

1. Introduction

In practical problems, we frequently encounter problem with non-regular domain such as re-entrant corner. In this case, the regularity of the solution is deteriorated. For example, when simulating crack, we see that the first derivative of the solution field blows up at the crack tip. It was observed that the convergence rate of finite element solution deteriorate for problem with this type of solution. One way to address this problem is to adaptively perform local grid refinement. In order to do that we need to compute an a posteriori error estimate to indicate which part of the grid induces large errors. Using this information we can then locally refine the corresponding grid.

1.1 PDE Solver

In all simulation, finite element method with quadratic triangular element will be used.

2. A Posteriori Error Estimates: Residual-Based

The following residual based error estimate will be used.

$$R_T(u_h) = \Delta u_h + f \quad \text{for } T \in \mathcal{T}_h$$

$$\begin{aligned} R_e(u_h) &= \left[\frac{\partial u_h}{\partial n} \right] \quad \text{for } e \in \Gamma_h \\ \eta_{T,R} &= \left\{ h_T^2 \|R_T\|_{0,T}^2 + \frac{1}{2} \sum_{e \subset \partial T} h_e \|R_e\|_{0,e}^2 \right\}^{1/2} \quad \text{for } T \in \mathcal{T}_h \\ \eta_R &= \left\{ \sum_{T \in \mathcal{T}_h} h_T^2 \|R_T\|_{0,T}^2 + \sum_{e \subset \Gamma_h} h_e \|R_e\|_{0,e}^2 \right\}^{1/2} \end{aligned}$$

where u_h is finite element solution, h_T is element size, h_e is edge size, T is a triangle element, and \mathcal{T} is a shape-regular triangulation. For this residual based estimator, there exist a constant $c = c(\Omega, \kappa)$ such that for all $T \in \mathcal{T}_h$,

$$\|u - u_h\|_{1,\Omega} \leq c \eta_R \tag{1}$$

$$\eta_{T,R} \leq c \left\{ \|u - u_h\|_{1,\omega_T}^2 + \sum_{T' \subset \omega_T} h_T^2 \|f - f_h\|_{0,T'}^2 \right\} \tag{2}$$

where u is the exact solution and

$$\omega_T = \cup \{T' \in \mathcal{T} \mid T \text{ and } T' \text{ have a common edge or } T' = T\}$$

The upper bound means that the estimator is reliable and the lower bound means that the estimator is also efficient.

2.1 Smooth problem

Now, we will test this estimate with the following BVP in a domain $\Omega = [-1, 1] \times [-1, 1]$.

$$\begin{aligned}\Delta u + f &= 0 \\ u(x, -1) &= 0\end{aligned}$$

where

$$f(x, y) = 6y$$

The solution of this problem is given by

$$u(x, y) = -y^3 + 3y + 2$$

Three mesh refinements will be used. The mesh, error with respect to exact solution, and error estimate are shown in Fig.1-3.

