

쿼드트리를 이용한 일반적인 3 차원 트림곡면에서의 유한요소 자동생성

유동진*, 윤정환**

Automatic Mesh Generation in the General Three-Dimensional Trimmed Surface using Quadtree

Dong-Jin Yoo* and Jeong-Whan Yoon**

ABSTRACT

In this work, a general method for the mathematical description of three-dimensional trimmed surface is proposed by introducing the base parametric surface and boundary curves. Since mesh density distribution for the analysis may vary by cases, a grid-based mesh generation algorithm using quadtree is proposed in the present work. For the assurance of connectivity of generated meshes among surfaces, a method for the pre-cleaning of boundary curves has been developed to be used in the automatic generation of the finite elements. In addition, mesh-smoothing algorithm is suggested which can be used in the general trimmed surface. In this algorithm nodes are moved on the original surface by the normal projection in each iterative smoothing procedure.

Key Words : Three-dimensional trimmed surface (3 차원 트림곡면), Quadtree (쿼드트리), Mesh Smoothing (요소 평활화)

1. 서론

유한요소의 자동생성에 관해서는 여러 가지 방법들이 보고되어 왔는데, 대표적인 방법으로는 위상분해법 (topology decomposition approach), 절점 연결법 (node connection approach), 기하분할법 (geometry decomposition approach), 격자이용법 (grid-based approach), 오프셋 이용법 (offset or paving approach), 전방전진법 (advancing front approach) 등이 있다. 위상분해법^[1]은 다각형의 해석영역이 주어졌을 때 다각형의 꼭지점을 연결하여 비교적 큰 형태의 삼각형으로 만든 후 이를 다시 원하는 정

도의 작은 삼각형으로 분할하는 방법이다. 절점 연결법^[2,3]은 해석영역의 경계와 내부에 절점을 규칙적인 또는 불규칙적인 방법으로 생성하고 이렇게 생성된 절점들을 연결하여 유한요소를 생성하는 방법이다. 절점을 생성하는 방법 및 생성된 절점들을 요소화 하는 방법의 차이에 따라 여러 가지 연구들이 보고되어 왔다. 기하분할법^[4,5]은 해석영역의 경계곡선 내 배치된 주요 절점들을 기준으로 가장 좋은 분할선으로 양분해 나가는 과정을 계속 하여 모든 분할 영역이 사각형이 될 때까지 수행한다. 비교적 양질의 사각 요소를 생성할 수 있다는 장점이 있다. 격자이용법^[6]은 균일한 간격의 격

* 대전대학교 컴퓨터응용 기계설계학과

** LG 생산기술원 Design Engineering Center

자들이 유한요소가 될 수 있겠다는 생각에서 시작되었는데 해석영역에 가상의 균일한 간격의 격자를 썩은 뒤 경계를 포함하지 않는 내부의 격자를 그대로 유한요소로 사용하고 물체의 경계를 포함하는 격자들은 여러 가지 방법에 의해 삼각형 또는 사각형 유한요소로 적당히 분할하는 방법이다. 내부에는 좋은 형상의 요소가 생성되는 반면 경계 곡면주위에 질이 좋지않은 요소가 생성될 수 있는 단점이 있다. 오프셋이용법⁷⁾은 경계곡면에 절점들을 생성시킨 후 해석영역으로 일정거리만큼 오프셋 해나가는 과정을 반복해 감으로서 양질의 사각형 요소를 얻고자 하는 방법이다. 2차원에서는 성공적인 연구 사례들이 다수 발표되어 있으나 3차원 트림곡면의 경우에는 오프셋 된 요소들간의 간섭처리, 절점들의 원래 곡면으로의 수직투영(normal projection)의 문제 등 보다 상세한 연구가 계속되고 있다. 전방전진법⁸⁾은 경계곡선에 절점들을 생성시킨 후 해석영역에 적당한 절점을 생성시킴과 동시에 요소를 생성해 나가는 방법으로서 전방 요소들의 모서리들을 새로운 경계 곡선으로 하여 해석영역이 모두 분할될 때 까지 반복적으로 수행된다. 최적의 절점을 매 스텝마다 찾아나간다는 특성 때문에 양질의 요소를 얻을 수 있으나 3차원 트림곡면의 경우에는 내부 홀(inner hole)의 처리, 절점들의 생성과 관련된 계산시간의 문제 등 지속적인 연구가 요망되고 있다.

본 연구에서는 일반적인 3차원 트림곡면에서의 유한요소 생성을 위하여 격자 이용법을 기본으로 하고 요소의 밀도 분포를 조절할 수 있도록 quadtree 기법을 혼합한 방법을 제안하였다. 일반적인 곡면 모델은 trimmed planar surface 또는 trimmed curved surface 등의 복수개의 곡면들이 위상학적으로 구성되어 있는 형태이므로 각 곡면별로 유한요소를 생성할 수 있지만 각 곡면의 연결부에 생성된 요소의 절점들이 일치하지 않는 경우가 대부분이므로 문제의 성격에 따라서는 각 곡면에서 생성된 요소의 절점의 위치를 일치시키는 수작업이 뒤따라야 한다. 본 연구에서는 유한요소 생성 전에 모든 곡면들간의 경계곡선상의 요소 절점수를 일치시키는 알고리즘을 개발하였다.

2. 3차원 트림곡면의 수식적 표현

단순한 해석적 표현에 의해 묘사되지 않는 임

의 복잡한 형상의 곡면은 매개변수 곡면의 조합으로서 표현될 수 있다. 일반적인 자유형상의 곡면은 크게 프리곡면 (free surface)과 트림곡면(trimmed surface)으로 나눌 수 있는데 Fig.1에서와 같이 프리곡면은 곡면의 범위를 정의하는 경계곡선이 필요 없는 형태의 곡면을 의미하고 트림곡면은 곡면의 사용범위를 정의하기 위한 바깥경계곡선(outer boundary curve)과 안쪽 경계곡선(inner boundary curve)이 존재하는 보다 일반적인 형태의 곡면으로서 일반 산업현장에서 사용되고 있는 대다수의 CAD 시스템은 이 두가지 형태의 매개변수 곡면의 조합으로서 복잡한 형상을 묘사하게 된다⁹⁾.

각 곡면은 다시 다수의 패치(patch)의 조합으로서 나타내어지는 데 매개변수 패치상의 한 점의 3차원 좌표는 다음과 같이 표시된다.

$$\mathbf{r}(u,v) = x(u,v)\mathbf{i} + y(u,v)\mathbf{j} + z(u,v)\mathbf{k} \quad (1)$$

$$(0 \leq u, v \leq 1)$$

여기서 $x(u,v)$, $y(u,v)$ 와 $z(u,v)$ 는 사용하고자 하는 패치의 형태에 따라 여러 가지 방법으로 수식화 될 수 있는데 본 연구에서는 일반 CAD 시스템과의 효율적인 연계성을 위하여 다음과 같은 u, v 에 관한 일반적인 다항식을 사용하였다^{[10],[11]}.

$$\mathbf{r}(u, v) = \begin{bmatrix} 1 \\ u \\ \vdots \\ u^{n-1} \end{bmatrix} \begin{bmatrix} A_{11}^i & \cdots & A_{m1}^i \\ \vdots & & \vdots \\ A_{1n}^i & \cdots & A_{mn}^i \end{bmatrix} \begin{bmatrix} 1 \\ v \\ \vdots \\ v^{m-1} \end{bmatrix} \quad (2)$$

여기서 계수행렬의 첨자 i 가 1일 때는 $x(u,v)$, i 가 2일 때는 $y(u,v)$, i 가 3일 때는 $z(u,v)$ 이 계산을 위한 계수를 의미하고, n, m 은 각각 u 방향의 차수, v 방향의 차수를 의미한다. 프리곡면은 u 방향의 패치수(n patch), v 방향의 패치수(m patch), u 방향의 order (n), v 방향의 order (m)로서 $(3 \times n \text{ patch} \times m \text{ patch} \times n \times m)$ 개의 계수가 주어지면 식(2)에 의하여 수식적으로 표현될 수 있다.

