A Precorrected-FFT method for Capacitance Extraction of Complicated 3-D Structures

J. R. Phillips J. White
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139.

Abstract

In this paper we present a new approach to three-dimensional capacitance extraction based on a precorrected FFT scheme. The approach is compared to the now commonly used multipole-accelerated algorithms for a variety of structures, and the new method is shown to have substantial performance and memory advantages.

1 Introduction

In the design of high performance integrated circuits and integrated circuit packaging, there are many cases where accurate estimates of the capacitances of complicated three-dimensional structures are important for determining final circuit speeds and functionality. Algorithms using method of moments [1] based discretizations of integral equation formulations are commonly used to compute these capacitances, but such approaches generate dense matrix problems which are computationally expensive to solve, and this limits the complexity of problems which can be analyzed.

In [2], a rapid method, based on the hierarchical multipole algorithm[3], for computing the capacitance of three-dimensional structures was presented. In this paper, we describe a precorrected-FFT approach which can replace the fast multipole algorithm for accelerating the dense matrix-vector product associated with discretized potential integral equations.

2 Problem Formulation

We consider the problem of capacitance extraction for conductors embedded in a homogenous dielectric medium. The capacitance of an m-conductor geometry is summarized by an $m \times m$ symmetric matrix C. To determine the j-th column of the capacitance matrix, one need only solve for the surface charges on each conductor produced by raising conductor j to one volt while grounding the rest.

These m potential problems can be solved using an equivalent free-space formulation in which the conductor-dielectric interfaces are replaced by a charge layer of density σ [4]. The charge layer in the free-space problem will be the induced charge in the original problem if σ satisfies the integral equation

$$\psi(x) = \int_{surfaces} \sigma(x') \frac{1}{4\pi\epsilon_0 \|x - x'\|} da', \qquad x \in surfaces. \tag{1}$$

where $\psi(x)$ is the known conductor surface potential, da' is the incremental conductor surface area, $x, x' \in \mathbf{R}^3$, and $\|x\|$ is the usual Euclidean length of x given by $\sqrt{x_1^2 + x_2^2 + x_3^2}$.

A standard approach to numerically solving (1) for σ is to use a piece-wise constant collocation scheme. The conductor surfaces are broken into n small panels, and it is assumed that on each panel i, a charge, q_i, is uniformly distributed. For each panel, an equation is written which relates the known potential at the center of that i-th panel, denoted $\overline{p}_i$, to the sum of the contributions to that potential from the n charge distributions on all n panels. The result is a dense linear system,

$$Pq = \overline{p} \tag{2}$$

where $P \in \mathbf{R}^{n \times n}$, q is the vector of panel charges, $\overline{p} \in \mathbf{R}^n$ is the vector of known panel potentials, and

$$P_{ij} = \frac{1}{a_j} \int_{panel_j} \frac{1}{4\pi\epsilon_0 \|x_i - x'\|} da', \tag{3}$$

where x_i is the center of the i-th panel and a_j is the area of the j-th panel.

To avoid the $O(n^3)$ cost of solving (2) withe Gaussian elimination, iterative solvers such as GMRES [5] are commonly used. The dominant costs of using GMRES to solve (2) are in calculating the n^2 entries of P using (3) before the iterations begin, and

© 1994 ACM 0-89791-690-5/94/0011/0268 $3.50

performing n^2 operations to compute a matrix-vector product on each iteration. Described below is our precorrected-FFT algorithm which, through the use of carefully applied approximations and transform techniques, avoids forming most of P and reduces the cost of forming the matrix-vector product to order $n \log n$ operations.

3 The Precorrected-FFT Approach

To develop a faster approach to computing the matrix-vector product, consider the parallelepiped which contains a three dimensional problem after it has been discretized into n panels. The parallelepiped containing the problem could then be subdivided into an $k \times l \times m$ 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 somehow approximated to accelerate the computation [3].

One approach to computing distant interactions is to exploit the fact that evaluation points distant from a cube can be accurately computed by representing the given cube's charge distribution using a small number of weighted point charges. If the point charges all lie on a uniform grid, then the Fast Fourier Transform (FFT) can be used to compute the potential at these grid points due to the grid charges. Specifically, Pq may be approximated in order $n \log n$ operations 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 an FFT, (3) interpolate the grid potentials onto the panels, and (4) directly compute nearby interactions. This process is summarized in Figure 1.

