

〈論 文〉

삼차원 유한요소의 자동생성(2)

— 사면체 옥트리로부터의 유한요소 생성 —

정용호* · 이건우**

(1994년 6월 20일 접수)

Automatic Generation of 3-D Finite Element Meshes : Part(Ⅱ)

—Mesh Generation from Tetrahedron-based Octree—

Yoong-Ho Jung and Kunwoo Lee

Key Words : Tetrahedron(사면체), Octree, Automatic Generation(자동생성), Finite Element Mesh(유한요소), Boundary Moving(경계이동), Shape Value(형상계수)

Abstract

Given the tetrahedron-based octree approximation of a solid as described in part(Ⅰ) of this thesis, in this part(Ⅱ) a systematic procedure of 'boundary moving' is developed for the fully automatic generation of 3D finite element meshes. The algorithm moves some vertices of the octants near the boundary onto the exact surface of a solid without transforming the topology of octree leaf elements. As a result, the inner octree leaf elements can be used as exact tetrahedral finite element meshes. In addition, as a quality measure of a tetrahedral element, 'shape value' is proposed and used for the generation of better finite elements during the boundary moving process.

1. 서 론

본 논문의 (1)편에서 기술한 octree의 구성법은, 입체형상을 근사적으로 표현하는 방법으로써, CSG 방식을 사용하는 물체의 모델링(modeling),⁽¹⁾ 영상 처리(image processing),⁽²⁾ 광선투사(ray tracing), 이동중인 물체의 경로해석(path analysis) 등^(3,4) 다양한 응용분야를 갖는다. 이러한 octree 기법은 서로 다른 크기의 요소를 허용함으로써 효과적으로 입체를 나타낼 수 있는 장점을 갖고 있으므로, 이와 같은 특성은 유한요소의 자동생성에 이용될 수 있을 것이다. 즉 해석영역의 내부에는 큰 요소

를 허용하고 표면에서는 작은 요소를 자동으로 생성되게 할 수 있으므로, 해의 정확도 측면이나 경제적인 측면에서 유한요소 해법의 특성에 잘 부합되는 요소가 생성되게 할 수 있다.

그런데 주어진 입체형상을 근사화하는 octree의 요소들은 그 입체형상의 경계를 정확히 표현하지 못하므로 곧 바로 유한요소가 될 수는 없다. 이러한 문제를 해결하기 위해 육면체 octree를 사용하는 기존의 방법들^(5~11)에서는 여러가지 복잡한 과정을 수행하여 입체형상의 경계를 포함하는 육면체 요소를 사면체로 분할하여 입체의 경계를 정확히 나타내는 유한요소를 생성하도록 하였다. 그러나 이와 같은 방법은 각 육면체 단위로 수행되므로 서로 인접한 육면체 요소로부터 생성되는 유한요소가 서로 모순되는 경우도 발생할 수 있으므로 이를 해결하기 위한 별도의 과정이 필요하다.

*정희원, 삼성항공(주) 항공우주연구원

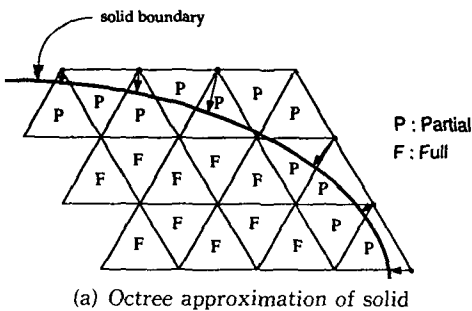
**정희원, 서울대학교 기계설계학과

그러나 본 논문의 (1)편에서 기술한 사면체에 근거한 octree로써 입체형상을 근사화하였을 경우에는, 해석하고자 하는 입체형상의 표면을 포함하는 요소의 몇몇 꼭지점들만 입체형상의 표면으로 이동하면 이웃하는 요소와의 적합성이 유지되는 유한요소가 생성될 수 있다. 이러한 이동 과정을 '경계이동'이라고 하겠는데, 본 논문의 (2)편에서는 이러한 '경계이동'을 체계적으로 수행하는 방법을 기술하고자 한다.

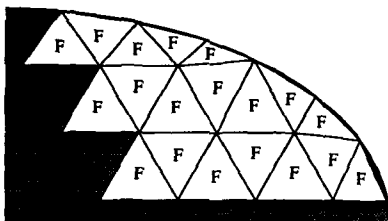
2. 경계이동

2.1 개요

본 논문의 (1)편에서 기술한 방법으로 입체형상을 근사화한 팔분체들은 다음과 같은 과정에 의해 정확한 유한요소가 될 수 있다. 즉, 2차원의 경우를 예를 들어 설명하면, Fig. 1(a)와 같이 입체형상을 근사화한 결과로서 부분요소(P)와 내포요소(F)가 있을 경우, 인접한 요소의 위상을 유지한 채로 부분요소의 적절한 꼭지점들을 화살표방향과 같이 입체형상의 경계로 이동시킨다. 그 다음 꼭지점의 위치이동의 결과로서 경계부분에 있는 요소들의 입체형상에 대한 포함관계가 변경될 수 있으므로 이를 재분류한다. 결과적으로 Fig. 1(b)와 같이, 생성된 모든 내포요소(F)만을 취하면 입체형상의 모든 영역을 정확히 나타내는 유한요소가 된다.



(a) Octree approximation of solid



(b) After boundary moving

Fig. 1 Boundary moving

본 논문의 (2)편에서는 이와 같은 '경계이동'의 체계적인 방법을 제안하는데, 이를 위해 먼저 다음과 같은 사항을 고려하였다.

첫째, 이동과정에서 빈번히 필요로 하는 입체형상과 octree의 단말요소와의 면/면 교차판정 과정과 사면체의 각 꼭지점에 대해 입체형상에 대한 내/외판정 및 이동과 관련한 요소들의 포함관계에 대한 판정을 용이하게 하고 소요되는 계산과정을 줄이기 위해, 본 논문 (1)편에서 octree생성을 위해 사용한 절면모델을 대상으로 경계이동을 수행한다. 이러한 절면모델에 대한 이동이 완료된 후, 이동된 각 절점들을 실제 입체형상의 정확한 표면으로 이동시킴으로써 해석 모델을 정확히 근사하는 유한요소가 되도록 하였다.

둘째, 절면모델로 근사화하였을 때, 입체형상의 원래 모서리를 이루는 절면의 모서리 외에도, 한 평면상에 놓이지 않는 서로 이웃한 절면과의 경계를 이루는 모서리도 절면모델을 나타내는 모서리로서의 역할을 수행한다. 따라서 이러한 선분의 집합을 따로 구성하여 2.4절의 모서리로의 이동시 고려하여야 한다. 이에 대한 보다 상세한 설명은 2.2절의 절면선분의 구성과 2.3절의 모서리로의 이동에서 언급하겠다.

셋째, 경계이동과정은 입체형상의 표면을 포함하는 부분인 요소에 대해 수행하므로 이를 효율적으로 수행하기 위해, 먼저 논문의 (1)편에서 생성된 octree의 단말요소들 중 부분요소들만의 집합을 구성하여 이를 Oct-If 리스트라고 하고 이 집합에 대해 경계이동을 수행한다. 경계이동 과정을 수행하는 동안 생성되거나 소멸되는 부분요소를 Oct-If 리스트에 삽입 또는 제거하기 용이하도록 Oct-If 리스트는 Fig. 2와 같이 연결리스트(linked list)의 구조를 갖도록 하였다.

