

2차원 자동요소분할 프로그램 개발에 관한 연구 (A Study on the two-dimensal Automatic Mesh Generation Programming)

조 명 철, 유 형 선
(Myung-Cheol Joe, Hyeong-Seon Yoo)

인하대학교 공과대학 자동화 공학과
(Dept. of Industrial Automation , Inha University)

Abstract

This paper is concerned with the program of the automatic mesh generation for 2-dimensional domain which contains the curved boundaries and holes.

This program treats a new vertical - line drawing method. This method starts with 4-subdivisions of problem domain and the classification of the cross points of grid lines and boundaries. And the new node is generated by the vertical line to the line connecting the two intersections of a boundary and two grid lines in general.

The boundaries are piecewise-curves composed of lines, circles, arcs, and free curves. The free curves are generated by B-Spline form.

Although there were some bad elements for the complex boundary, it was possible to obtain the acceptable elements for the given boundaries.

The results of automatic mesh generation can be verified directly by drawing on the computer monitor in excuting the program. And it is possible to add the processes - that is, editing , hard copying, etc - using the script file in Auto-CAD.

1. 서 론

미분방정식의 해를 수치해석적으로 구하는 유한요소법에서, 필요한 데이터들 직접하게 입력시키는 전처리부(Preprocessor)에 관한 연구로서, 주어진 영역의 경계선이 곡선 및 영역내의 빈 영역(Hole)이 포함된 경우에 요소를 자동으로 분할하는 방법(Automatic mesh generation)에 관한 연구이다.^{1, 2, 3}

본 논문에서의 요소 분할 방법은 고정된 격자를 주고, 그 격자와 주어진 영역의 경계곡선이 만나는 점들을 검색하여, 각 격자를 -IN(주어진 영역 내부), -OUT(주어진 영역 외부), 와 -BOU(경계와 주어진 격자가 만남)로 분류하여, -OUT 인 격자는 제거하고, -IN 인 격자는 유지하며, -BOU 인 격자는 격자와 경계가 만나는 경우에 따라 처리하려고 하는데, 가장 일반적으로인 경우에 격자와 경계가 만나는 두점사이를 통하는 직선에 -BOU(경계와 격자가 만남)인 격자내의 절점에서 수선을 내려 만나는 절점을 그 격자 요소의 새로운 절점으로 택하여, 4절점 사각요소 및 8절점 사각요소를 생성하겠끔 하려고 한다.

또한, 되도록 최소한의 영역의 경계 DATA들 주어 경계곡선을 생성하였고, 특히 B-spline 곡선식을 이용하여 최소의 입력으로 다양한 형태의 경계곡선을 생성, 처리하고자 하는데 그 목적을 둔다.

2. 요소분할 방법

본 논문의 요소분할 방법은 다음과 같다.

우선 초기에 $257((2^8)^2 + 1)$ 개의 격자를 주고, 그 격자와 주어진 경계곡선과 어떤 격자에서 만나는 지를 판하여, 각 격자가 외부인지 내부인지를 판별한다.

외부, 내부의 판별은 각 격자를 이루고 있는 한 쪽 방향의 17개 직선으로 이루어지는데, 어떤 직선이 경계곡선과 홀수번 만나는 지점 이전은 외부, 짝수번 만나는 지점 이전은 내부로 인식한다. (도안. 1, 도안. 2)

도안. 1.에서는 절점 및 격자를 분류하였고, 도안. 2에서는 외부와 내부의 판별이후의 결과를 나타내었다.

일단, 격자가 외부, 내부 그리고 경계곡선과 만나는 격자인지의 판별이 끝나면 모든 절점이 외부에 있는 격자는 제거되고, 모든 절점이 내부에 있는 격자는 유지되며, 격자와 경계곡선이 만나는 격자는 도안. 3과 같이 완전히 IN 인 요소이거나, 완전히 OUT 인 요소를 제외한 13가지 형태로 분류된다. 도안. 3에서와 같이 격자와 경계곡선이 만나는 격자들은 -BOU 격자와 -MOD 격자로 나누어지는데, 그것은 격자를 이루고 있는 절점들의 형태에 따라 나누어진다. 절점들의 형태도 -OUT, -IN, -BOU로 나누어지는데, 절점의 -IN, -BOU의 분류는 그 절점에서부터 경계곡선까지의 거리로 판단한다.