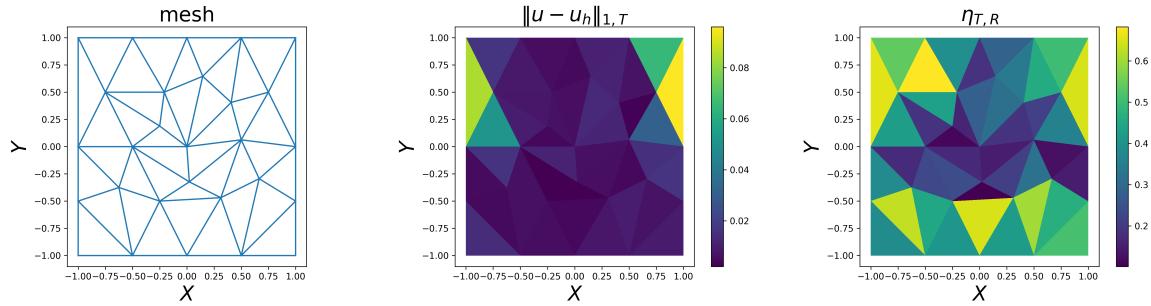


Figure 1: Mesh, error, and error estimate for mesh 1

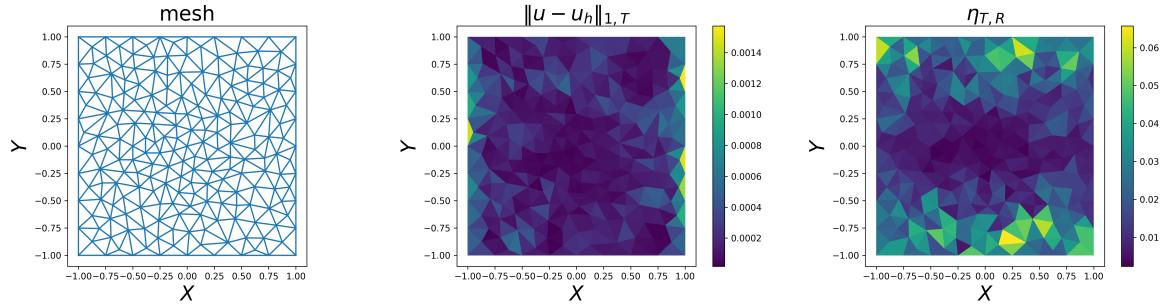


Figure 2: Mesh, error, and error estimate for mesh 2

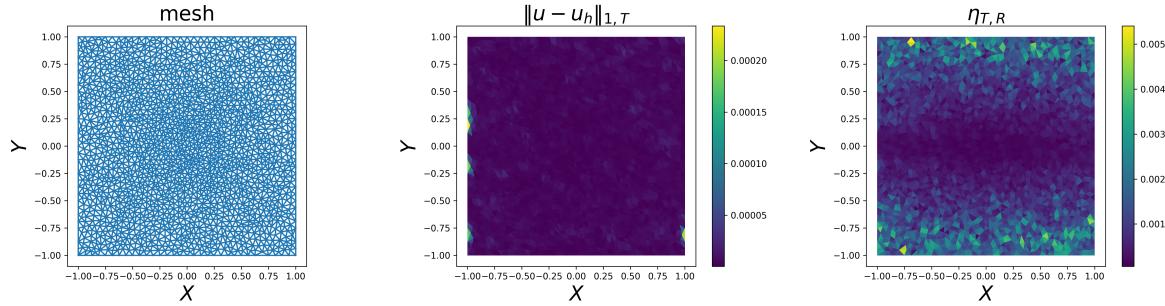


Figure 3: Mesh, error, and error estimate for mesh 3

We can see that the estimate is predicting high error in the region where the derivative of the solution is high. The finite element solution is given in Fig.4.

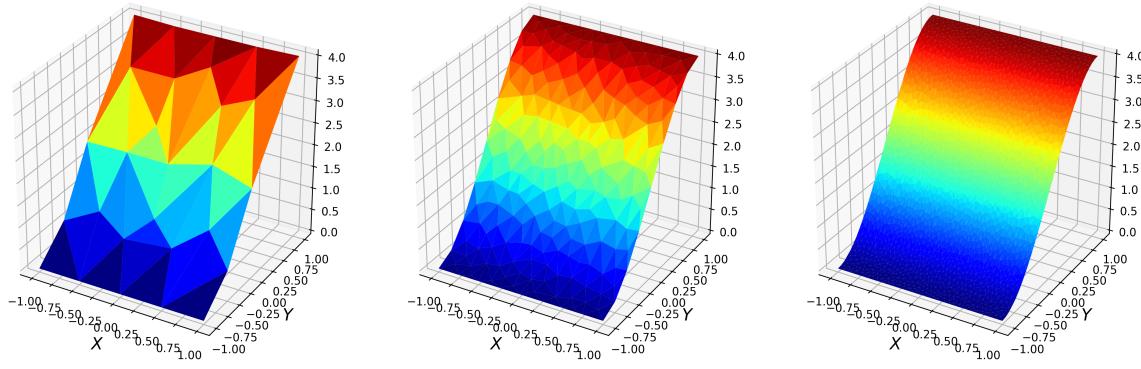


Figure 4: Finite element solution for three different refinements

The efficiency will be shown here. Using $c = 200$, we plot

$$R_{eff} = c \left\{ \|u - u_h\|_{1,\omega_T}^2 + \sum_{T' \subset \omega_T} h_T^2 \|f - f_h\|_{0,T'}^2 \right\} - \eta_{T,R} \quad (3)$$

Fig.5 shows the plot of R_{eff} . The values are all positive, that means eq.2 is satisfied.

