

버블패킹방법을 이용한 2차원 자동격자 생성 및 재구성 알고리즘 개발 (I)

- 선형해석 -

정순완* · 김승조†

(2000년 11월 6일 접수, 2001년 4월 9일 심사완료)

Development of Algorithm for 2-D Automatic Mesh Generation and Remeshing Technique Using Bubble Packing Method (I)

- Linear Analysis -

Soon Wan Chung and Seung Jo Kim

Key Words: Bubble Packing Method(버블패킹방법), Mesh Generation(격자생성), Remeshing(격자재구성), Refinement(세밀화), Error Estimation(오차예측), Refining Circle(세밀화 원)

Abstract

The fully automatic algorithm from initial finite element mesh generation to remeshing in two dimensional geometry is introduced using bubble packing method (BPM) for finite element analysis. BPM determines the node placement by force-balancing configuration of bubbles and the triangular meshes are made by Delaunay triangulation with advancing front concept. In BPM, we suggest two node-search algorithms and the adaptive/recursive bubble controls to search the optimal nodal position. To use the automatically generated mesh information in FEA, the new enhanced bandwidth minimization scheme with high efficiency in CPU time is developed. In the remeshing stage, the mesh refinement is incorporated by the control of bubble size using two parameters. And Superconvergent Patch Recovery (SPR) technique is used for error estimation. To verify the capability of this algorithm, we consider two elasticity problems, one is the bending problem of short cantilever beam and the tension problem of infinite plate with hole. The numerical results indicate that the algorithm by BPM is able to refine the mesh based on a posteriori error and control the mesh size easily by two parameters.

1. 서론

유한요소법의 발전으로 점차 거대하고 복잡한 구조물의 해석도 가능해지고 있다. 하지만 해석을 위한 모델링 과정도 함께 복잡해지기 때문에 자동적으로 격자를 생성하는 많은 알고리즘이 개발되어지고 있다.^(1~3) 대표적인 방법으로 공간분할법⁽¹⁾과 절점연결법^(2,3)이 있는데, 공간분할법은

영역을 원하는 격자크기에 도달할 때까지 반복적으로 나누는 방법을 말하고, 절점연결법은 먼저 절점을 배치한 다음, 이들을 Delaunay 삼각화와 같은 방법으로 연결하는 방법을 말한다. 이러한 자동 격자생성 알고리즘 개발도 현재 수치해석 분야에서 중요한 위치를 차지하고 있다. 또한 유한요소해석 결과에서 수반되는 근사 오차 (approximation error)를 줄이기 위해 유한요소 결과로 구한 오차를 바탕으로 격자를 재구성하는 기법에 대한 연구도 활발히 이루어지고 있다.^(4~6) 오차를 계산하기 위한 여러 복원기법(recovery technique)이 제안되었는데 최근에는 SPR(superconvergent patch recovery)⁽⁷⁾과 REP(recovery by

* 서울대학교 대학원 항공우주공학과

† 책임저자, 회원, 서울대학교 기계항공공학부

E-mail : sjkim@snu.ac.kr

TEL : (02)880-7388 FAX : (02)887-2662

equilibrium patch)⁽⁸⁾가 개발되었다. 격자 재구성 과정으로 오차가 큰 부분에서만 격자는 세밀화되기 때문에 최소의 자유도 증가로 정확한 해를 구할 수 있다. 따라서 *h*-convergence 측면에서 볼 때, 전체적으로 일정하게 세밀화 했을 때의 해석결과보다 오차를 바탕으로 한 적응 세밀화 결과가 수렴률이 더 높다.

본 연구에서는 앞서 언급한 절점연결법에서 최적의 절점을 얻기 위한 한 가지 방법인 버블패킹 방법⁽³⁾을 사용하여 2차원 자동격자 생성 및 재구성 알고리즘을 개발하였다. 배치된 절점들을 Delaunay 삼각화 방법으로 연결하여 삼각형 격자를 만드는 데, 전진경계기법(advancing front technique)을 사용하여 경계에서부터 안쪽으로 격자를 만들어 나갔다. 이와같이 얻은 절점들을 유한요소 해석에 사용하기 위해 효율적인 밴드폭 최소화 기법(bandwidth minimization technique)을 제안하였다. 그리고 SPR을 통해 구한 복원값으로 오차를 계산하였고 오차가 큰 영역에 세밀한 격자를 생성하기 위해 두 개의 변수를 사용하였다. 본 알고리즘의 타당성을 검증하기 위해 두 개의 선형 탄성 예제를 해석하였고, 격자 크기를 조절할 수 있는 두 변수의 변화에 따른 재구성 격자의 변화도 함께 보였다.

2. 자동격자 생성기법

본 연구에서 사용한 자동격자 생성기법은 버블패킹방법으로 최적의 절점 배치를 구하고 Delaunay 삼각화로 경계에서부터 삼각형 격자를 만들어 나간다. 그리고 생성된 격자의 품질을 높이기 위해 적응 및 반복적 버블 조절 기능도 고려하였다. 본 알고리즘에서는 Shimada와 Gossard⁽³⁾의 알고리즘을 바탕으로 하면서 새로운 절점 탐색방법과 밴드폭 최소화 기법을 제안하고 있다.

2.1 버블패킹방법

이 방법은 절점을 질량이 있는 버블의 중심으로 생각하고 버블사이의 인력과 척력을 통해 힘의 평형을 이루는 위치를 최적의 절점 위치로 고려한다. 버블에 작용하는 힘(인력/척력)은 버블 크기(지름) 함수로 계산된다. 절점 위치를 구하는 과정은 다음과 같다.

- (1) 영역의 경계를 나타내는 최소정보의 사용
- (2) 경계선 버블 채움
- (3) 면적 버블 채움
- (4) 동적 시뮬레이션

[1단계] 영역의 경계를 나타내는 최소정보 사용
자동격자 생성 알고리즘은 경계를 나타내는 최소의 정보를 통해 사용자가 원하는 격자를 생성해야 한다. 본 알고리즘에서는 경계의 꼭지점(Fig. 1), 이를 연결하는 선분 그리고 선분으로 이루어진 평면, 영역의 중심을 지정하면 된다. 각 절점은 버블의 지름 정보를 가지고 있고 위치에 따른 격자 크기 변화는 버블크기함수로 해결하였는데, 이는 격자 재구성 과정에서도 마찬가지이다. 다각형 외의 원형 경계는 원의 중심과 반지름 그리고 원주상에 놓일 절점의 개수로 지정한다. 그리고 유한요소해석에서 경계조건 등을 부가하기 위해 특정한 위치의 절점(고정점)이 필요할 수 있는데, 이러한 절점은 버블크기를 음수로 하여 구별하였다.