이러한 과정을 수행하는 궁극적인 목적은 영역의 분할을 4각 요소로 하기 위함이다. 종종 발생할 수도 있는 삼각 요소를 없애고, 주어진 영역에 충실하게 접근한 4각요소의 생성을 위해서는 새로운 절점이 필요적이었다.

대표적인 경우의 -BOU 및 -MOD 요소의 요소 변화를 도안. 4, 도안. 5와 도안. 6에 나타내었다.

-BOU와 -MOD 요소의 IN인 절점에서 내려지는 수선은 일반적으로 만나는 두 교점사이에서 하나의 선을 긋고 그 선으로 절점에서 수선을 긋는 것이다. (도안. 4)

또한, 모퉁이 각이져진 부분의 요소수정은 도안.5와 같은데, 모퉁이에 가장 근접한 IN 절점에서 두개의 수선이 내려져, 새로운 두개의 절점이 생성되고, 나머지 하나의 절점은 경계와 격자가 만나는 두점사이의 중간 지점에 생성되도록 하였다. (도안. 5)

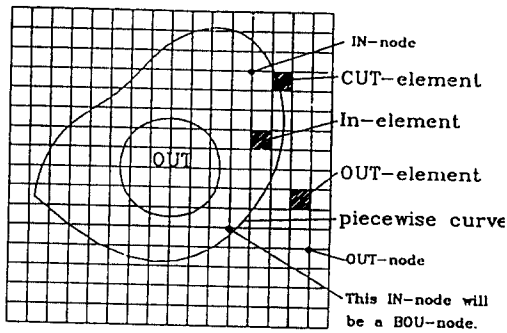
3개의 IN 절점이 존재하는 요소에서는 3개의 IN 절점에서 경계로 수선을 내려, 결과적으로는 해당요소가 두개의 4각 요소로 나누어지게 하였다. (도안. 6)

위와같은 요소생성 과정이후의 중요한 작업은 생성된 각 절점과 요소에 번호를 매기는 일이었다.

절점의 번호는 격자를 이루는 선들의 교점에 의해, 1부터 257까지 매겨지며, OUT 인 격자는 제거되고, IN 인 절점에서 내려지는 수선에 의해 생성된 절점에는 주어진 번호에 300이 더해지고, 만약 도안. 5와 같이 같은 절점에서 또다른 수선이 내려질 경우 600을 이미 주어진 번호에 더하는 방법으로 번호를 정했다.

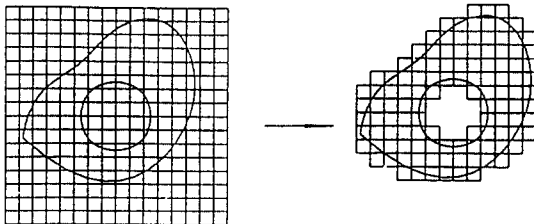
격자의 번호도 우선 격자를 이루는 각 선에 따라서 1111번에서 4444번까지 설정하고, OUT 인 요소는 제거하며, 수정된 요소는 기존에 주어진 번호에 4를 더하고, IN 절점이 3개인 요소(도안. 6)는 원래의 번호에 44를 더하게 하였다.

하지만, 위와같은 번호매김이 순차적이라고는 할 수 없으며, 단지, 요소와 절점의 구분을 위한 것이다.



[도안.1] 격자속의 주어진 영역과 절점의 분류

[Fig. 1] A problem domain with grid line & the classification of node



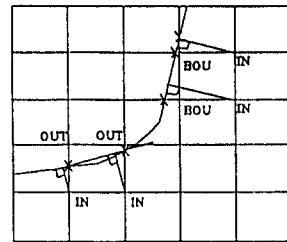
[도안.2] -IN 요소와 -OUT 요소에 따른 격자 구분

[Fig. 2] A division of elements by -IN & -OUT

No.	Feature	Group	No.	Feature	Group
1	4 OUT-nodes 	OUT-element	9	1 BOU-, 2 IN-, 1 OUT-node 	MOD-element
2	1 BOU-, 3 OUT-nodes 	BOU-element	10	3 IN-, 1 OUT-node 	MOD-element
3	1 IN-, 3 OUT-nodes 	BOU-element	11	4 BOU-nodes 	MOD-element
4	2 BOU-, 2 OUT-nodes 	BOU-element	12	3 BOU-, 1 IN-node 	MOD-element
5	1 BOU-, 1 IN-, 2 OUT-nodes 	BOU-element	13	2 BOU-, 2 IN-nodes 	MOD-element
6	2 IN-, 2 OUT-nodes 	MOD-element	14	1 BOU-, 3 IN-nodes 	MOD-element
7	3 BOU-, 1 OUT-nodes 	MOD-element	15	4 IN-nodes 	IN-element
8	2 BOU-, 1 IN-, 1 OUT-node 	BOU-element			
		BOU-element			
		MOD-element			

