

Delaunay 삼각화에 의한 유한요소 자동 생성 코드 개발에 관한 연구

Code Development of Automatic Mesh Generation for Finite Element Method
Using Delaunay Triangulation Method

○박 병호¹⁾, 사 종엽²⁾

Pyong-Ho Park and Jong-Youb Sah

The Delaunay triangulation technique was tested for complicated shapes of computational domain. While a simple geometry, both in topology and in geometry, was discretized well into triangular elements, a complex geometry often failed in triangularization. A complex geometry should be divided into smaller sub-domains whose shape is simple both topologically and geometrically. The present study developed the data structures not only for relationships among neighboring elements but also for shape information, and coupled these into the Delaunay triangulation technique. This approach was able to enhance greatly the reliability of triangularization specially in complicated shapes of computational domains. The GUI (Graphic User Interface) and OOP (Object-Oriented Programming) were used in order to develop the user-friendly and efficient computer code.

1. 서론

최근 컴퓨터 하드웨어의 발달과 수치해석 알고리즘의 발달로 인하여 복잡한 물리적 현상을 해석하고 실제로 설계에 응용하는 사례가 크게 증가하고 있다. 구조물의 해석 등에서는 유한요소법이 널리 사용되어 왔으며, 전산유체역학 분야에 있어서도 유한요소법의 적용이 꾸준히 증가하고 있는 추세이다. 유한 요소법은 작은 요소로 전체 계산 영역을 분할하고 각 요소에 형상함수를 정의하여 응력이나 유동장을 계산하는 것으로써, 유한요소법에 의한 프로그램은 식을 유도하여 코드화하는 과정이 복잡하지만 일단 코드가 완성되면 사용자들이 복잡한 형상의 문제에 적용하기에 편리하므로 여러 분야의 상용 소프트웨어를 개발하는데 널리 사용되고 있다. 그러나 이러한 유한 요소 프로그램을 사용하기 위하여서는 계산영역을 작은 요소로 분할하는 요소 생성이 필요한데, 계산 영역의 형상이 복잡한 경우에 이러한 요소 생성이 수치계산보다 사용자에게 과중한 부담을 주는 경우가 흔히 있다. 이러한 문제점을 해결하기 위하여 유한 요소를 자동으로 생성하는 여러 가지 방법들이 제안되었으며, 삼각형 요소를 만드는 방법으로는 크게 프론트 전진 기법^[1]이나 Delaunay 삼각화 기법^[2,3] 등이 일반적으로 사용되고 있다. 이 중에서 복잡한 형상에 대하여 보다 높은 신뢰도를 가질 수 있는 것이 Delaunay 방법으로 많은 연구자들에 의하여 활발하게 연구되어 왔으며, Delaunay 방법을 구현하는 구체적 부분에서 많은 변형들이 제안되어 상당히 다양한 종류의 Delaunay 방법들이 존재한다^[4-7]. 이러한 많은 선행 연구가 있었음에도 불구하고 복잡한 형상에 대하여 한 번에 정확한 요소를 생성할 수 있는 신뢰도는 아직 충분히 높지 않은 것으로 판단된다. 대부분의 경우에 복잡한 형상을 몇 개의 단순한 부영역(sub-domain)으로 분할하여 각 부영역에 대하여 요소를 자동 생성하고 이를 전체적으로 조합하는 방법이 널리 사용되고 있으며, 이는 형상이 복잡할 경우 사용자가 계산영역을 몇 개의 부영역으로 일일이 분할하여야 하므로, 진정한 의미의 유한요소 자동생성이라 보기 어렵다.

본 연구에서는 Delaunay방법에 대하여 기존의 여러 가지 변형들을 조사하고 이들을 통합하며, 또한 B-rep 방식의 solid modeler의 데이터 구조 등을 활용하여 계산 영역의 위상 정보 및 기하 정보를 격자 생성 방법과 효율적으로 연계하여, 복잡한 계산영역에 대하여서도 부영역으로 분할하지 않고 한 번에 안전하고 정확한 격자를 자동으로 생성할 수 있도록 Delaunay 방법의 신뢰도를 향상시키고자 하였다. 또한, 복잡한 계산영역을 사용자들이 CAD 식으로 쉽고 편리하게 입력하

1) 영남대학교 기계공학과 대학원 (712-749, 경북 경산시 대동 214-1 영남대학교 기계공학과)

2) 영남대학교 기계공학과 조교수

는 것이 가능하도록 윈도우즈 환경 하에서 그래픽스와 GUI를 이용하여 자동 격자 생성 프로그램을 개발하였다.

