형상 모델러의 자료구조에 의한 수정 Delaunay 삼각화

채 은미[*1], 사 종엽[*2]

Modified Delaunay Triangulation Based on Data Structure of Geometric Modeller

E.-M. Chae and J.-Y. Sah

A modified Delaunay triangulation technique is tested for complicated computational domain. While a simple geometry, both in topology and geometry, has been well discretized into triangular elements, a complex geometry having difficulty in triangulation had to be divided into small sub-domains of simpler shape. The present study presents a modified Delaunay triangulation method based on the data structure of geometric modeller. This approach greatly enhances the reliability of triangulation, especially in complicated computational domain. We have shown that efficiency of Delaunay triangulation can be much improved by using both the GUI (Graphic User Interface) and OOP (Object-Oriented Programming).

Key Words: 수정 Delaunay 삼각화(Modified Delaunay triangulation), 자료구조(Data structure), 형상 모델러(Geometric modeller)

1. 서 론

최근 컴퓨터 하드웨어의 발달과 수치해석 알고리즘의 발달로 인하여 실제적인 복잡한 형상을 직접 해석하고 설계하는 사례가 크게 증가하고 있다. 구조물의 해석 등에는 유한요소법이 널리 사용되어 왔으며, 전산유체역학 분야에 있어서도 유한요소법의 적용이 꾸준히 증가하고 있는 추세이다. 유한요소법은 작은 요소로 전체 계산영역을 분할하고 각 요소에 형상함수를 정의하여 응력이나 유동장을 계산하는 것으로써, 유한요소법에 의한 프로그램은 복잡한 형상의 문제에 적용하기에 편리하므로 여러 분야의 상용 소프트웨어를 개발하는데 널리 사용되고 있다. 그러나 유한요소법은 계산영역을 작은 요소로 분할하는 요소생성 과정이 필요하며 형상이 복잡한 계산영역의 경우에는 사용자에게 과중한 부담을 준다. 이러한 문제점을 해결하기 위해 유한요소를 자동으로 생성하기 위한 방법이 필요하다. 자동 격자 생성 알고리즘은 적절히 분포된 양질의 격자를 생성할 수 있는 신뢰성과 안정성을 지녀야 한다.

Lo[1]의 프론트 전진기법은 계산영역의 경계에서 분할하고자하는 내부로 프론트(Front)를 이동시키며 격자를 채워가는 방법이다. 내부로 격자를 생성해 나가는 과정에서 매번 프론트 간의 교차검사를 처리해야 하며 양질의 격자를 생성할 수 있는 기준이 모호하다.

Preparate Shamos[2]는 주어진 절점들에 대하여 등변 삼각형에 가장 가까운 유일한 삼각형 집합을 보장할 수 있는 Delaunay 성질을 격자 생성에 적용시켜 Delaunay 삼각화 방법을 제안하였다. 알고리즘 자체가 간단하고 수학적으로 잘 정의되어 있으므로, 프론트 전진기법에 비해 효과적이나, 영역의 경계를 보장하기 어려운 단점이 있다. 이것을 해결하기 위해 Georage[3]는 edge-swaping을 이용하여

*1 학생회원, 영남대학교 대학원
*2 정회원, 영남대학교 기계공학과

Delaunay 삼각형 요소를 국소적으로 수정함으로써, 특정영역의 Delaunay 성질을 제한하는 방법을 사용하였다. 그리고 Baker[4]는 경계절점의 절점밀도를 높이는 방법등을 제안하였다. 프론트 전진기법과 Delaunay 삼각화 이외의 자동격자 생성 방법으로는 계산영역을 정밀하게 표현할 수 있을때까지 공간분할을 통해 격자를 생성해가는 Quad/octree 방법[5]이 있으며 이 방법은 알고리즘이 간단하고 내부영역에서 양질의 격자를 얻을수 있으나 경계부분에서 격자가 불규칙적으로 분포하는 단점이 있다. 또한 Rebay[6]는, 삼각화 과정은 Delaunay 알고리즘을 사용하고 절점삽입 방법은 프론트 전진 기법을 사용하는 복합방법도 제시하였다.

