Fast Capacitance Extraction of General Three-Dimensional Structures

Keith Nabors, Songmin Kim, and Jacob White, *Member, IEEE*

Abstract—In [1], a boundary-element based algorithm is presented for computing the capacitance of three-dimensional m-conductor structures whose computational complexity grows nearly as mn, where n is the number of elements used to discretize the conductor surfaces. In that algorithm, a generalized conjugate residual iterative technique is used to solve the $n \times n$ linear system arising from the discretization, and a multipole algorithm is used to compute the iterates. In this paper, several improvements to that algorithm are described which make the approach in [1] applicable and computationally efficient for almost any geometry of conductors in a homogeneous dielectric. In particular, a new adaptive multipole algorithm is described, along with a strategy for accelerating the iterative algorithm by exploiting electrostatic screening. Results using these techniques in a program which computes the capacitance of general three-dimensional structures are presented to demonstrate that the new algorithm is nearly as accurate as the more standard direct factorization approach, and is more than two orders of magnitude faster for large examples.

I. 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 [2] or weighted-residuals [3], [4] based discretizations of integral equation formulations, also known as boundary-element methods [5], 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 [1], a fast algorithm for computing the capacitance of three-dimensional structures of rectangular conductors in a homogeneous dielectric is presented. The method solves the discretized capacitance problem using an iterative technique with iterates computed by a hierarchical multipole algorithm [6], [7]. This general strategy was first suggested in [8]. The computation time for the al-

gorithm was shown to grow nearly as mn, where n is the number of panels used to discretize the conductor surfaces, and m is the number of conductors. In this paper, we describe several improvements to that algorithm and present computational results on a variety of examples to demonstrate that the new method is accurate and can be as much as two orders of magnitude faster than standard direct factorization approaches.

The outline of the paper is as follows. The boundary-element formulation and a standard iterative algorithm for solving the generated matrix problem are briefly reviewed in Section II. A simplified version of the hierarchical multipole algorithm is described in Section III, and our new adaptive multipole algorithm tuned to the boundary-element formulation is given in Section IV. A new preconditioning strategy for accelerating the iterative algorithm, based on the idea of screening, is presented in Section V. Experimental results using our program FASTCAP to analyze a wide variety of structures, made possible by a link to the M.I.T. Micro-Electro-Mechanical Computer Aided Design (MEMCAD) system [9], are presented in Section VI. Finally, conclusions and acknowledgements are given in Section VII.

II. PROBLEM FORMULATION

Capacitance extraction is made tractable by assuming the problem contains conductors embedded in a homogenous dielectric medium, though the techniques described below can be extended to the piecewise-constant dielectric case using the approach in [10]. The capacitance of an m-conductor geometry can then be summarized by an $m \times m$ symmetric matrix C, where the jth column of C has a positive entry C_{jj}, representing the self-capacitance of conductor j, and negative entries C_{ij}, representing coupling between conductors j and i, $i = 1, 2, \cdots, m$, $i \neq j$. To determine the jth 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. Then C_{ij} is numerically equal to the charge on conductor i, $i = 1, 2, \cdots, m$. Repeating this procedure m times gives the m columns of C.

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 [11, 10]. Assuming a homogenous dielectric, the charge layer in the free-space problem will be the induced

charge in the original problem if σ satisfies the integral equation

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

$$x \in \text{surfaces}. \tag{1}$$

where $\psi(x)$ is the known conductor surface potential, da' is the incremental conductor surface area, $x, x' \in R^3$, 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. That is, the conductor surfaces are broken into n small panels or tiles, and it is assumed that on each panel i, a charge, q_i, is uniformly distributed. Then for each panel, an equation is written which relates the known potential at the center of that ith panel, denoted $\bar{p}_i$, to the sum of the contributions to that potential from the n charge distributions on all n panels [10]. The result is a dense linear system,

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

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

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

where x_i is the center of the ith panel and a_j is the area of the jth panel. For such a point-collocation scheme, in general $P_{ij} \neq P_{ji}$, that is P is unsymmetric[1].

The dense linear system of (2) can be solved to compute panel charges from a given set of panel potentials, and the capacitances can be derived by summing the panel charges. If Gaussian elimination is used to solve (2), the number of operations is order n^3. Clearly, this approach becomes computationally intractable if the number of panels exceeds several hundred. Instead, consider solving the linear system (2) using a conjugate-residual style iterative method like GMRES [12]. Such methods have the form given below:

Algorithm 1: GMRES algorithm for solving (2)
Make an initial guess to the solution, q_0.
Set $k = 0$.
do {
    Compute the residual, $r^k = \bar{p} - Pq^k$.
    if $\|r^k\| < tol$, return q^k as the solution.
    else {
        Choose α's and β in
            $q^{k+1} = \sum_{j=0}^{k} \alpha_j q^j + \beta r^k$
        to minimize $\|r^{k+1}\|$.
        Set $k = k + 1$.
    }
}

[1]If a Galerkin scheme rather than point collocation is used [11], the resulting P matrix will be symmetric, but computing the P_{ij}'s will require evaluating a more complicated integral.

The dominant costs of Algorithm 1 are in calculating the n^2 entries of P using (3) before the iterations begin, and performing n^2 operations to compute Pq^k on each iteration. Described below is our adaptive hierarchical multipole algorithm which, through the use of carefully applied approximations, avoids forming most of P and reduces the cost of forming Pq^k to order n operations. This does not necessarily imply that each iteration of the GMRES algorithm can be computed with order n operations. If the number of GMRES iterations required to achieve convergence is order n, then to perform the minimization in each iteration will require order n^2 operations. For this reason, it is important to insure that the number of GMRES iterations required to achieve convergence remains bounded. A preconditioning scheme is described below which accomplishes this, at least for all the examples tested, by reducing the number of GMRES iterations required to achieve one percent accuracy to fewer than 30, even for problems where n is as large as 10 000.