2. 형상정보의 표현 및 처리

Fig. 1에서 보는 바와 같이 위상적으로나 기하적으로 다양하고 복잡한 형상들에 대하여 쉽고 편리하게 삼각형 요소를 자동 생성하기 위하여서는 다음과 같은 사항들이 필요하다. 먼저 계산 영역의 형상들의 위상정보와 기하정보를 저장하고 처리하기 위한 형상정보 데이터 구조를 설계하여야 한다. 이러한 데이터 구조가 정의되면 이를 입력시키는 방법을 찾아야 하는데, 본 연구에서는 경계곡선에 의하여 계산영역의 형상을 표현하는 방식을 사용하였다. 그러므로, 복잡한 경계를 잘 표현할 수 있도록 다양한 종류의 곡선들을 입력하고 이를 변형할 수 있어야 한다. 또한, 경계 곡선들을 이용하여 형상정보의 데이터 구조의 위상 및 기하정보와 상호 연결시켜 주어야 한다. 그리고, 경계 곡선 상에서 경계요소를 생성하고 이들의 조밀성을 사용자가 원하는 바대로 변경할 수 있어야 한다.

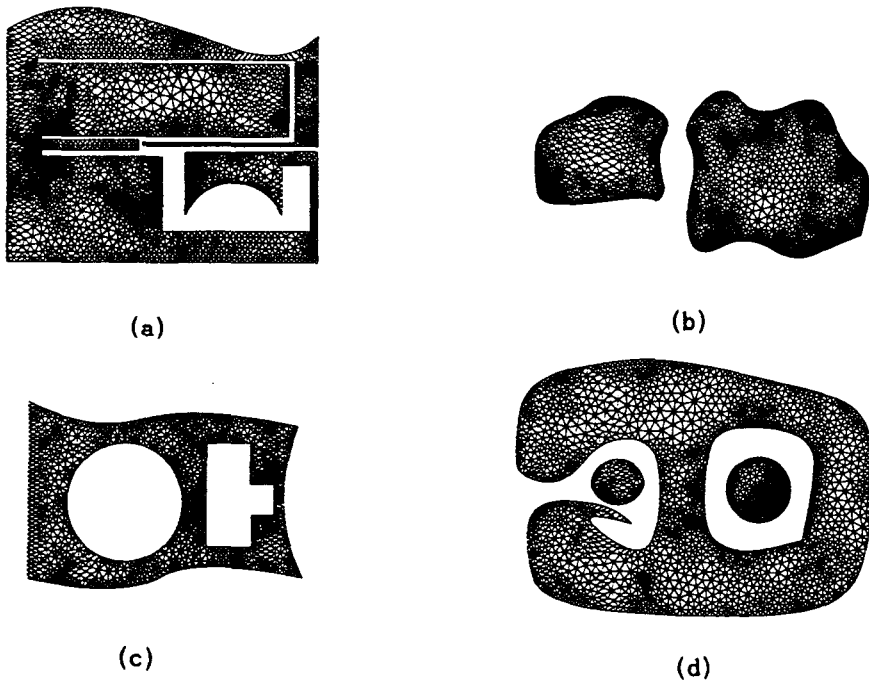


Fig. 1 Generation of triangular element for complicated shapes of computational domain

2.1 형상정보 데이터 구조

계산 영역의 형상을 표현하기 위하여서는 본 연구에서는 Fig. 2와 같은 데이터 구조를 사용하였다. 계산 영역을 먼저 블록(block)들의 집합으로 구성한다. 블록은 다음과 같은 경우에 서로 분리하여 정의하는데, 영역이 위상적으로 분리되어 있는 경우 각각을 서로 다른 블록으로 지정하며, 위상적으로 분리되어 있지 않더라도 두가지 이상의 다른 특성을 지닌 영역이 서로 인접하여 있는 경우 각각을 서로 다른 블록으로 지정하며, 그렇지 않은 경우 사용자가 내부 격자의 조밀성을 조절하거나 기타의 목적으로 편의상 하나의 영역을 두 개 이상의 영역으로 분할하고자 할 때 각각을 서로 다른 블록으로 지정할 수 있다. 블록은 루프(loop)들로 구성되며, 루프는 블록의 경

