

프로파일 감소를 위한 절점 번호 넘버링 기법 개발

이병채 · 김형민*

Development of a Node Numbering Algorithm for Reducing Profile

Byung Chai Lee · Hyung Min Kim

〈요 약〉

유한 요소 시스템 행렬의 프로파일을 줄일 수 있는 새로운 절점 번호 자동 부여 알고리즘을 개발하였다. 이 알고리즘은 절점 연결 행렬 압축 기법과 웨이브 프론틱 개념에 의한 번호 부여 기법을 복합한 방법으로 알고리즘이 간단, 명료하다. 대표적인 유한 요소 모델을 선정하여 본 알고리즘과 기존 알고리즘을 비교하였다. 새로운 알고리즘의 번호 부여 시간은 기존의 우수한 알고리즘과 비슷하였으나 프로파일 크기는 작았다.

주요어 : 유한 요소 해석, 절점 번호 자동 부여, 프로파일 감소

1. 서론

지난 40년동안 활발히 연구되어온 유한 요소법은 범용성, 수렴성, 강건성, 다양성 등의 장점으로 인해 현재 가장 널리 사용되고 있는 공학해석 방법 중 하나이다. 이 방법이 적용될 수 있는 문제도 응력해석, 진동해석, 열해석, 유체해석, 전자장해석, 음향해석 문제 등 매우 다양하다. 유한 요소법은 근본적으로 미분 방정식의 근사해법이며 미분 방정식을 적절한 적분형 표현으로 변환한 후, 문제가 정의된 영역을 비교적 단순한 형상으로 분할하고 분할된 영역에 형상 함수를 도입하여 미지 함수를 유한차원으로 근사하는 방법이다. 이러한 근사에 의해 시스템의 거동을 나타내는 대수방정식을 얻을 수 있고 이를 풀어서 최종 해를 구한다. 이 때 분할된 단순 형상 하나하나를 요소, 미지 함수를 근사하는 유한차원 근사점을 절점이라

정의한다. 유한요소 형상함수는 요소 내에 국부적으로 정의되기 때문에 한 요소 내에 정의되는 미지변수 사이에만 상관관계가 있고, 요소를 통해 연결되지 않은 미지변수 간에는 아무런 관계도 없게 된다. 유한 요소법에서 최종적으로 푸는 다원 1차 연립 방정식은 다음과 같다.

$$[K] \{u\} = \{F\} \quad (1)$$

여기서 $[K]$ 는 구조의 강성행렬, $\{u\}$ 는 미지변수 벡터, $\{F\}$ 는 구조하중 벡터이다.

일반적으로 유한요소 시스템 방정식의 강성행렬은 전술한 유한 요소 근사의 특성에 의해 거의 모든 요소가 0인 매우 성긴(sparse) 행렬이 되므로, 이 성질을 잘 이용하면 계산 효율이나 저장 효율을 크게 높일 수 있다. 보통 모든 요소를 저장하지 않고 밴드형

* 한국과학기술원 기계공학과

대 또는 스카이라인 형태로 저장한다. 이때 저장 효율이나 계산시간은 저장되는 행렬의 형태학적 특성에 따라 크게 달라지며, 밴드 형태의 경우 밴드폭이, 스카이라인 형태의 경우 프로파일인 이러한 형태학적 특성을 나타내는 양이 된다. 강성행렬의 어떤 행에서 0 아닌 값을 가진 최대 열 번호와 대각선 번호의 차를 그 행의 밴드폭이라 할 때, 행 밴드폭의 최대값을 시스템 행렬의 밴드폭이라하고, 강성행렬의 각 열에서 대각선 항으로부터 0이 아닌 값까지의 최대 높이를 열의 높이라 할 때, 모든 열의 높이의 합을 프로파일이라 정의하고 있다. 강성행렬의 밴드폭이나 프로파일은 문제의 유한 요소 모델과 절점 번호 부여에 따라 유일하게 결정된다.

대체적으로 저장용량은 밴드폭이나 프로파일에 비례하며, 계산량은 이 양들의 제곱에 비례하는 것으로 알려져있다. 따라서, 절점번호를 잘 부여하여 강성행렬의 밴드폭이나 프로파일을 줄이면 컴퓨터의 주기억부를 효율적으로 이용할 수 있으며, 컴퓨터의 계산 시간과 계산 상의 오차를 줄일 수 있게 된다[1]. 특히 식(1)을 반복해서 풀어야 하는 비선형 문제나 최적 설계 문제의 경우 각 절점에 적절히 번호를 부여하면 계산시간을 크게 줄일 수 있게 된다.

요소와 절점이 많지 않은 문제의 경우에는 밴드폭이나 프로파일이 가능한한 작게 되도록 유한요소 모델 작성자가 직접 번호를 부여할 수 있으나, 요소와 절점수가 증가하면 수동으로 좋은 번호를 부여하는 것이 거의 불가능하게 된다. 특히 최근 경향으로 나타나고 있는 전처리 프로그램의 이용, 적응형 격자 재분할 기법의 효율적 이용에는 자동 번호 부여 방법이 필수적이라 할 수 있다.

유한 요소법의 발전과 함께 밴드폭 감소 방법들도 여러가지로 개발되어 왔는데, 이러한 방법들을 크게 세가지로 구분할 수 있다. 첫째는 축차적으로 절점 번호 부여 상태를 바꾸고 밴드폭을 비교해 가며 개선하는 방법으로 Akynuz[2], Grooms[3], Collins[4] 등의 방법들이 여기에 해당되며, 둘째는 이상적 형태에 대한 관찰, 수치적 경험으로부터 얻은 번호 부여 기준에 따라 모든 절점들에 일괄적으로 부여하는 방법으로 Akhras[5], Puttonen[6]의 방법들이 여기에 해당된다.

세째는 그래프 이론에 기초하여 기존의 번호와 상관 없이 일괄적으로 번호를 다시 부여하는 방법으로 Cuthill-McKee의 방법, Reverse Cuthill-McKee의 방법[7], Gibbs-Poole-Stockmeyer(GPS)의 방법[8], 구분용의 방법[9]이 여기에 해당된다. 일반적으로 축차방법은 일괄 번호 부여 방법에 비해 소요시간, 결과면에서 모두 뒤떨어지며 그래프 이론에 근거한 방법이 가장 우수한 것으로 알려져 있다.

프로파일 감소 방법들로는 선형연립 방정식의 풀이 과정인 Gauss 소거법의 개념을 이용한 King[10], Gibbs[11]의 방법이 있다. 밴드폭을 줄이는 방법 중에서 Reverse Cuthill-McKee의 방법[7], GPS의 방법[8] 등도 프로파일을 상당히 감소시키는 것으로 알려져 있다. 최근에 발표된 구분용의 방법[9]은 그래프 이론과 웨이브프론트(wave front) 개념을 이용하여 요소 넘버링과 절점 넘버링을 차례로 수행하는 방법으로 효율과 결과 면에서 매우 우수한 방법이다. 그러나 1번 절점을 선정하기 위한 요소 번호 부여의 필요성, 웨이브 프론트 개념에 의한 번호 부여시 복잡한 기준의 적용 등이 단점으로 지적될 수 있다.

본 논문에서는 구분용 방법의 결과를 더욱 개선하고 효율을 높이며 알고리즘을 간단 명료하게 하는 것을 목표로 하고 있으며 이를 위해 경험 기준에 의한 Puttonen의 번호 부여 방법과 그래프 이론에 근거한 일괄적 번호 부여 방법을 차례로 적용하는 방법을 시도하였다.

2. 프로파일 감소 알고리즘

2.1 절점연결행렬의 구성

유한 요소법에서 유한 요소의 갯수가 m 일 때 강성행렬 $[K]$ 는 다음과 같이 구한다.

$$[K] = \sum_{i=1}^m [K^{(i)}] \quad (2)$$

여기서 $[K^{(i)}]$ 는 구조 자유도로 표현된 i 번째 요소의 강성행렬이며 그 크기는 $[K]$ 와 같고, j 번째 자

유도와 k번째 자유도가 i번째 요소에 없으면 $K_{jk}^{(i)} = 0$ 이다[12].

