

(論 文)

이단계 번호 부여 방법과 선단집합 이용방법을 결합한 밴드폭 감소 알고리즘 개발

이 병 채* · 구 본 응**

(1990년 8월 24일 접수)

Development of a Bandwidth Reduction Algorithm by Combining the Two-Step Approach and the Frontal Ordering Scheme

Byung Chai Lee and Bon Ung Koo

Key Words : Finite Element Analysis(유한요소해석), Bandwidth Reduction(밴드폭 감소), Two-step Approach(이단계 번호 부여), Frontal Ordering Scheme(선단집합 이용 방법), Graph Theory(그래프 이론)

Abstract

A new bandwidth reduction algorithm is developed by combining the two-step approach and the frontal ordering scheme. In the two-step approach, finite elements are numbered first, followed by nodal numbering based on the graph theory. The concept of wave front is incorporated into it to control the cardinality of the set of adjacent nodes and the bandwidth to be achieved. They are controlled systematically by rational selection of next candidates with the purpose of getting the smaller bandwidth efficiently. Eighteen meshes are renumbered and the results are compared with those of well-known algorithms. The results demonstrate the efficiency and the reliability of the proposed algorithm.

기 호 설 명

- A : 인접집합, 현재 번호가 부여된 꼭지점에 인접한 번호가 부여되지 않은 꼭지점들의 집합
- A_k : 요소 k 의 절점에 처음으로 번호가 붙어질때 새롭게 인접집합에 속한 절점들의 집합
- b_k : 요소 k 의 절점에 처음으로 번호가 붙어질때 인접집합의 크기
- K : 유한 요소법의 구조 강성 행렬
- $K^{(i)}$: i 번째 유한 요소의 구조 전체 자유도로 표현한 강성 행렬
- $L(u)$: 그래프에서 꼭지점 U 로부터 위상학적 거리가 같은 꼭지점들의 집합, 동위구조

- n_e : 유한 요소의 총 수
- v : 꼭지점
- w : 동위구조의 폭
- β : 강성행렬의 밴드폭, 대각선을 포함한 반폭

1. 서 론

컴퓨터의 발달과 함께 공학문제 해석에도 컴퓨터가 활발히 이용되고 있다. 컴퓨터를 이용한 공학문제 해석방법으로 가장 일반적으로 이용되는 방법은 유한요소법이다. 유한요소법은 해석하고자하는 영역을 적절한 부영역으로 분할하고 형상함수를 도입하여 독립변수를 절점변수로 근사함으로써 풀고자하는 미분방정식의 해를 구하는 방법이다. 영역 분

*정회원, 한국과학기술원 기계공학과

**한국과학기술원 기계공학과

할과 형상함수의 도입에 의해 미분방정식은 강성행렬과 하중벡터로 구성된 대수방정식으로 근사되고 이들 강성행렬과 하중벡터로부터 절점변수를 구하게 된다.

강성행렬은 일반적으로 대칭이며 한 유한요소 내의 독립변수값은 그 요소를 이루고있는 절점에서의 변수값 만으로 근사되기 때문에 요소를 통해 서로 연결되지 않은 자유도에 해당되는 강성행렬항은 영이된다. 따라서 전 구조의 강성행렬은 대부분의 항이 영인 행렬이 된다. 강성행렬에서 영이 아닌 항이 대각선 부근에 집중되게하여 영이 아닌 경계까지만 저장, 이용하면 컴퓨터의 기억용량을 효율적으로 이용할 수 있으며 계산량과 계산오차도 줄어진다. 이때 행렬의 영이 아닌 항들이 모두 포함되는 대각선으로 부터의 거리를 밴드폭이라 하며 연결방정식을 푸는 시간은 대체적으로 총자유도에 비례하고 밴드폭의 제곱에 비례하게 된다. 특히 최적 설계 또는 비선형 문제의 경우와 같이 반복해서 해를 구해야하는 경우에는 계산시간과 계산오차의 관점에서 밴드폭을 줄이는 과정이 거의 필수적이라 할 수 있다.

유한요소법을 적용할 때 강성행렬의 밴드폭은 근본적으로 유한요소의 절점에 번호를 부여하는 방법에 따라 크게 차이가난다. 절점에 효율적으로 번호를 부여하는 방법은 1960년대 이후 본격적으로 개발되기 시작하였으며⁽¹⁻⁸⁾ 대표적인 방법으로는 Gibbs-Poole-Stockmeyer(GPS)의 방법⁽³⁾, Puttonen의 방법⁽⁴⁾, 이단계 번호부여 방법⁽⁵⁾등이 있다.

GPS방법과 이단계 번호부여 방법은 그래프 이론에 기초하여 시작점으로부터 위상학적 거리가 가까운 절점부터 차례로 번호를 부여하는 방법으로 초기에는 밴드폭이 작게 번호가 부여되지만 진행해감에 따라 차츰 밴드폭이 커지는 단점이 있으며 GPS방법은 시작점 선정과정이 복잡하고 이단계 번호 부여 방법은 결과가 좋지않다. Puttonen의 방법은 강성행렬의 대각선 쪽에 각 항이 차곡차곡 집중되도록 절점에 번호를 부여하는 방법으로 초기 번호의 영향이 크고 번호부여 시간도 오래걸리고 결과도 별로 좋지않다.

