

< 논문 >

얇은 두께의 3 차원 형상에 경계층 요소 생성

권기연* · 이병채† · 채수원**

(2004년 9월 17일 접수, 2005년 3월 7일 심사완료)

Boundary Layer Mesh Generation for Three-Dimensional Geometries with Thin Thickness

Ki-Youn Kwon, Byung-Chai Lee and Soo-Won Chae

Key Words : Boundary Layer Meshes(경계층 요소), Injection Molding(사출 성형), Prisms(프리즘), Offsetting(오프셋), Tetrahedral(사면체)

Abstract

A method of generation boundary layer mesh has been presented. This paper describes the generation of semi-unstructured prismatic/tetrahedral meshes for three-dimensional geometries with thin thickness. By offsetting of surface triangle elements prismatic/tetrahedral meshes are generated and using the node relocation method of this research intersected meshes can be efficiently improved. Finally tetrahedral meshes are automatically generated at the rest of the domain. Sample meshes are constructed to demonstrate the mesh generating capability of the proposed algorithm.

1. 서 론

최근들어 고정밀도와 특수한 기계적 성질을 요구하는 사출 성형품이 많이 생산되고 있어 금형 및 제품 설계시 컴퓨터를 이용한 수치실험이 부각되고 있다. 특히 플라스틱 제품은 두께가 얇아 이를 셀로 표현하여 삼각형 또는 사각형 요소로 모델링 한 후 해석에 이용한다. 그러나 대부분의 모델들은 매우 복잡하여 모델링 과정에 많은 시간이 소요되고 셀로 표현하기에는 부적합할 정도의 두께를 가진 부분도 함께 포함하고 있다. 따라서 최근들어서는 솔리드 요소를 이용하여 실제 형상과 똑 같이 유한 요소 모델링을 생성하는 추세이다. 사출 성형을 비롯한 유동 해석에서는 정확한 해를 얻기 위해서는 얇은 부분에도 여러층의 요소를 생성하는 것이 필요하다. 일반적인 요소망 생성 기법으로는 이러한 문제를 해결하기 어려우므로 경계층 요소(boundary layer mesh)를 생성한 후 내부

에 요소를 생성하는 것이 필요하다. 경계에는 오프셋 방법을 이용하여 여러층의 사면체 또는 프리즘 요소를 생성하고, 내부에는 자동으로 사면체 요소를 생성한다.⁽¹⁻⁶⁾ Fig. 1은 직육면체 형상에 경계층 요소를 생성한 예의 단면을 나타내고 있다. 여기서 삼각형은 사면체 요소이고, 사각형은 프리즘 요소를 나타낸다. 내부에 있는 사면체 요소는 요소 사이의 연결성을 맞추기 위해서 생성된 프리즘 요소의 삼각형 면을 이용하여 요소를 생성한다. 오프셋을 시킬 때 형상의 모든 부분에 균일한 크기를 지정하면 얇은 부분과 두꺼운 부분이 함께 존재하는 모델에 대해서는 오프셋 깊이가 얇은 부분에 의해 정해져 적절하지 못하다[Fig. 2(a)]. 그래서 모델의 두께에 따라 요소들이 서로 교차하지 않은 범위 내에서 오프셋 깊이를 적절히 수정할 필요가 있다[Fig. 2(b)]. 기존의 방법들은 교차하는 영역에 형상비(aspect ratio)가 매우 큰 요소들을 생성시켜 상대적으로 두께가 얇은 영역에 많은 수의 요소를 생성시켰다. 그리고 교차된 요소를 수정하는 과정을 순차적으로 진행시켜 많은 시간이 소요되었다. 본 연구에서는 일정 깊이로 오프셋을 시킨 후 교차하는 부분을 효율적으로 찾아서 요소를 수정함으로써 두께가 얇은 형상에 대해 유동 해석이 가능한 요소망을 생성할 수 있도록 하였다.

* 한국과학기술원 기계공학과

† 책임지자, 회원, 한국과학기술원 기계공학과

E-mail : bchlee@kaist.ac.kr

TEL : (042)869-3031 FAX : (042)869-3095

** 고려대학교 기계공학과

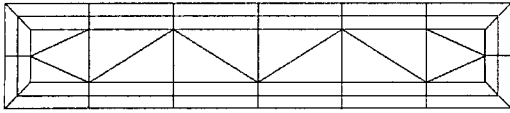
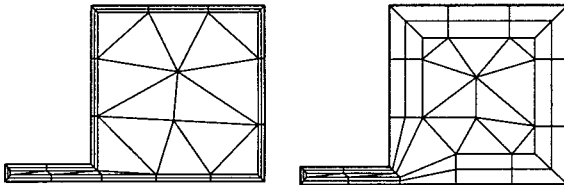


Fig. 1 Boundary layer mesh generation



(a) Applied uniform offset (b) Applied different offset depth by thickness

Fig. 2 Offsetting

2. 본 론

얇은 형상에 두께를 고려하여 경계층 요소를 생성하는 방법은 곡면에 생성된 삼각형 요소를 이용하여 균일한 사면체 또는 프리즘 요소를 생성시키고, 남은 영역에 사면체 요소를 자동으로 생성시키는 것이다. 개략적인 순서는 다음과 같다.

