정자계의 유한요소해석을 위한 병렬계산

김근환*0 최경** 이기식*** 한송엽*
* 서울대학교 ** 강원대학교 *** 단국대학교

Parallel Computations for Finite Element Analysis of Magnetostatic Fields

Kim Keunhwan*0 Choi Kyung** Lee Kisik*** Hahn Songyop*
* Seoul National Univ. ** Kangweon National Univ. *** Dankuk Univ.

Abstract

In the field of structural analysis so-called substructure methods have been applied extensively to solve large and complex structures by splitting them up in substructures. This substructure method is applicable to electromagnetic field analysis and highly parallel in nature.

In this paper substructure FEM is implemented for magnetostatic field computation using parallel computer consists of many transputers. Parallel substructure method is a promising tool as a solution of not only compuation speed problem but also memory problem.

1. 서론

유한요소법을 이용하여 전자계해석을 할때 해석대상이 복잡하거나 대형화되면(특히 3차원해석의 경우) 컴퓨터의 계산속도와 메모리의 한계에 부딪히게 된다. 최근 공학용 워크 스테이션이 싼 가격에 매우 빠른 계산속도를 보이고 있지만 한개의 CPU를 사용하는 컴퓨터는 그 성능에 한계가 있어 점차로 고성능 컴퓨터는 다수의 CPU를 쓰는 경향이다.[9]

최근 트랜스퓨터를 이용한 병렬계산을 하여 유한 요소해석을 한 논문들이 나오고 있는데 다수의 트랜스퓨터를 사용하여 병렬시스템을 구축하면 저가격에 막강한 계산성능을 얻어낼 수 있다.[5],[6]

본 연구에서는 트랜스퓨터를 이용하여 유한요소해석을 하는 방법을 제시하고자 한다. 계산 구조역학 분야에서 일찌기 컴퓨터 메모리 한계때문에 많이 사용해오고 있는 부분구조법(substructure method)을 병렬화하여 동시계산의 시간절약효과와 영역분할에 의한 메모리문제 해결 방법을 연구하였다.[1]-[4]

2. 병렬 부분구조법

2.1 부분구조법

유한요소해석시 직접법을 써야할 경우가 있는데 미지수가 많으면 메모리와 계산시간면에서 매우 불리하다. 직접법의 효율을 높이는 방법에는 skyline storage법과 wave front법, 부분구조법등이 있는데, 부분구조법은 메모리를 많이 필요로 하는 대형계산문제를 여러부분으로 나누어 계산을 하여 메모리를 크게 절약하는 방법이다.[8]

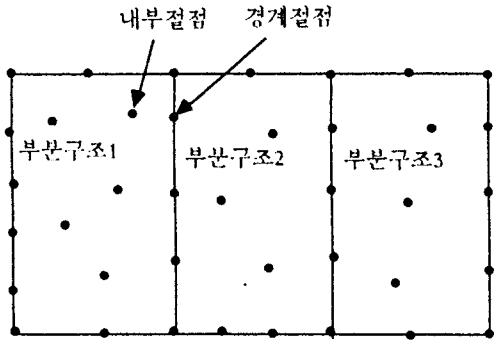

그림 1 세 부분구조를 가진 모델
Fig. 1 A model which has three substructures.

그림1과 같이 세부분으로 나누어져 있는 해석영역이 있다고 할때 각 부분구조에서 $\{U_e\}$를 경계절점의 포텐셜벡터, $\{U_i\}$를 내부절점의 포텐셜벡터, $[A]$를 그 부분구조의 계행렬(system matrix)이라고 하고 $\{B\}$를 구동벡터라고 하면, $[P]\{U\}=\{B\}$이다. 내부절점, 경계절점에 대해 분리해서 쓰면,

$$\begin{bmatrix} [A_{ee}][A_{ei}] \\ [A_{ie}][A_{ii}] \end{bmatrix} \begin{bmatrix} \{U_e\} \\ \{U_i\} \end{bmatrix} = \begin{bmatrix} \{B_e\} \\ \{B_i\} \end{bmatrix} \quad (1)$$