본 논문에서는 그래프 이론에 기초한 이단계 번호 부여 방법에 목표밴드폭을 도입하여 계산시간이 짧으면서도 효율적으로 번호를 부여할 수 있는 방법을 제안하고자 한다.

2. 그래프 이론

2.1 강성행렬의 구성

유한요소해석에서 유한요소의 갯수를 n_e 라고 하면 강성행렬 $[K]$ 는 다음과 같이 구성된다.

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

여기서, $[K^{(i)}]$ 는 구조 전체 자유도로 표현된 i 번째 요소의 강성행렬이며 j 번째 자유도와 k 번째 자유도에 해당하는 절점들이 i 번째 요소에 존재하지 않으면 $K_{jk}^{(i)}=0$ 이 된다. 이와같이 구해진 강성행렬의 밴드폭 β 는 다음과 같이 정의된다.

$$\beta = \max_{i,j \in V} |i-j|+1, K_{ij} \neq 0 \quad (2)$$

식 (1)과 식 (2)에서 각 자유도가 유한요소를 통해서만 연결되는 것을 고려하면 밴드폭은 하나의 유한요소에서 나타나는 자유도 번호의 최대 차이에 1을 더한 값이 된다. 따라서 각각의 유한요소 내에 나타나는 최대 절점번호와 최소 절점번호의 차이가 작게 나타나도록 하면 밴드폭도 작아지게 된다.

2.2 그래프 이론

대칭행렬의 구조를 나타내는 방법으로 그래프 이론이 널리 이용되고 있으며 그래프는 꼭지점(V)과 모서리(E)집합으로 구성되고 꼭지점 간의 위상학적 연결상태에만 관심이 있을 때 유용하게 이용된다^(2,3,8). 행렬의 그래프 표현은 각 자유도를 꼭지점으로 일대일 대응시켰을 때 $K_{ij} \neq 0$ 이면서 $i \neq j$ 일때 꼭지점 v_i 와 꼭지점 v_j 를 모서리 (v_i, v_j) 로 연결하여 나타낼 수 있다. 이때 그래프의 꼭지점 수는 행렬의 행 또는 열의 수와 같아지고 모서리 수는 행렬의 상부 삼각 행렬의 영이 아닌 항의 수와 같게 된다.

꼭지점 간의 위상학적 거리를 표현하기 위해 경로를 정의한다. 꼭지점 v_i 로 부터 꼭지점 v_j 로 가는 경로는 그래프 G 가 완전히 연결되어있을 때 최소한 한개 이상 존재하며 그 경로는 $\{(v_i, v_1), (v_1, v_2), \dots, (v_s, v_j)\}$ 로 나타낼 수 있다. 꼭지점 사이의 거리는 두 꼭지점을 잇는 경로 중 가장 작은 수의 모서리를 갖는 경로의 모서리 수로 한다. 이와 같이 꼭지점 간의 거리를 정의하면 어떤 꼭지점을 기준으로 모든 꼭지점의 거리를 구할 수 있고 이때 이 값에 1을 더한 값을 꼭지점의 수준(level)이라고 표현한다.

어떤 꼭지점으로 부터 수준이 k 인 꼭지점들의 집합을 L_k 라고 할때 L_k 의 집합을 u 를 기준 꼭지점으로 하는 동위구조(level structure), $L_{(u)}$ 라고 하며 이 구조를 이용하여 가까운 점부터 조직적으로 번호를 부여할 수 있다. 동위구조 $L_{(u)}$ 의 최대 수준을 d 라고 할때 동위구조 $L_{(u)}$ 의 깊이는 d 이고 수준이 최대인 꼭지점들의 집합을 끝점집합이라고 하며 동위구조 $L_{(u)}$ 의 폭 w 와 꼭지점 v 에 인접한 꼭지점들의 집합, $adj(v)$,은 다음과 같이 정의된다.

$$w = \max_{i=1,2,\dots,d} |L_i| \quad (3)$$

$$adj(v) = \{v_i \in V | (v, v_i) \in E\} \quad (4)$$

동위구조의 폭과 깊이는 번호 부여 후의 밴드폭 크기와 밀접한 관계를 가지고있으며 폭이 작고 깊이가 큰 동위구조 일수록 보다 작은 밴드폭을 갖게 번호를 부여할 수 있다⁽³⁾. 식 (3)에서 $|L_i|$ 는 집합 L_i 에 속한 꼭지점의 갯수를 나타낸다.

