

2차원 자동요소분할 프로그램 개발에 관한 연구

조 명 철*, 유 형 선**

A Study on the Two-Dimensional Automatic Mesh Generation Programming

Myung Cheol Joe*, Hyeong Seon Yoo**

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. 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. And the node very close to the boundary is moved to the boundary. The automatic mesh generation composed of only rectangular elements is achieved by this procedure.

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 rectangular elements for the given boundaries.

1. 서 론

유한요소법(Finite Element Method)은 공학에서 주로 다루는 미분방정식의 해를 수치해석법을 이용하여 구해내는 것으로서, 그 구성은 첫째 전처리부(Preprocessor)로 미분방정식 및 문제의 영역에 관련된 정보를 받아 처리부에서 해를 구할 수 있도록, 미분방정식의 영역을 유한개의 요소로 나누고, 이 요소들에 대한 정보를 생성하여 유한요소법에서 해를 구할 수 있는 자료를 입력하는 것이며, 두번째 처리부(Processor)는 전처리부에서 받은 정보를 이용하여 미분방정식의

해를 구하는 것이고, 마지막 후처리부(Postprocessor)는 처리부에서 받은 해를 화일(File)에 저장하거나, 혹은 컴퓨터 모니터상에 시각적으로 출력하는 과정을 수행한다.^{1,2}

본 논문은 유한요소법에 있어서의 전처리부에 해당하는 것으로서 주어진 문제영역의 경계가 곡선이거나 주어진 문제의 영역의 내부에 빈 영역 즉, 구멍(Hole)이 있는 경우에도 자동으로 문제의 영역을 유한개의 요소로 분할하는 방법(Automatic mesh generation)에 관한 연구이다.

본 논문에 앞서 1986년 N. Kikuchi는 자동요소분할

* 仁荷大學校 自動化工學科 (정회원)

** 仁荷大學校 自動化工學科 (정회원)

을 적용 사상, 차분방정식 그리고 대수적 기법을 이용한 방법을 제시하였으며,³ P. L. Baehmann과 S. L. Wittchen등은 기하학적·기초를 둔 상태하에서의 자동요소분할에 대한 논문^{4,5}을 1987년 발표한 바 있다.

그러나, 기존의 사상(Mapping)³이나, 유한차분법(Finite Difference Method)등에서 사용하는 대차식완화법(S. O. R.) 방식의 차분법⁶ 등은 복잡한 형상의 경계나, 빈 영역이 있는 경우의 요소분할이 어려우며, Delaunay 법등을 이용할 경우⁷ 3각형 요소만의 요소분할에는 용이하나, 4각형요소만으로 이루어진 자동요소분할에는 부적합하였다.

본 논문은 연속적으로 4연속등분하는 Quadtree방식의 초기격자를 생성하고, 그 격자를 주어진 문제 영역에 알맞게 변형함으로써 자동요소분할하고자 하였으며, 2차원에서의 격자의 상태를 Tree구조로 정렬한 Quadtree방식을 이용한 것은 방대한 양의 정보를 체계적으로 정리하고 또한, 그 관리기법에서 얻을 수 있는 Memory의 감소로 소형컴퓨터 즉, P. C. 에서 본 과정을 수행하기 위해서이다.

초기격자의 생성은 기존의 Quadtree방식과 유사하게 적용하였으나, 경계부분의 격자변형방식은 새로운 수선작도방식을 제시, 적용하여, 기존의 연구에서는 얻을 수 없는 비교적 다양한 영역에 대한 경계에 충분히 수렴하는 4각형요소만으로 이루어진 요소분할을 이룩하였다.

또한, 되도록 최소한의 경계입력을 주어 영역의 경계곡선을 생성하였고, 특히 B-spline 곡선식¹⁷을 이용하여 최소의 입력으로 다양한 형태의 경계곡선을 생성, 처리하였다.

2. 요소분할 방법

2.1 초기격자의 생성과정

본 연구에서 사용한 초기격자 생성 방법을 기술하면, 연속적으로 정사각형영역을 4등분해 나가는 것이다. (Fig. 1)

Fig. 1의 (a)에 나타난 것과 같이 연속적으로 등분하면, Fig. 1의 (b)에서와 같이 격자의 생성은 Tree구조를 갖게된다. 본 논문에서의 초기격자는 4연속 등분, 즉 4차에 걸쳐 4등분하여, 257 ((2⁴)²+1)개의 격자로 이루어진 초기격자를 생성하였다.

