xi_n^{-\lambda_2} & ... & c_n\xi_n^{-\lambda_n} \end{bmatrix}^T \qquad with\ \xi_i \in (0, 1] \tag{14}$$

The result will be a matrix $n \times m$ where column m^{th} corresponds to ξ_i and row n^{th} corresponds to η_i.

2.2 Traditional and Hierarchical shape function for higher-order elements

Implementing an approximating solution to represent the precise solution requires high-order elements introduced into the formula of the scaled boundary finite element method. There are two techniques to have an additional degree of freedoms in the mesh's element. The first technique termed as the spectral elements approach which the Lagrange procedure is used to interpolate within each element. The second technique called as the hierarchical higher-order element approach which deploys a hierarchical shape function to introduce new DOFs "on top" of the existing one, and therefore preserve the original data structure. In another word, the new shape function of order p consists a single shape function of order p, while it keeps the remain of the lower order shape function.

For simplicity, only two-dimensional elastostatics problem is discussed here to incorporate higher-order traditional and hierarchical elements in the scaled boundary finite element method. For a complete expansion of the scaled boundary finite element method, the paper published by Deeks and Wolf can be used as a reference, another source can be used is the study of Vu and Deeks for a construction of the high order element in the method (Deeks & Wolf, 2002c; Vu & Deeks, 2006).

Traditional shape function

The traditional shape function or Lagrange shape function has been used widely in finite element method and scaled-boundary finite element method (Vu & Deeks, 2006). The shape function is implemented to approximate and interpolate the value of points within each element. Each node in an element has a separated shape function, and the value of that shape function equals one at that node and equals zero at the remained nodes. As this report only focuses on two-dimen+sional problem, therefore it only needs to look at a one-dimensional Lagrange shape function. Given an element with order p, it will has $p+1$ nodes indexed with local coordinate η from $\eta_0,\ \eta_1,\ ...,\ \eta_p$ (with $\eta_0 = -1$ and $\eta_1 = +1$, refer figure 1 for more information). The general shape function can be derived as

$$N_i(\eta) = \prod_{j=0 \cap j \neq i}^{p} \frac{\eta - \eta_i}{\eta_i - \eta_j} \ with\ i = 0\ ...\ p \tag{15}$$

From the equation (15), the shape matrix $[N]$ for a 2-node linear element and 3-node quadratics element can be assembled as follow

$$\text{2-node linear element: } [N(\eta)] = [N_0(\eta)\ N_1(\eta)] \tag{16}$$

$$= \left[\frac{1}{2}(1-\eta)\ \frac{1}{2}(1+\eta)\right] \tag{17}$$

$$\text{3-node quadratics element: } [N(\eta)] = [N_0(\eta)\ N_2(\eta)\ N_1(\eta)] \tag{18}$$

$$= \left[-\frac{1}{2}\eta(1-\eta)\ (1+\eta)(1-\eta)\ \frac{1}{2}\eta(1+\eta)\right] \tag{19}$$

Figure 2 and 3 plot the Lagrange shape functions for $p = 1$ and $p = 2$. One disadvantage of Lagrange shape function is that it is required to re-calculate when a new node is introduced or repositioned. Also, Vu and Deeks experimented that using Lagrange shape function with high order results a poor conditioned matrices (Vu & Deeks, 2006).

Hierarchical shape function

Using the second approach is favoured for p-adaptivity as the procedure enriches the solution by incorporating new features to the existing one while retaining the mesh size the same. Vu and Deeks investigated various types of hierarchical shape functions and recommended the use of the Lobatto polynomials when hierarchical shape functions are desired to save the computational effort (Vu & Deeks, 2006). The second technique

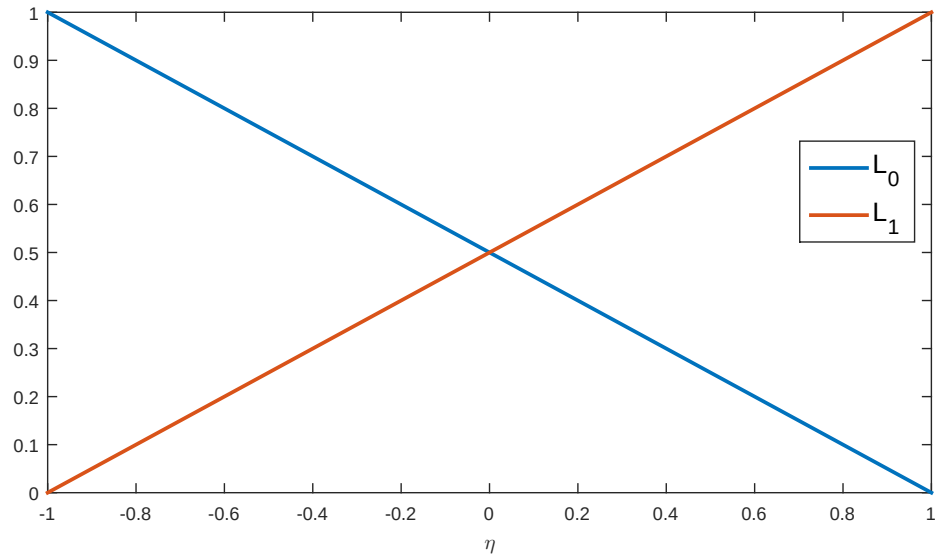

Figure 2: Lagrange shape functions for 2-node linear element

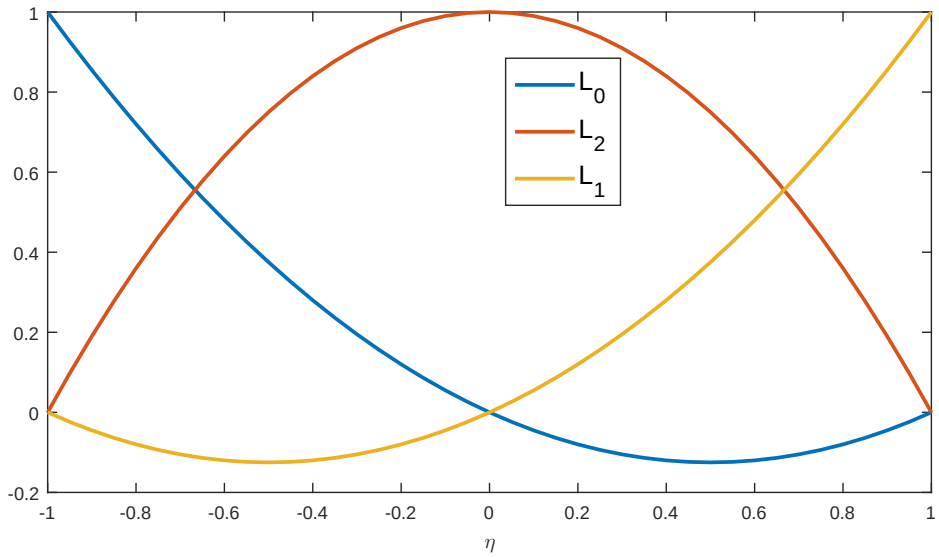

Figure 3: Lagrange shape functions for 3-node quadratics element

has its name as the additional shape function is added to the original functions and therefore preserve the original data structure. However, with h-adaptivity, the mesh size is subdivided in each step, and the order of the shape function alters only one order higher throughout the process. Therefore, both types of approach are implementable.

The hierarchical shape functions or also called Lobatto shape functions of order p is defined as follow (Vu & Deeks, 2006)

$$N_0(\eta) = -\frac{1}{2}(\eta - 1) \tag{20}$$

$$N_1(\eta) = -\frac{1}{2}(-\eta - 1) \tag{21}$$

$$N_k(\eta) = \frac{1}{\|P_{k-1}\|}\int_{-1}^{\eta} P_{k-1}(\psi)d\psi \text{ with } k = 2...p \tag{22}$$

With P_k is Legendre polynomials and can be evaluated by following formula (Demkowicz, 2006)

$$P_0(\psi) = 1 \tag{23}$$

$$P_1(\psi) = \psi \tag{24}$$

$$P_n(\psi) = \frac{2n-1}{n}\psi P_{n-1}(\psi) - \frac{n-1}{n}P_{n-2}(\psi),\ n = 3, ... \tag{25}$$

$$\text{and } \|P_{k-1}\| = \sqrt{\int_{-1}^{1} P_{k-1}^2(\psi)d\psi} \tag{26}$$

With the formula (20-22), the Lobatto shape matrix for a 3-node quadratics element can be assembled as follow

$$[N(\eta)] = [N_0(\eta)\ N_2(\eta)\ N_1(\eta)] \tag{27}$$

$$= \begin{bmatrix} -\frac{1}{2}(\eta - 1) \\ \frac{\sqrt{6}}{4}(x^2 - 1) \\ \frac{1}{2}(\eta + 1) \end{bmatrix} \tag{28}$$

As the scaled boundary method provide an analytical result in the radial direction, only the domain boundary is required to be split and refined. It leads to an advantage of the method is that a two-dimensional problem is able to solve by using one-dimensional boundary elements. In the hierarchical element, while the degree of freedoms at both the end always represents the actual displacement, the addition degree of freedoms no longer

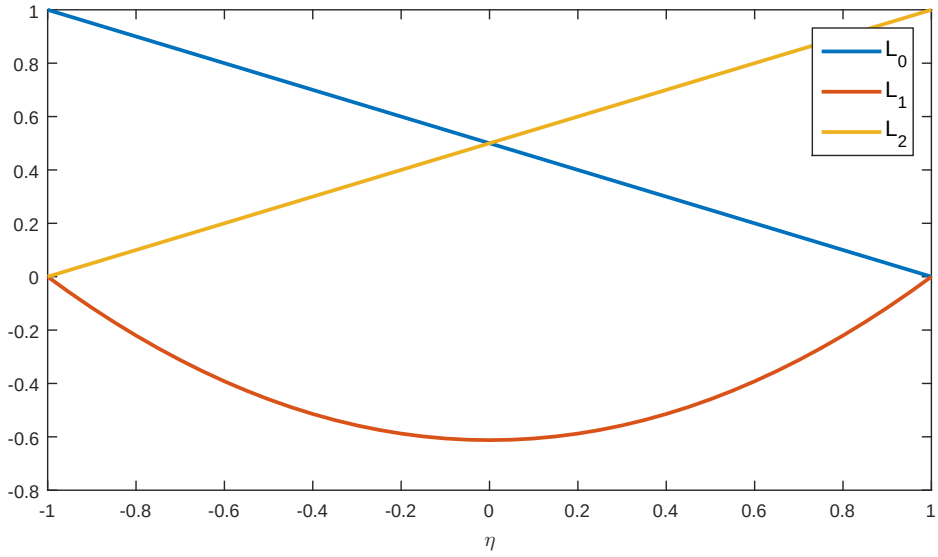

Figure 4: Lobatto shape functions for 3-node quadratics element

resemble the alteration of real movement correlated with the boundary's real nodes. Instead, they are representing the weight of the new added Lobatto shape functions and attached to a virtual node located at $\eta = 0$. Formula (1) can be expressed for order p.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \xi \begin{bmatrix} x_s(-1) & \alpha_2 & \dots & \alpha_p & x_s(+1) \\ y_s(-1) & \beta_2 & \dots & \beta_p & y_s(+1) \end{bmatrix} [N_0(\eta)\ N_2(\eta) \dots N_p(\eta)\ N_1(\eta)]^T \quad (29)$$

The (x_0, y_0) represents for the scaling centre, while the co-ordinates of the intersection between the radial line and the defining curve are represented by $x_s(s)$ and $y_s(s)$. On the other hand, the variables α_k and β_k are zero when a straight line of the border is described by a higher-order element, and they are different than zero as they represent the weights of the shape functions to scope the curve of the boundary. The shape function implemented in the process is restrained to quadratic order for simplification. The study will provide a comparison of the performance between the h-adaptive method based on projection solution and the conventional h-adaptive based on superconvergent patch recovery technique. Similar to the finite element method, the solution field $u(\xi, s)$ is determined by interpolating between these radial nodal nodes.