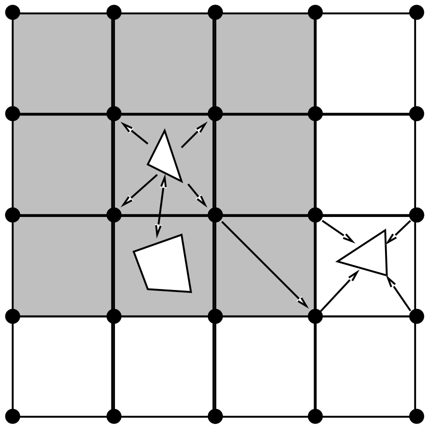


Figure 1: 2-D Pictorial representation of the four steps of the precorrected-FFT algorithm. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid.

3.1 Projecting onto a grid

Consider the cube embedded in the center of a $3 \times 3 \times 3$ array of cubes, and assume that the potential will be evaluated at points exterior to the 27 cube array. Then, since the potential satisfies Laplace's equation, the error in the point charge approximation over the entire exterior can be minimized by minimizing the potential error on the surface of the cube array.

The above observation suggests a scheme for computing the grid charges used to represent charge in a given cube a. First, test points are selected on the surface of the $3 \times 3 \times 3$ cube array which has cube a as its center. Then, potentials due to the grid charges are forced to match the potential due to the cube's actual charge distribution at the test points. Since such collocation equations are linear in the charge distribution, this projection operation which generates a subset of the grid charges, denoted q_a^g, can be represented as a matrix, W_a, operating on a vector representing the panel charges in cube a, q_a.

The sum of grid charges is also required to be equal to the net charge in the cube. This is equivalent to matching the order-zero multipole expansion coefficient, which insures that the error in the calculated potential decays to zero far away from the grid charge cell. The grid projection scheme is summarized in Figure 2. For an alternative approach, based more generally on matching multipole expansion coefficients, see [6].

3.2 Using the FFT

Once the charge has been projected to a grid, computing the potentials at the grid points due to the grid charges is a three-dimensional convolution. We denote this as

$$\psi_g(i,j,k) = \sum_{i',j',k'} h(i-i', j-j', k-k') q_g(i',j',k'). \tag{4}$$

where i,j,k and i',j',k' are triplets specifying the grid points, ψ_g is the vector of grid point potentials, q_g is the vector of grid point charges, and $h(i-i', j-j', k-k')$ is the inverse distance between grid points i,j,k and i',j',k'. As will be made clear below, $h(0,0,0)$ can be arbitrarily defined, and is set

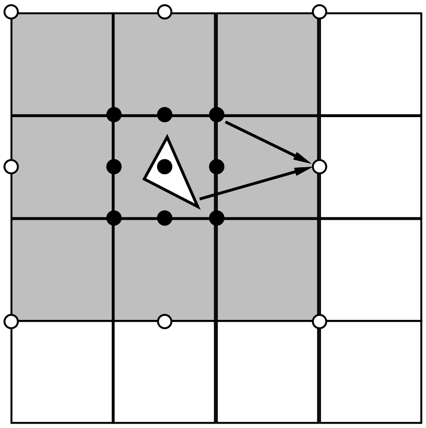


Figure 2: 2-D Pictorial representation of the grid projection scheme. The black points represent the point charges being used to represent the triangular panel's charge density. The white points are the the points where the potential due to the black point charges and the potential due to the triangular panel's charge density are forced to match.

to zero. The above convolution can be computed in $O(N \log N)$ time, where N is the number of grid charges, by using the FFT.

Fast multipole algorithms also effectively create a uniform grid by constructing multipole expansions at the center of each cube, but due to sharing, the point charge approach can be more efficient. For example, a point charge at a cube vertex is used to represent charge in the eight cubes which share that vertex, whereas each multipole coefficient is associated only with charges in a single cube.

Once the grid potentials have been computed, they can be interpolated to the panels in each cube using the transpose of W_a [7]. Therefore, projection, followed by convolution, followed by interpolation, can be represented as

$$\psi_{fft} = W^t H W q, \tag{5}$$

where q is the vector of panel charges, ψ_{fft} is an approximation to the panel potentials, W is the concatenation of the W_a's for each cube, and H is the matrix representing the convolution in (4).