이것은 각 격자요소의 관련 정보줄 Tree구조의 형식

으로 관리할 수 있으므로 각 정보의 관리가 수월하였고, 각 격자의 상태가 변해감에 따라 그 정보를 유지하거나 제거하기 편리하였으며, 변형시의 정보수정도 편리했다. 또한 이러한 방법의 활용으로 인해 Memory의 절약이 가능하였으므로 본 연구가 추구하는 소형컴퓨터, 즉 P. C. 상에서도 경계에 적절한 4각형 요소만의 자동요소분할이 가능하였기 때문이다.

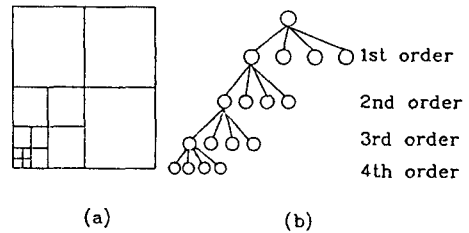


Fig. 1 The generating procedure of initial grid line by quadtree.

2.2 격자와 경계의 겹침을 통한 절점 및 격자의 분류과정

Fig. 1과 같은 방식으로 생성된 격자들 속에 경계곡선을 생성시키면, Fig. 2에서와 같은 형태로 나타난다.

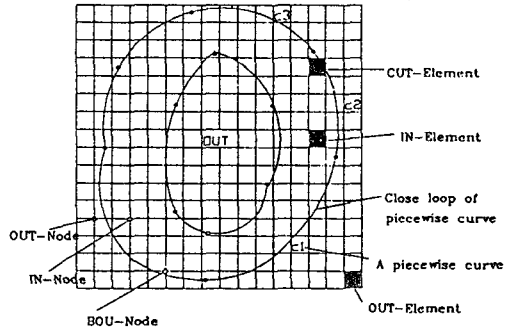


Fig. 2 The division of elements and nodes.

격자와 주어진 경계곡선과 어떤 격자에서 만나는 지를 확인하여, 각 격자를 구성하는 절점이 외부인지 내부인지를 판별한다.

외부, 내부의 판별은 각 격자를 이루고 있는 한 쪽 방향의 17개 직선으로 이루어지는데, 어떤 직선에서 경계곡선과 홀수번 만나는 지점 이전은 외부, 짝수번 만나는 지점은 내부로 인식한다.

우선 격자를 구성하는 절점의 구분은, 절점이 완전히

경계영역 밖(OUT-Node)인 경우와 완전히 경계영역 안(IN-Node)로 나누어지고, 격자는 격자를 구성하는 4개의 절점이 모두 밖(OUT-Node)인 OUT-Element인 경우와 절점 4개가 모두 안(IN-Node)인 IN-Element, 그리고, 격자를 구성하는 4개의 절점중 1개에서 3개만이 안(IN-Node)인 CUT-Element로 구분된다.

4개의 절점 모두가 밖(OUT-Node)인 경우는 경계영역과는 아무런 관계가 없으므로, 그 격자를 제거하면, Fig. 3의 오른쪽 그림과 같은 격자만이 남게된다.

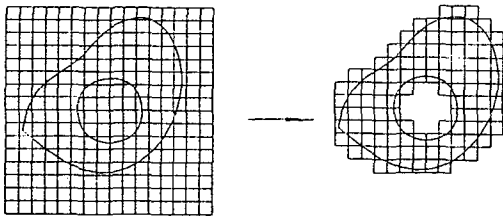


Fig. 3 The eliminating procedure of OUT-elements.

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			
3	1 IN-, 3 OUT-nodes 	BOU-element	10	3 IN-, 1 OUT-node 	MOD-element
4	2 BOU-, 2 OUT-nodes 	BOU-element	11	4 BOU-nodes 	MOD-element
5	1 BOU-, 1 IN-, 2 OUT-nodes 	BOU-element	12	3 BOU-, 1 IN-node 	MOD-element
6	2 IN-, 2 OUT-nodes 	MOD-element	13	2 BOU-, 2 IN-nodes 	MOD-element
7	3 BOU-, 1 OUT-nodes 	MOD-element	14	1 BOU-, 3 IN-nodes 	MOD-element
8	2 BOU-, 1 IN-, 1 OUT-node 	BOU-element	15	4 IN-nodes 	IN-element

Fig. 4 The summary of the elements by nodes-combination.