경계이동의 전체과정은 Fig. 3과 같이 수행된다. 즉, 먼저 Oct-If 리스트에 저장되어 있는 부분요소의 개수(PaCnt)를 구한다. 그 다음, 절면 선분을 구성하고 (MK-SEG), 다음의 각 소절에서 설명될 단계들을 순차적으로 수행하게 되는데, 부분요소가 포함하는 물체의 경계면의 절면의 개수가 큰 경우부터 작아지는 경우의 순서로 수행한다. 즉, 꼭지

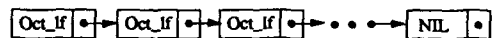


Fig. 2 Linked list of Oct-If

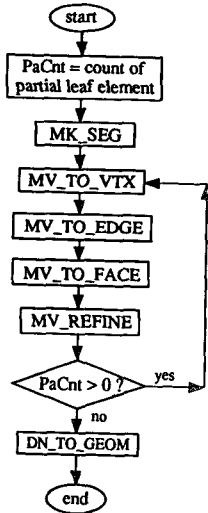


Fig. 3 Overall procedure of boundary moving

점으로의 이동(MV-TO-VTX) 단계에서는 부분요소가 절면의 면을 3개 이상 만나는 경우를 처리하게 하고 모서리로의 이동(MV-TO-EDGE) 단계에서는 2개의 절면과 만나는 경우를 처리하게 한다. 그리고 면으로의 이동(MV-TO-FACE) 단계에서는 부분요소가 절면의 면을 단 하나 포함하는 경우를 처리하게 한다. 마지막으로 세분화 이동(MV-REFINE) 단계에서는 앞의 세단계에서 이동할 수 없었던 부분요소에 대한 처리를 수행하게 된다.

각 단계에서 생성되거나 소멸되는 부분요소는 Oct-If 리스트에 첨삭되고 PaCnt를 다시 구한다. 이러한 과정을 Oct-If 리스트에 부분요소가 하나도 남지 않을 때까지 반복하여 수행한다.

각 단계에서, 팔분체의 한 꼭지점이 이동되면 그 꼭지점을 공유하는 모든 팔분체도 따라서 변하게 되는데, 이동되는 꼭지점을 공유하는 모든 요소는 그 꼭지점의 좌표를 참조하는, 본 논문의 (2)편에서 기술한, Node의 mvpt를 의미하고 이와 같은 과정은 자동으로 이루어진다. 각 단계에 대한 보다 상세한 설명은 다음과 같다.

2.2 절면선분의 구성(MK_SEG)

본 논문에서 제안되는 방법은 평면뿐만 아니라 곡면도 포함할 수 있는 입체형상을 대상으로 하고 있으므로, 전술한 바와 같이 사면체 octree의 형성 과정과 경계이동을 수행하는 과정에서 계산상의 효율을 위해 절면모델을 사용하도록 하였다.

솔리드 모델링 시스템으로 생성한 입체형상의 모서리는 두 개의 면이 만나는 곡선으로 정의할 수 있겠는데, 절면모델로 변환되면 연속한 곡면이 유한개의 다면체로 표현되므로 한 평면에 놓일 수 없는 서로 이웃하는 절면들의 모든 경계가 모서리가 된다. 2.4절에서 설명될 모서리로의 이동은 이러한 선분들 모두를 대상으로 하여야 한다. 그런데 경계 이동 과정은 Oct-If 리스트를 계속 순환하면서 탐색하도록 하므로 모서리로의 이동단계에서 매번 이러한 선분을 찾는다는 것은 비효율적이다. 따라서 모서리로의 이동을 수행하기 전에 이러한 선분들만의 집합을 미리 구성해 두고 이에 대해 적용하는 것이 보다 효율적일 것이다. 이를 위해 아래와 같이 C언어로 표현된 자료구조로써 이러한 선분들의 집합을 구성하였다.

```
struct Seg-Ln
```

```
{ Integer    onflag ;
  Point     start ;
  Point     end ;
  Edge      *ed ;
  Face      *fc ;
  struct Seg-Ln *next ; }
```

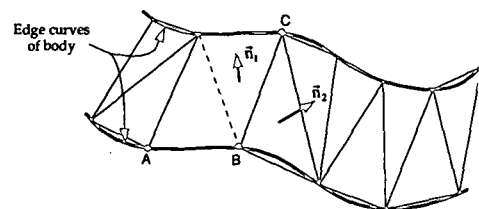
여기서, Seg-Ln의 각 항목에 대한 설명은 다음과 같다.

onflag : 해당 선분이 입체형상의 실제 모서리로부터 생성되었는지, 아니면 절면의 윤곽을 나타내는 모서리인지를 나타낸다.

start : 절면선분의 시작점의 좌표값을 저장한다.

end : 절면선분의 끝점의 좌표값을 저장한다.

ed : Fig. 4의 선분 AB와 같이 절면선분이 실제 모서리로부터 생성되었을 때, 실제 모서리에 관한 정보를 참조할 수 있는 지시인자이다.



— : Segment line

--- : Boundary between two facets on a plane

\vec{n}_1 : normal vector of facet i

Fig. 4 Illustration of segment lines

fc: Fig. 4의 선분 BC와 같이 절면선분이 입체형상의 실제 모서리를 나타내지 않고 절면의 윤곽을 나타내는 모서리일 때, 절면이 근사하는 원래 입체형상의 면을 참조할 수 있는 지시인자이다.

next: 연결리스트로 만들기 위한 지시인자이다.

한편 이와 같은 자료구조에 절면선분의 집합을 구성하는 과정은 다음과 같이 이루어진다. 즉, 절면모델의 모든 절면을 탐색하면서 각 절면의 세개의 모서리가 먼저 입체형상의 모서리 상에 존재하는지를 검사한다. 만약 Fig. 4의 A, B를 잇는 선분과 같이 모서리 상의 선분이고 절면선분의 집합에 이미 등록되어 있지 않으면 이를 등록하고 그 모서리로의 지시인자를 저장한다. 만약 Fig. 4의 B, C를 잇는 선분처럼 모서리 상의 선분이 아니면 그 선분을 공유하는 이웃 절면들이 한 평면상에 놓일 수 있는지를 조사한다. 즉, 그 선분이 절면 모델의 윤곽으로 작용하고 절면선분의 집합에 등록되어 있지 않으면, 이를 절면선분의 집합에 삽입하고 그 절면이 근사하고 있는 입체형상의 면으로 지시인자를 저장한다. 이와 같은 과정을 모든 절면에 대해 수행하여 절면선분의 집합을 구성한다.

2.3 꼭지점으로의 이동(MV.TO.VTX)

본 단계에서는 부분요소가 물체의 꼭지점을 포함하는 경우에 대한 처리 과정이다. 이를 위해 먼저 Oct-If 리스트를 탐색하여 입체형상의 꼭지점(이하 SV)을 포함하는 부분요소를 찾아낸다. 각 부분팔분체가 SV를 포함하는지에 대한 계산은 본 논문 (1)편의 4.4절에서 소개한 꼭지점의 내/외 판정에서 사용한 방법을 이용한다. 즉, Fig. 5와 같이 SV에서 임의의 방향으로 충분한 길이의 직선을 생성하고, 이 직선과 그 팔분체의 4개의 면과의 교점의 갯수를 구하고 이 개수가 홀수이면 그 팔분체는 SV를 포함하는 것으로 판정한다. 직선과 팔분체 면과의 교차여부는 본 논문 (1)편의 4.3절의 면/면 교차판정에서 설명된 벡터선분과 절면과의 교차 문제로 구한다.

그 다음, 찾아진 부분요소에 대해 SV가 존재하는 위치에 따라 다음의 해당되는 경우가 찾아진다.