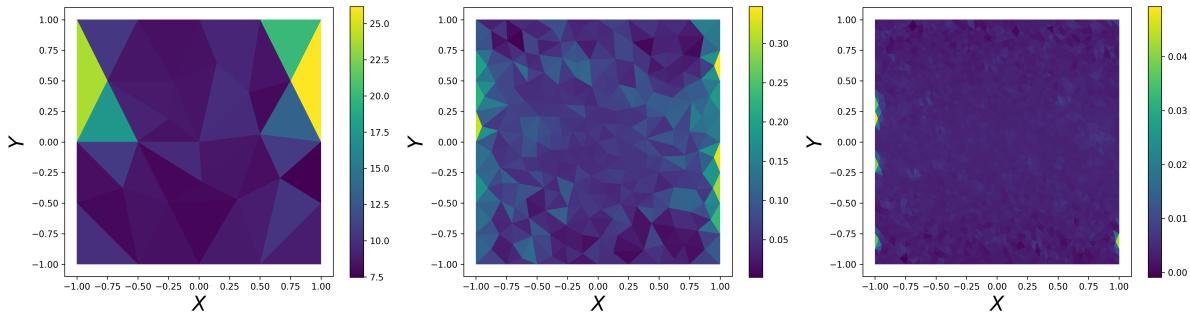


Figure 5: Efficiency of error estimator

Next, we want to show that the reliability claim is also satisfied. Fig.6 shows the evolution of error and the estimate. We can see that the estimate has the roughly the same slope as the error and provide upper bound although quite loose when we incorporate constant c . This plot also shows that the FEM solver has optimal convergence (convergence rate is two in the H_1 norm).

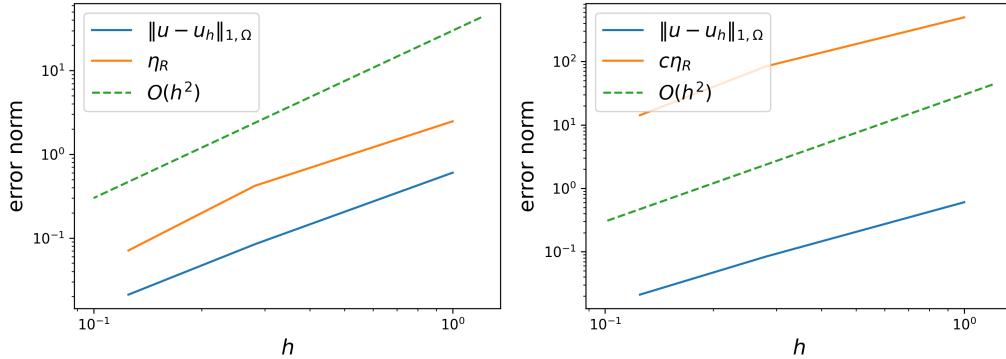


Figure 6: Error evolution in H_1 norm

At last, we also check the convergence rate with relative error in the l_2 norm. It is indeed has optimal convergence.

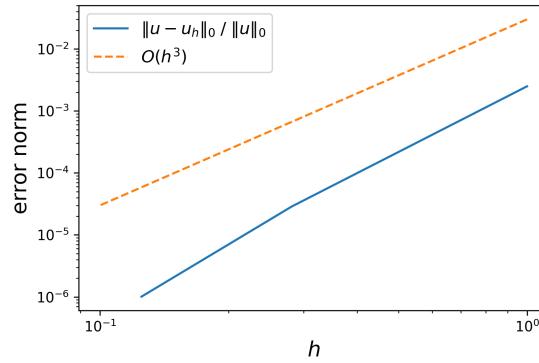


Figure 7: Relative error in l_2 norm

2.1 Domain with re-entrant corner

Now consider the two-dimensional domain with re-entrant corner shown in Fig.8.

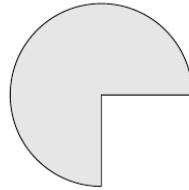


Figure 8: Domain with re-entrant corner

Consider laplace problem in a domain is given by

$$\Omega = \{(x, y) \in \mathbb{R} \mid x^2 + y^2 < 1, x < 0 \text{ or } y > 0\}$$

The BVP is given by

$$\begin{aligned}\Delta u &= 0 && \text{in } \Omega \\ u(e^{i\varphi}) &= \sin\left(\frac{2}{3}\varphi\right) && \text{for } 0 \leq \varphi \leq \frac{3}{2}\pi \\ u &= 0 && \text{elsewhere on } \partial\Omega\end{aligned}$$

The exact solution to this problem is

$$u(z) = \operatorname{Im}(w(z))$$

$$w(z) = z^{2/3}$$

$$z = x + iy$$

Since $w'(z) = 2/3z^{-1/3}$, the first derivative of u are not bounded as $z \rightarrow 0$. Three different mesh refinement will be used to solve this problem. For all refinements, we use 700 element around the curved boundary to minimize error from approximation geometry of the domain. The meshes, error, and error estimates are shown in Fig.

