

# 전기기기의 유한 요소 해석을 위한 효율적인 요소 생성에 관한 연구

論 文  
48B-4-5

## A Study on the Efficient Mesh Generation for Finite Element Analysis of Electric Machinery

金 眞 泰\* · 鄭 泰 庚\*\* · 金 炯 碩\*\*\*  
(Jintae Kim · Tae Kyung Chung · Hyeong Seok Kim)

**Abstract** - To obtain more accurate result in the finite element analysis for electric machinery, it is important to have a mesh being of good quality. This paper describes a new technique of mesh generation for the finite element method. When the list of points defining the region of analysis is given, an appropriate distribution of interior points is generated first. Secondly the points are connected to form the triangles. Finally the connectivity data are used to reposition the interior points using Laplacian smoothing and mesh relaxation technique. In this paper, a mesh searching technique of Lawson which modifies the start mesh is proposed in addition to the above three steps. This algorithm is simple and produces the meshes being of good quality with high speed in comparison with the existing one.

**Key Words** : Finite element method, Electric machinery, Mesh generation, Laplacian smoothing, Mesh relaxation

### 1. 서 론

유한요소법(Finite Element Method)[1]을 사용하여 전기기기를 해석하기 위한 전제조건인 해석영역을 유한개의 삼각형 요소로 나누는 요소분할은 자동 요소분할 방법[2], 라플라스 차분 연산자를 이용한 방법[3], 다항보간 함수를 이용한 사상에 의한 방법[4] 등 다양한 여러 가지 방법에 의하여 행하여 졌다. 전술된 방법은 해석하려는 영역의 형상에 따라 각각 적용되는 내부함수를 정해 주어야 하고, 복잡한 계산이 요구되므로 적용하기가 불편하다. 또한 삼각형 요소를 사용할 경우 너무 큰 둔각이 존재하면 유한요소법 적용시 계산의 정확도가 떨어지게 되는데, 위에 언급된 방법은 변환에 의해 요소모양이 찌그러져 계산의 정확도가 떨어질 우려가 있다. 근래에는 해석영역을 경계의 점으로 표현하고 이 점에 의해 해석 영역을 삼각형 요소로 분할하였을 경우 세변의 길이가 비슷한 삼각형을 만드는 들러니삼각화(Delaunay triangulation)를 이용한 요소분할 방법이 제안되었다[5]. 이 방법이 현재 2차원 평면의 요소분할에서 가장 많이 사용되고 있는 방법이다. 또한 신경 회로망을 이용한 방법[6][7]도 시도되고 있으나 요소의 형상과 요소생성 속도, 각도율 등을 고려해 볼 때 아직까지는 장점이 없다고 할 수 있다.

다른 하나의 방법으로 유한요소해석후 해석오차를 고려한 후처리 과정을 통하여 요소의 분포를 이분법을 이용하여 늘

려가는 적응요소생성(adaptive mesh generation)[8][9]이 사용되고 있다. 들러니삼각화를 이용한 방법으로는 오차가 큰 요소의 무게중심에 점을 발생시켜 세분하는 방법과 오차가 큰 요소의 외접원의 중심에 점을 발생시켜 세분하는 방법, 서로 겹쳐지는 외접원을 가지는 삼각형 요소를 통합하여 분할하는 방법등이 있다. 삼각형의 무게중심에 점을 발생시켜 세분하는 방법의 경우는 오차가 큰 요소가 경계에 붙어 있는 경우에 새로 생성되는 점으로 인하여 삼각형 요소의 형상이 더욱 찌그러지지 않도록 경계면에 점을 발생시키는 부수적인 과정이 필요하고, 언급된 나머지 두 방법의 경우는 외접원의 중심이 요소 밖에 있는 경우 그 점이 경계를 벗어나는지 여부를 계산하여야 하므로 상대적으로 계산시간이 길어지게 된다.

해석모델 형상의 다양성으로 인해 등변에 가까운 요소를 생성하는 것이 요소분할 속도를 개선하는 것 만큼 중요하게 되었다. 그러나, 요소수가 많은 해석모델의 비선형 유한요소 해석을 통한 최적화 전기기기 설계는 여전히 많은 시간을 요구한다. 그러므로 해석영역의 요소망의 요소들을 효율적으로 분포되도록 한다면, 그것이 요소생성의 최적이라 할 수 있다[8].

본 논문에서는 비구조화된요소망(unstructured mesh) 중 구속들러니삼각화(constrained Delaunay triangulation)[10] 알고리즘에 근거하여 수정 보완된 자동요소분할 방법을 실현하였다. 이분법(bisection method), 고정조각화(Dirichlet Tessellation)나 기존의 증가알고리즘(incremental algorithm)에 의해 생성된 요소망에 비해, 본 방법을 통해 얻어진 요소망은 요소의 균등분포와 등변 삼각형요소의 비율이 상대적으로 증가하였다. 기존 증가알고리즘에선 실현하기 힘든 요철 형상, 원형 형상의 효율적인 요소분할을 가능하게 하였다.

\* 正 會 員 : 한글과 컴퓨터 연구원 · 碩士

\*\* 正 會 員 : 中央大 電子電氣工學部 教授 · 工博

\*\*\* 正 會 員 : 順天鄉大 電氣電子工學部 副教授 · 工博

接受日字 : 1998年 12月 23日

最終完了 : 1999年 3月 9日

## 2. 요소의 생성과 개선

### 2.1 요소의 생성

유한요소해석에서 쓰이는 삼각형 요소수의 증가는 시스템 방정식을 푸는데 걸리는 시간을 증가 시킨다. 그러나 최근 컴퓨터 성능의 향상으로 삼각형 요소 생성시에 걸리는 시간은 비선형 유한요소해석을 계산하는 시간에 비하여 그 비율이 상대적으로 무시할 만 하다. 그러나 초기에 생성되는 삼각형요소의 형상과 분포는 해의 정확도에 영향을 미친다[8].