- (a) 꼭지점이 팔분체 모서리에 존재하는 경우(Fig. 6(a))
- (b) 꼭지점이 팔분체 면에 존재하는 경우(Fig. 6(b))
- (c) 꼭지점이 팔분체 내부에 존재하는 경우(Fig. 6(c))

(a)의 경우, SV와 접하는 팔분체의 모서리의 양 끝점 중 하나가 이동될 수 있는데, 한 꼭지점이 이

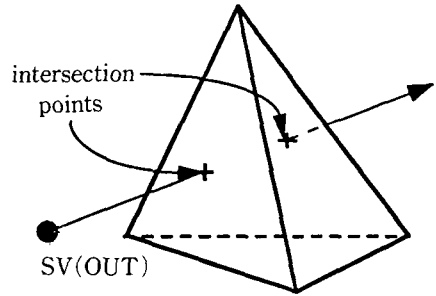
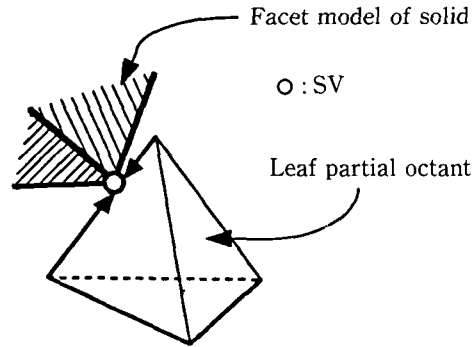
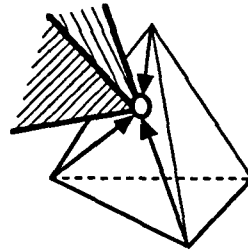


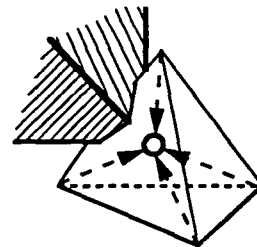
Fig. 5 IN/OUT test of a SV SV(OUT)



(a) SV(solid vertex) on edge of octant



(b) SV on face of octant



(c) SV in octant

Fig. 6 Cases where MV·TO·VTX is applied

동될 때 형성되는 요소들의 형상이 보다 좋아지는 순서로 이동될 우선순위가 결정된다. (b)와 (c)의 경우도 마찬가지로, 팔분체의 적절한 꼭지점들의

이동될 우선순위가 정해진다. 단 차이점은 Fig. 6(b), (c)와 같이 선택될 수 있는 꼭지점들이 (a)의 경우와 다르다는 것이다. Fig. 6에서 화살표는 이동될 수 있는 꼭지점과 그 이동 방향을 나타낸다.

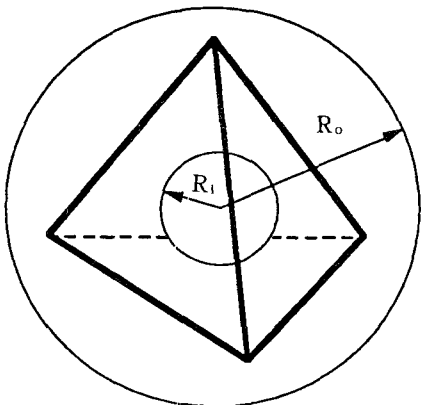
여기서, 좋은 형상의 요소란 정사면체에 가까운 형상으로 정의하고, 본 논문에서는 사면체의 왜곡 정도를 판단하기 위해, 사면체의 내접구와 외접구의 반경의 비를 형상계수(shape value)로 사용할 것을 제안하였다.(Fig. 7) 이 비값은 정사면체일 때 3이므로, 식(1)과 같이 3으로 정상화(normalize)한 값의 역수를 형상계수값으로 정의하여 사용하였다. 따라서, 사면체 요소는 정사면체로부터 왜곡될수록 1보다 점점 더 작은 값을 갖는다.

$$S = \left[\frac{R_o/R_i}{3} \right]^{-1} \quad (1)$$

여기서,

- S : 사면체의 형상계수
- R_o : 사면체의 외접구의 반경
- R_i : 사면체의 내접구의 반경
- 3 : 정상화계수(normalizing factor),

따라서 이동될 수 있는 팔분체의 각 꼭지점에 대해, 입체형상의 꼭지점으로 이동시켰다고 가정했을 때, 이동되는 점을 공유하는 모든 팔분체의 형상계수를 합한 수치가 큰 순서로 우선순위가 정해진다. 그 다음, 우선순위에 따라 선택된 부분요소의 꼭지점이 이동된 후 관련된 모든 요소들의 위상의 변화가 있는지를 검사한다. 위상의 변화를 감지하는 방법은 2.6절에서 설명하겠다. 만약 위상의 변화가



R_o : Radius of the smallest sphere enclosing tetrahedron
 R_i : Radius of inner largest sphere

Fig. 7 Shape value

감지되면 선택된 꼭지점의 이동을 포기하고, 그 다음 우선순위의 이동 가능한 꼭지점에 대해 같은 방법으로 검사한다. 만약 이동으로 인한 관련 요소의 위상의 변화가 없으면 이동을 실행한다. 이때 이동되는 꼭지점을 공유하는 모든 요소들 중 octree에서 다른 계층수(level)를 가진 요소가 존재하면 이들이 모두 같은 계층수를 갖도록 세분화한 후 이동한다. 실제로 프로그램 내부에서 꼭지점의 이동이란 그 꼭지점에 해당하는 절점, 즉 본 논문의 (1)편에서 기술한 자료구조의 Node에 새로운 좌표값을 부여하고 관련된 정보들을 부여함으로써 이루어진다. 이동 후 관련된 각 요소는, 입체형상에 대한 포함관계를 조사하여, 부분이면 Oct-If 리스트에 삽입하고 그렇지 않으면 Oct-If 리스트에서 제거한 다음 PaCnt를 다시 구한다.

이상과 같이 Oct-If 리스트를 탐색하면서 수행되는 이 과정은 해당하는 경우가 발생하지 않을 때까지 순환하므로 Fig. 8과 같이 부분인 요소가 둘 이상의 SV를 포함하는 경우는, 각 SV에 대해 순차적으로 처리된다. 이에 대한 전체적인 흐름도가 Fig. 9에 소개되어 있다.

2.4 모서리로의 이동(MV.TO.EDGE)

본 단계에서는 Fig. 10과 같이 부분인 팔분체가 모서리 역할을 수행하는 절면의 모서리(이하 SE)를 포함하거나 접촉하는 경우에 대한 처리 과정이다. 모서리 역할을 수행하는 SE의 집합은 2.2절에서 이미 구성되었다.

이 경우에도 Oct-If 리스트를 탐색하면서, 부분 요소가 SE를 포함하거나 접촉할 가능성이 있는 경우를 찾아낸다. 이를 위해 부분요소를 포함하는 최소한의 상자와 SE를 포함하는 최소한의 상자가 겹치는지를 먼저 검사하게 된다. 만약 두 상자가 겹치고 찾아진 부분요소가 SE의 끝점 중의 하나를