트림곡면은 식(2)로 표현되는 각 패치의 수치 정보 외에 사용되어질 부분과 사용되지 않을 부분을 구분 짓기 위한 경계곡선의 데이터가 추가적으로 요구되어 지는데 매개변수 t 에 관한 함수로 표

현된다. 경계곡선 역시 많은 세그먼트(segment)로 이루어 지는데 한 세그먼트는 다음과 같이 표현된다⁹⁾.

$$u(t) = \sum_{i=0}^{n-1} a_i t^i \quad (3a)$$

$$v(t) = \sum_{i=0}^{n-1} b_i t^i \quad (3b)$$

여기서 a_i, b_i 는 각각 u, v 에 관한 다항식의 계수들이고 n 은 곡선의 오더이다. 매개변수 t 의 범위는 시작 세그먼트와 마지막 세그먼트를 제외하고는 $0 \leq t \leq 1$ 로 정의된다. 즉 트림곡면 등 패치의 수치정보 외에 바깥쪽 경계곡선, 안쪽 경계곡선의 매개변수 u, v 를 계산하기 위한 계수들이 필요하고 식(3)에서 구한 u, v 를 식(2)에 대입함으로써 경계곡선상의 임의의 매개변수 값 t 에서의 3 차원 좌표값을 구할 수 있다.

트림면(trimmed surface)을 더 세분화하면 Fig.2에서와 같이 트림평면 (trimmed planar surface) 과 트림곡면 (trimmed curved surface)으로 분류할 수 있는데 트림곡면의 경우는 식(2)와 식(3)과 같이 정의되는 반면 트림평면의 경우는 식(3)은 필요로 하나 곡면에서나 필요한 식(2)에서의 계수들 대신 다음과 같은 평면의 방정식을 구성하기 위한 수치 데이터를 필요로 한다.

$$\mathbf{r} \cdot \mathbf{n} = P \quad (4)$$

여기서

\mathbf{r} : 평면상의 한 점을 나타내는 위치벡터

\mathbf{n} : 평면에 수직인 단위벡터

P : 좌표계의 원점과 평면과의 수직거리이다.

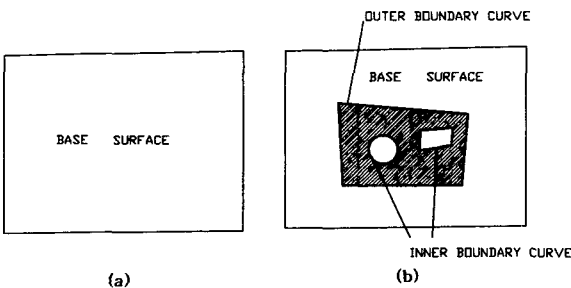


Fig. 1 Classification of general three-dimensional surface; (a) Free surface, (b) Trimmed surface

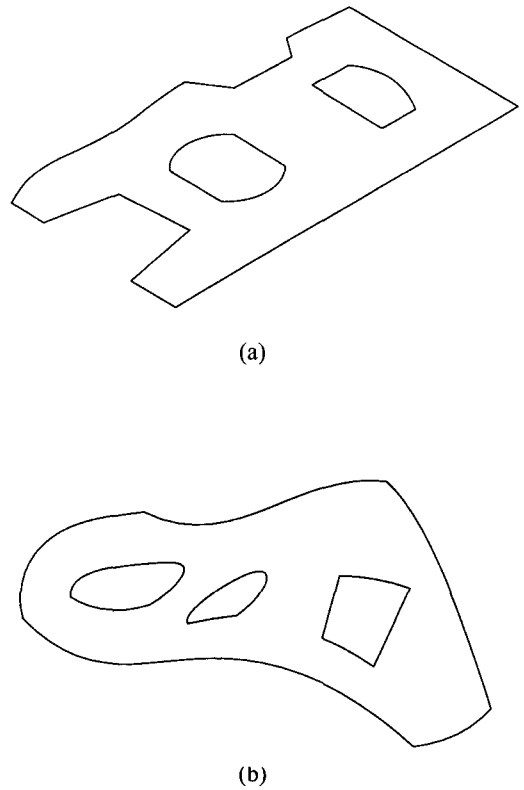


Fig. 2 Classification of trimmed surface; (a) Trimmed planar surface, (b) Trimmed curved surface

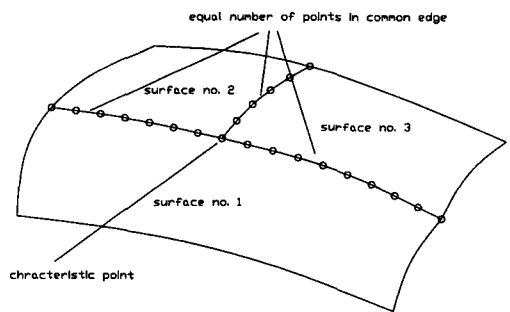


Fig. 3 Pre-Treatment of boundary edges between surfaces

3. 유한요소 자동생성

3.1 경계곡선의 전처리 알고리즘

종래의 자동 유한요소 생성에 관한 연구 결과들을 살펴보면 곡면들이 수백 개, 수천 개 모여 있는 경우 곡면 별로 유한요소를 생성하므로 각 곡면간의 접합부에서는 생성된 요소의 절점들이 일치하지 않게 되고 따라서 각 곡면에서 생성된 요소들의 절점들의 위치를 많은 시간과 단순노동을 요하는 수작업에 의하여 일일이 변경시켜 주어야 하는 문제점이 있었다. 이는 해석자에게 과중한 부담이 되어 왔고 창의적인 설계작업에 투여해야 할 시간을 낭비하는 결과를 초래하였다. 본 연구에서는 이를 해결하기 위해 Fig.3 과 같이 유한요소 생성 전에 모든 곡면들간의 경계곡선상의 요소 절점수를 일치시키는 효율적인 경계곡선 전처리 알고리즘을 개발하였다. 프리곡면은 물론 트림곡면의 모든 경계곡선들상의 특성점(characteristic points)들을 구해서 경계곡선들의 수치 데이터와 더불어 데이터 베이스에 저장시킨 후 곡면 단위로 유한요소 생성시 각 곡면의 경계곡선들을 모든 특성점과의 수직투영을 통해서 일련의 새로운 곡선들의 집합으로 변환시킨다. 이렇게 구해진 각 곡선들의 실장은 다음과 같은 식으로 표현될 수 있다.