한편 대칭행렬 [K]에서 i 번째 열의 높이 θ_i 를

$$\theta_i = \max\{ i-j+1 : K_{ij} \neq 0, j \leq i \} \quad (3)$$

로 정의할 때 프로파일 p 는 각 열의 높이의 총합으로 다음과 같이 정의된다.

$$p = \sum_{i=1}^n \theta_i \quad (4)$$

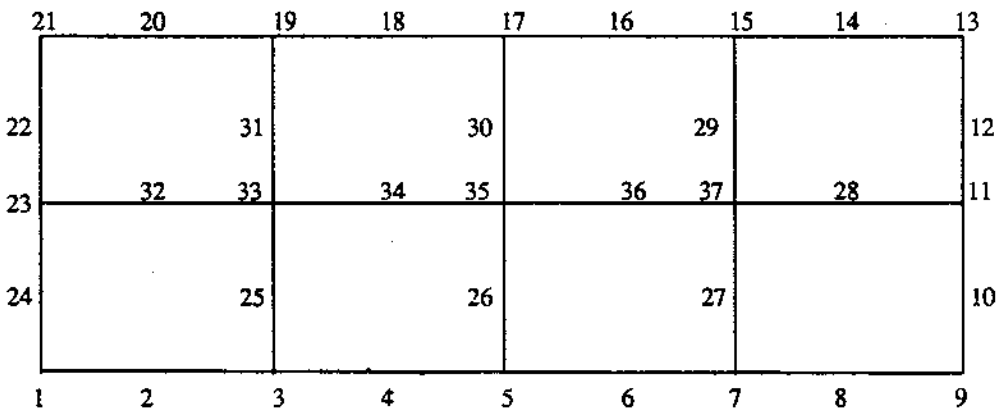
유한요소 모델에서 절점의 자유도 수가 1 이 아니거나 자유도 수가 서로 다른 절점들이 있으면 절점 번호 보다는 자유도 번호를 새롭게 부여해야 밴드폭이나 프로파일의 관점에서 더 좋은 결과를 얻을 수 있겠지만, 본 논문에서는 문제를 간단히 하기 위해서 모든 절점의 자유도가 1 인 경우로 가정하였다. 이 경우 절점 번호 부여의 기초가 되는 절점 연결 행렬은 강성 행렬과 같은 형태학적 구조를 갖는다.

그래프 이론에 의한 번호 부여 기법은 효율적이고 결과도 우수하나 1번 절점의 선택에 어려움이 많다. 이를 개선하기 위해 구본웅 등은 요소를 넘버링하고 그 결과를 이용하였다. 그러나 이를 위해서는 요소 연결 행렬을 새로 구성해야 하고 요소 넘버링에서 1 번

요소를 구하는 어려움이 여전히 남는다. 본 논문에서는 이의 개선을 위해 Puttonen의 절점 연결 행렬 압축 기법을 적용하여 새로운 번호를 구한다. 여러가지 방법 중 Puttonen의 방법을 선택한 것은 이 방법의 결과가 밴드 폭의 관점에서 비교적 우수하고 넘버링 소요 시간이 매우 작기 때문이다. 이 번호는 1번 절점을 찾는 데, 또 그래프 이론에 의한 넘버링에서의 절점 선정의 한 기준으로 사용된다.

2.2 절점연결행렬 압축기법에 의한 번호부여

절점 연결 행렬 압축 기법은 밴드폭을 줄이기 위해 Puttonen이 제안한 방법으로 절점 연결 행렬에서 0이 아닌 항들을 가능한 한 대각선 쪽으로 압축시키는 경험적 기준에 의한 일괄 번호 재부여 방법이다[6]. 절점 연결 행렬이란 각 절점이 유한 요소를 통해 서로 관련되어 있는지를 나타내는 행렬로서 절점수를 n이라 하면 $n \times n$ 행렬이 된다. i번 절점과 j번 절점이 요소를 통해 관련되어 있으면 이 행렬의 ij 항이 0이 아닌 값을 갖고 그렇지 않으면 0이 된다. 각 행의 압축은 절점연결 행렬의 각 행에서 0이 아닌 값들이 대각선 쪽으로 최대한 접근하도록 번호를 다시 부여함으로써 이루어진다. 이때 이미 번호가 부여된 절점은 번호가 변하지 않도록 하며 연결되어 있는 절점수가 작은 것부터 먼저 대각선 쪽으로 압축시킨다. 일단 이



<그림 1> 외팔보의 유한요소 모델. 2차 사각형 요소 8개, Ex.1.

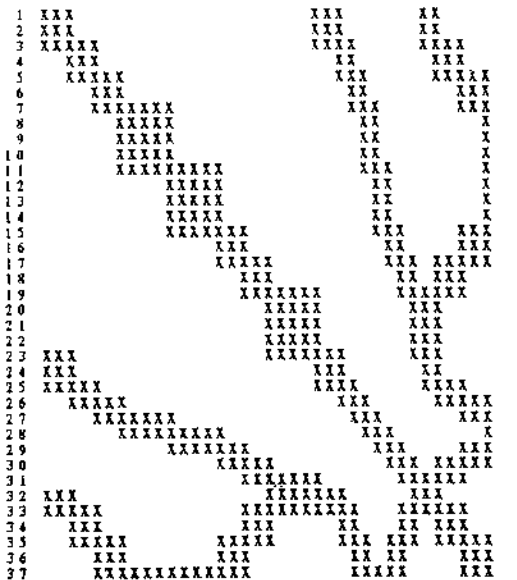
렇게 첫행부터 마지막 행까지 차례로 옮기고 나면, 절점 연결 행렬의 밴드폭은 상당히 줄어든다. 그러나, 첫번째 절점 근처에서는 별 변화가 없을 것이므로 이의 개선을 위해 마지막 행부터 역순으로 한번 더 연결 행렬을 압축한다. 원래의 Puttonen의 방법은 이후 밴드폭을 더욱 감소시키기 위해 1 행부터의 압축과정, 비교, 교환에 의한 축차 단계를 거치게 되나 본 연구에서는 1번 절점 추출과 비교적 우수한 넘버링을 얻는 것이 이 과정의 목적이므로 전방 압축 1회와 후방 압축 1회로 이 방법을 끝낸다. 일반적으로 밴드폭 감소를 위해 넘버링한 결과를 역으로 하면 프로파일의 감소하므로 전후방 압축에 의해 얻어진 절점들의 순서를 역으로 한다. 절점 연결 행렬의 압축에 따른 결과를 예시하기 위해 <그림 1>과 같은 평면 2차 사각형 요소로 구성된 간단한 유한 요소 모델에 대해 초기 번호와 압축 후의 번호에 대한 절점 연결 행렬을 <그림 2>에 보였다. 그림에서 X 표시는 해당 행과 열의 자유도가 요소를 통해 연결되었음을 의미한다. 그림에서 볼 수 있는 바와 같이 절점 연결 행렬이 대각선 쪽에 밀집되어 있고 이때 밴드폭이나 프로파일은

원래 번호에 비해 상당히 감소된다. 그러나 복잡한 문제의 경우, 이 방법만으로는 반복 적용을 통해서도 좋은 결과를 얻기가 어렵다. 따라서 다음에 기술하는 그래프 이론에 근거한 방법을 고려할 필요가 있게 된다.