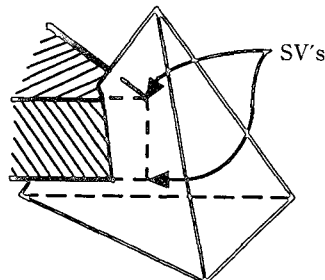


Fig. 8 Octant enclosing more than two SV's

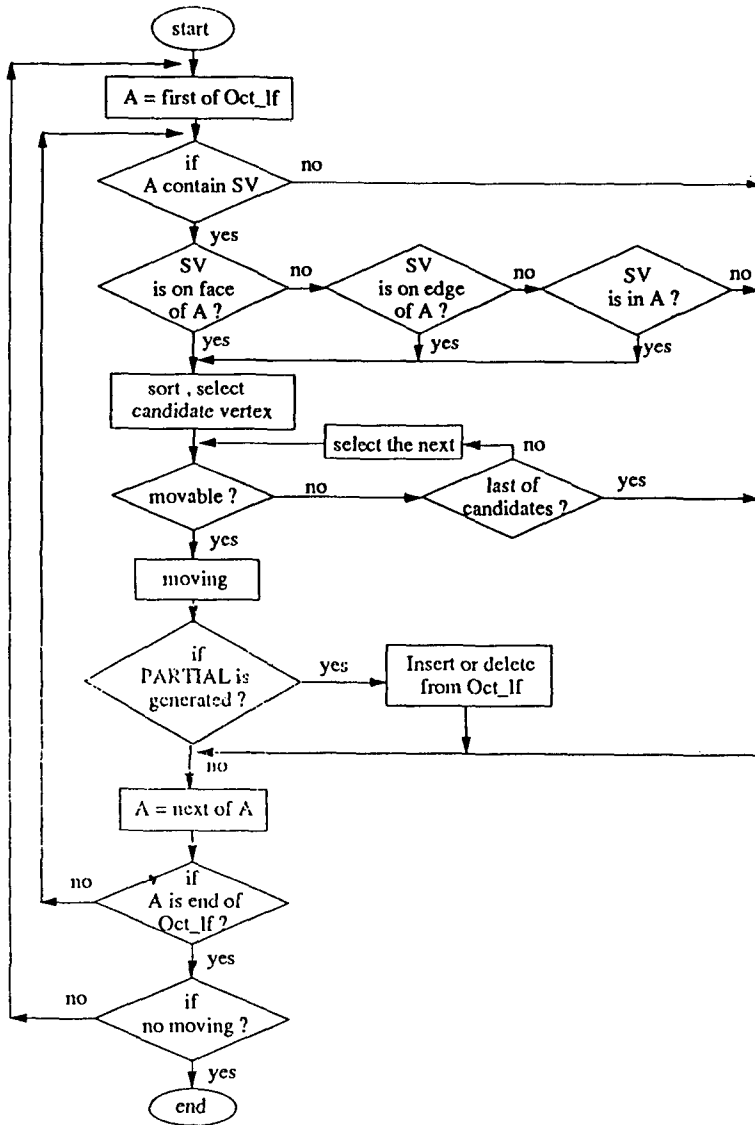


Fig. 9 Algorithm of MV_TO_VTX

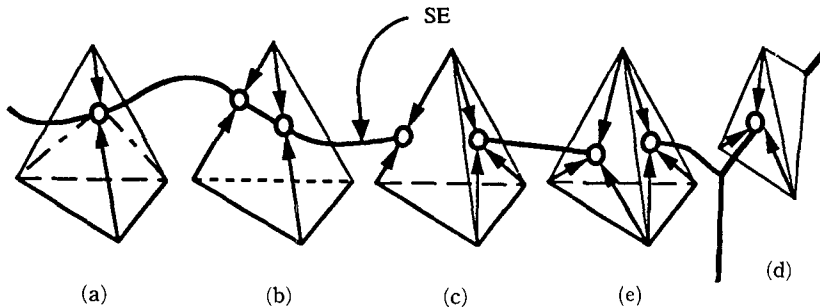


Fig. 10 Cases where MV TO EDGE is applied

포함하고 있으면 이는 2.3절의 꼭지점으로의 이동에 해당하는 경우와 같은 경우이다. 따라서 이 경우에는 꼭지점으로의 이동에서 처리한 것과 같은 과정이 이루어진다. SE의 양 끝점이 부분요소에 포함되지 않고 SE가 부분요소를 지나는 경우는 다음의 다섯가지로 분류된다.

(a) SE가 팔분체의 한 모서리와 교차하는 경우 (Fig. 10(a))

(b) SE가 팔분체의 두 모서리와 교차하는 경우 (Fig. 10(b))

(c) SE가 팔분체의 한 모서리를 지나고 한 면을 관통하는 경우(Fig. 10(c))

(d) SE가 팔분체의 한 면을 관통하는 경우(Fig. 10(d))

(e) SE가 팔분체의 두 면을 관통하는 경우(Fig. 10(e))