3.3 Precorrecting

In ψ_{fft} of (5), the portions of Pq associated with neighboring cube interactions have already been computed, though this close interaction has been poorly approximated in the projection/interpolation. Before computing a better approximation, it is necessary to remove the contribution of the inaccurate approximation. Denote as $P_{a,b}$ the portion of P associated with the interaction between neighboring cubes a and b, denote the potential at grid points in cube a due to grid charges in cube b as $H_{a,b}$, and denote ψ_a and q_b as the panel potentials and charges in cubes a and b respectively. Then

$$\psi_a = \psi_{a_{fft}} + \left(P_{a,b} - W_a^t H_{a,b} W_b\right) q_b \tag{6}$$

will be a much better approximation to ψ_a.

We define

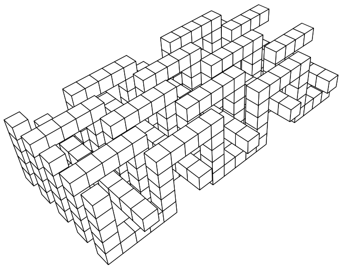
$$P_{a,b}^{cor} \equiv P_{a,b} - W_a^t H_{a,b} W_b \tag{7}$$

to be the "precorrected" direct interaction operator. When used in conjunction with the grid charge representation $P_{a,b}^{cor}$ results in exact calculation of the interactions of panels which are close. Assuming that the Pq product will be computed many times in the inner loop of an iterative algorithm, $P_{a,b}^{cor}$ will be expensive to initially compute, but will cost no more to subsequently apply than $P_{a,b}$.

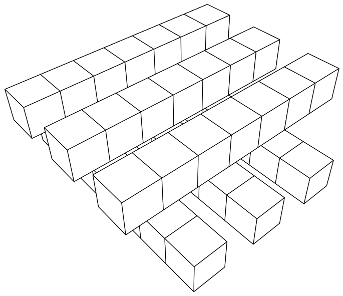
4 Results

In this section we present results comparing the multipole-based FASTCAP program to the precorrected-FFT method for a variety of examples. As is clear from the results in Figure 3, the precorrected-FFT method can be as much as 40% faster and can use as little as one fourth the memory of FASTCAP. We have also presented a figure-of-merit defined as the product of the memory- and speed- advantages of the precorrected-FFT method. It is important to consider this figure because it is often possible to trade memory for speed by changing the size of the grid used in the precorrected-FFT method (this is true of the via structure, for example). In terms of the speed/memory product, the precorrected-FFT method is superior to FASTCAP for all the examples shown, in some cases by more than a factor of six. It should also be noted that since the method is based on the FFT, and not on spherical-harmonics based shifting operators, the approach can also be combined with modified Green's function techniques for handling ground planes or flat dielectric interfaces.

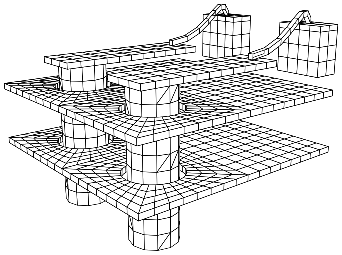
The authors are grateful to Dr. Keith Nabors for making his FASTCAP program available. Many of the experiments were conducted by using substantial portions of the FASTCAP program.



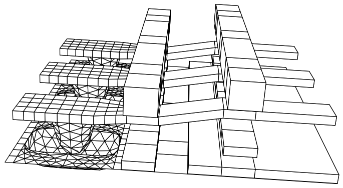
5x5 woven bus			
Panels	Multiply†	Memory‡	Product
9360	0.82	0.42	0.34



3x3 bus crossing			
Panels	Multiply†	Memory‡	Product
11520	0.59	0.26	0.15



via structure			
Panels	Multiply†	Memory‡	Product
6126	2.26	0.37	0.84



RAM cell			
Panels	Multiply†	Memory‡	Product
4881	0.88	0.73	0.64

Figure 3: Comparison of FASTCAP to precorrected-FFT on various examples. Actual discretizations are finer than shown in figures. † Ratio of Grid/FFT CPU time to multipole CPU time. ‡ Ratio of Grid/FFT memory use to multipole memory use.

References

[1] R. F. Harrington, *Field Computation by Moment Methods.* New York: MacMillan, 1968.

[2] K. Nabors, S. Kim, and J. White, "Fast capacitance extraction of general three-dimensional structure," *IEEE Trans. on Microwave Theory and Techniques*, vol. 40, pp. 1496–1507, July 1992.

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

[4] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 21, pp. 76–82, February 1973.

[5] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.

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

[7] A. Brandt, "Multilevel computations of integral transforms and particle interactions with oscillatory kernals," *Computer Physics Communications*, no. 65, pp. 24–38, 1991.