When using the higher-order Lobatto shape functions, there are various changes

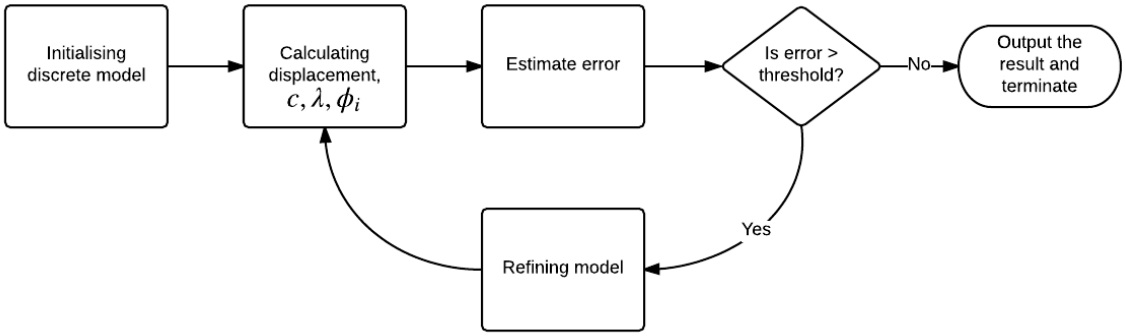

Figure 5: Algorithm for refinement process

which are needed to account in equation (2)

$$u(\xi,\eta) = [u(\xi,-1)\ x_2(\xi)\ ...\ x_p(\xi)\ u(\xi,+1)] \begin{bmatrix} N_0(\eta) \\ N_2(\eta) \\ ... \\ N_p(\eta) \\ N_1(\eta) \end{bmatrix} \tag{30}$$

The coefficients $x_p(\xi)$ are named as the internal degree of freedom which is different with the degree of freedom of the physical displacements of the end nodes. Note that this study solely considers the case of single refinement which means if an element having the total degree of freedom k_{dof} is refined, its total degree of freedom only increase by one $k_{dof} + 1$.

2.3 Refinement process

A refinement process or adaptivity process is a well known technique to achieve a better accuracy of a discrete model (it could be a finite element or scaled boundary finite element model). It requires an adequate level of refinement to obtain a desirable accuracy. The solution becomes more accurate when the model being discretised in higher level. Basically, an adaptivity process needs an error estimator and a error threshold to automatically drive the process to meet the specific requirement (O. Zienkiewicz & Taylor, 2000; St Doltsinis, 1987).

Figure 5 shows a basic algorithm and its components. The process begins with a minimum description of the problem and an error target or threshold. Then it performs the analysis to obtain the solution such as displacement, c, λ, *and* ϕ. From this solution, the process estimates the error according to the error estimator implemented. If the current error is higher than the target error, the algorithm selects the elements that contribute high inaccuracies to refine. The process then repeat again until the current error meets the target, then it stops and return the result.

Deeks and Wolf has covered numerous error estimators for implementing in refinement procedures for scaled-boundary finite element problems (Deeks & Wolf, 2002b). In their work, they propose using superconvergent patch recovery to construct the error estimator. Details of this method will be covered in the coming sections. Another method to implement the error estimator is using projection-based method and reference solution as Vu and Deeks did in their work for p-hierarchical adaptive procedure (Vu & Deeks, 2008). In this project, the same method will be implemented in h-adaptive procedure and compare with superconvergent patch recovery method.

Another important component in adaptive process is the refinement method. The refinement methods can be categorized regarding to its effect on various characteristic of the mesh. For example, h-method reduces the size of the elements that have higher error by introducing new nodes (Krishnamoorthy & Umesh, 1993); p-method increases the order of the polynomial shape function while retaining the same configuration of the elements (Gui & Babuska, 1985); r-method alternates and redistributes nodes' position to high error areas (Kita, Higuchi, & Kamiya, 2000). There are also combining methods such as h-p methods which can simultaneously add new nodes and increase the polynomial order or h-r method (Gui & Babuska, 1985; Kita et al., 2000). This project will implement h-method and reference solution in adaptivity procedure.

There are two types of h-methods which are global re-meshing type and h-hierarchical type. In h-hierarchical methods, when refining a coarse mesh to obtain a finer mesh, the basic structure of the shape function is reserved (Cramer, Rudolph, Steinl, & Wunderlich, 1999). Similar to p-hierarchical, the adaptivity process introduces an additional auxiliary nodes to the original mesh which corresponds to add a h-hierarchical shape function into the original shape function. There are various researchs developing the shape functions for h-hierarchical method, especially for boundary finite element method (Guiggiani & Lombardi, 1992; Charafi, Neves, & Wrobel, 1995).

In global re-meshing type, rather than adding an additional shape function, it will re-create the whole mesh based on the basic definition and the error indicators. This is the most popular method due to its ease in implementing to the adaptivity process for analysis. This project will also adopt this method.

2.4 Error estimation and Reference solution

As mentioned, it is crucial to estimate the error between the exact solution $\{u_{exact}(\xi, s)\}$ and the discretised solution $\{u_h(\xi, s)\}$. This error reflects the reliability and accuracy of the solution, and can become an indicator to drive the adaptivity process in finding a better mesh. However, this exact solution is usually unavailable in a practical problem, and therefore, it is popular to substitute the exact solution $\{u_{exact}(\xi, s)\}$ by an approximate solution $u(\xi, s)$ in (31) (Deeks & Wolf, 2002b).

$$\{u_{exact}(\xi, s)\} \approx \{u(\xi, s)\} \tag{31}$$

Modelling the unknown exact solution is essential to evaluate the error in the current solution as the discrepancy between the exact and current solution illustrates the

contribution of each element to the error. To obtain a better approximation of the exact solution, Deeks and Wolf chose to use the recovery-based estimators and recovered the nodal stresses on the scaled boundary by employing the superconvergent patch recovery method, which was developed by Zienkiewics and Zhu (O. C. Zienkiewicz & Zhu, 1992; Deeks & Wolf, 2002b). This technique is based on the premise that an element has points where stresses converge at a similar rate as the displacement field does, and at a higher rate than any other points within the same element. It has been shown that the stresses are superconvergent at the Gauss points (Zhu & Zienkiewicz, 1990). The procedure attempts to use the stresses at the Gauss points to recover the nodal stresses, and then use the shape function to interpolate the whole stress field. The resulting stress field is smooth and can be used to obtain an error estimator to drive the adaptive process.

Using the reference solution can obsolete the requirement for a stress recovery procedure. To represent the exact solution, the reference solution $u_{ref}(\xi, s)$ is required to be closer to the exact solution $u(\xi, s)$ than the coarse mesh solution. In the p-adaptive process, the reference solution $u_{ref}(\xi, s)$ can be achieved by uniformly increasing the order of shape function by one (Demkowicz et al., 2002). The solution $u_{fine}(\xi, s)$ of the fine mesh is made to coincident with the reference solution $u_{ref}(\xi, s)$. The same idea can be applied in h-adaptive as each element of the coarse mesh will be separated into two children to obtain the reference solution. It is noted that the number degree of freedom of each element also increases one which is the same with the p-methods. Within each iteration, despite a generally better solution obtained from the fine mesh, the optimal mesh is the main mesh that the adaptivty process works on. Since using the fine mesh to advance the process will lead to a uniform refinement, and therefore, will not reduce the computational expense. As Vu and Deeks mentioned in their work that this process takes one step forward then half a step back in the direction of the optimal refinement. At the end of the procedure, the final fine mesh is the final solution of the problem (Vu & Deeks, 2007).

As mentioned above, the superconvergent patch recovery approximates the exact solution by a "smoother" version of the current solution. This "smoother" solution still has the same configuration as the current solution. On another hand, the fine mesh used as the reference solution has one degree of freedom more in each element comparing to the coarse mesh. Therefore, the solution of fine mesh needs to be projected to the coarse mesh to presents the effect of coarse mesh on the solution. The projection process also means that there is losing information in the projecting result, and it is desirable to retain the information as much as possible (Demkowicz, 2006). Mathematically, the best projection solution $u_{project}$ should minimise the L^2 norm of the distance between it and the function of the reference solution u_{ref} (32)

$$\|u_{ref} - u_{projection}\| \rightarrow min \tag{32}$$

More details to derive the projection solution will be introduced in the next chapter.

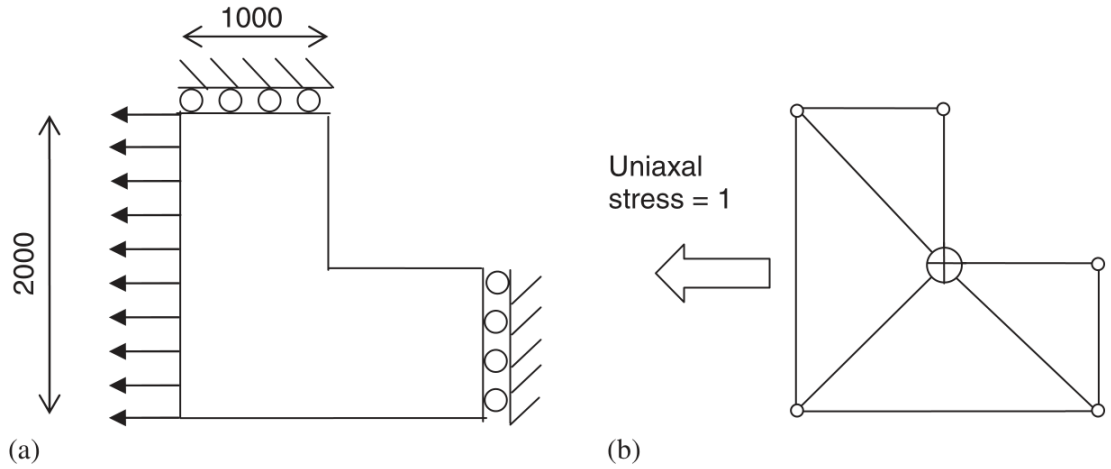

Figure 6: Square hole inside the square plate problem

2.5 Examples

In order to test and compare the result from refinement process, this report will use two well known example problems. The first one is the plane stress problem, and the second one is a plain strain problem.