Fig. 3에서와 같이 유지된 격자를 살펴보면, 절점 4개가 모두 안(IN-Node)인 IN-Element와 절점 1개에서 3개가 안(IN-Node)인 CUT-Element로 구분되며, CUT-Element의 OUT-Node와 IN-Node 중에서 경계곡선에 비교적 가까운 절점은 BOU-Node로 구분하며, 그 판단 기준은 격자를 구성하는 정사각형의 한번의 길이의 1/3보다 가까운 것으로 판단하였다.

BOU-Node의 결정이 끝나면, CUT-Element와 MOD-Element로 나누어져 처리되게 된다.

Fig. 4에 경계에 따른 격자를 분류하여 나타내었다.

2.3 수선작도법에 의한 격자의 변형과정

경계와 격자가 만나는 CUT-Element는 Fig. 4에서 나타난 BOU-Element와 MOD-Element로 나누어져, 본 논문이 추구하는 4각형요소로 경계에 충분히 적절하게 변형된다.

일반적으로 BOU-Element는 MOD-Element에 비하여, 경계영역내부에 속하는 격자의 영역이 작은 것이다. 이 BOU-Element에 속하는 격자들은 MOD-Element에 흡수되어 재정렬되고, 이 MOD-Element의 변형과정이 본 논문에서 가장 중요한 수선 작도법으로, 경계에 적절한 4각형 요소로 수정되는 것이다.

일반적으로 수선작도법이란 격자와 경계가 만나는 격자내의 IN인 절점으로부터 경계곡선과 격자가 만나는 두점사이에 한 선분을 긋고 선분상으로 수선을 내리는 것이다. 수선을 내리는 절점은 항상 IN인 절점이며, BOU절점은 격자 변형시 소멸되거나 경계로 이동하여 경계에 적합 하게된다.

대표적인 경우의 MOD-Element의 격자 변화를 Fig. 5, Fig. 6와 Fig. 7에 나타내었다.

MOD-Element의 IN인 절점에서 내려지는 수선은 일반적으로 경계와 격자가 만나는 두 교점사이에 하나의 선을 긋고 그 선상으로 IN인 절점에서 수선을 긋는 것이다.

또한, 모퉁이 부분의 격자수정은 Fig. 6과 같은데, 모퉁이에 가장 근접한 IN 절점에서 두개의 수선이 내려져, 새로운 두개의 절점이 생성되고, 나머지 하나의 절점은 경계와 격자가 만나는 두점사이의 중간 지점에 생성되도록 하였다. 또한 그림에서 보는 바와 같이 BOU절점은 경계로 이동함을 볼 수 있다. (Fig. 6)

Fig. 7과 같이 3개의 IN절점이 존재하는 격자에서는 3개의 IN절점에서 경계로 수선을 내려, 결과적으로는

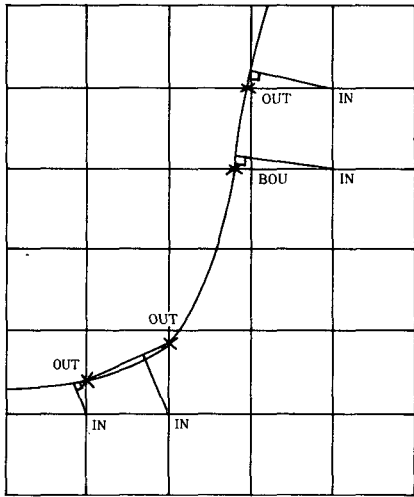


Fig. 5 The vertical line drawing procedure from IN-node of MOD-element.

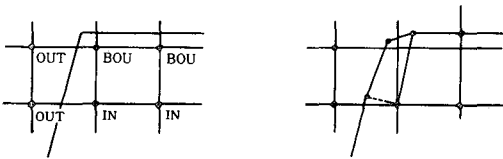


Fig. 6 The modifying procedure of the corner MOD-elements.