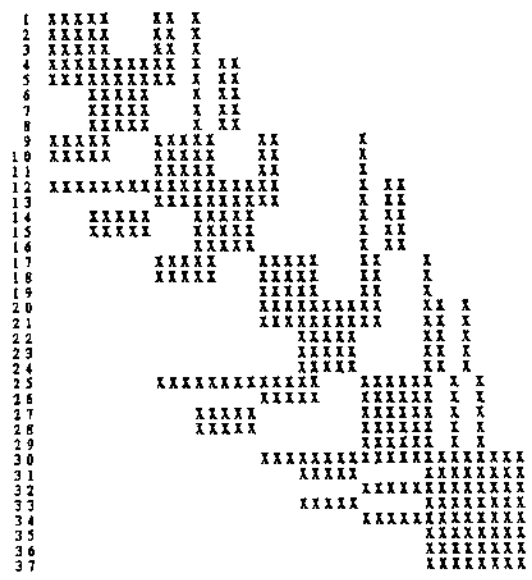
2.3 번호 재부여를 위한 그래프 이론

그래프 이론은 절점 간의 실제 거리보다는 연결 상태에 보다 관심을 두는 이론이며 절점 번호 부여시 절점 간의 형태학적 거리 측정에 유효한 이론으로 일괄적인 번호 재부여 방법, 대칭 행렬의 구조를 나타내는 방법으로 널리 이용되고 있다. 그래프 G는 꼭지점들의 집합 V와 각 꼭지점들 사이의 연결을 나타내는 모서리들의 집합 E로 정의되는 기하학적 실체이다.

행렬의 구조는 자유도를 꼭지점, 0아닌 값을 모서리로 형상화하면 그래프로 나타내어진다. 대칭행렬 [K]에서 $K_{ij} \neq 0$ 이면서 $i \neq j$ 일 때 꼭지점 v_i 와 꼭지점 v_j 는 모서리 (v_i, v_j) 로 연결되어 있다고 정의한다. 연결 행렬과 이를 꼭지점, 모서리로 표



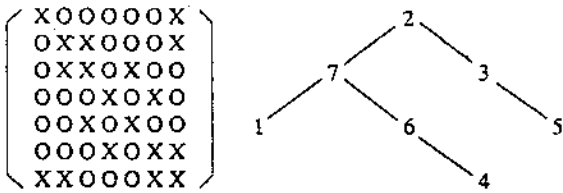
(a) 초기 절점 번호에 대한 절점 연결 행렬.



(b) 압축 기법 수행 후의 절점 연결 행렬.

(x 표시는 해당 자유도가 유한 요소를 통해 서로 연계되어 있음을 의미함.)

<그림 2> 절점 연결 행렬의 예. Ex. 1.



(그림 3) 대칭 행렬의 그래프 표현 예.

현한 예를 <그림 3>에 보였다.

꼭지점 v_i 로부터 꼭지점 v_j 로 가는 경로는 그래프 G 가 완전히 연결되어 있을 때 최소한 한개 이상 존재하며 그 경로는 $\{(v_i, v_1), (v_1, v_2), \dots, (v_{s-1}, v_s), (v_s, v_j)\}$ 과 같이 모서리들의 집합으로 나타낼 수 있다. 꼭지점 v_i 로 부터 꼭지점 v_j 로 가는 경로의 최소 모서리 수를 꼭지점 간의 거리로 정의한다. <그림 3>의 예에서 꼭지점 4와 꼭지점 1 사이의 거리는 3이고 꼭지점 4와 꼭지점 5 사이의 거리는 5이다.

또한 꼭지점 u 를 기준 꼭지점으로 할 때, 각 꼭지점들의 수준(level)을 다음과 같이 정의한다. 꼭지점 u 의 수준을 1로 하고 $v_j \neq u$ 인 v_j 의 수준은 u 와 v_j 의 거리에 1 을 더한 것으로 한다. 수준이 k 인 꼭지점들의 집합을 L_k 로 표현하고, L_k 의 집합(class)을 꼭지점 u 를 기준 꼭지점으로 하는 동위구조(level structure), $L_{(u)}$ 라 한다. <그림 3>에서 꼭지점 1을 기준 꼭지점으로 할 때 $L_1 = \{1\}$, $L_2 = \{7\}$, $L_3 = \{2, 6\}$, $L_4 = \{3, 4\}$, $L_5 = \{5\}$ 이며 $L_w = \{L_1, L_2, L_3, L_4, L_5\}$ 가 된다. 이 동위구조를 실제 프로그램에서는 각 꼭지점의 수준을 유일하게 정의하는 것으로 구현할 수 있다.

꼭지점 u 를 기준으로 구성된 동위구조에서 수준이 가장 큰 점들의 집합 중에서 한 점을 기준점으로 선택하고, 이 점을 기준으로 다시 만든 동위구조를 역 동위구조라고 한다.

한편, 동위구조 $L_{(u)}$ 에 나타나는 최대수준을 d 를 동위구조 $L_{(u)}$ 의 깊이라 하고, 수준이 최대인 꼭지점들의 집합을 끝점집합이라고 하며, 동위구조 $L_{(u)}$ 의 폭 w 는 다음과 같이 표현된다.

$$w = \max |L_i|, \quad i = 1, 2, \dots, d \quad (5)$$

어떤 꼭지점이 모서리 하나를 통해 다른 꼭지점과 연결되어 있을 때 두 꼭지점은 인접하고 있다고 표현한다. 꼭지점 u 에 인접하고 있는 꼭지점들의 집합 $adj(u)$ 를 다음과 같이 정의한다.

$$adj(u) = \{ u_i \in V \mid (u, u_i) \in E \} \quad (6)$$

유한 요소 모델에 번호를 부여해 갈 때 이미 번호가 부여된 꼭지점들에 인접하고 있으면서 아직 번호가 부여되지 않은 꼭지점들의 집합을 $adj(u)$ 와 구별하여 인접집합이라고 정의한다. 번호부여 과정을 파도가 진행되어 나가는 것으로 묘사하여 다음에 번호를 부여할 꼭지점을 보다 잘 찾기 위해 다음과 같이 인접집합 요소를 선단집합과 비선단 집합으로 분류한다. 선단집합의 요소는 인접집합의 요소 중에서 번호를 붙였을 때 인접집합의 크기를 증가시키는 요소이며 비선단집합의 요소는 그렇지 않은 요소이다[9]. 이렇게 인접집합의 요소를 나누면 절점에 번호를 부여할 때, 선단집합보다는 비선단집합 요소의 번호를 먼저 부여하게 할 수 있고 이것이 밴드폭이나 프로파일의 감소를 초래한다.

2.4 새로운 알고리즘

새로운 알고리즘은 압축 알고리즘과 그래프 이론에 의한 방법을 복합한 방법이며 알고리즘의 효율적 수행을 위해 약간의 변형을 가하였다. 실제 번호 부여 시에 번호 부여 후보가 여러 개가 되면 이 중 하나를 선택해야 한다. 본 알고리즘에서 이러한 선택 기준은 수치 경험으로부터 추출하였으며 알고리즘 설명 후에 이를 기술하겠다.

준비 단계. 절점들의 연결 관계 행렬의 구성

유한요소 입력 자료로부터 절점 연결행렬을 구성한다.

제 1 단계. 압축 알고리즘을 이용한 번호부여

스텝 1. 전방 압축을 수행한다.

스텝 2. 후방 압축을 수행한다.

스텝 3. 압축알고리즘을 통하여 새롭게 번호가 부여된 절점들의 역순서를 저장한다.

스텝 4. 동위 구조 구성을 위한 기준 절점을 선택한다. 새로운 넘버링의 1번 절점과 끝번

절점 근처에서 연결절점 수가 가장 작은 절점을 1번 절점으로 선택한다.

제 2 단계. 그래프 이론에 기초한 번호부여

스텝 1. 역동위구조의 구성

- 1) 제 1 단계의 과정을 통하여 선택된 1번 절점을 기준으로 동위구조를 구성한다.
- 2) 끝점집합의 절점들 중에서 연결 절점수가 최소인 점을 기준점으로 다시 동위구조를 만든다(역동위구조).
- 3) 역동위구조에서 수준이 i 인 꼭지점의 수준을 (역동위구조의 깊이 - $i + 1$) 로 바꾸어 준다.