Step 1. 표면의 삼각형 요소를 읍셋시켜 프리즘 요소를 생성한다.

Step 2. 필요에 따라 프리즘 요소를 사면체 요소로 분할한다.

Step 3. 요소가 서로 교차하거나 요소 사이의 거리가 너무 가까우면 절점을 이동시켜 요소를 적절히 수정한다.

Step 4. 남은 영역에 자동으로 사면체 요소를 생성한다.

초기에 입력 받는 데이터는 형상 표면의 삼각형 요소이고 다음과 같은 조건을 만족시켜야 요소망 생성이 가능하다. 모든 삼각형 요소의 법선 방향은 일정하고, 페루프를 구성하고 있어야 한다. 또한 다중 영역이 존재해서는 안된다.

2.1 프리즘 요소 생성

프리즘 요소는 경계의 삼각형 면을 요소의 법선 방향을 고려해 내부로 읍셋시켜 생성한다(Fig. 3).

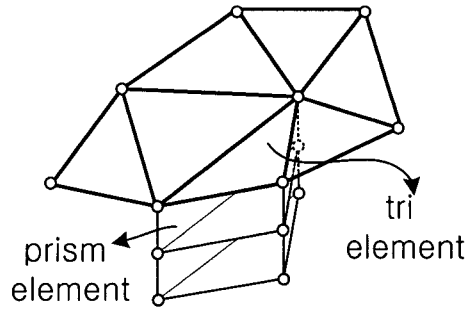


Fig. 3 Prism meshes generation by offsetting

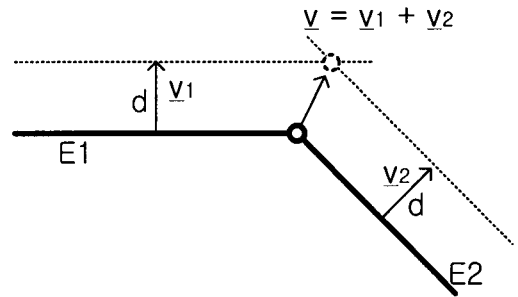


Fig. 4 New node positions

새로운 절점의 위치는 요소 E1, E2 가 각각의 법선 방향으로 거리 d 만큼 이동했을 때 일치하는 곳이다. 2 차원일 경우는 유일한 점이 존재하지만 3 차원인 경우는 그렇지 않아 최적의 위치를 찾는 것이 필요하다. 우선 절점이 초기에 이동할 방향은 인접한 요소의 법선방향을 평균해서 구한다 (Fig. 4). 그리고 기존 절점과 인접한 요소의 법선 방향을 이용해 현재 이동한 상태의 절점으로 직선 방정식을 구성하고, 요소로부터 평면 방정식을 구성하여 만나는 위치를 구한다. 이 위치가 현재 이동한 상태의 절점이 되고, 다른 인접한 요소와 같은 방법으로 새로운 절점의 위치를 구한다. 이와 같은 방법으로 여러 번 반복하여 절점의 위치를 결정하게 된다.

여러층이 생성될 경우에는 마지막에 생성될 층의 절점만을 계산하며 나머지 층의 절점의 위치는 초기 절점과 마지막 층의 절점을 이용해서 구한다. Fig. 5 에서 살펴보면 절점 n4 는 절점 n1 을 주변 요소 정보를 이용하여 새롭게 생성시킨 절점이고, 절점 n2, n3 는 이 두 절점을 이용해서 구한다.

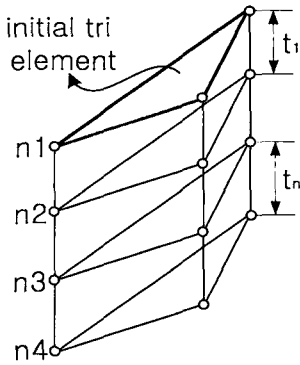


Fig. 5 Recalculate position of nodes

가운데 층의 절점들은 식 (1)을 이용하여 구해진다. 여기서 α 는 요소의 치우침 비를 의미하며 마지막 층의 두께(t_n)와 첫번째 층의 두께(t_1) 비이다. 그리고 d 는 읍셋 깊이를 나타내고, N 은 생성될 요소의 층수를 나타낸다. 각 층의 두께(t_i)를 구한 후 절점의 위치(P_i)를 계산한다.

$$P_i = P_1 + (P_n - P_1) \times \frac{t_i}{d} \quad (1)$$

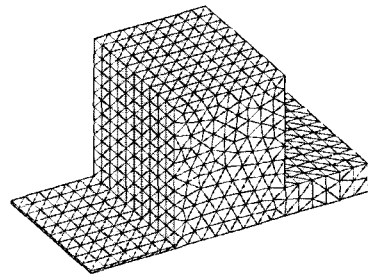
여기서 t_i 는 i 번째 층의 두께를 나타내며 다음과 같이 계산된다.