또한, 적응요소생성 방법을 적용할 때 요소세분 방법으로 이분법을 적용하면, 세분된 요소망의 요소들의 형상 왜곡 정도는 초기요소망에 절대적으로 의존한다[8]. 이에 본 논문에서 개선된 요소망 생성방법을 제안한다. 삼각형 요소생성을 일반적인 경우에 이분법을 통하여 3~5 회 요소생성을 반복함으로써 형상의 왜곡이 더욱 심해지게 된다. 이 경우 초기 요소망의 질은 아주 중요하다. 만족할만한 삼각형 요소를 얻기 위하여 어떻게 점을 생성할 것인가에 대한 많은 알고리즘들이 연구되었다.

#### 2.1.1. 기존의 요소생성 방법

삼각형 요소생성의 가장 기본이 되는 Solid Grid법은 전 영역을 덮는 격자를 사용하여 점을 생성하는 것이다. 이 생성된 격자에 규칙적으로 점을 생성하거나 혹은 단위 격자내의 임의의 위치에 점을 생성한다. 이런 균등 요소(Uniform mesh)의 생성은 전 영역을 덮고 있는 격자로부터 쉽게 생성할 수 있으며, 생성된 요소의 외형에 대해 수정을 가하여 최종 요소를 얻을 수 있다[11]. 다양한 영역에 대한 각기 다른 일정한 격자를 사용하는 방법과 사상에 의한 방법등도 사용되었으나 이는 임의적인 영역에 대한 자동화가 실행되기 어렵다는 단점을 가지고 있다.

이분법을 적응요소 생성을 위하여 적용하기 위해서는 초기 요소의 형상이 아주 중요하다. 그러므로 초기 요소 생성시에 등변에 가까운 삼각형 요소를 생성할 수 있는 알고리즘을 사용한 후, 유한요소해석을 통하여 오차율을 계산하고 삼각형 요소수를 증가시킬 경우에는 이분법을 사용하여 요소수를 증가시킴으로써 안정적으로 삼각형 요소수의 증가와 요소의 형상에 대한 보장을 할 수 있다.

루퍼트(Ruppert)의 들러니세분(Delaunay refinement)알고리즘[12]은 다음과 같다.

1. 삼각형의 가장 긴 변을 가장 짧은 변으로 나눈 aspect ratio가 일정한 한계를 가진다. 이것은 유한요소 해석을 위해 고려해야 할 요소의 질을 높이는 역할을 한다.

2. 생성되는 요소의 개수를 일정하게 조절할 수 있다. 이것은 유한요소 해석에 걸리는 총 시간을 조절할 수 있게 한다.

이 알고리즘은 무한 요소분할의 문제점을 포함하고 있다. 무한 분할을 막는 방법은 들러니세분알고리즘에서는 제안하고 있지 않으나 초기 입력 값으로부터 이런 문제가 생긴다는 것을 고려해 보면, 문제의 해결 방법은 입력치로부터 삼각형 요소생성의 전처리 상태에서 이런 각을 만들어 주는 점들을 제거하고 제거한 점에 인접한 무한 분할되지 않는

점을 생성하는 특별한 고려를 알고리즘에 추가하는 것이다.

위의 두 가지 방법은 해석모델의 요소분할이 초기 점과는 크게 상관없이 요소의 밀도를 높이고자 하는 부분에서는 요소수가 적게 생성되고, 의도하지 않은 영역에서 요소수가 많게 생성될 수 있다는 것을 알 수 있다.

기존의 알고리즘에 근거한 요소분할은 그림. 1 과 같은 요소분할을 가지게 된다. 두 개의 원형 요소가 뜻대로 분할되지 않는 것을 볼 수 있다. 원 B는 요소분할이 많이 필요한 부분에서 많은 요소를 분할한 부분이고 원 A부분은 주변의 요소들은 많이 분할된 반면 원 A의 부분은 요소 분할이 많이 되지 않는 결과를 보이고 있다. 기존 자동 요소분할 방법의 큰 단점인 생성 요소분포의 임의성을 극복하기 위해서 절점배분(node spacing)을 고려한 알고리즘[13]을 사용하였다.

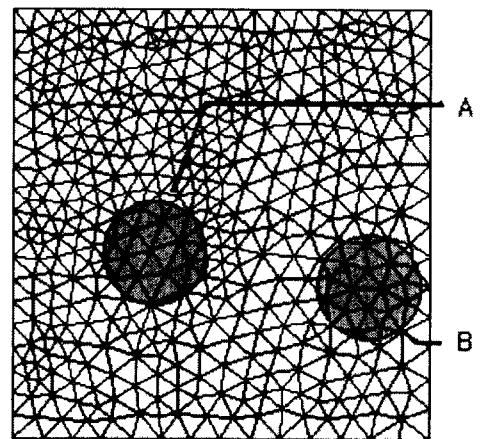


그림 1 기존의 방법(Tesselation)으로 요소분할 한 형상  
Fig. 1 Element mesh generated by Tesselation method.