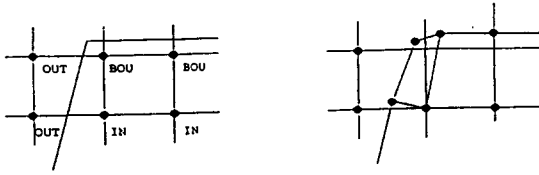
[도안.3] 절점 조합의 정리

[Fig. 3] A summary of node combination

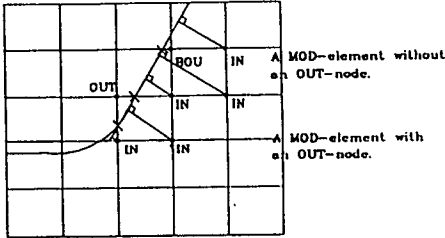


[도안.4] MOD 요소의 IN 노드로부터 수선작도 과정

[Fig. 4] A vertical-line drawing process from IN-nodes of MOD-elements



[도안.5] 모퉁이 부분의 요소 수정과정.
[Fig. 5] A modifying process on the corner



[도안.6] 3개의 IN 절점이 있는 경우의 요소 수정과정
[Fig. 6] A modification process of 3-IN-node cases

3. 경계곡선의 정의

본 논문에서는 경계곡선을 크게,

- ① 직선만으로 구성된 경우.
- ② 원 및 원의 일부와 직선으로 구성된 경우.
- ③ 자유곡선으로 이루어진 경우

로 구분하였다.

직선만으로 구성된 경우는 입력값을 통해서 시작점과 끝점의 좌표로 경계좌표를 구하였고, 원 및 원의 일부와 직선으로 이루어진 경우는 원의 경우 중심점과 반지름, 회전방향 및 회전량으로 정의하였으며, 원의 일부와 직선으로 이루어진 경우는 원과 직선의 접점을 구하여 그 경계좌표를 얻었다. 마지막으로 자유곡선은 위의 경우에서 벗어나는 임의의 곡선을 정의하는데 사용하였다.

일반적으로 자유곡선은 B-spline 곡선식과 Bezier 곡선식으로 많이 표현되어지는데, 여기서는 조정점을 이용하여 좀더 정확한 곡선을 얻을 수 있는 B-spline 곡선식을 이용하였다.

1) 직선만으로 구성된 경우.

곡선의 차수를 1로 정의하여 시작점 및 끝점의 좌표를 입력함으로써 경계를 정의하였다.

2) 원 및 원의 일부와 직선으로 구성된 경우.

원의 경계 입력은 곡선의 차수를 2로 정의하고, $(X - X_c)^2 + (Y - Y_c)^2 = r^2$ 인 원의 방정식에 맞게 중앙점의 좌표 X_c 와 Y_c 및 반지름 r 을 입력하고 원의 회전방향에 따라 반시계 및 시계방향으로 구분하였으며, 회전량을 주기위해 시작점과 끝점의 위치를 그 원에 대한 라디안 값으로 나타내었다.

또한 원과 그 원에 접한 직선의 경계입력을 위해 경계선상의 좌표가 원과 직선이 접한 경우 입력치에 따라 그 접점의 좌표를 모르더라도 자동으로 계산하여 각 곡선의 시작점과 끝점으로 정의 하였다.

3) 자유 곡선.

자유곡선은 차수를 3으로 정의하고 인식하게 하였다.

자유곡선의 보간은 B-spline 곡선식을 이용하였다. B-spline 곡선식은 다음과 같이 기술된다.⁴

$$P(u) = \sum_{i=1}^n P_i N_{i,k}(u)$$

여기서 n : 조정점의 갯수 - 1

u : 매개변수 ($0 \leq u \leq 1$)

B-spline의 보간 함수는 아래의 표현식에 의해 연차적으로 기술된다.

$$N_{i,1}(u) = \begin{cases} 1 & \text{if } t_i < u < t_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

이고,

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}}$$

인데,

여기에서 k 는 u 에 의해 겹과외 다항식의 $(k-1)$ 도항 결정하고, 따라서 또한 곡선의 연속성도 조정한다.

t_i 는 결절치라 하며, 이것은 조정점 P_i 의 매개변수 u 와 관련되어 있다.