Figure 6 illustrates the problem which includes a square plate with a square hole undertaken a uniformly distributed force on the horizontal axis. The scaling centre is situated in the corner to take advantage of the method and reduce the number of elements. The constants using in the problems are the Poisson ratio $\upsilon = 0.3$, the Young modulus $E = 250,000 Nmm^{-2}$ and the target error is 3%. The dimension of the mesh will set in mm, and the uniaxial tension will be specified in $\frac{N}{mm}$.

The next example is a plane strain square plate subjected a vertical load. The square plate has dimension 1000×1000 and under a uniform vertical patch load $1\frac{N}{mm}$. The plate is modeled with a scaling center at the origin $(0, 0)$, with two side faces and the boundary being discretised by nine nodes. Those nodes are numbered counter clockwise. The Young modulus $E = 250,000 Nmm^{-2}$ and the Poisson ratio $\upsilon = 0.3$. Figure 7 illustrates this problem.

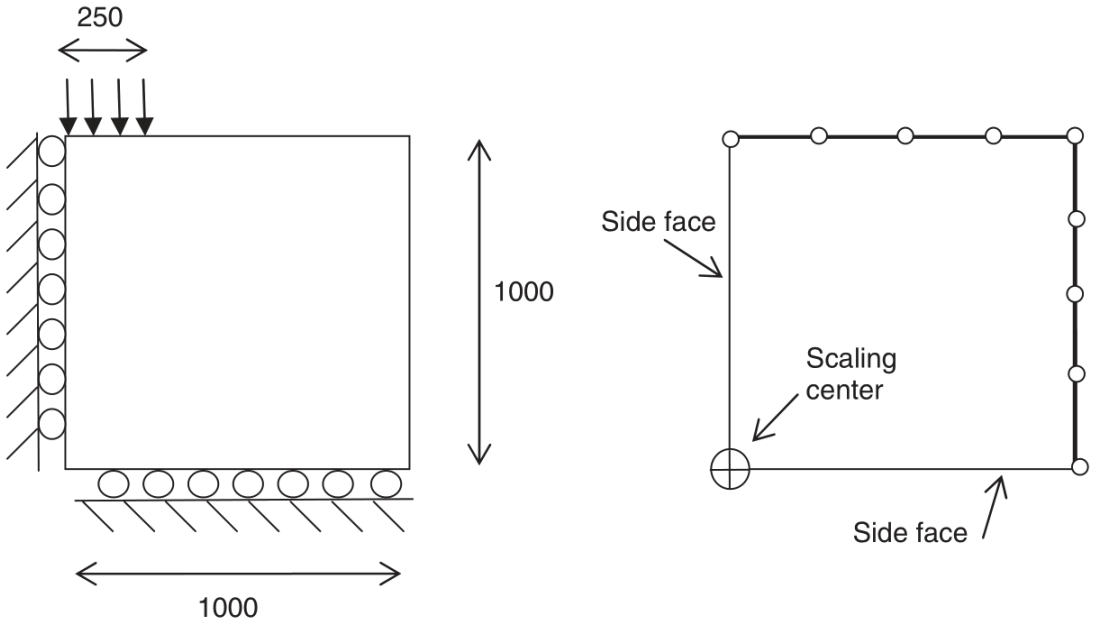

Figure 7: Plane strain square plate with a vertical load

3 Proposed Approach

To evaluate the effiency of projection-based method in h-adaptivity, this report will first attempt to implement the superconvegent patch recovery. This method is widely used in finite element method and deployed to scale boundary finite element problem by Deeks and Wolf (Deeks & Wolf, 2002a). As superconvegent patch has been described in details in Deeks and Wolf's work, this report only walk through the implement steps and formula works. Finally, the report developes the theory to obtain the projecting solution from the reference solution, and form the formula to calculate the error estimator.

3.1 Superconvergent patch recovery method in h-adaptivity

Superconvergent patch recovery first calculates the stress at Gauss points within an element which are proved that has a faster converged rate or more accuracy (O. C. Zienkiewicz & Zhu, 1992). Then from those sampling points, it applies least square method to interpolate the other nodes. Finally, from the recovered nodes, the shape functions will be used to regenerate the stress field. Figure (8) illustrates the Gauss points in both linear element and quadratics element, and the nodal value recovered by the procedure.

Note that, most of the work done in this topics used the traditional Lagrange shape functions, however, in this report, the Lobatto shape functions is used to utilise its advantages in calculation. As Lobatto shape function is a hierarchical shape function, the high-order nodes adding into the element is an auxiliary node and have no physical meaning. Hyung-Seok and Batra has used this p-hierarchical shape function for the

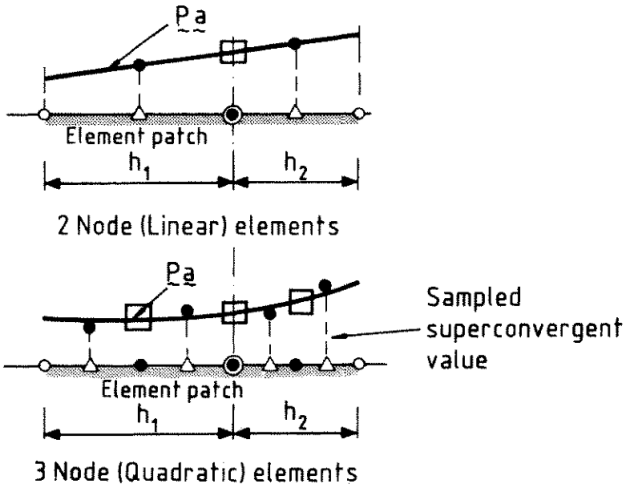

Figure 8: One dimension with 2-Node elements and 3-Node elements: △ Superconvergent Gauss point; ⊙ Patch assembly point; [] Nodal values determined by recovery procedure; Image courtesy of (O. C. Zienkiewicz & Zhu, 1992)

superconvergent patch recovery with an assumption that the stress of the hierarchical element at the optimal point is also accurate and confirmed their statement and the end of the report (Oh & Batra, 1999).

Least square fitting

To recovery the nodal values from Gauss points, the least square method is used to fit a polynomial to the Gauss points then interpolate the value of the nodal nodes. The procedure for recovering nodal nodes for quadratic elements is described below

$$\sigma^* = Pa \tag{33}$$

With σ^* is the nodal nodes, P is the coordinate of the nodal nodes and a is the coefficient matrix. Matrix P for a quadratic element can be assembled as

$$P = [1 \; X_{node} \; X_{node}^2] \tag{34}$$

And matrix a can be computed by

$$K = [1 \; X_{sample} \; X_{sample}^2] \tag{35}$$

$$A = K^T K \tag{36}$$

$$b = K^T Y \tag{37}$$

$$\rightarrow a = A^{-1} b \tag{38}$$

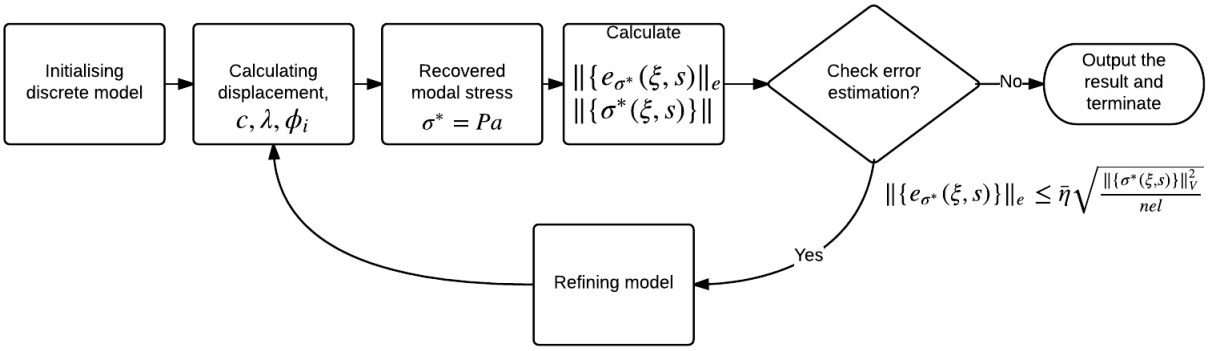

Figure 9: Flow chart of h-refinement using superconvergent patch recovery

Note that, P and K can be normalised to simplify the calculation. Although matrix a requires more calculation to assemble the matrix, it only need to compute once for each patch, and the calculation is mainly vector operation, therefore the calculation will be fast and efficient.

Error estimation in the scaled-boundary finite element method

Figure (10a) and (11a) show raw stress fields of the two examples introduced in previous section. It is noted that while the stress is continuous along the radial direction within an element, it is discontinuous across the radial lines which corresponds to the inter-element nodes between two elements. Deeks and Wolf suggests that the recovery process should be done along the circumferential direction to produce a smooth stress field (Deeks & Wolf, 2002b). For nodal nodes located at the sharp conner, it is recommended that these nodes is calculated independently from two adjoining element, then its two estimates can be averaged. Finally, as the main difference between finite element method and scaled boundary finite element is that scaled boundary method's solution is combined from displacement modal values. Therefore, the stress recovery need to be applied independently for each mode but with the same least square coefficient matrix for all the modes (Deeks & Wolf, 2002b).

If $\{\sigma_i^*\}$ is the nodal stress value recovered from the least square process. The smooth stress field then can be calculated from the recovered nodal stress by fomula

$$\{\sigma^*(\xi,s)\} = [D]\sum_{i=1}^{n} c_i \xi^{-\lambda_i - 1}\{\sigma_i^*\} \tag{39}$$

Forming an error estimation is the next step after obtaining the smooth stress field. A details proof was carefully described in Deeks and Wolf's papers (Deeks & Wolf, 2002a), this report only refers the formula to evaluate the error estimation. In the conventional h-adaptive method, the process attemps to obtain an optimal mesh by ensuring that all the elements within share the same distribution to the error. The criteria can be