한편, 유한 요소 모델의 번호 부여 과정을 고찰해보면 절점을 임의로 선택하여 번호를 붙이는 것보다 한쪽에서 부터 차례로 붙이는 것이 밴드폭의 관점에서 나은 것을 알 수 있다. 이를 조직적으로 이용하기 위해 다음과 같이 번호 부여 진행에 따라 변하는 몇개의 집합을 정의한다. 번호를 부여함에 따라 이미 번호가 부여된 꼭지점에 인접하고 있으면서 아직 번호가 부여되지 않은 꼭지점이 존재하게 되는데 이 꼭지점들의 집합을 인접집합이라고 정의하며 인접집합 중에서 이미 번호가 부여된 꼭지점이나 인접집합에 속해있는 꼭지점에만 인접하고있는 꼭지점들의 집합을 비선단집합이라고 하고 인접집합에서 비선단집합을 뺀 집합을 선단집합이라고 한다. 번호를 부여하는 동안 인접집합의 요소 수는 어떤 꼭지점에 번호를 붙이는 가에 따라 증가하기도 하고 감소하기도 하며, 이때 나타나는 최대 인접집합의 요소 수는 밴드폭에 큰 영향을 미치며 실제로 결과 밴드폭은 최대 인접집합의 요소 수보다 작을 수 없게된다. 따라서, 목표 밴드폭을 설정하여 인접집합의 요소 수를 잘 제어하면 좋은 결과를 얻을 수 있다⁽²⁾.

3. 번호부여 과정

여기서 제시하는 알고리즘은 크게 두개의 과정으로 나누어지며 제 1단계는 유한요소의 번호를 새롭게

부여하는 과정이며 제 2단계는 제 1단계에서 얻어진 정보를 이용하여 절점번호를 부여하는 과정이다.

3.1 제 1단계-유한요소의 번호 부여

제 1단계에서는 각 유한요소들 사이의 연결관계 행렬을 구성하고 연결관계행렬을 이용하여 동위구조를 구성한 다음 Cuthill-McKee 알고리즘을 이용하여 각각의 요소에 새롭게 번호를 부여한다. 이와 같이 유한요소의 번호를 새롭게 부여하면 시작점의 결정과 동위구조 구성이 용이해지고 절점번호 부여를 위한 기준으로도 응용할 수 있게된다.

(1) 유한요소들의 연결관계행렬구성

유한요소들의 연결상태를 나타내는 방법으로는 요소 e_i 와 요소 e_j 가 한개 이상의 절점을 공유하면 e_i 와 e_j 가 서로 연결되어 있다고 할 수 있다. 그러나 이렇게 구한 연결관계행렬은 유한요소들 사이의 연결관계가 매우 복잡하게 나타나므로 다음과 같이 연결상태를 완화한다⁽⁵⁾.

요소 e_i 는 n_i 차원이고 e_j 는 n_j 차원 요소일때 e_i 와 e_j 는 $\{\min(n_i, n_j) - 1\}$ 차원 경계를 공유할 때 서로 연결되어 있다고 한다. 예를들어 e_i 와 e_j 가 모두 3차원 요소일때 서로 공유하는 경계면이 있으면 서로 연결 되었다고 하며 모두 2차원 요소일 때에는 공유하는 경계선이 있을때 서로 연결 되었다고 한다.

(2) 유한요소들에 대한 동위구조 구성

유한요소들의 연결관계행렬을 이용하여 유한요소들에 대한 동위구조를 구성한다. 동위구조는 폭이 작은 것일수록 결과가 양호하게 나타나며 동위구조의 폭을 작게하기 위하여 다음과 같은 과정을 거친다. 유한 요소의 동위구조에서는 각 유한 요소가 꼭지점이 된다.

- (i) 연결꼭지점 수가 가장 적은 꼭지점을 찾는다.
- (ii) (i)에서 구한 꼭지점들을 기준꼭지점으로 하여 동위구조를 구성한 다음 동위구조의 깊이가 가장 깊으면서 동위구조의 폭을 가장 작게 만드는 꼭지점을 찾아 u 라고 한다.

(iii) 꼭지점 u 를 기준꼭지점으로 하여 동위구조 $L_{(u)}$ 를 구성한다.

(iv) 동위구조 $L_{(u)}$ 의 끝점집합을 구하고 끝점집합을 연결꼭지점 수가 적은 꼭지점부터 차례로 재배열 한다.

(v) 끝점집합 각각의 꼭지점을 기준으로 동위구조 $L_{(v)}$ 를 구성한다. 이때 동위구조 $L_{(v)}$ 의 깊이가 동위구조 $L_{(w)}$ 의 깊이보다 깊으면 v 를 w 로 대체시킨 다음 (iii)으로 진행한다.

(3) Cuthill-McKee 알고리즘

유한요소들에 대한 동위구조를 이용하여 각각의 요소에 새롭게 번호를 부여한다. 이 과정에서는 비교적 간단한 방법으로 Cuthill-McKee 알고리즘을 이용하여 각각의 요소에 번호를 부여한다. Cuthill-McKee 알고리즘은 수준이 낮은 꼭지점부터 차례로 번호를 부여하는 방법으로 다음과 같은 과정으로 진행된다⁽⁷⁾.

(i) (2)에서 구한 꼭지점 u 에 번호 1을 부여한다. 수준 $i=1$

(ii) 수준 i 에 속한 꼭지점들에 번호를 부여한다. 가장 낮은 번호가 부여된 꼭지점에 인접한 꼭지점 부터 번호를 부여하며 후보가 여러점 일때는 연결 꼭지점 수가 작은 꼭지점에 먼저 번호를 부여한다.