III. THE HIERARCHICAL MULTIPOLE ALGORITHM

A complete description of the hierarchical multipole algorithm is not given here; the original description is in [6], [7], and its application to capacitance extraction is described in [1]. Instead we describe the expansion approximation and examine a simplified two-dimensional example which both exhibits the method's salient features, and motivates the adaptive algorithm and the preconditioner described in subsequent sections.
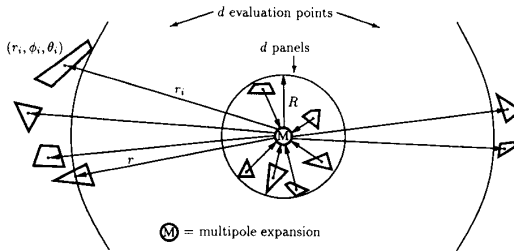
A. Multipole Expansions

Multipole expansions are often used to approximate the far field due to a confined charge distribution [13]. For example, consider evaluating the potential p_i at the center of a panel i, (r_i, ϕ_i, θ_i), due to a collection of d distant panels, as in Fig. 1. The potential due to the surface charges on those d panels is given approximately by the truncated multipole expansion

$$\psi(r_i, \phi_i, \theta_i) \approx \sum_{n=0}^{l} \sum_{m=-n}^{n} \frac{M_n^m}{r_i^{n+1}} Y_n^m(\phi_i, \theta_i), \tag{4}$$

where the spherical coordinates of the evaluation location are measured relative to the origin of the multipole expansion, $Y_n^m(\phi_i, \theta_i)$ are the surface spherical harmonics, M_n^m are the multipole coefficients determined from the panel charges, and l is the expansion order.

Given the multipole coefficients, the same multipole expansion can be used to quickly, but approximately, evaluate the potential at many panel centers. For example, in Fig. 1, there are d charged panels, and d panel centers where the potential must be evaluated. A direct calculation of those potentials requires order d^2 operations, but only order d operations are needed if the multipole expansion is used (assuming the expansion order l is fixed).

Fig. 1. The evaluation of the potential at (r_i, ϕ_i, θ_i).

In the Fig. 1 case, the error due to truncating the multipole expansion is bounded [7], as in

$$\left| \psi(r_i, \phi_i, \theta_i) - \sum_{n=0}^{l} \sum_{m=-n}^{n} \frac{M_n^m}{r_i^{n+1}} Y_n^m(\phi_i, \theta_i) \right|$$

$$\leq K_1 \left(\frac{R}{r_i}\right)^{l+1} \leq K_1 \left(\frac{R}{r}\right)^{l+1} \qquad (5)$$

The quantities r and R are as in Fig. 1 and K_1 is a constant independent of the multipole expansion order, l. The bound indicates that the multipole potential evaluations converge more rapidly with expansion order as the minimum distance between the panel charges and the evaluation points increases.

In order to ensure that the error bound in (5) tightens sufficiently with each increase in expansion order l, the hierarchical multipole algorithm uses a multipole expansion to represent the effect of charge in a region only if the radius of the region, R, is less than half the distance between the region's center and the evaluation point, denoted r. For example, in Fig. 2 two groups of panels are represented by a multipole expansions of order l, and by the above criteria, both can be used to evaluate the potential at panel i's center, as $R/r = 3R/3r < 0.5$.

B. A Two-Dimensional Example

The aggregation of distant panels into multipole expansions which can be used to evaluate potentials at many panel centers is the source of the hierarchical multipole algorithm's efficiency. Maintaining this efficiency for general distributions of panels while controlling error is ensured by exploiting a hierarchical partitioning of the problem domain, the smallest cube containing all the conductors.
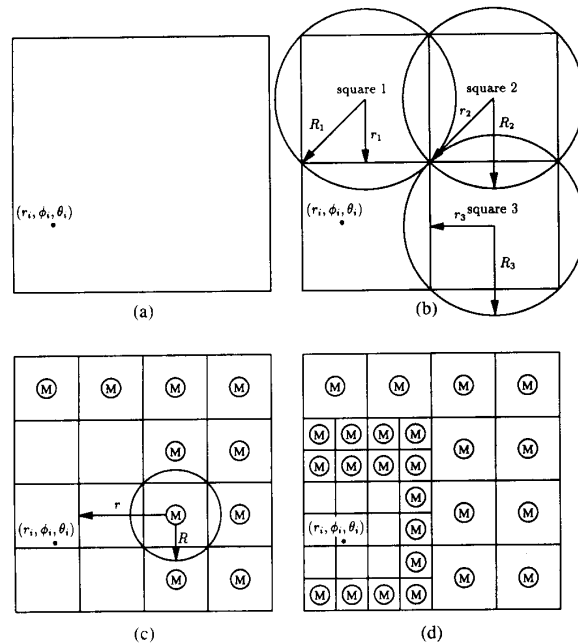
Consider, for example, evaluating the potential at some point (r_i, ϕ_i, θ_i) in Fig. 3 due to panel charges inside the illustrated problem domain. A first partitioning would be to break the problem into four smaller squares, leaving (r_i, ϕ_i, θ_i) somewhere in the lower left square (Fig. 3(b)[2]. To ensure that the errors due to truncating the multipole expansion shrink rapidly with increasing expansion order, multipole expansions will not be used to represent the

[2]In the three-dimensional problem, the equivalent partitioning would be to divide a cube into eight smaller cubes.
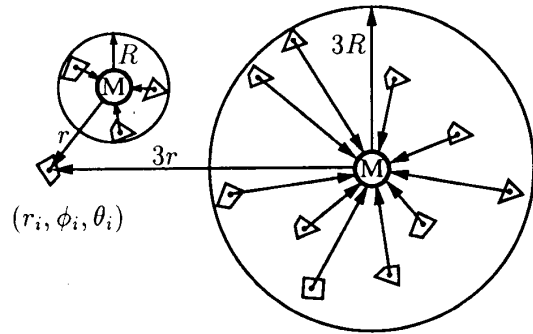
$\textcircled{M}$ = multipole expansion

Fig. 2. The evaluation of two multipole expansions with the same error bound.