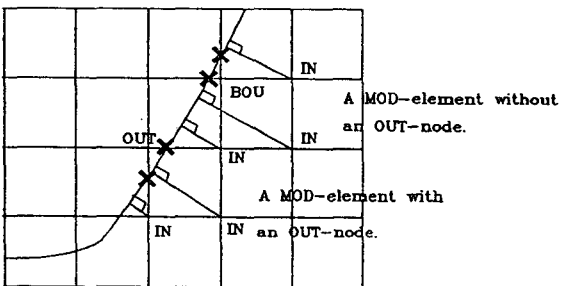


Fig. 7 The vertical line drawing procedure from 3 IN-nodes of MOD-element.

해당격자는 두개의 4각형 격자로 나누어지게 하였다. 위와같은 과정을 통해 격자를 수정하는 이유는 본 논문이 추구하는 4각형 격자만으로 구성된 요소분할이 목적이기 때문이다.

경계와 격자가 만나는 부분에서 볼 수 있는 여러 형태의 경우를 정리하고 구분하여 처리함으로써 다양한 형태의 경계곡선에도 충실히 접근하여 자동요소분할하려 하였다.

2.4 절점과 격자의 번호 매김 과정

요소생성 과정이후의 중요한 작업은 생성된 각 절점과 격자에 번호를 매기는 일이었다. 초기격자에 따라 나타난 번호매김과정을 Fig. 8에 나타내었다.

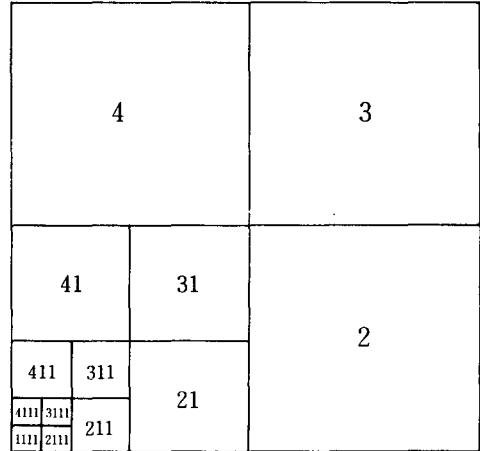


Fig. 8 The numbering of initial elements.

절점의 번호는 격자를 이루는 선들의 교점에 의해, 1부터 257까지 매겨지며, OUT-Element를 구성하는 OUT 절점은 제거되고, IN-Element를 구성하는 IN인 절점은 그 번호를 유지하며, IN인 절점에서 내려지는 수선에 의해 생성된 절점에는 주어진 번호에 300이 더하여지고, 만약 그림 6와 같이 같은 절점에서 또다른 수선이 내려질 경우 600을 이미 주어진 번호에 더하는 방법으로 번호를 정했다.

격자의 번호도 우선 격자를 이루는 각 선에 따라서 1111에서 4444번까지 설정하고, OUT인 격자는 제거하며, 수정된 격자는 기준의 주어진 번호에 4를 더하고, Fig. 7과 같이 IN 절점이 3개인 격자에 의하여 생성된 또 하나의 격자는 원래의 번호에 44를 더하게 하였다.

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

3. 경계곡선의 정의

본 논문에서는 경계곡선을 크게,

- ① 직선만으로 구성된 폐곡선,
 - ② 원 및 원의 일부와 직선으로 구성된 폐곡선,
 - ③ 자유곡선으로 이루어진 폐곡선
- 로 구분하여 처리하였다.

3.1 직선만으로 구성된 폐곡선

곡선의 형태를 1로 정의하여 시작점 및 끝점의 좌표를 입력함으로써 경계를 정의하였다. 폐곡선의 경우 한 직선의 끝점은 다음 직선의 시작점으므로 시작점만 입력하였고, 끝점의 입력은 생략하였다.

3.2 원 및 원의 일부와 직선으로 구성된 폐곡선.

원의 경계 입력은 곡선의 형태를 2로 정의하고,

$$(X - X_c)^2 + (Y - Y_c)^2 = r^2 \dots (1)$$

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 점점의 좌표를 모르더라도 자동으로 계산하여 각 곡선의 시작점과 끝점으로 정의 하였다.

3.3 자유 곡선으로 구성된 폐곡선

자유곡선은 곡선형태를 3으로 정의하고 인식하게 하였다.