$$L = \sum_{i=1}^{n-1} \int_0^1 r_i(t) dt \quad (5)$$

여기서 n 은 곡선을 이루는 3차원 점의 개수이고 $r_i(t)$ 는 곡선의 각 세그먼트로서 본 연구에서는 3차 퍼거슨 곡선 보간 (cubic Ferguson's curve fitting) 방법^[12,13]을 이용하여 복합곡면을 만든 후 식(5)에 (Gauss 수치적분을 적용하여 각 단위 곡선들의 실장의 합을 구함으로써 각 곡선들의 실장을 구하였다. 이렇게 얻어진 각 곡선들의 실장과 현재의 곡선에 요구되는 절점들간의 갯수로부터 3차원 곡선상에서 각 절점들간의 간격 δ 는 다음과 같이 구해진다.

$$\delta = \frac{L}{m-1} \quad (6)$$

여기서 L 은 식(5)로부터 구해진 현재 곡선의 실장

이고, m 은 요구되는 절점의 개수를 나타낸다. 절점들 간의 간격이 δ 가 되도록 복합 곡선상에서 새롭게 m 개의 3차원 점을 다음과 같은 뉴턴-랩슨 법(Newton Raphson Method) 및 수치 적분의 혼합적 수치 해법^[14]에 의해 구하였다. 식 (5)를 통한 복합 곡선의 실장 계산 시 각 단위 곡선들의 실장이 이미 구해져 있고 현재 구하고자 하는 n 번째 절점까지는 실장이 $(n-1)\delta$ 가 되므로 이 n 번째 절점이 위치하게 될 단위 곡선은 쉽게 찾을 수 있다. 위치하게 될 단위 곡선이 k 번째라 하면 다음과 같은 매개변수 t 로 구성된 비선형 방정식을 풀어서 n 번째 절점의 정확한 위치를 구할 수 있다.

$$F(t) = \int_0^t r_x(t) dt - \Delta L = 0 \quad (7)$$

여기서

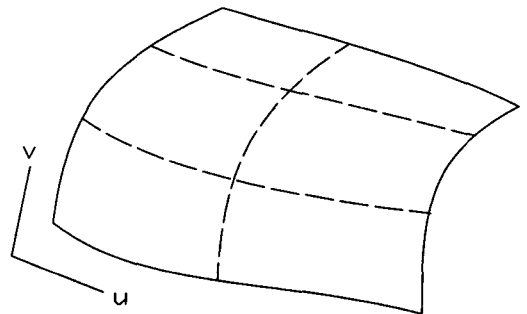
$$\Delta L : (n-1)\delta - L_k$$

L_k : $(k-1)$ 번째 까지의 단위곡선 실장의 합이다.

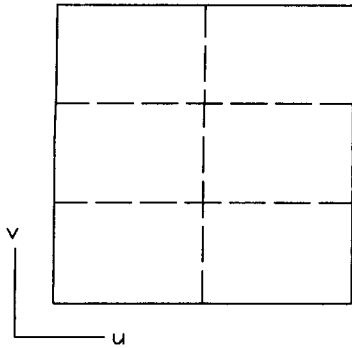
식 (7)의 근을 구하기 위한 뉴턴-랩슨 반복 계산식은 다음과 같이 표현된다.

$$t_{\text{new}} = t_{\text{old}} - \left[\frac{F}{\frac{dF}{dt}} \right]_{t=t_{\text{old}}} \quad (8)$$

즉 (8)의 반복식을 통하여 실장이 ΔL 이 되는 매개변수 t 의 값을 해당 단위 곡선 내에서 구하게 된다.



(a)



(b)

Fig. 4 Mapping of three-dimensional surface to two-dimensional domain; (a) Surface description in 3-D space, (b) Surface description in u, v 2-D domain

3.2 쿼드트리 (Quadtree)

쿼드트리는 2 차원 영역을 분할하는 기법으로 영상처리 (image processing), 2 차원에서의 유한요소 생성 등 다양한 용도로 사용되어 왔는데 본 연구에서 다루고자 하는 일반적인 3 차원 트림곡면도 Fig.4 에서와 같이 두 매개변수 u, v 에 관해서는 2 차원 영역으로 간주할 수 있다는 생각에서 국부적인 요소 밀도의 조절이 가능하도록 쿼드트리 기법을 도입하였다. Fig.5 는 쿼드트리의 형성과정을 보여주고 있는데 주어진 2 차원 형상을 충분히 둘러싸는 직사각형 또는 정사각형을 형성하여 이것의 가로, 세로를 각각 이등분 시켜 가면서 궁극적으로는 주어진 2 차원 형상을 직사각형 또는 정사각형의 합집합으로 근사 시키는 방법이다^[15,16]. 쿼드트리를 만들어가는 과정은 다음과 같다.

주어진 2 차원 형상을 포함하는 최소크기의 사각형을 구한다. 이 사각형은 쿼드트리의 뿌리 (root)에 해당한다. 그런 다음 이 사각형을 분할하는데 한번의 분할을 통하여 가로, 세로를 각각 이등분함으로써 4 개의 작은 사각형을 만들게 된다. 이렇게 분할되어 생성된 사각형요소를 사분체 (quadrant)라 한다. 사분체가 생겨나면 그것이 대상 물체에 의해 완전히 포함되는지, 부분적으로 포함되는지, 아니면 외부에 있는지를 파악하여 각각 포함 (interior), 부분포함 (partial), 외부(exterior)형

으로 분류하게 된다. 이렇게 분할을 반복 수행하여 최소 사분체의 크기가 기준크기 이하로 되면 분할을 마치게 된다.

쿼드트리의 자료 구조는 한번 분할에 4 개씩의 일정한 자식 (child)들이 생겨나는 것 외에는 일반적인 트리구조와 같은 구조를 갖고 있다. 따라서, 일반적인 트리구조와 마찬가지로 뿌리에 대한 현재 사분체의 깊이를 레벨 (level)이라 하고 전체 트리의 깊이를 depth 라 한다. 쿼드 트리는 일반트리와는 다른 필드 (field)를 갖고 있는데, 그것은 부모사분체에 대한 자식 사분체들의 위치를 정의하는 필드이다. 쿼드트리는 기하학적인 정보를 표현하고 있으므로 이러한 사분체들간의 위치 관계를 나타낼 필요가 있다^[17,18].

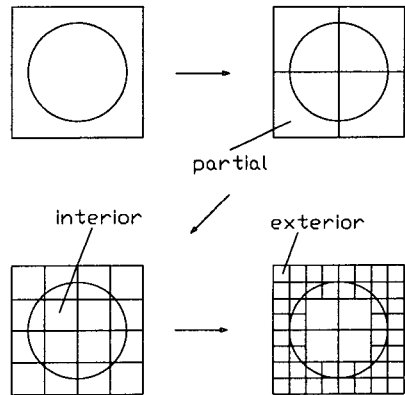


Fig. 5 Procedure of quadtree generation

3.3 쿼드트리를 이용한 유한요소 생성

본 연구에서는 유한요소 생성 방법 중 격자이용법^[19]을 기본으로 하고 국부적인 요소의 밀도 조절이 가능하도록 쿼드트리 기법을 혼합한 방법을 제안하고자 한다. 즉 곡면의 u, v 매개변수 영역에 요소길이가 형상오차를 고려한 가상의 격자를 씌운 후 완전히 포함되는 내부의 격자는 그대로 두고 경계곡선에 교차되는 사분체는 최소 사분체의 크기가 가장 가까운 경계곡선상의 두 절점사이의 길이 이하로 될 때까지 분할을 계속한 후 경계 곡선들과 내부의 격자 사이에는 Delaunay 삼각화 방법^[20-22]을 이용하여 요소를 생성한다. 최종적으로 요소 평활화 (mesh smoothing)작업을 통하여 요소의 형질을 개선하게 된다.