본 연구에서는 Delaunay 삼각화를 기반으로 한 자동 비정렬격자 생성방법을 개발하였다. Delaunay 알고리즘은 삼각화 과정에서 계산영역의 경계를 고려하지 않음으로써 발생하는 문제이므로 Delaunay 삼각화의 신뢰도를 높이기 위해 형상 모델러에서 사용하고 있는 B-rep 방식의 자료 구조를 도입하였다. 데이터 구조의 위상정보와 기하정보를 격자생성 과정에 사용함으로써 Delaunay 삼각화 방법의 신뢰도를 높였다. 본 연구는 자료구조에 의한 수정 Delaunay 방법을 제안한 것으로써, Delaunay 방법에 관한 자세한 내용은 생략하기로 한다. Delaunay 방법에 대한 보다 구체적인 내용은 참고문헌[6-9]를 참조할 수 있다.

2. 자료구조

본 연구에서 사용된 자료구조는 크게 두가지로 나눌 수 있다. 첫째는 복잡한 형상의 기하 및 위상정보를 다루기 위한 형상정보 자료구조이고, 다른 하나는 생성된 삼각형 요소들과 절점들의 관계를 효율적으로 규정하기 위한 요소정보 자료구조이다.

2.1 형상정보 자료구조

복잡한 형상에 대하여 항상 신뢰성있게 격자를 구성하려면 격자를 생성할 영역의 위상정보와 기하정보들을 효율적으로 다룰 수 있어야 한다. 계산영역의 형상을 표현하기 위하여 본 연구에서는 Fig. 1과 같은 자료구조를 사용하였다. 계산영역을 먼저 블록들의 집합으로 구성한다. 블록은 다음과 같은 경우에 서로 분리하여 정의하는데, 영역이 서로 분리되어 있는 경우 각각을 서로 다른 블록으로 지정하며, 분리되어 있지 않더라도 두가지 이상의 특성을 지닌 영역이 서로 접하여 있는 경우 각각을 서로 다른 블록으로 지정한다. 그렇지 않은 경우 사용자가 내부 삼각형 요소의 조밀성을 조절하거나 기타의 목적으로 편의상 하나의 영역을 두 개 이상의 영역으로 분할 하고자 할 때 각각을 서로 다른 블록으로 지정할 수 있다. 블록은 루프(loop)들로 구성되며, 루프는 블록의 경계에 의하여 닫혀진 폐곡선을 의미한다. 블록이 single-connected 영역인 경우에는 외부 경계에 의한 하나의 외부 루프를 갖지만, 내부에 구멍이 있는 multi-connected 영역인 경우에는 외부 루프 이외에도 내부 구멍에 의한 폐곡선으로 이루어진 내부 루프를 추가로 갖게 된다. 루프는 half-curve들로 이루어져 있고 이들 중 첫 번째 half-curve의 주소를 루프가 간직하고 있다. 모든 curve는 서로 반대 방향으로 향하는 한쌍의 half-curve들로 이루어져 있으므로, curve의 좌우에 있는 각 블록의 루프에 half-curve 하나씩이 대응된다. 만약 어떤 curve가 인접한 두 블록의 경계에 있지 않고 하나의 블록에 속하여 있는 경우 나머지 하나의 half-curve는 NULL로 처리한다. 이러한 자료 구조는 B-rep 방식의 형상모델러에서 사용되고 있는 자료구조 방식 중의 하나인 half-edge 구조[10]를 변형한 것이다.