(1)식에서 $\{U_i\}$를 소거하면

$$\{U_i\} = [A_{ii}]^{-1}(\{B_i\} - [A_{ie}]\{U_e\}) \quad (2)$$

$$([A_{ee}] - [A_{ei}][A_{ii}]^{-1}[A_{ie}])\{U_e\} = \{B_e\} - [A_{ei}][A_{ii}]^{-1}\{B_i\} \quad (3)$$

(3)식을 이용하여 전체 경계절점에 대한 계행렬을 조립하여 경계조건을 대입하고 풀어서 $\{U_0\}$를 구하고 이것을 (2)식에 대입하여 $\{U_1\}$를 구한다.

2.2 병렬 알고리즘

그림2의 흐름도의 첫번째 과정에서는 각 부분구조의 계행렬을 조립하고 내부점 소거를 하는데 각 트랜스퓨터가 완전히 독립적으로 수행한다. 이 과정에서는 내부점 소거를 위해서 $[A_{11}]$의 역행렬을 계산하는데 이 과정이 계산시간의 상당한 부분을 차지한다. 두번째과정은 매스터 트랜스퓨터가 각 트랜스퓨터로부터 내부점소거에 의해 변형된 경계절점행렬을 받아 재조립하고 경계치 대입후 병렬 가우스 소거법을 이용하여 전영역의 $\{U_0\}$를 구한다. 세번째과정은 계산된 $\{U_0\}$를 받아서 각

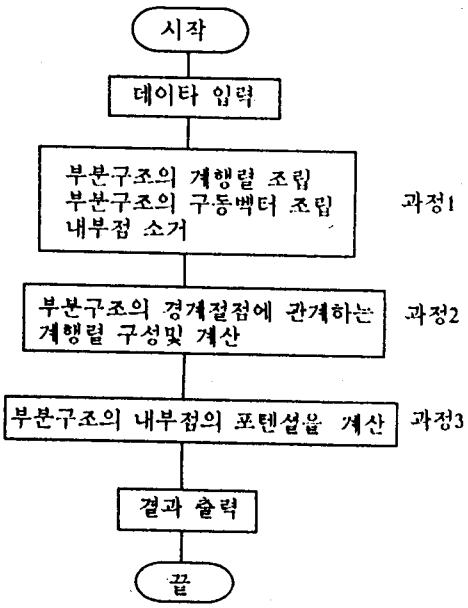

그림 2 병렬부분구조법의 흐름도
Fig. 2 Flowchart of parallel substructure method.

부분구조의 내부점의 포텐셜을 계산하는 과정인데 역시 각 트랜스퓨터가 동시에 계산한다.

실제로 시간이 많이 걸리는 부분은 첫째와 둘째과정인데 총 경계절점과 총내부절점을 적절히 조절해서 두과정에서 걸리는 시간을 합한것이 최소가 되도록 해야 한다. 또한 주의할 것은 부하균등(load balance)문제인데 각 트랜스퓨터가 담당하는 부분구조의 절점의 갯수가 모두 같거나 비슷해야 제대로 효율이 나온다.

여기서 병렬계산의 속도증가와 효율을 정의하면,

$$\text{속도증가} = \frac{\text{M개 프로세서의 계산시간}}{\text{1개 프로세서의 계산시간}}$$

$$\text{효율} = \frac{\text{속도 증가}}{\text{M}} \times 100[\%]$$

이다.

3. 병렬 가우스 소거법

본 연구에서는 직접법에서 가장 계산량이 적은 가우스 소거법을 사용하였다. 계산시간의 거의 대부분을 차지하는 전진 소거과정만 병렬화하고 후퇴대입과정은 매스터 트랜스퓨터에서 수행했다. 왜냐하면 후퇴대입과정은 병렬성이 좋지 않기 때문이다.

