An Efficient Error Indicator with Mesh Smoothing for Mesh Refinement: Application to Poisson and Laplace Problems

ZHAOCHENG XUAN *, YAOHUI LI, HONGJIE WANG

Tianjin University of Technology and Education
Department of Computational Science
Tianjin 300222
China
* Corresponding email: xuanzc@tute.edu.cn

Abstract: An efficient mesh refinement method for *h*-version finite element analysis is presented based on both an a-posteriori error indicator and the geometrical quality of mesh. The first step is to refine the meshes on which the a-posteriori error indicators are relatively higher than the others. The error indicators are obtained by simplifying the computation of error bounds which are obtained by solving elemental Neumann type subproblems with the averaged flux for the consistency of the Neumann problems. The simplification of computation means that the functional space on the mesh uniformly refined with only half size of the coarse mesh is chosen as the test functional space in the elemental residual form of error equations, thus the cost for computing the error indicators is quite low. After refinement, some refined triangles will become poorly shaped or distorted, then the second step is to move the meshes to improve their geometrical quality with Laplacian smoothing algorithm. Two examples are computed to verify this method and the results show that the refined mesh obtained by both the a-posteriori error indicator and mesh smoothing gives the optimal convergence and higher accuracy for the results.

Key–Words: Finite elements; Mesh refinement; Error indicator; Mesh smoothing; Implementation

1 Introduction

The subject of adaptive mesh refinement is as old as finite element method, but now it has the potential to improve the quality of the bounding on the quantity of interest in engineering [1–3]. The main objective of the adaptive mesh refinement is to achieve the optimal rate of convergence and reach the desired accuracy with the minimum degrees of freedom. In *h*-version finite element analysis, the number of degrees of freedom or elements must reach a larger quantity in order to obtain more accurate results. When the accuracy of analysis results obtained from a coarser mesh dose not meet the demand, the mesh has to be refined for obtaining an improved solution with better accuracy.

There are two ways to refine the element mesh, one is uniform, which divides *all* elements to generate new elements to form a new mesh, another is adaptive, which generates new elements only at the place where *error* is relatively higher. The computing on uniform mesh will lead to not only huge cost of computing, but also less accurate results, as compared with the adaptive mesh with equivalent number of elements, at the place where intensity may occur. Adaptive mesh refinement strategies provide a good basis to improve the accuracy of solutions. The essential part of adaptive finite element calculations is the estimation of discretization error and the design of refined meshes. The energy norm of error, $a(e,e)$, which concerns to the derivative of solution, is often chosen to measure the error in usual error estimators. The asymptotically optimal mesh is defined by [4–6] as one in which all error indicators are equal, and the value of the optimal error is stable under perturbations of the optimal mesh. Adaptive refinement involves not only the introduction of new elements where needed, but also be combined with mesh regularization (i.e., the movement of elements) to further improve the equality of the mesh [7–9]. When an error estimator of an element is relatively larger, the element will be divided. There are two basic methods for dividing triangles used in practice: regular division and bisection. To divide a triangle by regular division, one connects the midpoints of the sides of the triangle to obtain four triangles similar to the original. The incompatibilities will arise when only some of the triangles are divided by regular division, the hanging nodes will appear. The method often used for solving the problem of incompatibilities is to connect the hanging node and the opposite vertex to form two elements, like the bisection method that is used to divide triangles by connecting a vertex to the midpoint of the opposite side. The bisection method is usually performed to divide the longest edge, this choice is intuitively reasonable since it divides the largest angle. No matter what method, the regular method or the bisection method, is used to refine the mesh, some refined triangles will become poorly shaped or distorted, therefore, some node point adjustment methods such as mesh smoothing are used to improve the geometrical quality of the mesh [10–14].

The commonly used mesh smoothing technique comprises local methods that operate on vertex at a time to improve mesh quality in a neighborhood of that vertex. Some number of sweeps over the adjustable vertices are performed to achieve an overall improvement for the mesh. Local mesh smoothing technique operates using data from the neighborhood of a grid point that is being adjusted. Laplacian smoothing technique is such a method, which moves the free vertex to the geometric center of its incident vertices with an inexpensive cost of computation.

In this paper, we develop an adaptive algorithm based on a simplified a-posteriori error estimation and mesh smoothing. The adaptive refinement procedure includes two steps, one is to refine the mesh by a-posteriori error estimator, another is to move the mesh to improve its geometrical quality. The paper is organized as follows, in Section 2, the basic concepts and the error estimation are introduced; in Section 3, the algebra form of the error estimation in an element is presented; in Section 4, examples are given to illustrate the algorithm; in the last section, the conclusions are drew.

2 A posteriori error estimator

We consider the standard self-adjoint, positive definite, elliptic equation

$$\begin{aligned} -\nabla(p\nabla u)+qu &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma_D, \\ p\frac{\partial u}{\partial n} &= g \quad \text{on } \Gamma_N, \end{aligned}$$

with $p > 0$, $q \geq 0$, and f and g the given smooth functions in the space $L^2(\Omega)$. Domain Ω is a bounded region in $\Re^2$. The boundary of the region, $\partial\Omega$, is assumed piecewise smooth and composed of Dirichlet portion Γ_D and Neumann portion Γ_N, i.e. $\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}$. The weak form of the above equation is: find u in $H^1(\Omega)$ such that

$$a(u,v) = (f,v)+\langle g,v\rangle, \quad \forall v \in H^1(\Omega) \tag{1}$$

in which

$$\begin{aligned} a(u,v) &= \int_\Omega p\nabla u\cdot\nabla v + quv\,\mathrm{d}\Omega, \\ (f,v) &= \int_\Omega fv\,\mathrm{d}\Omega, \\ \langle g,v\rangle &= \int_{\Gamma_N} gv\,\mathrm{d}\Gamma. \end{aligned}$$