(iii) 아직 번호를 부여할 꼭지점이 남아있으면 수준 $i=i+1$ 로 하고 (ii)로 진행한다.

요소의 번호를 부여하는 과정은 요소들로 구성된 그래프가 절점들의 그래프보다는 훨씬 단순하기 때문에 별로 복잡하지 않고 컴퓨터 처리시간도 짧다.

3.2 제 2단계-절점의 번호 부여

절점에 대한 번호부여과정에서는 새롭게 부여된 요소번호, 절점들에 대한 동위구조와 적절하게 설정된 목표밴드폭을 이용하여 각 절점에 번호를 부여한다.

(1) 절점들에 대한 동위구조 구성

절점에 대한 동위구조는 깊이가 깊으면서 폭이 작은것이 유리하나 여기에서는 좋은 동위구조 탐색을 위한 시간을 줄이기 위해 이단계 방법을 이용하였고 따라서, 전술한 방법에 따라 새롭게 구한 요소번호를 이용하여 다음과 같이 간단하게 동위구조를 구성한다⁽⁸⁾.

(i) 요소번호 1번에 속한 절점 중에서 연결꼭지점 수가 가장 적은 절점을 기준꼭지점으로 하여 동위구조를 구성한다.

(ii) 끝점집합의 각 꼭지점을 기준꼭지점으로 하여 동위구조를 구성하고 동위구조의 폭이 가장 작은 것을 선택한다.

(iii) 수준이 i 인 꼭지점들의 수준을 (동위구조의

깊이 $-i+1$)로 바꾸어준다.

(2) 목표밴드폭의 설정

목표밴드폭은 번호부여 과정중 번호를 부여할 절점을 선택하는데 이용된다. 원칙적으로 번호는 인접집합에 먼저 들어온 절점에 먼저 부여되지만 다른 절점에 번호를 먼저 붙여도 밴드폭이 목표밴드폭을 초과하지 않는 경우 나중에 들어온 절점 또는 인접집합이 아닌 절점에 먼저 번호를 부여하여 최종 밴드폭을 보다 줄일 수 있도록 하였다.

목표밴드폭은 수정이 가능하며 번호를 부여하는 과정에서 인접집합의 크기가 목표밴드폭보다 커지게되면 최종 밴드폭은 목표밴드폭보다 커지게되므로 목표밴드폭은 이때의 인접집합의 크기와 같게 수정된다. 한편, 절점에 번호를 붙이기 시작할 때의 목표밴드폭이 너무 크게 설정되면 그만큼 최종 밴드폭도 크게 나타나며 너무 작게 설정되면 반복 과정이 많아지게 된다. 따라서 적절한 초기목표밴드폭이 필요하게되며 다음과 같이 초기목표밴드폭을 설정하게 된다.

(i) 절점에 대한 그래프에서 연결꼭지점의 수가 가장 적은 꼭지점의 연결꼭지점수를 m_1 , 연결꼭지점수가 가장 많은 꼭지점의 연결꼭지점수를 m_2 라고 하면 초기목표밴드폭은 식 (5)와 같이 정한다.

$$band_{initial} = \max\left(m_1, \frac{m_2}{2}\right) \quad (5)$$

(ii) 초기목표밴드폭을 이용하여 절점에 번호를 부여하고 최종 밴드폭 $band_{final}$ 를 얻는다.

(iii) $band_{min} = band_{final}$ 로 한다.

(iv) 초기목표밴드폭을 수정한다.

$$band_{initial} = \frac{band_{min} + band_{initial}}{2} \quad (6)$$

만약, 식 (6)으로 구한 초기목표밴드폭이 전단계의폭과 같으면 수행을 완료한다. 식 (6)에서 목표밴드폭은 정수이어야 하므로 계산된 소수부분은 버린다.

(v) 식 (6)으로 구한 초기목표밴드폭을 이용하여 번호를 부여하고 최종 밴드폭을 얻는다. 만약 최종 밴드폭이 $band_{min}$ 보다 작게되면 $band_{min} = band_{final}$ 로 하고 (iv)로 진행한다. 그렇지 않으면 $band_{min}$ 을 최종값으로 하여 수행을 완료한다.

(3) 절점에 대한 번호 부여과정

절점 번호 부여는 어떤 절점에 번호를 부여하는가에 따라 다음과 같이 크게 세가지 경우로 나눌 수 있다. 첫번째 경우는 비선단집합에 속한 절점에

번호를 부여하는 경우로 이 과정에서 인접집합의 크기는 작아지게 된다. 두번째 경우는 선단집합에 속한 절점에 번호를 부여하는 과정으로 인접집합의 크기는 작아지지 않는다. 마지막으로 세번째 경우는 인접집합에 속하지 않은 절점에 번호를 부여하는 과정으로 이 경우에는 인접집합의 크기는 커지게 된다.

한편, 인접집합의 크기는 가능한 한 목표밴드폭보다 작게 유지되어야하므로 위의 세가지 경우를 적절히 조합하여 번호를 부여하면 좋은 결과를 얻을 수 있으며 다음과 같은 과정에 따라 번호를 부여하여 그 크기를 제어한다.