Fig. 3. The evaluation of the potential at (r_i, ϕ_i, θ_i).

charges in squares 1, 2, and 3, when evaluating the potential at points in the lower-left square, because R_1/r_1, R_2/r_2 and R_3/r_3 in Fig. 3(b) are all greater than 0.5. For the particular example evaluation point in the lower-left square, the charge in square 2 is distant enough to satisfy the criteria for using multipole expansions. However, a more detailed study of the hierarchical multipole algorithm than we will consider here would show that it is not efficient to exploit such special cases.

Squares 1, 2, and 3 are each divided into four squares, as in Fig. 3(c), to produce smaller regions which can possibly satisfy the criteria for representation by a multipole expansion. In fact, many of the smaller squares do satisfy the criteria, as can be seen by examining the illustrated case, for which R/r is less than 0.5. Thus, at the end of

this partitioning step, all the charges in the squares marked with an M in Fig. 3(c) will be represented with a multipole expansion when evaluating the potential at points in the square containing (r_i, ϕ_i, θ_i).

In order for the multipole expansions to be used to represent the potential due to panel charges contained in the unmarked squares of Fig. 3(c), these squares are partitioned further, as in Fig. 3(d). Then, as before, the distance criteria implies that multipole expansions can be used to represent the panel charges in all but a few squares near the square containing the evaluation point. If it is determined not to partition any further than is indicated in Fig. 3(d), the potential p_i, at (r_i, ϕ_i, θ_i), can then be computed by summing a "near" or direct term and a "far" or multipole term. That is, the "near" contribution to p_i is due to panel charges in the nine unmarked squares in Fig. 3(d), and is computed directly from $P_{ij} q_j$ products. The "far" contribution to p_i is due to distant panel charges and is determined by evaluating the 25 multipole expansions indicated in Fig. 3(d). In the next section, we will refer to the list of squares associated with those 25 multipole expansions as the *multipole list* for the square containing (r_i, ϕ_i, θ_i).

In general, the number of partitioning levels, L, for a given problem domain is selected so that the squares on the finest level each have no more than k panels (typically k is of the order of ten). Then for a uniform distribution of panels, the number of partitioning levels will be given by $L = \log(n/k)$. Since the number of multipole expansions on each partitioning level which contribute to p_i is bounded by a constant, each potential evaluation involves order $\log n$ multipole expansion evaluations. Also, since each lowest level square has no more than k panels, the direct contribution to p_i is bounded by a constant. Therefore, as evaluating the entire potential vector requires n evaluations of this type, the above multipole approach is an order $n \log n$ algorithm for computing an approximation to Pq.[3] The hierarchical multipole algorithm given in [7], and used in the FASTCAP program mentioned below, is more sophisticated than the above description suggests. In particular, multipole evaluations are efficiently combined into local expansions in such a way as to reduce the number of operations to order n. However, for purposes of describing the adaptive algorithm and the preconditioning techniques below, the simplified algorithm above is sufficiently detailed.

IV. BOUNDARY-ELEMENT ORIENTED ADAPTIVE CALCULATION

In general, when representing the charge in a region by a multipole expansion, the coefficients M_n^m in (4) are determined from the charge density, $q(r, \phi, \theta)$, as

$$M_n^m = \int_{\text{region}} q(r, \phi, \theta) r^n Y_n^{-m}(\phi, \theta) \, da'. \tag{6}$$

[3]The analysis is similar for the three-dimensional case. The primary difference is that in the three-dimensional case space is partitioned into cubes, and when cubes are subpartitioned, they generate eight smaller cubes.

In this section, computing the multipole expansion coefficients for panel charges is examined in more detail, and is shown to lead naturally to an adaptive multipole algorithm.

A. Computing the Multipole Expansions

Returning to Fig. 1, the potential at the Cartesian equivalent of (r_i, ϕ_i, θ_i), (x_i, y_i, z_i), due to d distant panels may be approximated with a zeroth-order multipole expansion, which is equivalent to computing the potential due to a single charge equal to the sum of the d panel charges, and located at the center of the smallest ball enclosing the panels,

$$\psi(x_i, y_i, z_i) \approx \overline{M}_0^0 \frac{1}{r_i}. \tag{7}$$

Here (x_i, y_i, z_i) is the ith panel's center point relative to the center of the smallest ball enclosing the distant panels, $r_i \triangleq \sqrt{x_i^2 + y_i^2 + z_i^2}$, and

$$\overline{M}_0^0 \triangleq \sum_{j=1}^{d} q_j. \tag{8}$$

The approximation

$$\psi_0(x_i, y_i, z_i) \triangleq \overline{M}_0^0 \frac{1}{r_i} \tag{9}$$

is the zeroth-order multipole expansion for the potential due to the distant panels. For accuracy reasons, higher order expansions are typically used. For example, the first-order multipole expansion for ψ is

$$\psi(x_i, y_i, z_i) \approx \psi_0(x_i, y_i, z_i) + \psi_1(x_i, y_i, z_i), \tag{10}$$

with

$$\psi_1(x_i, y_i, z_i) \triangleq \overline{M}_1^0 \frac{z_i}{r_i^3} - \overline{M}_1^1 \frac{x_i}{2r_i^3} - \tilde{M}_1^1 \frac{y_i}{2r_i^3}, \tag{11}$$

where, for panels on which the charge is assumed uniformly distributed,

$$\overline{M}_1^0 \triangleq \sum_{j=1}^{d} \frac{q_j}{a_j} \int_{\text{panel}_j} z' \, da'; \tag{12}$$

$$\overline{M}_1^1 \triangleq -2 \sum_{j=1}^{d} \frac{q_j}{a_j} \int_{\text{panel}_j} x' \, da'; \tag{13}$$

$$\tilde{M}_1^1 \triangleq -2 \sum_{j=1}^{d} \frac{q_j}{a_j} \int_{\text{panel}_j} y' \, da'. \tag{14}$$

The added potential ψ_1 is the field due to a single dipole aligned along the vector

$$(x, y, z) = (\overline{M}_1^0, -\overline{M}_1^1/2, -\tilde{M}_1^1/2). \tag{15}$$

In general, the lth order multipole expansion in (4) can be rewritten in the form

$$\psi(x_i, y_i, z_i) \approx \sum_{n=0}^{l} \psi_n(x_i, y_i, z_i), \tag{16}$$

with each ψ_n corresponding to the potential due to a 2^n-pole charge constellation.