In order to obtain the approximate solutions of the weak problem, a finite dimensional counterpart of all these variational forms above must be built using the finite element method. Two triangulations of the computational domain Ω are considered: the working or coarse H-mesh, $\mathscr{T}_H$, consisting of K_H elements T_H; the “truth” or fine h-mesh, $\mathscr{T}_h$, consisting of K_h elements T_h, and one fine mesh is only in one coarse mesh. To each of these meshes we associate regular piecewise linear continuous finite element subspaces,

$$\begin{aligned} X_H &= \{v \in X \mid v|_{T_H} \in P_1(T_H), \quad \forall T_H \in \mathscr{T}_H\}; \\ X_h &= \{v \in X \mid v|_{T_h} \in P_1(T_h), \quad \forall T_h \in \mathscr{T}_h\}, \end{aligned}$$

where $P_1(T)$ denotes the space of linear polynomials over T. As we require that a fine mesh T_h be only within one coarse element T_H, it is obvious that $X_H \subset X_h \subset H^1(\Omega)$.

Let $\mathscr{E}_I$ represent the coarse element edges in the interior of Ω, here we assume that boundary $\partial\Omega$ consist of the edges of $T_H \in \mathscr{T}_H$. Let $\mathscr{N}_{T_H}$ denote all neighbor elements sharing common edges ∂T_H with the coarse element T_H, i.e.

$$\mathscr{N}_{T_H} = \left\{T'_H \in \mathscr{T}_H \mid T'_H \cap T_H = \mathscr{E}_I\right\}.$$

Let $\mathscr{E}(T_H)$ and $\mathscr{E}(T_h)$ denote the set of open edges in the triangulation $\mathscr{T}_H$ and $\mathscr{T}_h$, respectively, then according to the above assumption, for coarse mesh we have $\mathscr{E}(T_H) = \mathscr{E}_I \cup \Gamma_D \cup \Gamma_N$.

Let u_H be the solution for the coarse mesh

$$a(u_H,v) = (f,v)+\langle g,v\rangle, \quad \forall v \in X_H \tag{2}$$

and u_h be the solution for the refined mesh

$$a(u_h,v) = (f,v)+\langle g,v\rangle, \quad \forall v \in X_h \tag{3}$$

then we have the residual

$$R(v) = (f,v)+\langle g,v\rangle - a(u_H,v), \quad \forall v \in X_h, \tag{4}$$

and the error

$$e = u_h - u_H,$$

then according to (2) and (3), we have

$$R(v) = a(e,v).$$

There will be *jump* between elements, the bilinear form of *jump* is $j : \hat{X}_h \times Q_h \mapsto \Re$,

$$j(v,v) = \sum_{\partial T_H \in \mathscr{E}_I} \int_{\partial T_H} \Upsilon_{\partial T_H}(v)\, v|_{\partial T_H}\, d\Gamma, \tag{5}$$

where $\Upsilon_{\partial T_H}(v)$ is the jump in v across ∂T_H when $\partial T_H \in \mathscr{E}_I$. We will use an approximate to the *jump* of the fluxes on the element edges for constructing the error indictors in this study. There are *fluxes* on the edges of elements, and the *jump* of *fluxes* is

$$j(v,v)|_{\partial T_H} = \int_{\partial T_H} \frac{1}{2}\left(\left.\frac{\partial u_H}{\partial n}\right|^{+}_{\partial T_H} - \left.\frac{\partial u_H}{\partial n}\right|^{-}_{\partial T_H} \right) v|_{\partial T_H} d\Gamma.$$

Exactly solving the error that concerns the *jump* is difficult or impossible, an alternative is to smooth the *jump* and form an approximate equation for the error on element T_H, and to make overall bound on the error of whole finite element model.

Let us define the approximate flux by smoothing the fluxes of two neighbor elements on a common boundary with averaging of the two fluxes, i.e.

$$\begin{aligned} \overline{\frac{\partial u_H}{\partial n_T}} &= \frac{1}{2}\left(\left.\frac{\partial u_H}{\partial n}\right|^{+}_{\partial T_H} + \left.\frac{\partial u_H}{\partial n}\right|^{-}_{\partial T_H} \right) \\ &= \frac{1}{2}\left(\left.\frac{\partial u_H}{\partial n}\right|_{\partial T_H} + \left.\frac{\partial u_H}{\partial n}\right|_{\partial T'_H \in \mathscr{N}_{T_H}} \right), \end{aligned}$$

where n_{T_H} is the outward normal of the boundary of element T_H.

Let $\hat{e} \in X_h$ be an error functional, which satisfies the following equation define on element $T_H \in \mathscr{T}_H$

$$a_{T_H}(\hat{e}, v) = \bar{R}_{T_H}(v - \mathscr{I}_H v), \quad \forall v \in X_h \tag{6}$$

where $\mathscr{I}_H : X_h \mapsto X_H$ is a polynomial interpolation operator, and

$$\begin{aligned} &\bar{R}_{T_H}(v - \mathscr{I}_H v) = R_{T_H}(v - \mathscr{I}_H v) \\ &\quad + \left\langle p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, v - \mathscr{I}_H v \right\rangle_{\partial T_H \cap \varepsilon_I} \\ &\quad = (f, v - \mathscr{I}_H v)_{T_H} + \langle g, v - \mathscr{I}_H v \rangle_{\partial T_H \cap \varepsilon_B} \\ &\quad - a_{T_H}(u_H, v - \mathscr{I}_H v) + \left\langle p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, v - \mathscr{I}_H v \right\rangle_{\partial T_H \cap \varepsilon_I}. \end{aligned} \tag{7}$$

Let us examine the consistency of equation (6). If $v = C$ is a constant, then we have $C = \mathscr{I}_H C$, this is because C belongs to both X_H and X_h, therefore from equation (7), its right hand side is 0, and then $\bar{R}_{T_H}(v - \mathscr{I}_H v) = 0$ for v being a constant.