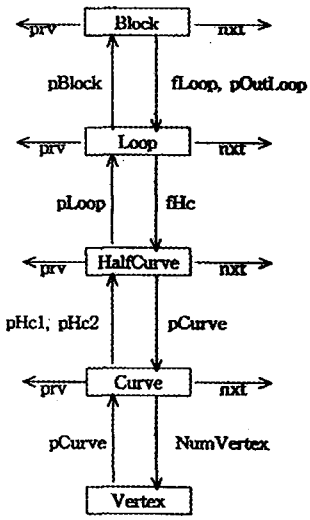


Fig. 1 Data structure for shape information

Fig. 2는 3개의 블럭들(#1, #2, #3)로 구성되어 있는 영역의 예에 대하여 자료구조의 예를 보여준다. 각 블럭은 경계곡선에 의하여 둘러싸여 있고, 각 곡선들은 서로 반대방향으로 향하는 한 쌍의 half-curve로 이루어져 있다. 각 블록들의 외부 경계는 반시계방향의 half-curve들로 표현되며, 블록 #2처럼 내부에 구멍이 있는 경우 내부 경계는 시계방향의

half-curve들로 표현된다. 이러한 half-curve들로 이루어진 폐곡선을 loop라고 부르며, 모든 블록들은 반드시 하나의 외부 loop를 가져야 하며, 내부에 구멍이 있는 경우 구멍의 수만큼 내부 loop를 갖는다.

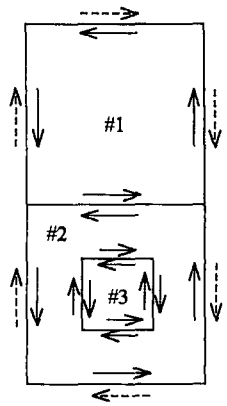


Fig. 2 Example of data structure for shape informations

이러한 자료구조를 이용하여 얻을 수 있는 장점 중의 하나는 주어진 점이 특정 블록의 내부에 있는지를 쉽게 판단할 수 있도록 하는 것이다. Fig. 3에서 보는 바와 같이 복잡한 형상이든 단순한 형상이든 상관없이 주어진 점에서 임의의 방향으로 반무한 직선을 가정하고, 이 반무한 직선과 블록의 루프를 구성하는 curve들과의 교점의 수를 계산하여, 주어진 점이 블록의 내부에 위치하여 있는지를 판단할 수 있다. 그 이외에도, 형상정보 자료구조를 통하여 얻을 수 있는 여러가지 장점들을 참고문헌 [10]에서 참조할 수 있다.

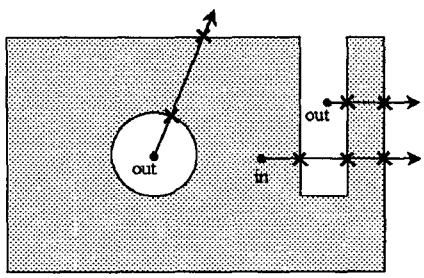


Fig. 3 A containment test based on the ray-firing

형상정보 자료구조가 Table 1에 나타나 있다.

Table 1 Data structure for shape information

```
class CFemHalfCurve
{
   public:
      CiCurve                 *Curve;
      CFemNode                *Node;
      CFemLoop                *pLoop;
      CFemHalfCurve           *pre;
      CFemHalfCurve           *nxt;
      ......
};

class CFemLoop
{
   public:
      int                     id;
      CFemHalfCurve           *fCurve;
      CFemBlock               *pBlock;
      CFemLoop                *pre;
      CFemLoop                *nxt;
      ......
};

class CFemBlock
{
   public:
      int                     id;
      CFemLoop                *Outer;
      CFemLoop                *fInner;
      CconFemNode             *fNode;
      CconFemElement          *fElem;
      CFemBlock               *pre;
      CFemBlock               *nxt;
      ......
};
```