스텝 2. 번호부여

- 1) 제 1 단계에서 선택한 1 번에 절점번호 1 번을 부여한다.
- 2) 인접집합을 구한다.
- 3) 인접집합의 요소수가 0이면 번호 부여 과정을 끝낸다.
- 4) 인접집합 요소를 선단 집합과 비선단 집합으로 분류한다.
- 5) 비선단집합의 요소에 번호를 부여한다. 비선단 집합 요소수가 0이 될 때까지 반복한다.
- 6) 선단집합의 한 절점에 번호를 부여하고, 2)로 되돌아 간다.

제 3 단계. 최종 번호의 확정.

초기 번호, 단계 1, 단계 2의 프로파일을 비교하여 최소의 프로파일을 갖는 번호를 택한다.

1번 절점이 정해지고 나면, 1번을 기준으로 1번의 주위에서부터 그래프 이론에 기초하여 번호를 부여해 나기는데, 이 때 절점이 부여되는 방향이 한방향으로 일관성을 갖게 하는 것이 유리하다. 따라서 같은 조건을 가진 여러개의 절점들에 번호를 부여해 나갈 때 마지막 절점의 위치를 미리 파악할 수 있다면 가능한 한 마지막 절점으로부터 먼곳에 있는 절점에 먼저 번호를 부여하는 것이 효과적이다. 이를 위해 본 논문에서는 역동위구조를 구성하고 이를 다음의 선택 기

준에 이용하였다.

스텝 3 에서 비선단 집합에 번호를 부여하는 기준은 다음과 같다.

- 1) 역동위구조가 낮은 절점.
- 2) 연결절점수가 작은 절점.
- 3) 인접집합에 먼저 들어온 절점.

또한, 선단집합에 번호를 부여하는 기준은 다음과 같다.

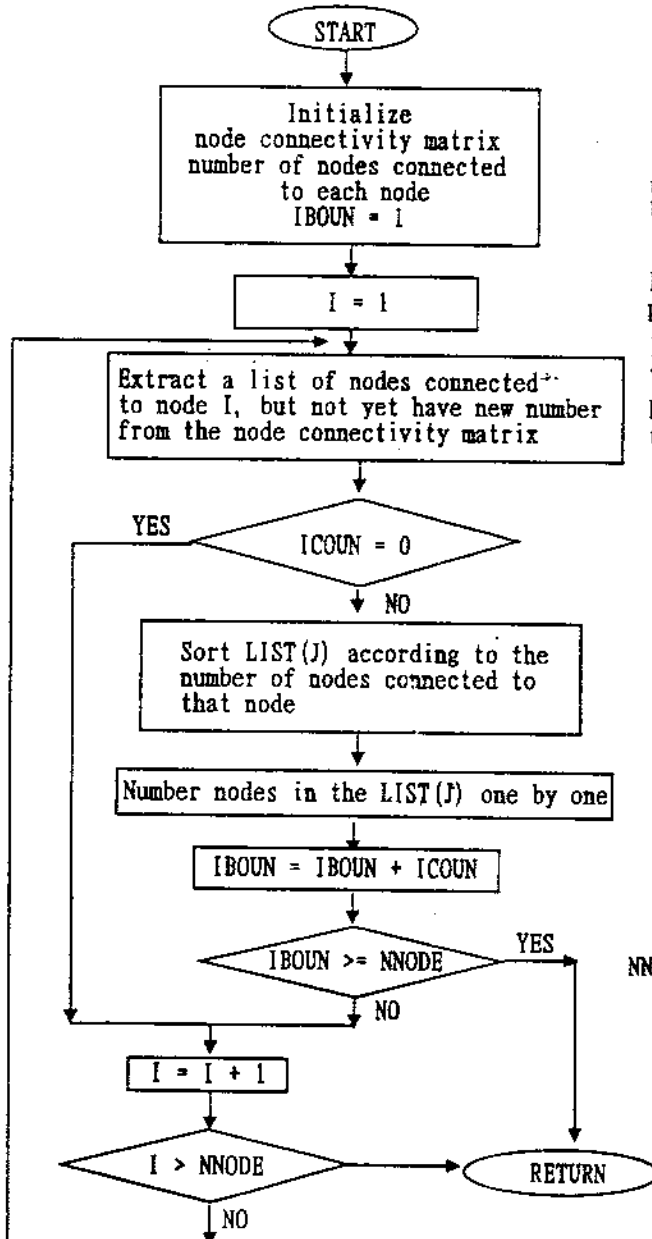
- 1) 인접 집합의 크기를 작게 증가시키는 절점
- 2) 스텝 2에서 구한 압축알고리즘으로 구한 절점 순서가 빠른 절점.

번호 부여 기준은 최종적으로 한 절점이 선정될 때까지 차례로 적용하고 한 점이 결정되지 않을 때에는 최종 후보 절점 중에서 하나를 임의로 선택한다. 전술한 알고리즘의 흐름도를 전방 압축단계, 그래프 이론에 의한 번호 부여 단계, 절점 선택 단계에 대해 각각 <그림 4>, <그림 5>, <그림 6>에 그렸다.

3. 결과 및 고찰

본 논문에서는 18 가지의 유한 요소 모델에 여러가지 번호 부여 알고리즘을 적용하여 알고리즘의 효용성을 비교, 검토하였다. 18 가지의 예제에 대해 표 1에 그 형태 및 초기 조건을 요약하였고, <그림 1>과 <그림 7~23>까지에 각 예제의 유한 요소 모델 격자 모양을 그렸다. 선요소, 삼각형 요소, 사각형 요소, 육면체 요소, 1차요소, 2차요소 등 다양한 유한 요소들이 복합된 모델을 택하였다.

프로파일 결과치는 GPS 방법[8], GK 방법[10], 구본웅의 방법[9]과 비교하였으며 그 결과를 표 2에 나타내었다. GPS 방법은 참고문헌 [13]의 프로그램을 이용하였으며, 구본웅의 방법은 참고문헌 [9]에 제시된 알고리즘을 이용하였다. 프로파일의 결과치를 비교해 볼 때 18개의 예제 중에서 제안된 알고리즘이 11개의 예제에 대하여 가장 우수하고 구본웅의 방법은 5개, GK 방법은 4 개가 가장 우수하다. 구본웅의 알고리즘은 예제 9, 14, 16, 18 번에서, GK 방법은 예제 3, 5, 8, 13, 14, 17, 18 번에서 제시한 알고리즘보다 상당히 좋지 않으며 예제 11, 17 번은 구본웅의 알고



IBOUN is the column number up to which the matrix has been condensed.