개곡면에 대하여 살펴보면,

t_i 는

$$\begin{cases} t_i = 0 & \text{if } i < k \\ t_i = i - k + 1 & \text{if } k \leq i < n \\ t_i = n - k + 2 & \text{if } i > n \end{cases}$$

이고 여기서, i 는

$$0 \leq i \leq n + k$$

인 특성을 갖는다.

이 곡선의 특징은 매개변수를 이용하므로, 주어진 좌표에 독립적으로 궤적의 좌표를 구하고, 특히 주어진 점들이 이루는 다각형안에 궤적이 생기는 특성이 있어, 임의의 경계곡선을 보간하는데 매우 유용하다.⁵

4. 연구의 구성

1) 프로그램의 구성

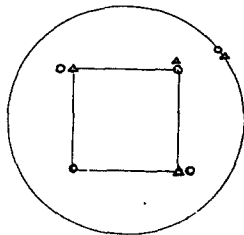
본 논문에 사용된 프로그램은 도안.7과 같이 구성되어 있으며 프로그램의 각 Subroutine은 다음과 같은 역할을 한다.

- subrountine srinpu
경계곡선의 정보를 얻기위한 렉을 받아 처리할 수 있는 형태로 한다.
- subrountine srtang
원과 직선이 접하는 경우 그 접점을 구하여 그것을 곡선의 시작점과 그 곡선의 끝점으로 정한다.
- subrountine srcros
경계 곡선과 격자가 만나는 지를 판별하고 어떤 격자인지를 판별한다.

- subroutine sralig
경계곡선과 격자가 만나는 교점들을 정렬한다.
- subroutine srinsi
격자들의 절점들이 -IN인가를 판별한다.
- subroutine srseta
원에 관한 입력 정보로부터 각도를 계산하고 그것을 라디안 값으로 변환한다.
- subroutine srdoub
교점이 정상적으로 만나는 지, 아니면 증근을 가지는 형태로 만나는 지를 판단한다.
- subroutine srcomb
격자들이 -OUT, -IN, -BOU 인지를 판별한다. 필요따라서 같은 형태의 격자를 결합하기도 한다.
- function ipcasc
생성된 격자들을 재 정렬하여, 그것에 번호를 정할때 각각 격자들의 절점들의 상태를 파악한다.
- subroutine srchan
경계 요소들을 재 정렬한다.
- subroutine srnumb
정렬된 요소에 번호를 정한다.
- function neside
재 정렬되는 요소 격자의 각변의 숫자 및 위치를 정한다.
- subroutine srpost
입력의 집집에 내려져야하는 수선의 수를 검사하고 그것에 따라 정렬한다.
- subroutine bsplinp
B-spline curve 에 관련된 subroutine으로서 곡선이 주기적인 경우의 궤적을 생성하여 경계좌표들을 구한다.
- subroutine bsn
B-spline curve 에 관련된 subroutine으로서 곡선이 비주기적인 경우의 궤적을 생성하여 경계좌표들을 구한다.
- subroutine dwg
Auto-CAD에서 출력력이 가능하도록 script file을 생성한다.

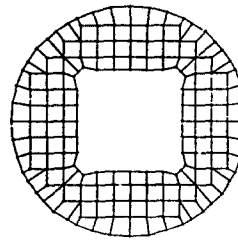
2) 입력의 예

- ① 직선만의 폐곡선과 원 (도안.8-A)
- ② 원 및 원의 일부와 그것에 접하는 직선으로 구성된 폐곡선 (도안.9-A)
- ③ B-spline 곡선으로 이루어진 경우.
 - 주기적 특성을 가지는 B-spline 곡선 (조정점 6개, 도안.10-A)
 - 비주기적 특성을 가지는 B-spline 곡선 (조정점 9개, 도안.11-A)

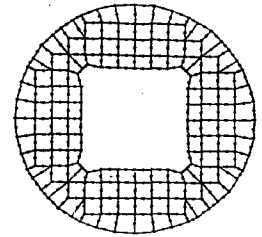


(a) Boundary Type

△ : Start point
○ : End points

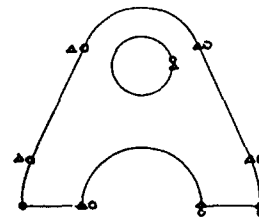


(b) 4-node mesh



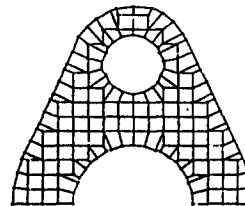
(c) 8-node mesh

[도안.8] 4각 구멍이 있는 링의 요소분할
((a) 경계 형태 (b) 4절점 요소분할 (c) 8절점 요소분할)
[Fig. 8] A mesh generation of the ring with a rectangular hole.
(a) Boundary type (b) 4-node mesh (c) 8-node mesh

