

그래픽 에디터를 이용한 자동 요소 분할에 관한 연구 (A Mesh Generation Method using PC Graphic Editor)

김 동 일, 유 형 선
(Dong-Il Kim, Hyeong-Seon Yoo)

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

Abstract

In this paper it is discussed about the program which makes automatically the preprocessing data on the graphic editor of PC monitor. Three kinds of colors are used in order to make know a boundary_element, in_element, out_element. By initial mesh generation, three kinds of elements are made; the element with three nodes, four nodes, five nodes. The node of boundary element draw a vertical line to the boundary. At that time the element with three nodes is removed and the element with four nodes is changed to the element with four nodes. The domain graphic on this paper is compose of piecewise curves which has many lines and a hole. The data of element is written on the memory of PC computer. The program controls the memory address according to the change of element's quantity; disappearance and addition of element. All this work is operated on the PC computer.

1. 서론

유한 요소 해석에서 문제의 영역을 분할하고 자료를 입력하는 과정을 쉽고 정확하게 처리하기 위한 연구가 많이 수행되고 있다.

본 논문에서는 요소분할에 필요한 입력자료를 그래픽 에디터에 의하여 작성된 도면으로 대처하는 자동 요소분할법을 다룬다. 기존의 프로그램은 대형 컴퓨터에서 사용 가능하였으나, 본 프로그램은 컴퓨터의 메모리를 경제적으로 사용함으로써 PC에서 사용이 가능하며, 출력도면은 널리 사용되고 있는 AUTO/CAD와 연결이 가능하다. 사용되는 요소의 형태는 사각형이며, 요소의 경계점에서 발생하는 요소의 형태에 따라 프로그램 에러를 최소화 하기위해 요소를 최적설계 하였다.

2. 요소의 분할

2.1 그래픽 작업

대상물이 유한요소로 분할되어 질때 외부점, 경계점, 내부 점을 인식할 수 있도록 3가지 색깔 color-in, color-bou,

color-out등으로 구분 되어진다. 작업은 크게 Draw작업, File의 load와 save작업, 내부의 색깔(color-in)을 칠하는 작업으로 이루어지며, Mouse와 interface가 되어진다.

2.2 초기의 요소분할

화면의 기준점으로 부터 그림 1 에서와 같이 사각형의 불력을 쌓아가는 방식으로 분할되며, 이때 각 절점의 색깔과 주어진 영역과 불력사이의 교점을 구하게 된다.

각 절점의 색깔에 따라 내부점 외부점이 구분되어 지며 영역 경계와의 교점에 따라 요소의 형태가 결정되어진다. 예를 들어 한 불력에서 2개의 절점을 택했을 경우 두 절점이 모두 color-in이면 2개 모두 내부 절점으로 인식되어 지며, 한 절점은 color-in이고 다른 절점이 color-out이면 두 절점 사이와 영역 경계와의 교점을 구하게된다. 이때 두 절점의 중간 점을 택하여 색깔을 check하고 이전과 먼저 두점중에 색깔이 다른 두점을 택하여 위 작업을 두 점이 모두 color-bou로 일치 될 때까지 반복한다.

발생하는 요소의 형태는 그림 2 에서와 같이 삼각형, 사각형, 오각형등 3가지 모양으로 경계요소를 얻을 수 있다. 4개의 절점의 색깔이 모두 color-in인 경우는 내부요소로 인식되어지며, 각요소의 데이터는 절점의 갯수와 절점의 위치가 structure로서 memory에 기억되다.

2.3 경계요소의 최적화

1) 삼각형 요소의 소거

2가지 방법으로 삼각형 요소는 소거된다. 이웃하는 요소가 삼각형, 경계점을 두개 갖는 사각형, 오각형일 경우 그림 3-A 에서와 같이 a와 b점에 해당하는 데이터점 c점의 데이터로 바꾸고 삼각형 요소의 데이터는 메모리에서 소거된다.

경계점이 1개인 사각요소가 이웃했을 때는 그림 3-B 에서와 같이 a점은 그대로 있고 a점의 새로운 절점을 사각형 요소에 추가해 오각형 요소가 된다.

2) 사각형 요소의 최적화

영역의 경계점(color-in)을 가진 요소의 각 절점간의 거리를 검토하여 임의의 한계보다 작은 경우는 두 점중 내부점을 경계점 위치로 이동시켜 절점이 2개인 경우는 그림 4-A 와

같이 소멸되고 절점이 3개로 변화 되는 경우는 그림 4-B 와 같이 변환후 위 1)의 작업을 계속한다.