[2단계] 경계선 버블 채움(Fig. 2)

우선 꼭지점을 연결하는 경계선분에 대해 버블이 채워진다. 이 때 버블크기에 맞추어 차례로 밀착되게 놓여진다.

[3단계] 면적 버블 채움

1단계에서 주어진 영역의 중심과 꼭지점들을 연결한 선분이 기준이 된다. 먼저 이 선분들을 2단계에서처럼 버블로 채우고(Fig. 3), 한 선분 위의 버블과 다른 선분 위의 버블을 연결하는 선분들을 경계에서 안쪽으로 채워 나간다(Fig. 4). 이 때 빈 내부공간에 채워지는 영역은 나중에 설명할 vertical ray shooting 알고리즘으로 제거된다.

[4단계] 동적 시뮬레이션

1~3단계에서 구한 버블들은 서로 겹치거나 간격이 벌어질 수 있으므로 이들 사이의 인력과 척력을 가정하여 힘의 평형을 이루는 최적의 버블 위치를 찾는 과정이 본 단계에서 수행된다. 작용하는 힘은 버블중심간의 거리에 대한 3차함수로 가정하였고,⁽³⁾ 경계선 상의 절점은 경계선분 위에서만 이동해야 하므로 이 절점에 작용하는 하중을 구한다음, 경계선분과 나란한 하중성분만을 구하여 절점에 작용하였다. 식 (1)과 같은 *i*-번째 버블의 운동방정식을 4차 Runge-Kutta법으로 적분하였다.

$$m_i \ddot{x}_i + c_i \dot{x}_i = f_i \quad (1)$$

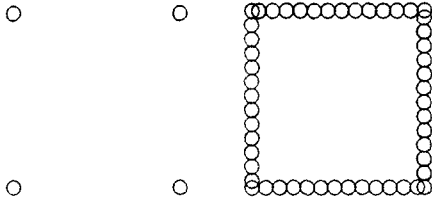


Fig. 1 Nodes given in input data

Fig. 2 Boundary line bubble packing

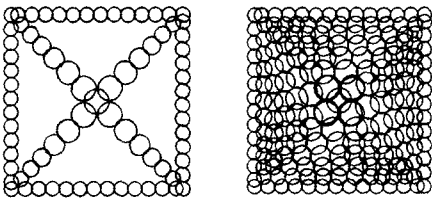


Fig. 3 Surface bubble packing 1

Fig. 4 Surface bubble packing 2

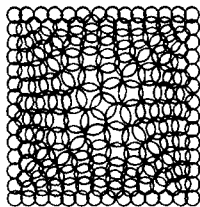


Fig. 5 Bubble distribution after dynamic simulation

상수 m_i, c_i 는 변위가 점차 감소되도록 정하였고 수렴속도가 느린 경우 최대 100회까지만 반복 계산을 수행하였다. 그리고 버블간의 거리가 버블크기의 평균의 5분의 1 내지 2분의 1(사용자에 의해 결정)보다 작은 경우 둘 중 하나를 제거하였다. 그리고 반복계산 과정 중에 영역 밖으로 나가는 버블도 제거하였다. Fig. 5는 이상으로 얻은 최종 절점배치를 도시하고 있다.

2.2 Delaunay 삼각화

버블패킹방법으로 구한 절점들의 배치는 Voronoi 다각형과 유사하므로 Delaunay 삼각화를 이용하여 정삼각형에 가까운 삼각형을 만들었다. 이를 위해 최대내각 성질⁽⁹⁾을 경계선분에서부터 사용하였다(Fig. 6). 경계선분에서 구한 나머지 두 변을 기준으로 다시 최대각을 이루는 절점을 찾으므로, 전진경계기법처럼 경계에서 안쪽으로 삼각형이 형성된다. 절점 탐색시, 각 절점에 연결되는 내각의 총합을 계산하여 360도에 도달한 절점(경계선 상의 절점은 180도)은 제외하였다.

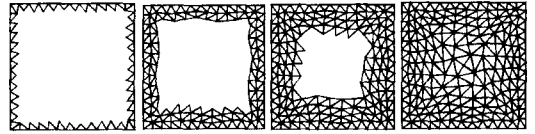


Fig. 6 Delaunay triangulation with advancing front concept

2.3 절점 위치 탐색을 위한 알고리즘

삼각화 과정을 하기 전에 절점들이 영역내에만 존재함을 확인하는 과정이 필요하다. 따라서 2.1절의 3번째 단계에서 빈 내부영역에 존재하는 버블이나 4번째 단계에서 영역 밖에 존재하는 버블을 제거하는 과정을 고려하였는데, 이 때 사용한 방법은 vertical ray shooting 알고리즘이다.⁽¹⁰⁾ 절점에서 수직으로 내린 직선이 경계선과 교차하는 점의 개수를 구하여, 홀수이면 그 절점은 영역내에 존재하고 짝수이면 영역밖에 존재하게 된다. 2.2절의 Delaunay 삼각화 과정에서 최대각은 이루지만 부적합 영역에 절점이 위치할 수 있는데, 이를 제거하기 위해 새로운 절점 탐색 알고리즘(Node Search Algorithm-1)을 추가하였다. Fig. 7의 점c는 선분ab에 대해 최대각을 이루지만 선분ab에 이미 연결된 점e와 같은 방향에 있으므로 연결될 수 없다. 따라서 NSA-1은 선분ab와 연결하는 점으로서 점c가 아닌 점d를 선택한다. 또한 경계선분과 같이 선분을 기준으로 한쪽 방향에 있는 절점만을 탐색하는 경우를 위해 NSA-2를 제안하였다. Fig. 8과 같이 반시계 방향으로 정의된 경계선에 대해 적합 영역과 부적합 영역은 바깥쪽 경계선과 안쪽 경계선에 따라 달라진다. 방향벡터(v_x, v_y)를 반시계 방향으로 90도 회전한 수직벡터는 ($-v_y, v_x$)이므로, 경계선분의 직선의 방정식은 다음과 같다.

$$-v_yx + v_xy + c_1 = 0 \tag{2}$$

여기서 c_1 는 경계선 상의 두 절점 중의 임의의 한 점을 지나도록 하는 상수이다. 한 점(x_1, y_1)이 방향벡터의 왼쪽에 있으면 식 (3)을 만족하고, 오른쪽에 있으면 식 (4)를 만족한다.