B. The Adaptive Algorithm

The simple multipole algorithm discussed in Section III uses multipole expansions to represent potentials due to panel charges inside cubes which are far enough away from the evaluation point. The efficiency of this procedure depends on the number of panels in the cubes. Consider, for example, a cube containing three panels whose potential is to be approximated using a first-order multipole expansion of the form (16),

$$\psi(x_i, y_i, z_i) \approx \overline{M}_0^0 \frac{1}{r_i} + \overline{M}_1^0 \frac{z_i}{r_i^3} - \overline{M}_1^1 \frac{x_i}{2r_i^3} - \tilde{M}_1^1 \frac{y_i}{2r_i^3}.$$

(17)

The exact potential due to the cube's panels has the form

$$\psi(x_i, y_i, z_i) = P_{i1}q_1 + P_{i2}q_2 + P_{i3}q_3,$$ (18)

assuming the three panels are numbered 1, 2, and 3. In the capacitance calculation, the geometry-dependent quantities in (17) and (18) are calculated once and stored for repeated use in computing the iterates of Algorithm 1. Thus evaluating (17) involves multiplying four, fixed, geometry-dependent quantities

$$\frac{1}{r_i}, \quad \frac{z_i}{r_i^3}, \quad \frac{x_i}{2r_i^3}, \quad \frac{y_i}{2r_i^3}$$

(19)

by the charge-dependent multipole coefficients $\overline{M}_0^0$, $\overline{M}_1^0$, $\overline{M}_1^1$ and $\tilde{M}_1^1$, while (18) involves computing only the three products, multiplying the geometric quantities P_{ij}, $j = 1$, $\cdots$, 3, by the corresponding charges. Therefore, for this case it will be more efficient to evaluate the potential due to panels in this cube using $P_{ij}q_j$ products rather than by evaluating multipole expansions.

An adaptive multipole algorithm can be derived from the simplified approach described in Section III if the potential due to panel charges in a cube is always evaluated directly, rather than with a multipole approximation, whenever the number of expansion coefficients would exceed the number of panels. A more precise definition of the computational procedure is given in Algorithm 2 below, which uses some notation which we now introduce.

The cube which contains the entire problem domain is referred to as the level 0 cube. If the volume of the cube is divided into eight equally sized child cubes, referred to as level 1 cubes, then each has the level 0 cube as its parent. The panels are distributed among the child cubes by associating a panel with a cube if the panel's center point is contained in the cube. This process can be repeated to produce L levels of cubes, and L partitionings of panels starting with an 8-way partitioning and ending with an 8^L-way partitioning. The number of levels, L, is chosen so that the maximum number of panels in any finest, or Lth, level cube is less than some threshold (nine is a typical default). A neighbor of a given cube is defined as any cube which shares a corner with the given cube or

shares a corner with a cube which shares a corner with the given cube (note that a cube has a maximum of 124 neighbors). Finally, in the algorithm below it is assumed that for each finest-level cube, a multipole list has been constructed using a recursive approach similar to that given in the two-dimensional example above.

Algorithm 2: Adaptive Algorithm for Computing $p = Pq$.

  Comment: Compute the potential due to nearby charges directly.
    For each finest-level cube $i = 1$ to 8^L {
      For each panel j in finest-level cube i {
        Set $p_j = 0$.
        For each panel k in cube i or its neighbors {
          Add $P_{jk}q_k$ to p_j.
        }
      }
    }

  Comment: Compute the multipole coefficients from the charge vector q.
  Comment: *order* is the order of the multipole expansion (typically 2).
    For each level $j = L$ to 2 {
      For each level j cube $i = 1$ to 8^j {
        If cube i contains more than $(order + 1)^2$ panels {
          Compute the multipole coefficients for cube i using panel charges and/or coefficients of child cube multipole expansions.
        }
      }
    }

  Comment: Compute the potential due to distant panels.
    For each finest-level cube $i = 1$ to 8^L {
      For each cube j in cube i's multipole list {
        If cube j contains more than $(order + 1)^2$ panels {
          For each panel k in cube i {
            Evaluate the multipole expansion for cube j at (x_k, y_k, z_k) and add to p_k.
          }
        }
        Else {
          For each panel k in cube i {
            For each panel l in cube j {
              Add $P_{kl}q_l$ to p_k.
            }
          }
        }
      }
    }

It should be again noted that, as mentioned at the end of Section III, Algorithm 2 is a simplified version of the
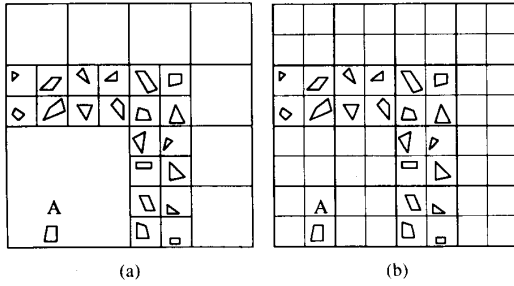
Fig. 4. Partitionings used in Algorithm 2, (b), and by the adaptive algorithm of [14], (a).

hierarchical multipole algorithm [14], [7] used in FAST-CAP [1]. The complete algorithm also adaptively applies local expansions to improve the efficiency of the process of gathering together multipole expansions.

C. Comparison to Previous Work

The above approach to making the multipole algorithm adaptive is specialized to the boundary-element problem. A more general, but not as efficient, approach would be to extend to three-dimensions the two-dimensional adaptive algorithm described in [14]. In this earlier work, the multipole algorithm is made adaptive by breaking up the problem domain nonuniformly, in which case lowest level squares are of different sizes, where the sizes are chosen so that each lowest-level square has roughly the same number of panels.