2.2 요소정보 자료구조

삼각형 요소를 생성할 때, 일반적으로 사용될 수 있는 자료구조는 삼각형 요소가 세 절점의 절점번호를 기억하는 것이다. 그러나 이러한 방식은 어떤 한 절점을 둘러싸고 있는 요소들을 찾고자 할 때, 모든 요소들을 검색하여야 하며, 주위의 요소들을 모두 찾았다 하더라도 그들이 어떤 순서로 서로 배치되어 있는지 알기 위하여 또 다시 계산을 수행하여야 한다. 그러나 half-edge 구조를 사용하면, 위와 같은 정보를 가장 빠른 시간에 효율적으로 찾을 수 있다. Fig. 4는 그 예를 보여준다. 주어진 절점 v는 자신에 연결된 임의의 half-edge he1을 기억하고 있다. half-curve는 자신이 속한 루프를, 루프는 자신이 속한 요소를 알고 있으므로 he1이 속한 삼각형 요소를 알아낼 수 있다. half-edge는 ring 형태의 이중구조를 가지고 있으므로 he1의 이전 링크인 he2를 쉽게 찾을 수 있으며, he2는 자신이 속한 edge를, edge는 자신에 포함된 한쌍의 half-edge를 알고 있으므로 he2에 대응하는 he3를 쉽게 알 수 있다. 따라서, he3를 포함한 삼각형 요소를 알아낼 수 있으며, 이와 같은 과정을 반복하면, 절점

주위의 모든 삼각형 요소를 효율적으로 빠른 시간에 찾을 수 있다. 단지, 삼각형 요소에서는 내부 루프가 존재하지 않기 때문에 형상 모델러의 half-edge 구조체를 있는 그대로 사용하지 않고 삼각형 요소를 다루는데 필요한 부분만을 남기고 다른 부분은 모두 제거하여 변형시켜 사용하였다.

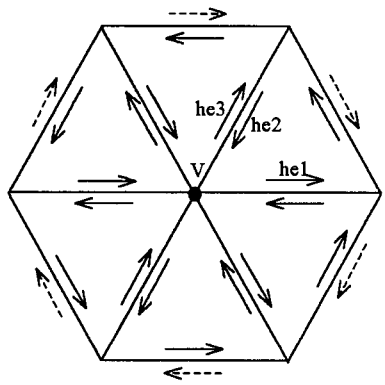


Fig. 4 Finding neighbering elements around a vertex using data structure for elemnet information.

요소정보 자료구조가 Table 2에 나타나 있다.

Table 2 Data structure for element information

```
class CFemNode
{
  public:
    int                    id;
    double                 x,y;
    CFemNode               *pre;
    CFemNode               *nxt;
    ......
};

class CFemElement
{
  public:
    int                    id;
    CFemNode               *node[3];
    CFemElement            *adjacent[3];
    CFemElement            *pre;
    CFemElement            *nxt;
    ......
};
```

3. 수정 Delaunay 삼각화

Delaunay 삼각화는 주어진 절점들에 대하여 가장 등변 삼각형에 가까운 삼각형들의 집합을 제공하며, 그 해는 유일하다. 즉, 주어진 삼각형 요소의 외접원 내부에 어떤 절점도 포함되는 것을 허용하지 않기 때문이다. 새로 도입된 절점에 대하여 기존의 삼각형 요소들의 외접원이 이 절점을 포함하면 해당되는 삼각형들을 모두 검색하여 해체한다. 해체된 삼각형들을 모으면 항상 볼록 다각형이 되며, 그 볼록 다각형의 각 꼭지점에서 새로운 절점까지 연결하여 새로운 삼각형 요소들을 체결한다. 하지만, 이러한 조건은 일반적인 복잡한 형상에 대하여 Delaunay 삼각화를 수행할 때 문제를 일으킬 수도 있으며, 그러한 전형적인 예들이 Fig. 5에 나타나 있다.