$$-v_yx_1 + v_xy_1 + c > 0 \tag{3}$$

$$-v_yx_1 + v_xy_1 + c < 0 \tag{4}$$

바깥쪽 경계선의 적합 영역은 왼쪽이므로 식 (3)을 만족하는 절점만을 고려하고, 반대로 안쪽

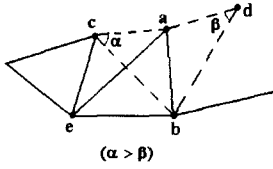


Fig. 7 Sketch for NSA-1

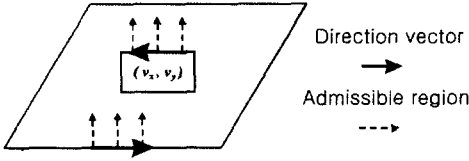


Fig. 8 Sketch for NSA-2

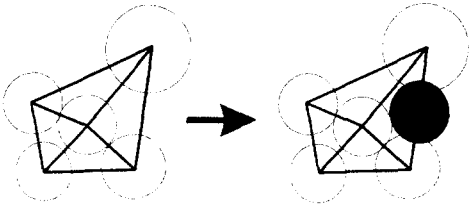


Fig. 9 Adaptive bubble control: bubble addition

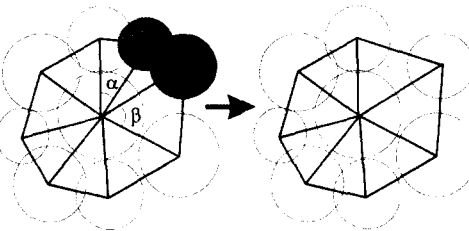


Fig. 10 Adaptive bubble control: bubble deletion

경계선의 적합 영역은 오른쪽이므로 식 (4)를 만족하는 절점만을 고려한다.

2.4 적응 및 반복적 버블 조절

삼각형 격자의 품질(quality)을 보다 높이기 위해 다음과 같은 적응 조절법을 사용하였다. 우선 동적 시뮬레이션 과정에서 경계선분과 연결된 삼각형의 최대내각이 150도 이상이 될 때 해당 버블을 제거하였다. 그리고 한 내부 절점에 연결된 격자의 수를 계산하고 그 평균값이 5.9 이하이거나 6.1 이상이면 버블 추가 또는 제거가 수행되었다. Fig. 9와 같이 한 점에 연결된 격자의 수가 5보다 작으면 가장 내각이 큰 격자의 대변 중점에 버블을 추가한다. 그리고 Fig. 10과 같이 연결

된 격자의 수가 7보다 크면 가장 내각이 작은 격자와 연결된 두 버블 중에 이웃한 격자의 내각이 작은 쪽(α)의 버블이 제거된다. 이와같이 버블의 위치가 재조정되면 다시 2.1절의 [4단계]인 동적 시뮬레이션 과정으로 돌아간다. 이 때 다시 추가/제거 과정이 필요한 경우가 생길 수 있는데 실제 프로그램에서는 이 과정을 5번까지만 반복하였고, 이러한 점은 차후 연구에서 개선할 계획이다.

2.5 밴드폭 최소화 기법

자동 격자 생성기법으로 얻은 절점번호는 불규칙하게 분포되어 있으므로 유한요소해석시 전역강성행렬의 밴드폭(bandwidth)이 매우 커질 수 있다. 따라서 본 연구에서는 Puttonen⁽¹¹⁾의 밴드폭 감소 알고리즘을 바탕으로 한 수정된 알고리즘을 개발하였다. Puttonen 알고리즘의 기본 아이디어는 한 요소를 구성하는 절점들의 번호를 나타내는 절점 연결 행렬의 행과 열을 가능한 한 대각 성분 쪽으로 이동시키는 것이다. 그런데 이 행렬이 정방행렬(Square Connectivity Matrix, SCM)이면 요구되는 메모리와 계산시간이 증가한다. 만약 0이 아닌 성분만을 저장하는 축약된 행렬(Condensed Connectivity Matrix, CCM)을 사용한다면 메모리와 계산시간에서 상당한 이득을 얻을 수 있을 것이다. 예를 들어, 10000개의 절점을 가진 모델에서 SCM 사용시의 요구 메모리는 100Mbyte이고(연결여부만을 판단하기 때문에 1 Byte 문자형 변수를 사용하고 SCM의 성분개수는 10^8 개이므로 필요한 메모리 크기는 100Mbyte이다), CCM 사용시의 요구 메모리는 기껏해야 100Kbyte 정도이다. 참고문헌 (11)에서는 CCM의 언급이 있었지만 구체적인 설명이 없으므로 여기서 사용한 방법을 간단히 소개하고자 한다. Puttonen의 알고리즘은 다른 절점과의 연결정도가 낮은 행(열)을 우선적으로 대각원소쪽으로 이동시키는 행(열)들 사이의 교환으로 이루어지므로, CCM에서 생각할 수 있는 경우의 수는 교환하는 행(열)에 절점번호가 있느냐 없느냐에 따라 다음 4가지로 나눌 수 있다.

Fig. 11에서 임의의 k -번째 행에서 i -번째 열과 j -번째 열이 서로 바뀐다고 가정해 보자. Case 1과 Case 2에서는 아무 변화가 없고, Case 3과 Case 4에서의 변화는 다음과 같다.

Case 3 : CCM(k,1)=i , CCM(k,2)=i+1 , CCM(k,3)=i+3

→ CCM(k,1)=i+1 , CCM(k,2)=i+3 , CCM(k,3)=j

Case 4 : CCM(k,1)=i+1 , CCM(k,2)=i+2 , CCM(k,3)=j

→ CCM(k,1)=i , CCM(k,2)=i+1 , CCM(k,3)=i+2

이 CCM의 효율을 보이기 위해, Fig. 12의 세 가지 모델에 대해 CPU 시간을 SCM과 비교하였다. Table 1에서 보는 바와 같이 절점수가 증가할수록 CCM을 사용하면 SCM을 사용할 때보다 계산시간이 급격히 줄어든다. 참고로 Alpha-533MHz를 장착한 컴퓨터를 사용하였다. 밴드폭 최소화 과정으로 절점번호가 다시 지정되므로 이 과정 전후의 절점번호 변화를 저장하는 별도의 기억장소가 필요하다. 이에 대한 이유는 다음 장의 격자 재구성 알고리즘에서 설명할 것이다.