계에 의하여 닫혀진 폐곡선을 구성한다. 블록이 simple-connected 영역인 경우에는 외부 경계의 폐곡선으로 이루어진 하나의 루프를 갖지만, 내부에 구멍이 있는 double-connected 영역인 경우에는 외부 경계의 폐곡선으로 이루어진 외부 루프 이외에도 내부 구멍의 경계에 의한 폐곡선으로 이루어진 내부 루프를 추가로 갖게 되어 두 개의 루프를 갖는다. 루프는 half-curve들로 이루어져 있고 이 들 중 첫번째 half-curve의 주소를 loop가 간직하고 있다. 모든 curve는 최대 두 개의 half-curve들로 이루어져 있으며, 이들 각각의 half-curve들은 서로 방향이 반대이다. curve는 block에 대하여 two-manifold 이며, 이는 평면 상에서 curve가 자신의 좌우에 각각 하나씩 두 개의 block에 공유될 수 있음을 뜻하며, 각 half-curve가 각 block의 loop에 하나씩 대응된다. 만약 어떤 curve가 인접한 두 개 블록들 간의 경계에 있지 않고 하나의 블록에만 속하여 있는 경우 나머지 하나의 half-curve는 NULL로 처리된다. 이러한 데이터 구조는 B-rep 방식의 solid modeler에서 사용되고 있는 데이터 구조 방식들 중의 하나인 half-edge 구조^[8]를 평면 유한요소 생성에 알맞게 변형한 것이다.

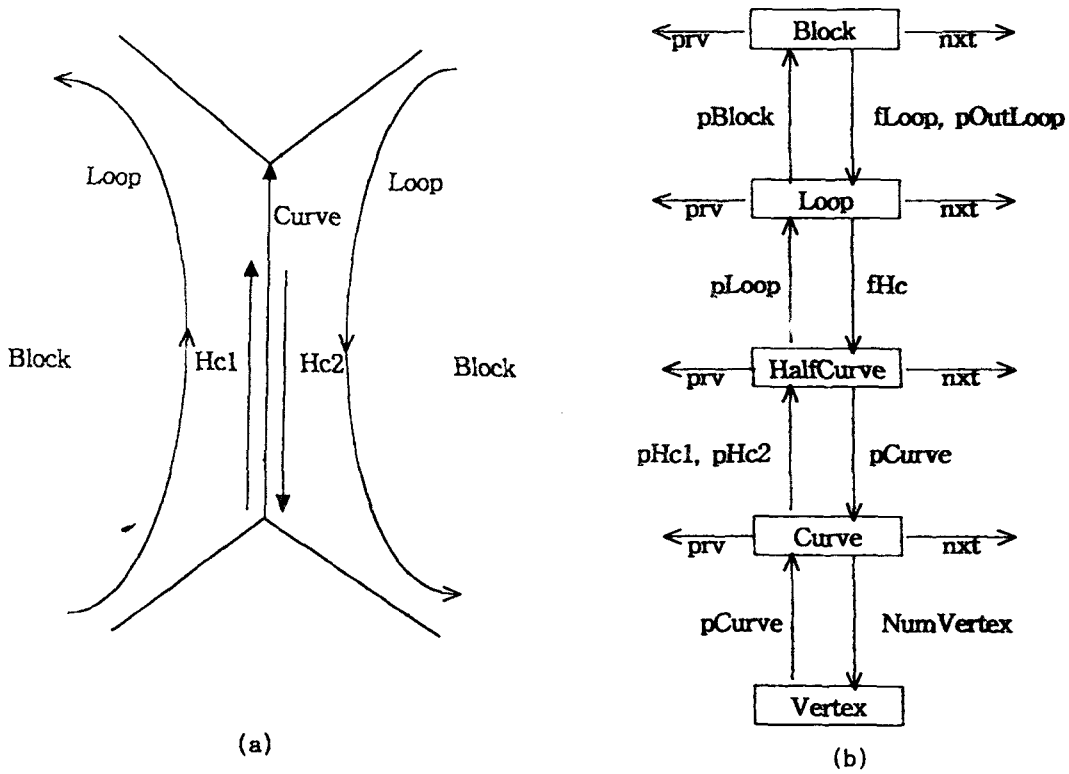


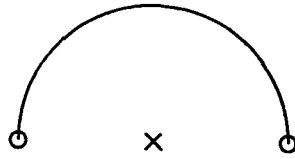
Fig. 2 Data structure for shape information