$$r = \left(\frac{1.0}{N-1.0} \right)^\alpha, \quad \alpha = \frac{t_n}{t_1}$$

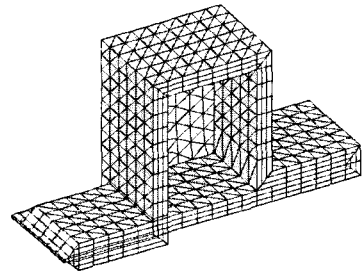
$$a = \frac{d}{N^r - 1.0} \quad (2)$$

$$t_i = a \times \left(\frac{i^r - 1}{r - 1} \right)$$

Fig. 6(a)는 표면이 삼각형으로 이루어진 모델로 높이가 100 이고 가장 얇은 부분의 두께는 2 이다. Fig. 6(b)는 읍셋 깊이 값을 6, 요소 생성 층의 수를 3 개로 설정해 프리즘 요소를 생성한 결과의 일부분을 나타내고 있다. 그림에서 알 수 있듯이 얇은 영역과 구석부분에서 요소들이 서로 교차하였고 일부 요소들이 매우 인접하여 생성되었다. 이와 같은 경우에는 내부에 자동으로 사면체 요소를 생성할 수 없으므로 3 단계에서 요소가 교차하거나 너무 가까이 있지 않도록 2.3 절의 방법을 이용하여 절점을 이동시킨다.



(a) Initial surface elements



(b) Prism elements

Fig. 6 Prism elements generation

2.2 프리즘 요소를 사면체 요소로 분할

모델 전체에 사면체 요소만을 생성할 경우에 프리즘 요소를 사면체 요소로 분할해야 한다. 프리즘 요소가 사면체로 분할될 수 있는 타입은 Fig. 7 과 같이 여덟가지가 있다. 타입 1~6 은 3 개의 사면체 요소로 분할되며, 타입 7, 8 은 내부에 하나의 절점이 생성되면서 다섯 개의 사면체 요소로 분할된다. 타입 7 과 8 은 형상이 좋지 않은 사면체 요소를 생성할 가능성이 많고 불필요하게 많은 수의 요소를 생성하므로 가급적 이용되지 않도록 해야 한다. 또한 요소가 분할될 때 인접요소와의 연결성이 일치해야 한다. 따라서 본 연구에서는 Dompierre⁽⁷⁾가 제안한 절점의 전역 번호를 이용하여 요소를 분할하는 방법을 사용하였다. 프리즘이 사면체로 분할되기 위해서는 세 개의 사각면이 각각 두개의 삼각형으로 분할되는 것이 필요하다. 이 방법은 요소내에서 가장 큰 전역 번호를 가지는 절점을 기준으로 두 개의 사각형 면을 대각선 방향으로 분할하고, 나머지 한 개의 면은 임의로 분할한다. 그림 8 에서 살펴보면 여섯 개의 절점 중 N1 이 가장 큰 절점 번호에 해당한다면, 두 개의 사각형 면 (N1, N2, N5, N4), (N1, N3, N6, N4)은 각각 그림과 같이 2 개의 삼각형 면으로 분할된다. 나머지 면 (N2, N5, N6, N3)은 어떤 방향으로 분할되어도 타입 7, 8 이 아니다. 그리고 두 면(N1, N2, N5, N4), (N1, N3, N6, N4)을 포함하고 있는 다른 요소에서도 N1 이 가장 큰 절점 번호에 해당하므로 같은 방향으로 2 개의 삼각형으로 분할되어 인접