첫 번째, 수치오차에 의하여 볼록 다각형 생성에 실패할 수 있는 예가 Fig. 5(a)에 나타나 있다. 새로운 절점 P가 삼각형 요소 A와 B에는 포함되지만 삼각형 요소 C와 D의 외접원의 경계에 있는 경우, 특히 요소 C는 수치오차에 의하여 포함된 것으로 간주되고 요소 D는 수치오차에 의하여 포함되지 않은 것으로 간주될 때, 해체될 삼각형 요소 A, B, C는 볼록 다각형을 이루지 않는다. 요소정보의 자료구조를 이용하면 먼저 절점 P를 포함하는 삼각형을 찾을 수 있으며, 각 삼각형의 edge에 인접한 삼각형들을 Fig. 4에서와 비슷한 방법으로 검색하여 외접원의 포함여부를 조사한 뒤, 볼록 다각형을 구성하므로 삼각형 요소 C가 해체되지 않는다. 즉, 삼각형 요소 A와 edge를 인접하고 있는 삼각형들을 조사하여 해체시키므로, 요소 C는 요소 A와 인접하고 있지 않으므로 요소 D가 해체되지 않는한 해체 여부를 조사하지 않는다. 이러한 요소정보 자료구조를 이용한 Delaunay 삼각화 방법은 안정성과 신뢰도를 크게 향상시킨다.

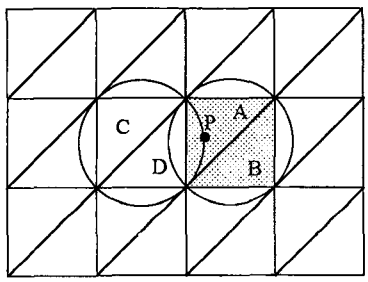
두 번째, 영역의 경계를 통하여 볼록 다각형이 잘못 생성될 수 있는 경우가 Fig. 5(b)에 나타나 있다. 경계 영역 외부에 있는 삼각형 요소 A의 외접원이 점 P를 포함하는 경우, 삼각형 요소 A도 해체 삼각형의 대상이 되어 볼록 다각형을 만드는데 참여하게 되면 계산이 실패하게 된다. 이러한 경우를 제거시키려면 일반적인 경우 많은 계산을 수행하여 조사하여야 하지만, 앞의 방법과 같이 요소정보 자료구조를 사용하면, 해체 삼각형들의 인접 삼각형들을 통하여서만 해체될 삼각형을 조사하므로 결코 삼각형 요소 A가 해체 대상으로 고려될 수 없다. 점 P가 포함된 삼각형 요소의 edge들에 인접한 삼각형들을 대상으로 탐색하기 때문이다.

세 번째, 삼각형의 경계면이 깨어질 수 있는 예가 Fig. 5(c)에 나타나 있다. 이러한 경우는 초기 삼각화 과정에서 주로 나타나며, 초기 삼각화를 수행하면서 형상의 경계가 깨어져 삼각형 요소 A와 B로 분리되었다. 이러한 경우는 앞의 방법과 같이 해체할 삼각형을 검색할 때,

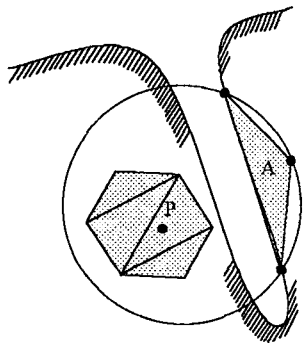
삼각형 요소의 edge가 경계인 경우, 그 경계 edge 외부에 인접한 삼각형 요소는 검색 대상에서 제외시키면 볼록 다각형을 생성할 때 이러한 오류가 발생하지 않는다.

따라서, 수정 Delaunay 삼각화는 삼각형을 조사하고 해체하여 이들을 볼록 다각형으로 만들 때, 먼저 새로운 절점 P를 포함하는 삼각형 요소를 찾고, 그 삼각형의 edge들을 조사하여 경계 edge가 아니면, 각 edge의 인접한 삼각형을 요소정보 자료구조로부터 찾아내어 인접 삼각형의 외접원이 절점 P을 포함하는지 여부를 조사하고, 만일 포함하면 주변의 인접 삼각형 요소를 계속 검색한다. 그러므로 검색된 삼각형들은 반드시 경계가 아닌 edge들을 통하여 서로 인접하여 있으며, 이들 edge를 제거시켜 쉽게 볼록 다각형을 만들 수 있다.