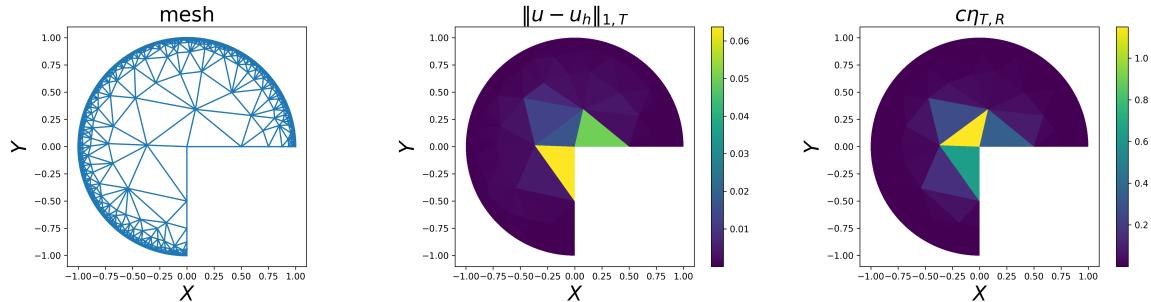


Figure 9: Mesh, error, and error estimate for mesh 1

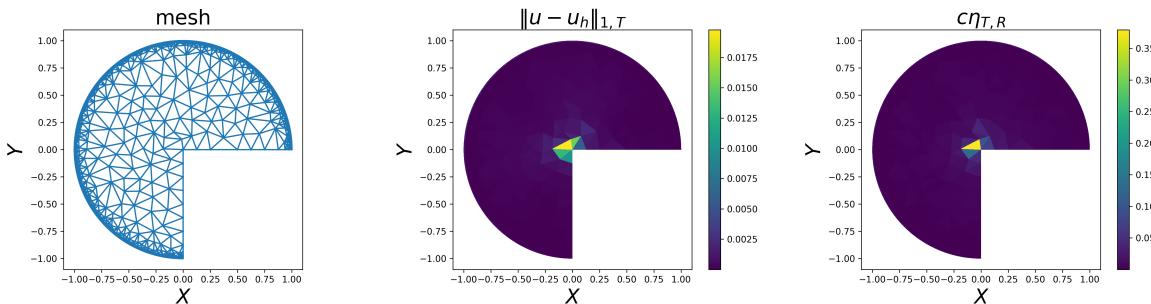


Figure 10: Mesh, error, and error estimate for mesh 2

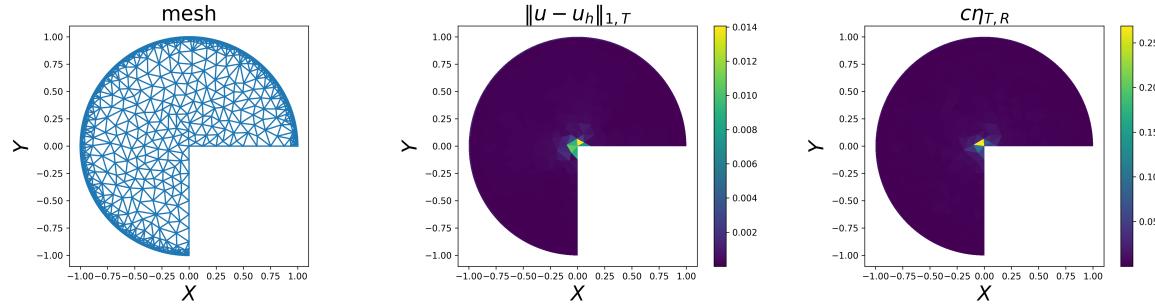


Figure 11: Mesh, error, and error estimate for mesh 3

We can see that high error concentration in the region with singularity. In this case, the estimator performed quite well to predict the high error regions. The solution of this problem with several different refinements are shown in Fig.12.

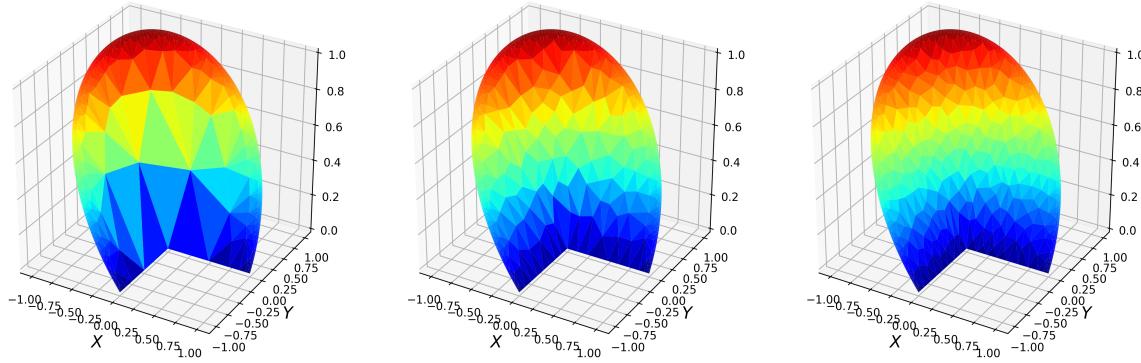


Figure 12: Finite element solution for three different refinements

Next, we check efficiency of the estimator the same way as in the previous section. Fig.13 shows the plot of R_{eff} according to eq.3 with $c = 200$. This shows that the estimator is indeed satisfies eq.2.