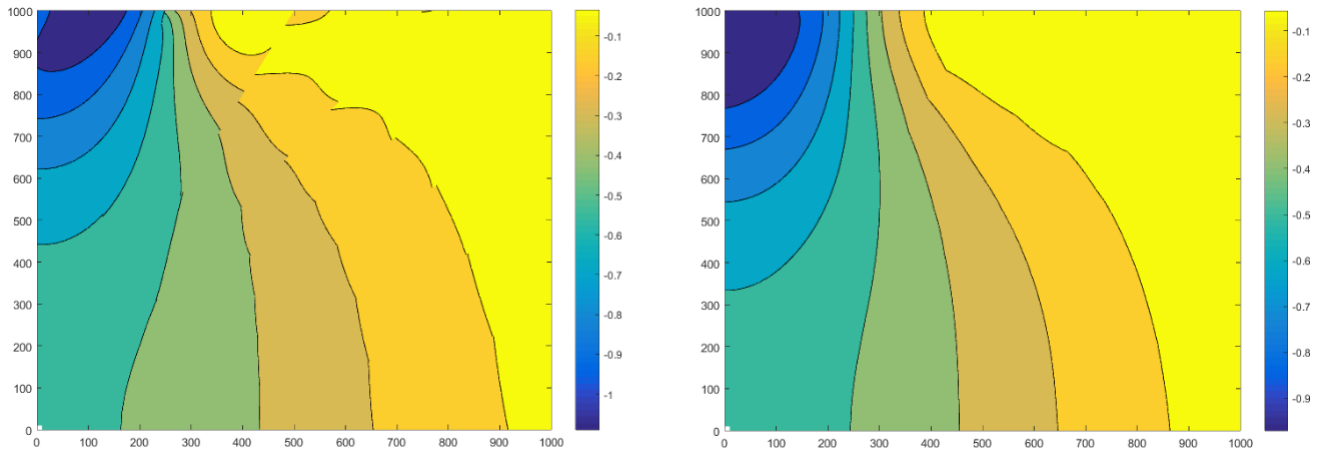

Figure 10: a) Raw stress of square plate (left) b) Smooth stress field after recovery process (right)

expressed the following form (Deeks & Wolf, 2002a)

$$\|\{e_{\sigma^*}(\xi,s)\}\|_e \leq \bar{\eta}\sqrt{\frac{\|\{\sigma^*(\xi,s)\}\|_V^2}{nel}} \tag{40}$$

In this equation, the $\|\{e_{\sigma^*}(\xi,s)\|_e$ represents the contribution of each element into the stress error, the term $\|\{\sigma^*(\xi,s)\}\|$ indicates the total energy norm, nel is the number of element over the domain, and $\bar{\eta}$ is the target error. During the process, if an element is failed to meet the requirement specified by 40, it will be selected for refinement. The energy norm and error energy norm can be achieved by

$$\|\{\sigma^*(\xi,s)\}\| = \mp\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{c_i c_j}{\lambda_i+\lambda_j}\int_S \{\sigma_i^*(s)\}^T[D]^{-1}\{\sigma_j^*(s)\}|J|ds} \tag{41}$$

$$\|\{e_{\sigma^*}(\xi,s)\}\| = \mp\sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{c_i c_j}{\lambda_i+\lambda_j}\int_{S_e} \{e_{\sigma^* i}^*(s)\}^T[D]^{-1}\{e_{\sigma^* j}^*(s)\}|J|ds} \tag{42}$$

The h-adaptivity procedure using superconvergent patch recovery can be summarise in figure (9). Figure (10) and (11) show the results of smoothing method over two problems which are a square plate with a square hole and a plane strain square plate.

3.2 Projecting solution technique and criteria for refinement

Projection-based interpolation is a technique which can locate the optimal representation of a given function that can be conducted by a particular discretized model. This technique is used to project the reference solution onto desired mesh. It is common that the exact solution is unavailable in practical problems, therefore, the reference solution

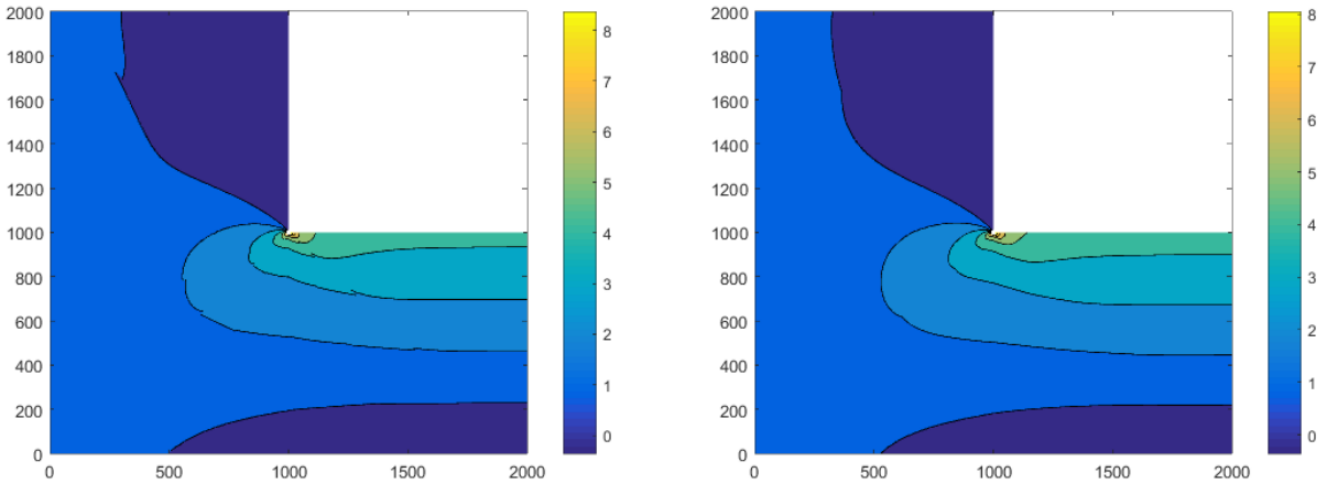

Figure 11: a) Raw stress of square plate with square hole(left) b) Smooth stress field after recovery process (right)

$u_{ref}(x)$ is necessary to replace for the exact solution u(x). In the case of h-adaptivity, the solution of a fine mesh, which has two sub-elements split from each original element, is selected as the reference solution, or $u_{ref}(x) \approx u_{fine}(x)$. Projecting the reference solution on the current coarse mesh $u_{fine\_coarse}(x)$ is the first step to compute the error decrease (Vu & Deeks, 2008).

There are three properties of projection-based interpolant $u_{fine\_coarse}(x)$ (Demkowicz, 2006):

Locality $u_{fine\_coarse}(x)$ is computed element by element using values of function $u_{fine}(x)$ over the element iel. This allows the participating of each element to the error decrease to be calculated element by element.

Optimality The residual between the function $u_{fine}(x)$ and the projection-based interpolant $u_{fine\_coarse}(x)$ is minimum in an appropriate norm $\|\|_{iel}$.

Global continuity The result of local projections has to be ensure the global continuity which means it is free to determine the middle node (or nodes as higher order), however the vertex nodes have to be intact to prevent the effect on the neighbouring elements.

A constrained minimization problem can be established to solve for the projection-based interpolant based on the three properties discussed above

$$\int_a^b (u(x) - u_{project}(x))^2_{,x} dx \rightarrow min \tag{43}$$

with

$$u_{project}(x) \in l_x, \quad where\ l_x = [a, b]$$
$$u_{project}(x) = u(x),\ at\ x = a, x = b$$

where $u(x)$ indicates the exact solution and $u_{project}(x)$ represents the projected solution. Equation (43) can be formed for local element and replaces the exact solution $u(x)$ by $u_{fine}(x)$, a discrete minimization problem is derived as

$$\int_{-1}^{1} (u_{fine}(\eta) - u_{fine\_coarse}(\eta))^2_{,\eta} d\eta \rightarrow min \tag{44}$$

with

$$u_{fine\_coarse}(\eta) \in l_\eta, \; where \; l_\eta = [-1, 1]$$
$$u_{project}(\eta) = u(\eta), \;\; at \; \eta = -1, \eta = 1$$

The local projection-based interpolant $u_{fine\_coarse}(\eta)$ is the reference solution optimally projected onto the element, and can be recovered by determining the minimization of each element in the coarse mesh. For a linear element, the projection is specified by the values of the fine solution at the end nodes. For higher order element, the degrees of freedom at the internal node (at $\eta = 0$) are determined to minimise the integral above.

The error decrease formed by refining any particular element *iel* is

$$\begin{aligned} \Delta err_{iel} &= \int_{-1}^{1} ((u - u_{exact\_coarse})^2_{iel,\eta} - (u - u_{exact\_fine})^2_{iel,\eta}) d\eta \\ &\cong \int_{-1}^{1} ((u_{fine} - u_{fine\_coarse})^2_{iel,\eta} - (u_{fine} - u_{fine\_fine})^2_{iel,\eta}) d\eta \\ &\cong \int_{-1}^{1} ((u_{fine} - u_{fine\_fine})^2_{iel,\eta}) d\eta \end{aligned} \tag{45}$$

When element *iel* is refined from coarse to fine, its error decrease is described by the first line of (45), and involves calculating the exact solution. As the solution is not available, it is approximated by the reference solution which is substituted into the second line. The optimal projection-based interpolant of the fine solution on the fine mesh is the fine solution. Therefore, the last term of the second line will be zero. As a result, there is the simplified representation on the final line.

In each iteration of the adaptive procedure, the refinement criteria are used to select the elements that required to be refined to optimise the mesh. The concept is the element having the significant contribution to the global error decrease rate will be chosen.

$$\Delta err_{iel} > \frac{1}{3} \Delta err_{max} \tag{46}$$

Conventionally those elements satisfying (46) are refined (Demkowicz et al., 2002), where Δerr_{max} is the maximum element error decrease in the mesh. In order to identify

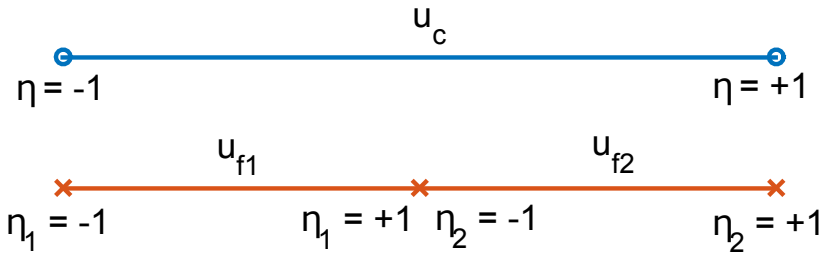

Figure 12: Coarse element u_c and its corresponding fine u_f

elements of high error magnitude, an additional refinement criteria was suggested to account for the actual magnitude of the projection-based error appearing in the coarse solution (Solin, Segeth, & Dolezel, 2003).

$$err_{iel} > 10 err_{ave} \tag{47}$$

With err_{ave} is the average element projection-based error magnitude. When a mesh includes several elements having high error decrease, it will distort the criteria introduced in (46) and might return only a few elements refined. The criteria in (47) was developed to forestall the problem. The last line of (45) indicates that the error decrease Δerr_{iel} matches with the estimated error magnitude err_{iel}, resulting in an alternate form of refinement criteria specified in (47).

$$\Delta err_{iel} > 10 \Delta err_{ave} \tag{48}$$

It is evident from the above analysis that the projection-based interpolation technique presents an essential role in forming the refinement criteria to drive the mesh refinement following the steepest decrease approach.

3.3 Projection-based recovery method

This section describes briefly the procedure to project the reference solution u_{fine} onto the coarse mesh. It consists a development of the projection-based technique for the one-dimensional finite element method, and extending to two-dimensional scaled boundary finite element method. The process is inspired from the work of Vu and Deeks in projection-based interpolant for p-adaptive process (Vu & Deeks, 2008). Note that, this implement chose Lobatto shape function as the main shape function during the process.