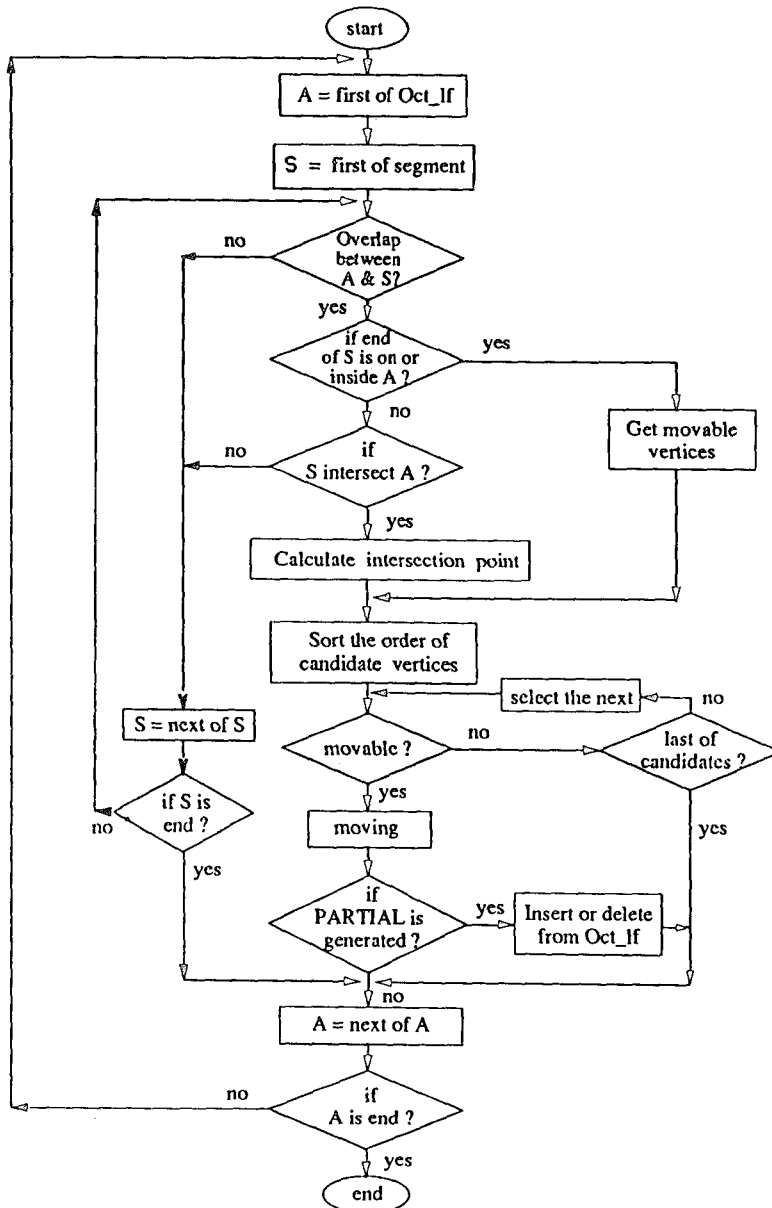


Fig. 11 Algorithm of MV TO EDGE

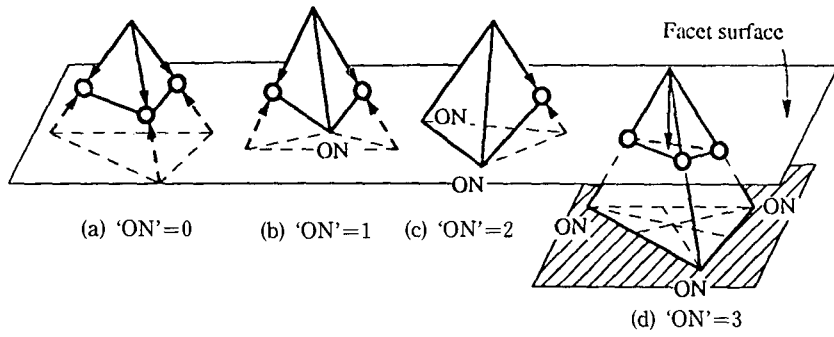


Fig. 12 Cases where MV TO FACE is applied

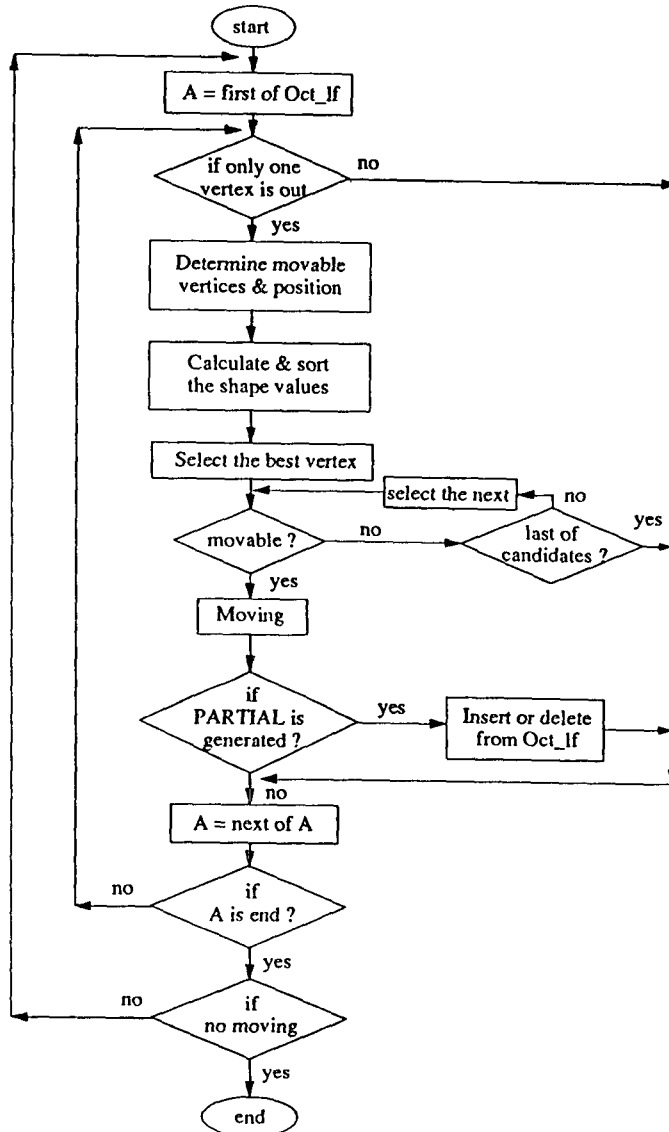


Fig. 13 Algorithm of MV TO FACE

(a)의 경우, SE와 교차하는 팔분체의 모서리의 양끝점 중 하나가 이동될 수 있는데, 그 꼭지점의 이동에 따라 형성되는 요소들의 형상이 보다 좋아지는 순서로 이동될 우선순위가 결정된다. 나머지의 경우도 마찬가지로, 팔분체의 이동될 적절한 꼭지점들의 우선순위가 정해진다. Fig. 10에서 화살표는 이동될 수 있는 꼭지점과 이동될 위치를 나타낸다.

우선순위의 결정은 앞의 단계와 마찬가지로 형상 계수값에 근거하여 정해진다. 그 다음, 우선순위에 따라 선택된 부분요소의 꼭지점을 입체형상의 꼭지점으로 이동시켰다고 가정했을 때, 관련되는 모든 요소들의 위상의 변화가 있는지를 검사한다. 만약 이동으로 인한 관련 요소의 위상의 변화가 없으면 이동을 실행한다. 이때도 이동되는 꼭지점을 공유하는 모든 요소들 중 octree에서 다른 계층수를 가진 요소가 존재하면 이들이 모두 같은 계층수를 갖도록 세분화한 후 이동한다.

실제로 프로그램 내부에서 이동은 앞의 단계와 마찬가지로 그 꼭지점에 해당하는 절점, 즉 Node에 새로운 좌표를 부여하고 관련된 정보들을 부여함으로써 이루어진다. 단, 앞의 단계와 다른 점은 SE가 입체형상의 모서리로부터 생성되었으면 이동된 꼭지점의 자료구조인 Node의 onEd는 입체형상의 모서리에 대한 지시인자를 저장하고, 만약 SE가 입체형상의 모서리가 아닌 절면간의 경계로부터 생성되었으면 이동된 꼭지점의 자료구조인 Node의 onFc는 입체형상의 면에 대한 지시인자를 저장한다. 이는 2.7절에서 설명될 요소절점의 위치 단계를 위해 필요하다.

이동 후 관련된 각 요소는, 입체형상에 대한 포함 관계를 조사하여, 부분이면 Oct-If 리스트에 삽입하고 그렇지 않으면 Oct-If 리스트에서 제거한 다음 PaCnt를 다시 구한다.

이상과 같이 Oct-If 리스트를 탐색하면서 수행되는 이 과정은 해당하는 경우가 발생하지 않을 때까지 순환한다. 본 단계에 대한 전체적인 흐름도가 Fig. 11에 소개되어 있다.

2.5 면으로의 이동(MV.TO.FACE)

꼭지점으로의 이동과 모서리로의 이동이 수행된 후 Oct-If 리스트에 잔존하는 팔분체는, 절면모델의 한 면과 교차하는 것들이다. 왜냐하면, 절면모델의 3개 이상의 면이 교차하는 점을 포함하는 팔

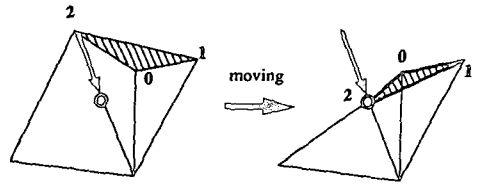


Fig. 14 Example of distorted mesh

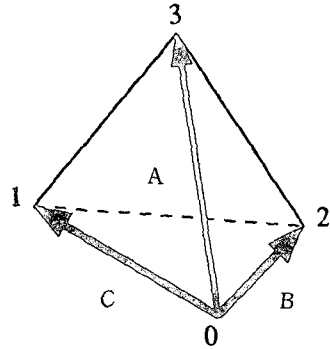


Fig. 15 Vectors used in detecting distorted elements

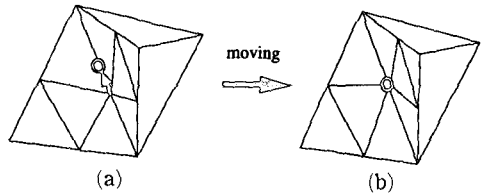
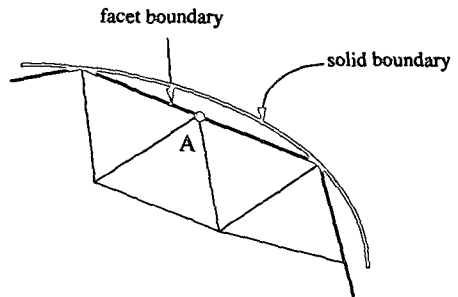
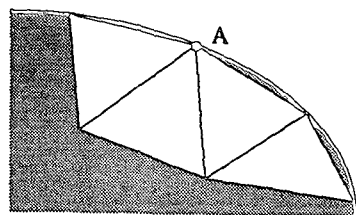


Fig. 16 Moving after refinement



(a) Before moving to exact geometry



(b) After moving to exact geometry

Fig. 17 Relocation of nodes onto exact geometry

분체는 꼭지점으로의 이동 단계에서 제거되었고, 2개의 면이 교차하는 상태를 포함하는 팔분체는 모서리로의 이동단계에서 이미 제거되었기 때문이다. 따라서 나머지 부분요소와 절면모델과의 관계는 부분요소가 포함하는 절면모델의 면(facet surface, 이하 FS)과의 관계에 따라 다음과 같이 네 가지로 분류된다.