요소를 생성하기 위한 점의 예상 위치는 삼각형의 외접원의 중심과 내접원의 중심 중에서 선택적으로 사용하게 된다. 외접원의 중심에 점이 위치하게 되면 이 점은 삼각형의 모든 점에서 같은 거리를 가지게 된다는 특징이 있다. 삼각형이 정삼각형에 가까울수록 내접원의 중심에 점이 위치하는 것도 삼각형의 다른 세 꼭지점과 같은 거리를 가지게 된다. 외접원의 중심에 점을 발생시키는 경우는 삼각형 요소의 외부에 점이 위치할 수 있는 반면, 내접원의 경우에는 언제나 삼각형의 내부에 점이 존재하기 때문에 해석영역에 점이 포함 되는가를 계산할 필요가 없어진다..

이런 점을 고려해서, 공간함수(spacing function)을 이용한 삼각형의 요소분할 방법은 삼각형 요소분할시 거리 개념이 도입되어 요소가 의도하지 않은 쪽에서 많이 생성되는 현상을 막을 수 있으며, 또한 최종 요소의 생성수를 간접적으로 제어할 수 있고, 초기 요소망의 형상에 크게 관계없이 요소형상의 질을 보장할 수 있다는 장점이 존재한다. 그러나 이 요소생성 방식은 다음과 같은 개선 해야할 부분이 있다. 삼각형 요소에 둔각이 존재하는 경우에는 삼각형 요소 외접원의 중심이 삼각형 요소의 외부에 존재하게 되므로 해석 영역 내에 존재하는 모든 요소에 대해서 내부에 속하는가를 검색하거나 혹은 그 점이 영역 바깥의 점인가를 검사

해야 한다. 이와같은 판정 루틴은 실행시간을 증가시키므로 일반적으로는 내부에 외접원의 중심을 가지는 요소에 대해서만 요소분할을 시행하게 된다. 이 방법은 계산량이 줄어들고 연산이 간략해지는 장점이 있으나 삼각형 요소의 질을 보장할 수 없게 된다.

그림. 2에서와 같은 길게 늘어진 삼각형 ABC는 초기 요소망 생성과정에서 많이 발생되는데, 이와같은 삼각형을 등변에 가깝도록 하기 위해 생성된 내부점의 위치는 내접원(반경=r)의 중심 I와 외접원(반경=R)의 중심 O 사이에 위치하게 된다. 바람직한 요소를 만들기 위해 생성하게 되는 점 P의 위치는 가중치를 내접원의 중심과 외접원의 중심에 각각 두어 생성점의 위치를 결정하게 된다.

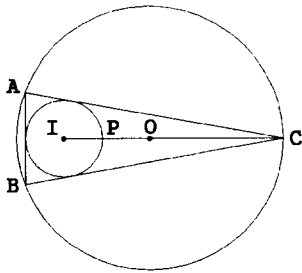


그림. 2 내접원과 외접원 사이에 생성점의 위치.

Fig. 2 A node placement between the inscribed circle and the circumcircle in a triangle element.

즉,  $\rho = \frac{2r}{R}$  이라 할 때, 선분  $\overline{IP}$  와 선분  $\overline{IQ}$  의 길이의 비가  $(1-\rho) : \rho$  가 되도록 새로 생성된 점 P의 위치를 정한다.

## 2.2. 요소의 형상 개선

2.1 절에서 생성된 점들의 위치와 선분들의 연결을 조정함으로써 요소의 형상을 개선한다..

### 2.2.1 라플라스 조정(Laplacian smoothing)

거의 모든 요소분할 알고리즘의 최종단계에서 사용되는 라플라스 조정은 내부점을 그 점과 연결된 인접 점들의 좌표 값들의 평균값의 위치로 이동시키는 과정을 말한다. 이 과정의 적용을 통해 유한요소해석에서 요구하는 등각의 삼각형으로 초기에 생성된 요소들의 형상을 조정할 수 있다. 실제 계산 수식은 .

$$\vec{X}_i^{new} = \frac{1}{n_i} \sum_{j=1}^{n_i} \vec{X}_j \quad (1)$$

로 표현된다.

여기서  $\vec{X}_i^{new}$  는 이동될 내부점 i의 좌표이며  $n_i$  는 점 i에 연결된 점의 개수이고  $X_j$  는 점 i에 연결된 점의 좌표이다.

## 2.2.2 Mesh relaxation

모든 삼각형 요소에서 임의의 점의 degree  $\delta$ 는 그 점에 연결되어 있는 인접한 점의 수를 의미한다. 삼각형 요소 내부에 점이 발생되었다고 하면 이것은 그 점을 공유하고 있는 삼각형 요소의 개수를 나타내기도 한다. 이 degree  $\delta$ 는 실질적으로 그 점에 연결된 요소의 평균적인 각도를 의미한다. 그러므로 이 degree  $\delta$ 를 제어함[14]으로써 전반적인 요소의 질을 향상시킬 수 있는 것이다. 이 degree  $\delta$ 의 제어는 변교환(edge swapping) 알고리즘을 사용하며 그 원리는 다음과 같다.

변교환(edge swapping) 알고리즘은 전체 삼각형 요소에서  $4 < \delta < 7$  인 수가 가장 많도록 계속해서 변교환(edge swapping)을 하는 것이다. 일반적인 경우에는 이 알고리즘을 통해서 모든 degree  $\delta$ 가 5, 6, 7 사이에 위치하게 할 수 없으며 이런 변교환(edge swapping)은 순환적인 맞교환(swapping)을 일어나게 하는 경우가 있으므로 한 번 실행한 점에 대해서는 다시 실행하지 않음으로서 삼각형 요소생성 시 속도의 개선과 삼각형 요소의 형상 향상을 동시에 얻을 수 있게 된다.