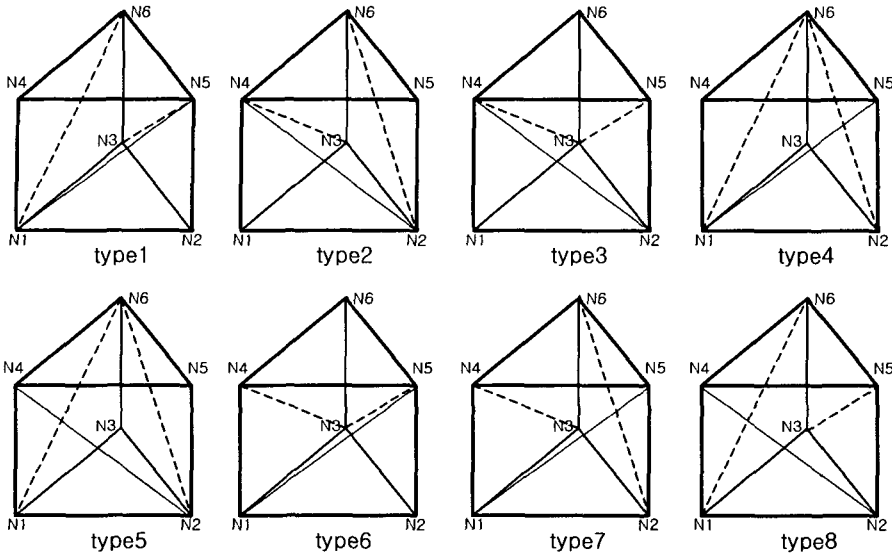


Fig. 7 Eight cases to split the tetrahedral element

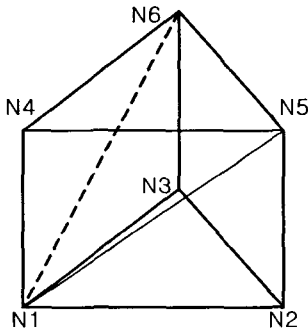


Fig. 8 The smallest vertex N1

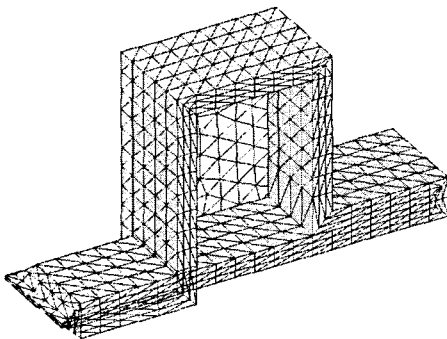


Fig. 9 Constructed mesh with tetrahedral elements

한 요소와 연결성이 일치된다. 이 방법은 별도의 계산 작업이 필요 없으므로 분할 속도가 빠르다. Fig. 9 는 5004 개의 프리즘 요소가 15012 개의 사면체 요소로 분할된 예를 보여주고 있다.

2.3 절점 이동

모델에서 움푹 깊이를 크게 주면 Fig. 6(b)에서와 같이 요소들이 서로 교차하거나 요소 사이의 거리가 너무 가까워 사면체 요소망 생성이 불가능하거나 요소의 품질이 매우 떨어지는 요소가 생성될 수 있다. 따라서 이와 같은 방법을 해결하기 위해서 절점을 이동시키는 방법이 필요하다. 본 연구에서는 여러 단계에 걸쳐 교차 체크를 하여 국부적으로 절점을 이동시키는 방법⁽²⁾과는 달리 한번의 과정을 통해 필요한 절점을 이동시키도록 하였다. 요소를 움푹 시켜 프리즘 또는 사면체 요소가 생성되었을 때 수정이 필요한 경우를 Fig. 10 과 같이 네 가지 경우로 나누고, 각각의 방법에 대해 절점을 이동시켜 사면체 요소를 생성시킬 수 있도록 하였다. 첫번째 경우는 형상의 모서리 부분에서 나타날 수 있으며 요소의 절점 자체가 서로 엇갈려서 생성된다. 두번째 경우는 서로 다른 요소가 영역 내부에서 교차하는 경우이고, 세번째 경우는 서로 다른 요소가 영역 외부까지 생성되어 서로 교차하는 경우이다. 그리고 마지막 경우는 요소가 교차하는 하지 않지만 요소 사이의 거리가 너무 가까워 사면체 요소를 생성시킬 경우 요소 품질이 매우 떨어질 수 있다.

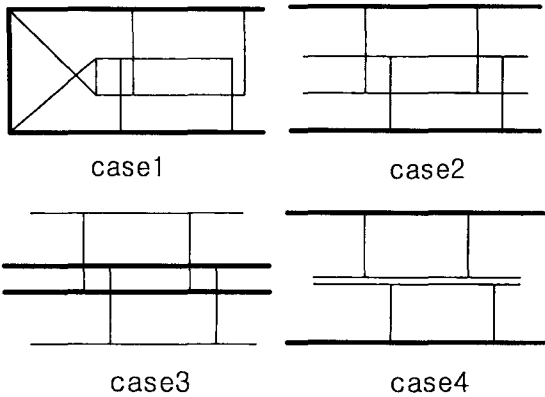


Fig. 10 Four cases to move nodes

이런 경우들을 해결하기 위해서는 교차하거나 거리가 가까운 부분의 절점들을 찾아서 경계쪽으로 밀어주어 이를 해결한다.

2.3.1 수정이 필요한 요소 찾기

모든 영역에 대해 요소가 서로 교차하는지, 너무 가까이 있는지를 확인하면 많은 시간이 소요되므로 가능한 찾고자하는 요소에 인접한 요소들에 대해서만 체크를 수행하도록 해야 한다. 이 체크를 효율적으로 하기 위하여 다음과 같은 방법을 사용하였다. 여기서 설명하는 방법은 2D 를 예로 들어 설명한 것이고, 3D 로 확장하기 위해서는 같은 방법을 사용하고 높이에 해당하는 변수를 추가시키면 된다.⁽⁹⁾ 우선 영역의 크기를 정하여 가로, 세로 격자 개수를 정한다. 영역의 최대 최소 범위를 정할때는 약간의 오차를 두어 경계에 걸쳐진 대상이 오작동을 범하지 않도록 한다. 격자개수는 대상의 요소에지 길이와 가로세로 크기(l_1, l_2)비를 고려하여 구하여 모델에 따라 유동적으로 격자개수가 설정되도록 한다. 이렇게 최대 최소값 및 격자 개수가 정해지면 식 (3)을 이용하여 절점이 어떤 그리드에 속하는지를 판단한다.