(a) 팔분체의 꼭지점이 모두 FS에 존재하지 않는 경우(Fig. 12(a))

(b) 팔분체의 하나의 꼭지점이 FS에 존재하는 경우(Fig. 12(b))

(c) 팔분체의 두 개의 꼭지점이 FS에 존재하는 경우(Fig. 12(c))

(d) 팔분체의 세 개의 꼭지점이 FS에 존재하지만 부분인 경우(Fig. 12(d)). 즉, 절면모델의 표면에 있는 꼭지점이, 실제로 팔분체와 절면모델과 교차하는 FS에 존재하지 않고 다른 면에 존재하는 경우이다.

전술한 바와 마찬가지로, 꼭지점의 이동 가능 위치는 Fig. 12에 화살표로 표시하였다. 어느 꼭지점을 이동시킬 것인가는 앞의 단계들과 마찬가지로 형상계수값에 따라 우선순위가 결정된다. Fig. 12(d)의 경우, FS에 존재하는 3개의 꼭지점의 무게 중심과 FS에 있지 않은 꼭지점을 잇는 선분을 생성한다. 이 선분과 실제로 팔분체와 절면모델이 교차하는 FS와의 교점을 구하고, 표면에 있지 않은 꼭지점을 구해진 교점으로 이동시킨다. 이동 단계에서 처리되는 과정은 앞의 단계들과 마찬가지로, 단 이동되는 꼭지점에 해당하는 Node의 onFc는 FS가 근사하는 원래 입체형상의 면에 대한 지시인자를 갖는다.

본 단계는 앞의 단계들과 마찬가지로 상기의 네 가지 경우에 해당되는 팔분체가 모두 제거될 때까지 Oct-If 리스트를 탐색하는 과정이 반복된다. 또한 이동 후 부분 팔분체가 새로 생성되거나 소멸될 때마다 Oct-If 리스트는 이를 반영하게 된다. 본 단계에 대한 전반적인 과정은 Fig. 13에 소개되어 있다.

2.6 세분화 이동(MV.REFINE)

본 논문의 (1)편에서 정의한 사면체의 꼭지점의 번호체계에 부합하는 요소를 정상(normal) 요소라고 하고, 그렇지 않은 요소를 비정상(abnormal) 요소라고 정의한다. 따라서 octree생성과정 중 발

생한 모든 팔분체는 정상의 위상을 유지하고 있으므로, 유한요소 해법에서 일관된 유한요소로 허용된다.

그런데 경계이동의 과정에서 꼭지점의 이동 거리가 너무 큰 경우 비정상요소가 생성될 수 있다. 이와 같은 경우를 2차원 문제에 대해 Fig. 14에 나타내었다. 즉, Fig. 14에서 2번 꼭지점을 이동하면 빗금친 요소의 꼭지점의 순서가 뒤바뀌어, 결과적으로 중첩된 영역의 요소가 생성되는 경우를 보여준다. 이는 유한요소 해법에서 허용되지 않는 요소이다. 이와 같은 요소를 검출하기 위한 방법으로서

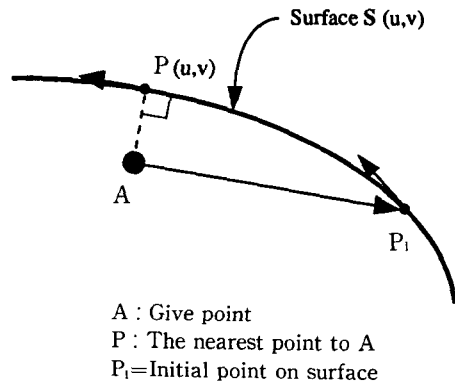


Fig. 18 Calculation of the nearest position on a surface to a given point

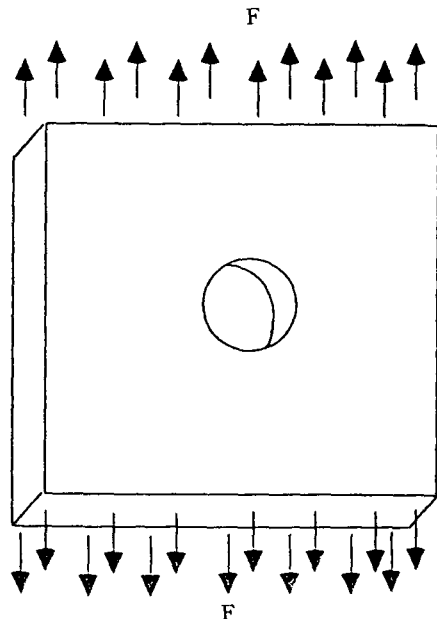


Fig. 19 Plate with hole in tension

해당 요소의 체적의 부호를 조사한다. 즉, 본 논문의 (1)편에서 정의한 사면체의 번호체계를 만족하는 '정상'요소의 꼭지점의 순서에 따라 Fig. 15와 같은 벡터를 구성한 다음, 이로부터 식(2)를 계산하여 요소의 체적을 계산하는데 정상요소의 경우 계산된 체적이 양의 값을 갖지만 비정상요소의 경우 체적은 음의 값을 갖는다.

$$volume = \frac{1}{6} \vec{A} \cdot (\vec{B} \times \vec{C}) \quad (2)$$

만약 전술한 경계이동의 각 단계에서, 해당되는 부분요소의 꼭지점들 중 이동이 고려되는 꼭지점들 모두가, 이동으로 인하여 비정상요소를 발생시키는 경우라면 그 부분요소에 대해서는 이동을 수행할 수 없다. 따라서 이와 같은 경우에는 이동할 꼭지점을 포함하는 모든 요소를 세분화한 후, 전술한 이동 과정들을 순차적으로 적용하도록 한다. 예를 들어 Fig. 14와 같은 경우에는, 2번 꼭지점과 관련된 모든 요소를 Fig. 16(a)와 같이 세분화한 후, 경계이동 과정중의 해당되는 경우에 따라 Fig. 16(b)와 같이 이동이 수행된다.

2.7 요소 절점의 위치(DN.TO.GEOM)

Oct-If 리스트에 부분요소가 하나도 남지 않을 때까지 전술한 단계들을 순차적으로 수행하게 되면, 최종적으로 남게 되는 octree의 단말요소들 중 내포인 요소들의 집합은 유한요소가 갖추어야 할 최소한의 조건을 만족하는 집합이다. 그러나 이러한 경계이동으로 형성된 유한요소는 삼차원 모델링 시스템이 생성한 입체형상으로부터 변환된 절면모델에 대한 것이다. 절면모델은 형성 과정에서 입체형상을 삼각형의 절면으로 근사화한 것이므로 입체형상의 표면이 평면이 아닌 경우, 실제 입체형상과는 간격이 존재할 수 있다. 따라서 경계이동 단계에서 절면모델의 경계로 이동된 각 요소의 절점들을 실제 입체형상의 경계면 상에 위치하도록 하여 얻어진 유한요소가 실제의 정확한 입체형상을 표현하도록 할 필요가 있다. 이 작업은 전술한 대로 경계이동의 각 단계에서 절면모델로 이동된 요소의 꼭지점들이 이동된 좌표값뿐 아니라 해당된 절면이 생성된 입체형상의 기하학적 정보에 대한 지시인자도 함께 갖기 때문에 가능하다. 즉, Fig. 17(a)에서와 같이 만약 요소의 꼭지점 A가 절면선분으로 이동된 경우, 그 절면선분이 입체형상의 모서리를 근

사하는 것이면, A점으로부터 그 절면선분이 근사하는 입체형상의 모서리 상의 최단 거리에 있는 점으로 이동한다. 만약 이동된 절면선분이 입체형상의 모서리가 아닌 절면간의 경계로부터 생성되었다면 그 절면선분이 근사하는 원래 입체형상의 면으로 이동된다.