- (i) 요소번호 1번에 있는 절점 중에서 연결꼭지점수가 가장 적은 절점을 찾아 절점번호 1번을 부여한다. $k=1$
- (ii) 인접집합(A), 새롭게 인접집합에 주입된 절점들의 집합(A_k), 이때 인접집합의 크기(b_k)를 구한다.
- (iii) 인접집합의 크기가 목표밴드폭보다 작거나 같으면 (iv)로 진행한다. 그렇지 않으면 목표밴드폭을 인접집합의 크기로 바꾸고 (i)로 되돌아간다.
- (iv) 식 (7)을 만족하는 j 를 찾는다.

$$j = \min(1, 2, \dots, k) \quad (7)$$

$$\{A_j \cap A\} \neq \emptyset$$

만약, b_j 가 목표밴드폭보다 작으면 (vi)으로 진행한다.

(v) A_i 의 비선단집합에 번호를 부여하고 A_j 에 비선단집합이 없으면 $k=k+1$ 로 하고 A_j 의 선단집합 중에서 한점을 선택하여 번호를 부여한 후 (ii)로 진행한다.

만약, 모든 절점에 번호가 부여되었다면 (2)에서 기술한 대로 초기목표밴드폭을 수정한 다음 (i)로 되돌아간다.

- (vi) $\max(b_{j+1}, b_{j+2}, \dots, b_k) < \text{목표밴드폭}$ 이면 (ix)로 진행, 아니면 $m=j+1$.
- (vii) A_m 의 비선단집합에서 한점을 선택하여 번호를 부여한다.
- (viii) 밴드폭 조절변수를 수정한다.

$$b_i = b_i + 1, i = j, j+1, \dots, m-1 \quad (8)$$

A_m 의 모든 비 선단집합에 번호가 부여되었으면 $m=m+1$ 로 한다.

$\max(b_j, b_{j+1}, \dots, b_{n-1}) < \text{목표밴드폭}$ 이면 (vii)로 진행, 아니면 (v)로 진행한다.

(ix) 인접집합에 없는 절점에 번호를 부여하더라도 인접집합의 크기가 목표밴드폭보다 커지지 않는다면 $k=k+1$ 로 하고 해당 절점에 번호를 부여한 후 (ii)로 진행, 그렇지 않으면 $m=j+1$ 로하여 (vii)로 진행한다.

과정 (v)와 과정 (ix)에서 번호 부여 대상 절점이 여러개일 경우 어떤 절점에 번호를 부여 하느냐에 따라 최종 밴드폭은 상당한 차이가 나므로 적절한 절점 선택의 기준이 필요하다. 많은 예제를 통하여 다음과 같은 절점 선택의 기준을 제안한다.

- (1) 인접집합을 가장 적게 증가시키는 절점.
- (2) 인접집합에 없는 요소중에서 비선단집합에 인접하고 있으면서 인접집합을 가장 적게 증가시키는 절점.
- (3) 수준이 가장 낮은 절점.
- (4) 유한요소의 번호가 가장 작게 부여된 요소에 속해있는 절점.

과정 (v)에서는 (1), (3), (4)의 순서를, 과정 (ix)에서는 (2), (3), (4)의 순서를 따른다.