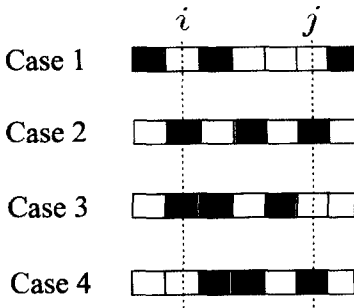


Fig. 11 4-cases of row/column exchanges for CCM

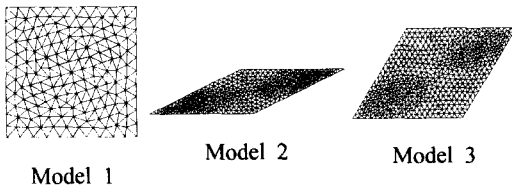


Fig. 12 Models used for comparison of CPU time

Table 1 Comparison of CPU time

	절점수	요소수	SCM (sec)	CCM (sec)	밴드폭 변화
Model 1	173	304	0.141	0.068	129→22
Model 2	399	716	0.640	0.139	273→14
Model 3	593	1104	3.080	0.425	534→52

3. 격자 재구성 알고리즘

격자 재구성 알고리즘은 오차나 요소의 찌그러짐이 큰 부분을 중심으로 보다 세밀한 요소를 많이 배치하여 수렴성과 정확성을 높이는 것을 그 목적으로 하고 있다. 일반적으로 탄성해석에서 일정하게 격자를 세밀화하는 것(uniform refinement)보다 정해로의 높은 수렴률을 나타낸다. 본 연구에서는 사후(a posteriori) 오차를 계산하여 세밀화할 영역과 그 영역에서의 격자 크기를 결정하였다. 버블패킹방법에서는 버블크기가 곧 격자 크기이므로 버블크기함수를 조정함으로써 쉽게 새로운 격자크기를 결정할 수 있다. 그리고 본 알고리즘은 절점배치나 삼각화 과정이 경계에서 시작되고 세밀화하는 과정에서 경계선분에 절점이 추가될 수 있기 때문에 경계정보의 전달이 중요하다. 앞의 2.5절에서 언급한 추가 기억저장 장소는 실제 해석에 사용된 절점번호 가운데 경계 절점을 구별하여 새로운 절점 추가로 변경되는 경계정보를 수정하기 위해 필요하다. 경계정보의 전달과 버블크기함수가 다시 구해지면 자동격자 생성단계로 넘어가서 격자를 생성하게 된다.

3.1 복원기법과 오차예측

복원기법으로 구한 값을 이용한 오차는 정해를 이용한 오차에 수렴한다는 사실이 알려져 있으므로, (12) 일반적으로 이러한 근사 오차가 격자 세밀화의 기준이 될 수 있다. 본 연구에서는 복원기법으로 SPR(7)을 사용하였는데 그 과정을 간단히 설명하면 다음과 같다. 1차 삼각형 요소의 경우, 변형률의 선형 전개식은 식 (5)와 같다.

$$\epsilon_i^* = P a \tag{5}$$

여기서, $P = [1, x, y]$, $a = [a_1, a_2, a_3]^T$ 이다.

미지수 a 는 ϵ_i^* 과 유한요소해($\hat{\epsilon}$) 사이의 차를 최소화함으로써 다음과 같이 구해진다.

$$a = A^{-1} b \tag{6}$$

여기서, $A = \sum_i P^T(x_i, y_i) P(x_i, y_i)$, $b = \sum_i P^T(x_i, y_i) \hat{\epsilon}(x_i, y_i)$ 이고 n 은 패치(patch)를 구성하는 요소의 적분점 개수이다. 미지수가 3개이므로 적분점은 3개 이상이 되어야 한다. 그리고 경계선 부근에서는 복원값의 급격한 증가를 막기 위해 4개 이상의 요소를 선택하였다. 만약 경계절점의 패치를 구성하는 요소가 4보다 작으면 패치상의 임의의 두

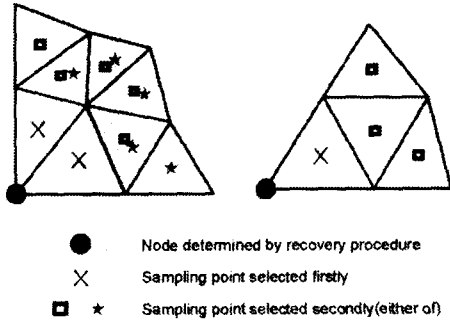


Fig. 13 Recovery at boundary nodes

개의 절점과 연결된 이웃한 요소들 패치에 포함시켰다(Fig. 13). a 를 계산한 다음, 절점에서의 복원값 $\bar{\epsilon}^*$ 은 식 (5)에 절점좌표를 대입하여 구한다. 따라서 요소내의 변형률은 $\bar{\epsilon}^*$ 을 보간하여 얻어진다. 여기서 보간함수(interpolation function)는 변위의 보간에 사용된 것과 같은 것을 사용한다.

$$\bar{\epsilon}^* = N \bar{\epsilon} \quad (7)$$

복원값을 이용한 변형률 오차는 식 (8)과 같고,

$$e_\epsilon = \bar{\epsilon}^* - \hat{\epsilon} \quad (8)$$

변형률 오차의 L_2 norm($\|e_\epsilon\|$)과 예상 상대오차(η^*)는 다음 식으로 표현할 수 있다.

$$\|e_\epsilon\| = \sqrt{\sum_1^m \|e_{\epsilon i}\|^2} = \sqrt{\sum_1^m \int_{\Omega_i} (\bar{\epsilon}^* - \hat{\epsilon})_i^2 d\Omega} \quad (9)$$

$$\eta^* = \sqrt{\frac{\|e_\epsilon\|^2}{\|\bar{\epsilon}^*\|^2 + \|e_\epsilon\|^2}} \quad (10)$$

여기서 m 은 전체 요소수를 의미한다. 만약 오차가 전체에 걸쳐 일정하게 분포하고 있다고 가정하면, 주어진 최대허용 상대오차 η_{\max}^* 에 대해 임의의 i -번째 요소에서의 최대허용오차는 다음 식과 같이 표현될 수 있다.

$$\|e_{\epsilon i}\|_{i, \max} = \frac{\eta_{\max}^*}{\sqrt{m}} (\|\bar{\epsilon}^*\|^2 + \|e_\epsilon\|^2)^{1/2} \quad (11)$$

3.2 세밀화 기준

식 (11)의 요소당 최대허용오차보다 큰 오차를 갖는 요소들 중심으로 격자가 세밀화된다. 만약 η_{\max}^* 가 크면 오차가 큰 부분만이 세밀화되고, 작으면 세밀화되는 영역이 넓어진다. 따라서 η_{\max}^* 는 오차분포에 따라 선택될 수 있으며, 이 η_{\max}^*