To see why an adaptive algorithm based on a nonuniform partitioning of the problem domain can be inefficient, consider computing the potential at the center of the panel labeled A in Figure 4(a). For this example, a two-dimensional square problem domain has been recursively quartered, the recursion being halted when a square had no more than one panel. To compute the potential at the center of panel A using a multipole algorithm based on this nonuniform partitioning requires: nine direct potential evaluations, for the nine panels in nearest-neighbor squares bordering the square containing A, and eleven multipole evaluations for the eleven nonempty squares not bordering A. Alternatively, if a uniform partitioning is used, as in Fig. 4(b), then it is easily seen that only *five* multipole evaluations and no direct evaluations are required.

Such an increase in computational cost can not occur with the adaptive algorithm given above, as the following theorem states:

Theorem 1: The computational cost of the adaptive approach given in Algorithm 2 is never greater than that of the corresponding nonadaptive algorithm.

The proof follows directly from the fact that the adaptive multipole algorithm, Algorithm 2, evaluates the potential due to panel charges in a cube directly, rather than with a multipole approximation, whenever the number of expansion coefficients would exceed the number of panels.

It should be noted that the overhead cost of maintaining

a uniform domain partitioning has been ignored in this comparison. If the distribution of panels is very nonuniform, a matching nonuniform partitioning, like the approach used in [14], will naturally generate fewer lowest-level partitions, and hence reduce bookkeeping overhead. Careful data organization can make this overhead negligible for typical capacitance calculation problems. In the examples presented in Section VI, for which FASTCAP generates uniform domain partitionings with as many as 250 000 lowest-level cubes, the bookkeeping overhead is insignificant.

V. PRECONDITIONING THE ITERATIVE METHOD

In general, the GMRES iterative method applied to solving (2) can be significantly accelerated by *preconditioning* if there is an easily computed good approximation to the inverse of P. We denote the approximation to P^{-1} by $\bar{C}$, in which case preconditioning the GMRES algorithm is equivalent to using GMRES to solve

$$P\bar{C}x = \bar{p}. \tag{20}$$

for the unknown vector x, from which the charge density is computed by $q = \bar{C}x$. Clearly, if $\bar{C}$ is precisely P^{-1}, then (20) is trivial to solve, but then $\bar{C}$ will be very expensive to compute.

A. A Simple Example

A good approximation to P^{-1} that is easily computed, and fits with the hierarchical multipole algorithm described previously, can be derived by exploiting the fact that P^{-1} is approximately the *detailed* capacitance matrix, by which we mean the $n \times n$ capacitance matrix for the problem in which every panel or tile used to represent the conductor surfaces is treated as an independent conductor. To see why this point of view leads to a preconditioner, consider the 7×7 P matrix, denoted $P(7)$, for the seven panel example in Fig. 5(a). The fourth row of $P(7)$, which is associated with the center panel in Fig. 5(a), can be computed using the definition in (3) and is, in inverse-farads,

$$
\begin{array}{cccc}
P(7)_{4,1} & P(7)_{4,2} & P(7)_{4,3} & P(7)_{4,4} \\
0.5785 & 0.8346 & 1.4261 & 3.1686 \\
P(7)_{4,5} & P(7)_{4,6} & (7)_{4,7} \\
1.4261 & 0.8346 & 0.5785
\end{array}
\tag{21}
$$

where all the values have been multiplied by 10^{-10}. The fourth row of $P(7)^{-1}$ is, in picofarads,

$$
\begin{array}{cccc}
P(7)_{4,1}^{-1} & P(7)_{4,2}^{-1} & P(7)_{4,3}^{-1} & P(7)_{4,4}^{-1} \\
-1.3080 & -1.5898 & -15.4544 & 46.7864 \\
P(7)_{4,5}^{-1} & P(7)_{4,6}^{1} & P(7)_{4,7}^{-1} \\
-15.4544 & -1.5898 & -1.3080
\end{array}
\tag{22}
$$

From the definition in (3), the matrix elements $P_{4,j}$ must decay as $|j - 4|$ grows, but notice that the terms in $P_{4,j}^{-1}$
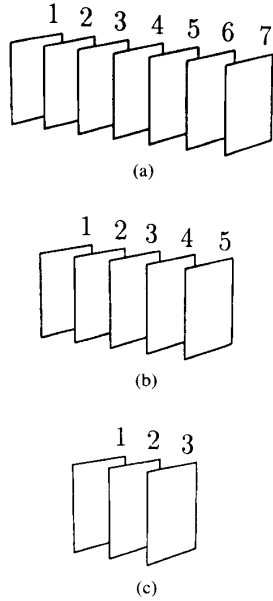
Fig. 5. Simple panel systems with potential coefficient matrices $P(7)$, (a), $P(5)$, (b), and $P(3)$, (c). The parallel, 1 m × 1 m panels are spaced 0.5 m appart.

decay *much faster* as $|j - 4|$ increases. Viewing P^{-1} as an approximation to the detailed capacitance matrix makes clear that the very fast decay of terms in $P(7)^{-1}$ is just an example of classical electrostatic screening. The effect of screening can also be seen by examining the row of $P(5)^{-1}$ associated with the center panel in the five panel problem in Fig. 5(b) and in the row of $P(3)^{-1}$ associated with the center panel in the three panel problem in Fig. 5(c), which are, in picofarads,

$$P(5)_{3,1}^{-1} \quad P(5)_{3,2}^{-1} \quad P(5)_{3,3}^{-1} \quad P(5)_{3,4}^{-1} \quad P(5)_{3,5}^{-1}$$

$$-2.1593 \quad -15.5547 \quad 46.6990 \quad -15.5547 \quad -2.1593$$
$$\tag{23}$$

and

$$P(3)_{2,1}^{-1} \quad P(3)_{2,2}^{-1} \quad P(3)_{2,3}^{-1} \tag{24}$$

$$-16.5499 \quad 46.4573 \quad -16.5499$$

respectively. Comparing (22), (23), and (24) leads to the observation that a good estimate for the fourth row of $P(7)^{-1}$ can be derived from the third row of $P(5)^{-1}$, which is a *smaller* problem, and that even the second row of $P(3)^{-1}$ provides a reasonable estimate. Specifically, the estimate based on the five-panel problem is

$$\tilde{C}_{4,j} = \begin{cases} P(5)_{3,3+(j-4)}^{-1}, & |j - 4| < 3; \\ 0, & \text{otherwise}; \end{cases} \tag{25}$$

where $\tilde{C}$ denotes the estimate to $P(7)^{-1}$.