Another way to examine the consistent is to use equation (4), because C satisfies $a(u_h, v) = (f, v) + \langle g, v \rangle$, for all $v \in X_h$, and $a(u_H, v) = (f, v) + \langle g, v \rangle$, for all $v \in X_H$, so that $a(u_h, C) = (f, C) + \langle g, C \rangle$, and $a(u_H, C) = (f, C) + \langle g, C \rangle$. Therefore $a(u_h, C) = a(u_H, C)$, $a(e, C) = a(u_h - u_H, C) = 0$. We can solve e that is in the infinite functional space with an approximate solution $\hat{e}$ in finite functional space, that is $a_{T_H}(\hat{e}, v) = 0$.

Let $e = v$, and sum all elements in $\mathscr{T}_H$, the following equations are obtained

$$\begin{aligned} a(\hat{e}, e) = R(e) &+ \sum_{T_H \in \mathscr{T}_H} \left\langle p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, e \right\rangle_{\partial T_H \cap \varepsilon_I} - R(\mathscr{I}_H e) \\ &- \sum_{T_H \in \mathscr{T}_H} \left\langle (\mathscr{I}_H e)\, p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, e \right\rangle_{\partial T_H \cap \varepsilon_I}, \end{aligned}$$

while

$$\sum_{T_H \in \mathscr{T}_H} \left\langle p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, e \right\rangle_{\partial T_H \cap \varepsilon_I} = 0; \tag{8}$$

$$R(\mathscr{I}_H e) = \sum_{T_H \in \mathscr{T}_H} R_{T_H}(\mathscr{I}_H e) = 0; \tag{9}$$

$$\sum_{T_H \in \mathscr{T}_H} \left\langle (\mathscr{I}_H e)\, p\overline{\frac{\partial u_H}{\partial n_{T_H}}}, e \right\rangle_{\partial T_H \cap \varepsilon_I} = 0; \tag{10}$$

(8) is due to $\sum_{T_H \in \mathscr{T}_H} \left.\overline{\frac{\partial u_H}{\partial n_{T_H}}}\right|_{\varepsilon_I} = 0$, this is because on an inner edge of elements T_H and T'_H,

$$\overline{\frac{\partial u_H}{\partial n_T}} = -\overline{\frac{\partial u_H}{\partial n_{T'}}}.$$

(9) is due to $\mathscr{I}_H e \in X_H$, and (10) is for the same reason as (8). Then we have

$$a(\hat{e}, e) = R(e) = a(e, e).$$

Because

$$\begin{aligned} 0 &\leq (\hat{e} - e, \hat{e} - e) \\ &= a(\hat{e}, \hat{e}) + a(e, e) - 2a(e, \hat{e}) \\ &= a(\hat{e}, \hat{e}) - a(e, e), \end{aligned}$$

then

$$a(e, e) \leq a(\hat{e}, \hat{e}).$$

If we use $||u||^2 = a(u, u)$ to express the energy norm of solution, then the above inequality means $||\hat{e}||$ is an upper bound of $||e||$.

3 The Calculation of local problems

Let us assume there will be n finite element nodes on the mesh, and $\phi_H^i(x,y)$ (for $i=1,2,\cdots,n$) is the linear basis function corresponding to node i, that satisfies $\phi_H^i(x,y)=1$ at node i, and $\phi_H^i(x,y)=0$ at the others nodes. That means it is one at node i, and on the boundary of the patch formed by the elements that contain node i, it is zero. The basis functions satisfy

$$\sum_{i=1}^{n}\phi_H^i(x,y)=1.$$

The error estimator on each element is used as a criterion for adaptive refining the mesh. We consider each element T_H as a subproblem to implement the algorithm for the error estimator. Now let the finite element solution of element T_H be $\{u_H^1,u_H^2,u_H^3\}^T$, and $\phi_H^i(x,y)$ be the basis functions corresponding to the nodes $i=1,\cdots,3$ of element T_H, respectively, that is

$$\begin{aligned}
\phi_H^1(x,y)&=\frac{1}{2S_{T_H}}\left[(y_2-y_3)x+(x_3-x_2)y+\right.\\
&\quad\left.(x_2y_3-x_3y_2)\right];\\
\phi_H^2(x,y)&=\frac{1}{2S_{T_H}}\left[(y_3-y_1)x+(x_1-x_3)y+\right.\\
&\quad\left.(x_3y_1-x_1y_3)\right];\\
\phi_H^3(x,y)&=\frac{1}{2S_{T_H}}\left[(y_1-y_2)x+(x_2-x_1)y+\right.\\
&\quad\left.(x_1y_2-x_2y_1)\right],
\end{aligned}$$

in which S_{T_H} is the area of element T_H

$$S_{T_H}=\frac{1}{2}\begin{vmatrix}1 & x_1 & y_1\\ 1 & x_2 & y_2\\ 1 & x_3 & y_3\end{vmatrix}.$$