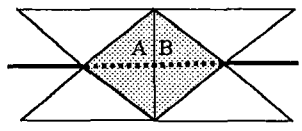
형상정보 자료구조는 주로 영역의 내부에 속하는지 외부에 있는지를 판정할 때 사용한다. 이러한 경우는 초기 삼각화 과정에서 영역 내부의 삼각형만을 남기고 외부 삼각형을 해체하고자 할 때, 그리고 새로운 절점을 추가하였을 때 그 점이 영역의 내부에 있는지를 조사할 때 사용한다.



(a)



(b)



(c)

Fig. 5 Failures in making a convex polygon by degenerating inadequate trangles.

4. 계산 예

본 연구를 통하여 개발된 삼각형 유한요소 격자 자동생성 코드를 이용하여 격자 생성의 안정성 및 신뢰도를 조사하였다. Fig. 6은 네 가지 일반적인 형상에 대한 격자 생성 예를 보여주며, Fig. 7은 횡류팬의 형상에 대한 격자 생성 예를 보여준다.

5. 결 론

수정된 Delaunay 삼각화 기법에 의한 자동 격자 생성 방법은 복잡한 형상에 대해서도 높은 신뢰도와 정확도, 그리고 빠른 계산속도를 유지할 수 있다. 이러한 특징은 계산영역의 형상정보 자료구조와 요소정보 자료구조의 적절한 설계로부터 가능할 수 있다. 특히, 형상모델러에서 사용되고 있는 B-rep 방식의 half-edge 데이터 구조를 본 연구에 알맞게 변형하여 사용함으로써, 어떠한 형상의 계산 영역에 대한 위상 및 기하 정보도 빠르고 완벽하게 처리할 수 있으며, 요소정보의 자료구조와 연계하여 격자생성의 신뢰도를 크게 향상시킬 수 있었다. 또한 Delaunay 삼각화 방법들의 문제점들을 연구한 후 각 문제점의 해결방법을 제시하였다. 수정된 Delaunay 방법은 그러한 연구결과들을 적용한 삼각화 기법이며, 격자생성이 갖추어야 할 신속성과 안정성에 있어 만족할 만한 성과를 얻을 수 있었다.

본 연구를 통하여 개발된 GENMESH V1.0은 객체지향형 프로그래밍 기법을 사용하여 강력한 GUI 환경을 지원하는 삼각형 격자 자동생성 코드로써, http://caflab.yeungnam.ac.kr에서 인터넷을 통하여 다운로드할 수 있다.

후기

이 논문은 1996학년도 영남대학교 학술연구조성비에 의한 것임.