$$X = \min x + (x - \min x) / l_1 \times n_1 \quad (3)$$

$$Y = \min y + (y - \min y) / l_2 \times n_2$$

여기서

$$l_1 = \max x - \min x$$

$$l_2 = \max y - \min y$$

n_1 : number of horizontal grid

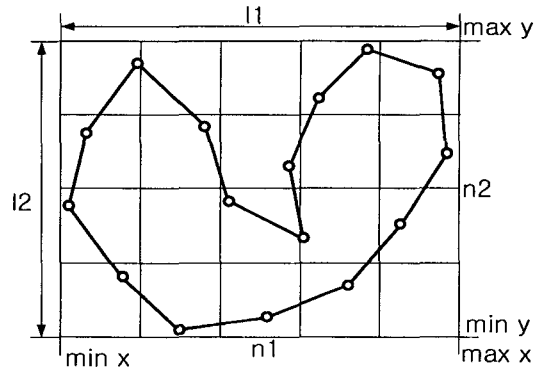


Fig. 11 A grid generation in 2D

n_2 : number of vertical grid

(x, y) : coordinate of node

이다.

여기서 X, Y 는 그리드의 위치를 나타내며, (X, Y)번째 그리드에는 이 절점을 포함하는 요소에지가 리스트로 저장된다. 이와 같이 하면 대상 요소의 절점에 인접한 요소가 저장되어 있다. 이 작업은 절점의 좌표값 자체가 그리드 위치가 되므로 별도의 찾기(searching) 작업 없이 인접한 요소를 찾을 수 있다.

2.3.2 절점 밀어내기

절점을 경계 쪽으로 밀어내기 위해서는 이동 범위를 정하는 것이 필요하다. 또한 이동된 절점이 경계 밖으로 나가면 안되고, 요소가 균일하게 생성될 수 있도록 해야한다. 절점을 밀어내는 방식은 다음과 같다. 여기서 P_i' 는 i 번째 절점의 새로운 위치를 나타내고 식 (4)와 같이 구해진다. 여기서 N 은 요소 층 수를 나타내고, β 는 이동량을 나타내는 변수이다. 따라서 각각의 경우에 대해 적절한 β 값을 구해야 한다. α 는 여러층이 생성될 경우 처음 층에 생성되는 요소의 옵셋 방향거리와 마지막층에 생성되는 요소의 거리비를 나타내는 치우침비(biasing level)이다. 값이 1 인 경우는 요소가 균등한 간격으로 생성되며 1 보다 큰 경우에는 경계로 갈수록 요소 간격이 좁아진다. 우선은 치우침 비가 1 인 경우로 가정하여 절점을 이동시킨 후, 치우침비를 고려하여 절점의 위치를 재수정한다.

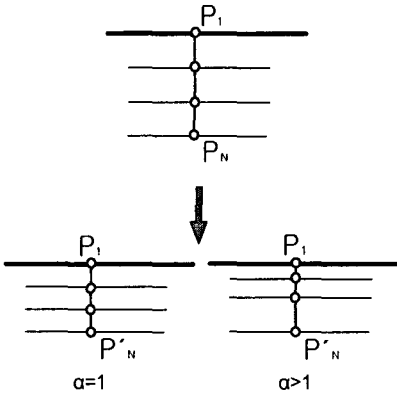


Fig. 12 Nodes relocation

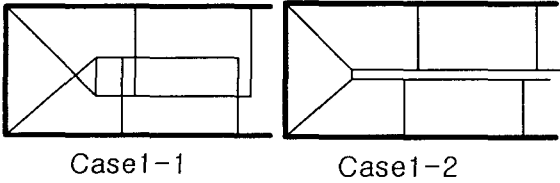


Fig. 13 The first case

$$P'_i = P_1 + \frac{P_N - P_1}{N} \times i \times \frac{1}{\beta} \quad (4)$$

Case 1

이 경우는 Fig. 13 과 같이 두 가지 경우가 있을 수 있다. 즉, 요소의 절점 번호가 엇갈리는 경우와 요소의 한 면이 매우 작아지는 경우가 있다. 첫번째 경우를 판단하는 방법은 읍셋 시키기 전의 절점에 의한 요소의 법선 방향과 읍셋 시킨 후의 절점에 의한 요소의 법선 방향이 반대이다. 두 번째 경우는 읍셋 시킨 후의 절점을 가지고 구한 면적이 읍셋 전의 면적에 비해 많이 작아지는 경우이다. Fig. 14 는 두 층을 가지는 요소가 교차하는 예이다. 두 번째 그림과 같이 교차점까지 균등하게 N+1 층이 생성될 수 있도록 β 값을 구한다. 줄어든 길이(L_r)는 길이 l_2 에 T 를 더한 값이며, 이를 정리하면 β 값은 식 (5)로 표현된다.