## 2.3. 요소생성의 속도개선

### 2.3.1 로슨(Lawson)의 찾기(search) 알고리즘

대부분의 삼각화 과정에서는 어느 삼각형 요소의 내부에 점을 발생시켰는지 확인하기 위하여 전 영역의 삼각형을 검색하는 것이 필요하다. 요소생성 과정 중에서 대부분의 시간을 소모하는 것이 이 찾기 과정이다. 로슨(Lawson)의 찾기 알고리즘[15]을 사용하면 효과적으로 내부에 생성된 점을 포함하는 삼각형 요소를 찾을수 있어 전반적인 요소생성 시간을 줄일 수 있게 된다. 로슨(Lawson)알고리즘을 구현하기 위해서는 인접 삼각형의 데이터를 구축하고 있어야 하나 속도의 개선이라는 장점이 훨씬 크기 때문에 사용되어 지고 있다.

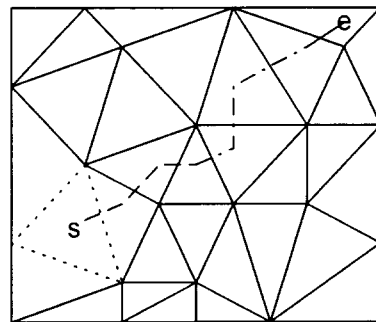


그림. 3 로슨의 찾기 알고리즘

Fig. 3 Lawson's searching algorithm.

로슨(Lawson)알고리즘은 다음과 같은 특징을 가진다. 그림. 3에서 생성된 점 e를 포함하는 삼각형 요소를 찾기 위하여 선택된 점선으로 표시한 최초의 삼각형 S를 시작점으로 할 때 내부의 점을 가지는 삼각형을 찾아가는 가장 짧은 거리는 직선이 지나가는 삼각형들을 따라 검색할 때이다. 한

개의 삼각형 요소가 인접한 세 개의 삼각형 요소의 데이터를 가지도록 구축해 놓고 방향성을 고려하여 인접 삼각형 요소로 옮겨가는 알고리즘을 구현할 수 있다.

그림. 3에서의 시작점을 전체 영역에서 비슷한 개수의 요소들이 들어가는 여러 개의 영역으로 나누고 각각의 영역에 시작점을 두어 새로 생성된 점과 가장 가까운 삼각형을 시작점으로 하는 변형 알고리즘[10]도 제안되었다.

### 3. 요소 생성의 구현

일반적으로 요소의 생성은 기본점 생성, 외형 초기점 생성, 내부점 생성의 3단계로 구분할 수 있다.

#### 3.1 프로그램 흐름도

그림. 4는 요소생성 프로그램의 흐름도를 나타낸 것이다. 본 논문에서는 요소의 검색부분에는 후술된 변형 알고리즘을 이용하였고, 요소 생성부분에서는 생성된 인접요소를 기준으로 점간의 간격을 전파시키는 방법을 사용하였다. 그림. 5의 개선된 찾기 알고리즘은 그림. 4의 요소생성 흐름도 중 내부에 점발생 블록에 적용된다.

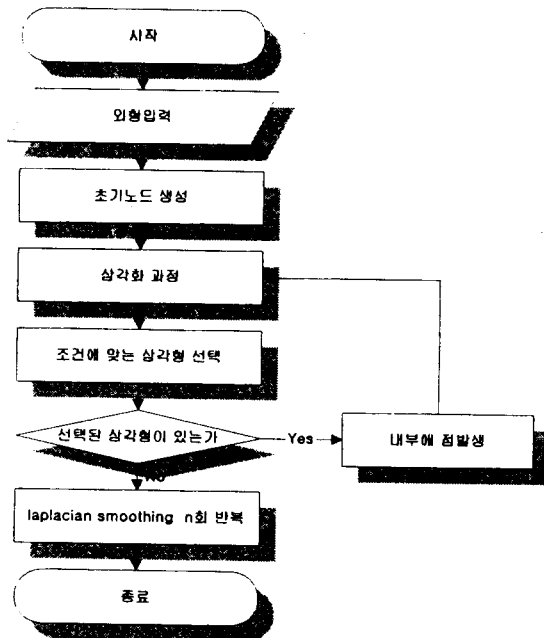


그림. 4 요소생성 프로그램의 흐름도  
Fig. 4 Flowchart of mesh generation

### 3.2. 알고리즘의 적용과 개선

#### 3.2.1 요소 찾기의 개선

일반적인 찾기 알고리즘의 구현은 해석 대상 영역 내부의 모든 삼각형 요소에 대해서 각각의 요소가 생성점을 내부에 가지고 있는가를 판정함으로써 이루어진다. 알고리즘의 수행시간이 초기 요소를 생성할 때 보다 내부점의 개수가 증가함에 따라 현저하게 증가한다.

찾기 알고리즘의 속도를 개선하기 위하여 로슨(Lawson)의 찾기 알고리즘이 구현되었다. 본 논문에서는 로슨(Lawson)의 찾기 알고리즘의 기본 개념만을 취하고, 구현에서는 다른 방법을 적용하였다. 개선된 알고리즘의 흐름도는 그림. 5와 같다. 또한 찾기 속도의 개선을 위하여 기본 알고리즘을 수정하여 보다 빠른 찾기를 가능하게 하였다. 새로 생성된 점에 가까운 삼각형 요소를 찾기 위한 방법으로 그래픽 이론에서 적용되는 인아웃테스트(in-out test) 알고리즘[16]을 사용하였다.