The interpolation function of element T_H based on the basis functions is expressed as

$$u_H|_{T_H}=\sum_{i=1}^{3}u_H^i\phi_H^i(x,y). \tag{11}$$

In order to calculate the error estimator which is one higher order than the finite element solution, we will select the middle points on each element edge as the new nodes to generate 4 new fine elements on T_H. Let the basis functions for the new mesh be $\phi_h^1(x,y),\cdots,\phi_h^6(x,y)$,

$$\begin{aligned}
\phi_h^1=&\frac{1}{S_{T_H}}\left[(y_2-y_3)x+(x_3-x_2)y+\frac{1}{2}(x_1+x_2)(y_1+y_3)\right.\\
&\left.-\tfrac{1}{2}(x_1+x_3)(y_1+y_2)\right] \text{ on } T_h^1;\\
\phi_h^2=&\frac{1}{S_{T_H}}\left[(y_3-y_1)x+(x_1-x_3)y+\frac{1}{2}(x_2+x_3)(y_1+y_2)\right.\\
&\left.-\tfrac{1}{2}(x_1+x_2)(y_2+y_3)\right] \text{ on } T_h^2;\\
\phi_h^3=&\frac{1}{S_{T_H}}\left[(y_1-y_2)x+(x_2-x_1)y+\frac{1}{2}(x_1+x_3)(y_2+y_3)\right.\\
&\left.-\frac{1}{2}(x_2+x_3)(y_1+y_3)\right] \text{ on } T_h^3;\\
\phi_h^4=&\frac{1}{S_{T_H}}\left[(y_1-y_2)x+(x_2-x_1)y+(x_1+x_2)y_2\right.\\
&\left.-x_2(y_1+y_2)\right] \text{ on } T_h^2;\\
\phi_h^4=&\frac{1}{S_{T_H}}\left[(y_3-y_1)x+(x_1-x_3)y+x_3(y_1+y_3)\right.\\
&\left.-(x_1+x_3)y_3\right] \text{ on } T_h^3;\\
\phi_h^4=&\frac{1}{S_{T_H}}\left[(y_3-y_2)x+(x_2-x_3)y+\frac{1}{2}(x_1+x_3)(y_1+y_2)\right.\\
&\left.-\frac{1}{2}(x_1+x_2)(y_1+y_3)\right] \text{ on } T_h^4;\\
\phi_h^5=&\frac{1}{S_{T_H}}\left[(y_1-y_2)x+(x_2-x_1)y+x_1(y_1+y_2)\right.\\
&\left.-(x_1+x_2)y_1\right] \text{ on } T_h^1;\\
\phi_h^5=&\frac{1}{S_{T_H}}\left[(y_2-y_3)x+(x_3-x_2)y+(x_2+x_3)y_3\right.\\
&\left.-x_3(y_2+y_3)\right] \text{ on } T_h^3;\\
\phi_h^5=&\frac{1}{S_{T_H}}\left[(y_1-y_3)x+(x_3-x_1)y+\frac{1}{2}(x_1+x_2)(y_2+y_3)\right.\\
&\left.-\frac{1}{2}(x_2+x_3)(y_1+y_2)\right] \text{ on } T_h^4;\\
\phi_h^6=&\frac{1}{S_{T_H}}\left[(y_3-y_1)x+(x_1-x_3)y+(x_1+x_3)y_1\right.\\
&\left.-x_1(y_1+y_3)\right] \text{ on } T_h^1;\\
\phi_h^6=&\frac{1}{S_{T_H}}\left[(y_2-y_3)x+(x_3-x_2)y+x_2(y_2+y_3)\right.\\
&\left.-(x_2+x_3)y_2\right] \text{ on } T_h^2;\\
\phi_h^6=&\frac{1}{S_{T_H}}\left[(y_2-y_1)x+(x_1-x_2)y+\frac{1}{2}(x_2+x_3)(y_1+y_3)\right.\\
&\left.-\frac{1}{2}(x_1+x_3)(y_2+y_3)\right] \text{ on } T_h^4.
\end{aligned}$$

They form a basis of the space X_h. In fact we will not really divide the mesh T_H into four elements in this step. Based on the four *virtual* elements

$T_h^1, T_h^2, T_h^3, T_h^4$, the error estimator can be spanned by the basis functions $\phi_h^1(x,y)$, $\cdots$, $\phi_h^6(x,y)$

$$e_{T_H} = \sum_{i=1}^{6} e_{T_H}^i \phi_h^i(x,y). \qquad (12)$$

There will be m refined elements and n nodes in the working mesh. Let the basis functions for the refined element space in a working mesh be $\phi_h^1(x,y)$, $\phi_h^2(x,y)$, $\cdots$, $\phi_h^n(x,y)$, thus the solution function of element T_H can also be spanned by the basis functions of the refined element space, i.e.

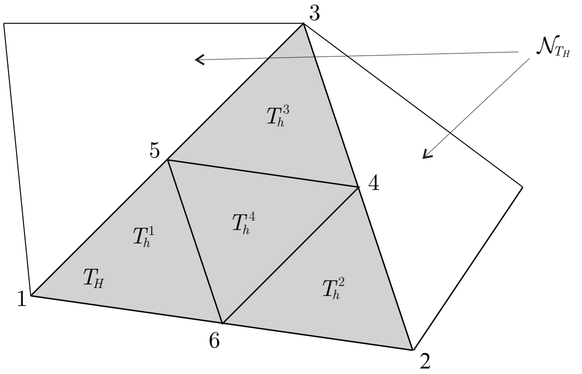


Figure 1: A piecewise finite element space is constructed with the uniform mesh refinement on element T.

$$u_H = \sum_{i=1}^{n} u_H|_{T_H}(P_h^i)\phi_h^i(x,y)$$

in which P_h^i is the coordinate of the ith node of refined mesh. the function v on the working element T_H is

$$v = \sum_{j=1}^{6} v_j \phi_h^j(x,y), \qquad (13)$$

then on element T_H

$$\begin{aligned} v - \mathscr{I}_H v &= \sum_{j=1}^{n} \phi_h^j(x,y) - \mathscr{I}_H \phi_h^j(x,y)) \\ &= \sum_{j=1}^{n} v_j \bar{\phi}_h^j(x,y). \quad \forall v \in X_h \qquad (14) \end{aligned}$$