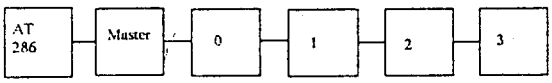

그림 3 트랜스퓨터의 선형연결구조
Fig. 3 Linear topology of transputer network.

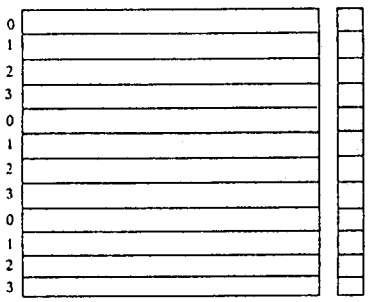

그림 4 병렬 가우스 소거법의 행렬배치법
Fig. 4 Matrix distribution scheme for parallel Gaussian elimination.

트랜스퓨터의 연결구조는 그림3과 같은 선형구조이다.[9],[10] 행렬은 각 트랜스퓨터의 부하균등을 위해서 각행을 돌아가면서 배치한다.(그림4) 구동벡터도 순환적으로 배치되어 있다. 전진소거의 맨 바깥쪽 루프가 한 스텝씩 수행될 때마다 방송(broadcasting)을 한다. 방송은 현재 처리하고자 하는 행을 가지고 있는 트랜스퓨터가 그 행의 요소들을 대각선 요소로 나누어서 그 값을 다른 트랜스퓨터에 보내고 다른 트랜스퓨터들은 이것을 받음으로써 이루어진다. 모든 트랜스퓨터가 방송된 정보를 가지게 되면 곱하기와 빼기를 수행하여 자신이 가지고 있는 부분 행렬의 내용을 변경한다.

N x N행렬의 경우 (N-1)번의 방송을 하고 한번의 방송시 길이 N의 실수벡터를 다른 트랜스퓨터에전송하므로 전진소거과정중 전송하는데 걸리는 시간은 최소한 (N-1) x N x 4 x 8 x (M/2) / 20^6 [초]이 된다. 단, 여기서 N은 행렬의 크기,M은 트

랜스퓨터의 수, 2^6은 트랜스퓨터의 전송속도이다. 행렬의 크기가 너무 크면 즉 경계절점수가 너무 많으면 통신에서 오버헤드가(overhead)가 많아서 효율이 떨어진다.

언어는 트랜스퓨터용 병렬처리 언어인 Occam을 사용했다.[11],[12]

4. 사례연구

병렬 부분구조법의 효용성을 입증하기 위해서 전영역을 8개, 16개, 24개의 부분구조로 나눈 모델을 트랜스퓨터를 8개, 16개, 24개 써서 계산하였다. 또한 트랜스퓨터 한개에서 skyline storage법을 이용한 가우스 소거법을 써서 계산하고 그 계산시간들을 병렬 부분구조법의 계산시간과 비교하였다. 그림5에 나와 있는 것처럼 모델은 해석적인 해가 나와있는 2차원 모델로서 skyline storage를 위해 재정렬을 하면 저장할 부분이 매우 작아져서 메모리 측면에서 효율적인 모델이다.[7] 요소분할은 부하균등을 고려해서 각 부분구조에서 거의 같은 요소수가 되게 하였다. 계산결과 두방법 모두 자속의 특이점 부분인 원점부분을 제외하고는 오차가 1%이하 정도로 해석적인 해와 일치했다.

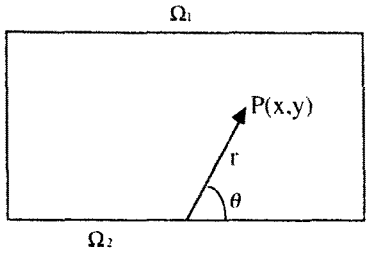

그림 5 해석적인 해가 존재하는 2차원 모델
Fig. 5 A two-dimensional model which has analytic solution.