그림. 6은 새로 삽입될 목표점을 포함하는 삼각형 요소에 도달할 때까지 검색한 삼각형 요소의 수를 나타낸다. 그림. 6에서 전영역 찾기 알고리즘의 그래프가 심하게 기복하는 것은 어느 삼각형 요소에서 찾기가 끝날지 예측할 수 없는 알고리즘 특성 때문이다. 이 특성은 요소의 번호와 위치가 전혀 연관성없이 존재하는 비구조화된 요소망(unstructured mesh)의 삼각형 데이터 저장방법에 원인이 있다.

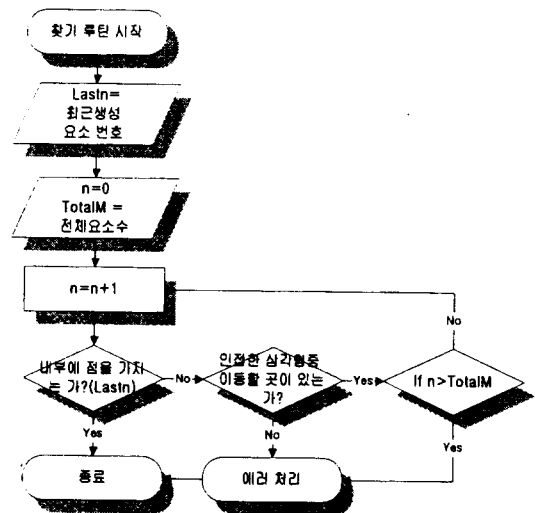


그림. 5 개선된 찾기 알고리즘의 흐름도  
Fig. 5 Flowchart of improved searching algorithm.

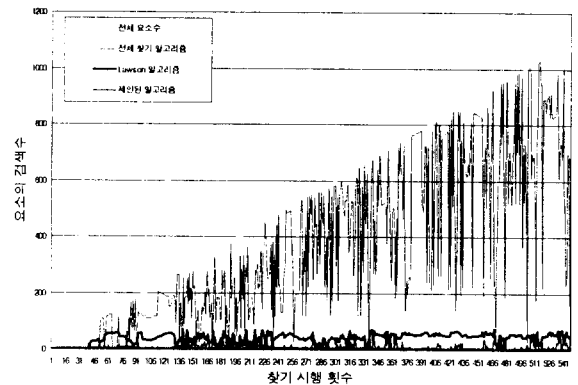


그림. 6 새로 생성될 요소를 찾기까지 검색한 요소의 수  
Fig. 6 Number of elements inspected by looking for the newly producing element.

기존의 변형 로슨(Lawson)알고리즘에서는 요소들을 n개의 영역으로 분할하여 각 영역에 대표 값을 두고 이 대표

값들과 새로 삽입될 점과의 거리를 비교하고 가장 가까운 점을 가진 삼각형 요소에서부터 새로 생성된 점을 포함한 삼각형 요소를 찾기 시작하는데 이 알고리즘은 선행조건에서 자료의 x축, y축 별 sort와 분할된 영역에 균등한 갯수의 삼각형 요소를 배치하기 위한 선행 계산이 필요하다.

이 알고리즘은 찾기 자체에서는 우수한 성능을 발휘하는데 반해 그 찾기를 가능하게 하는 선행 조건에서 요소의 수가 증가함에 따라 다시 영역을 분할해야 하는 등 여러 가지 복잡한 계산을 수행해야 함으로 구현에 어려움이 있다. 그러나 본 논문에서는 단지 최근에 생성한 삼각형 요소를 기억하고 요소를 찾는데 기억된 삼각형 요소를 시작 삼각형 요소로 선택함으로써 보다 효과적인 찾기가 이루어진다.

### 3.2.2 요소생성의 개선

그림. 7은 요소생성을 위한 흐름도이다. 흐름도에서 알 수 있듯이 생성점 간의 간격을 제한하고, 생성된 요소가 필요 없는 부분에 많이 형성되는 것을 막기 위하여 이미 생성된 요소의 간격으로부터 새로 형성될 점의 간격을 전파해 가는 형태로 수정하였다. 기존의 증가(incremental) 알고리즘은 생성된 요소의 형태에서 개선된 결과를 얻을 수 있다.