B. *Preconditioning Algorithm*

The above example suggests an approach to estimating P^{-1} for a general configuration of panels which fits with

the hierarchical multipole algorithm in that the preconditioner $\tilde{C}$ can be constructed and applied in a cube-by-cube fashion. The preconditioner is formed by inverting a sequence of reduced P matrices, one associated with each finest-level cube, as in Algorithm 3 below.

Algorithm 3: Forming $\tilde{C}$.
    For each finest-level cube $i = 1$ to 8^L {
        Form P^i, the potential coefficient matrix for the reduced problem considering only the panels contained in cube i and cube i's neighbors.
        Compute $\tilde{C}^i = (P^i)^{-1}$.
        For each panel k in cube i or cube i's neighbors
        {
            if panel k is not in cube i {
                delete row k from $\tilde{C}^i$.
            }
        }
    }

Note that $\tilde{C}^i$ is not a square matrix and that

$$\sum_{i=1}^{8L} (\# \text{ rows in } \tilde{C}^i) = n \tag{26}$$

where again n is the total number of panels. By comparing Algorithm 3 with Algorithm 2, it is clear that P^i uses only those elements of the full P matrix which are already required in Algorithm 2, and therefore the computational cost in computing the preconditioner is only in inverting small P^i matrices. Then computing the product $P\tilde{C}x^k$, which would be used in a GMRES algorithm applied to solving (20), is accomplished in two steps. First, the preconditioner is applied to form $q^k = \tilde{C}x^k$ using Algorithm 4 below. Then, Pq^k is computed using Algorithm 2 in the previous section[4].

Algorithm 4: Forming $q = \tilde{C}x$.
    For each finest-level cube $i = 1$ to 8^L {
        For each panel j in finest-level cube i {
            For each panel k in cube i or its neighbors {
                Add $\tilde{C}_{jk}^i x_k$ to q_j.
            }
        }
    }

VI. Experimental Results

In this section, results from computational experiments are presented to demonstrate the efficiency and accuracy of the preconditioned, adaptive, multipole-accelerated (PAMA) 3-D capacitance extraction algorithm described above [16]. In particular, the program FASTCAP, which can use both direct factorization and multipole-accelerated techniques, has been developed and incorporated into MIT's MEMCAD (Micro-Electrical-Mechanical Computer-Aided Design) system [9]. The structures described

---

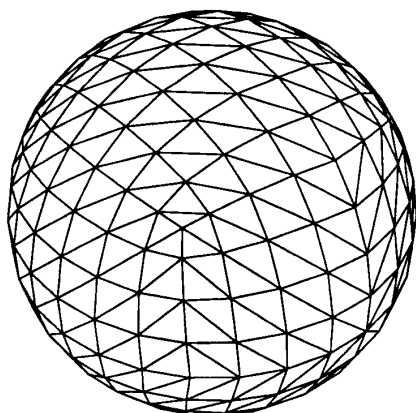[4]A similar approach in a somewhat different context was suggested in [15].

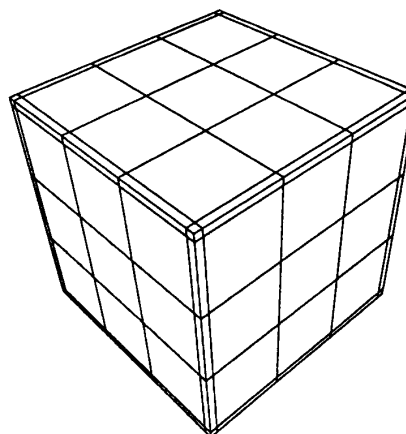Fig. 6. The sphere1 discretization of the unit sphere.

Fig. 7. The cube1 discretization of the unit cube.

below were created with the solid modeling program in the MEMCAD system, PATRAN, or by computer program, and all capacitance calculations were performed using FASTCAP. The multipole-accelerated algorithms in FASTCAP use, by default, second-order multipole expansions and a GMRES convergence tolerance (see Algorithm 1) of 0.01.

To demonstrate absolute accuracy, the FASTCAP program was used to compute the capacitance of a unit sphere, discretized as in Fig. 6, and a unit cube, discretized as in Fig. 7. In Table I, the capacitances computed using the PAMA algorithm are compared with the capacitances computed using direct factorization of P in (2) (Direct), and with analytic results for the unit sphere and with reference results for the unit cube. As can be seen from the table, the results using the PAMA algorithm are easily within one percent of the analytic or reference results.

The PAMA algorithm is nearly as accurate as the direct factorization method even on more complex problems, such as the 2 × 2 woven bus structure in Fig. 8. The capacitances computed using the two methods are compared in Table II, using coarse, medium, and fine discretizations of the woven bus structure, also shown in Fig. 8. Note that the coupling capacitance C_{12} between conductors one and two, which is forty-times smaller than the self-capacitance C_{11}, is computed nearly as accurately with the PAMA algorithm as with direct factorization.

The computational cost of using the FASTCAP program is roughly proportional to the product of the number of conductors, m, and the number of panels n. This is experimentally verified by computing the capacitances of the 2 × 2 woven bus structure in Fig. 8, with progressively finer discretizations. In Fig. 9, the execution times required to compute these capacitances are plotted as a function of mn, and as the graph demonstrates, the execution time does grow nearly linearly.