자유곡선의 보간은 B-spline 곡선식을 이용하였다. B-spline 곡선식은 다음과 같이 기술된다.¹⁷

$$P(u) = \sum_{i=1}^n P_i N_{i,k}(u) \dots (2)$$

여기서 n : 조정점의 갯수 - 1
 u : 매개변수 ($0 \leq u \leq 1$)

B-spline의 보간 함수는 다음의 표현식에 의해 연차적으로 기술된다.

$$\begin{cases} N_{i,1}(u) = 1 \text{ if } t_i < u < t_{i+1} \\ = 0 \text{ otherwise} \end{cases} \dots (3)$$

이고,

$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_{i+k} - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \dots (4)$$

인데,

여기에서 k 는 u 에 의해 결과적 다항식의 $(k-1)$ 도를 결정하고, 따라서 또한 곡선의 연속성도 조정한다. t_i 는 결절치라 하며, 이것은 조정점 P_i 의 매개변수

와 관련되어 있다.

개곡선에대하여 살펴보면,

t_i 는

$$\begin{cases} t_i = 0 & \text{if } i < k \\ t_i = i - k + 1 & \text{if } k \leq i < n \dots (5) \\ t_i = n - k + 2 & \text{if } i > n \end{cases}$$

이고, 여기서, i 는

$$0 \leq i \leq n + k \dots (6)$$

인 특성을 갖는다.

이 곡선의 특징은 매듭수를 이용하므로, 주어진 좌표에 독립적으로 궤적의 좌표를 구하고, 특히 주어진 점들이 이루는 다각형안에 궤적이 생기른 특성이 있어, 임의의 경계곡선을 보간하는데 매우 유용하다.¹⁷

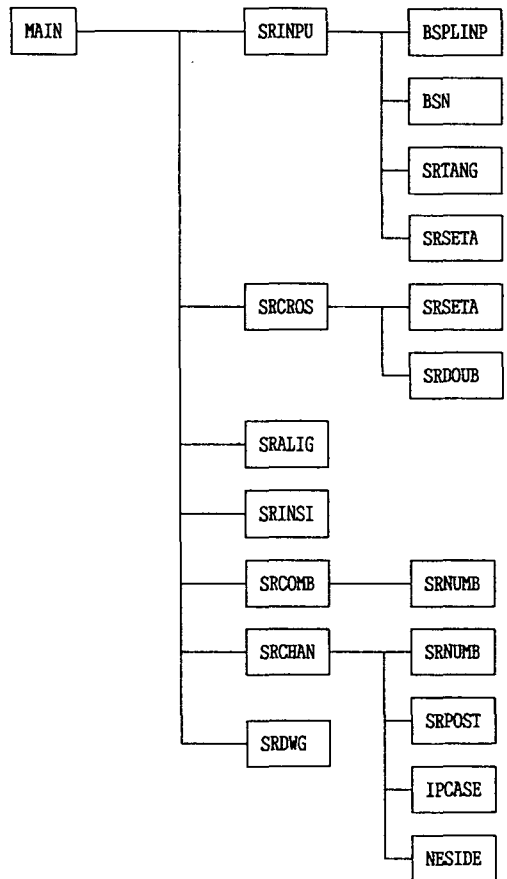


Fig. 9 Block-diagram.

4. 연구의 구성

본 논문에 사용된 프로그램은 Fig. 9와 같이 구성되어 있으며 프로그램의 각 Subroutine 및 함수(Function)는 다음과 같은 역할을 한다. (Fig. 9)