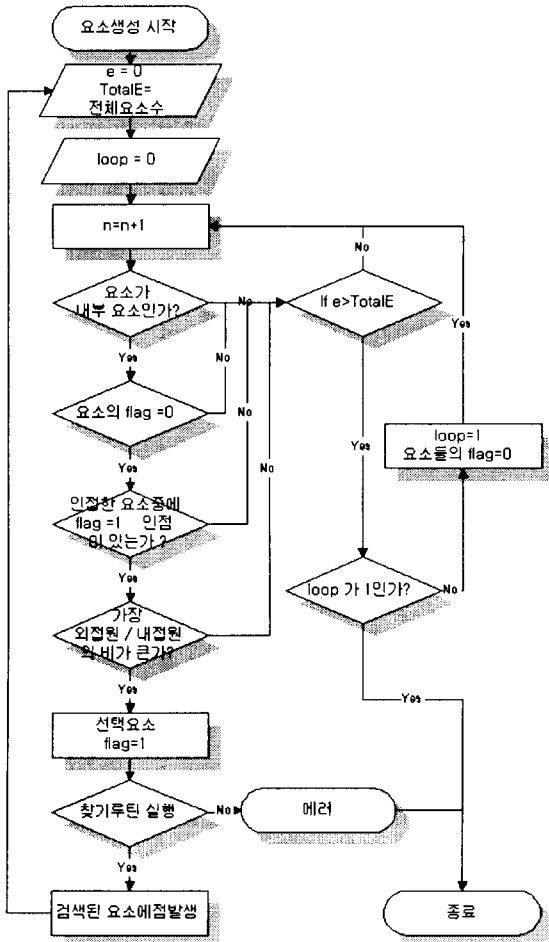
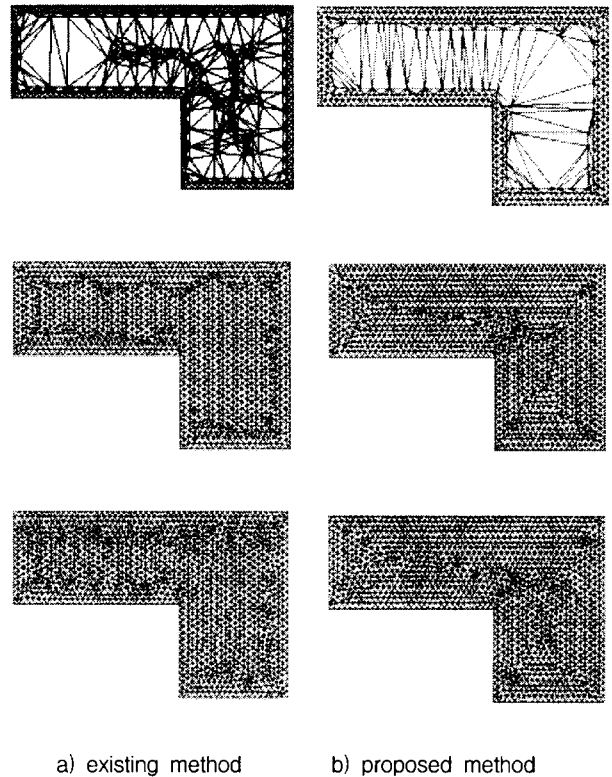


그림. 7 개선된 요소생성 알고리즘의 흐름도  
Fig. 7 Flowchart of improved element producing algorithm.

증가(incremental) 알고리즘은 알고리즘의 특성상 발생된 내부점에 의해 분할된 요소를 인접요소로 갖는 삼각형 요소 중에 가장  $r/R$  비율이 작은 삼각형으로 요소 생성이 진행된다. 따라서, 그림. 8 a)와 같은 요소의 생성 과정을 보이게 되며, 삼각형 요소의 생성은 요소가 한 부분으로 자라감에 따라 삼각형 요소가 자란 흔적이 전체 형상 중에 특정 영역에 집중되는 현상과, 요소가 모여야 할 부분에 요소가 적어지는 현상을 나타낸다. 이것을 개선하기 위하여 본 논문에서는 그림. 7 흐름도의 Loop flag를 사용하여 제어하는 방법을 제안하였다.

그림. 8 b)는 본 논문에서 새로 제안한 알고리즘에 의해 형성된 요소망이다. 그림. 8 b)를 보면 제안된 방법은 새로 생성된 요소가 인접한 요소로부터 자라나기 때문에 그것이 채택한 요소간의 최소거리를 유지할 수 있으나 이렇게 자라난 요소 때문에 결국은 요소간의 거리가 완전하게 일정할 수는 없다. 제안된 방법은 해석모델의 외형에서부터 시작하여 일괄적으로 요소의 형상이 제일 나쁜 요소에 내부점을 발생시켜, 모든 요소에 한 번의 요소분할 기회를 제공함으로써 요소망에서 요소의 엉킴 현상을 최소화 할 수 있다.



a) existing method      b) proposed method  
그림. 8 기존의 방법a)과 제안된 방법b)의 비교  
Fig. 8 Comparison with a proposed and existing methods.

그림. 8의 중간 두 요소망은 외형 삼각형을 포함하여 약 2400개 정도의 삼각형 요소를 생성하여 최종 요소분할을 한 형상이다. 이 결과는 라플라스 조정(Laplacian smoothing)과 요소망 이완(mesh relaxation)과정을 거치지 않은 순수 출력 결과이다. 그림. 8의 최종 두 요소망은 각각 후처리를 거친 증가(incremental) 알고리즘의 최종요소망과 제안된 알고리즘의 최종 생성요소망을 보여주고 있다.

그림. 8의 마지막 그림은 요소망 이완(mesh relaxation) 작업 후에 라플라스 조정(Laplacian smoothing)을 행하여 출력된 결과이다. 이 실행 결과에서 기존 알고리즘에 의해 얻어진 요소망인 그림. 8 a)는 초기에 형성된 입력점의 주변에 요소가 몰려있는 것을, 제안된 알고리즘에 의해 얻어진 요소망인 그림. 8 b)는 가운데 부분에 삼각형 요소가 몰려있는 것을 볼 수 있다. 두 알고리즘 모두에서 나타나는 요소의 집중 현상은 요소의 간격 정보가 다음 요소로 전파되는 과정에서 발생하며 이것은 삼각형을 구성하는 단 세 점의 간격 정보만을 참조하지 않고 세 개의 인접 요소의 간격 정보도 참조함으로써 더욱 향상될 수 있다.

제안한 알고리즘에서 사용한 삼각형의 최적판정루틴은 r/R 비율을 0.7을 사용하였으며, 이상적인 삼각형의 경우에는 약 0.5773의 값을 가지게 된다. 이 비율을 초기와 후기, 외형의 서로 다른 부분에서 자라나기 시작한 요소들이 만났을 경우에, 서로 다르게 적용함으로써 요소의 형상 개선 효과를 얻을 수 있다.