표1에서 두방법의 계산시간을 비교했다. 표1에서 보면 효율이 낮은 경우와 매우 높은 경우를 볼 수 있는데 이것으로써 경계절점수와 내부절점수의 크기를 잘 조절해야 높은 효율을 얻을 수 있다는 것을 알 수 있다. 즉 각 부분구조를 맡은 트랜스퓨터가 내부점 소거를 위해 역행렬 계산을 하는데 드는 시간과 전체 경계절점에 대한 행렬을 가우스 소거법으로 계산하는데 드는 시간을 고려해서 내부절점수와 경계절점수의 비를 적절히 조절해야 효율이 좋아진다.

16개의 경우는 효율이 좋은 경우로서 전체 경계절점 405혹은 495개를 16개의 트랜스퓨터로 해를 계산하는 시간과 각 트랜스퓨터가 역행렬을 계산하는 시간이 둘다 적게 걸리는 경우이다.

표 1 계산시간 비교표
Table 1 Comparison of execution times

부분구조 갯수	절점수	요소수	계산시간 (초)	효율 [%]	경계절점
8	1914	3694	256.3 66.2	48.4	235
	966	1702	49.7 16.1	38.8	411
16	2265	4360	374.86 22.29	105	405
	2526	4846	472.6 30.3	97.5	495
24	2019	3728	235.7 38.4	25.6	803
	3021	5798	622.9 29.8	87.1	626

단,계산시간에서 윗것은 skyline법에 걸린시간
아랫것은 병렬부분구조법에 걸린시간.

5. 결론

본 연구에서는 유한요소해석에 병렬 부분구조법을 적용하여 계산속도면과 기억용량면에서 효율을 높일 수 있었다. 2차원 문제로 그 효용성을 검정하였으나 3차원 문제에도 쉽게 적용이 가능하리라고 생각한다.

이 방법의 장점으로는 skyline storage법에 비해 전처리가 적게 걸리고 계산속도가 매우 빠르며 대용량의 메모리가 요구되는 문제를 해결하는데 사용될 수 있다는 것이다. 단점으로는 각 트랜스퓨터의 부하균등을 고려해야 하는데 이를 위해서 요소분할이 된 데이타를 자동으로 부분구조로 분할하는 전처리 과정이 필요하다는 것이다. 또한 경계절점과 내부절점의 수를 적절히 조절해서 총계산시간을 최소로하는 구성비를 찾아야 한다.

6. 참고문헌

[1] K. R. Weeber and S. R. H. Hoole, "The subregion method in magnetic field analysis and design optimization", COMPUMAG 8th Conference , July 7-11, 1991, Sorrento, Italy.

[2] J.S. Przemieniecki, "Matrix structural analysis of substructures", Journal of American Institute of Aerospace and Aeronautics, Vol1, No1,1963.

[3] R. Rosen, M. Rubinstein, "Substructure analysis by matrix decomposition", Proc. of the ASCE, Structural Division, Mar. 1970.

[4] C. Farhat and E. Wilson, "A new finite element concurrent computer program architecture", IJNME, Vol.24, 1987.

[5] C.F.Bryant, M.H.Roberts, and C.W.Trowbridge, "Implementing a boundary integral method on a transputer system", IEEE Trans. on Magnetics, Vol.26, No.2, March 1990.

[6] G. Yagawa, N. Soneda and S. Yoshimura, "A large scale finite element analysis using domain decomposition method on parallel computer", Computers and Structures, Vol. 38, 1991.

[7] 고창섭,정현교,한송엽, "적응 경계요소법을 이용한 2차원 정자계 해석", Trans. KIEE, Vol.40, No.3, MAR. 1991.

[8] G. Yagawa and S. Yoshimura,"有限要素法",培風館,1991.

[9] Kai Hwang and Faye A. Briggs, "Computer architecture and parallel processing", McGraw Hill International Editions, 1985.

[10] Dimitri P. Bertsekas and John N. Tsitsiklis, "Parallel and distributed computation", Prentice-Hall International Editions, 1989.

[11] Pountain, Dick and David May, "A tutorial introduction to Occam programming", BSP Professional Books, London, 1987.

[12] INMOS, "Occam 2 Reference Manual", Prentice Hall, 1988.