A Pollution Adaptive Mesh Generation Algorithm in r-h Version of the Finite Element Method

Soo Bum Pyun and Hyeong Seon Yoo

Department of Computer Science, Inha University,
Inchon, 402-751, South Korea
hsyoo@inha.ac.kr

Abstract: In this paper, we propose a simplified pollution adaptive mesh generation algorithm, which concentrate on the boundary node based on the element pollution error indicator. The automatic mesh generation method is followed by either a node-relocation or a node-insertion method. The boundary node relocation phase is introduced to reduce pollution error estimates without increasing the boundary nodes. The node insertion phase greatly improves the error and the factor with the cost of increasing the node numbers. It is shown that the suggested r-h version algorithm converges more quickly than the conventional one.

1 Introduction

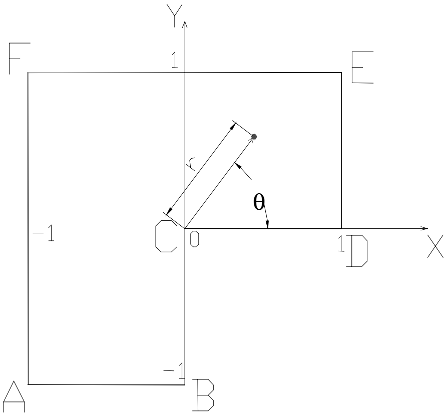
Most engineering problems are described in polygonal domains with geometric singularities. These singularities make the solution diverge to infinity and cause the conventional error estimators to severely underestimate the error in any patch outside the neighborhood of the singular point. Since Babuska's works about error estimators and pollution errors it is known that the pollution error estimates are much more than the local error ones [1,2,3,4]. It was demonstrated that the conventional Zienkiewicz-Zhu error estimator [5,6,7,8] was insufficient and should include a pollution error indicator [1,4]. The pollution-adaptive feedback algorithm employs both local error indicators and pollution error indicators to refine the mesh outside a larger patch, which includes a patch and one to two surrounding mesh layers [2,3]. The conventional pollution adaptive algorithm bisects the element for every iteration and needs a lot of iterations to converge.

We concentrate only on a problem boundary since the singularities exist on the boundary and mesh sizes change gradually regardless of the mesh generation algorithm. A mesh generation algorithm, which uses a node relocation method (r-method) as well as h-method of the finite element method for boundary elements, is proposed. The algorithm employs a boundary-node relocation at first and then does a node insertion based on the pollution error indicator.

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2073, pp. 928-936, 2001.

2 The Model Problem

Consider a typical *L*-shaped polygon, $\Omega \subseteq \mathbf{R}^2$, with boundaries $\partial\Omega = \Gamma = \Gamma_D \cup \Gamma_N$, $\Gamma_D \cap \Gamma_N = \{\,\}$ where Γ_D is the *Dirichlet* and Γ_N is the *Newmann* boundary (Fig.1).



$$\Gamma_D = CD$$

$$\Gamma_N = AB \cap BC \cap DE \cap EF \cap FA$$

Fig. 1. The L-shaped domain for the model problem

We will consider Laplacian with mixed boundary conditions. Let us $H^1_{\Gamma_D} \equiv \{u \in H^1(\Omega) \,|\, u = 0 \;\; on \;\; \Gamma_D\}$. Then the variational formulation of this model problem satisfies (1).

Find $u_h \in S^p_{h,\Gamma_D}(\Omega) := H^1_{\Gamma_D} \cap S^p_h$ such that

$$B_\Omega(u_h, v_h) = \int_{\Gamma_N} g v_h \quad \forall v_h \in S^p_{h,\Gamma_D} \qquad (1)$$

A patch error was expressed only by a local error, but it was demonstrated that the pollution error should include the patch error . The local error was improved by considering a mesh patch ω_h with a few surrounding mesh layers. The equilibrated residual functional is the same for the local error and the pollution error. But the pollution error was calculated by considering the outside of the larger patch, ω_h.

$$e_h\Big|_{\omega_h} = V_1^{\tilde{\omega}_h} + V_2^{\tilde{\omega}_h} \qquad (2)$$

where, $V_1^{\tilde{\omega}_h}$; local error on $\tilde{\omega}_h$

$V_2^{\tilde{\omega}_h}$; pollution error on $\tilde{\omega}_h$

$\tilde{\omega}_h$; ω_h + a few mesh layers

Let us denote $|||v|||_S = \sqrt{B_S(v,v)}$ energy norm over any domain $S \subseteq \Omega$, then the equation (3) can be a pollution estimator with $\bar{x} \in \omega_h$, [1,2,3].