To demonstrate the effectiveness of various aspects of the PAMA algorithm on a range of problems, in Table III the execution times required to compute the capacitances

TABLE I
CAPACITANCE VALUES (IN pF) ILLUSTRATING FASTCAP's ACCURACY (†BY ANALYTIC CALCULATION; †FROM [11], [17])

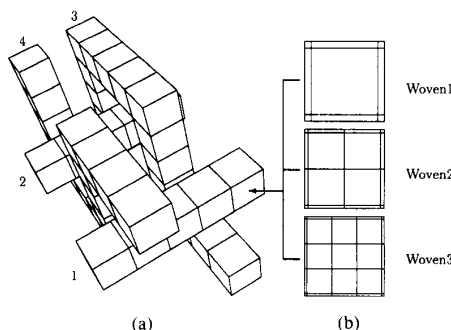|	Problem	
Method	Sphere1 768 Panels	Cube1 150 Panels
Direct	110.6	73.26
PAMA	110.5	73.28
Other	111†	73.5, 73.4†

Fig. 8. The 2 × 2 woven bus problem: bars have 1 m × 1 m cross sections. The three discretizations are obtained by replacing each square face in (a) with the corresponding set of panels in (b).

TABLE II
CAPACITANCE VALUES (IN pF) ILLUSTRATING FASTCAP's ACCURACY FOR THE COMPLICATED GEOMETRY OF FIG. 8.

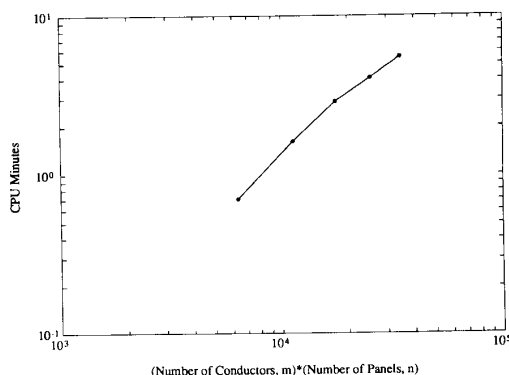|	Problem					
	Woven1 1584 Panels		Woven2 2816 Panels		Woven3 4400 Panels	
Method	C_{11}	C_{12}	C_{11}	C_{12}	C_{11}	C_{12}
Direct	251.6	−6.353	253.2	−6.446	253.7	−6.467
PAMA	251.8	−6.246	253.3	−6.334	253.9	−6.377

(Number of Conductors, m)*(Number of Panels, n)

Fig. 9. Executione time as a function of mn for the MA, AMA, and PAMA algorithms applied solving to progressively finer discretizations of the 2 × 2 woven bus problem.

TABLE III
CPU TIMES IN MINUTES ON AN IBM RS6000/540 (TIMES IN PARENTHESES ARE EXTRAPOLATED)

Method	Cube2 294 Panels	Sphere2 1200 Panels	2 × 2 Woven Bus 4400 Panels	Via 6185 Panels	Diaphragm 7488 Panels	5 × 5 Woven Bus 9630 Panels
Direct	0.11	3.2	185	(490)	(890)	(1920)
MA	0.06	0.3	6.0	11	8.7	42
AMA	0.05	0.2	3.3	4.7	5.9	23
PAMA	0.05	0.2	2.3	3.2	1.3	11

of six different examples using four different methods are given. The examples Cube2 and Sphere2 are finer discretizations of the unit cube and sphere in Figs. 6 and 7; the examples 2 × 2 Woven Bus and 5 × 5 Woven Bus are described above; the example Via, shown in Fig. 10, models a pair of connections between integrated circuit pins and a chip-carrier; and the example Diaphragm, shown in Fig. 11, is a model for a microsensor [18].

From Table III, it can be seen that using the adaptive multipole algorithm (AMA) improves FASTCAP's execution time by a factor of two over using the multipole algorithm (MA) alone, and that using the preconditioner can reduce the execution time by nearly a factor of five. The improvement due to the adaptive algorithm is small because it is being compared to our MA algorithm [1], which ignores empty cubes, and is therefore already somewhat adaptive. Exploiting empty cubes is trivial to implement, and makes an enormous difference. For the largest problem, the 5 × 5 Woven Bus, more than 252 000 out of 262 000 cubes used to partition the problem domain are empty. A truly, nonadaptive multipole algorithm would therefore be *twenty-five* times slower than the our MA algorithm.

The reduction in execution time afforded by the adaptive algorithm (AMA) over the normal multipole algorithm (MA) stems from more efficiently computing the Pq product in each iteration of the GMRES algorithm; and using the preconditioner reduces execution time by reducing the number of iterations required to achieve convergence. Because of various program overheads, comparing
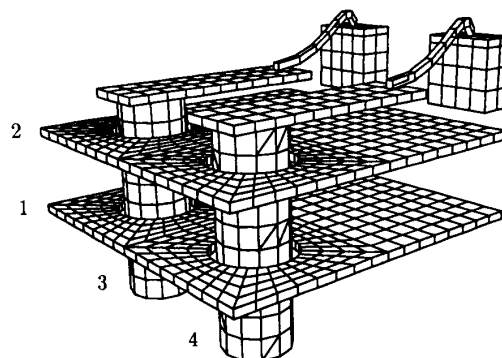

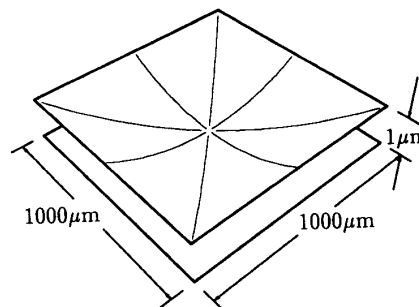Fig. 10. Two signal lines passing through conducting planes; via centers are 2 mm appart.