Substitute (13) and (14) to (7), we have

$$\mathbf{A}\mathbf{e}_{T_H} = \mathbf{c} \qquad (15)$$

in which

$$\mathbf{c} = \mathbf{f} + \mathbf{g} - \mathbf{B}^T \mathbf{u}_H|_{T_H} + \mathbf{d}$$

and

$$\begin{aligned} &\mathbf{u}_H|_{T_H} = \left\{u_H^1, \cdots, u_H^6\right\}^T \\ &= \left\{ u_H^1, u_H^2, u_H^3, \frac{1}{2}(u_H^2 + u_H^3), \frac{1}{2}(u_H^3 + u_H^1), \frac{1}{2}(u_H^1 + u_H^2) \right\}^T, \end{aligned}$$

$$\begin{aligned} \mathbf{A} &= \int_{T_H} p\nabla\phi_h(x,y)\nabla\phi_h^T(x,y) + q\phi_h(x,y)\phi_h^T(x,y)\mathrm{d}\Omega, \\ \mathbf{f} &= \int_{T_H} f\bar{\phi}_h(x,y)\mathrm{d}\Omega, \\ \mathbf{g} &= \int_{\partial T_H \cap \Gamma_N} g\bar{\phi}_h(x,y)\mathrm{d}\Gamma, \\ \mathbf{B} &= \int_{T_H} p\nabla\phi_h(x,y)\nabla\bar{\phi}_h^T(x,y) + q\phi_h(x,y)\bar{\phi}_h^T(x,y)\mathrm{d}\Omega, \\ \mathbf{d} &= \int_{\partial T_H \cap \varepsilon_I} \frac{1}{2}\left(p\frac{\partial u_H}{\partial n}\bigg|_{\partial T_H} + p\frac{\partial u_H}{\partial n}\bigg|_{\partial T_H'} \right)\bar{\phi}_h(x,y)\mathrm{d}\Gamma. \end{aligned}$$

in which $\bar{\phi}_h = \phi_h - \phi_H$, and

$$\begin{aligned} \phi_h =& \{\phi_h^1(x,y), \phi_h^2(x,y), \cdots, \phi_h^6(x,y)\}^T, \\ \phi_H =& \{\phi_H^1(x,y), \phi_H^2(x,y), \phi_H^3(x,y), 0, 0, 0\}^T. \end{aligned}$$

Then we obtain the error estimator on element T_H

$$||e_{T_H}||^2 = \mathbf{e}_{T_H}^T \mathbf{A}\mathbf{e}_{T_H}.$$

It is well known that in (15) $\mathbf{A}$ is a 6×6 semi positive definite matrix with rank being 5, no unique solution can be found from solving problem (15). Let's take the Poisson problem for an example to illustrate how to add condition to make the problem well posed. For Poisson problem $\mathbf{A} = \int_{T_H} p\nabla\phi_h(x,y)\nabla\phi_h^T(x,y)\mathrm{d}\Omega$, if we add the zero mean condition on e_{T_H}, i.e. $\int_{T_H} e_{T_H}\mathrm{d}\Omega = 0$, the unique solution can be insured [6]. According to (12) we have

$$e_{T_H}^1 + e_{T_H}^2 + e_{T_H}^3 + 3e_{T_H}^4 + 3e_{T_H}^5 + 3e_{T_H}^6 = 0.$$

Let $\mathbf{h} = (1,1,1,3,3,3)^T$, adding $\mathbf{h}^T\mathbf{e}_{T_H} = 0$ to (15) leads to

$$\bar{\mathbf{A}}\mathbf{e}_{T_H} = \bar{\mathbf{c}}$$

in which

$$\bar{\mathbf{A}} = \left\{ \begin{array}{c} \mathbf{A} \\ \mathbf{h}^T \end{array} \right\}, \quad \bar{\mathbf{c}} = \left\{ \begin{array}{c} \mathbf{c} \\ 0 \end{array} \right\}.$$

Now there are 7 equations used to determine 6 unknown variables, because the row rank of $\bar{\mathbf{A}}$ is 6, the unique solution can be obtained.

4 Examples

The first example is a Poisson problem, looking for u satisfying

$$-\nabla^2 u = \sin(16xy),$$

within the domain of a square $\Omega = [0,1] \times [0,1]$, and restricted to the Dirichlet boundary $u = 0$ on $\Gamma_D = AB \cup BC \cup CD \cup CA$. The Fig. 2 shows the initial mesh of the problem [15]. The second example is a Laplace equation, looking for u satisfying

$$-\nabla^2 u = 0,$$

and restricted to the Dirichlet boundary $u = 0$ on AF, and the Neumann boundary $\partial u/\partial n = y(1-y)$ on ED, refer to the Fig. 3.

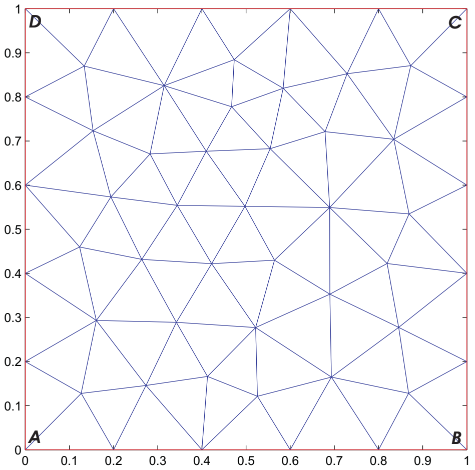


Figure 2: Initial mesh of example 1 with 54 nodes and 86 elements.