$$\left\|V_2^{\tilde{\omega}_h}\right\|_{\omega_h} \cong \sqrt{|\omega_h|}\sqrt{\left(\frac{\partial V_2^{\tilde{\omega}_h}}{\partial x_1}(\bar{x})\right)^2 + \left(\frac{\partial V_2^{\tilde{\omega}_h}}{\partial x_2}(\bar{x})\right)^2} \tag{3}$$

3 The Proposed Algorithm

3.1 The Basic Idea

For adaptive control of the pollution error in a patch of interest, the conventional algorithm fixes meshes in the patch and refines meshes outside the patch especially near singularities. The algorithm calculates an element pollution indicator and regularly divides $\gamma\%$ of elements whose pollution indicators are high [2]. This algorithm is as following Fig.2.

```
Let $T_h = T_h^0$
 Compute the finite element solution on $T_h$, $\varepsilon_{\omega_h}$ and
$M_{\omega_h}$;
 While ($M_{\omega_h} > t\%\varepsilon_{\omega_h}$) do
      For(each element )do
          Compute $\mu_\tau$, $\tau \in T_h, \tau \notin \tilde{\omega}_h$;
          If ( $\mu_\tau \geq \gamma \max \mu_\tau$ )
              Subdivided $\tau$ regularly;
          Endif
      Endfor
    Compute the finite element solution on $T_h$ and
    $\varepsilon_{\omega_h}$, $M_{\omega_h}$;
 Endwhile
```

Fig. 2. Structure of the conventional algorithm

In Fig.2 we denote the element pollution error M_{ω_h}, the local error ε_{ω_h} and the element pollution indicator μ_τ [2]. Since the conventional algorithm bisects the element length, it could be accelerated if we have smaller boundary elements near the singular points. Therefore it is natural to think about combining r and h method.

In our proposed algorithm, we concentrated only on boundary nodes and whole interior area is triangulated automatically by the constrained Delaunay algorithm [9]. Our algorithm employs two ideas for the control of the boundary-nodes. The first is to relocate a boundary node. It makes boundary nodes near a singular point close to the point. The other is to insert a node between the boundary nodes of elements whose pollution indicators are larger than the specified value. In the relocation phase, the new boundary element length is calculated by using the following relationship between the pollution error estimator and the element size [1,11].

Let $$\left|\!\left|\!\left| V_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h} \approx h^{2\lambda+1} \qquad (4)$$

where λ ; the exponent for singular point.

From this expression, we can deduce old and new element length as following,

$$\left|\!\left|\!\left| \mathrm{V}_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,\mathrm{old}} = \mathrm{Ch}_{\mathrm{old}}^{2\lambda+1} \qquad (4')$$

$$\left|\!\left|\!\left| \mathrm{V}_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,\mathrm{new}} = \mathrm{Ch}_{\mathrm{new}}^{2\lambda+1} \qquad (4'')$$

Combining two equations, we obtain $\mathrm{h}_{\mathrm{new}}$,

$$h_{new} = h_{old} \times \left(\left|\!\left|\!\left| V_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,old} / \left|\!\left|\!\left| V_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,new} \right)^{-\frac{1}{2\lambda+1}} \qquad (5)$$

In order to get the pollution error smaller than the local error we use $t\varepsilon_{\omega_h} \approx t \left|\!\left|\!\left| V_1^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h}$ instead of $\left|\!\left|\!\left| \mathrm{V}_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,\mathrm{new}}$. t is a user-specified constant between 0 and 1. And $\left|\!\left|\!\left| \mathrm{V}_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h,\mathrm{old}}$ will be $\mu_\tau \approx \left|\!\left|\!\left| V_2^{\tilde{\omega}_h} \right|\!\right|\!\right|_{\omega_h} / \sqrt{|\omega_h|}$ since the pollution error consists of the element pollution error indicators outside $\tilde{\omega}_h$. Finally the new element size becomes,