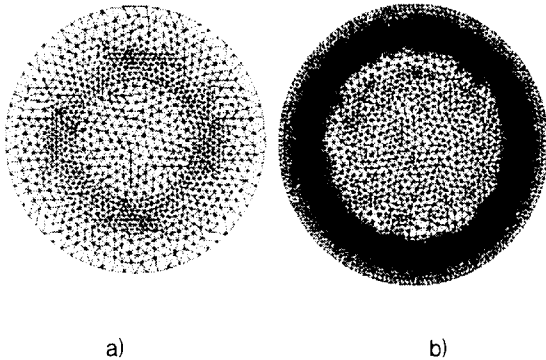


그림. 9 기본적인 모터 형상을 요소분할한 모습 :  
a) 요소수2017, 점 1028 b) 요소수 17322, 점 8214  
Fig. 9 A shape of element of general motor.

그림. 9는 기본적인 모터형상에 대하여 제안된 알고리즘을 적용하여 삼각형 요소로 분할한 모습이다. 그림. 9 a)의 요소망은 약 2000개의 요소를, 그림. 9 b)의 요소망은 약 18000개의 요소를 갖고 있다.

3.4. 생성 요소의 결과비교

최종 결과는 PC에서 stand-alone 모드, normal priority에서 Windows Message를 제거하고 실행 되었다. 최종 요소망 생성시간은 프로그램 시작에서 끝날 때까지 걸린 시간이 아닌 실제 연산에 소요된 절대 시간을 의미한다.

표 1은 그림. 8의 요소생성 과정에서 출력된 자료이다. 표 2에서 볼 수 있듯이 증가(incremental) 알고리즘에 대해서 약 10%의 속도 개선 효과가 있었다.

표 2는 80 x 50의 직사각형에 가로, 세로의 초기점으로 80, 50개를 입력하였을 때 발생하는 요소 수와 점수이다. 내부에 발생하는 점의 수가 다른 것은 요소를 발생하기 위하여 구현된 알고리즘 차이에서 기인한다.

그림. 10은 최종 요소망으로부터 구한 요소의 각도 분포도이다. 개선된 알고리즘은 60도에 약 38%정도의 비율을 가

지고 있으나 본 논문에서 제안된 방법은 약 48%정도의 비율을 차지하고 있다. 비교를 위하여 첨가된 tessellation방법을 이용한 삼각형 요소분할에서는 오히려 48도에서 약 11%정도의 비율을 가지고 있다는 것을 알 수 있다. 그러나 평균각은 이 방법도 거의 60도에 해당한다.

표 1 각각의 방법에 대한 요소 생성시 해석결과  
Table 1 Analysis result of each method.

	Incremental 방법	본 논문에서 제안된 방법
최종 생성시간	11초	9초
최종 요소수	2755	2684
요소의 평균각도	59.9988	60.0029
최종 점 수	1380	1327
최빈각	60	60
최빈각의 비율	36%	48%
전체요소중 최대각	90	90
전체요소중 최소각	40	41
최빈연결수(Degree)	6	6
점의 최대 연결수	7	7
점의 최소 연결수	5	5

표 2 점수를 증가 시켰을 때 해석결과  
Table 2 Analysis result increasing the points.

	Incremental 방법	본 논문에서 제안된 방법
최종 생성시간	3분 31초	2분 27초
최종 요소수	9737	9575
최종 점 수	4871	4790

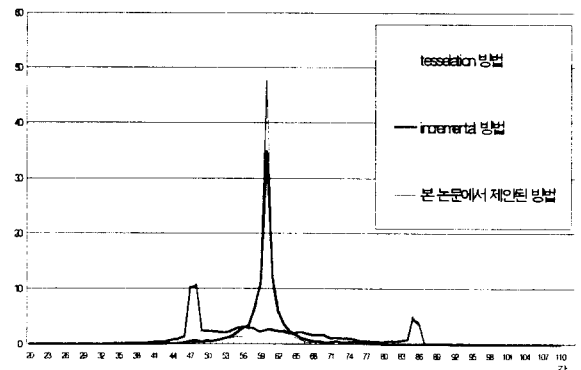


그림. 10 세가지 방법으로 생성된 최종요소에 대한 각도분포도  
Fig. 10 Distribution of degrees with respect to the three methods

그러나 각도의 비율에서 알 수 있는 degree의 분포를 보면 최대 degree는 9, 최소 degree는 4이다. 그러므로 이론적인 각도는 최대각은 90도, 최소각은 40 도이나 실제 요소생

성 과정에서는 이론에서와 같이 완벽한 삼각형이 만들어지는 것이 아니므로 108도와 26도를 가지게 되었다.

본 논문에서 사용된 방법으로 얻어진 요소는 결과적으로 최대 degree는 7, 최소 degree는 5이며 이론적으로는 최대 72도, 최소 51.43도의 각도를 가지나 전술한 바와 같이 항상 이상적인 삼각형을 생성하는 것은 아니므로 실제 각도는 최대 90도, 최소 41도를 가지며 최빈각 60도는 전체 요소 중에서 약 50%를 차지한다.

제안된 알고리즘에서 요소분할 대상의 외형이 원형이며 생성하고자 하는 점의 수가 많을 경우에 점 거리 제한에 걸려 내부에 점이 적게 생성되거나 혹은 초기점 간격 설정에 의한 강제점 생성 조건에 의해 점이 과다하게 발생하는 문제가 있다. 그러나 이것은 삽입될 점과 내부에 점이 발생되기 전의 삼각형의 꼭지점을 구성요소로 하는 작은 삼각형들의 비율로 구해지는 점 거리조절 함수의 값을 새로 삽입된 점에 의해서 생성되는 삼각형들의 정삼각형 비율에 따르도록 고침으로써 방지할 수 있다.

