PRECORRECTED-FFT METHODS FOR ELECTROMAGNETIC ANALYSIS OF COMPLEX 3-D INTERCONNECT AND PACKAGES *

J. R. PHILLIPS
DEPARMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, ROOM 36-881, MIT
JPHILL@RLE-VLSI.MIT.EDU

The method of moments, commonly used to discretize potential-integral equations describing the electromagnetic interactions in complicated 3-D interconnect structures, generates large, dense matrix problems. If Gaussian elimination is used to solve method-of-moments matrices, $O(n^3)$ operations and $O(n^2)$ storage is required, where n is the number of panels into which the structure has been discretized. Typical engineering problems may have thousands or tens of thousands of panels, so that Gaussian elimination is not a feasible approach. Instead, the method-of-moments equations can be solved using a Krylov-subspace technique such as GMRES[5]. The dominant costs of such an algorithm are in calculating the n^2 entries of the method-of-moments matrix, P, before iterations begin, and performing n^2 operations to compute the dense matrix-vector product on each iteration.

To develop even faster approaches to solving the method-of-moments equations, it is necessary to combine the Krylov-subspace iterative method with a faster approach to computing the matrix-vector products. To achieve this acceleration, after discretizing the problem into n panels, the problem domain is subdivided into an array of small cubes so that each small cube contains only a few panels. Several sparsification techniques for P are based on the idea of directly computing only those portions of Pq associated with interactions between panels in neighboring cubes. The rest of Pq is then approximated to accelerate the computation, typically using multipole expansions [2, 3, 4]. In this paper we describe a precorrected-FFT based algorithm which is avoids the high memory overhead associated with the multipole algorithms, is generalizable to a wide variety of Green's functions, and for reasonably homogeneous problems requires $O(n)$ memory and $O(n \log n)$ computation time.

The basic idea is to compute distant interactions using a representation of the given cube's charge distribution using a small number of weighted point charges[1]. Pq can then be approximated in four steps: (1) project the panel charges onto a uniform grid of point charges, (2) compute the grid potentials due to grid charges using the FFT, (3) interpolate the grid potentials onto the panels, (4) directly compute nearby interactions. This process is summarized in Figure 1. The table below demonstrates the effectiveness of the precorrected-FFT methods as compared to fast-multipole algorithms for capacitance extraction.

REFERENCES

[1] L. Berman, *Grid-multipole calculations*, Tech. Rep. RC 19068(83210), IBM Research Report, 1993.

[2] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, M.I.T. Press, Cambridge, Massachusetts, 1988.

[3] V. Jandhyala, E. Michielssen, and R. Mitra, *Multipole-accelerated capacitance computation for 3-d structures in a stratified dielectric medium using a closed form green's function*, tech. rep., University of Illinois, 1994.

[4] C.C. Lu and W. C. Chew, "Fast Algorithms for solving Hybrid Integral Equations," IEEE Proceedings-H Vol. 140, No. 6, December 1993, pp. 455-460.

[5] Y. Saad and M. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

* This work was supervised by Professor Jacob White and supported by ARPA contracts N00014-91-J-1698, N00174-93-C-0035, and J-FBI-92-196 as well as grants from the Semiconductor Research Corporation (SJ-558), IBM and Digital Equipment Corporation.

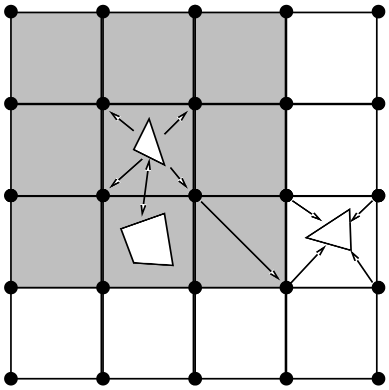

FIG. 1. *2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid.*

Example	Speed	Memory
micromotor	0.68	0.81
cube	0.73	0.31
woven bus	0.63	0.42
bus crossing	0.43	0.26
via	1.42	0.37
DRAM cell	0.80	0.73

FIG. 2. *Comparison of performance of FFT-based to multipole-based codes for* $1/r$ *Green function. "Speed" is ratio of matrix-vector product time of pre-corrected FFT method to fast multipole based method, "memory" the ratio of required storage*