This projection can be conducted through each element by addressing the discrete minimization problem introduced in (44). In the manner of h-adaptivity, the coarse mesh is uniform refined to obtained the fine mesh. Figure 12 demonstrates a coarse element and its corresponding fine element. As the configuration of the element is not the same anymore, therefore (44) also need to be modified to account for this change.

$$u_0 = u_{f1}(-1),\ u_1 = u_{f2}(1) \text{ with } u_0, u_1 \in u_c \tag{49}$$

$$\int_{-1}^{1} (u_{project} - u_f)^2_{,\eta} d\eta \to min \tag{50}$$

Recall the three properties discussed in the previous sections to form a projection interpolant, equation (49) satisfies the Global continuity by directly projecting vertex nodes of fine elements to coarse element. From figure 12, it is clear that the first vertex node $\eta = -1$ of u_{f1} and second vertex node $\eta = +1$ of u_{f2} are coincident with two vertex nodes of u_c. Equation (50) also satisfies the Locality and Optimality properties. Differentiating the left hand side of (50) subsequently with respect to each variable $u_k \in u_c$ and setting the equation to zero

$$\int_{-1}^{1} (u_{project} - u_f)_{,\eta} [N_i]_{,\eta} d\eta = 0 \tag{51}$$

From equation (30), $u_{project}, u_{f1}, u_{f2}$ can be expressed as follow

$$\begin{aligned} u_{project} &= u_c(-1)N_0 + u_c(1)N_1 + \sum u_k N_k && -1 < \eta < 1 \\ u_{f1} &= u_{f1}(-1)N_0 + u_{f1}(1)N_1 + \sum u f_{1k} N_k && -1 < \eta < 0 \\ u_{f2} &= u_{f2}(-1)N_0 + u_{f2}(1)N_1 + \sum u f_{2k} N_k && 0 < \eta < 1 \end{aligned}$$

Multiplying out and moving the known terms to the right hand side, notice that $u_0, u_1 \in u_c$ and alreay determined in the analysis process, (51) becomes

$$\sum_{k=2}^{p} (\int_{-1}^{1} [N_i]_{,\eta} [N_k]_{,\eta} d\eta) u_k = \int_{-1}^{1} (uf - u_0 * N_0 - u_1 * N_1)_{,\eta} [N_i]_{,\eta} d\eta \tag{52}$$

If the Lobatto shape functions are implemented, the derivative of shape functions, which are Legendre polynomials, are orthogonal

$$\int_{-1}^{1} (N_i(\eta))_{,\eta} (N_j(\eta))_{,\eta} d\eta = 0 \text{ with } i \neq j \tag{53}$$

Multiplying out and applying the (53) to (52)

$$\sum_{k=2}^{p} (\int_{-1}^{1} [N_i]_{,\eta} [N_k]_{,\eta} d\eta) u_k = \int_{-1}^{1} (uf)_{,\eta} [N_i]_{,\eta} d\eta \tag{54}$$

The big element uf can be split into the sum of two child element u_{f1} and u_{f2}

$$\sum_{k=2}^{p}(\int_{-1}^{1}[N_i]_{,\eta}[N_k]_{,\eta}d\eta)u_k = \int_{-1}^{0}(u_{f1})_{,\eta}[N_i]_{,\eta}d\eta + \int_{0}^{1}(u_{f2})_{,\eta}[N_i]_{,\eta}d\eta \tag{55}$$

The variable can be changed to obtain the integral from $[-1, 1]$

$$\begin{aligned}\sum_{k=2}^{p}(\int_{-1}^{1}[N_i]_{,\eta}[N_k]_{,\eta}d\eta)u_k = \int_{-1}^{1}(uf_1(\eta))_{,\eta}[N_i(\frac{\eta}{2}-\frac{1}{2})]_{,\eta}d\eta \\ + \int_{-1}^{1}(u_{f2}(\eta))_{,\eta}[N_i(\frac{\eta}{2}+\frac{1}{2})]_{,\eta}d\eta\end{aligned} \tag{56}$$

Finally, turn the right hand side to matrix

$$\begin{aligned}\sum_{k=2}^{p}(\int_{-1}^{1}[N_i]_{,\eta}[N_k]_{,\eta}d\eta)u_k = \sum_{j=0}^{p}(\int_{-1}^{1}[N_j]_{,\eta}[N_i(\frac{\eta}{2}-\frac{1}{2})]_{,\eta}d\eta)\{u_{f1}\} \\ + \sum_{j=0}^{p}(\int_{-1}^{1}[N_j]_{,\eta}[N_i(\frac{\eta}{2}+\frac{1}{2})]_{,\eta}d\eta)\{u_{f2}\}\end{aligned} \tag{57}$$

with

$$uf_1 = \begin{bmatrix} uf_1(-1) \\ uf_1(1) \\ uf_1(2) \\ ... \\ uf_1(p) \end{bmatrix} \tag{58}$$

$$u_{f2} = \begin{bmatrix} u_{f2}(-1) \\ u_{f2}(1) \\ u_{f2}(2) \\ ... \\ u_{f2}(p) \end{bmatrix} \tag{59}$$

In equation (57), the left hand side $\sum_{k=2}^{p}(\int_{-1}^{1}[N_i]_{,\eta}[N_k]_{,\eta}d\eta) = 0$ when $k \neq i$ and $\sum_{k=2}^{p}(\int_{-1}^{1}[N_i]_{,\eta}[N_k]_{,\eta}d\eta) = 1$ when $k = i$ due to Lobatto's properties (53). This propertie makes (57) more straightforward to evaluate, the equation (57) can expressed in words as the middle node of the projection interpolant can be calculated by the linear combination

of the two corresponding fine elements' nodes. Note that Finally, the projection-based displacement field $\{u_{fine\_coarse}\}$ and the projection-based stress field $\{\sigma_{fine\_coarse}(\xi, s)\}$ can be recovered by these equations

$$\{u_{fine\_coarse}(\xi, s)\} = [N_{coarse}(s)] \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}} \{\phi_{fine\_coarse,i}(s)\} \tag{60}$$

$$\begin{aligned} \{\sigma_{fine\_coarse}(\xi, s)\} = [D] \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} [-\lambda_{fine,i} [B^1_{coarse}(s)]] \\ + [B^2_{coarse}(s)] \{\Phi_{fine\_coarse,i}\} \end{aligned} \tag{61}$$

The equations (60) and (61) have the same forms as the equations developed for the displacement and stress in the basic scaled boundary finite element method.

3.4 Energy norm based error estimator

For 2D elastostatic problems, error indicators based on minimising the error for the displacement field are not favourable, but an error estimator based on the stress field $\sigma(\xi, s)$. The energy norm of the difference between the fine mesh stress field $\sigma_{fine}(\xi, s)$ and the coarse mesh stress field $\sigma_{coarse}(\xi, s)$ indicates the error of the current coarse mesh. Referring to Equation (7), the stresses in the fine mesh and the coarse mesh are recovered by

$$\begin{aligned} \sigma_{fine}(\xi, s) = [D] \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} [-\lambda_{fine,i} [B^1_{fine}(s)]] ... \\ + [B^2_{fine}(s)] \{\Phi_{fine,i}\} \end{aligned} \tag{62}$$

$$\begin{aligned} \sigma_{coarse}(\xi, s) = [D] \sum_{i=1}^{n_{coarse}} c_{coarse,i} \xi^{-\lambda_{coarse,i}-1} [-\lambda_{coarse,i} [B^1_{coarse}(s)]] ... \\ + [B^2_{coarse}(s)] \{\Phi_{coarse,i}\} \end{aligned} \tag{63}$$

In the equations (62) and (63), n_{fine} and n_{coarse} represent the number of DOFs arising in the fine mesh and coarse mesh respectively. As the fine mesh is a result of uniformly h-refining the coarse mesh by one order, the relationship between n_{fine} and n_{coarse} is described by the equation

$$n_{fine} = n_{coarse} \tag{64}$$

However, regarding to the figure 12, within the same coarse element, there are two corresponding fine elements. This is the main difference between the h-adaptivity and p-adaptivity. In h-adaptivity, the polynomial order does not change but the number of fine element per coarse element. Therefore, it is critical to update the error energy norm

accounting this difference. The energy norm of the stress field over a finite volume V represents a weighted root-mean-square of the stresses and is defined as

$$\|\sigma(\xi,s)\|_V = \left(\int_V \sigma(\xi,s)^T \epsilon(\xi,s) dV \right)^{\frac{1}{2}} = \left(\int_V \sigma(\xi,s)^T [D]^{-1} \sigma(\xi,s) dV \right)^{\frac{1}{2}} \tag{65}$$

with

$$dV = |J| \xi d\xi ds \tag{66}$$

Hence, if the approximation of the point-wise stress error is defined as

$$e_\sigma(\xi,s) = e^*_\sigma(\xi,s) = \sigma_{fine}(\xi,s_f) - \sigma_{coarse}(\xi,s) \tag{67}$$

Note that $\sigma_{fine}(\xi,s_f)$ here is defined to account for every fine elements fit in the same original coarse element. The parameter of $\sigma_{fine}(\xi,s_f)$ is s_f which can be map from $s :\rightarrow s_f$. The energy norm of the discretization error in the stress field is

$$\|e^*_\sigma(\xi,s)\|_V = \left(\int_V e^*_\sigma(\xi,s)^T [D]^{-1} e^*_\sigma(\xi,s) dV \right)^{\frac{1}{2}} \tag{68}$$

The error estimator is calculated as

$$\eta^* = \frac{\|e^*_\sigma(\xi,s)\|_V}{\|\sigma_{fine}(\xi,s)\|_V} \times 100\% \tag{69}$$

The terms defined in (69) is close to the calculation of the energy norm of the difference between the raw stress field and there covered stress field introduced by Deeks and Wolf (Deeks & Wolf, 2002b). By specifying an error tolerance $\bar{\eta}$ which $(0 < \bar{\eta} < 1)$, the terms (69) can drive the adaptivity process until it meets the target error, i.e. $\eta^* \leq \bar{\eta} \times 100\%$. Next section will illustrates the evaluation of the element projection-based error decrease rate Δerr_{iel}.