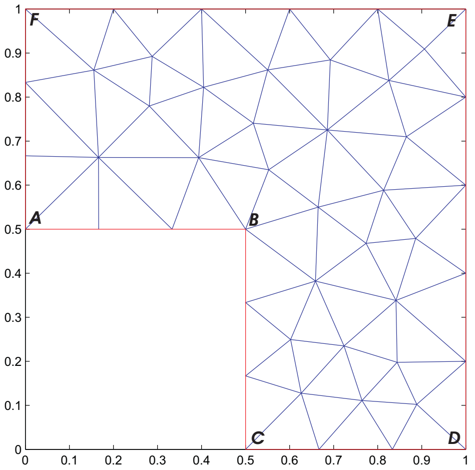


Figure 3: Initial mesh of example 2 with 48 nodes and 72 elements.

Table 1: The results of $||u||^2$ in Example 1.

Smoothed		Non smoothed	
NoE	$\lvert\lvert u\rvert\rvert^2$	NoE	$\lvert\lvert u\rvert\rvert^2$
86	3.13433E-3	86	3.13433E-3
165	3.39858E-3	165	3.39244E-3
376	3.69418E-3	425	3.66329E-3
981	3.85979E-3	801	3.77872E-3
1310	3.99103E-3	1503	3.91938E-3
2134	4.08233E-3	2438	4.06015E-3
4356	4.13219E-3	6119	4.12822E-3
6350	4.15239E-3	8574	4.14138E-3
11470	4.17093E-3	21016	4.17172E-3

Table 2: The results of $||u||^2$ in example 2.

Smoothed		Non smoothed	
NoE	$\lvert\lvert u\rvert\rvert^2$	NoE	$\lvert\lvert u\rvert\rvert^2$
72	0.830817	72	0.830817
106	0.926109	104	0.929284
205	0.986871	258	0.990728
417	1.017258	509	1.020839
684	1.031505	1318	1.036129
1227	1.037550	2052	1.040352
3763	1.044715	5129	1.045198
5444	1.046297	8286	1.046685
8315	1.047112	14653	1.047680

For the mesh smoothing, we use JIGGLEMESH function of MATLAB to do the smoothing in each iteration, each node that is not located on an edge segment is moved toward the center of mass of the polygon formed by the adjacent triangles. The grammar of the function is `NP=jigglemesh(DP, DE, DE, OPT, ITER)`, in which `NP` is the jiggled coordinates of the nodes, `DP`, `DE` and `DE` are the non jiggled mesh data. This process is repeated according to the setting of the `Opt` and `Iter` variables.

Table 1 gives the results of $||u||^2$ with respect to NoE, the number of elements, in the two situations of smoothing and non-smoothing of meshes for example 1. Fig. 4 and Fig. 5 show the adaptive meshes in example 1 with smoothing and non-smoothing in some iterations of computing, respectively. In the captions of the figures, we also give the relative error of the energy norm of solutions, $||e_r||$, on different meshes.

Table 2 gives the results of $||u||^2$ with respect to NoE, in the two situations of smoothing and non-smoothing of meshes for example 2.

Fig. 6 and Fig. 7 show the adaptive meshes in example 2 with smoothing and non-smoothing in some

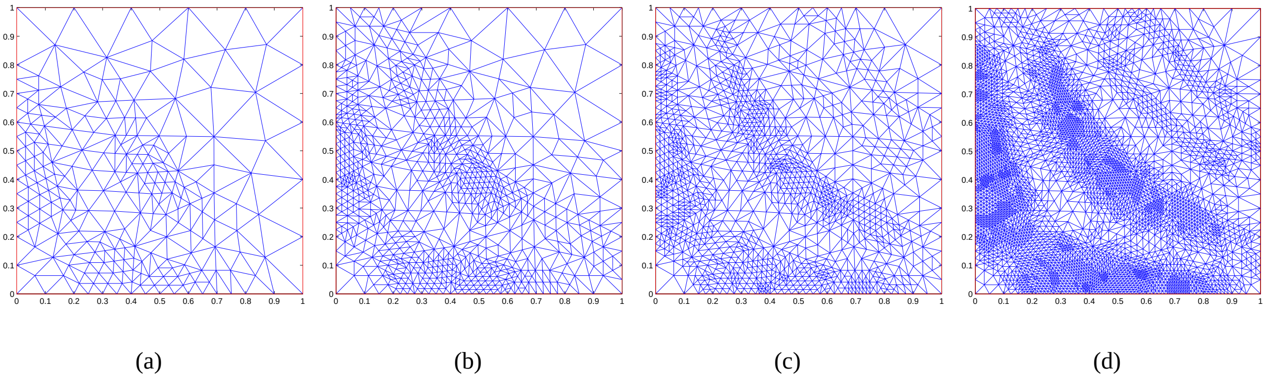

(a) (b) (c) (d)

Figure 4: Non smoothed meshes in example 1. (a) 234 nodes, 425 elements, $||e_r|| = 3.5716\%$; (b) 794 nodes, 1503 elements, $||e_r|| = 2.5801\%$; (c) 1274 nodes, 2438 elements, $||e_r|| = 1.8178\%$; (d) 4388 nodes, 8574 elements, $||e_r|| = 1.1705\%$.