주어진 점으로부터 최단거리에 있는 곡면상의 점은 아래의 비선형방정식 (3)을 풀어 구한다. 즉, Fig. 18에서와 같이 주어진 꼭지점 A로부터 곡면 S상의 최단거리에 있는 점 P를 구하기 위해, S곡면 상에 존재하면서 A점의 근처에 있는 임의의 점 P_i 를 취한다. 그 다음, P_i 에서 출발하여 A점에서 곡면상의 점까지의 벡터와 그 점에서 각 매개변수 방향으로의 접선 벡터와의 내적이 0인 조건에 해당하는 점을 Newton-Raphson방법으로 찾아간다.

이러한 이동과정은 새롭게 생성되거나 소멸되는 요소를 발생시키지 않고 단지 요소의 절점의 좌표를 정확히 위치시키는 효과가 있다.

$$f_1 = \{P(u, v) - A\} \cdot \frac{dS(u, v)}{du} = 0$$

$$f_2 = \{P(u, v) - A\} \cdot \frac{dS(u, v)}{dv} = 0 \quad (3)$$

3. 적용 예

3차원 입체형상에 대해 본 논문의 (1)편에서 기술한 사면체 octree구성법으로 입체형상을 근사화하였을 때, 이로부터 본 논문의 (2)편에서 기술한 경계이동 방법에 의해 자동으로 유한요소가 생성되는 예를 보이겠다. 예제의 적용은 10Mips(personal iris: Silicon Graphics사)의 시스템에서 수행되었다.

먼저 논문의 (1)편에서 고려한, 인장하중을 받는 얇은 평판에 구멍이 있는 경우(Fig. 19)를 해석하는 문제를 고려하자. 이 문제는 대칭이므로 그 4분의 1만을 취한 해석모델에 대해 Fig. 20과 같이 사면체 octree로 근사화하였을 때, 이에 대해 경계이동을 수행하여 자동생성된 유한요소의 결과를 Fig. 21에 보였다. 이 경우 경계이동을 수행하는 데는 약 10초가 소요되었으며, 생성된 요소 중 가장 형상이 조악한 경우는 형상계수치가 0.4를 나타내었다.

두번째 예는 Fig. 22와 같이 육면체의 블럭위에 실린더 형상이 비대칭의 위치에 놓인 입체형상의 경우이다. 이에 대해 사면체 octree로 근사화한 결

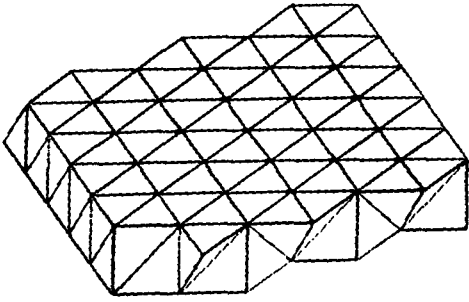


Fig. 20 Tetrahedron-based octree representation of quadrant model

과가 Fig. 23과 같이 주어졌을 때, 이를 ‘경계이동’하여 생성된 유한요소를 Fig. 24에 보였다. 이 경우 경계이동을 수행하는데는 약 78초가 소요되었다.

그 다음의 예는 기계 시스템의 대표적인 형상이라고 할 수 있는 connecting rod의 해석 모델에 대해(Fig. 25), 논문 (1)편에서 제안된 사면체 octree