#### 4. 결 론

본 논문에서는 전기기기의 유한요소 해석을 위한 사전 단계인 요소생성에 있어서 삼각형 요소의 생성속도와 형상의 질 향상을 구현하였다. 증가(incremental) 알고리즘에 대해서도 최종 요소망의 요소의 개수에 따라 약 10% - 40%까지의 속도 개선의 효과를 얻을 수 있었다. 요소 수를 증가시켜 해석의 정확도를 높이고자 하는 경우에 속도 개선 효과를 뚜렷하게 볼 수 있었다.

본 논문에서 제안된 알고리즘은 변형 알고리즘에 비해 알고리즘 구현면에서 단순함과 실행 속도면에서 우위를 가진다.

요소의 형상면에서는 각도의 분포가 보다 60도쪽에 첨예하게 집중되어 있는 것을 그림. 10에 나타난 각도 분포 그래프를 통해서 확인할 수 있었다. 또한 60도의 비율이 증가(incremental) 알고리즘은 약 35%인데 반하여 본 논문에서 제안된 방법은 48%가량임을 알 수 있었다. 삼각형 요소의 형상을 향상시킴으로써 전기기기의 유한요소해석 결과의 정확도를 개선할 수 있을 것으로 사료된다.

#### 감사의 글

본 연구는 1997년도 과학재단 핵심전문연구(과제 번호: 971-0910-061-3)의 지원에 의하여 수행되었음.

#### 참 고 문 헌

[1] A. J. Davies, *The Finite Element Method : A First Approach*, CLARENDON. OXFORD 1989.  
 [2] P. J. Green and R. Sibon, "Computing Dirichlet tessellations in the plane, *Computer Journal*, Vol. 21, pp. 168-173, 1977.

[3] L. R. Herrmann, "Laplacian isoparametric grid generation scheme," *J. Eng. Math. Div. ASCE*, Vol. 102, pp. 749-756, 1976.  
 [4] G. Steinmueller, "Restrictions in the application of automatic mesh generation scheme by isoparametric coordinates," *Int. J. Num. Methods Eng.*, Vol. 3, pp. 519-528, 1971.  
 [5] 김형석, 정현교, 이기식, 한송엽, "Delaunay의 삼각화를 이용한 3차원자동요소분할", *대한전기학회지*, 37권, 12호, pp.847-853, 12월, 1988.  
 [6] D. N. Dyck, D. A. Lowther, and S. McFee, "Determining an approximate finite element mesh density using neural network techniques," *IEEE Trans. Magn.*, Vol.28, pp. 1767-1770, March 1992.  
 [7] S. Alfonzetti, S. Cocco, S. Cavalieri and M. Malgeri, "Automatic mesh generation by the let-It-grow neural network," *IEEE trans. Magn.*, Vol. 32, pp. 1349-1352, May 1996.  
 [8] M. C. Rivara, "algorithm for refining triangular grids suitable for adaptive and multigrid techniques," *Int. J. Num. Methods Eng.*, Vol. 20, pp. 745-756, 1984.  
 [9] 김형석, 최홍순, 한송엽, "정전자장의 적용유한요소해석을 위한 오차측정", *대한전기학회지*, 38권, 2호, pp.22-28, 1월, 1989.  
 [10] S. W. Sloan, "A fast algorithm for generating constrained Delaunay triangulations," *Computers & Structures*, Vol. 47, No. 3, pp. 441-450, 1993.  
 [11] S. H. Lo, "A new mesh generation scheme for arbitrary planar domains," *Int. J. Num. Methods Eng.*, Vol.21, pp. 1403-1426, 1985.  
 [12] Jim Ruppert, "A delaunay refinement algorithm for quality 2-dimensional mesh generation," *Journal of algorithms*, Feb. 2, 1994.  
 [13] William H. Frey, "Selective refinement: A new strategy for automatic node placement in graded triangular meshes," *Int. J. Num. Methods Eng.*, Vol. 24, pp. 2183-2200, 1987.  
 [14] David A. Field and William H. Frey, "Mesh relaxation : A new technique for improving triangulations," *Int. J. Num. Methods Eng.*, Vol. 30, pp. 1121-1133, 1991.  
 [15] C. L. Lawson, *Software for C surface interpolation*. In *Mathematical Software III*, pp.161-194, Academic Press 1977.  
 [16] F. S. Hill Jr., *Computer Graphics*, Macmillan Publishing company, 1990.

## 저 자 소 개



김진태(金眞泰)

1973년 1월 22일 생. 1996년 중앙대 공대 전기공학과 졸업. 1998년 2월 동 대학원 석사. 현재 한글과 컴퓨터 연구원.



정태경(鄭泰庚)

1957년 3월 28일생. 1981년 서울대학교 전기공학과 졸업. 1983년 동대학원 석사. 1987년 2월 동대학원 박사. 1986년 - 1988년 대우중공업 중앙연구소 선임연구원, 1991년~1992년 7월, 미국 Harvey

Mudd College Visiting Scholar. 1988년 - 현재 중앙대 전자전기공학부 교수.

H · P : 016-590-5291

E-mail : tkchung@cau.ac.kr



김형석(金炯碩)

1962년 10월생, 1985년 2월 서울대 전기공학과 졸업. 1987년 2월 동대학원 석사. 1990년 8월 동대학원 박사. 현재 순천향대 전기전자공학부 부교수 및 본대학 부설 소프트웨어교육센터소장. 1997년 2월~1998년 2월

미국 R.P.I. 방문교수. 주요연구분야 : 전자장수치해석, 적응유한요소기법, 초고주파 소자 해석 및 설계, 상용소프트웨어 교육 및 교육용 프로그램 개발.