SOR 방법을 이용한 효율적인 FEM 계산

Efficient FEM analysis by using an SOR method

김운식, 이종창
홍익대학교 전자전기공학부
wave@wow.hongik.ac.kr

유한요소법(Finite-element method; FEM)은 광통신소자의 특성해석 및 설계에 광범위하게 사용되어 왔다. 특히 복잡한 2차원 및 3차원 구조를 갖는 소자에서 광모드의 분포 및 전위분포 계산에 FEM은 다른 수치해석 방법인 FDM(Finite-difference method) 및 FDTD(Finite-difference time domain)에 비해 보다 정확하고 소요계산시간이 작은 해석방법으로 인식되어 왔다. 그러나 FEM은 소요 메모리가 상대적으로 커서 3차원 구조의 해석을 위해서는 용량이 큰 workstation이 필수적인 단점이 있다. 본 논문에서는 이러한 대용량 FEM의 계산에서 행렬의 sparse 특성을 이용하여 소요 메모리 용량을 줄일 수 있는 효율적인 방법을 제시하려 한다.

일반적으로 전위분포 해석과 같은 static FEM에서는 각 유한요소에서의 Lagrangian functional을 통합하고 경계치 조건을 적용하면 $AX=B$ 와 같은 global matrix equation을 얻게 된다[1]. 이때 A 는 node의 수가 M 개일 때 $M\times M$ 인 행렬로 주어진다. 예를 들면 그림 1과 같은 rectangular waveguide 에서 가로 세로로 20개씩의 grid로 그림 1(b)와 같이 분할하면 행렬 A 의 크기는 861×861이 된다. 행렬 A 의 non-zero 항의 위치를 나타내면 그림 1(c)와 같이 되어 861×861개의 항 중에서 non-zero 항의 개수는 5,297개로서 sparse matrix 가 됨을 알 수 있다. 이때 각 node의 전위분포는 $X=A^{-1}B$ 의 역행렬식으로부터 얻게 되는데 일반적인 Gauss elimination 방법이나 LDU decomposition 방법을 사용하게 되면 행렬 A 의 non-zero 항까지 모두 저장하여야 하는 비효율적인 면이 발생한다[2].

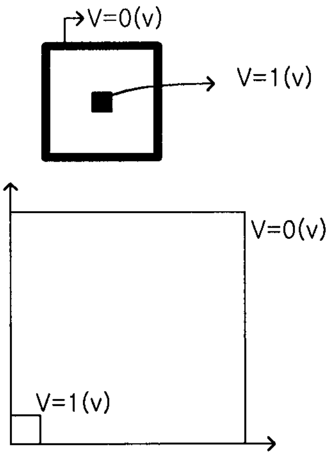

그림 1(a). Rectangular waveguide

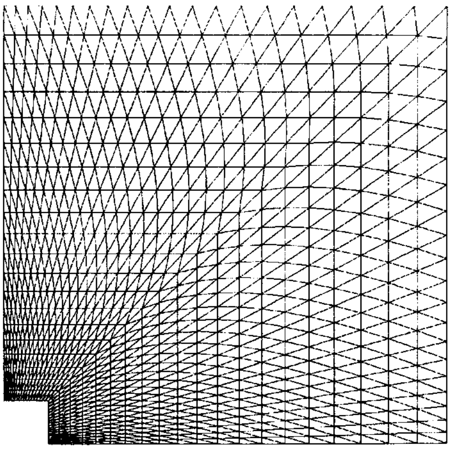

그림 1(b). Grid geometry.

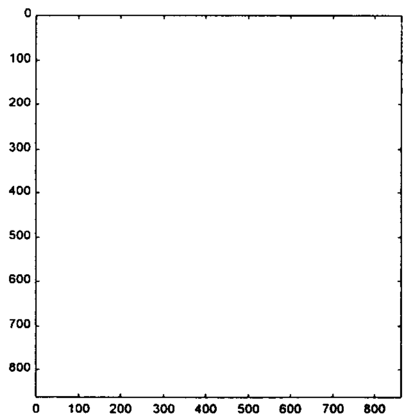

그림 1(c). 행렬 $A[861\times861]$의 non-zero 항 (5,297개)의 위치.