3.5 Formulating the new refinement criteria for the scaled boundary finite element method

The error decrease presented by element *iel* specified in (45) is capable to calculate in energy norm due to the relation between the displacement u and σ presenting in equation (10). Therefore, formula (45) becomes

$$\Delta err_{iel} = \|\sigma_{fine}(\xi,\eta_f) - \sigma_{fine\_coarse}(\xi,\eta)\|^2_{iel} \tag{70}$$

Note that the function above calculate with σ_{fine} at η_f. This is also the participation of the element to the energy norm of the projection-based error measurement,

$$err_{iel} = \|\sigma_{fine}(\xi, \eta_f) - \sigma_{fine\_coarse}(\xi, \eta)\|^2_{iel} \tag{71}$$

From (61), the expression for which the norm is considered is

$$\begin{aligned}\sigma_{fine}(\xi, \eta_f) - \sigma_{fine\_coarse}(\xi, \eta) =& [D] \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} [-\lambda_{fine,i} [B^1_{fine}(\eta_f)] \\ &+ [B^2_{fine}(\eta_f)]] \{\phi_{fine,i}\} \\ &- [D] \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} [-\lambda_{fine,i} [B^1_{coarse}(\eta)] \\ &+ [B^2_{coarse}(\eta)]] \{\phi_{fine\_coarse,i}\}\end{aligned} \tag{72}$$

The radial terms in the expression above can be taken out as common terms, the function can be simplified to

$$\sigma_{fine}(\xi, \eta_f) - \sigma_{fine\_coarse}(\xi, \eta) = \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} \{\Delta_{\sigma,i}(\eta)\} \tag{73}$$

where

$$\begin{aligned}\{\Delta_{\sigma,i}(\eta)\} = [D] \Big\{ & [-\lambda_{fine,i} [B^1_{fine}(\eta_f)] + [B^2_{fine}(\eta_f)]] \{\phi_{fine,i}\} \\ & - [-\lambda_{fine,i} [B^1_{coarse}(\eta)] + [B^2_{coarse}(\eta)]] \{\phi_{fine\_coarse,i}\} \Big\}\end{aligned} \tag{74}$$

Substituting (73) into (70) and multiplying out the summations, the energy norm of element discrete error becomes

$$\begin{aligned}\Delta err_{iel} &= \int_{\xi_0}^{\xi_1} \int_{-1}^{1} \left\{ \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} \{\Delta_{\sigma,i}(\eta)\} \right\}^T [D]^{-1} \\ &\qquad \times \left\{ \sum_{i=1}^{n_{fine}} c_{fine,i} \xi^{-\lambda_{fine,i}-1} \{\Delta_{\sigma,i}(\eta)\} \right\} \xi |J| d\xi ds \\ &= \sum_{i=1}^{n_{fine}} \sum_{j=1}^{n_{fine}} c_{fine,i} c_{fine,j} \int_{\xi_0}^{\xi_1} \xi^{-\lambda_{fine,i}-1-\lambda_{fine,j}-1} \\ &\qquad \times \left(\int_{-1}^{1} \{\Delta_{\sigma,i}(\eta)\}^T [D]^{-1} \{\Delta_{\sigma,j}(\eta)\} |J| d\eta \right) d\xi\end{aligned} \tag{75}$$

For bounded domain, the integration calculates from $\xi_0 = 0$ to $\xi_1 = 1$ and the real parts of λ are all non-positive, while in the unbounded case, the limits are $\xi_0 = 1$ and $\xi_1 = \infty$ and the the λ are all non-negative. The calculation also excludes the rigid body modes and modes for restrained side faces, which have eigenvalues $\lambda_i = 0$, and the boundary stresses for these modes are zeros at all point. Therefore, the denominator term $(\lambda_{fine,i} + \lambda_{fine,j})$ is always different from zero. The equation (75) can be expressed in the same form for both bounded and unbounded domain with the signs corresponding to the bounded and unbounded cases respectively.

$$\Delta err_{iel} = \mp \sum_{i=1}^{n_{fine}} \sum_{j=1}^{n_{fine}} \frac{c_{fine,i} c_{fine,j}}{\lambda_{fine,i} + \lambda_{fine,j}} \times \int_{-1}^{1} \{\Delta_{\sigma,i}(\eta)\}^T [D]^{-1} \{\Delta_{\sigma,j}(\eta)\} |J| d\eta \tag{76}$$

Note that the integral part of (76) needs to account for different in η and η_f. Therefore, when calculating the term $\{\Delta_{\sigma,i}(\eta)\}$, the relation between η and η_f within a original local element can be expressed as follow

$$\eta = \frac{1}{2}(\eta_f - 1) \text{ with } -1 \leq \eta_f \leq 1 \tag{77}$$

$$\eta = \frac{1}{2}(\eta_f + 1) \text{ with } -1 \leq \eta_f \leq 1 \tag{78}$$

With the expression for Δerr_{iel} and err_{iel}, the refinement criteria in (46) and (47) can now be implemented in the adaptive procedure (Demkowicz et al., 2002; Solin et al., 2003). The vector form of this (76) can find in the code.

3.6 The h-adaptive algorithm

Initialize condition $\bar{\eta}$, start with step $k = 1$, initialize coarse mesh $Mesh^k_{coarse}$
Start Find the coarse mesh solution σ^k_{coarse}
Uniformly split every element of $Mesh^k_{coarse}$ to obtain the fine mesh $Mesh^k_{fine}$.
Find the stresses σ^k_{fine}.
Project the fine mesh solution. Calculate $\sigma^k_{fine\_coarse}$
Compute the error estimator η^{*k} (69)
If $\eta^{*k} \leq \bar{\eta} \times 100\%$
then Stop the loop
else $(\eta^{*k} > \bar{\eta} \times 100\%)$
Calculate Δerr^k_{iel} for each element in the current coarse mesh.
Determine elements for refinement using (46) and (47).
Refine elements and store new mesh
Set $k = k + 1$
Back to Start
endif

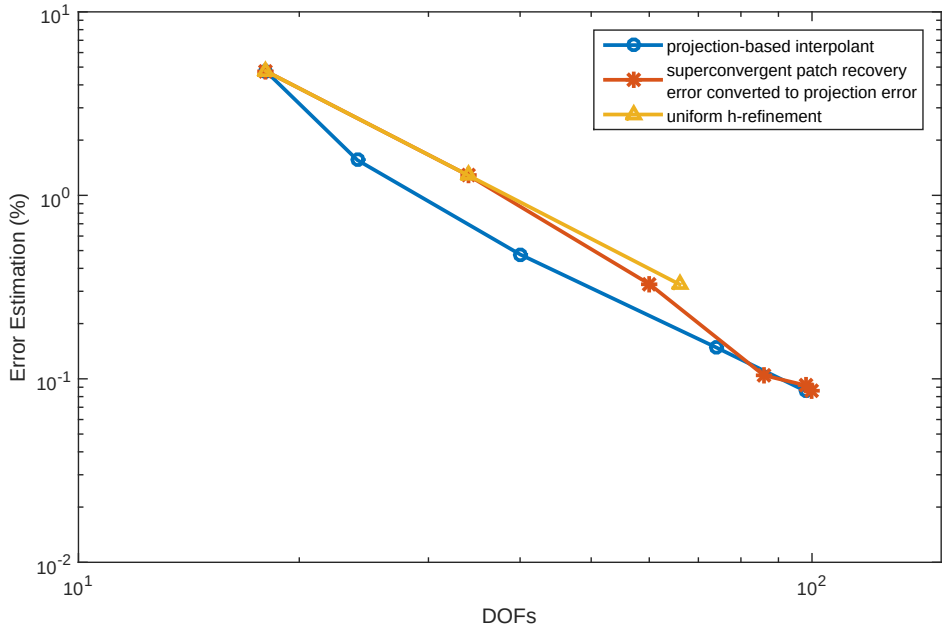

Figure 13: Square plate with a hole problem. Comparing projection-based method with superconvergent recovery and uniform refinement

4 Results and Discussion

This section will consider the plane square plate with a square hole problem and plane square plate under vertical load problem. Although, the reference method can benefit the process by starting with large size elements, the initialized square plate will be split into eight elements to be compatible with error estimators using superconvergence patch recovery.

In the conventional h-adaptive method, the process attemps to obtain an optimal mesh by ensuring that all the elements within share the same distribution to the error. Recall the equation (40), the criteria can be expressed the following form (Deeks & Wolf, 2002a)

$$\|\{e_{\sigma^*}(\xi, s)\}\|_e \leq \bar{\eta} \sqrt{\frac{\|\{\sigma^*(\xi, s)\}\|_V^2}{nel}}$$

Due to the difference in the error estimation between reference solution using projection-based interpolation and superconvergent patch recovery, it is necessary to compute the result in one scale for more accuracy comparing.

Figure 13 compares the results from h-adaptive method based on projection technique with the conventional h-adaptive method based on superconvergence patch recovery technique and the projection technique with equidistribution criteria. The error

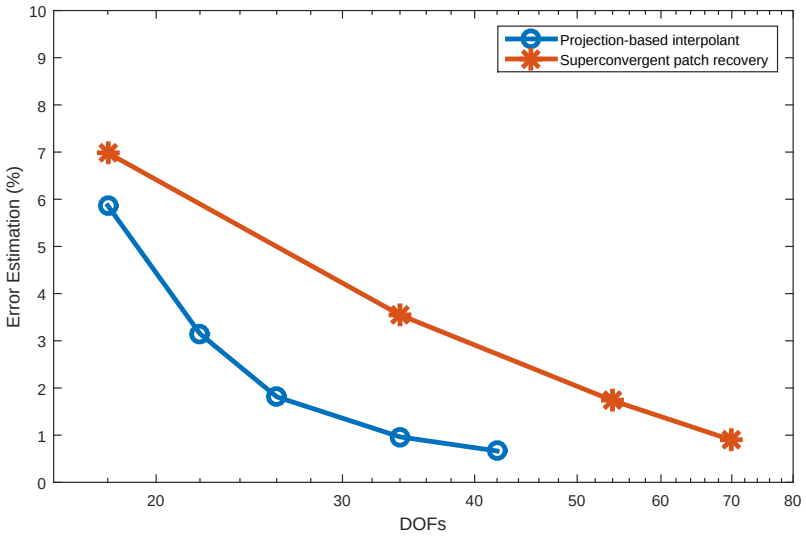

Figure 14: Strain square plate under vertical load. Comparing convergent rate between projection method and superconvergence method

estimators $\eta^*(\%)$ are plotted against the number degrees of freedom. Although, the error at the initilize mesh is different between two methods due to the method was used to model the exact solution, the error estimation in the resulted mesh of superconvergence method is re-evaluated using projection method, figure 13 indicates a significant convergence of the projection-based method comparing to the other techniques. It also noted that projection-based method is considered to be more accurate in estimating the error than superconvergence method (Demkowicz, 2006; O. Zienkiewicz & Taylor, 2000), as the superconvergence method overestimates the error in the same element comparing to projection method.

In figure 13, the blue line indicates the projection method, the red line indicates the patch recovery method, and the yellow line is the uniform refinement. As figure 13 shows the fast converged speed of projection method comparing to the uniform refinement and superconvergent refiment, it also show that after five iterations, the superconvergent refinement starts to catch up with the projection error estimation. This behavior can be explained due to the refinement criteria of superconvergent method which tries to refine every elements that has error larger than the threshold, while projection-based method focuses to element that causes the most error and refine it first. It is straightforward to predict that the superconvergence method will behave like a uniform refinement if the initial target is set significant low. Refer to figure 11 representing the stress on the plate, the square plate has a singular point near the square hold. This singular point has been captured by the defining the center of scaled-boundary at that point. Therefore, in this example, the error happens mostly around the corner.