2.2 경계 곡선

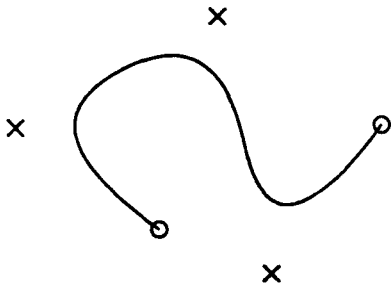
경계곡선은 복잡한 형상을 표현하기 위하여 직선, 원호, cubic-spline, NURBS를 기본적으로 제공하고 사용자가 함수를 정의하여 필요한 곡선을 생성할 수 있는 기능을 추가하였다. Fig. 3은 본 프로그램에서 제공하는 경계곡선들을 보여주고 있으며, 객체지향 구조의 추상 클래스를 사용하여 curve를 정의하고 이로부터 실제 곡선의 종류를 파생시킴으로써 새로운 곡선의 종류를 추가하는 것이 편리하도록 하였다. 또한 모든 곡선은 사용자가 원하면 항상 수정이 가능하도록 하였으며, 필요한 경우에 삭제하거나 또는 곡선을 들로 분할할 수 있도록 하였다^[9].



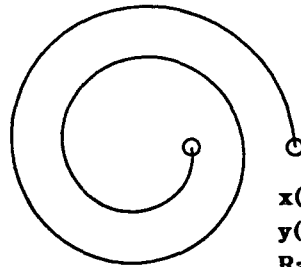
(a) line



(b) arc



(c) NURBS



$$\begin{aligned}
 x(t) &= (1+t/2\pi) \cdot \cos(t) \\
 y(t) &= (1+t/2\pi) \cdot \sin(t) \\
 \text{Range} &: 0 \leq t \leq 4\pi
 \end{aligned}$$

(d) function-defined curve

Fig. 3 Types of boundary curve

2.3 경계 요소

경계 요소는 curve 상에서 요소의 개수와 분할함수의 계수들을 입력함으로써 생성하거나 조절할 수 있다. 특히, 중간 지점을 밀집시키기 위하여서는 곡선을 분할하여 두 개의 곡선에 대하여 각각의 밀집 정도를 제어할 수 있도록 하였다. 경계요소 분할 시의 조밀성을 제어하기 위하여 다음과 같은 분할 함수를 사용하여 분할 계수(coefficient)등을 정의하였다^[9].

(A) 멱함수

$$s_i = S \frac{r^i - 1}{r^n - 1}$$

(B) 지수함수

$$s_i = S \frac{e^{ai/n} - 1}{e^n - 1}$$

(C) 초월 싸인 함수

$$s_i = S \left(1 + \frac{\sinh(a(\frac{i}{n} - 1))}{\sinh(a)} \right)$$

(D) 초월 탄젠트 함수

$$s_i = S \left(1 + \frac{\tanh(a(\frac{i}{n} - 1))}{\tanh(a)} \right)$$

2.4 내부 요소의 조밀성 조절

경계 요소가 생성되면 이로부터 내부 요소를 자동으로 생성하므로 요소 생성 과정은 편리한 반면, 내부 요소의 밀집 정도를 제어하기 어려운 면도 있다. 본 프로그램을 이용하여 내부요소의 밀집 정도를 조절하기 위하여서 다중 블록을 사용할 수 있다.