전체 알고리즘은 크게 두 부분으로 구성되는데 첫번째는 u, v 매개변수 영역에 경계곡선상의 요소길이와 형상오차를 고려하여 쿼드트리 기법을 통하여 절점을 생성하는 단계이고 두번째는 생성된 절점으로부터 요소를 생성하고 평활화하는 단계이다. 이를 전체적인 순서도로 표현하면 Fig.6 과 같다.

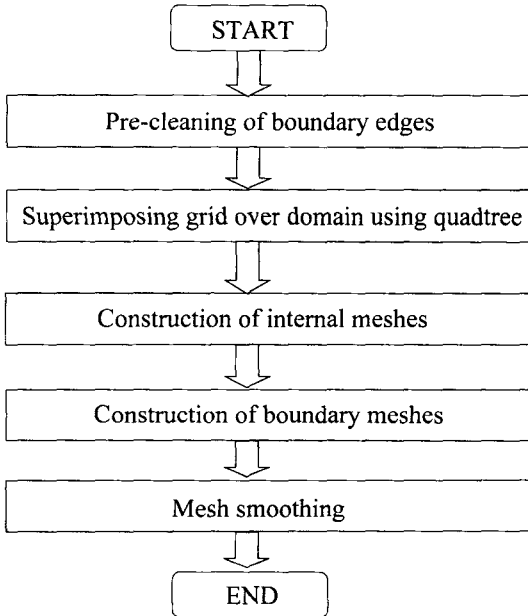


Fig. 6 Overall procedure of mesh generation

3.3.1 절점생성 알고리즘

사용자의 목적 및 의도에 부합되는 절점 생성을 위하여 최대 요소 길이와 원하는 형상오차를 입력변수로 처리하였다. Fig.7 에서와 같이 최대 요소길이 및 형상 오차를 작게 하면 할수록 가상 격자 수는 증가하게 된다. 실제 산업현장에서 작성되는 곡면 데이터에는 Fig.8 과 같이 수치적으로 크기가 매우 작은 패치(patch)들이 많이 존재하는데 본 연구에서는 필요이상의 절점이 생성되지 않도록 패치 단위가 아닌 전체 영역으로부터 국부적으로 세분화해 가는 방법을 적용하였다. u 또는 v 방향의 곡선상에서 먼저 최대 요소 길이가 만족되 때까지 절점을 생성한 후 형상오차가 만족되지 않는 구간은 만족될 때까지 절점을 추가적으로 삽입하

게 된다.

u 방향 곡선인 경우는 v 는 상수로 일정하게 되고 v 방향 곡선인 경우는 u 가 일정하게 되므로 형상오차 ϵ 는 다음과 같은 비선형 방정식을 풀어서 구할 수 있다.

i) u 방향 곡선인 경우 ($v=v_1$)

$$F(u, v_1) = 2 \sum_{\alpha=1}^3 (X_{\alpha}(u, v_1) - P_{\alpha}) \frac{\partial X_{\alpha}(u, v_1)}{\partial u} = 0 \quad (9a)$$

$$u_{i+1} = u_i - F(u_i, v_1) / \frac{\partial F(u_i, v_1)}{\partial u} \quad (9b)$$

ii) v 방향 곡선인 경우 ($u=u_1$)

$$F(u_1, v) = 2 \sum_{\alpha=1}^3 (X_{\alpha}(u_1, v) - P_{\alpha}) \frac{\partial X_{\alpha}(u_1, v)}{\partial v} = 0 \quad (10a)$$

$$v_{i+1} = v_i - F(u_1, v_i) / \frac{\partial F(u_1, v_i)}{\partial v} \quad (10b)$$

식 (10a)와 식(10a)에서 P_{α} 는 두 절점 사이를 잇는 직선의 중간점으로서 식(9a)와 식(10a)의 근을 식(9b) 또는 식(10b)의 뉴턴-랩슨 반복계산식을 이용하여 구하게 되면 중간점 P_{α} 에 가장 근접한 곡선상의 점, 즉 수직투영법 Q_{α} 를 찾게 되므로 형상오차 ϵ 를 다음과 같이 표현할 수 있다.

$$\epsilon = \sqrt{\sum_{\alpha=1}^3 (P_{\alpha} - Q_{\alpha})^2} \quad (11)$$

이와 같은 방법으로 각 패치의 중앙을 지나는 u, v 두 방향으로의 곡선들에 대해서 절점을 구한 후 가장 많은 절점이 생성된 두개의 곡선상의 u, v 값을 최종적인 가상 격자점들로 취하였다.

이상의 과정은 3 차원 곡면상에서 이루어지지만 u 또는 v 값들은 이미 구해져 있으므로 Fig.9 에서와 같이 u, v 에 관한 2 차원 영역의 문제로 단순화 시킬 수 있다. 이렇게 전체적인 요소밀도의 조절이 완료되면 최소 사분체의 크기가 가장 가까운 경계 곡선상의 두 절점사이의 길이 이하로 될 때까지 쿼드트리 기법을 이용하여 사분체를 분할해 감으로써 국부적인 요소밀도를 조절하게 된다.