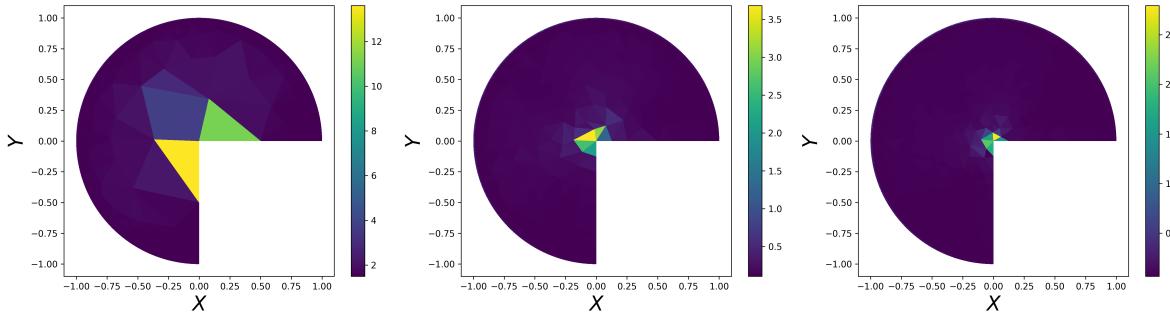
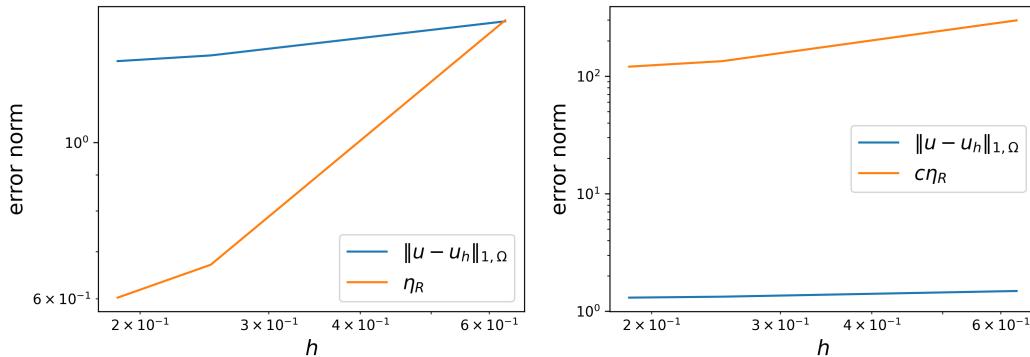
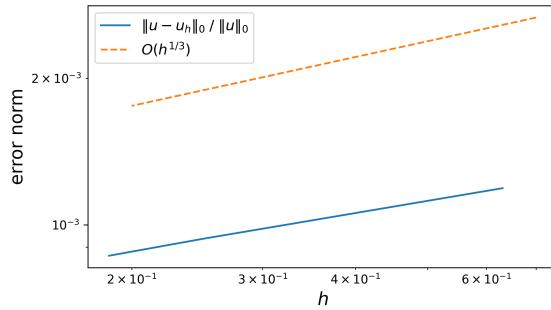


Figure 13: Efficiency of error estimator

We also check the reliability of the error estimator. Using the same value of c as the one used for efficiency, we can see in Fig.14 that the estimator is indeed providing upper bound for the error.

Figure 14: Error evolution in H_1 norm

From Fig.14 it seems the real error decrease very slowly. In Fig.15, we can see that we do not have optimal convergence. This is because the regularity of the solution of this problem. The high error was concentrated in one small region such that refining the whole domain doesn't do much improvement. We will address this problem using adaptive refinement in the next section.

Figure 15: Relative error in the l_2 norm

3. Adaptive Refinement in Finite Element Method

In this section, we will see how adaptive finite element method address the convergence issue in the previous section. The strategy is to have an initial mesh, run the simulation with the mesh, do error estimate, and locally refine the mesh in the region where the estimator gives high error. The following algorithm from Rivara (1984) will be used. Let \mathcal{T}_0 be any conforming initial triangulation and let $\mathcal{T}_0 \subset \mathcal{T}_0$ be the set of triangles to be refined. Then a new (locally) refined triangulation is obtained as follows:

1. Bisect T by its longest side, for each $T \in \bar{\mathcal{T}}_0$.
2. Find $R_1 \subset \mathcal{T}_0$ the set of non-conforming triangles generated in step 1. Set $k \leftarrow 1$.
3. For each $T \in R_k$ (with the non-conforming point $P \in T$): (a) bisect T by the longest side; (b) if P is not over the longest side of T , then join P with the midpoint of the longest side.
4. Let \mathcal{T}_0^k be the triangulation generated in step 3. Find $R_{k+1} \subset \mathcal{T}_0^k$ the set of non-conforming triangles generated in step 3.