3) 오각형 요소의 최적화

그림 5와 같이 b,c,d점에서 수선을 내려 5각형 요소는 절점 b' b' c' c'를 갖는 요소로 변환시키고 절점 c' c' d' d'를 갖는 새로운 요소를 새로이 생성한다.

3. 프로그램의 구성

3.1 그래픽 작업

Mouse를 사용한 그래픽 작업은 다음과 같은 Subroutine에 의하여 작업이 이루어 진다.

- Subroutine Initialize
그래픽 화면으로 초기화하며 화면의 최대, 최소의 범위틀 구한다.
- Subroutine Make-Screen
선택 Menu 등 그래픽에 필요한 화면을 구성한다.
- Subroutine Locator
mouse를 load시킨다.
- Subroutine Draw
Line, circle 등 각종 명령으로 도면을 그린다.
- Subroutine Save
작성된 도면을 save 한다.
- Subroutine Load
save 된 file을 load시킨다.
- Subroutine Fill-in
영역의 내부를 표시한다.
- Subroutine Fill-out
영역의 외부를 표시한다.
- Subroutine Mouse-set
Mouse의 작동을 읽는다.

3.2 영역의 분할작업

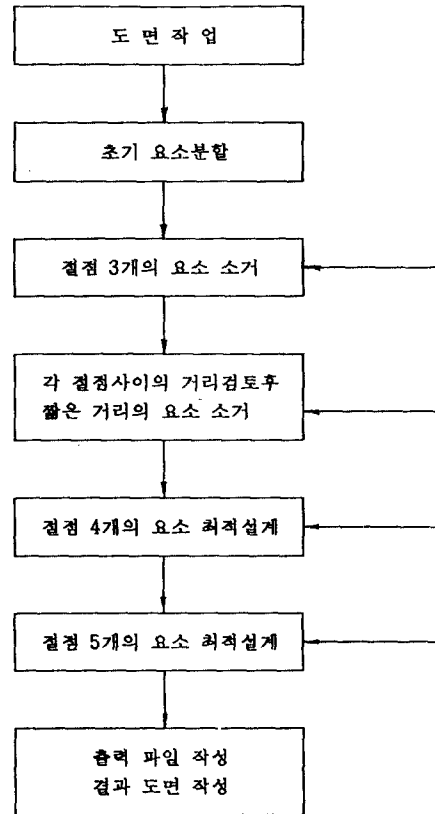
요소의 자동 분할은 아래와 같은 Subroutine 에 의하여 이루어 작업이 이루어 진다.

- Subroutine Mesh-Initial
주어진 요소의 크기로 초기요소를 분할한다.
- Subroutine Mesh-3node
절점 3개의 요소를 최적화 한다.
- Subroutine Mesh-4node
절점 4개의 요소를 최적화 한다.
- Subroutine Mesh-5node
절점 5개의 요소를 최적화 한다.
- Subroutine Node-Length-Check
각 요소의 절점 사이의 길이를 검토한다.
- Subroutine Node-Array
각 요소의 절점의 순서를 배열한다.
- Subroutine Change-XY
변화되는 절점의 값을 변화시킨다.
- Subroutine Intersection-Point
절점사이의 경계가 있는 경우 교점을 구한다.

- Subroutine Dwg

최종 결과를 화면에 표시한다.

3.3 프로그램 블록 선도



4. 결과 및 고찰

유한 요소법으로 해를 구하고자 하는 대상을 도면화 하면 또다른 input-data작성이 없이 주어진 화면 정보로써 요소를 분할하여 빠른 시간내에 해를 구할 수 있다.

그림 7,8,9에서 보는 바와 같이 주어진 영역이 free-curve 이거나 내부에 hole을 가진 경우라도 모두 사각형으로 요소 분할이 이루어 졌다. 분할되어진 요소중 한 절점에서 예각이 너무작은 요소는 좀더 정확한 해를 얻기위해 계속 연구되어야 한다.