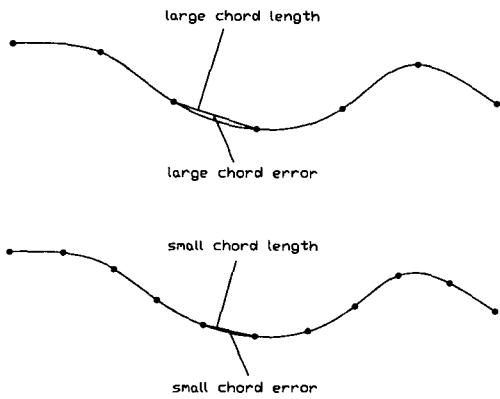


Fig. 7 Effect of chord length and chord error on the generation of nodes

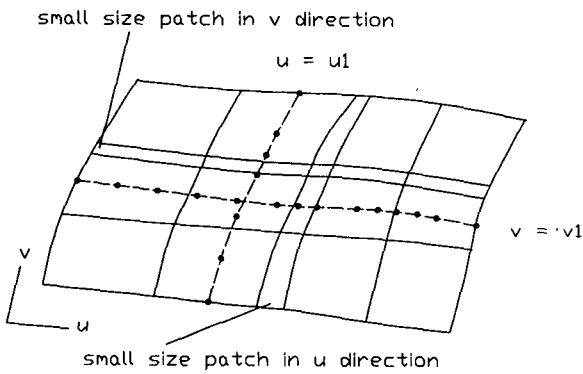


Fig. 8 Generation of nodes in surface considering chord length and chord error

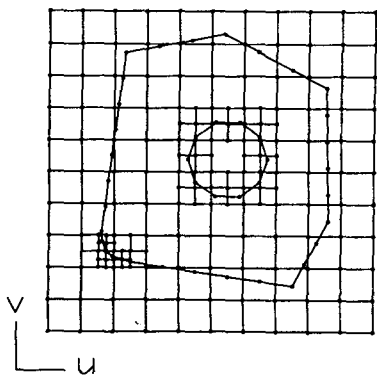
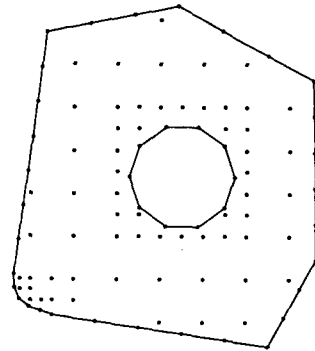
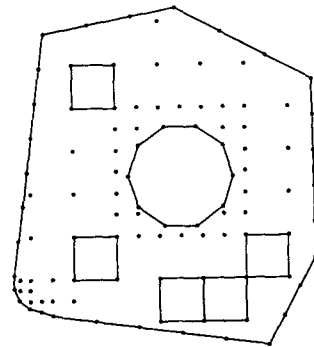


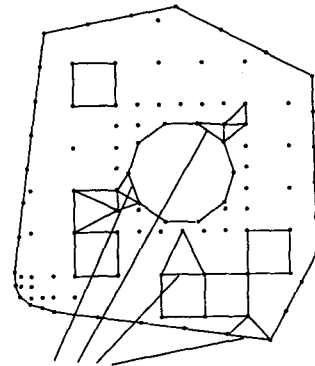
Fig. 9 Superimposing grid over domain using quadtree



(a)



(b)



(c)

Fig. 10 Schematic diagram illustrating the procedures mesh generation; (a) Filtering of interior nodes, (b) Construction of internal meshes using root quadrant, (c) Delaunay triangulation

3.3.2 요소생성 알고리즘

앞에서 생성된 절점들을 연결하여 요소를 생성하는 알고리즘은 크게 두 단계로 분류할 수 있다. 첫번째는 일차적 사분체 (root quadrant) 중 사용 영역에 포함되어 있으면서 네 방향에 인접해 있는 사분체가 분할되어 있지 않은 경우 이 사분체를 그대로 유한요소격자로 구성하는 단계이며, 두번째 단계는 경계곡선상에 형성된 절점들에 의해 생성된 직선들과 내부 요소의 바깥 경계면을 이루는 직선들을 기준선 (base line)으로 하고 나머지 절점들을 이용하여 Delaunay 삼각화 방법을 이용하여 경계곡선과 내부격자사이의 요소들을 생성하는 단계이다 (Fig.10).

Fig.10a 과 같이 u, v 매개변수 영역에 평균요소 길이와 요구되는 형상오차를 만족하는 가상 격자점들을 찍은 후 모든 격자점들에 대하여 바깥쪽 경계조건과 안쪽 경계곡선상의 절점들의 수치 데이터를 토대로 트림곡면의 사용 영역에 포함될지의 여부를 계산한다. 한 점 Q가 점열 (P_1, P_2, \dots, P_n) 로 구성되어 있는 경계곡선 안에 포함되어 있는지의 여부는 다음과 같은 식으로 구할 수 있다.

$$\theta = \left| \sum_{i=1}^{n-1} \angle P_i Q P_{i+1} \right| \quad (12)$$

식(12)에서 구한 θ 가 2π 이면 포함되어 있는 것이고 θ 가 0이 되면 바깥쪽에 있는 것이므로 바깥쪽 경계곡선 및 안쪽 경계곡선이 존재하는 일반적인 트림곡면의 경우 바깥쪽 경계곡선에 대해서는 2π , 안쪽 경계곡선이 대해서는 0이 되어야 사용 영역에 존재하게 되는 것이다. 이상과 같이 각 절점의 포함여부가 결정되면 내부에 존재하는 네 절점은 두개의 삼각형 또는 한 개의 사각형 요소로 되고 경계곡선 상에 형성된 절점들에 의해 분할된 직선들과 내부요소의 바깥 경계면을 이루는 직선들은 Delaunay 삼각화를 위한 기준선으로 등록된다. 각각의 기준선에 대해 꼭지점으로 선택되어 삼각형을 형성할 수 있는 점들 중에서 그 삼각형이 외접원 내에 다른 절점이 존재하지 않는 점을 선택하여 삼각형을 구성한다. 이렇게 생성된 삼각형의 새로운 두 변을 새로운 기준선으로 등록된다. 그러나, 기준선이 바깥쪽 경계곡선 또는 안쪽 경계곡선을 나타내는 선분이거나 이미 존재하는 삼

각형의 한 변이라면 해당 기준선은 기준선의 집합에서 제외된다. 이와 같은 방법으로 기준선의 집합에 기준선이 하나도 남지 않을 때까지 삼각화를 계속 해 나간다 (Fig.10c). Delaunay 삼각화법은 각 삼각형의 외접원내에 다른 절점이 존재하지 않는 점을 선택하여 삼각형을 구성하는 방법으로서 삼각형들의 최소각을 최대로 하므로 날카로운 요소의 생성을 피할 수 있다 (Fig.11). 만약 선택된 꼭지점에 의해 생성될 삼각형이 외접원내에 다른 절점이 존재하지 않는 점이 없는 경우에는 생성될 삼각형의 최소내각이 최대가 되는 절점을 선택하여 삼각형을 생성하였다^[23].

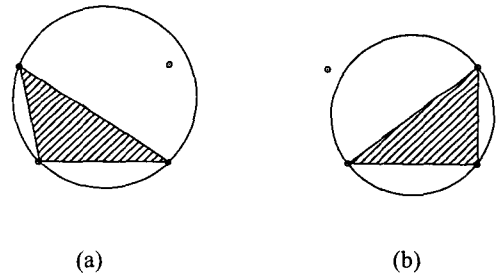


Fig. 11 Basic concept of Delaunay triangulation; (a) Mesh of bad quality, (b) Mesh of good quality

3.3.3 요소 평활화 (mesh smoothing)

요소의 질 (mesh quality)이 좋지 않은 경우 해석을 성공적으로 수행한다 하더라도 해석 결과의 정확도 및 신뢰도 관점에서 좋은 결과를 기대하기 곤란한 경우가 자주 발생하는데 따라서 가능한 한 계산을 수행하기 전에 요소의 질을 개선할 필요가 있다. 이와 같이 요소들의 절점의 위치를 재조정함으로써 요소의 질을 향상시키는 작업을 요소 평활화라고 한다. 재위치 과정을 전 절점에 대해 수회 반복 수행하여 각 절점들의 위치가 수렴성을 가질 때까지 계속하게 되는데 본 연구에서는 다음과 같이 각 절점과 연결된 다각형들의 무게중심으로 이동시키는 방법을 사용하였다^[8].

$$P = \frac{\sum_{i=1}^N A_i C_i}{\sum_{i=1}^N A_i} \quad (13)$$

여기서

- P: 절점의 새로운 위치
- A_i: i 번째 다각형 면적
- C_i: i 번째 다각형의 무게중심
- N: 연결된 다각형의 개수

이다.