5. Is cardinality of R_{k+1} equal to 0? (a) If yes, stop. The conforming triangulation was obtained.
 (b) If not, set $k \leftarrow k + 1$ and go to 3.

Next, we will solve the following Laplace problem in a V-notched domain Ω .

$$\begin{aligned}\Delta u &= 0 \\ u(x, -1) &= 0 \\ u(x, 1) &= 0 \quad \text{for } x \geq 0 \\ u(x, 1) &= 0.5 \quad \text{for } x < 0\end{aligned}$$

The domain and initial mesh is shown in Fig.16. This figure also show the error estimate. Based on this error estimate, we can set a threshold to determine which element needs to be refined. For this case, the threshold is

$$\eta_{T,R} > 0.5 \left(\max_{T \in \mathcal{T}_h} \eta_{T,R} \right)$$

Which means element with more than fifty percent of maximum element error in the domain will be refined

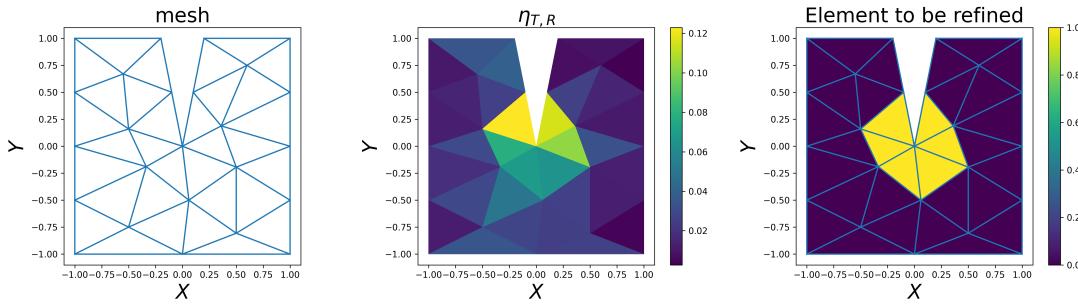


Figure 16: Initial mesh, error estimate, and list of elements that needs refinement

Fig.17-21 shows several stages of the refinement process. Let's denote the number of iteration by k such that $k = 1$ means the first iteration.