Similarly, figure 14 provides the result from the square plate under vertial uniform

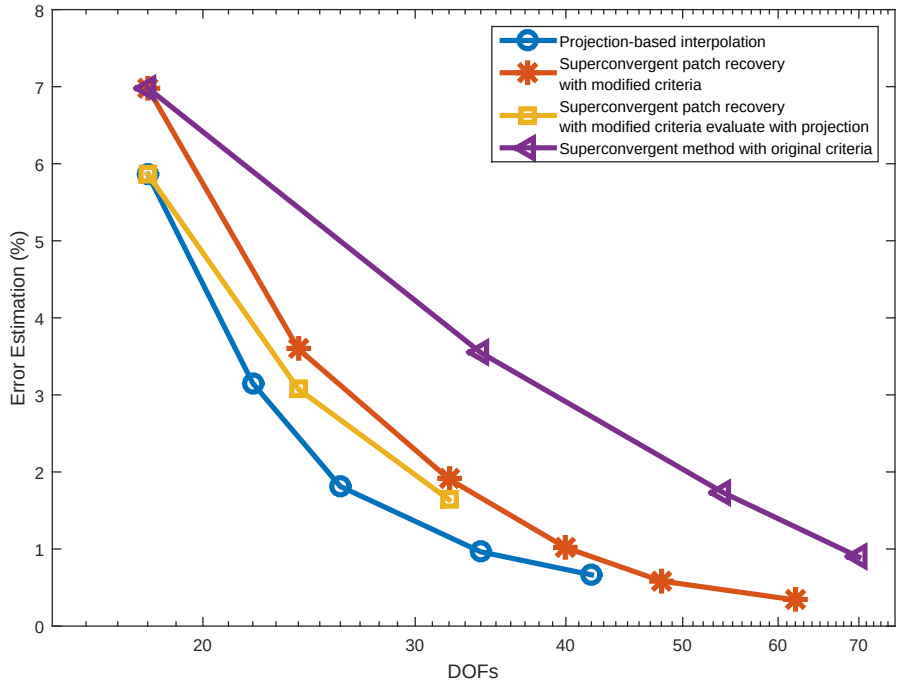

Figure 15: Strain square plate under vertical load. Comparing convergent rate between projection method and modified criteria superconvergence method

load. Due to numerial issue, in this example, it currently can not re-evaluate the result from superconvergent method in projection method. However, the figure still shows a high rate of convergence of projection technique comparing to superconvergence method. This figure also illustrates the discrepancy between error estimation based on patch recovery and projection recovery as the latter method provide a better estimation of error. Unlike figure 13, in this problem, patch recovery has a poor performance and has no chance to catch up with projection method. As the stress field shown in figure 10, the error is high around the load position. However, as the criteria of superconvergence recovery attemps to refine the error of all elements below the threshold, it refine a large number of elements before it starts to converge, which projection method have done a better job.

As the poor performance of the superconvergence method in the second example comes from the difference target in the error estimation. Figure 15 shows the result comparing the performance of the original superconvergence method (purple line) and a modified criteria superconvergence method (red colour). The new criteria adopts from equation (46, 47) and (42). This new criteria attempts to drive the adaptivity to the elements that contribute the largest error first. From figure 15, it is clear that the new criteria has improved the performance of the convergenent recovery method comparing

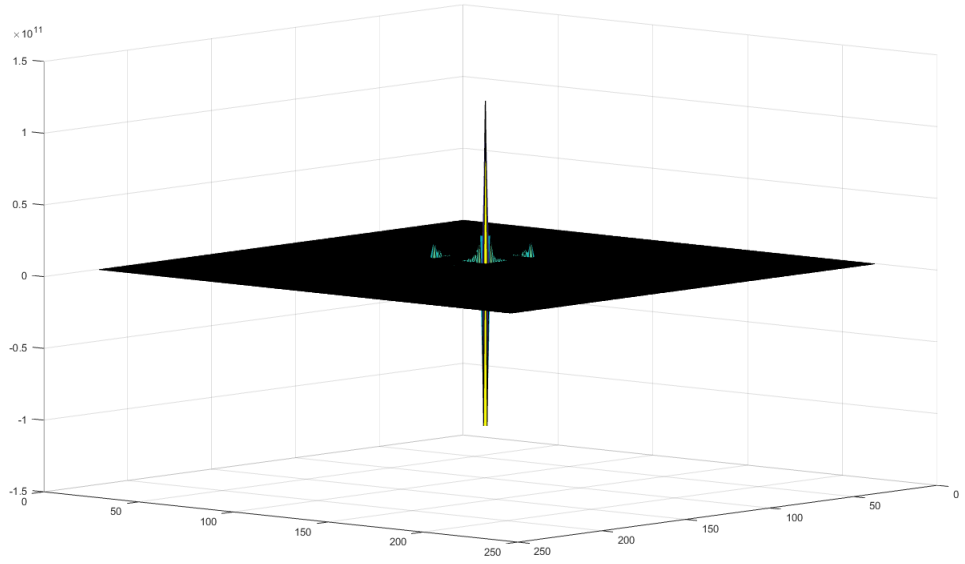

Figure 16: Numerical problem due to compute extremely large value

to the original version. To compare the new result with projection method, an effort has been made to re-evaluate the result mesh error in projection fashion. Although there are only three points for comparison (the yellow line), it is enough to expect that projection method still has a better rate than the patch method.

The next three figures illustrate step 1, 3 and 5 of the adaptivity process. Step 1 (figure 17) is the initial mesh with 8 elements and 9 nodes. In the next 2 steps, the h-refinement process two elements around the load position while leaves the other elements relative large (figure 18). After 5 iterations, in figure (19) the stress field becomes smooth and indicates the concentrated load position. Also, in the last step, when the overall error reduces significantly, the adaptivity process starts to refine the element at the corner. The process ends at target error $e \approx 0.7$.

Regarding to the numerical instability mentioned above, figure 16 demonstrates the numerical values required to evaluate the Δerr_{iel} in formula (76) for an element. The total number of values are $n_{fine} \times n_{fine}$. The figure is a 3D picture with x-axis and y-axis are the index of the value, while the z-axis is the value of that variable. It is clear that, this matrix contains some extremely large/small values. Those values will affect the accumulation process when Matlab tries to balance the memory to store the big and small number. It leads to error in computation and reduce the liability of the result.