이때 새로이 이동된 절점은 원래 곡면상에 있지 않으므로 곡면위로 수직 투영하여야 한다 (Fig.12). 절점의 좌표값을 P_α라 하면 절점과 가장 가까운 곡면상의 점은 u, v로 구성된 비선형 방정식을 풀어서 구할 수 있다.

$$F_1(u, v) = 2 \sum_{\alpha=1}^3 (X_{\alpha}(u, v) - P_{\alpha}) \frac{\partial X_{\alpha}(u, v)}{\partial u} = 0 \quad (14a)$$

$$F_2(u, v) = 2 \sum_{\alpha=1}^3 (X_{\alpha}(u, v) - P_{\alpha}) \frac{\partial X_{\alpha}(u, v)}{\partial v} = 0 \quad (14b)$$

여기서 X_α(u, v)는 곡면의 한 패치의 x, y, z 성분 에 관한 다항식을 나타내며 식(2)로부터 구할 수 있다. 식(14)의 근을 구하기 위한 뉴턴-랩슨 반복 계산법은 다음과 같이 표현된다.

$$\begin{bmatrix} u_{i+1} \\ v_{i+1} \end{bmatrix} = \begin{bmatrix} u_i \\ v_i \end{bmatrix} - J^{-1} \begin{bmatrix} F_1(u_i, v_i) \\ F_2(u_i, v_i) \end{bmatrix} \quad (15)$$

여기서

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial u} & \frac{\partial F_1}{\partial v} \\ \frac{\partial F_2}{\partial u} & \frac{\partial F_2}{\partial v} \end{bmatrix}$$

이다. 본 연구에서 사용된 최근접 패치 선별법 및 반복계산을 위한 초기치의 계산 등은 Yoo 등^[24]이 제안한 방법을 이용하였다.

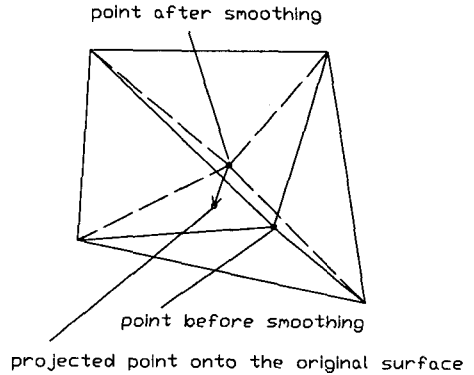


Fig. 12 Schematic diagram illustrating mesh smoothing

4. 적용예

본 연구를 통하여 개발된 유한요소 자동생성 알고리즘의 효용성 및 타당성을 검증하기 위하여 실제 곡면 모델에 대하여 유한요소생성을 해 보았다. 곡면 모델에 대한 데이터는 CAD 상용 시스템의 하나인 CATIA를 이용하여 모델링한 후 본 연구에서 채택한 매개변수에 대한 다항식 형태의 곡면 데이터로 변환시켜 사용하였다^[25,26].

4.1 트림 평면 (Trimmed planar surface)의 유한요소형성

제한된 제반 알고리즘의 타당성을 확인하기 위하여 Fig.13과 같은 내부 홀이 존재하는 트림 평면에 대하여 요소를 생성해 보았다. Fig.13(a)는 쿼드트리 및 요소평활화 기법을 적용하지 않은 경우를 보여주고 있는데 바깥쪽 경계곡선 및 안쪽 경계곡선 부근에서 요소 밀도의 분포가 급격히 변하고 내각의 크기가 작은 질이 좋지 않은 요소가 다수 발생하고 있다. Fig.13(b)는 쿼드트리를 적용하였을 때의 결과를 나타내고 있는데 경계곡선 부근에서 요소의 밀도가 유연하게 변하고 있음을 볼 수 있다. 그러나, 경계곡선 부근에 여전히 질이 좋지 않은 요소가 상당수 존재하고 있다, 이러한 질이 좋지 않은 요소들은 요소평활화 작업을 통하여 Fig.13(c)에서 보는 바와 같이 양질의 요소로 개선될 수 있다. 트림평면의 경우는 요소평활화시 절점들의 위치이동과 더불어 수행되는 수직 투영시 기본면이 평면이므로 기본면이 곡면인 경우보다 평활화를 위한 수치해석 알고리즘이 보다 강건하

고 계산시간에 있어 효율적이다.

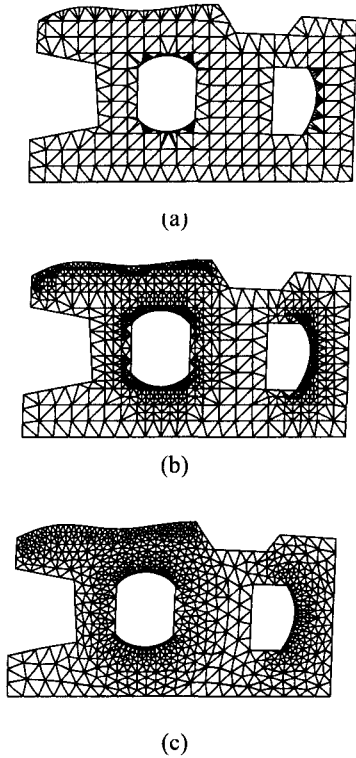
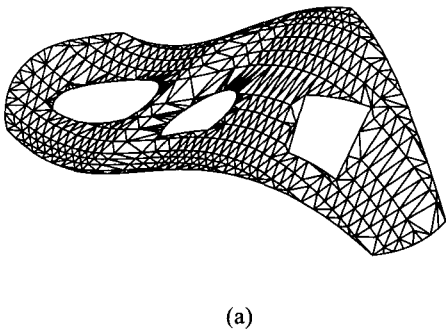


Fig. 13 Mesh generation in the trimmed planar surface with multiple holes; (a) the case using no quadtree and no smoothing, (b) the case using quadtree and no smoothing, (c) the case using quadtree and smoothing



(a)

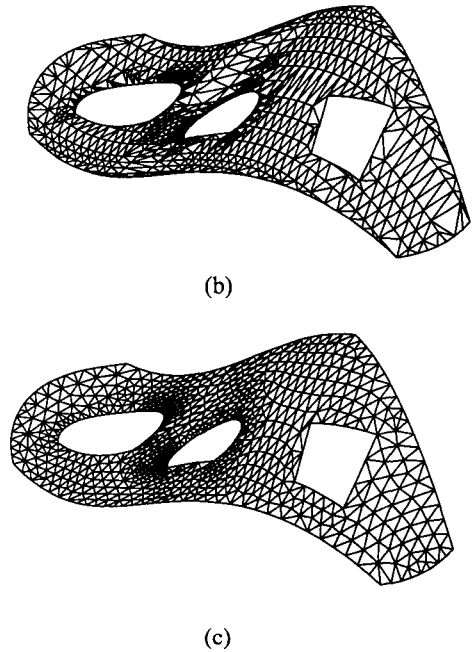
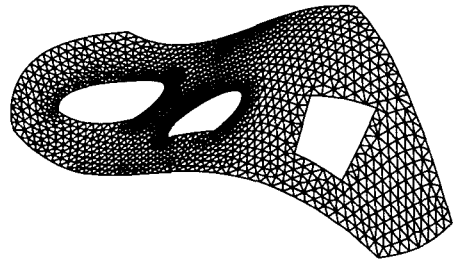