$$\beta = \frac{(N+1) \times (\frac{l_2}{l_1} + 1)}{N} \quad (5)$$

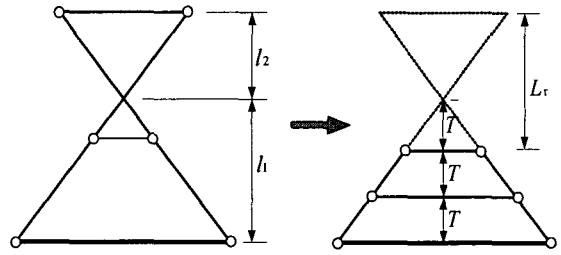


Fig. 14 An example of the first case

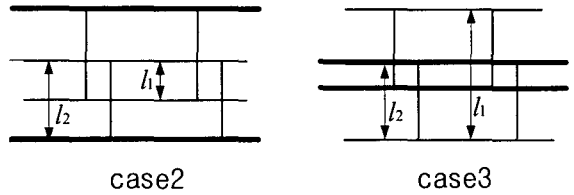


Fig. 15 The second and third cases

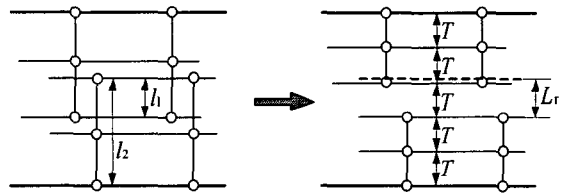


Fig. 16 Nodes relocation of case2, case3

여기서

$$L_r = l_2 + T, \quad T = \frac{l_1}{N+1}$$

$$L_r = (l_1 + l_2) - (l_1 + l_2) \times \frac{1}{\beta} \quad \text{이다.}$$

Case2,3

타입 2, 3 은 같은 경우이므로 동일한 식을 적용시켜 β 값을 구한다. 요소가 서로 교차하는지를 판단하는 방법은 표면상의 절점과 읍셋시킨 후의 절점으로 선분을 구성하고, 요소 면이 사각형인 경우는 2 개의 삼각형으로 분리하고 삼각형인 경우는 그대로 이용하여 선분과 삼각형이 교차하는지를 판단한다. β 값을 계산하는 방법은 절점을

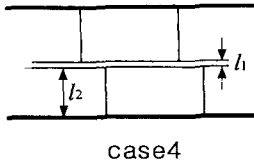
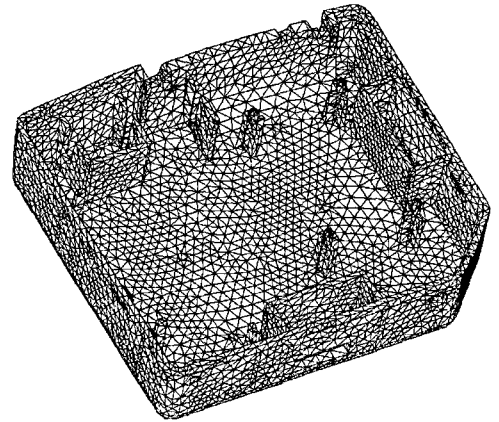


Fig. 17 Nodes relocation of case 4



(a) Initial input mesh with triangular elements

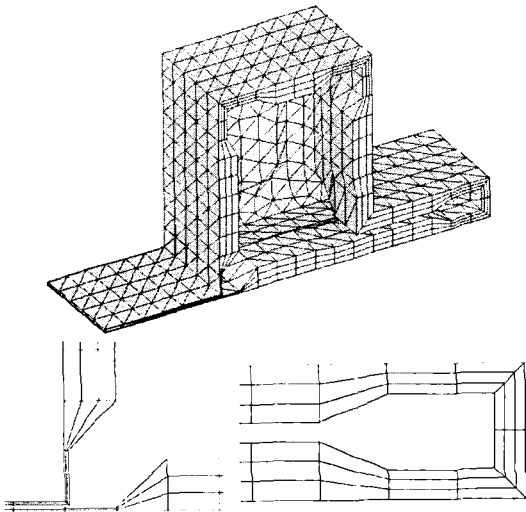
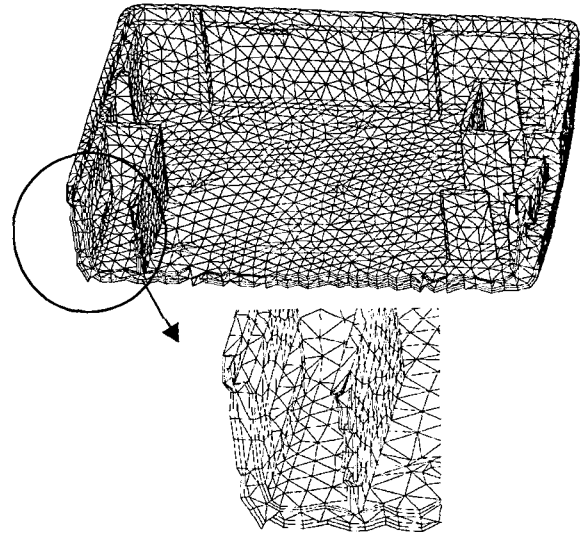


Fig. 18 An example of nodes relocation. (biasing level = 1)