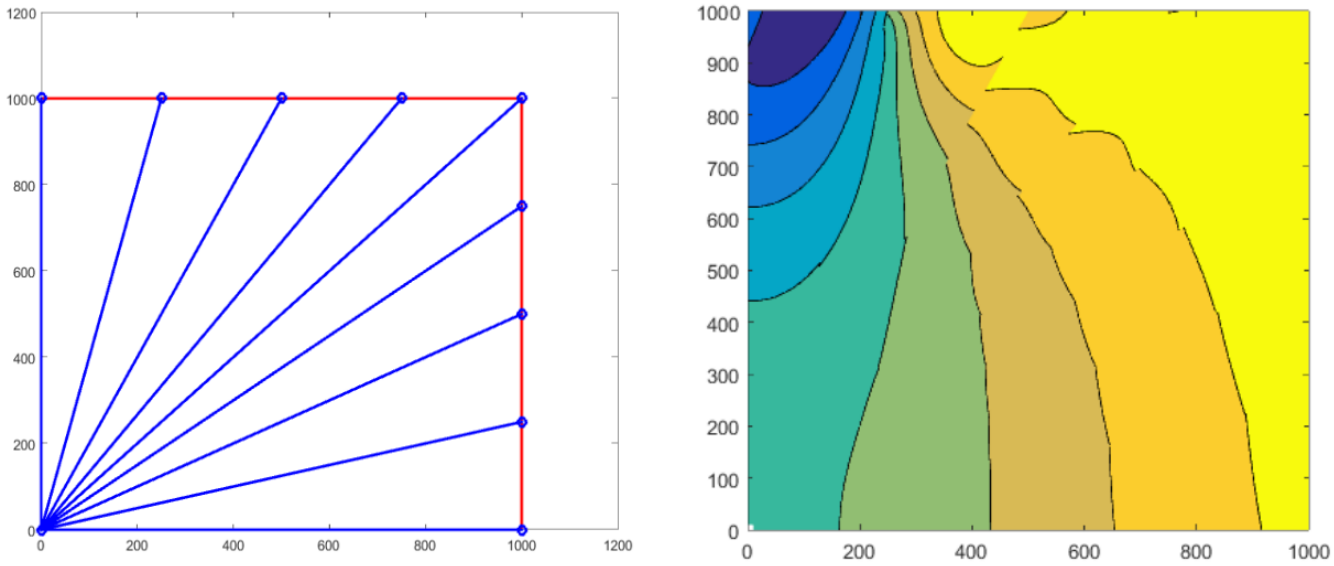

Figure 17: Strain square plate problem. Iteration 1

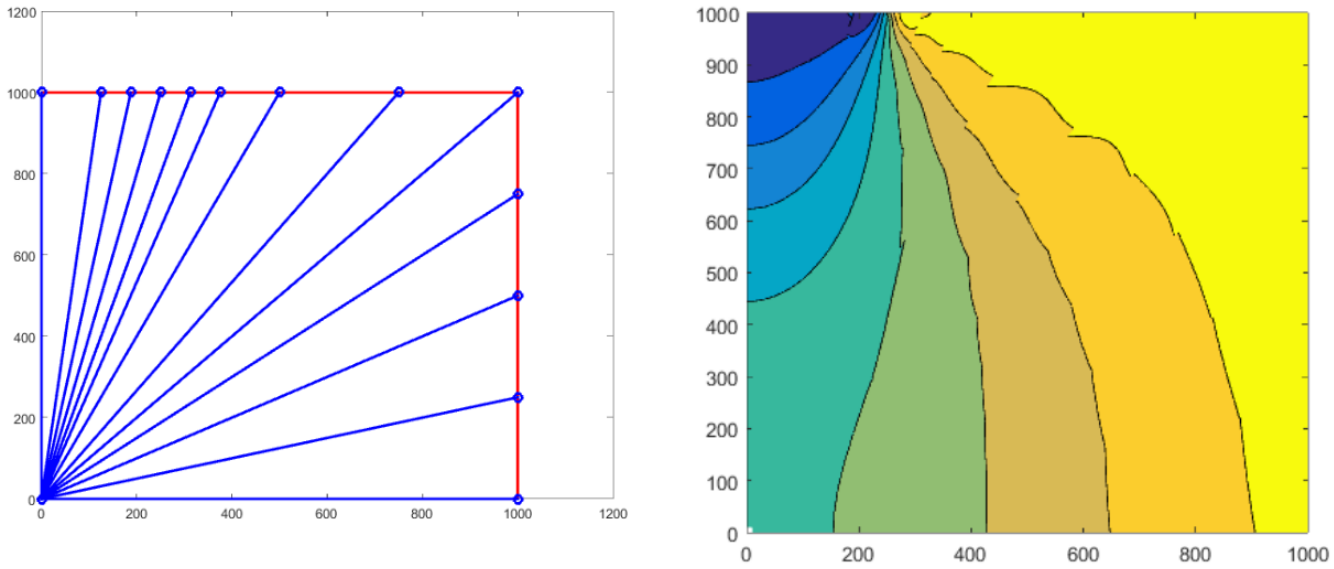

Figure 18: Strain square plate problem. Iteration 3

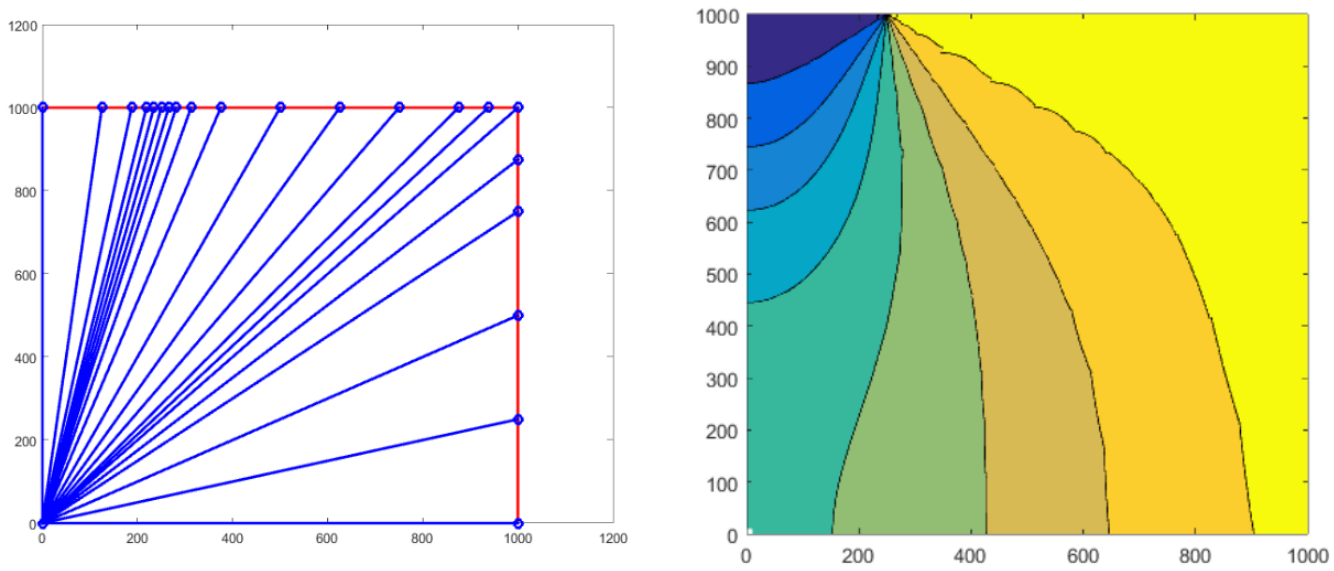

Figure 19: Strain square plate problem. Iteration 5

5 Concluding Remarks

5.1 Conclusions

This report has provided a brief background about scaled-boundary finite element method. It includes the definition of the scaled boundary problem, and equation to calculate displacement and stress field which is also the main difference between traditional finite element method and scaled boundary method. Next, the report introduced about various type of shape function and placed an emphasis on Lobatto shape function due to its properties to save memory and calculation when dealing with high order polynomials (Vu & Deeks, 2008). Another useful feature of Lobatto shape function is its orthogonal derivatives derived from Legendre polynomials which proved to ease the calculation in both p-adaptive and h-adaptive method. The report also introduced various popular refinement methods, the concept of the reference solution and the role of error estimation in adaptivity process.

In order to evaluate the performance of projection-based interpolation method in h-adaptivity, the report first reimplemented the superconvergent patch recovery method. This method uses the Gauss points within an element to recover the nodal nodes. Due to a distinct property of scaled boundary finite element method, the recovery process needs to apply in separated displacement modes to smooth the final stress field. The report also presented the error estimation based on Deeks and Wolf's work to drive the h-adaptivity method (Deeks & Wolf, 2002a). It is worth to note that this report tries to convert all the formula to vector form. As in programming environments, a vector form is easier to read and also outperforms the traditional loop control.

Next, the report expands the theory in projection-based interpolation to make it

capable of implementing in h-adaptivity. First, the report forms the equation to project the fine mesh to the coarse mesh. Although this h-projecting was mentioned in another research (Demkowicz, 2006), it has not been derived explicitly. Then, the report reviews the criteria and error estimation based on energy norm. It suggests that the formula needs to modify the η parameters of the fine mesh to account for the difference in configuration between the coarse mesh and fine mesh. In this case, the difference is for every element in the coarse mesh; there are two corresponding element in the fine mesh which was the same coarse element split into two.

The result was taken from two examples which are a plate with a square hole and a square plate under uniform patch load. Each example is applied h-adaptivity but with different methods modelling the error estimation. The result shows that the projection-based method produced the faster convergent speed than the superconvergent patch recovery method. Its effect in refining the high error element is well indicated in the second example which is the square plate under uniform patch load. In this example, the h-adaptivity attempts to refine elements under the load which are introducing the largest error. Finally, for the second example, the report experimented that by modifying the criteria of the original superconvergent patch recovery following the error criteria based on distribution style of projecting method, the modified version shows a better result than the original one. However, the modified version still has a lower converged rate comparing to projection method. Therefore, this report has confirmed that implementation projection-based method into h-adaptivity is possible and has a better performance than the traditional superconvergent patch recovery.

5.2 Future Work

As discussed in Result section, numerical stability is the remained problem as the calculation includes extreme large/small numbers. Beside the problem, hp-adaptivity and artificial intelligence enable are two suggested future work after this papers.

Numerical stability

Numerical stability is the problem happened when the mesh is finer, and the error estimation becomes sensitive with computation accuracy. As it is discussed in the previous section, the accuracy is exposed to the high error in calculation which affected the final result. This error requires more study in numerical method to stabilise the calculation and reduce the error.

hp-adaptivity

This work has confirmed that it is possible to implement a projection-based method to h-adaptivity procedure. It enables to develop the hp-adaptivity in scaled boundary method which incorporates complex mesh's configurations. Especially, this work utilised Lobatto shape functions in forming error estimator and recover projecting stress which is highly recommended when dealing with p-adaptivity.

Artificial intelligent enable

It is clear that the adaptivity process focuses on elements at the sharp conner or under load. A senior engineer with his/her experiments can determine which areas/elements should be refined to achieve, might not numerically, but an adequate accuracy. Therefore, it is promising that a reinforcement learning system can learn to map the physical information of the mesh to a refining action. Especially, a raw stress field from a coarse mesh is an image which AI has recently proved how efficient it is to learn.

References

Charafi, A., Neves, A., & Wrobel, L. (1995). h-hierarchical adaptive boundary element method using local reanalysis. *International journal for numerical methods in engineering*, *38*(13), 2185–2207.

Cramer, H., Rudolph, M., Steinl, G., & Wunderlich, W. (1999). A hierarchical adaptive finite element strategy for elastic–plastic problems. *Computers & structures*, *73*(1), 61–72.

Deeks, A. J., & Cheng, L. (2003). Potential flow around obstacles using the scaled boundary finite-element method. *International journal for numerical methods in fluids*, *41*(7), 721–741.

Deeks, A. J., & Wolf, J. P. (2002a). An h-hierarchical adaptive procedure for the scaled boundary finite-element method. *International Journal for Numerical Methods in Engineering*, *54*(4), 585–605.

Deeks, A. J., & Wolf, J. P. (2002b). Stress recovery and error estimation for the scaled boundary finite-element method. *International Journal for Numerical Methods in Engineering*, *54*(4), 557–583.

Deeks, A. J., & Wolf, J. P. (2002c, June). A virtual work derivation of the scaled boundary finite-element method for elastostatics. *Computational Mechanics*, *28*(6), 489–504. doi: 10.1007/s00466-002-0314-2

Demkowicz, L. (2006). *Computing with hp-adaptive finite elements: Volume 1 one and two dimensional elliptic and maxwell problems.* CRC Press.

Demkowicz, L., Rachowicz, W., & Devloo, P. (2002). A fully automatic hp-adaptivity. *Journal of Scientific Computing*, *17*(1-4), 117–142.

Gui, W.-z., & Babuska, I. (1985). *The h, p and hp versions of the finite element method in 1 dimension. part 1. the error analysis of the p-version.* (Tech. Rep.). DTIC Document.

Guiggiani, M., & Lombardi, F. (1992). Self-adaptive boundary elements with h-hierarchical shape functions. *Advances in Engineering Software*, *15*(3-4), 269–277.

Kita, E., Higuchi, K., & Kamiya, N. (2000). Application of r-and hr-adaptive bem to two-dimensional elastic problem. *Engineering analysis with boundary elements*, *24*(4), 317–324.

Krishnamoorthy, C., & Umesh, K. R. (1993). Adaptive mesh refinement for two-dimensional finite element stress analysis. *Computers & structures*, *48*(1), 121–133.

Oh, H.-S., & Batra, R. (1999). Application of zienkiewicz–zhus error estimate with superconvergent patch recovery to hierarchical p-refinement. *Finite Elements in Analysis and Design*, *31*(4), 273–280. doi: 10.1016/s0168-874x(98)00063-8

Solin, P., Segeth, K., & Dolezel, I. (2003). *Higher-order finite element methods.* CRC Press.

Song, C. (2004). A matrix function solution for the scaled boundary finite-element equation in statics. *Computer Methods in Applied Mechanics and Engineering*, *193*(23), 2325–2356.

Song, C., & Wolf, J. P. (1996). Consistent infinitesimal finite-element cell method: Three-dimensional vector wave equation. *International Journal for Numerical Methods in Engineering*, *39*(13), 2189–2208.

Song, C., & Wolf, J. P. (1999). The scaled boundary finite element method—alias consistent infinitesimal finite element cell method—for diffusion. *International Journal for Numerical Methods in Engineering*, *45*(10), 1403–1431.

St Doltsinis, J. (1987). Accuracy estimates and adaptive refinements in finite element computations i. babuska, oc zienkiewicz, j. gago, er de arantes oliveira, eds.(wiley, new york, 1987) isbn 0 471 90862 2. *Computer Methods in Applied Mechanics and Engineering*, *65*, 194.

Tveito, A., Langtangen, H. P., Nielsen, B. F., & Cai, X. (2010). *Elements of scientific computing* (Vol. 7). Springer Science & Business Media.

Vu, T. H., & Deeks, A. J. (2006). Use of higher-order shape functions in the scaled boundary finite element method. *International Journal for Numerical Methods in Engineering*, *65*(10), 1714.

Vu, T. H., & Deeks, A. J. (2007, December). A p-adaptive scaled boundary finite element method based on maximization of the error decrease rate. *Computational Mechanics*, *41*(3), 441–455. doi: 10.1007/s00466-007-0203-9

Vu, T. H., & Deeks, A. J. (2008, January). Ap-hierarchical adaptive procedure for the scaled boundary finite element method. *International Journal for Numerical Methods in Engineering*, *73*(1), 47–70. doi: 10.1002/nme.2055

Zhu, J., & Zienkiewicz, O. (1990). Superconvergence recovery technique and a posteriori error estimators. *International Journal for Numerical Methods in Engineering*, *30*(7), 1321–1339.

Zienkiewicz, O., & Taylor, R. (2000). *The finite element method: The basis*. Butterworth-Heinemann. Retrieved from `https://books.google.com.au/books?id=x4BGAAAAYAAJ`

Zienkiewicz, O. C., & Zhu, J. Z. (1992). The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, *33*(7), 1331–1364.