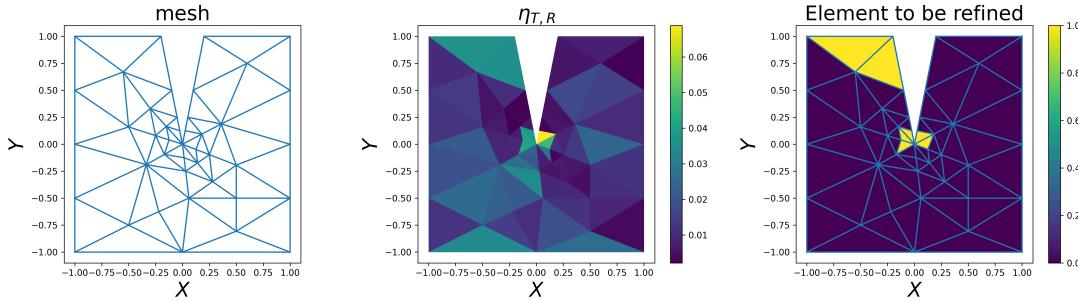
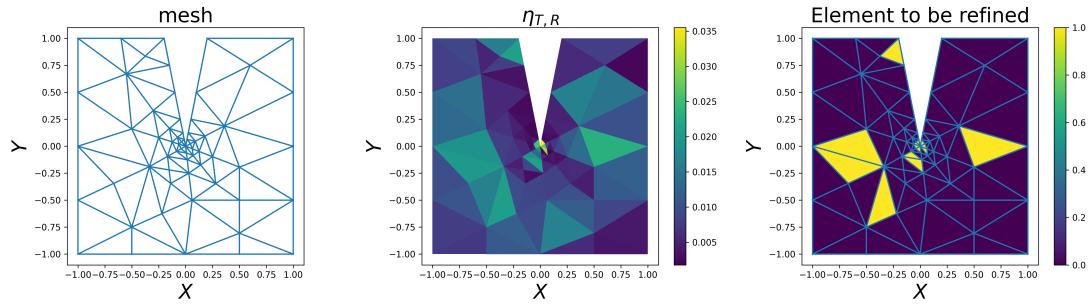
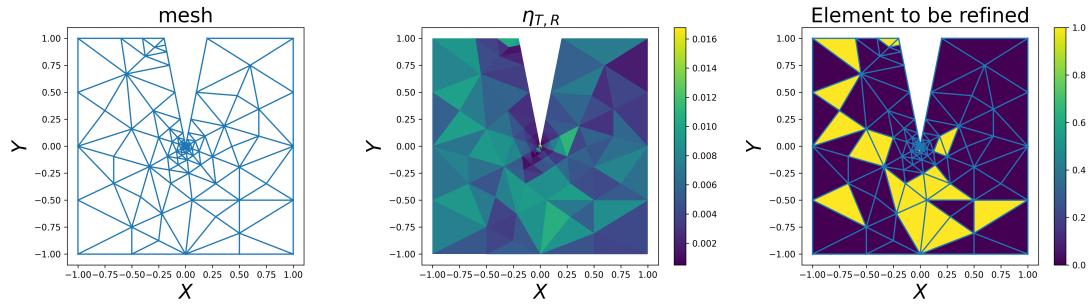
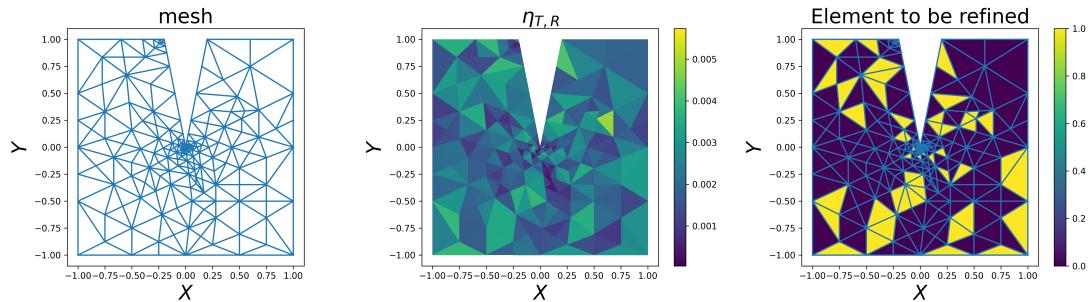
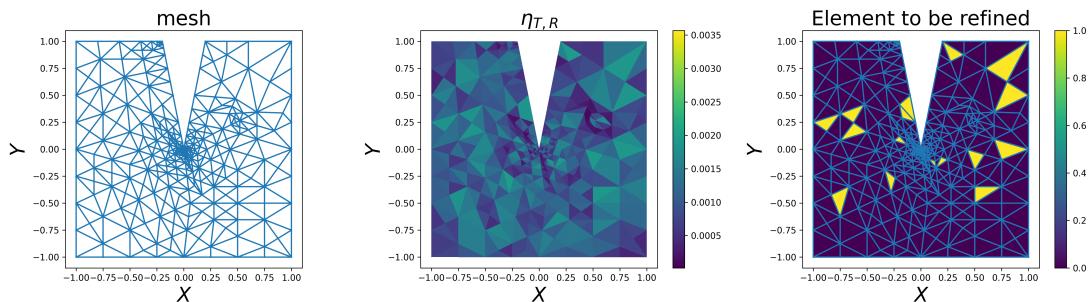


Figure 17: Mesh, error estimate, and list of elements that needs refinement ($k=3$)

Figure 18: Mesh, error estimate, and list of elements that needs refinement ($k=6$)Figure 19: Mesh, error estimate, and list of elements that needs refinement ($k=10$)Figure 20: Mesh, error estimate, and list of elements that needs refinement ($k=16$)Figure 21: Mesh, error estimate, and list of elements that needs refinement ($k=19$)

We can see that in the beginning, the refinement focused on elements near the notch. Eventually, elements near the notch become small enough that the error of the larger elements away from notches become comparable to error around the notch. At this point, the algorithm starts refining bigger elements away from the notch while still refining elements around the notch while doing so. The finite element solution of this problem is shown in Fig.

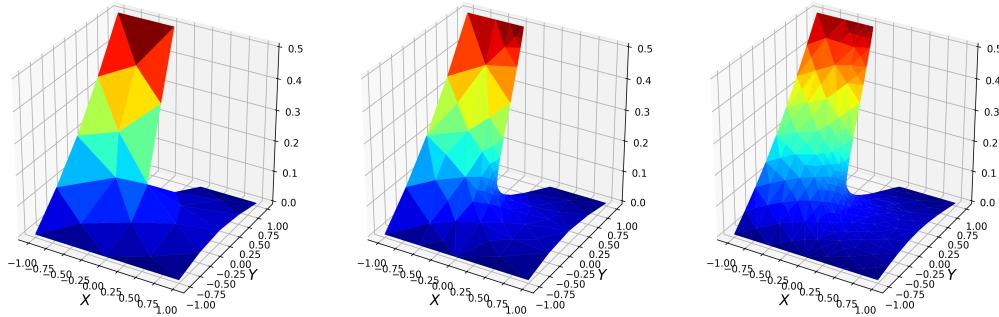


Figure 22: Adaptive finite elements solution for three different iteration ($k = 0$, $k = 10$, $k = 19$)

Fig.23 shows error reduction throughout iterations. In the early iteration, the reduction is slow. It is until the elements around the notch become small enough we have better error reduction because the algorithm starts refining bigger elements away from notch. Fig.24 shows error reduction with respect to number of dof. In 2-D, $O(N_{dof}^1) \approx O(h^{-2})$. This means, the figure shows that we achieve optimal convergence rate ($O(h^2)$) with adaptive mesh refinement.