본 논문에서는 SOR(successive over-relaxation) 방법[2]을 이용하여 FEM에서 효율적으로 역행렬을 계산할 수 있는 알고리듬을 개발하였다. 이 방법에서는 non-zero 항만을 저장하고 계산 시에도 non-zero 항만 고려해 주기 때문에 전산 저장용량과 계산시간을 10 배 이상 줄일 수 있다. 이 알고리듬에서는 행렬 A를 lower triangular matrix L과 upper triangular matrix U, 그리고 diagonal matrix D 등으로 구분한다. 즉 $A \equiv L+D+U$가 된다. 이 때 전위분포 행렬 X는 다음과 같은 iteration 방법으로 구할 수 있다.

$$(I+\omega D^{-1}L)X^{(m+1)} = \omega D^{-1}B + [(1-\omega)I - \omega D^{-1}U]X^{(m)}$$

여기서 $X^{(m)}$은 m 번째 iteration에서 얻은 값이며 ω는 relaxation constant로서 행렬차수가 M일 때 근사적으로 $\omega \simeq 2/(1+\pi/\sqrt{M/2})$의 값을 갖게 된다. 이때 허용오차를 10^{-p}라 할 때 iteration 회수를 $m \simeq p\sqrt{M/2}\ln 10/2\pi \simeq p\sqrt{M/2}/3$정도로 늘려주면 된다. 다른 Jacobi 방법이나 Gauss-Seidel 방법에서는 iteration 회수가 M에 비례하는 것에 비교하면 이 SOR 방법은 계산시간이 획기적으로 줄게 된다[2]. 아래 표에 그림 1(a)의 rectangular waveguide를 가로 세로로 여러 경우의 grid 수로 분할하였을 때 행렬 A의 크기와 ω의 값을 정리하였다. 각각의 grid 수에 대하여 SOR과 straight forward gauss algorithm[3]을 이용하여 위 문제를 해석한 결과를 비교하면 다음 그림 2와 같다. 그림 2(a)는 소요된 CPU time이며 그림 2(b)는 메모리 저장용량으로서 행렬의 크기가 커질수록 두 값 모두 현저히 줄어듦을 볼 수 있다. 그리고 그림 2(c)는 SOR을 통해서 얻은 전위분포 contour plot로서 원하는 오차범위 한계 내의 결과를 쉽게 얻을 수 있음을 보여준다.

grid	5	10	15	20	25
size of A	66×66	231×231	496×496	861×861	1326×1326
ω	1.909125	1.973165	1.987412	1.992728	1.995272

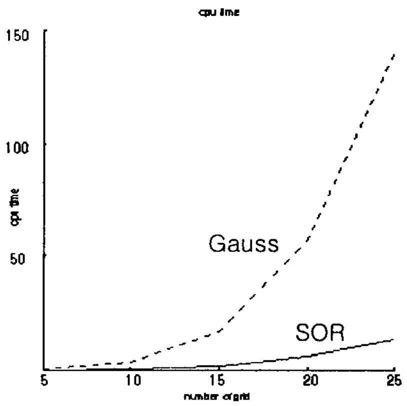

그림 2(a). CPU time.

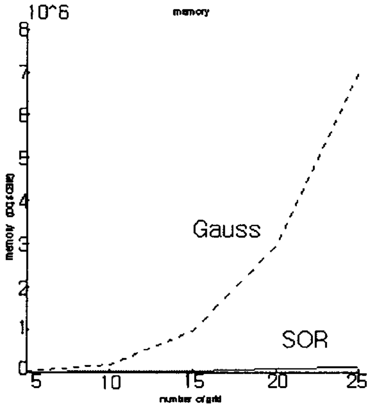

그림 2(b). 메모리 용량.

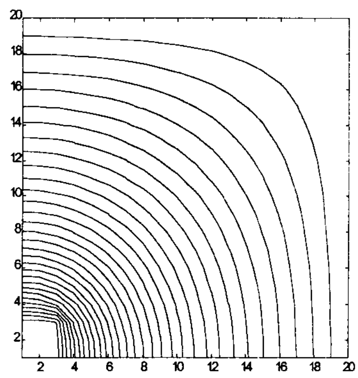
그림 2(c). 등전위분포.

[참고문헌]

1. G. Pelosi and R. Coccioli, *Quick Finite Elements for Electromagnetic Waves*, (1998).
2. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran,* 2nd ed., (1992).
3. S. C. Chapra, and R. P. Canale, *Numerical Methods for Engineers,* 2nd ed., (1990).