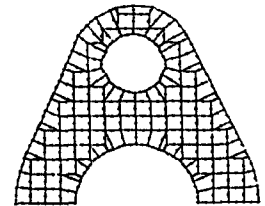


(a) Boundary type

△ : Start point
○ : End points

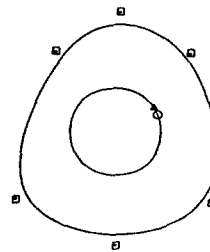


(b) 4-node mesh



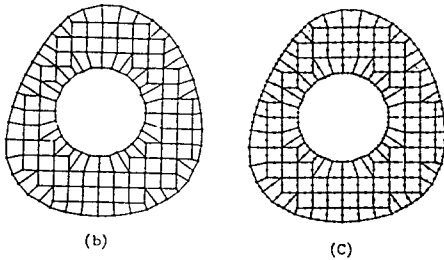
(c) 8-node mesh

[도안.9] 패킹 그랜드의 요소분할
((a) 경계 형태 (b) 4절점 요소분할 (c) 8절점 요소분할)
[Fig. 9] A mesh generation of the packing grand
(a) Boundary type (b) 4-node mesh (c) 8-node mesh

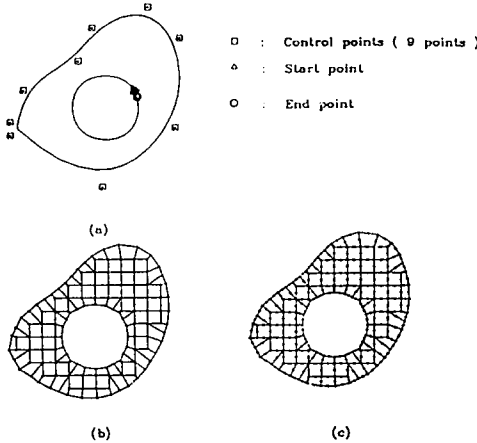


(a)

□ : Control points (6 points)
△ : Start point
○ : End point



[도안.10] 주기적 B-spline 곡선을 지닌 캠의 요소분할
 ((a) 경계 형태 (b) 4절점 요소분할 (c) 8절점 요소분할)
 [Fig. 10] A mesh generation of the cam with
 periodic B-spline curves
 ((a) Boundary type (b) 4-node mesh (c) 8-node mesh)



[도안.11] 비주기적 B-spline 곡선을 지닌 캠의 요소분할
 ((a) 경계 형태 (b) 4절점 요소분할 (c) 8절점 요소분할)
 [Fig. 11] A mesh generation of the cam with
 non-periodic B-splines curves
 ((a) Boundary type (b) 4-node mesh (c) 8-node mesh)

5. 결과분석 및 고찰

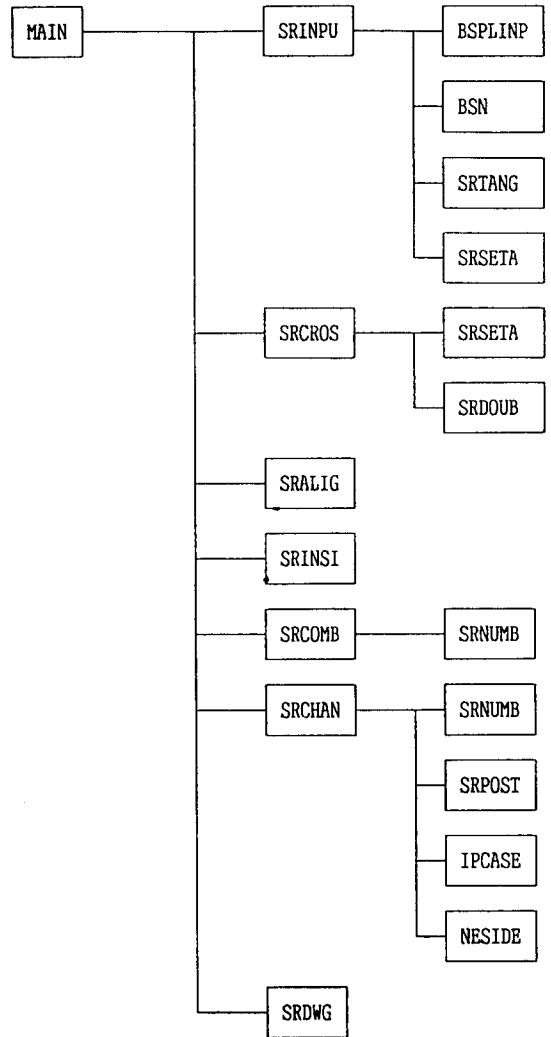
도안. 8 에는 4각 홈(HOLE)이 있는 링을 요소분할한 결과를 나타내었고, 도안. 9에는 packing grand의 형상에 대한 요소분할을 나타내었으며, 도안. 10에는 cam의 형태를 지닌 영역을, 도안. 11에는 또다른 형태의 cam을 요소분할하여 그 결과를 나타내었다.