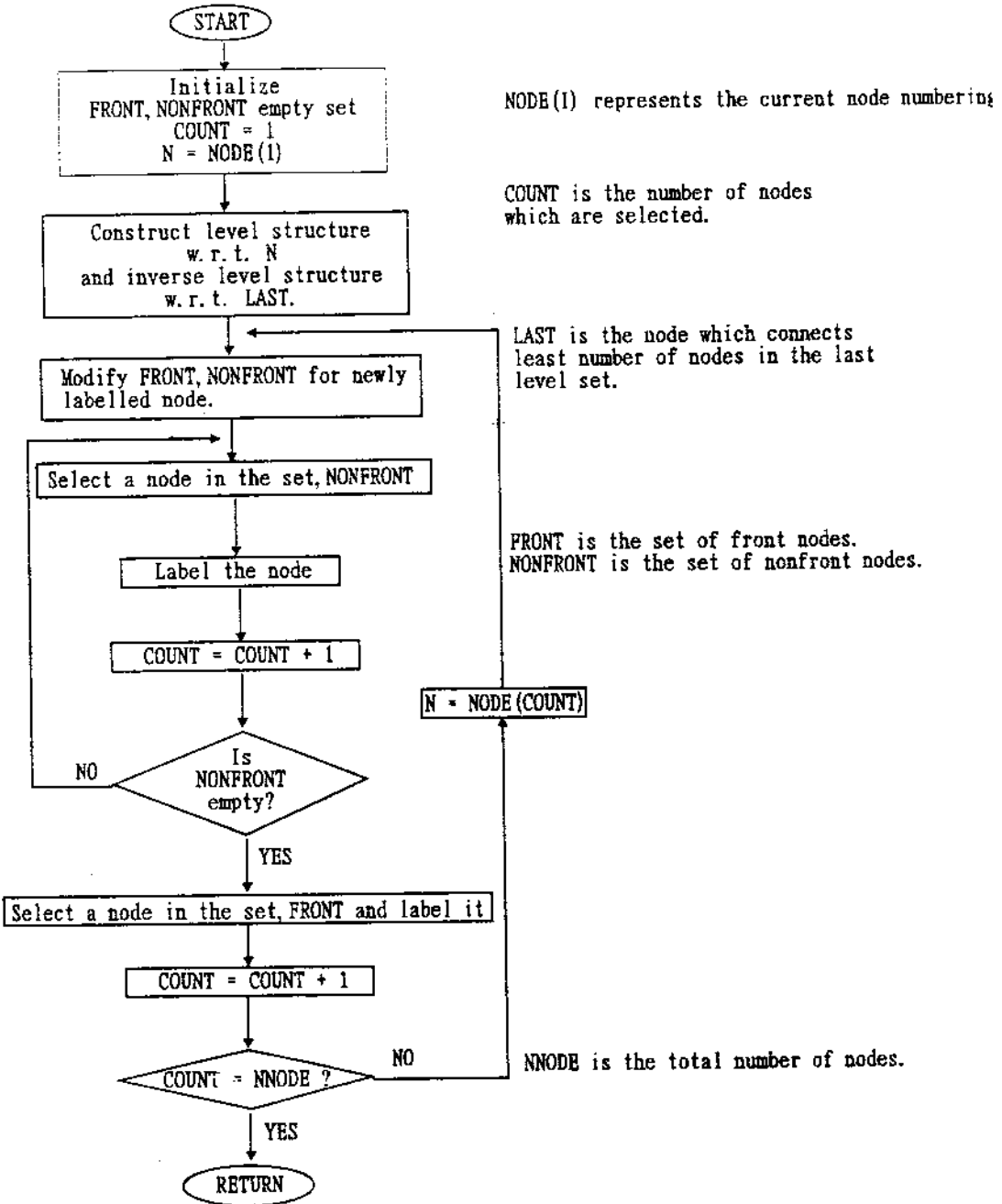
I is the row number to be processed.

ICOUN is the number of nodes to be numbered.

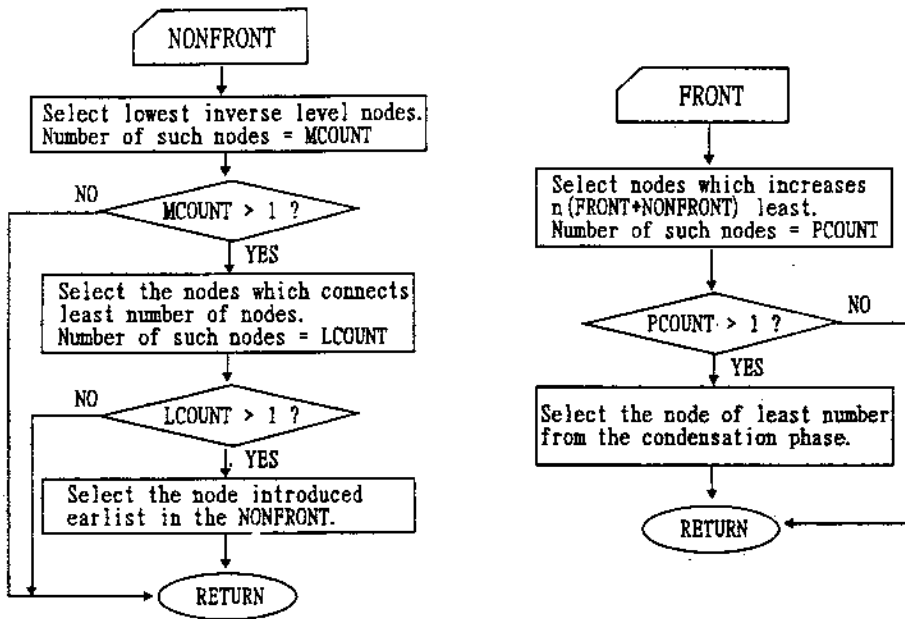
LIST(J) is the list of nodes to be numbered.

NNODE is the total number of nodes.

〈그림 4〉 압축 단계의 흐름도.



〈그림 5〉 웨이브 프론트 개념에 의한 번호부여 단계의 흐름도.



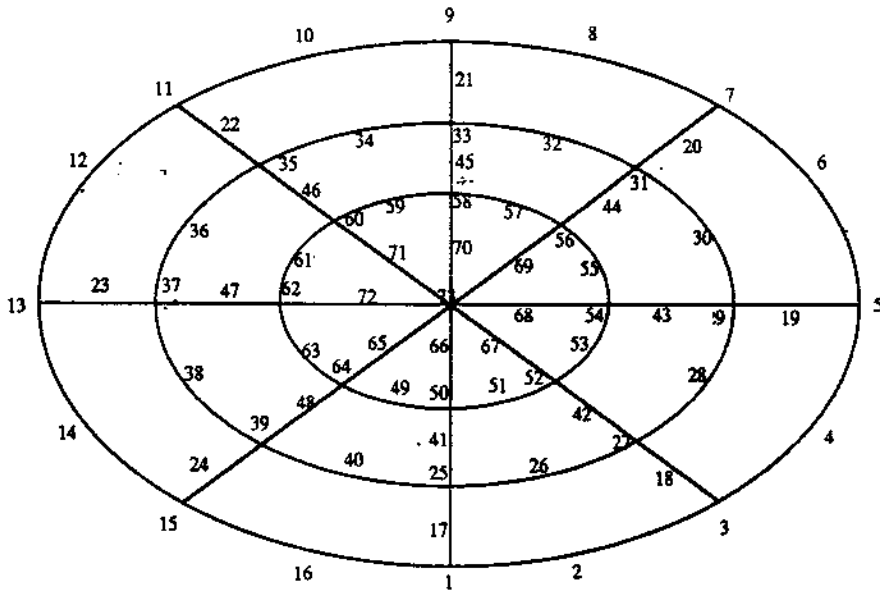
〈그림 6〉 절점 선정 단계의 흐름도.