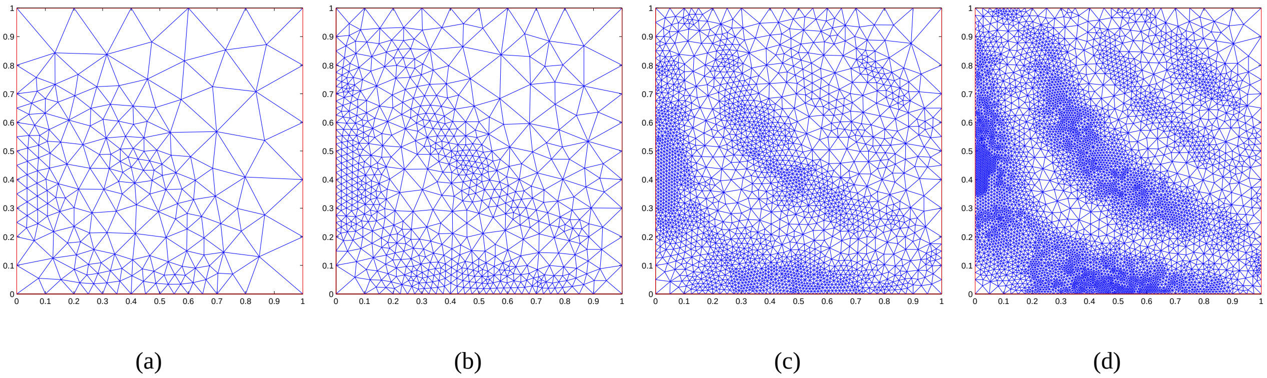

(a) (b) (c) (d)

Figure 5: Smoothed meshes in example 1. (a) 208 nodes, 376 elements, $||e_r|| = 3.4670\%$; (b) 695 nodes, 1310 elements, $||e_r|| = 2.2250\%$; (c) 2249 nodes, 4356 elements, $||e_r|| = 1.2605\%$; (d) 5854 nodes, 11470 elements, $||e_r|| = 0.8163\%$.

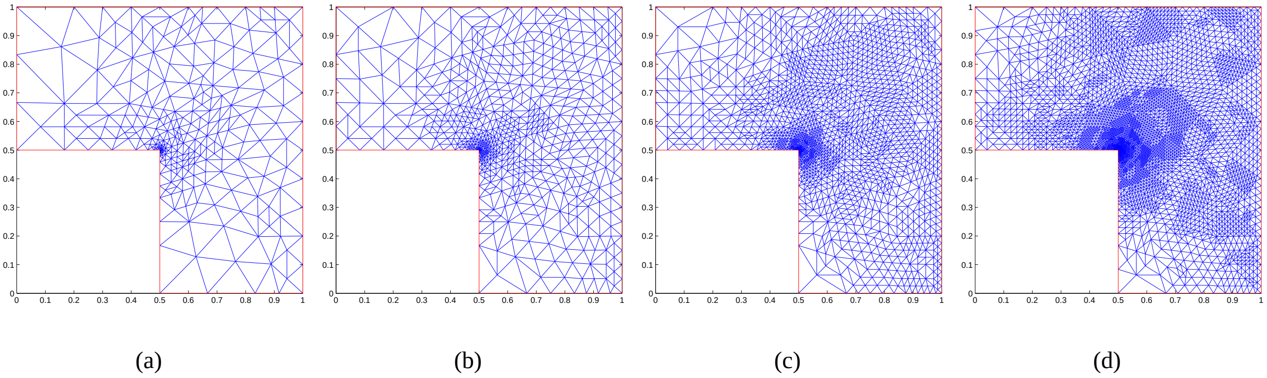
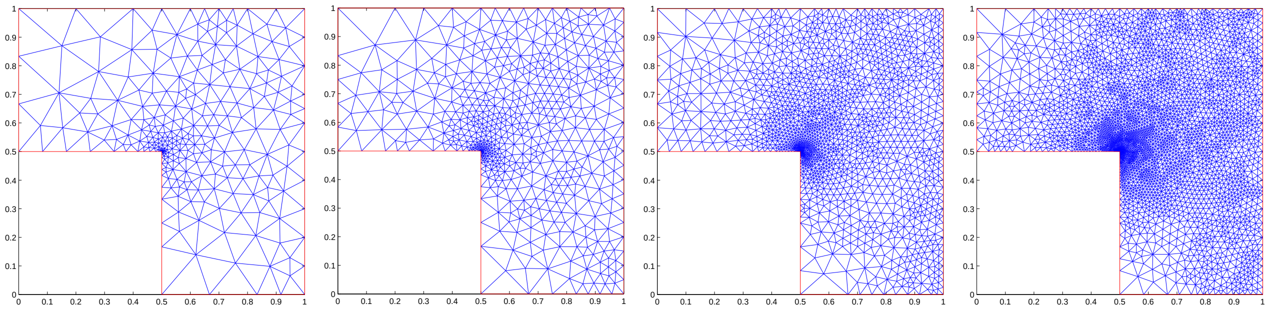
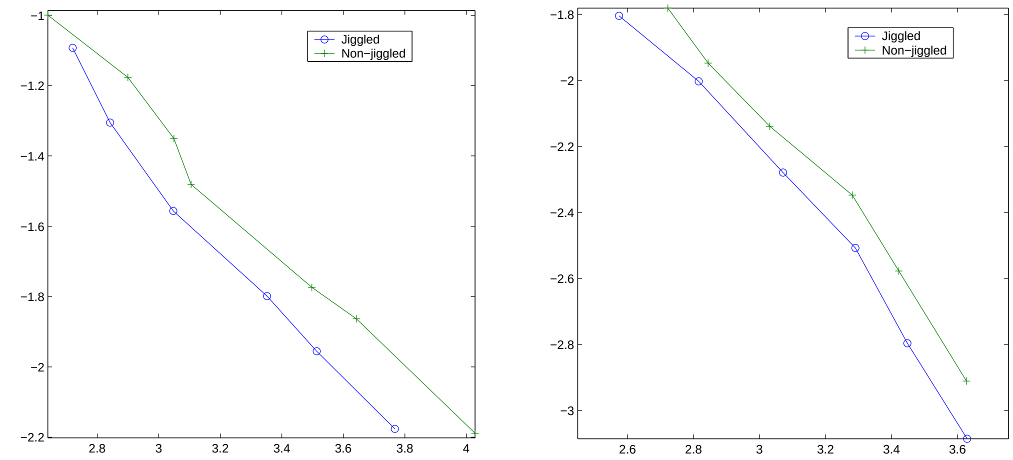