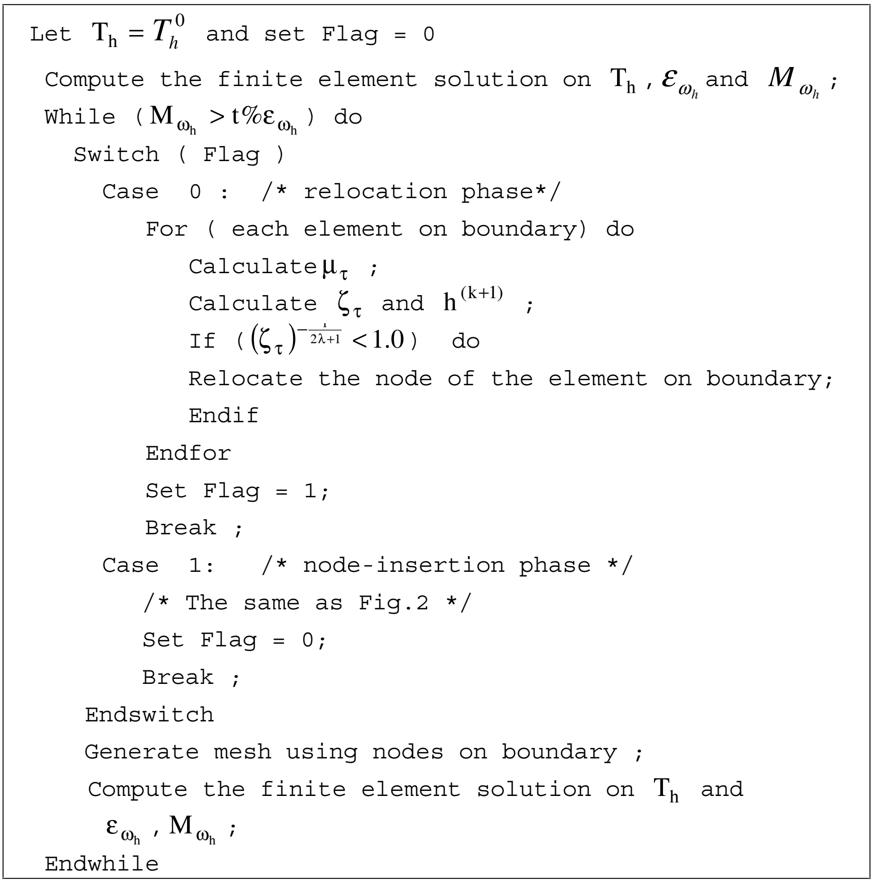
$$\mathrm{h}_{\mathrm{new}} = \mathrm{h}_{\mathrm{old}} \times (\zeta_\tau)^{-\frac{1}{2\lambda+1}} \qquad (6)$$

where $$\zeta_\tau \equiv \frac{\mu_\tau}{t\varepsilon_{\omega_h}}$$

This new element size has an effect on the location of the boundary node, especially the nodes on BC and CD in Fig 1. If the ratio of the element length $(\zeta_\tau)^{-\frac{\;}{2\lambda+1}}$ is less than 1, the algorithm moves the node to the singular point. But if it is greater than 1, the new length is discarded and the location of the node remains fixed to have stable solution. This relocation method is for reducing the number of iteration to get the final mesh. The boundary node insertion phase takes part in a high quality of the error estimator, this phase is the same as others [1,2,3].

3.2 The Proposed Algorithm

A binary number Flag is employed to alternate the boundary relocation and the node insertion process. If the flag is 0, the relocation phase is performed. Figure 3 shows the entire procedure.



```
Let $T_h = T_h^0$ and set Flag = 0
 Compute the finite element solution on $T_h$ , $\varepsilon_{\omega_h}$ and $M_{\omega_h}$ ;
 While ( $M_{\omega_h} > t\%\varepsilon_{\omega_h}$ ) do
   Switch ( Flag )
     Case  0 :  /* relocation phase*/
        For ( each element on boundary) do
           Calculate $\mu_\tau$ ;
           Calculate $\zeta_\tau$ and $h^{(k+1)}$ ;
           If ( $(\zeta_\tau)^{-\frac{1}{2\lambda+1}} < 1.0$ )  do
           Relocate the node of the element on boundary;
           Endif
        Endfor
        Set Flag = 1;
        Break ;
     Case  1:   /* node-insertion phase */
        /* The same as Fig.2 */
        Set Flag = 0;
        Break ;
    Endswitch
    Generate mesh using nodes on boundary ;
    Compute the finite element solution on $T_h$ and
     $\varepsilon_{\omega_h}$ , $M_{\omega_h}$ ;
 Endwhile
```

Fig. 3. Structure of the proposed algorithm

The algorithm starts with the initial mesh and set Flag 0. The boundary node relocation is controlled by $(\zeta_\tau)^{-\frac{1}{2\lambda+1}}$. If the value is below 1, the element shrinks to singular point. In the node insertion phase, a new node is added on the middle of the boundary element. This r-h method makes fewer nodes on the boundary than the h-version.