4. 결과 및 고찰

제안된 알고리즘의 타당성을 입증하기 위하여 Fig. 1로부터 Fig. 18까지 총 18개의 예제에 대해

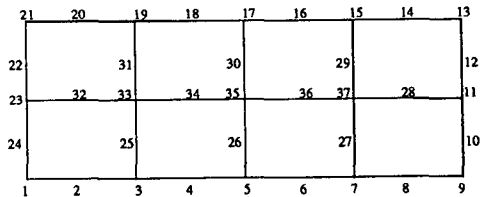


Fig. 1 Example 1. Cantilever beam

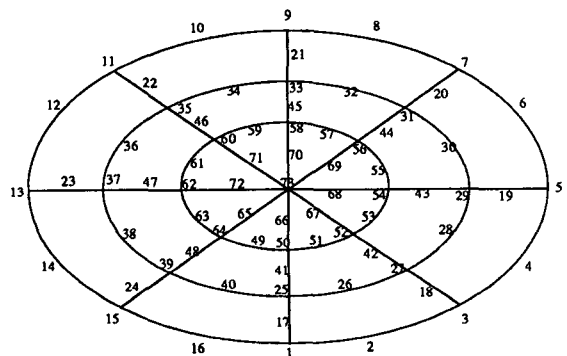


Fig. 2 Example 2. Circulate plate

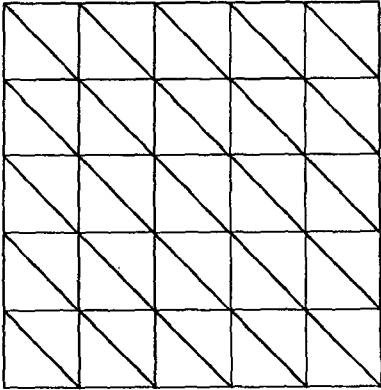


Fig. 3 Example 3. Rectangular plate

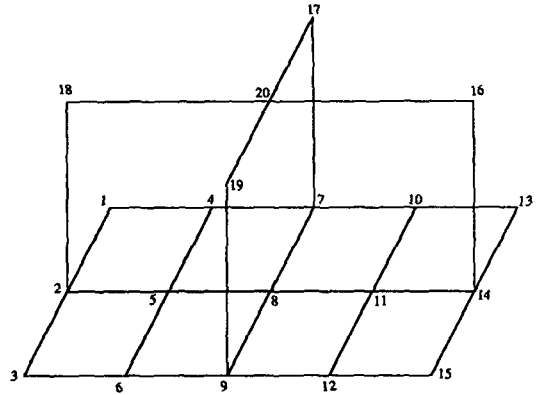


Fig. 6 Example 6. 3-D structure

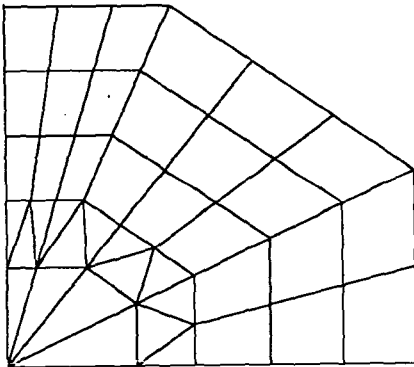


Fig. 4 Example 4. Grillage

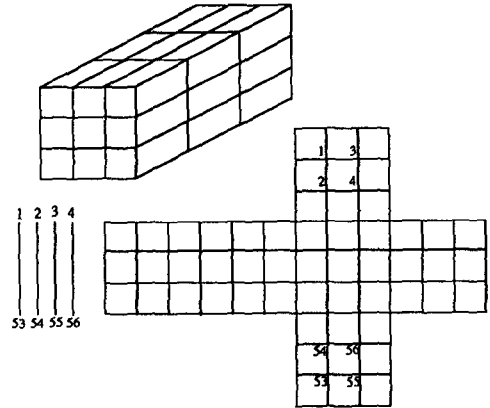


Fig. 7 Example 7. Box structure

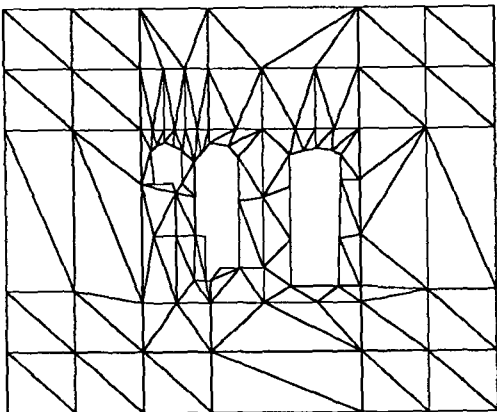


Fig. 5 Example 5. Underground cavities in rocks

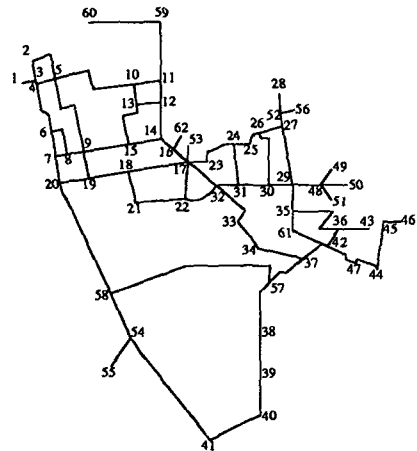


Fig. 8 Example 8. Water distribution network

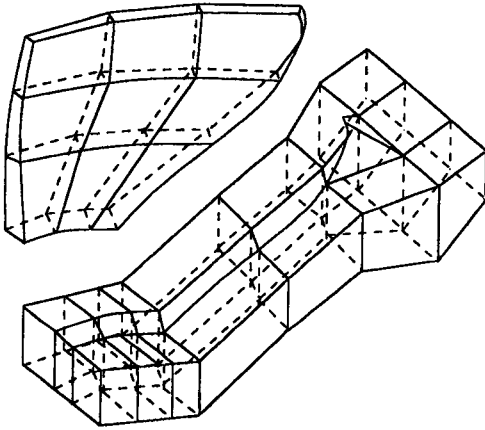


Fig. 9 Example 9. Arch dam with its foundation

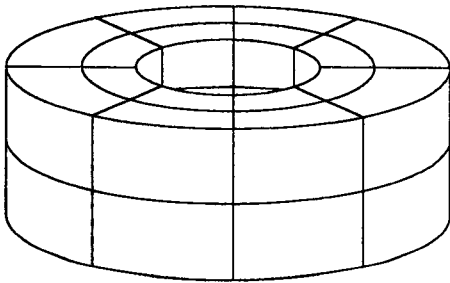


Fig. 10 Example 10. Wheel

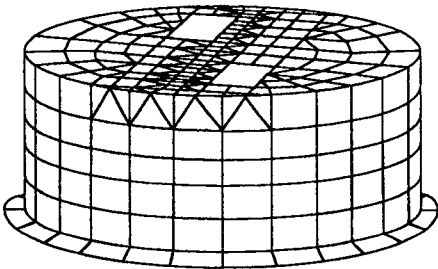


Fig. 11 Example 11. Power supply housing

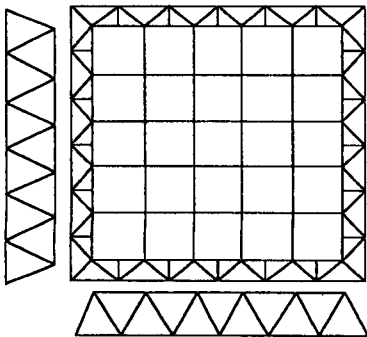


Fig. 12 Example 12. Roof of a factory

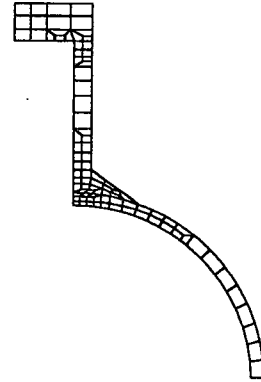


Fig. 13 Example 13. Pressure vessel

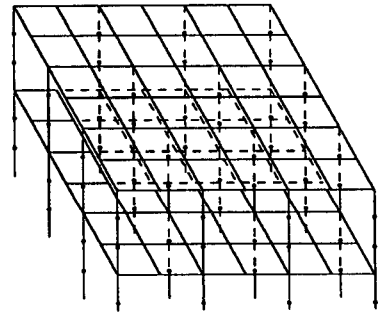


Fig. 14 Example 14. 2-story building

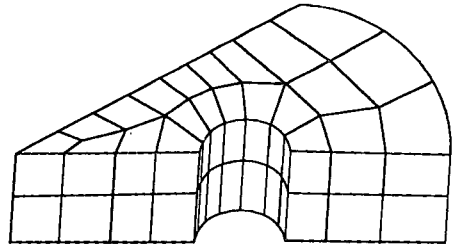


Fig. 15 Example 15. Circular plate with a hole

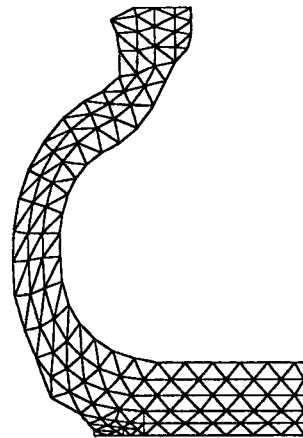