(a) (b) (c) (d)

Figure 6: Non smoothed meshes in example 2. (a) 279 nodes, 509 elements, $||e_r|| = 1.6253\%$; (b) 698 nodes, 1318 elements, $||e_r|| = 1.0879\%$; (c) 1909 nodes, 3691 elements, $||e_r|| = 0.7095\%$; (d) 4237 nodes, 8286 elements, $||e_r|| = 0.4204\%$.



(a) (b) (c) (d)

Figure 7: Smoothed meshes in example 2. (a) 285 nodes, 515 elements, $||e_r|| = 1.4868\%$; (b) 654 nodes, 1227 elements, $||e_r|| = 1.0237\%$; (c) 1952 nodes, 3763 elements, $||e_r|| = 0.6038\%$; (d) 4255 nodes, 8315 elements, $||e_r|| = 0.3688\%$.



(a) (b)

Figure 8: (a) The convergence rate for example 1, $\lg ||e_r||^2$ vs $\lg N$. (b) The convergence rate for example 2, $\lg ||e_r||^2$ vs $\lg N$.

iterations of computing, respectively. Fig. 8 shows the convergence rates for the two examples, $\lg ||e_r||e^2$ vs $\lg N$, and N is the number of DOFs. We can find that they have the same convergence rates for both the smoothed and non smoothed meshes, but the non smoothed meshes lead to a higher accuracy of the results.

5 Conclusions

An efficient mesh adaptive method is presented in this paper, compared with the other mesh refined methods, this method has some advantages. Firstly, this is an error indicator based on elemental computing, and it needs only to calculate the error between the coarse mesh and the regular refined mesh with 6 elements, thus the analytical equations can be obtained for the error indicator of each element. Secondly, the well posedness of the resulting equation for error indicators is insured, due to the mapping between the coarse elements. Thirdly, the implementation of the method is very simple, because the algebra equation has been derived, it is not a hard work to program the matrices computing with Matlab. Fourthly, it is very efficient, because on each element, we only need to solve a low dimensional algebra equation. At last, the examples show the method stable and efficient, and the smoothed mesh also increases the accuracy of results.

Acknowledgements: This work is supported by the National Natural Science Foundation of China under grant number 11172209.

References:

[1] Z.C. Xuan, Y.H. Li and M. Guan, The energy relaxation method for the verification of finite element analysis, *Applied Mechanics and Materials*, 478, 2014, pp 299-302.

[2] Z.C. Xuan, X.K. Zhang, M. Guan, The verification of the quantities of interest based on energy norm of solutions by node-based smoothed finite element method, *Advances in Engineering Software*, 71, 2014, pp 1-8.

[3] K.H. Lee, Z.C. Xuan, Computing a-posteriori bounds for stress intensity factors in elastic fracture mechanics, *Internationl Journal of Fracture*, 126, 2004, pp 123-142.

[4] I. Babuska, W.C. Rheinboldt, A posteriori error estimates for finite element computations, *SIAM Journal of Numerical Analysis*, 15, 1978, pp 736-754.

[5] R.E. Bank, A. Weiser, Some a posteriori error estimators for elliptic partial differential equations, *Mathematics of Computation*, 44, 1985, pp 283-301.

[6] J.T. Marti, *Introduction to Soblev Spaces and Finite Element Solution of Elliptic Boundary Value Problems*, Academic Press, London-New York, 1986.

[7] R. Verfurth, A posteriori error estimation and adaptive mesh-refinement techniques, *Journal of Computational and Applied Mathematics*, 50, 1994, pp 67-83.

[8] D.A. Field, Laplacian smoothing and Delaunay triangulations, *Communications in Applied Numerical Methods*, 4, 1998, pp 709-712.

[9] P. Ladevèze, J.T. Oden, *Advances in adaptive computational methods in mechanics*, Elsevier, 1998.

[10] L.R. Herrmann, Laplacian-isoparametric grid generation scheme, *Journal of the Engineering Mechanics Division*, 102, 1976, pp 749-756.

[11] G.A. Hansen, R.W. Douglass, A. Zardecki, *Mesh enhancement*, Imperial College Press, p. 404, 2005.

[12] M. Attene, M. Campen and L. Kobbelt, Polygon mesh repairing: An application perspective, *ACM Computing Surveys*, 45, 2013, 15:1-15:33.

[13] L. He, S. Schaefer, Mesh denoising via L_0 minimization, *ACM Transactions on Graphics*, 32, 2013, pp 64:1-64:8.

[14] H. Yu, S. Qin, G. Sun and D.K. Wright, On generating realistic avatars: dress in your own style, *Multimedia Tools and Applications*, 59, 2012, pp 973-990.

[15] Z.C. Xuan, Optimal finite element mesh refinement based on a-posteriori error estimator and the quality of mesh, Proceedings of the 2011 4th International Conference on Intelligent Networks and Intelligent Systems, Tianjin 2011, China.