The interior mesh generation phase is following the control of nodes on boundaries. This step is performed by the constrained Delaunay method. And the finite element analysis and error estimations are following.

4 Numerical Results and Discussions

We considered the mixed boundary-valued problem for the Laplacian over a L-shaped domain and applied boundary conditions consistent with the exact solution

$u(r,\theta) = r^{\frac{1}{3}} \sin(\frac{1}{3}\theta)$ [1]. An interior patch element ω_h far from the singular point as in Fig.4 is chosen. The L-shaped domain is meshed by uniform quadratic triangles (p=2) with h=0.125. In table.1, we show the numerical results for the model problem. Though the local error estimate (ε_{ω_h}) is almost constant, the pollution error decreases dramatically with iteration in the r-h version.

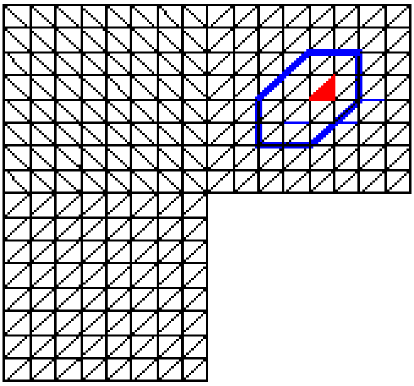


Fig. 4. The initial mesh for numerical example
ω_h : patch, a shaded triangular element
ω_h : large patch, elements enclosed by thick hexagonal line

Table 1. Results of the model problem

Iter.	ε_{ω_h} x E-05		M_{ω_h} x E-05	
	Fig.2	Fig.3	Fig.2	Fig.3
0	7.43	7.43	66.72	66.72
1	7.75	8.29	44.19	21.90
2	7.96	8.33	29.68	7.00
3	8.09	8.33	20.62	6.09
4	8.20	8.32	13.96	2.88
5	8.11		10.01	
6	8.12		7.28	
7	7.99		5.43	
8	8.00		4.45	
9	8.01		3.88	
10	8.01		3.55	
11	8.02		3.34	
12	8.02		3.26	
13	8.02		3.14	
14	8.02		3.13	
15	8.02		3.06	

After the second iterations in h version, the pollution error reduces about half of the initial value. But the pollution error of r-h version is decreased to one of ninth. This significant reduction makes the number of iteration less than that of the conventional one. The pollution factor is defined by the ratio of a pollution error estimate and the local error estimate, $p_{\tau} = IM_{\omega_h} / \varepsilon_{\omega_h}$. In Fig.5 we can see that the pollution factor decrease more rapidly for the proposed algorithm case. In case of the proposed algorithm the pollution factor becomes less than 0.4 only after 4 iterations. From this result, we note that the proposed algorithm controls the pollution error and is effective. The total number of iteration is 4, which is much smaller than one of the conventional algorithms. In Fig. 6 we show the final mesh, which is obtained by the proposed algorithm.