Fig. 14 Mesh generation in the trimmed curved surface with multiple holes (chord length : 30 mm, chord error : 0.05 mm); (a) the case using no quadtree and no smoothing, (b) the case using quadtree and no smoothing, (c) the case using quadtree and smoothing

4.2 트림 곡면 (Trimmed curved surface)의 유한요소 생성

기본면이 곡면인 트림곡면의 유한요소 생성을 통하여 요소 길이와 형상오차 등의 입력변수의 영향 및 트림곡면시의 쿼드트리 기법 및 요소평활화 알고리즘의 효율성을 검증해 보았다. Fig.14(a)는 최대 요소길이가 30mm 이고 형상오차는 0.05mm 일 때 쿼드트리 및 요소평활화 기법을 적용하지 않은 경우를 보여주고 있다. 역시 경계곡선 부근에서 요소 밀도의 분포가 유연하지 못하고 내각의 크기가 매우 작은 질이 좋지 않은 요소가 많이 생성되어 있다. 해석하고자 하는 문제의 특성에 따라서는 이러한 유한요소는 사용될 수 없을 수도 있다. Fig.14(b)는 쿼드트리를 적용한 후의 결과를 보여주고 있고 Fig.14(c)는 요소평활화 작업을 끝내고 난 후의 최종적인 모습을 보여주고 있다.

경계 곡선 부근의 요소 밀도의 분포가 유연화되어 있고 요소의 질도 많이 개선되어 있음을 볼 수 있다. 그러나 이와 같이 형상오차는 너무 작게 하고 최대 요소 길이는 너무 크게 하면 요소평활화 전의 요소들의 크기 분포가 고르지 않고 요소들의 질도 매우 좋지 않을 수 있다. 이러한 경우는 요소 평활화 작업을 통해서도 요소들의 질이 개선되는 정도에 한계가 있을 수 있는데 이러한 문제는 최대요소길이 및 형상오차를 적절히 선정함으로써 해소 시킬 수 있다. Fig.15(a)는 최대 요소길이가 5mm 이고 형상오차는 0.05mm 일 때 쿼드트리 및 요소평활화 기법을 적용하지 않았을 때의 요소 생성결과를 보여주고 있는데 곡면의 곡률 특성상 쿼드트리 및 요소 평활화 기법을 사용하지 않았는데도 매우 균질의 요소들을 보여주고 있다. Fig.15(b)에서는 쿼드트리 기법을 사용함으로써 경계곡선 부근의 곡률이 큰 부분에서의 요소 밀도를 국부적으로 조절하는 것이 가능하였으며 Fig.15(c) 처럼 최종적인 요소 평활화 작업을 통해서 정밀 구조해석이 가능할 정도의 양질의 요소를 생성할 수 있었다.

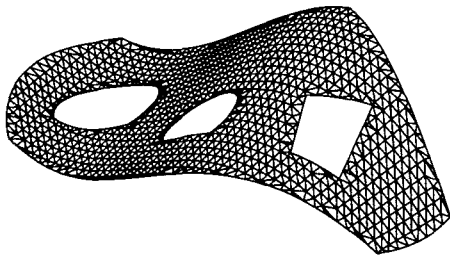


(c)

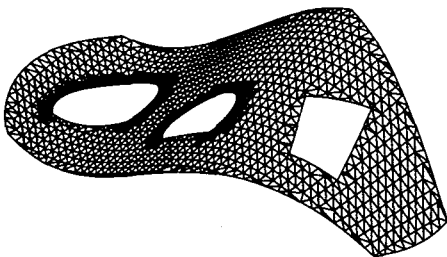
Fig. 15 Mesh generation in the trimmed curved surface with multiple holes (chord length : 5 mm, chord error : 0.05 mm) ; (a) the case using no quadtree and no smoothing, (b) the case using quadtree and no smoothing, (c) the case using quadtree and smoothing

4.3 복합 트림곡면 (Compound trimmed curved surface)의 유한요소 생성

본 연구에서 제안한 알고리즘 중 특히 곡면들간의 요소 연결성을 보장하는 경계곡선 전처리 알고리즘의 적합성을 확인해 보기 위하여 총 14 개의 트림곡면으로 구성된 실제 차체판넬의 하나인 프론트 펜더 (front fender)에 대하여 요소 생성을 해 보았다. 형상 오차는 0.1 mm, 최대 요소길이는 20 mm로 하였으며 쿼드트리 및 요소평활화 기법을 적용하였다. Fig.16 에서 보는 것과 같이 곡률이 작은 부분에서부터 큰 부분으로의 요소 밀도가 유연화 되어 있고 특히 곡면들간의 요소 연결성 (mesh connectivity)이 보장됨으로써 사용자의 후처리 작업이 전혀 필요 없는 완전한 양질의 요소를 얻을 수 있었다. Fig.17 은 8 개의 트림평면과 24 개의 트림곡면으로 구성된 복합곡면에 대하여 요소를 생성해 본 결과이다. 이러한 유형의 곡면 모델은 tetrahedron, hexahedron 등의 3 차원 유한요소를 생성하기 위한 입력자료로서 종종 이용되는데 3 차원 유한요소를 생성하기 앞서 요소연결성 및 형상의 정확도를 고려한 곡면상의 요소 (surface mesh)의 생성이 선결 과제이다. 그림에서 보는 것 처럼 곡면들간의 요소들의 위치가 경계곡선상에서 정확히 일치하고 있으며 형상의 정확도도 충실히 고려되어 있고 생성된 요소들의 질도 양호함을 확인할 수 있었다.



(a)



(b)

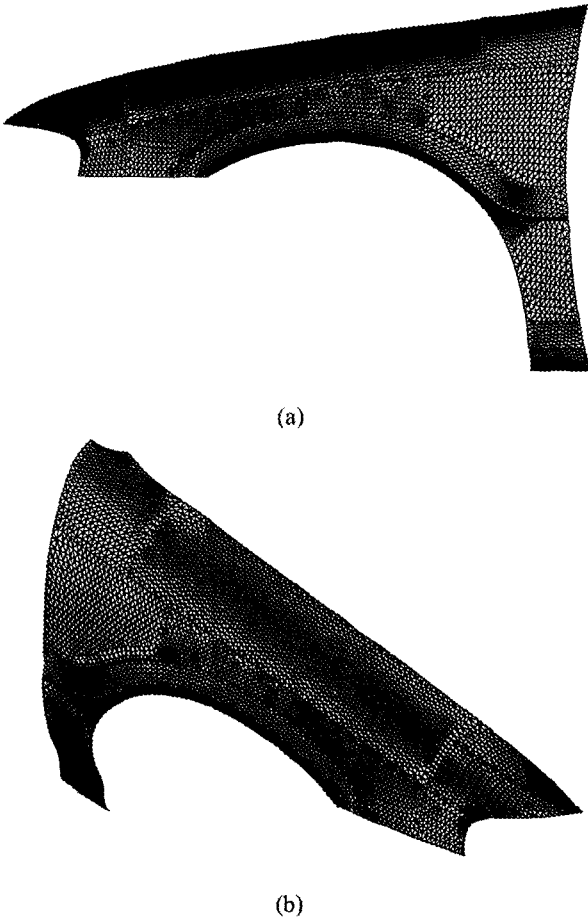


Fig. 16 Mesh generation in the multiple trimmed curved surfaces; (a) Top view, (b) Side view