Refining circle

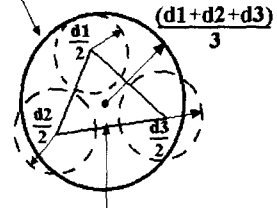


Fig. 14 Refining circle defined by bubble size

의 변화에 따른 격자분포의 변화를 뒤의 수치예제에서 확인할 것이다. 본 연구에서는 세밀화되는 영역을 표시하기 위해 세밀화 원(refining circle)을 도입하였다. 허용오차 이상의 오차를 갖는 요소의 무게중심을 원의 중심으로 하고, 세 꼭지점의 버블지름의 평균을 원의 반지름으로 정의하였다.(Fig. 14) 그리고 이 원의 내부에 속하는 새로운 점들의 버블크기는 식 (12)로 계산된다.

$$d_{new} = \left(\frac{\|e_\epsilon\|_{i, \max}}{\|e_\epsilon\|_i} \right)^q d_{ave} \quad (12)$$

여기서 d_{ave} 는 세 꼭지점에서의 버블지름의 평균이고, q 는 d_{new} 를 조절하는 인자이다. 인자 q 가 1보다 크면 버블크기의 감소율 $\frac{d_{ave} - d_{new}}{d_{ave}}$ 은 커지고, 1보다 작으면 감소율도 작아진다. 즉 η_{\max}^* 로 세밀화 영역을 선택하고 그 안의 버블크기는 인자 q 로 조절 가능하다. 따라서 버블패킹방법을 이용한 격자 재구성 알고리즘에서는 새로운 경계정보와 더불어 η_{\max}^* 와 q 를 이용한 새로운 버블크기함수가 핵심을 이룬다.

3.3 새로운 격자로의 변수 전달

격자 재구성 단계에서 고려해야 할 사항중의 하나는 이전 격자의 절점 정보나 상태변수를 새로운 격자에 정확히 전달하는 것이다. 탄성해석의 경우, 격자가 새로 생성되면 변형 전 상태에서 다시 힘을 가하기 때문에 경계조건과 하중조건이 작용하는 점들만을 고려하면 된다. 본 연구에서 사용한 알고리즘은 경계절점에서 시작하고 오차가 큰 요소가 경계선에 인접해 있는 경우 절점이 추가되어 경계 정보가 변할 수 있기 때문에 특히 경계절점의 정확한 전달이 필요하다. 그리

고 새로 추가된 절점은 이웃한 경계절점의 경계 조건과 하중조건을 따른다. 이전 격자에서의 경계절점 정보를 이용하기 위해 새 격자와 이전 격자의 절점번호 관계를 저장해야 하는데, 밴드폭 최소화 과정에 의한 절점번호 변화도 고려하기 위해 추가 기억장소가 필요하다.

4. 수치해석

지금까지 설명한 알고리즘을 바탕으로 두 개의 선형탄성문제를 해석하였다. 하나는 짧은 외팔보에 굽힘하중이 가해지는 문제이고, 다른 하나는 원형 구멍이 있는 평판에 인장하중이 작용하는 문제이다. 사용한 탄성계수와 포아송 비는 각각 $E=10^5$, $\nu=0.3$ 이다. 오차는 등가 변형률(equivalent strain)를 사용하여 예측하였다. 선형탄성해석이므로 격자 재구성 단계를 거치면 상태변수는 초기화되고 변형 전 형상에서 하중을 작용시켰다. 등가 변형률의 L_2 norm의 수렴률을 파악하기 위해 식 (13)을 바탕으로 정해를 구하였다.

$$\|e\| \leq Ch^{\min(p, \lambda)} \approx CN^{-0.5 \min(p, \lambda)} \quad (13)$$

즉, h -수렴 측면에서 일정하게 세밀화한 격자의 해석결과와 외삽(extrapolation)을 통하여 정해를 예측할 수 있다.⁽¹²⁾ 식 (13)에서 h 는 요소크기, N 은 자유도 수, C 는 문제에 따라 바뀌는 양수, p 는 형상함수의 차수 그리고 λ 는 특이성 정도를 나타낸다. 따라서 등가 변형률의 L_2 norm의 상대오차(η)는 다음 식과 같다.

$$\eta = \frac{\|e_e\|}{\|\epsilon\|} = \left(\frac{\|\epsilon\|^2 - \|\hat{\epsilon}\|^2}{\|\epsilon\|^2} \right)^{1/2} \quad (14)$$

4.1 짧은 외팔보의 굽힘문제

평면 응력 상태로 가정한 모델의 형상을 Fig. 15에 나타내었다. 분포하중 P 는 1000, 길이는 $L=15$, $c=5$ 를 사용하였다. 우선, 균일한 버블크기 2, 1, 0.5, 0.25를 사용하여 4가지의 격자를 만들었고, (Fig. 16) 이들의 절점수 및 요소수를 Table 2에 정리하였다. 각 격자에 대한 요소의 오차 ($\|e_e\|_i$) 분포도를 Fig. 17에 나타내었는데, 요소가 작아질수록 오차도 줄어들음을 볼 수 있다. Fig. 15의 (A),(B) 지점에서 오차가 크기 때문에 이 부분에 보다 많은 요소가 필요함을 알 수 있다.

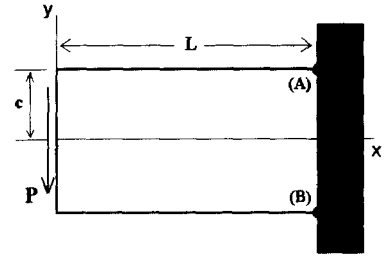


Fig. 15 Short cantilever beam under bending

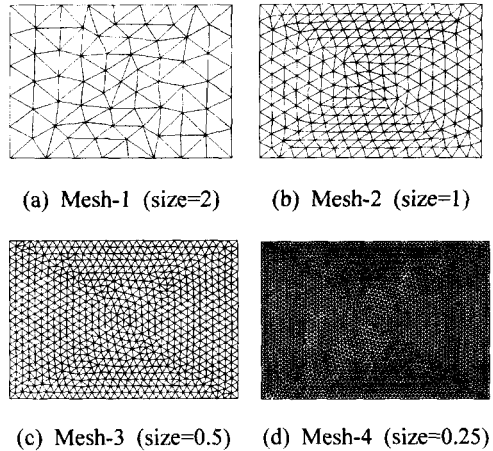


Fig. 16 Meshes generated by uniform refinement in bending problem

Table 2 Numbers of nodes and elements of uniform refinement in bending problem

	Mesh-1	Mesh-2	Mesh-3	Mesh-4
절점수	75	263	975	3749
요소수	122	474	1848	7296

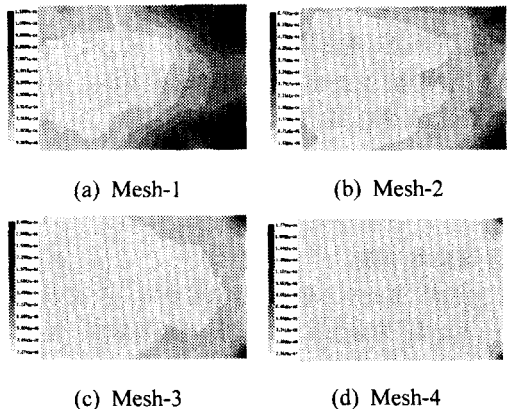


Fig. 17 Error ($\|e_e\|_i$) distribution of uniformly refined meshes in bending problem