참고문헌

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, (1983)
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)
6. Ben Ezzell, " Graphics Programming in Turbo c 2.0", Addison Wesley,(1989)

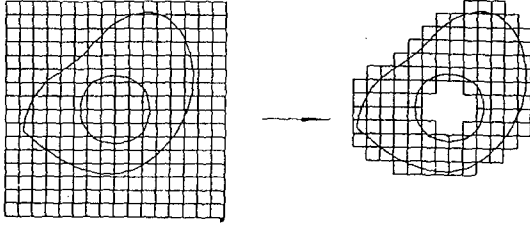
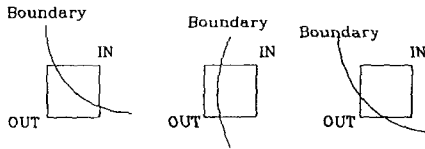
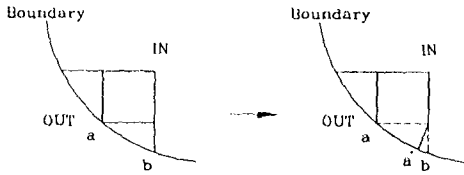


그림 1. 초기 Mesh Generation

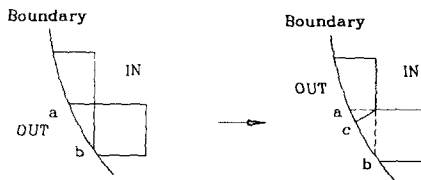


A.삼각형 경계요소 B.사각형 경계요소 C.오각형 경계요소

그림 2. 초기 생성된 요소형태

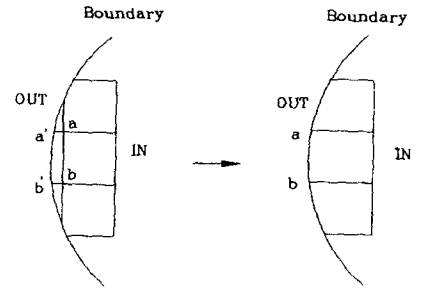


A. 경계점 1개의 사각형 요소의 소거

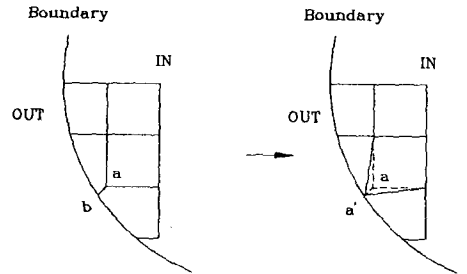


B. 삼각형,오각형,기타 사각형 요소가 이웃했을때

그림 3. 삼각형 요소의 소거



A. 두변의 길이가 짧은 경우



B. 한변의 길이가 짧은 경우

그림 4. 사각형 경계요소의 최적화

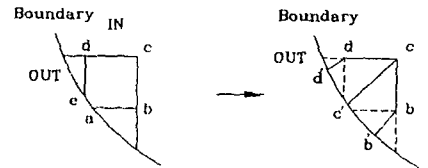
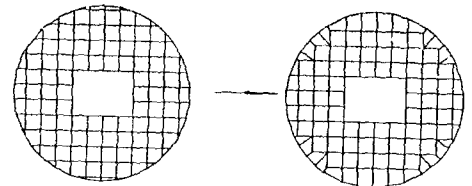
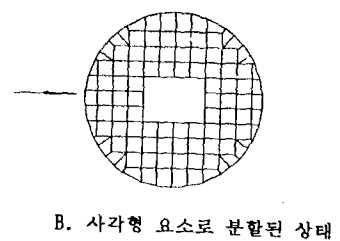


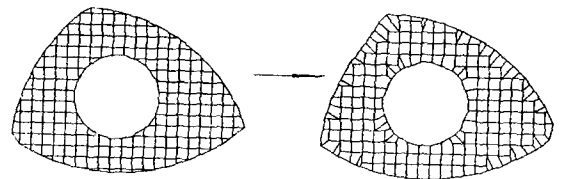
그림 5. 오각형 요소의 최적화



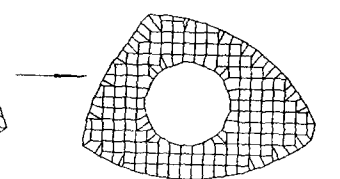
A. 초기 요소 분할 상태



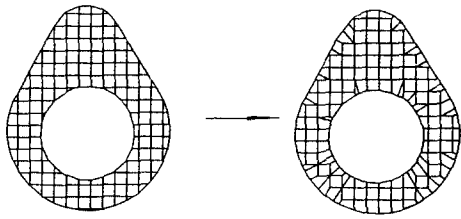
B. 사각형 요소로 분할된 상태



A. 초기 요소 분할 상태



B. 사각형 요소로 분할된 상태



A. 초기 요소 분할 상태 B. 사각형 요소로 분할된 상태

그림 6. 자동 요소 분할에 의해 분할된 상태