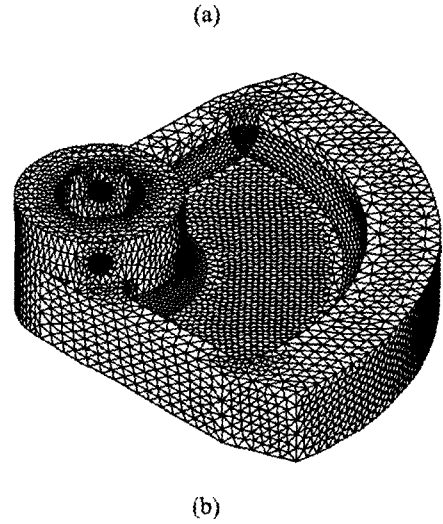
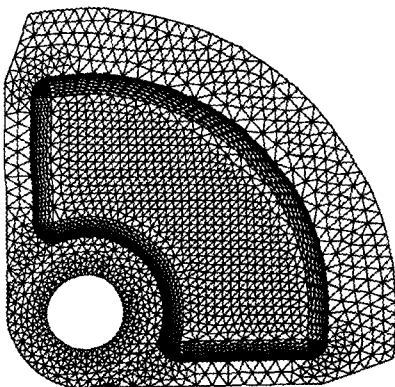


Fig. 17 Mesh generation in the compound surface composed of trimmed planar surfaces and trimmed curved surfaces; (a) Top view, (b) Side view

5. 결론

본 연구에서는 일반적인 형상의 트림곡면으로부터 유한요소를 자동 생성하는 알고리즘을 개발하였다. 전체적인 요소밀도 조절을 위하여 최대요소길이, 형상오차 등의 입력변수를 선정하였으며 경계곡선 부근의 국부적인 요소 밀도 조절을 위하여 쿼드트리 기법을 도입하였다. 한 곡면내의 유한요소 생성을 위하여 격자 이용법과 Delaunay 삼각화법의 복합적인 방법을 제안함으로써 요소 생성의 효율성 및 신뢰성을 향상시켰다. 유한요소 해석자의 큰 부담이 되어왔던 곡면들 간의 요소 연결성 확보를 위하여 경계곡선 전처리 알고리즘을 개발하여 실제 판넬등의 복합곡면에 적용해 본 결과 해석자의 후처리 작업이 전혀 필요 없을 정도의 양질의 요소를 얻을 수 있었다. 아울러 일반적인 트림 곡면에 적용할 수 있는 요소평활화 알고리즘을 개발하여 적용해 본 결과 요소의 질이 크게 문제되지않는 박관성형해석용 툴(tool)은 물론 사출성형해석, 구조해석 등 양질의 요소가 요구되는 문제에도 적용이 가능할 정도의 좋은 요소를 생성할 수 있음이 확인 되었다.

참고문헌

1. T.C.Woo and T.Thomasma, "An algorithm for generating solid elements in objects with holes," *Compt.&Struct.*, Vol. 18, pp. 333-342, 1984.
2. J.C.Cavendish, "Automatic triangulation of arbitrary planar domains for the finite element method," *Int. J. Numer. Meth. Eng.*, Vol. 8, pp. 679-696, 1974.
3. J.M.Nelson, "A triangulation algorithm for arbitrary planar domains," *Appl. Math. Modeling*, Vol. 2, pp. 151-159, 1978.
4. A.Bykat, "Design of recursive shape controlling mesh generator," *Int. J. Numer. Meth. Eng.*, Vol. 19, pp. 1375-1390, 1983.
5. J.A. Talbert and A.R. Parkinson, "Development of an automatic two-dimensional finite element mesh generator using quadrilateral elements and Bezier curve boundary definition," *Int. J. Numer. Meth. Eng.*, Vol. 29, pp. 1551-1567, 1990.
6. N. Kikuchi, "Adaptive grid-design methods for finite element analysis," *Comput. Methods in Applied Mech. Eng.*, Vol. 55, pp. 129-160, 1986.
7. R.J. Meyers and T.D. Blacker, "Generalized 3-D paving : an automated quadrilateral surface mesh generation algorithm," *Int. J. Numer. Meth. Eng.*, Vol. 39, pp. 1475-1489, 1996.
8. 이영규, "소성가공공정의 유한요소해석을 위한 3 차원 자동격자구성기법," 한국과학기술원 기계공학과 박사학위논문, 1998.
9. CATIA User's manual, Dassault Systems, Inc., 1996.
10. B.K. Choi, "Surface modeling for CAD/CAM," Elsevier Science Publishers B.V., 1991.
11. I.D. Faux and M.I. Pratt, "Computational geometry for design and manufacture," Ellis Horwood Ltd., 1979.
12. D.F. Rogers and J.A. Adams, "Mathematical elements for computer graphics," McGrawHill, 1990.
13. 이건우, "컴퓨터그래픽과 CAD," 영지문화사, 1994.
14. W.H.Press, B.P.Flannery, S.A.Teukolsky and W.T. Vetterling, "Numerical recipes: The art of scientific computing," Cambridge University Press, 1986.
15. M.A. Yerry and M.S. Shephard, "A modified quadtree approach to finite element mesh generation," *IEEE Comput. Graph. & Appl.*, Feb., pp. 39-46, 1983.
16. N. Nakajima, S. Tokumasu and Y. Kunitomo, "Feature-based heuristics for finite element meshing using quadtree and octrees," *Computer-Aided Design*, Vol. 24, pp. 677-690, 1992.
17. H. Samet, "Neighbor finding technique for images represented by quadtrees," *Comput. Graph. & Image Proc.*, Vol. 18, pp. 37-57, 1982.
18. 정용호, "사면체 Octree 에 근거한 3 차원 유한요소의 자동생성," 서울대학교 대학원 박사학위 논문, 1993.
19. S.H. Lo and C.K. Lee, "Generation of gradation meshes by the background grid technique," *Comput. & Struct.*, Vol. 50, pp. 21-32, 1994.
20. D.F.Watson, "Computing the n-dimensional Delaunay tessellation with application to Voronoi polygons," *Comp. J.*, Vol. 24, pp. 167-172, 1981.
21. S.H. Lo, "Delaunay triangulation of non-convex planar domains," *Int. J. Numer. Meth. Eng.*, Vol. 28, pp. 2695-2707, 1989.
22. T.P. Fang and L.A. Piegl, "Algorithm for Delaunay triangulation and convex-hull computation using a sparse matrix," *Computer-Aided Design*, Vol. 24, No. 8, pp. 425-436, 1992.
23. B.K. Choi, H.Y. Shin, Y.I. Yoon and J.W. Lee, "Triangulation of scattered data in 3D space," *Computer-Aided Design*, Vol. 20, No. 5, pp. 239-248, 1988.
24. D.J. Yoo, I.S. Song, D.Y. Yang and J.H. Lee, "Rigid-plastic finite element analysis of sheet metal forming processes using continuous contact treatment and membrane elements incorporating bending effect," *Int. J. Mech. Sci.*, Vol. 36, pp. 513-546, 1994.
25. Advanced Programming, Dassault Systems, Inc., 1996.
26. IUA Application Interface, Dassault Systems, Inc., 1996.