〈Table 1〉 Test Problem Statistics

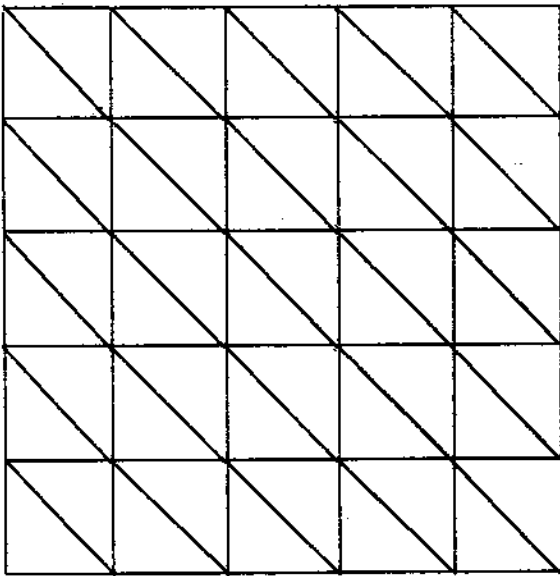
	No. of nodes	No. of elements	Type of element	Matrix density(%)	Initial bandwidth	Initial profile
Ex. 1	37	8	6	31.04	33	449
Ex. 2	73	24	4, 6	18.18	40	1332
Ex. 3	121	50	4	8.41	33	2314
Ex. 4	42	81	1	11.56	38	263
Ex. 5	101	151	3	5.97	37	947
Ex. 6	20	17	1, 5	28.50	17	116
Ex. 7	56	58	1, 5	15.82	53	867
Ex. 8	62	78	1	5.67	49	486
Ex. 9	84	31	7	17.91	65	1404
Ex. 10	96	48	7	18.23	48	2876
Ex. 11	307	334	3, 5	2.69	56	7928
Ex. 12	85	288	1	9.15	50	2059
Ex. 13	350	89	6	3.68	146	5850
Ex. 14	442	168	2, 6	2.22	370	30122
Ex. 15	351	44	8	10.29	152	32915
Ex. 16	490	213	4	2.14	243	30596
Ex. 17	529	186	4, 6	2.48	128	19397
Ex. 18	351	304	5	2.33	292	7308

* Type of elements

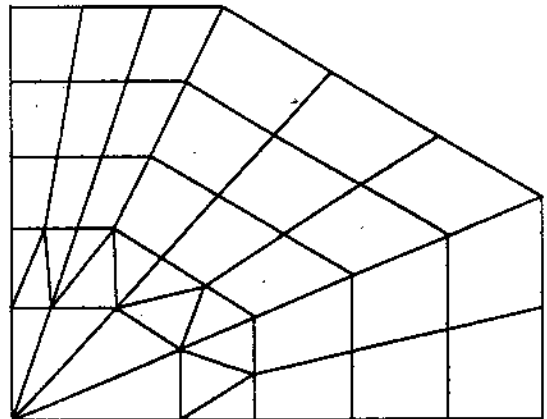
- 1 : Linear Line element (2 nodes/element)
- 2 : Quadratic Line element (3 nodes/element)
- 3 : Linear Triangular element (3 nodes/element)
- 4 : Quadratic Triangular element (6 nodes/element)
- 5 : Linear Quadrilateral element (4 nodes/element)
- 6 : Quadratic Quadrilateral element (8 nodes/element)
- 7 : Linear Hexahedral element (8 nodes/element)
- 8 : Quadratic Hexahedral element (20 nodes/element)



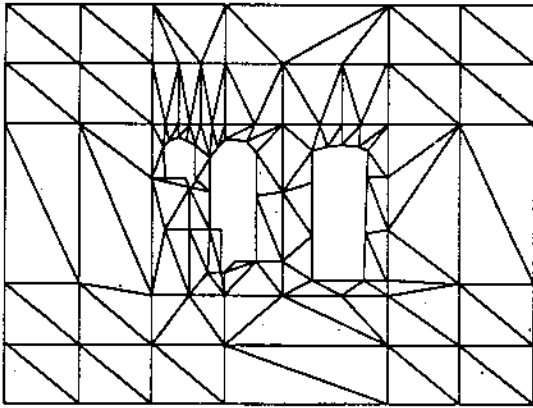
〈그림 7〉 원형판의 유일한 요소 모델. 2차 삼각형 요소 8개와 2차 사각형 요소 16개. Ex. 2.



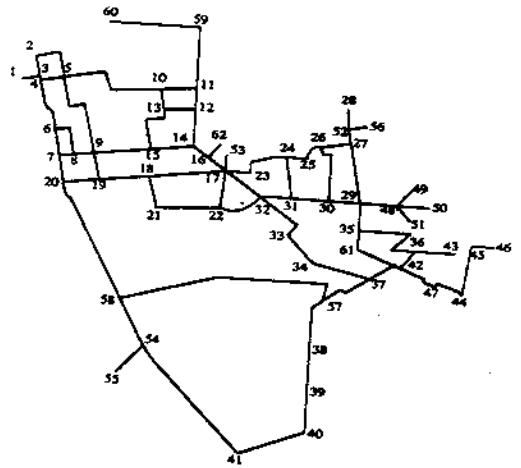
〈그림 8〉 사각형판의 유한 요소 모델.
2차 삼각형 요소 50개. Ex 3.



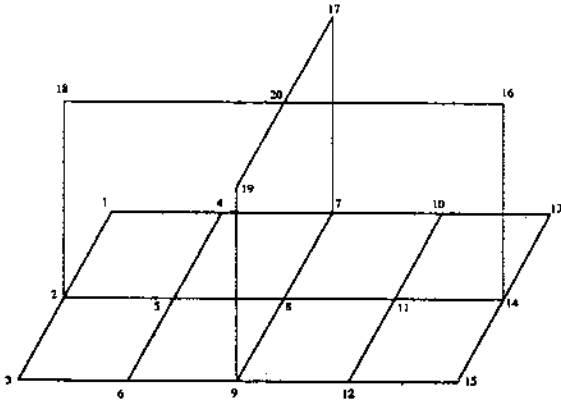
〈그림 9〉 격자 구조의 유한 요소 모델.
선형 선 요소 81개. Ex. 4.



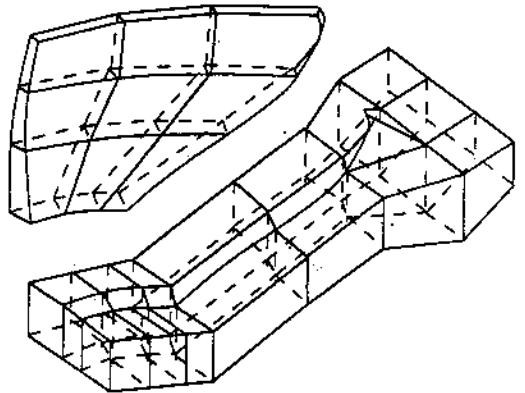
〈그림 10〉 압반 속의 지하 공간의 유한 요소 모델.
선형 삼각형 요소 151개. Ex. 5.



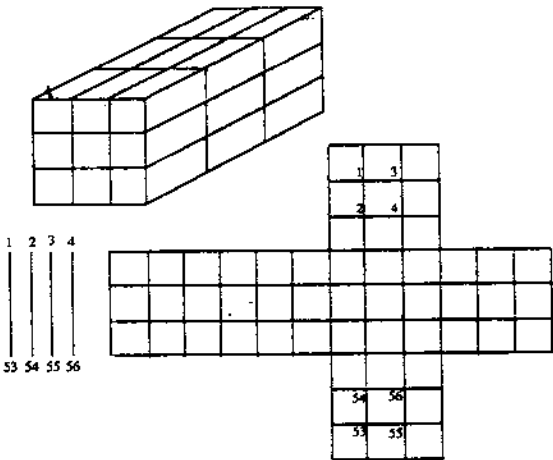
〈그림 13〉 수도관 망의 유한 요소 모델.
선형 선요소 78개. Ex. 8.