- subroutine srinpu
경계곡선을 얻기위한 입력 정보를 입력 파일로부터 받아, 주어진 곡선 형태를 인식하고 그 형태에 따라 필요한 Subroutine을 호출하여, 본 프로그램에서 처리할 수 있는 곡선의 형태로 한다.
- subroutine srtang
원과 직선이 접하는 경우 그 접점을 구하여 그것을 곡선의 시작점과 그 곡선의 끝으로 정한다.
- subroutine srcros
경계 곡선과 격자가 만나는 지를 판별하고 어떤 격자인지를 판별한다.
- subroutine sralig
경계곡선과 격자가 만나는 교점들을 정렬한다.
- subroutine srinsi
격자들의 결점들이 -IN인가를 판별한다.
- subroutine srseta
원에 관한 입력 정보로부터 각도를 계산하고 그것을 라디안 값으로 변환한다.
- subroutine srdoub
교점이 정상적으로 만나는지, 아니면 중근을 가지는 형태로 만나는지를 판단한다. 즉, 일반적으로 하나의 격자에 대하여 경계곡선이 두 점에서 만나는 반면, 어떤 특수한 경우 2차곡선과 같이 격자의 한면에 대하여 중근을 갖는 형태로 만날 수도 있는데 이런 경우를 판단하고, 이 경우의 격자의 변형을 위한 정보를 제공한다.
- subroutine srcomb
격자들이 -OUT, -IN, -BOU 인지를 판별한다.
- function ipcse
생성된 격자들을 재 정렬하여, 그것에 번호를 정할때 각각의 격자들의 절점들의 상태를 파악한다.
- subroutine srchan
경계 요소를 재 정렬한다. 즉, 어떤 절점에서 수선이 작도되든지, 또한 작도되는 수선의 갯수는 몇개인지를 판단하고 격자의 이동을 판별하며, 이런 변형을 통해 얻은 격자 및 절점의 상태를 정리 등, MOD격자와 BOU격자의 변형에

관련 전반적 과정을 수행한다.

- subroutine srnumb
정렬된 요소에 번호를 정한다.
- function neside
재 정렬되는 요소 격자의 각변의 순서 및 위치를 정한다.
- subroutine srpost
임의의 절점에서 내려져야하는 수선의 수를 판단하고 그것을 subroutine srchan에 전달한다.
- subroutine bsplinp
B-spline curve에 관련된 subroutine으로서 곡선이 주기적인 경우의 궤적을 생성하여 경계좌표를 구한다.
- subroutine bsn
B-spline curve에 관련된 subroutine으로서 곡선이 비주기적인 경우의 궤적을 생성하여 경계좌표를 구한다.
- subroutine dwg
Auto-CAD에서 출력이 가능하도록 script file을 생성한다.

5. 결과분석 및 고찰

5.1 결과예와 결과분석

Fig. 10에는 4각 홀(HOLE)이 있는 링을 요소분할한 결과를 나타내었고, Fig. 11에는 패킹 그랜드(Packing

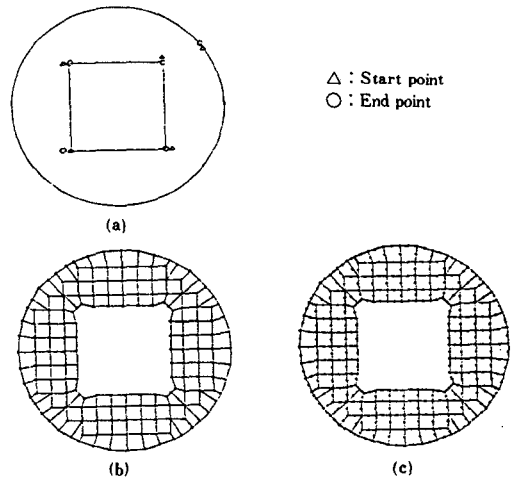


Fig. 10 The mesh-generation of the ring with rectangular-hole.
(a) boundary tupe (b) 4-node mesh (c) 8-node mesh

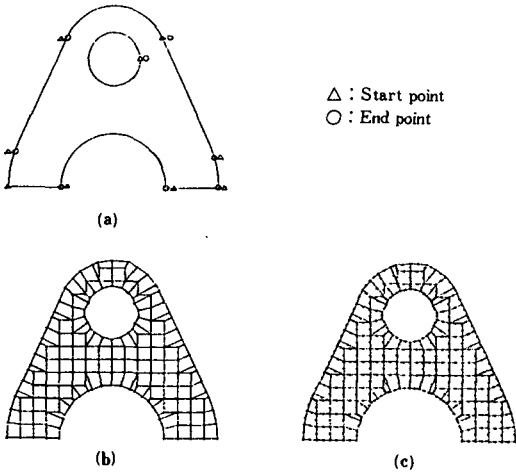


Fig. 11 The mesh-generation of packing grand.
(a) boundary tupe
(b) 4-node mesh (c) 8-node mesh

grand)의 형상에 대한 요소분할을 나타내었으며, Fig. 12에는 주기적 B-Spline 곡선으로 생성한 캠(Cam)의 형태를 지닌 영역을, Fig. 13에는 비주기적 B-Spline 곡선식으로 생성한 또다른 형태의 캠(Cam)을 요소분할하여 그 결과를 나타내었다. (Fig. 10, Fig. 11, Fig. 12, Fig. 13)의 4각 홀(Hole)부분에는 모따기를 한 형태와 같은 오차가 나타나는 데 그것을 Fig. 14에서 보는 것과 같이 3개의 IN 절점에서 경계와 격자가 만나는