3. 요소정보의 표현 및 처리

계산영역의 형상에 대한 정보와 경계 요소가 모두 입력되면, 이를 토대로하여 삼각형 요소를 생성한다. 그러나 복잡한 형상에 대하여 항상 신뢰성있게 격자를 구성하려면 생성된 요소들 간의 정보나 계산영역의 형상에 대한 정보들이 효율적으로 연계되어 계산영역 밖으로 격자가 걸치거나 입력된 경계 형상을 깨뜨리는지 등의 여부를 빠르고 정확하게 조사할 수 있어야 한다. 그러므로, 요소(element)와 정점(node)에 대한 데이터 구조를 효율적으로 설계하여야 하며, 이러한 데이터 구조 내에서 Delaunay 삼각화를 수행한다. 또한, 요소의 자동 생성의 신뢰도를 높이기 위하여 기존 삼각형 요소의 해체와 이에 따른 경계 격자의 해체방지를 위한 특별한 고려가 필요하다.

3.1 요소정보 데이터 구조

본 연구에서는 인접한 삼각형 요소들 간의 정보를 빠르고 정확하게 전달하기 위하여 요소정보 데이터 구조를 사용하고 있다. 요소정보는 요소를 총괄하는 클래스, 요소와 절점의 정보를 저장하는 클래스들로 이루어져 있다. 각 요소는 자신의 두 꼭지점 사이의 변에 인접하여 있는 요소에 대한 정보를 빠른 시간에 탐색할 수 있으며, 요소 상호 간의 간섭이나 요소와 경계 형상과의 간섭을 빠른 시간 내에 효율적으로 조사할 수 있도록 하였다. 따라서, 해체될 삼각형의 주변 삼각형들에 대한 정보를 이용하여 빠른 시간 내에 블록 다각형을 만들고, 이들 요소 간의 관계와 요소와 경계 사이의 관계를 조사하여 어떠한 복잡한 형상에 대하여서도 블록 다각형이 잘 못 생성되는 일이 없도록 하여 삼각형 요소 생성의 신뢰도를 확보하도록 하였다.

3.2 Delaunay 삼각화 기법

Delaunay 삼각화 기법은 주어진 점의 집합으로 형성할 수 있는 삼각형의 집합 중 가장 등변형의 삼각형 형상에 가깝게 생성 시켜주는 특징을 가지고 있다. 따라서 이 방법은 등변형의 삼각형의 형상이 바람직한 유한요소를 생성하기 위해 널리 사용되고 있다.

(A) Voronoi 다각형

P_1, P_2, \dots, P_n 를 평면 R^2 에서 정의된 서로 다른 점이라고 할 때, 다음과 같이 Voronoi의 다각형을 정의한다.

$$V_i = \{ x : d(x, P_i) \leq d(x, P_j), \text{ for all } i \neq j \}, \quad (1 \leq i \leq N)$$

여기서, V_i 는 i 번째 Voronoi의 다각형이고, P_i 는 2차원 평면 위에 위치한 i 번째 점이며, $d(x, P_x)$ 는 x 와 P_x 사이의 거리를 나타낸다. V_i 와 여러 개의 V_j 가 인접해 있을 때, V_i 는 점 P_i 와 각각의 P_j 를 잇는 선분의 수직 이등분선으로 경계가 지워진 블록 다각형(Voronoi Polygon)이 된다. 이러한 다각형 $\{V_i\}_{i=1}^N$ 의 집합을 Dirichlet Tessellation 이라고 한다. 일반적으로 Voronoi 다각형의 한 꼭지점은 세 개의 서로 다른 이웃 다각형이 공유하므로 각각의 인접한 다각형 내부의 생성 점을 서로 연결하면 삼각형 T_k 를 이룬다. 이 삼각형의 집합 $\{T_k\}$ 를 Delaunay 삼각형이라고 한다

[3]

(B) Watson 알고리즘

일직선상에 있지 않은 세 개의 주어진 점으로 구성된 삼각형에서 그 외접원 내에 다른 점이 포함되어 있지 않으면 그 세 점은 Delaunay 삼각형의 조건을 만족한다는 것이다. 먼저, 주어진 모든 점들을 꼭지점으로 하는 삼각형을 구성한다. 그 다음 새로운 점이 하나씩 도입되는데, 이미 구성되어 있는 삼각형의 외접원이 새로 도입된 점을 포함하는 삼각형들을 모두 찾아내어 해체하며, 이들 해체된 삼각형으로 이루어진 다각형을 삽입 다각형(insertion polygon)이라고 하는데 항상 볼록 다각형의 형태를 갖는다. 마지막 단계로서 삽입 다각형의 꼭지점들과 새로 도입된 점을 연결하여 새로운 삼각형 요소들을 생성한다. 이와 같은 과정을 모든 점에 대해 계속하면, 주어진 모든 점에 대해 Delaunay 특성을 유지하는 삼각화가 이루어진다. 따라서 Delaunay의 삼각화에 의해 형성된 삼각형들의 각각의 외심원 안에는 다른 삼각형을 형성하는 어떠한 점도 포함되지 않는다^[3].