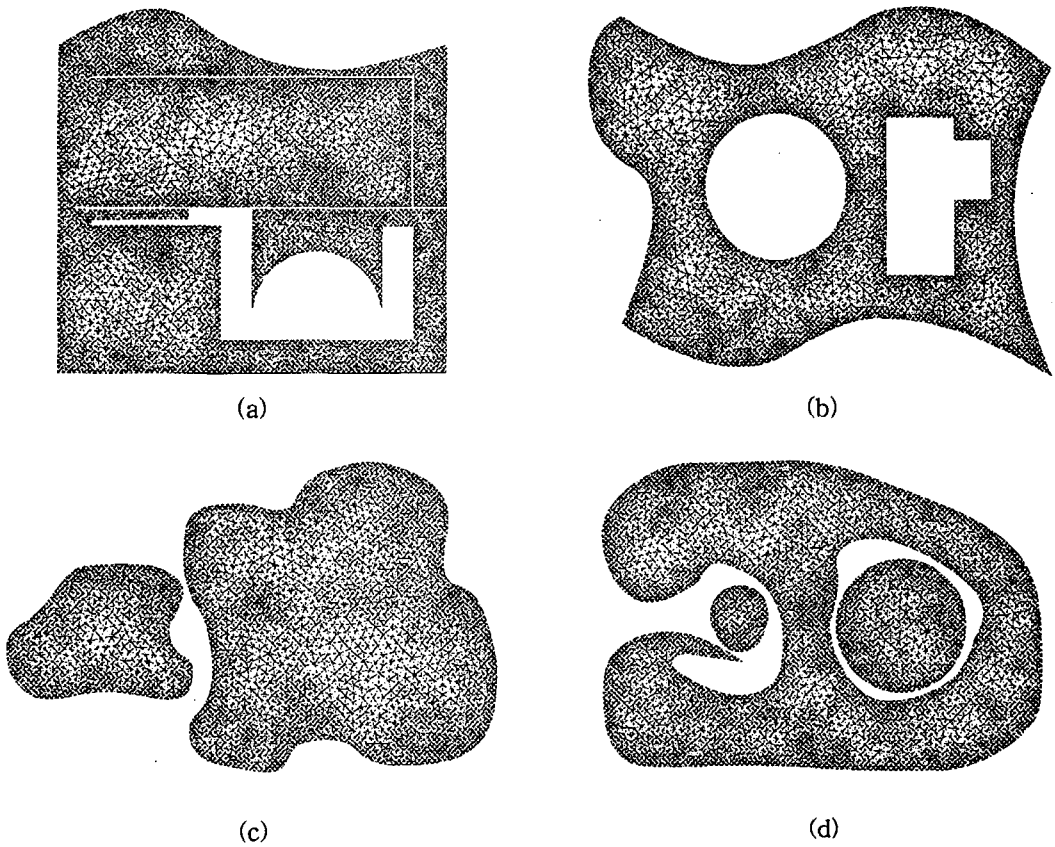


Fig. 6 Examples of triangulation for complicated computational domains

5. 참 고 문 헌

[1] Lo S.H., "A New Mesh Generation Scheme for Arbitrary Planar Domains," *Int. J. Numer. Methods Eng.*, Vol. 21 (1985), pp.1403-26.

[2] Preparata F.P., Shamos M.I., *Computational Geometry, An Introduction,* Texts Monogr. Comput. Sci. New York: Springer-Verlag (1985).

[3] George P.L. Hecht F., "Automatic Mesh Generatior with Specified Boundary," *Computer. Method Appl. Mech. Eng.*, Vol. 33 (1991), pp. 975-995.

[4] Barker T.J., *Mesh adaptation strategies for problems in fluid dynamics,* Finite Elem. Anal. Des. In press (1996).

[5] Yerry M.A., Shephard M.S. "Automatic Three Dimensional Mesh Generation by the Modified-Octree Technique," *Int. J. Numer. Methods Eng.,* Vol. 20 (1984), pp. 1965-90.

[6] Rebay S., "Efficient Unstructured Mesh Generation by means of Delaunay Triangulation and Bowyer/Watson Algorithm," *J. Comput. Phys.,* Vol. 106 (1993), pp. 125-138.

[7] Bowyer A., "Computing Dirichlet Tesselations", *The Comput. J.*, 24-2 (1981), pp. 162-166.

[8] Watson D.F., "Computing the n-Dimensional Delaunay Tesselation with Application to Voronoi Polytopes", *The Comput. J.*, 24-2 (1981), pp. 67-172.