Fig. 16 Example 16. Tire

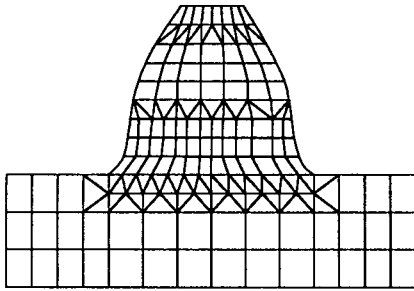


Fig. 17 Example 17. Gear tooth

여 적용하였고 문제의 특성을 Table 1에 보였다. 문제를 보다 간단히 하기 위하여 각 절점당 자유도를 1로 가정하였고 따라서, 절점연결행렬의 밴드폭

Table 1 Topological properties of the example

	No. of node	No. of element	Element type*	Original bandwidth
Example 1	37	8	6	33
Example 2	73	24	4, 6	40
Example 3	121	50	4	33
Example 4	42	81	1	38
Example 5	101	151	3	37
Example 6	20	17	1, 5	17
Example 7	56	58	1, 5	53
Example 8	62	78	1	49
Example 9	84	31	7	65
Example 10	96	48	7	48
Example 11	307	334	3, 5	56
Example 12	85	288	1	50
Example 13	350	89	6	146
Example 14	442	168	2, 6	370
Example 15	351	44	8	152
Example 16	490	213	4	243
Example 17	529	186	4, 6	128
Example 18	351	304	5	292

Element type*

- 1 : Linear line element(2node)
- 2 : Quadratic line element(3node)
- 3 : Linear triangular element(3node)
- 4 : Quadratic triangular element(6node)
- 5 : Linear quadrilateral element(4node)
- 6 : Quadratic quadrilateral element(8node)
- 7 : Linear hexahedral element(8node)
- 8 : Quadratic hexahedral element(20node)

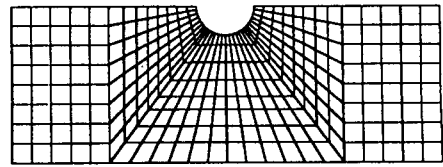


Fig. 18 Example 18. Notched plate

을 줄인 것이다. GPS방법은 참고문헌 (9)에 수록된 프로그램을 이용하였고, Puttonen 방법은 직접 프로그램 하였다. Table 2에서 각 문제에 대한 최종 밴드폭과 소요시간은 IBM PC AT에서 수행한 결과이며 CPU 처리시간은 절점들의 연결관계행렬이 구성된 이후 부터 알고리즘의 수행이 완료될 때까지 소요된 시간을 나타낸 것이다.