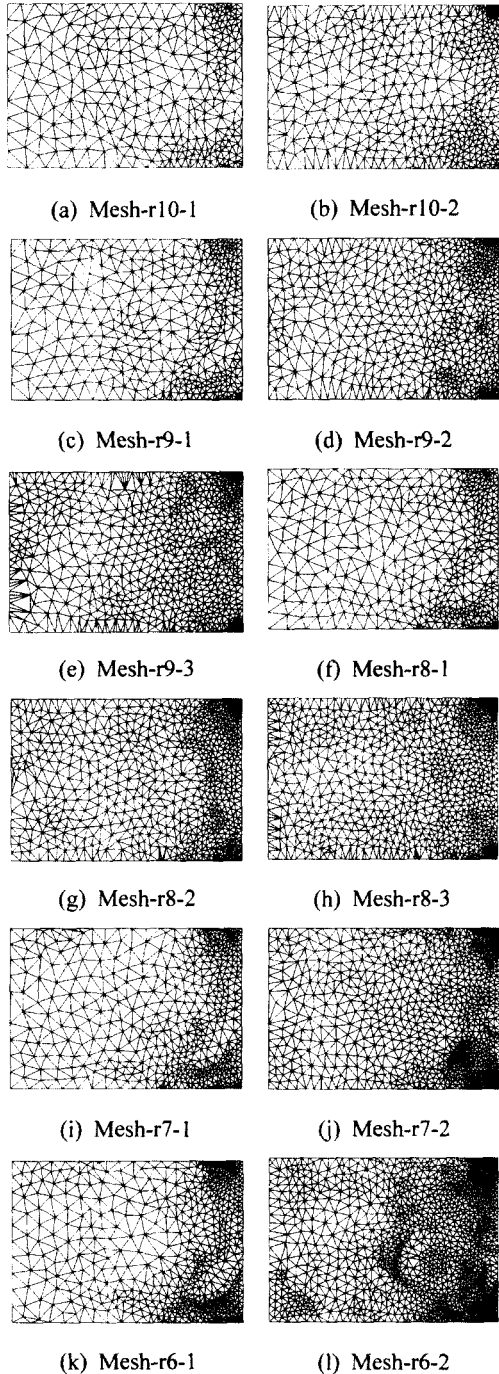


Fig. 18 Meshes generated by adaptive refinement in bending problem

적용 세밀화 결과를 얻기 위해, 5개의 η_{max}^* (10%~6%)를 사용하였다. Mesh-1을 초기격자로

Table 3 Numbers of nodes and elements of adaptive refinement in bending problem $q=1$ (M-r(N1)-(N2) , M : Mesh, N1 : η_{max}^* (%), N2 : remeshing order)

모델	절점수	요소수	모델	절점수	요소수
M-r10-1	338	598	M-r10-2	533	953
M-r10-3	655	1161	M-r10-4	755	1321
M-r9-1	412	735	M-r9-2	650	1176
M-r9-3	795	1429	M-r8-1	465	830
M-r8-2	788	1440	M-r8-3	939	1703
M-r7-1	551	994	M-r7-2	989	1830
M-r6-1	667	1215	M-r6-2	1408	2651

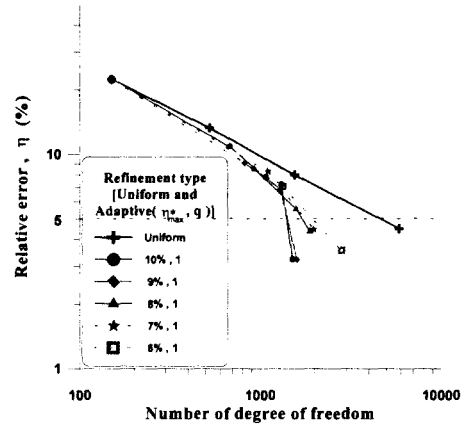


Fig. 19 Convergence rates of relative error(%) in bending problem

사용하였고 상대오차(η)가 5% 미만이 될 때까지 격자 재구성을 반복하였다. q 는 1을 사용하였고 세밀화된 격자들을 Fig. 18에 도시하였으며, 이들의 절점수와 요소수를 Table 3에 정리하였다. 예상한 바와 같이, (A),(B)지점 부근에서 세밀한 격자가 생성되었고, 재구성 횟수가 증가할수록 그 부근의 격자크기는 감소하였다. η_{max}^* 가 작을수록 보다 많은 요소가 세밀하게 되었고, 한번의 재구성 과정에 생성되는 격자수도 증가하였다.

일정하게 격자크기를 줄였을 때와 본 연구의 격자 재구성 알고리즘을 사용했을 때의 수렴률을 Fig. 19에서 비교하였다. η_{max}^* 값에 관계없이 적응 세밀화 결과가 수렴률 측면에서 확실히 우수함을 볼 수 있다. 이 결과로부터 본 격자 재구성 알고리즘은 오차가 큰 영역에서 격자를 효율적으로 세밀하게 만들고 높은 수렴률을 유지하면서 η_{max}^* 로 격자크기를 조절할 수 있음을 알 수 있었다.

4.2 원형 구멍이 있는 무한평판의 인장문제

Fig. 20의 왼쪽 그림과 같이, 원형 구멍이 있는 무한평판에 인장하중이 작용하는 문제를 해석하였다. 구멍의 반지름 a 가 1이고 분포하중 S 도 1이라고 가정할 때, 평면 변형률 상태에서의 응력 분포는 다음과 같이 탄성이론으로 구할 수 있다.

$$\sigma_{xx} = 1 - \frac{1}{r^2} \left(\frac{3}{2} \cos 2\theta + \cos 4\theta \right) + \frac{3}{2r^4} \cos 4\theta$$

$$\sigma_{yy} = -\frac{1}{r^2} \left(\frac{1}{2} \cos 2\theta - \cos 4\theta \right) - \frac{3}{2r^4} \cos 4\theta$$

$$\sigma_{xy} = -\frac{1}{r^2} \left(\frac{1}{2} \sin 2\theta + \sin 4\theta \right) + \frac{3}{2r^4} \sin 4\theta$$

해석에서는 Fig. 20의 오른쪽 그림 부분 ($l=4$)만을 사용하였고 위쪽과 오른쪽 변에 작용하는 하중은 위의 응력분포를 적분하여 계산하였다. 먼저, 균일한 버블크기 0.3, 0.2, 0.1을 사용하여 격자를 생성하였고(Fig. 21) 이에 대한 절점수와 요소수는 Table 4에 나타내었다.