(b) 33920 tetra elements, 28724 prism elements

Fig. 20 A thin plastic case model

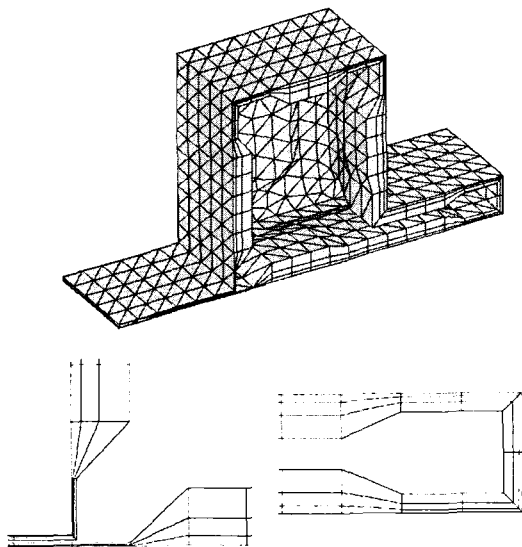


Fig. 19 An example of nodes relocation. (biasing level = 3)

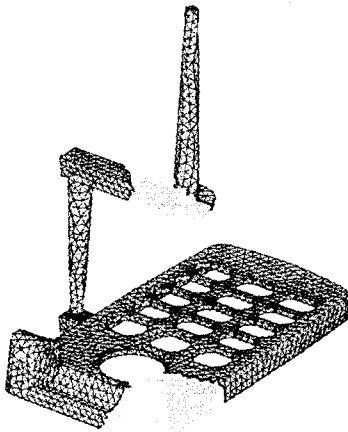
양쪽 방향으로 밀어낸 후 사이에 남은 영역의 길이 과 한층의 두께(T)와 같도록 하는 것이며, 이를 계산하면 β 값은 식 (6)와 같다.

$$\beta = \frac{2 \times l_2}{2 \times l_2 - l_1 - T} \quad (6)$$

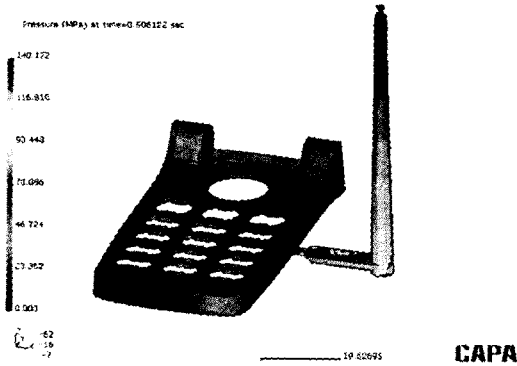
여기서

$$L_r \times 2 - l_1 = T, T = \frac{2 \times l_2 - l_1}{2 \times N + 1}$$

$$L_r = l_2 - l_2 \times \frac{1}{\beta} \text{ 이다.}$$



(a) 14674 tetra elements, 13744 prism elements



(b) CAPA(a pressure distribution)

Fig. 21 A phone case model 1

case 4

타입 4는 Fig. 17 과 같이 요소가 서로 교차하지는 않지만 거리가 너무 가까이에 있어 사면체 요소를 생성 할 경우 생성이 불가능하거나 품질이 매우 떨어지는 현상이 발생한다. 따라서 이 경우에도 절점을 이동할 필요가 있다. β 값을 계산하는 방법은 앞절의 방법과 유사하며 절점을 양쪽 방향으로 밀어낸후 사이에 남은 영역의 길이가 한 층의 두께(T)와 같도록 하는 것이며, 이를 계산하면 β 값은 식 (7)과 같다.

$$\beta = \frac{2 \times l_2}{2 \times l_2 + l_1 - T} \tag{7}$$

여기서 $L_r \times 2 + l_1 = T$, $T = \frac{2 \times l_2 - l_1}{2 \times N + 1}$

$L_r = l_2 - l_2 \times \frac{1}{\beta}$ 이다.

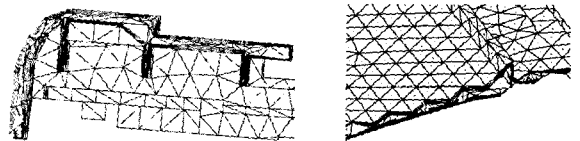
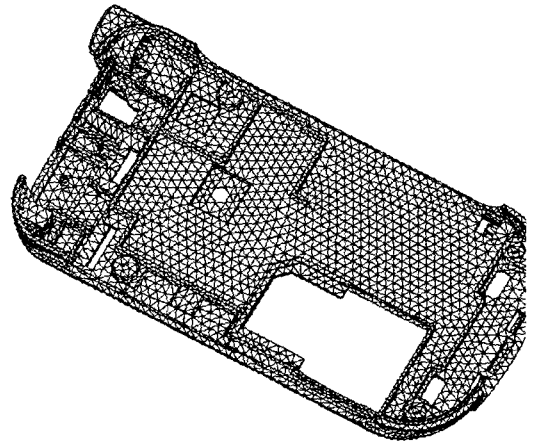


Fig. 22 A phone case model 2 (44375 tetra elements, 39570 prism elements)

Fig. 18, 19 는 교차하거나 거리가 가까운 요소의 절점들을 이동시킨 것이며, Fig. 18 은 치우침비가 1 이 적용되었고, Fig. 19 는 치우침비가 3 이 적용되었다.