(C) 초기 삼각화 과정

먼저, 주어진 계산 영역의 형상을 충분히 포함하는 직사각형을 선택하고 이 사각형의 대각선 방향의 꼭지점 두 개를 서로 연결하여 네 개의 꼭지점과 두 개의 직각 삼각형으로 이루어진 초기 삼각형 요소를 생성한다. 다음, 경계 요소의 절점들을 차례로 하나씩 도입하여 Watson 알고리즘에 의하여 외접원이 새로 도입된 점을 포함하는 삼각형들을 해체하여 볼록 다각형을 만들고 볼록 다각형의 꼭지점과 도입된 점들을 연결시켜 새로운 삼각형들을 생성한다^[6]. 모든 경계요소의 절점들이 다 적용되었으면, 형상정보 데이터 구조와 요소정보 데이터 구조를 이용하여 계산 영역 밖에 있는 삼각형들을 해체시킨다.

(D) 삼각형 요소 생성 방법

초기 삼각화 과정 후, 계산 영역의 형상 내부에 절점을 생성시켜 Delaunay의 삼각화 과정을 거친다. 여기서 새로운 내부 절점의 위치를 결정하는데 여러 가지 방법들이 있을 수 있지만, 본 연구에서는 요소의 형상비에 의한 절점 생성 방법과 이웃한 요소들 간의 면적비에 의한 절점 생성 방법 두가지를 코드화하였다.

형상비에 의한 절점 생성 방법은 다음과 같다. 정삼각형에 가까운 요소를 생성시키기 위하여 삼각형의 외접원(반경 R)과 내접원(반경 r)을 사용하여 형상비 $R/2r$ 를 정의하고, 이 값이 가장 큰 삼각형을 찾아내어 그 삼각형의 외접원의 중심에 새로운 내부 절점을 도입시킨다^[7]. 정삼각형의 경우 $R/2r = 1.0$ 이며, 본 연구에서는 $R/2r \leq 1.5$ 의 기준에 들어오는 삼각형은 더 이상 분할하지 않았다.

면적비에 의한 절점 생성 방법은 다음과 같다. 요소 생성 시에 이웃한 요소들 간의 면적이 서로 심한 차이를 보여서 큰 삼각형과 작은 삼각형이 인접하여 있는 경우, 인접한 삼각형 간의 면적비가 가장 큰 경우를 찾아내어 큰 삼각형의 내부에 새로운 절점을 도입시킨다. 본 연구에서는 면적비가 두 배 이하인 경우 더 이상 분할을 계속하지 않았다.

(E) smoothing (regulation)

요소 생성의 마지막 단계로, 생성된 요소를 정삼각형에 더욱 가깝게 하고 인접 삼각형과의 면적 차이를 줄여주는 과정이다. 본 연구에서는 두가지 방법의 smoothing을 사용하였다. 첫번째 방법은 영역 내부에 새로운 절점을 추가시킬 때, 해체된 삼각형으로 볼록 다각형을 얻고 난 후 그 다각형의 도심으로 새로운 절점의 위치를 이동시켜 삼각형을 생성하는 방법이다^[10]. 두번째 방법은 Laplacian-type의 smoothing으로 요소 생성이 완전히 끝난 후에 행하여지며 각 절점에 대하여 그 절점에 대하여 인접한 절점들까지의 거리벡터를 모두 합하여 평균한 뒤 완화계수를 곱하여 그 절점의 위치를 이동시키는 것이다.

$$x_i^{n+1} = x_i^n + \frac{\omega}{n} \sum_{k=1}^n (x_k - x_i^n)$$

$$y_i^{n+1} = y_i^n + \frac{\omega}{n} \sum_{k=1}^n (y_k - y_i^n)$$

여기서, x_i 와 y_i 는 현 절점의 위치이고, x_k 와 y_k 는 주변 인접한 절점들의 위치이며, n 은 인접한 절점들의 개수, ω 는 완화 계수이다. 경험적으로 완화 계수는 0.2를 사용하고 smoothing은 100 내지 200회 정도 반복하여 수행한다^[6].