Fig. 11. A schematic illustration of the diaphragm problem. The two plates are 0.02 μm appart at the center.
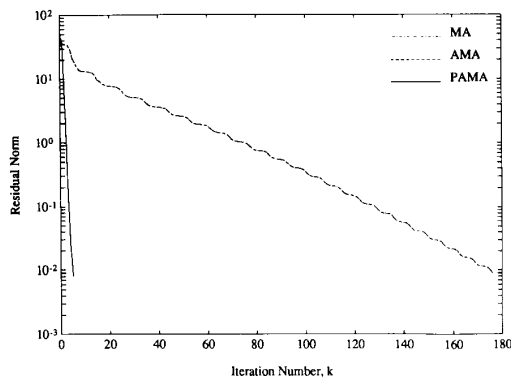
Fig. 12. The norm of the residual, $\| \bar{p} - Pq^k \|$, as a function of iteration for the MA, AMA, and PAMA algorithms applied to solving the Diaphragm problem.

total execution times can hide the sometimes dramatic effect the preconditioner can have on GMRES convergence. To show the impact of the preconditioner more directly, in Fig. 12 we plot the norm of the residual, $\| \bar{p} - Pq^k \|$, as a function of iteration for the MA, AMA, and PAMA algorithms applied to solving the Diaphragm problem. As is evident in the figure, the MA and AMA algorithms converge nearly identically, as expected, but the residuals computed with the PAMA algorithm drop considerably faster. It is this rapid convergence that easily offsets the fact that one PAMA iterate requires slightly more computation than an AMA iterate.

VII. CONCLUSIONS

In this paper several new algorithms were presented that make multipole-accelerated three-dimensional capacitance calculation applicable and computationally efficient for almost any geometry of conductors in a homogeneous dielectric medium. In particular, a new adaptive multipole algorithm was described, along with a strategy for accelerating iterative algorithm convergence by exploiting electrostatic screening. Results from using FAST-CAP, our program based on these techniques, to compute the capacitance of a wide range of examples were given, and they demonstrated that the new algorithms are nearly as accurate as the more standard direct factorization approach, and are more than two orders of magnitude faster for large examples. Current research in progress is in extending the above approach to solving problems with piecewise-constant dielectrics and ground planes.

REFERENCES

[1] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, pp. 1447–1459, Nov. 1991.
[2] R. F. Harrington, *Field Computation by Moment Methods*. New York: MacMillan, 1968.
[3] S. H. Crandall, *Engineering Analysis*. New York: McGraw-Hill, 1956.
[4] L. Collatz, *The Numerical Treatment of Differential Equations*. New York: Springer-Verlag, third ed., 1966.
[5] C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, *Boundary Element Techniques*. Berlin: Springer-Verlag, 1984.
[6] L. Greengard and V. Rokhlin, "A fast algorithm for particle simulations," *J. Computational Physics*, vol. 73, pp. 325–348, Dec. 1987.
[7] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, Cambridge, MA: M.I.T. Press, 1988.
[8] V. Rokhlin, "Rapid solution of integral equations of classical potential theory," *J. Computational Physics*, vol. 60, pp. 187–207, Sept. 15, 1985.
[9] F. Maseeh, R. M. Harris, and S. D. Senturia, "A CAD architecture for microelectromechanical systems," in *IEEE Micro Electro Mechanical Systems Conf. Rec.*, Napa Valley, CA, 1990, pp. 44–49.
[10] S. M. Rao, T. K. Sarkar, and R. F. Harrington, "The electrostatic field of conducting bodies in multiple dielectric media," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 1441–1448, Nov. 1984.
[11] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. 21, pp. 76–82, Feb. 1973.
[12] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.
[13] J. D. Jackson, *Classical Electrodynamics*, New York: Wiley, second ed., 1975.
[14] J. Carrier, L. Greengard, and V. Rokhlin, "A fast adaptive multipole algorithm for particle simulations," *SIAM J. Scientific and Statistical Computing*, vol. 9, pp. 669–686, July 1988.
[15] S. A. Vavasis, "Preconditioning for boundary integral equations (preliminary version)," tech. rep., Cornell University, Ithaca, NY, Mar. 1990.
[16] K. Nabors, S. Kim, J. White, and S. Senturia, "Fast capacitance extraction of general three-dimensional structures," in *Proc. 1991 IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors*, Oct. 1991, pp. 479–484.
[17] M. A. Jaswon and G. T. Symm, *Integral Equation Methods in Potential Theory and Elastostatics*. London: Academic Press, 1977.
[18] B. P. Johnson, S. Kim, S. D. Senturia, and J. White, "MEMCAD capacitance calculations for mechanically deformed square diaphragm and beam microstructures," in *Proc. of Transducers '91*, June 1991.
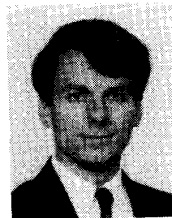
Keith Nabors received the B.E.S. degree from the Johns Hopkins University, Baltimore, Maryland, and the S.M. degree from the Massachusetts Institute of Technology. He is presently working towards the Ph.D. degree at the Massachusetts Institute of Technology.

Mr. Nabors' current interests are in numerical methods for circuit parameter extraction.

Songmin Kim was born in Seoul, Korea, on April 19, 1963. He received the B.Sc. degree in mathematics and the B.Sc. and M.Sc. degrees in electrical engineering from the Massachusetts Institute of Technology in 1986 and 1990.

Since 1990, he has been a Research Assistant in the Department of Electrical Engineering at the MIT. His research interests are in computational electromagnetics.

Mr. Kim is a member of Phi Beta Kappa, Tau Beta Pi and Sigma Xi.

Jacob White (S'80–M'83) received the B.S. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, and the S.M. and Ph.D. degrees in Electrical Engineering and Computer Science from the University of California, Berkeley.

He worked at the IBM T. J. Watson Research Center from 1985 to 1987, was the Analog Devices Career Development Assistant Professor at the Massachusetts Institute of Technology from 1987 to 1989, and is a 1988 Presidential Young Investigator. Dr. White is currently an Associate Professor at M.I.T., and his current interests are in theoretical and practical aspects of serial and parallel numerical methods for problems in circuit design and fabrication.