일반적으로 요소생성은 경계에 적합하였으나, 도안.12에서 보는 바와 같은 비정상적 요소가 발생하였다.

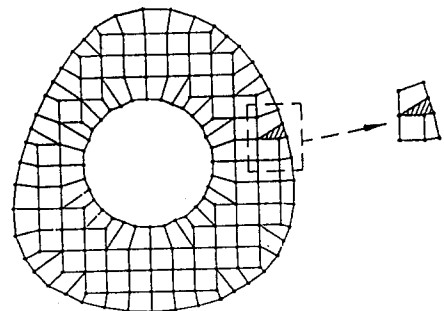
여기서, 비정상적 요소란 유한요소법으로 그 해를 구할 때 임의의 요소가 전체 영역의 좌표(Global coordinate)에서 국부 영역의 좌표(Local coordinate)로 사상(Mapping)될 경우 사상(Mapping)에 필요한 자코비안 전이 행렬(Jacobian matrix)의 행렬식이 양의 값을 갖지 못함을 의미한다.

단지, 경계의 형태에 따라 요소분할이 이루어진 관계로 각 요소의 생성된 모양이 형상함수에 주는 영향은 고려하지 못했기 때문이다. 또한, 전처리부(preprocessor)에서 받은 정보(data)를 직접 처리부(processor)에서 처리하기 위해서는 절점 및 요소의 순차적 번호 매김도 필요하겠다.

아울러, PC가 가지고 있는 memory의 한계도 충분히 고려된 프로그래밍이 있어야 하겠다.



[도안 7] 프로그램 블록 선도
 [Fig. 7] Program Block Diagram



[도안.12] 비정상적 요소
 [Fig. 12] Adnomality element

6. 결 론

자동으로 요소를 분할하는 가장 큰 이유는 주어진 영역을 매우 많은 요소로 분할할 경우에 방대한 양의 데이터를 수동으로 입력하는 번거러움을 줄이는 것에 있다.

본 논문은 그러한 번거러움을 많이 줄였다고 사료되며, 또한 기하학적 인식에 의한 경계의 정의 등은 기하학적 모델링에도 많은 도움이 되리라고 판단된다.

다만, 주어진 기하학적 형상의 요소분할에만 고려되어 비정상적인 요소가 꽤 많은 부분에서 발견되었다.

F.E.M solver를 이용하기 위해서는 분할된 요소의 적당함을 만족시켜줄 수 있는 강력한 판단 장치가 요구된다.

정당한 요소 판정의 기준을 언급해 본다면,

- 4각요소의 한각이 0° 에 가깝지 않아야함.
- Mapping Jacobian이 음이 압되도록 형상에 대하여 미리 염두해 두어야 함.

등을 언급할 수 있다.

참고 문헌

1. O. C. Zienkiewicz and R. L. Taylor, " The Finite Element Method, Vol.1 , McGRAW-HILL Book Company(1989)
2. Eric B. Becker, Graham F. Carey and J. Tinsley Oden, " FINITE ELEMENTS ", vol 1, 2 , PRENTICE-HAL, Inc. Englewood Cliffe, New Jersey) 07632. (1981)
3. M.A.Yerry and M.S.Shephard, " A Modified Quadtree Approach to Finite Element Mesh Generation " ,IEEE Computer graphics and applications, vol.3,1, (1993)
4. Micheal E. Mortenson, " Geometric Modeling ",98-125, John Wiley & Sons (1985)
5. Gerald Farin," Curves and Surfaces for Computer Aided Geometric Design ", Academic Press, Inc (1990)