2.4 사면체 요소망의 생성

내부의 영역에 대해서는 경계에 생성된 사면체 요소 또는 프리즘 요소의 삼각형 면을 이용하여 자동으로 사면체 요소를 생성한다.^(8,9)

3. 적용 예

본 연구에서 제안한 방법을 이용하여 두께가 얇은 형상에 적용해보았다. Fig. 20 은 14362 개의 삼각형으로 구성되어 있으며, 모델의 평균 두께는 1.8~2.2 정도이며, 최소 두께는 1 보다 작다. 읍셋 깊이를 0.7, 층의 개수를 2, 치우침 비를 1 로 주고, 경계의 프리즘 요소를 사면체로 분할하지 않았다. 생성 결과 표면에 28724 개의 프리즘 요소가 생성되었고, 내부에 33920 개의 사면체 요소가 생성되었다. Fig. 20(b)는 생성 결과의 단면을 보여주고 있다. 두께가 비교적 두꺼운 부분에서는 초기에 입력한 값만큼 읍셋이 이루어졌고, 얇은 영역에서는 상대적으로 적은량이 읍셋되었음을 확인할 수 있다. Fig. 21 은 표면이 6872 개의 삼각형으로 구성되어 있으며, 모델의 평균 두께는 1.0~1.5 정도이며, 최소 두께는 0.3 보다 작다. 읍셋 깊이를 1, 층의 개수를 2, 치우침 비를 1 로 주었다. Fig. 21(b)는 사출 성형 해석프로그램인 CAPA⁽¹⁰⁾를 이용하여 해석을 수

행한 결과를 나타낸다. Fig. 22 는 13190 개의 삼각형 요소로 구성되어 있으며, 평균 두께가 0.5~1.0 정도 이며, 최소 두께는 0.1 보다 작다. 읍셋 깊이를 0.3, 층의 개수를 3, 치우침 비를 1 로 주었다.

4. 결 론

본 논문에서는 얇은 두께를 가지는 형상에 사출성형을 비롯한 유동해석에 적합한 요소망을 생성할 수 있는 방법을 제안하였다. 이는 표면의 삼각형 요소를 읍셋이동시켜 프리즘 또는 사면체 요소를 생성하고, 내부에는 자동으로 사면체 요소를 생성시키는 기존의 방법에 얇은 부분에서 서로 요소가 교차하지 않도록 효율적으로 수정하는 방법을 제시하였으며 다양한 예제에 적용시켜 이를 검증하였다.

후 기

본 연구는 과학기술부 공학용 해석 소프트웨어 기술 개발 사업의 연구비 지원에 의해 수행된 것입니다.

참고문헌

- (1) Khawaja, A. and Kallinderis, Y., 2000, "Hybrid Grid Generation for Turbomachinery and Aerospace Applications," *Int. J. Numer. Meth. Engng.*, Vol. 49, pp. 145~166.
- (2) Garimella, R. V. and Shephard, M. S., 2000, "Boundary Layer Mesh Generation for Viscous Flow Simulations," *Int. J. Numer. Meth. Engng.*, Vol. 49, pp. 193~218.
- (3) Krause, J., Stresker, N. and Fichtner, W., 2000, "Boundary-Sensitive Mesh Generation Using an Offsetting Technique," *Int. J. Numer. Meth. Engng.*, Vol. 49, pp. 51~59.
- (4) Khawaja, A., Minyard, T. and Kallinderis, 2000, Y., "Adaptive Hybrid Grid Methods," *Comput. Methods Appl. Mech. Engng.*, Vol. 189, pp. 1231~1245.
- (5) Pirzadeh, S., 1966, "Three-Dimensional Unstructured Viscous Grids by the Advancing-Layers Method," *AIAA Journal*, Vol. 34, pp. 43~49.
- (6) Kallinderis, Y., Khawaja, A. and McMorris, H., 1996, "Hybrid Prismatic/Tetrahedral Grid Generation for Viscous Flows Around Complex Geometries," *AIAA Journal*, Vol. 34, pp. 291~298.
- (7) Dompierre, J., Labbe, P., Vallet, M. G. and Camarero, R., 1999, "How to Subdivide Pyramids, Prisms and Hexahedra into Tetrahedra," *The Proceedings of the 8th International Meshing Roundtable*, pp. 195~204.
- (8) Kim, Y. W., Kwon and G. W., Chae, S. W., 2002, "Unstructured Tetrahedral Meshing by an Edge-Based Advancing Front Method," *KSME International Journal*, Vol. 16, No. 2, pp. 211~218.
- (9) Kwon, K. Y., Chae, S. W. and Lee, B. C., 2004, "Automatic Tetrahedral Mesh Generation Using 3-D Operators", *Society of CAD/CAM Engineers*, Vol. 9, pp. 27~34.
- (10) CAPA v. 6.0 Manual, VMTech Co., 2003