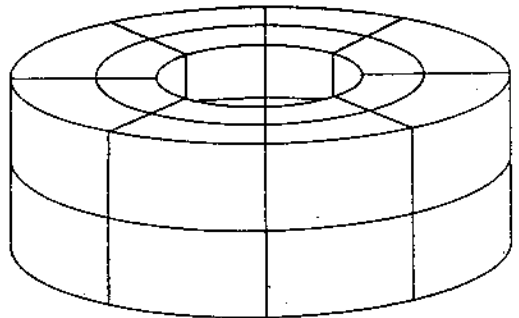
〈그림 11〉 3차원 구조의 유한 요소 모델. 선형 선요소 9개와
선형 사각형 요소 8개. Ex. 6



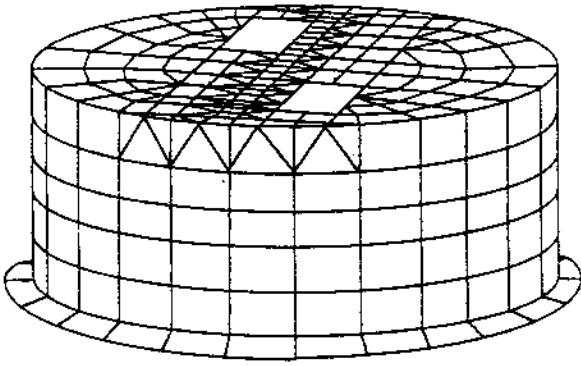
〈그림 14〉 아치형 맴과 기초의 유한요소 모델.
선형 육면체 요소 31개. Ex. 9



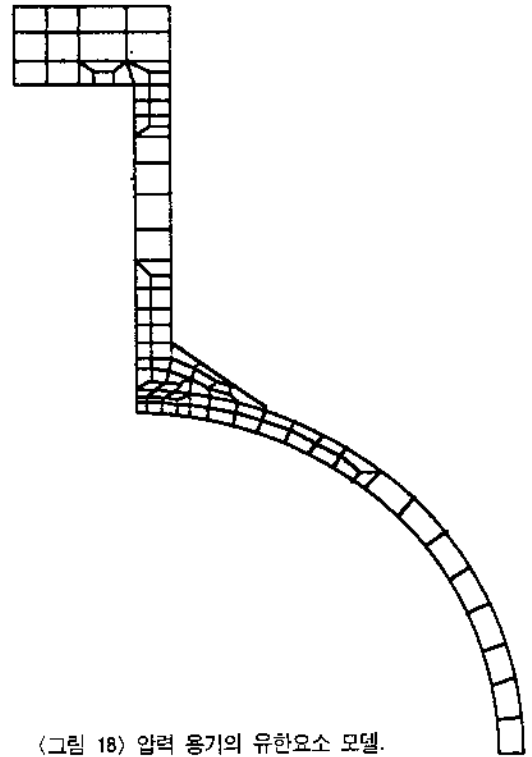
〈그림 12〉 상자 구조의 유한 요소 모델. 선형 선요소 4개와
선형 사각형 요소 54개. Ex. 7.



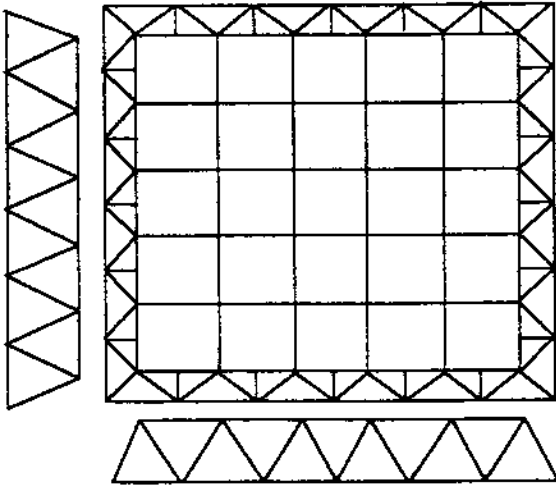
〈그림 15〉 바퀴의 유한요소 모델.
선형 육면체 요소 48개. Ex. 10.



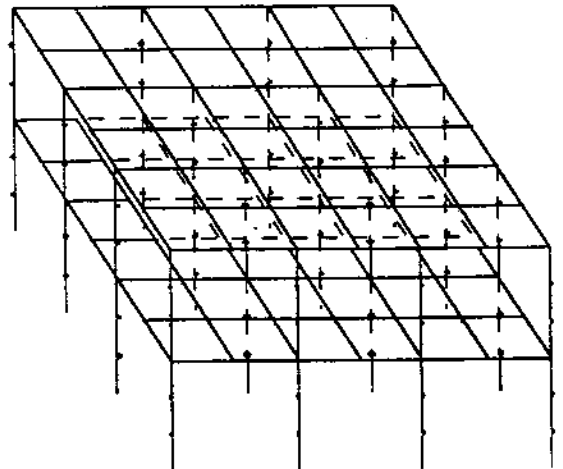
〈그림 16〉 발전소 건물의 유한요소 모델. 선형 삼각형 요소 99개와 선형 사각형 요소 235개. Ex. 11.



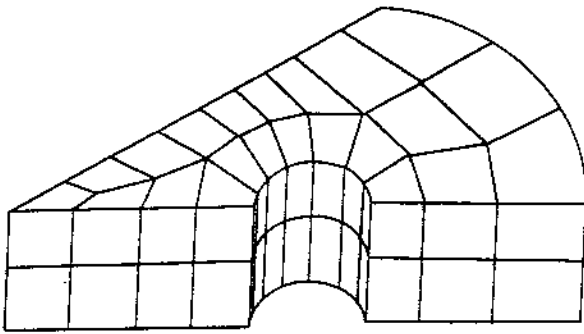
〈그림 18〉 압력 용기의 유한요소 모델. 2차 사각형 축대칭요소 89개. Ex. 13.



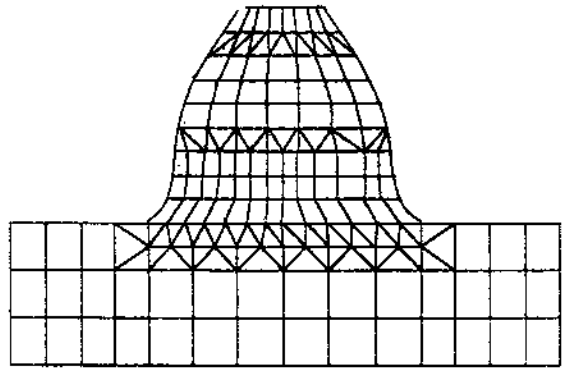
〈그림 17〉 공장 지붕의 유한요소 모델. 선형 선 요소 288개. Ex. 12.



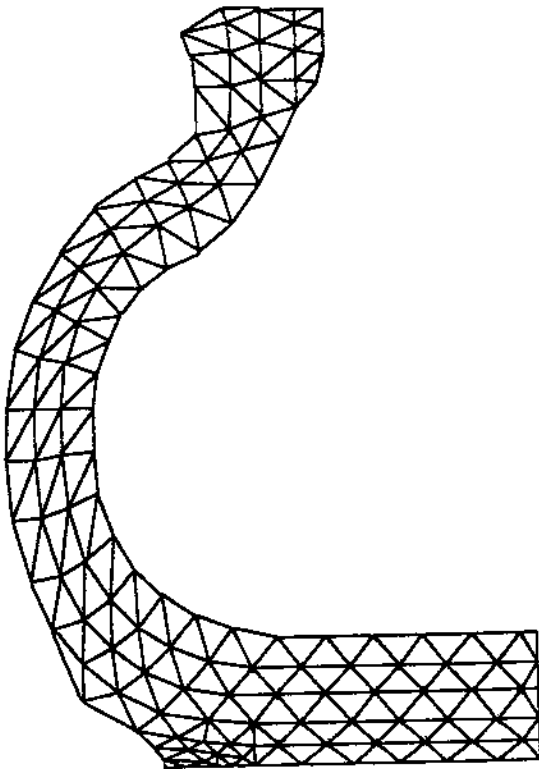
〈그림 19〉 2층 건물의 유한요소 모델. 2차 선요소 96개와 2차 사각형 요소 72개. Ex. 14.



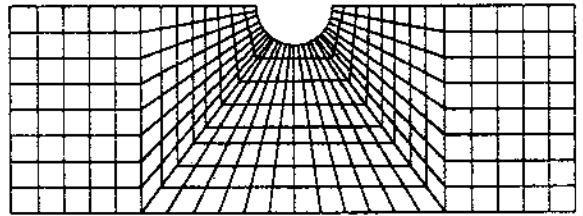
〈그림 20〉 구멍이 있는 원형판의 유한요소 모델.
2차 육면체 요소 44개. Ex. 15.