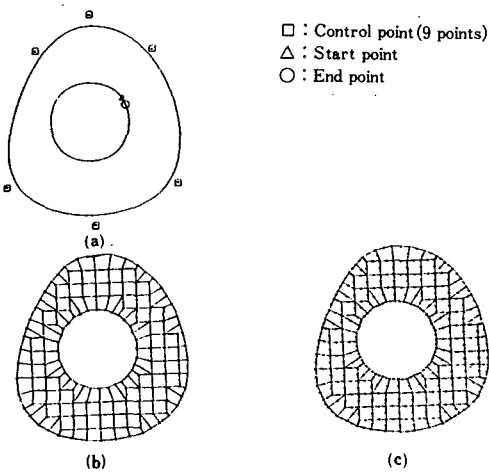


Fig. 12 The mesh-generation of the cam with periodic B-spline curve.
(a) boundary tupe
(b) 4-node mesh (c) 8-node mesh

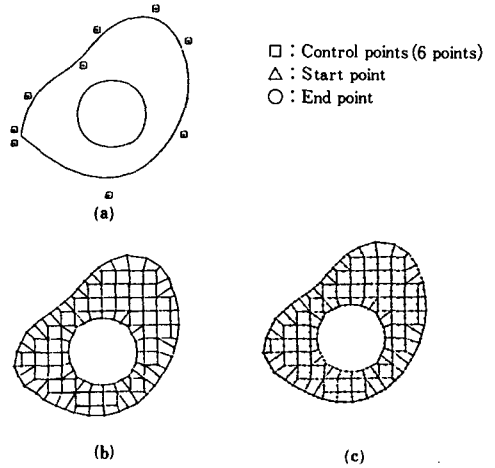


Fig. 13 The mesh-generation of the cam with non-periodic B-spline curve.
(a) boundary tupe
(b) 4-node mesh (c) 8-node mesh

두점을 지나는 선분상으로 수선을 내릴 경우 가장 최적화 하더라도 주어진 과정으로는 이와같은 오차를 피할 수는 없었기 때문이다. 즉, 격자의 크기 자체에 의하여 격자의 현재의 크기로는 위의 오차를 피할 수 없으리라 본다. (Fig. 14)

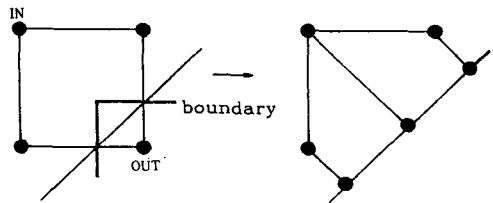


Fig. 14 The error of the corner.

5.2 결과예에서 나타난 이상요소

Fig. 15에는 결과예에서 나타난 대표적인 비정상적 요소를 나타내었다. 결과예의 많은 부분에서 이와 같은 이상요소가 나타났으나, 위의 경우가 가장 나쁘다고 생각하여 기술하였다. (Fig. 15)

주어진 경계에 충실 요소생성만을 본 연구는 수행하였기에 아래와 같은 요소가 발생하였다고 생각한다. 일반적으로 유한요소법의 해를 구하는 프로그램의 수행시 4각형요소로 구성된 요소를 처리하기 위해서는 두변사이의 각이 약 30도에서 150도 내의 경우가 Error를 피할 수 있다고 생각한다.

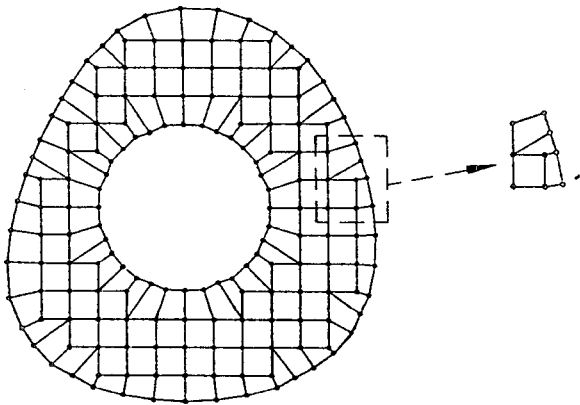


Fig. 15 The abnormal element.