Table 2에서 제안된 방법을 기준으로한 백분율의 평균치를 구해보면 계산시간의 경우 Puttonen 방법은 2160%이고 GPS 방법은 58.6%이며 밴드폭의 경우 Puttonen 방법은 110%, GPS 방법은 113%이다. 이로부터 Puttonen의 방법은 소요 시간이 매우 길고 밴드 폭도 별로 우수하지 않아 개

Table 2 Results of bandwidth reduction algorithms

	Puttonen		GPS		Present	
	Bandtime*	Bandtime*	Bandtime*	Bandtime*	Bandtime*	Bandtime*
Example 1	12	0.38	11	0.06	11	0.11
Example 2	25	2.91	28	0.17	24	0.22
Example 3	24	4.37	24	0.22	22	0.33
Example 4	10	0.22	9	0.06	8	0.11
Example 5	19	5.61	19	0.11	17	0.27
Example 6	8	0.11	8	0.05	7	0.05
Example 7	17	0.66	17	0.11	16	0.22
Example 8	8	0.71	9	0.11	9	0.11
Example 9	25	4.50	28	0.33	24	0.50
Example 10	40	10.11	44	0.44	33	0.82
Example 11	40	51.41	35	0.44	38	2.20
Example 12	16	1.05	15	0.17	15	0.55
Example 13	36	38.34	40	0.93	30	2.04
Example 14	74	467.64	83	1.37	62	8.02
Example 15	65	265.73	74	6.48	70	4.45
Example 16	38	69.81	36	1.21	33	2.04
Example 17	62	439.63	71	1.92	59	4.78
Example 18	22	29.99	20	0.77	18	2.25

time* : CPU time(sec) required for bandwidth reduction on the IBM PC AT

선이 필요한 방법이라 생각된다. GPS 방법의 경우, 소요 시간은 가장 짧으나 밴드폭이 제안된 방법에 비해 크고 특히 2차 유한 요소 모델이나 연결 관계가 복잡한 3차원 문제인 Example 10,13,14,17의 경우 밴드폭이 20~30% 정도 크게 나타나 안정성이 부족한 것으로 보인다.

결론적으로 계산 시간에 관해서는 제안된 방법이 GPS 방법에 비해 열등하나 번호를 붙이는 시간이 전 유한 요소 해석 시간에 비해 상당히 작은점과 GPS 방법에 비해 최종 밴드폭이 작고, 여러 문제에 대해 안정적으로 결과를 주고 있는 점을 고려하면 제안된 방법이 GPS 방법에 비해 실용성이나 신뢰성의 면에서 우수하다고 생각된다.

5. 결 론

유한요소법에 효과적으로 적용할 수 있는 밴드폭 감소 알고리즘을 제안하였다. 제안된 방법은 이단계 번호 부여 방법과 선단집합 이용 방법을 결합한 것으로 효율성과 신뢰성이 제고된 방법이다. 여러 예제에 대한 수치 실험을 통해 컴퓨터 처리 시간이 길지않으면서 요소의 종류에 관계없이 효과적으로 밴드폭을 줄일수 있는 방법임을 보였다.

참 고 문 헌

- (1) 구분용, 1990, "절점 및 요소 근접도에 근거한 유한 요소 모델의 효율적인 절점 번호 부여 방법 개발", 한국과학기술원 석사학위 논문.
- (2) 이병채, 1989, "유한요소 행렬의 밴드폭 감소를 위한 절점 번호 부여 알고리즘 개발", 한국항공우주학회지, 제17권, pp. 39~45.
- (3) Norman, E. Gibbs, William, G. Poole and Paul K. Stockmeyer, 1970 "An Algorithm for Reducing the Bandwidth and Profile of a Sparse Matrix", SIAM J. Numer. Anal., Vol. 13, pp. 236~250.
- (4) Jari Puttonen, 1983, "Simple and Effective Bandwidth Reduction Algorithm", International Journal for Numerical Methods in Engineering, Vol. 19, pp. 1139~1152
- (5) Steven J. Fenves and Kincho H. Law, 1983, "A Two-Step Approach to Finite Element Ordering", International Journal for Numerical Methods in Engineering, Vol. 19, pp. 891~911
- (6) Akhras and Dhatt, 1976, "An Automatic Node Relabelling Scheme for Minimizing a Matrix or Network Bandwidth", International Journal for Numerical Methods in Engineering, Vol. 10, pp. 787~797.
- (7) Cuthill, E. and McKee J. M. 1969 "Reducing the Bandwidth of Sparse Symmetric Matrices," Proc. 24th Nat. Conf., Assn. for Computing Machinery, ACM Pub., pp. 157~172.
- (8) Kee-Young Yoo, 1978, "An Algorithm for Bandwidth Minimization of Sparse Matrix", M. S. Thesis of KAIS,
- (9) Collected Algorithms from ACM, Vol. III, Algorithm 508, A Publication of the Association for Computing Machinery Inc., 1983, New York