[9] 박 병호, 사 종엽, "Delaunay 삼각화에 의한 유한요소 자동생성 코드개발에 관한 연구", 한국전산유체공학회 춘계학술대회 (1996)

[10] Mantyla M., "An Introduction to Solid Modeling", *Computer science press*, (1988)

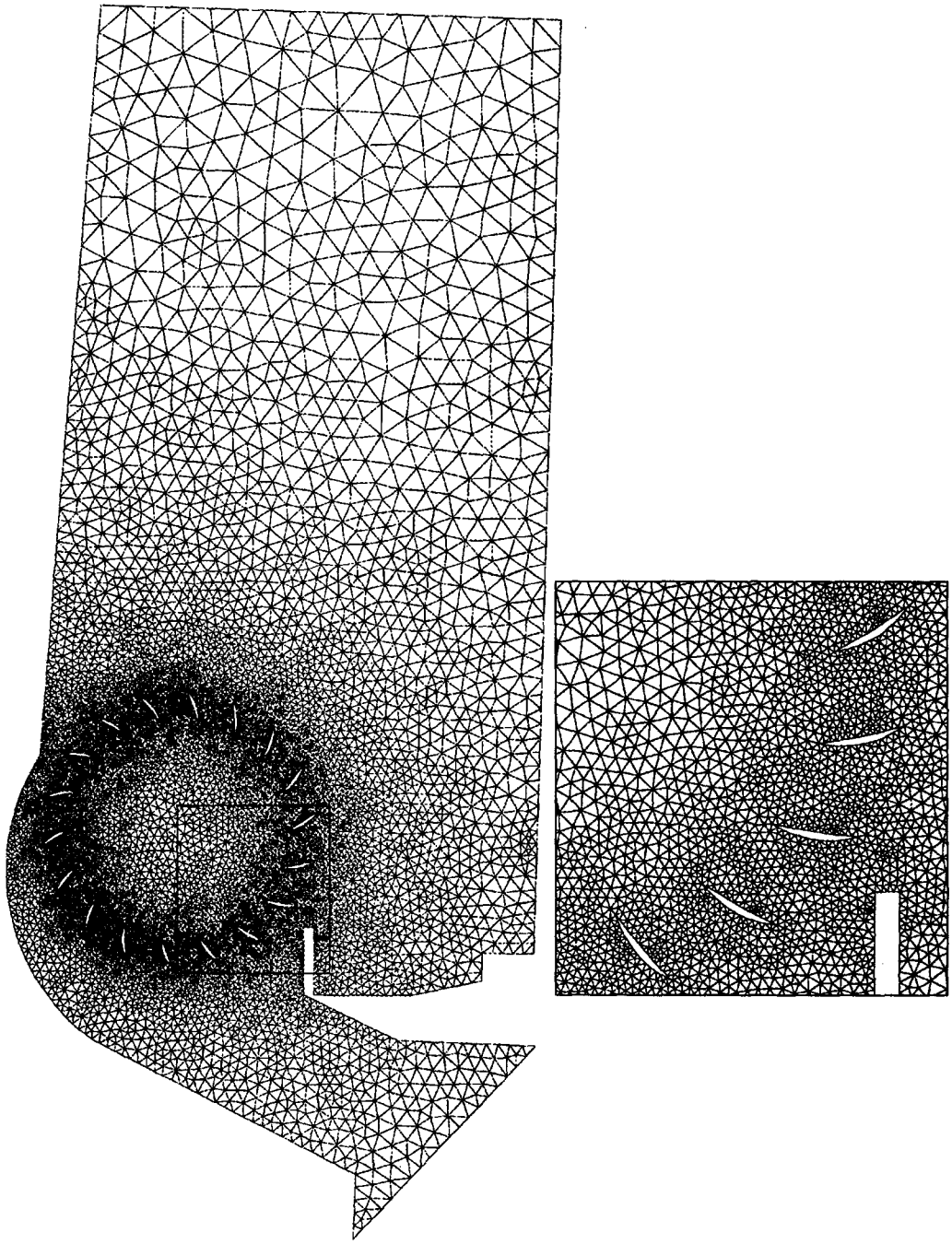


Fig. 7 Examples of triangulation for cross-flow fan geometry