3.3 삼각형 해체 및 경계격자 보존에 관하여

앞에서 설명한 방법으로 Delaunay 삼각화 과정을 수행할 때, 새로 추가된 절점에 대하여 기존 삼각형을 해체하여 볼록 다각형을 만들고 이 볼록 다각형의 꼭지점에서 새로 추가된 절점에 선을 연결시켜 새로운 삼각형을 만드는데, 볼록 다각형을 구성하기 위하여 해체될 삼각형을 선택할 때 이 과정의 신뢰도가 가장 문제가 된다. 단순하게 삼각형의 외접원이 삼입점을 포함하는지만을 검사하면 경계 요소가 깨어지거나 수치오차 등에 의하여 볼록 다각형이 제대로 형성되지 않는다. 본 연구에서는 요소정보와 형상정보를 서로 연계하여 경계요소가 손상되거나 수치오차에 의하여 볼록 다각형이 제대로 구성되지 못하는 경우를 빠른 시간 내에 효율적으로 탐색할 수 있게 하여, 빠른 계산 속도를 유지하면서 신뢰도를 보장할 수 있도록 하였다.

4. 결론

본 연구에서 개발된 delaunay 삼각화 기법에 의한 자동 격자생성 방법은 복잡한 형상에 대하여 서도 높은 신뢰도와 정확도를 유지할 수 있다. 이러한 특징은 계산영역의 형상에 대한 데이터 구조와 요소 및 절점에 대한 데이터 구조의 적절한 설계로부터 가능할 수 있다. 특히, B-rep 방식의 solid modeler에서 사용되고 있는 half-edge 데이터 구조를 본 연구에 알맞게 변형하여 사용함으로써 어떠한 형상의 계산 영역에 대한 위상 및 기하 정보도 완벽하고 빠르게 처리할 수 있으며, 요소의 데이터 구조와 이를 연계하여 격자 생성의 신뢰도를 크게 향상시킬 수 있었다.

또한 강력한 기능의 GUI 환경을 구축하여 사용자의 편의와 요소 생성의 효율을 증대시켰다. 그리고 객체지향형 구조를 채택하여 전체 프로그램의 모듈화와 유연성을 크게 향상시켰으며, 프로그램의 효율성을 높이고 수정 및 확장이 용이하도록 하였다.

참고문헌

- [1] J.Paire, M.Vahdati, K.Morgan and O.C.Zienkiewicz, "Adaptive remeshing for compressible flow computations", J. Comp. Phys., 72, (1987), p.449-466.
- [2] A.Bowyer, "Computing Dirichlet tessellations", The Comput. J. 24-2, (1981), p.162-166.
- [3] D.F.Watson, "Computing the n-dimensional Delaunay Tesselation with Application to Voronoi Polytopes", The Comput. J. 24-2, (1981), p.167-172.
- [4] S.H.Lo "Delaunay triangulation of non-convex planar domains", Int. J. Numer. Methods Engrg. 28, (1989), p.2695-2707.
- [5] T.P.Fang and L.A.Piegl, "Algorithm for Delaunay triangulation and convex-hull computation using sparse matrix", Computer-Aided Design, 24-8, (1992), p.425-436
- [6] S.Rebay, "Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm", J. Comput. Phys., 106, (1993), p.125-138.
- [7] W.K.Anderson, "A Grid Generation ans Flow Solution Method for the Euler Equations on Unstructured Grids", J. Comput. Phys., 110, (1994), p.23-38.
- [8] M. Mantyla, "An Inversion Algorithm for Geometric Models", Comput. Graph. 16-3, (1982), p.51-59.
- [9] 채 은미, 사 종엽, "해석적 자동격자생성방법에서 보간방법에 따른 조절함수의 영향에 관한 연구", 한국전산유체공학회, 1, (1996)
- [10] I.Zeid, "CAD/CAM Theory and Practice", McGraw-Hill, (1994)