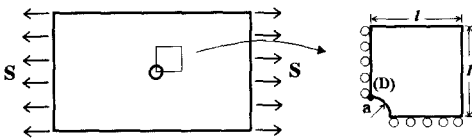
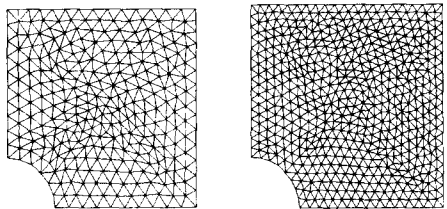
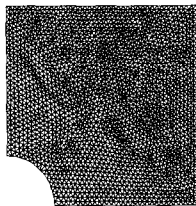


Fig. 20 Infinite plate with hole under uniform tension



(a) Mesh-1(size=0.3) (b) Mesh-2(size=0.2)



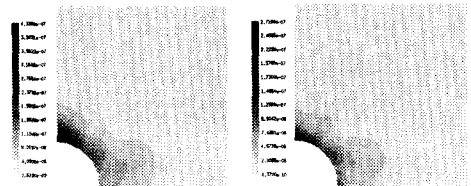
(c) Mesh-3(size=0.1)

Fig. 21 Meshes generated by uniform refinement in tension problem

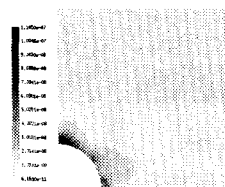
이 격자를 사용했을 때의 오차분포를 Fig. 22에 도시하였는데, Fig. 20의 (D)지점에서 오차가 큼을 볼 수 있다. 굽힘문제에서와 같이, Mesh-1을 초기 격자로 삼고, 5개의 η_{max}^* 를 사용하여 적응 세밀화를 수행하였다. 상대오차가 5% 이하가 될 때까지 재구성을 진행하였고, 버블크기 조절인자 q 의 효과를 보기 위해 1과 1.5 두 값에 대하여 계산하였다. 재구성 과정으로 얻은 격자들의 형상을 Fig. 23에 나타내었는데, η_{max}^* 와 q 에 상관없이 (D)지점을 비롯한 원주부근에서 격자가 세밀화됨을 볼 수 있다. 그리고 q 가 크면 세밀화시 버블크기의 감소율이 커지므로 재구성 과정으로 생기는 격자의 크기가 작음을 확인하였다. 그리고 두 개의 q 에 대한 상대오차의 수렴률을 Fig. 24과 Fig. 25에 각각 도시하였다. 앞선 예제처럼, 적응 세밀화에서 높은 수렴률을 보이고 있는데 $q=1$ 일 때는 η_{max}^* 에 따른 수렴률의 차이가 거의 없었지만, $q=1.5$ 일 때는 η_{max}^* 가 줄어들며 따라 수렴률이 uniform refinement의 결과에 접근함을 볼 수 있다. 이것은 $q=1.5$ 의 경우, 오차가 크지 않은 영역에서도 작은 격자크기가 얻어지므로 증

Table 4 Numbers of nodes and elements of uniform refinement in tension problem

	Mesh-1	Mesh-2	Mesh-3
절점수	292	601	2304
요소수	528	1122	4450



(a) Mesh-1 (b) Mesh-2



(c) Mesh-3

Fig. 22 Error($\|e_\epsilon\|_i$) distribution of uniformly refined meshes in tension problem

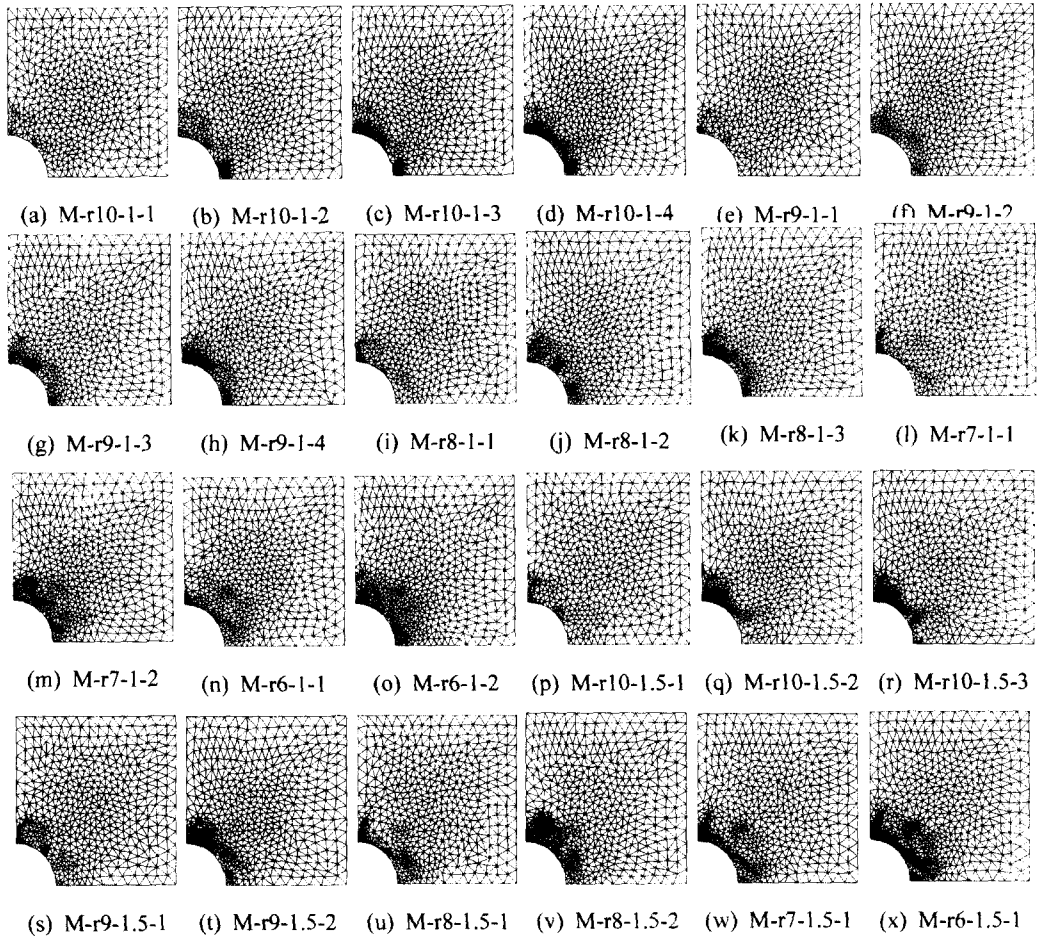


Fig. 23 Meshes generated by adaptive refinement in tension problem

가한 자유도에 비해 오차감소가 두드러지지 않기 때문이다. 이상의 해석으로부터, 본 알고리즘은 오차가 큰 영역에서 세밀한 격자를 잘 생성함과 동시에, 두 개의 변수(η_{max}^* , q)로써 세밀화 영역과 격자 크기를 쉽게 조절할 수 있음을 확인하였다.