6. 결론

자동으로 요소를 분할하는 가장 큰 이유는 주어진 문제의 영역을 매우 많은 요소로 분할할 경우에 방대한 양의 수동적 입력을 조금이라도 줄이자는데에 있다.

본 논문의 경우 매우 많은 요소를 요구하는 형상에 관한 요소분할을 단지 경계곡선의 정의만으로 실시간내에 영역을 분할하고 Auto-CAD와 같은 상용프로그램을 이용 그 결과를 그래픽적으로 처리하여 나타내었다. 또한, 경계에 적합한 격자의 생성을 위하여 본 논문에서 제시한 방법을 이용한 결과, 다양한 경계에 대해서도 적절한 형태의 4각요소로 구성된 자동요소분할을 이루었다. 기존의 논문들이 사상(Mapping)이나, Delaunay법 등을 사용하고 있지만, 사상(Mapping)의 경우 경계의 형상이 복잡해지면 그 결과의 신뢰성은 기대하기 어려우며, Delaunay법 등은 4각형 요소만의 요소분할에는 적합하지 않았다. 본 논문은 이와같이 문제를 해결하기 위하여, 경계부분의 요소를 그 경계의 형태에 따라 분류 처리하고, 그 경계요소의 변형을 위한 새로운 기하학적 수선작도법을 제시하고 적용하였으며, 그 결과 몇 개의 요소에서 오차가 발생하였으나, 비교적 다양한 영역의 경계에 적합한 4각형요소만의 자동요소분할을 이루었다.

향후과제로는 주어진 격자나 절점의 순차적 번호매김을 통해 처리부에 적용되어야 하며, 또한 비정상적 요소의 발생을 막을 수 있는 Program 기법 및 형상에 대한 오차를 줄일 수 있도록 더욱더 조밀한 격자로 구성된 초기격자를 이용하여 소형컴퓨터(P. C.)에서도 자

동요소분할할 수 있도록, Memory를 더욱 절약할 수 있는 Program 기법 및 적응해석법 등의 기법과 연결하여, 최적의 격자만으로 구성된 자동요소분할을 기대해 본다.

참고문헌

1. O. C. Zienkiewicz and R. L. Taylor, "The Finite Element Method", Vol. 1, McGRAW-HILL Book Company, PP. 1~3, PP. 183~191, PP. 290~297, 1989.
2. E. B. Becker, G. F. Carey and J. T. Oden, "FINITE ELEMENTS", Vol 1, 2, PRENTICE-HALL Inc., Englewood Cliffe, New Jersey, PP. 73~82, PP. 175~221, 1981.
3. N. Kikuchi, "Adaptive grid design methods for Finite Element Analysis", Computer methods in applied mechanics and engineering, PP. 129~160, 1986.
4. P. L. Baehmann, S. L. Wittchen, M. S. Shephard, K. R. Grice and M. A. Yerry, "Robust Geometrically Based, Automatic 2-dimensional Mesh Generation", International Journal Numerical Methods in Engineering, Vol. 24, PP. 1043~1078, 1987.
5. M. A. Yerry and M. S. Shephard, "A Modified Quadtree Approach to Finite Element Mesh Generation", IEEE Computer graphics and applications, Vol. 3, PP. 39~46, 1983.
6. M. E. Mortenson, "Geometric Modeling", John Wiley & Sons, PP. 98~125, 1985.
7. G. Farin, "Curves and Surfaces for Computer Aided Geometric Design", Academic Press, Inc, PP. 147~171, 1990.
8. 양준서, "곡면 경계를 포함한 영역내에서의 자동분할에 관한 연구", 인하대학교 기계공학과 자동화공학전공 석사학위청구논문, PP. 19~31, 1991.
9. C. F. Bryant, "Two dimensional automatic triangular mesh generation", IEEE Trans. Magnetics, Vol. MAG-21, No. 6, PP. 21~30, 1963.