〈그림 22〉 치차 일부의 유한요소 모델. 2차 삼각형 요소
84개와 선형 사각형 요소 102개. Ex. 17.



〈그림 21〉 타이어 단면의 유한요소 모델.
2차 삼각형 요소 213개. Ex. 16.



〈그림 23〉 노치가 있는 판의 유한요소 모델.
선형 사각형 요소 304개. Ex. 18.

리즘이 상당히 더 우수하다. GPS 방법의 결과는 전반적으로 본 방법에 비해 10 % 이상 차이가 난다.

번호 부여에 소요되는 시간을 표 3에 수록하였다. 수록한 시간은 66MHz 의 IBM PC 486 컴퓨터의 터보 기능을 끈 상태에서 절점 연결 행렬이 구성된 이후부터 알고리즘의 수행이 완료되는 시각까지의 CPU 처리 시간이다. 번호부여에 소요되는 시간은 전체적으로 구분용의 방법과 비슷하거나 약간 더 걸리는 것으로 나타났다. 전반적으로 GPS 방법과는 비슷하고 GK 방법보다는 빠르다.

계산결과, 계산시간, 프로그램의 난이도 등을 고려해 볼 때, 새로 제안한 방법이 기존의 방법들에 비해 우수하며 프로그램을 효율적으로 작성하면 번호 부여 소요시간을 보다 줄일 수 있을 것이다.

4. 결 론

유한 요소 해석시에 구성되는 강성행렬의 프로파일

(Table 2) Final Profiles of the Test Problems

	GPS[8]	GK[10]	KBU[9]	Present
Ex. 1	286	280	*271	*271
Ex. 2	1256	1043	*1005	*1005
Ex. 3	1761	1365	1232	*1220
Ex. 4	248	245	235	*233
Ex. 5	1207	1144	902	*886
Ex. 6	105	*97	*97	101
Ex. 7	716	*679	681	*679
Ex. 8	352	316	*277	291
Ex. 9	1280	1136	1166	*1088
Ex. 10	2442	*2217	2308	2308
Ex. 11	7970	7502	*7307	7659
Ex. 12	968	959	919	*905
Ex. 13	4400	4795	3729	*3652
Ex. 14	16653	12462	10018	*9390
Ex. 15	11818	*10904	11033	10983
Ex. 16	7588	*5800	6572	5810
Ex. 17	15785	19748	*13367	14428
Ex. 18	5149	5121	5046	*4407

*: means the best of the four

(Table 3) CPU Time Required for the Node Numbering (sec.)

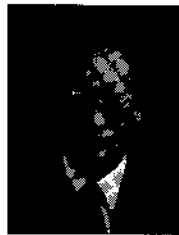
	GPS[8]	GK[10]	KBU[9]	Present
Ex. 1	0.12	0.14	0.06	0.05
Ex. 2	0.26	0.71	0.11	0.10
Ex. 3	0.35	0.74	0.16	0.23
Ex. 4	0.08	0.07	0.11	0.08
Ex. 5	0.19	0.41	0.22	0.28
Ex. 6	0.04	0.03	0.06	0.05
Ex. 7	0.15	0.23	0.11	0.15
Ex. 8	0.09	0.07	0.11	0.10
Ex. 9	0.45	0.78	0.16	0.15
Ex. 10	0.69	1.67	0.28	0.33
Ex. 11	0.70	3.35	1.32	2.71
Ex. 12	0.15	0.30	0.39	0.30
Ex. 13	1.30	3.20	0.66	0.94
Ex. 14	1.60	25.30	1.22	2.07
Ex. 15	8.73	31.24	2.41	1.14
Ex. 16	1.64	6.25	1.44	2.30
Ex. 17	2.42	23.38	1.59	3.02
Ex. 18	0.93	1.83	1.37	2.54

을 감소시키는 새로운 알고리즘을 개발하였다. 제안된 방법은 절점 연결 행렬을 대각선 방향으로 압축시키는 Puttonen의 방법과 그래프 이론에 근거한 일괄적 번호부여 방법을 차례로 적용하여 시스템 행렬의 프로파일을 줄이는 방법이다.

수치예제를 통해 제안된 방법이 기존의 방법들에 비해 프로파일 감소면에서 더 우수함을 보였다. 제안된 알고리즘은 알고리즘의 간단 명료성으로 인하여 프로그램 작성이 용이하고 실행 시간의 면에서도 비교적 우수하다. 이 방법은 요소의 종류에 관계 없이 다양한 유한 요소 모델에 적용이 가능하나, 유한 요소 모델의 각 절점 자유도 수가 상이한 경우에는 이를 고려할 수 있는 알고리즘의 개발이 필요하겠다.

[참고문헌]

- [1] 이병채, "유한요소 행렬의 밴드폭 감소를 위한 절점 번호 부여 알고리즘 개발," 한국항공우주학회지, 제 17권, 1989, pp. 39-45.
- [2] F. A. Akyuz and S. Utku, "An automatic node relabelling scheme for bandwidth minimization of stiffness matrices," AIAA Journal, Vol. 6, 1968, pp. 728-730.
- [3] Henry R. Grooms, "Algorithm for matrix bandwidth reduction," ASCE, J. Structural Division, Vol. 98, 1972, pp. 203-214.
- [4] R. J. Collins, "Bandwidth reduction by automatic renumbering," Int. J. Numer. Meth. Engng, Vol. 6, 1973, pp. 345-356.
- [5] G. Akhras and G. Dhatt, "An automatic node relabelling scheme for minimizing a matrix or network bandwidth," Int. J. Numer. Meth. Engng, Vol.10, 1976, pp. 787-797.
- [6] J. Puttonen, "Simple and effective bandwidth reduction algorithm", Int. J. Numer. Meth. Engng, Vol. 19, 1983, pp. 1139-1152.
- [7] W. H. Liu and A.H.Sherman, "Comparative analysis of the Cuthill-Mckee and the reverse Cuthill-Mckee ordering algorithm for sparse matrices," SIAM J. Numer. Anal., Vol. 13, 1976, pp. 198-213.
- [8] N. E. Gibbs, W. G. Poole, Jr. and P. K. Stockmeyer, "An algorithm for reducing the bandwidth and profile of a sparse matrix," SIAM J. Numer. Anal., Vol. 13, 1970, pp. 236-250.
- [9] B. U. Koo and B. C. Lee, "An efficient profile reduction algorithm based on the frontal ordering scheme and the graph theory," Computers and Structures, Vol. 44, No.6, 1992, pp. 1339-1347.
- [10] I. P. King, "An automatic reordering scheme for simultaneous equations derived from network analysis," Int. J. Numer. Meth. Engng, Vol. 2, 1970, pp. 523-533.
- [11] N. E. Gibbs, "Algorithm 509 : A hybrid profile reduction algorithm," Collected algorithm from ACM, ACM, New York, 1983.
- [12] Collected Algorithm from ACM, Vol.III, Algorithm 508, A publication of the Association for Computing Machinery Inc., New York, 1983.
- [13] G. C. Everstine, "A comparison of three resequencing algorithm for the reduction of matrix profile and wavefront," Int. J. Numer. Meth. Engng, Vol. 14, 1979, pp. 837-853.



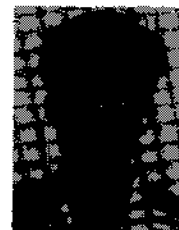
이병채

1977년 2월 서울공대 기계공학과 졸업(학사)

1984년 2월 한국과학기술원 졸업

1984년 6월~1988년 3월 한국과학기술 연구원 선임연구원

1988년 4월~현재 한국과학기술원 기계공학과 조교수, 부교수



김형민

1995년 2월 한국과학기술원 기계공학과 졸업(학사)