5. 결론

본 논문에서는 버블패킹방법을 이용하여 2차원 자동격자 생성 및 격자 재구성 알고리즘을 개발하였다. 버블패킹방법으로 절점의 위치를 결정하고, 전진경계기법을 이용한 Delaunay 삼각화로 격자를 생성하였다. 그리고 자동격자로 얻은 절점/요소 정보를 유한요소 해석에 적용하기 위해 효율적인 밴드폭 최소화 기법도 개발하였다. 격자

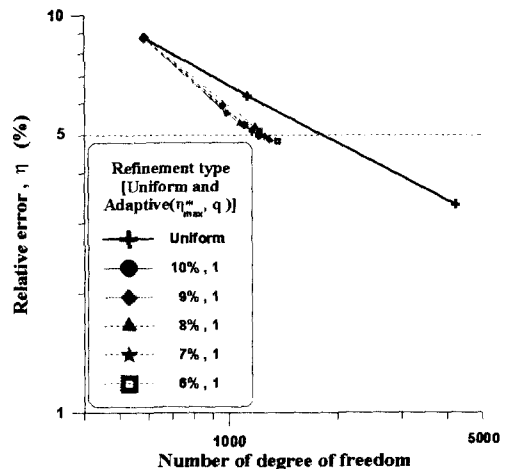


Fig. 24 Convergence rates of relative error(%) in tension problem ($q=1$)

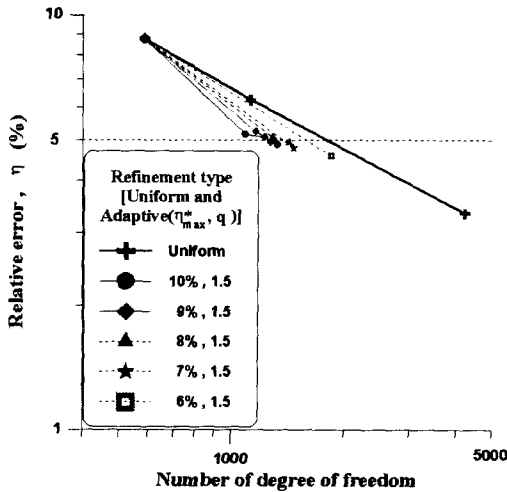


Fig. 25 Convergence rates of relative error(%) in tension problem ($q=1.5$)

재구성 과정에서는 세밀화 원을 도입하여 오차가 큰 영역을 표시하고 이 안에 놓이는 버블들의 크기를 두 개의 변수로 쉽게 조절하였다. 본 알고리즘은 경계절점과 버블크기함수만 새로 지정되면 격자 재구성을 쉽게 할 수 있는데, 그 성능을 두 개의 선형탄성예제에서 확인하였다. 오차가 큰 특이점 부근에서 세밀한 격자를 얻을 수 있었고 높은 수렴률을 나타냄을 확인하였다.

후 기

본 연구는 과학기술부의 국가지정연구실 사업(과제번호 00-N-NL-01-C-026)에 의해 지원되었으며, 이에 관계자 여러분께 감사 드립니다.

참고문헌

(1) Peggy L. Baehmann, Scott L. Wittchen, Mark S. Shephard, Kurt R. Grice and Mark A. Yerry, 1987, "Robust, Geometrically Based, Automatic Two- Dimensional Mesh Generation," *Int. J. Numer. Methods Eng.*, Vol. 24, pp. 1043~1078.
 (2) Lee, C.K. and Hobbs, R.E., 1999, "Automatic Adaptive Finite Element Mesh Generation over Arbitrary Two- Dimensional Domain Using

Advancing Front Technique," *Computers and Structures*, Vol. 71, pp. 9~34.
 (3) Kenji Shimada and David C. Gossard, 1998, "Automatic Triangular Mesh Generation of Trimmed Parametric Surfaces for Finite Element Analysis," *Computer Aided Geometric Design*, Vol. 15, pp. 199~222.
 (4) 정현석, 김용환, 1996, "Delaunay 삼각화기법을 이용한 유한요소망의 자동생성과 격자 재구성에의 응용," *대한기계학회논문집(A)*, 제20권, 제2호, pp. 553~563.
 (5) 채수원, 1994, "h-분할법에 의한 사각형 유한요소망의 적응적 구성," *대한기계학회 논문집*, 제18권, 제11호, pp. 2932~2943.
 (6) Paulino, G.H., Menezes, I.F.M., Cavalcante Neto, J.B. and Martha, L.F., 1999, "A Methodology for Adaptive Finite Element Analysis : Towards an Integrated Computational Environment," *Computational Mechanics*, Vol. 23, pp. 361~388.
 (7) Zienkiewicz, O.C. and Zhu, J.Z., 1992, "The Superconvergent Patch Recovery and a Posteriori Error Estimates. Part I : The Recovery Technique," *Int. J. Numer. Methods Eng.*, Vol. 33, pp. 1331~1364.
 (8) Boroomand, B. and Zienkiewicz, O.C., 1997, "Recovery by Equilibrium in Patches," *Int. J. Numer. Methods Eng.*, Vol. 40, pp. 137~164.
 (9) Chongjiang Du, 1998, "A Note on Finding Nearest Neighbours and Constructing Delaunay Triangulation in the Plane," *Communications in Numer. Methods Eng.*, Vol. 14, pp. 871~877.
 (10) Ketan Mulmuley, 1994, *Computational Geometry : An Introduction Through Randomized Algorithms*, Prentice-Hall, pp. 288~289.
 (11) Jari Puttonen, 1983, "Simple and Effective Bandwidth Reduction Algorithm," *Int. J. Numer. Methods Eng.*, Vol. 19, pp. 1139~1152.
 (12) Zienkiewicz, O.C., Zhu, J.Z. and Gong, N.G., 1989, "Effective and Practical h-p-Version Adaptive Analysis Procedure for the Finite Element Method," *Int. J. Numer. Methods Eng.*, Vol. 28, pp. 879~891.