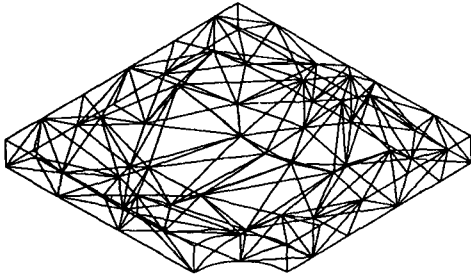


Fig. 21 Generated mesh of plate problem

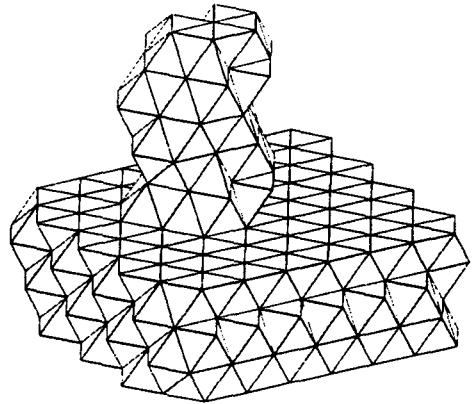


Fig. 23 Octree representation of the model

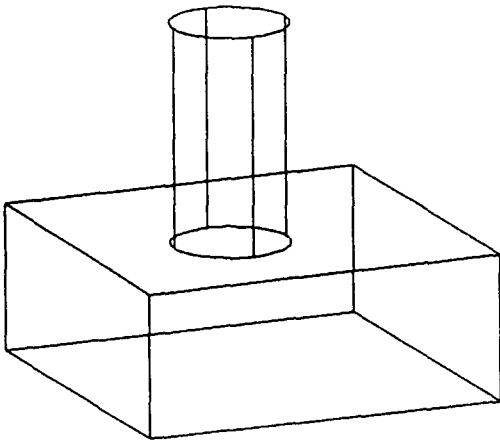


Fig. 22 Test model of cylinder on block

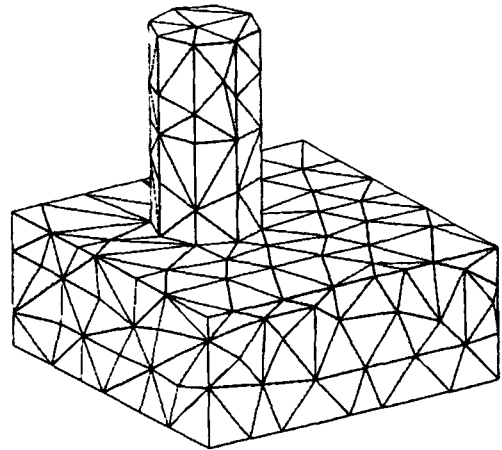


Fig. 24 Result of mesh generation

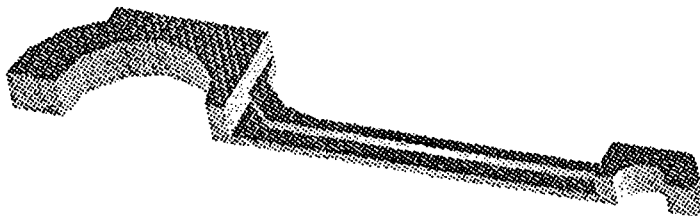


Fig. 25 3-D model of connecting rod

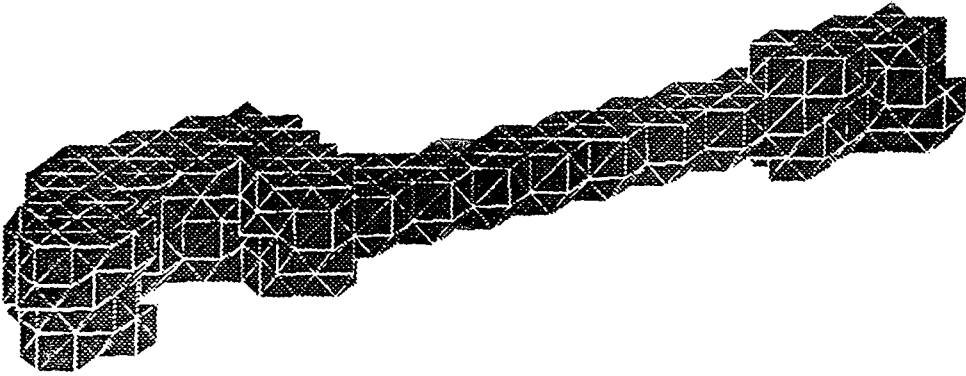


Fig. 26 Octree representation of the half model

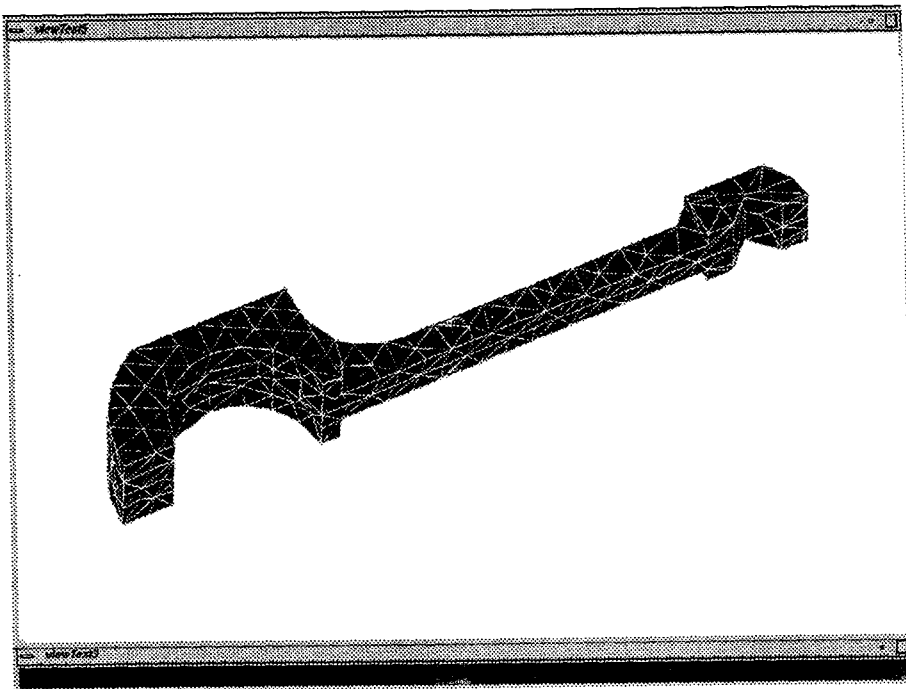


Fig. 27 Result of mesh generation for connecting rod

로써 근사화하였을 때(Fig. 26), 이에 대해 경계이동 과정을 수행한 결과로서 생성된 유한요소를 Fig. 27에 보였다. 이 경우 경계이동을 수행하는데 약 45분이 소요되었으며, 생성된 요소중 가장 형상이 조악한 경우는 형상계수치가 0.2를 나타내었다. 이상의 결과들은 모두 요소 평활화과정(smoothing)을 수행하기 전의 결과이므로 각 요소의 형상에 대한 개선의 여지를 갖고 있지만 초기 요소망으로서 는 비교적 좋은 요소가 생성되었다.

4. 결 론

본 연구에서는 3차원 물체를 유한요소 해법으로 해석하고자 할 때, 필요한 정보를 자동으로 생성하기 위해, 논문의 (1)편에서는 입체형상을 근사화하는 사면체 octree생성 방법을 새롭게 개발하였으며, 논문의 (2)편에서는 이로부터 사면체 유한요소를 생성하기 위한 체계적인 경계이동 방법을 제안

하였다. 논문의 (1), (2)편을 통하여 제안된 삼차원 물체에 대한 유한요소의 자동 생성 방법은 다음과 같은 장점을 갖는다.

(1) 기존의 육면체 octree기법의 단점인 육면체의 팔분체로부터 사면체의 유한요소로 변환하기 위해 복잡한 과정을 수행할 필요가 없다. 실제로 이 문제를 해결하기 위해 S-Y방법⁽⁵⁻¹¹⁾에서는 많은 수의 경우(예를 들면, 4096가지⁽¹²⁾)를 고려하거나 Wordenweber의 요소추출법(element extractor)⁽¹³⁾을 사용하였다. P-S-K방법⁽⁸⁾에서는 7종류의 형틀과 Wordenweber의 요소추출법을 사용하였다. 또 다른 방법^(9,10)에서는 Watson의 Delaunay 방법⁽¹⁴⁾을 사용하기도 하였다.

(2) 본 연구에서 제안된 사면체 octree로부터 생성되는 팔분체의 위상은 최종적인 유한요소의 위상과 같으므로, 육면체 octree를 사용하는 다른 방법들과는 달리 논문 (1)편의 Fig. 2에서 보인 것과 같은 인접한 요소 사이의 모순이 자동으로 제외된다.

(3) 해의 정확도를 위해, 국부적으로 원하는 곳에 유한요소를 세분화하는 것이 간단하여 적응 요소 생성 기능을 갖기가 용이하다. 즉, 생성되는 유한 요소의 위상이 최초의 뿌리 사면체의 위상과 동일하므로, octree를 형성하는 동일한 알고리즘으로 원하는 유한요소를 계속 세분화할 수 있다. 따라서 인접한 유한요소 사이의 계층수 차이를 1로 제한하면서 천이요소(transition mesh)를 허용하면, 특정한 요소를 계속 세분화하는 것이 가능하다.

참고문헌

- (1) Requicha, A. A. G., 1980, "Representation of Rigid Solid: Theory, Methods, and Systems," *ACM Computing Surveys*, Dec, pp. 437~464.
- (2) Meagher, D., 1984, "The Solid Engine: A Processor for Interactive Solid Modeling," *Proc. Nicograph 84 Conf.*, Tokyo, Nov.
- (3) Kambhampati, S. and Davis, L. S., 1986, "Multiresolution Path Planning for Mobile Robotics," *IEEE J. Robotics and Automation*, Sept., pp. 135~145.
- (4) Fujimura, K. and Samet, H., 1988, "Path Planning Among Moving Obstacles Using Spatial Indexing," *Proc. IEEE Int'l Conf. Robotics and Automation*, CS Press, Los Alamitos, Calif., pp. 1662~1667.
- (5) Shephard, M. S., Baehmann, P. L. and Grice, K. R., 1987, "Automatic Finite Element Modeling: Geometry Control for Direct Models," *Eng. Comput.*, Vol. 4, No. 2, pp. 119~125.
- (6) Yerry, M. A. and Shephard, M. S., 1985, "Automatic Mesh Generation for Three-Dimensional Solids," *Comput. Struct.*, Vol. 20, No. 1~3, pp. 31~39.
- (7) Kela, A., Saxena, M. and Perucchio, R., 1987, "A Hierarchical Structure for Automatic Meshing and Adaptive FEM Analysis," *Eng. Comput.*, Vol. 4, No. 2, pp. 104~112.
- (8) Perucchio, R., Saxena, M. and Kela, A., 1989, "Automatic Mesh Generation for Solid Models Based On Recursive Spatial Decompositions," *Int. J. Numer. Meth. Eng.*, Vol. 28, pp. 2469~2501.
- (9) Kela, A., 1989, "Hierarchical Octree Approximation for Boundary Representation-Based Geometric Models," *Computer Aided Design*, Vol. 21, No. 6, pp. 355~362.
- (10) Schroeder, W. J. and Shephard, M. S., 1990, "A combined Octree/Delaunay Method for Fully Automatic 3-D Mesh Generation," *Int. J. Numer. Meth. Eng.*, Vol. 29, pp. 37~55.
- (11) Shephard, M. S. and Baehmann, P. L., 1989, "Automated Adaptive Finite Element Modeling: Its Use and Misuse," ASME Winter Annual Meeting.
- (12) Yerry, M. A. and Shephard, M. S., 1984, "Automatic Three-Dimensional Mesh Generation by the Modified-Octree Technique," *Int. J. Numer. Meth. Eng.*, Vol. 20, pp. 1965~1990.
- (13) Wordenweber, B., 1981, "Automatic Mesh Generation of 2- and 3-Dimensional Curvilinear Manifolds," PhD Dissertation, Cambridge Univ., UK.
- (14) Watson, D. F., 1981, "Computing the N-Dimensional Delaunay Tessellation with Application to Voronoi Polygons," *Comput. J.*, Vol. 24, pp. 167~172.