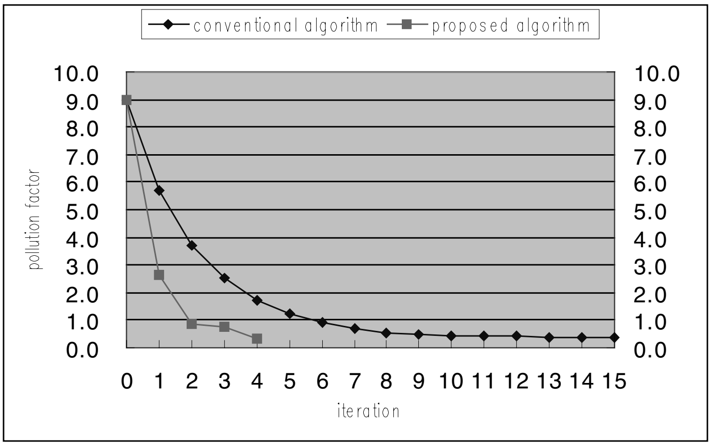


Fig. 5. The pollution factor, β_{τ} versus iteration

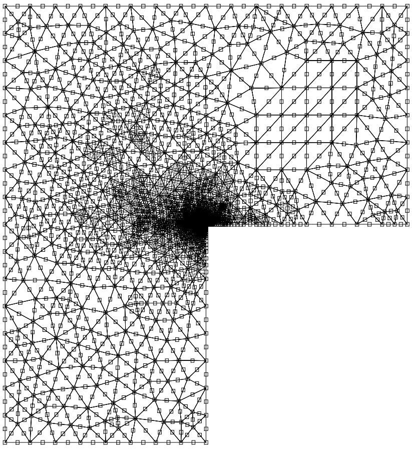


Fig. 6. The final mesh after 4 iterations by the proposed algorithm (N=3493,E=1696)

Table.2 shows the numerical results of two algorithms. From the table we knows that the proposed algorithm needs about 21.03% less time than the conventional one. It causes from more iteration of the conventional algorithm. Therefore the proposed algorithm has more effective results than the conventional one.

Table 2. Results of the model problem

	Conventional algorithm	Proposed algorithm
No. of iteration	15	4
No. of node	2597	3493
No. of element	1242	1696
Computation time	5462 sec.	4308 sec.

5 Conclusions

The pollution factor shows that the proposed algorithm converges after only 4 iterations comparing with 15 of the conventional one. The proposed r-h algorithm is easy to handle since it considers only the boundary elements. The boundary node-relocation phase is very effective for this fast convergence.

The pollution error estimates is improved from 66.72 to 2.88. Though the number of node is increased from 2597 to 3493, the total calculation time has been improved due to the decrease of the number of iteration. It is proved that the well known Delaunay method in this pollution adaptive algorithm is effective.

References

1. I. Babŭska , T. Strouboulis, A. Mathur and C.S. Upadhyay, "Pollution error in the h-version of the finite element method and the local quality of a-posteriori error estimates", Finite Elements Anal. Des.,17,273-321(1994)
2. I. Babŭska , T. Strouboulis, C.S. Upadhyay and S.K. Gangaraj, "A posteriori estimation and adaptive control of the pollution error in the h-version of the finite element method", Int. J. Numer. Method Engrg., 38, 4207-4235(1995)
3. I. Babŭska , T. Strouboulis, S.K. Gangaraj, "Practical aspects of a-posteriori estimation and adaptive control of the pollution error for reliable finite element analysis", http://yoyodyne.tamu.edu/research/pollution/index.html(1996)
4. I. Babŭska , T. Strouboulis, S.K. Gangaraj and C.S. Upadhyay, "Pollution error in the h-version of the finite element method and the local quality of the recovered derivatives", Comput. Methods Appl. Mech. Engrg.,140,1-37(1997)
5. O.C. Zienkiewicz, and J.Z.Zhu, "The Superconvergent Patch Recovery and a posteriori estimators. Part1. The recovery techniques", Int. Numer. Methods Engrg.,33,1331-1364(1992)

6. O.C. Zienkiewicz, and J.Z.Zhu, “The Superconvergent Patch Recovery and a posteriori estimators. Part2. Error estimates and adaptivity”, Int. J. Numer. Methods Engrg.,33,1365-1382(1992)
7. O.C. Zienkiewicz, and J.Z.Zhu, “The Superconvergent Patch Recovery(SPR) and adaptive finite element refinement”, Comput. Methods Appl. Mech. Engrg.,101,207-224(1992)
8. O.C. Zienkiewicz, J.Z.Zhu and J. Wu, “Superconvergent Patch Recovery techniques – Some further tests”, Comm. Numer. Methods Engrg., Vol. 9,251-258(1993)
9. B. Kaan Karamete, User manual of 2D Constrained Mesh Generation Mesh2d. http://scorec.rpi.edu/~kaan/mesh2d.tar
10. B.K. Karamete, T. Tokdemir and M. Ger, “Unstructured grid generation and a simple triangulation algorithm for arbitrary 2-D geometries using object oriented programming”, Int. J. Numer. Methods Engrg., 40,251-268(1997)
11. I. Babŭska , T. Strouboulis, and S.K. Gangaraj, “A posteriori estimation of the error in the recovered derivatives of the finite element solution”, Comput. Methods Appl. Mech. Engrg.,150,369-396(1997)