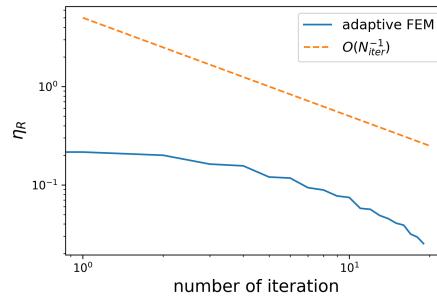


Figure 23: Evolution of error throughout iterations

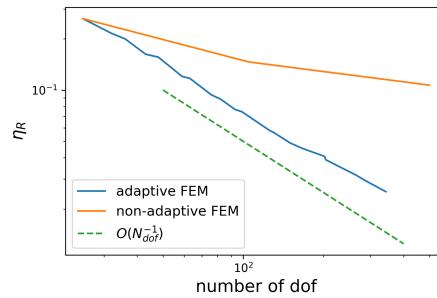


Figure 24: Error reduction with respect to number of dof

4. Discussion

This section includes topics that have not been investigated further because of time constraint.

4.1 Time complexity

We are also concerned about time complexity of this algorithm. Fig.25 shows the required computational time for refining a varying number of elements with constant total number of elements in the domain. From this figure, it seems that the computational time depends more on the total number of elements in the domain. This may be specific to my implementation because I loop over all elements to search for nonconforming elements. There should be a better way to implement this but here we will look at how this implementation perform. Fig.26, shows the required computational time for varying number of total element in the domain. The complexity of this implementation is $O(N_{elem}^2)$.

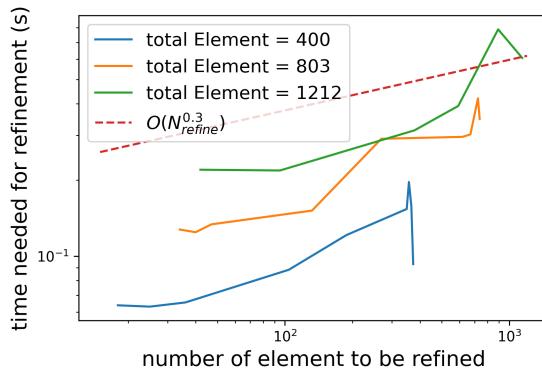


Figure 25: Computational time for varying number of refined element and constant number of total element in the domain

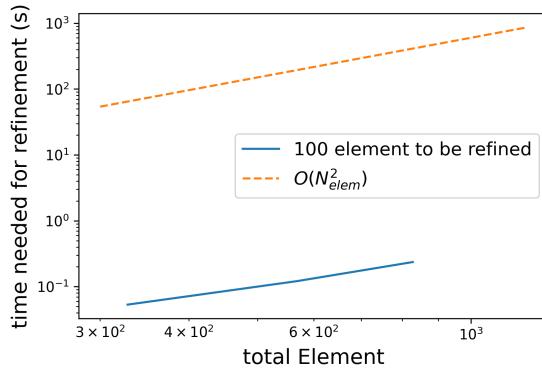


Figure 26: Computational time for constant number of refined element and varying number of total element in the domain

This is not an optimal implementation. The better way is to record neighbor elements when we do bisection and loop on this list instead of all elements. Further study will be needed to know the complexity of this algorithm with the best implementation.

4.1 Domain with cups

Although adaptive finite element can be used to address problem with re-entrant corner domain, it will not be able to address the problem where the domain has cups. In the re-entrant corner problem, the actual problem is related to efficiency. The high error is concentrated over a small region so refining in all region becomes really inefficient. This was addressed by the adaptive strategy. The source of problem for cups is different. It is a mathematical problem instead of efficiency problem. Therefore, the adaptive strategy is not the way to address it.

A. How to Reproduce Results

On the uploaded code, starting at line 1460, there are several commented function call. To run the simulation and recreate figures in the report, just un-comment the corresponding function and run the code.

References

- [1] Braess, D., (2007). Finite elements theory, fast solvers, and applications in elasticity theory.
- [2] Rivara, M.C., 1984. Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. *Int. J. Numer. Meth. Engng.* 20, 745–756.
<https://doi.org/10.1002/nme.1620200412>