Fast Capacitance Extraction of General Three-Dimensional Structures

K. Nabors S. Kim J. White S. Senturia
Research Laboratory of Electronics and the Microsystems Technology Laboratory,
Department of Electrical Engineering and Computer Science,
Massachusetts Institute of Technology.

Abstract

Several improvements to the multipole-accelerated 3-D capacitance extraction program described in [1] are presented in this paper. A new adaptive multipole algorithm is given, and a preconditioning strategy for accelerating iterative method convergence is described. Results using these algorithms to compute the capacitance of general three-dimensional structures are presented, and they demonstrate that the modified approach is nearly as accurate as the more standard direct factorization algorithm, and can be as much as two orders of magnitude faster.

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 or functionality. In these problems, capacitance extraction is made tractable by assuming the conductors are ideal, and are embedded in a piecewise-constant dielectric medium. Then to compute the capacitances, Laplace's equation is solved numerically over the charge free region with the conductors providing boundary conditions. The usual numerical approach is to apply a boundary-element technique to the integral form of Laplace's equation[2] as in

$$\psi(x) = \int_{surfaces} G(x,x')\sigma(x')da' \qquad (1)$$

where σ is the surface charge density, $x, x' \in \Re^3$, da' is the incremental surface area, ψ is the surface potential and is known, and $G(x,x')$ is the Green's function which in free space is $\frac{1}{\|x-x'\|}$.

To numerically solve (1) for σ, the conductor surfaces are broken into n small panels or tiles. It is assumed that on each panel k, a charge, q_k, is uniformly distributed. Then for each panel, an equation is written that relates the potential at the center of that k^{th} panel, denoted p_k, to the sum of the contributions to that potential from the n charge distributions on all n panels[3]. The result is a dense linear system,

$$Pq = \overline{p} \qquad (2)$$

where $P \in \Re^{n\times n}$ is the matrix of potential coefficients; q, $\overline{p} \in \Re^n$ are the vectors of panel charges and given panel potentials respectively, and

$$P_{kl} = \frac{1}{a_l}\int_{panel_l} \frac{1}{\|x' - x_k\|}\, da', \qquad (3)$$

where x_k is the center of the k^{th} panel and a_l is the area of the l^{th} panel.

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 from 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, and this limits the size of the problem that can be analyzed to one with a few conductors.

In [1], a fast algorithm for computing the capacitance of three-dimensional structures of rectangular conductors in a homogenous dielectric was presented. The computation time for the algorithm, which was based on the hierarchical multipole algorithm [4], 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 short paper, we briefly 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.

In the next section, we briefly describe an adaptive multipole scheme that improves the efficiency of the multipole approach, and in Section 3 we present a preconditioner which accelerates GCR convergence and fits naturally with the multipole algorithm. Experimental results using our program FASTCAP to

[0]This work was supported by the Defense Advanced Research Projects Agency contracts N00014-87-K-825 and MDA972-88-K-008, the National Science Foundation contract (MIP-8858764 A02), F.B.I. contract J-FBI-88-067, and grants from I.B.M. and Analog Devices.

analyze a wide variety of structures, made possible by a link to the M.I.T. Micro-Electro-Mechanical Computer Aided Design (MEMCAD) system [5], are presented in Section 4.

2 The Hierarchical Multipole Algorithm

A complete description of the hierarchical multipole algorithm is not given here, the original description is in [4], 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.

2.1 Multipole Expansions

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

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

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 coefficents 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 Figure 1, there are d charged panels, and d panel centers for which 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).

In the Figure 1 case, the error due to truncating the multipole expansion is bounded [4], as in

$$\left| \psi(r_i, \phi_i, \theta_i) - \sum_{n=0}^{l} \sum_{m=-n}^{n} \frac{M_n^m}{r_i^{n+1}} Y_n^m(\phi_i, \theta_i) \right| \leq K_1 \left(\frac{R}{r} \right)^{l+1} \qquad (5)$$

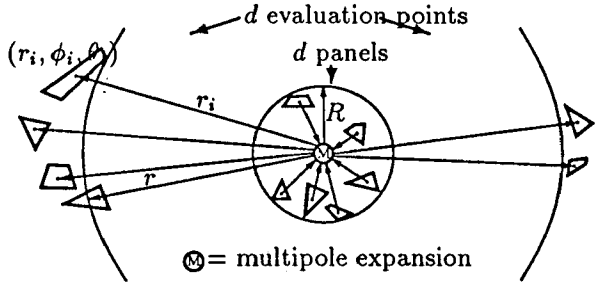

Figure 1: The evaluation of the potential at (r_i, ϕ_i, θ_i).

The quantities r and R are as in Figure 1 and K_1 is a constant independent of the multipole expansion order, l. The bound indicates that the multipole potential evaluations converge increasing 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 (6) 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 is less than half the distance between the region's center and the evaluation point.

2.2 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 insured by exploiting a hierarchical partitioning of the problem domain, the problem domain being defined as the smallest cube containing all the conductors.

Consider, for example, evaluating the potential at some point (r_i, ϕ_i, θ_i) in Figure 2 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 (Figure 2b)[1]. To insure that the errors due to truncating the multipole expansion shrink rapidly with expansion order, multipole expansions will not be used to represent the 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 Figure 2b are all greater than 0.5.

Squares 1, 2 and 3 are each divided into four squares,

[1] In the three-dimensional problem, the equivalent partitioning would be to divide a cube into eight smaller cubes.

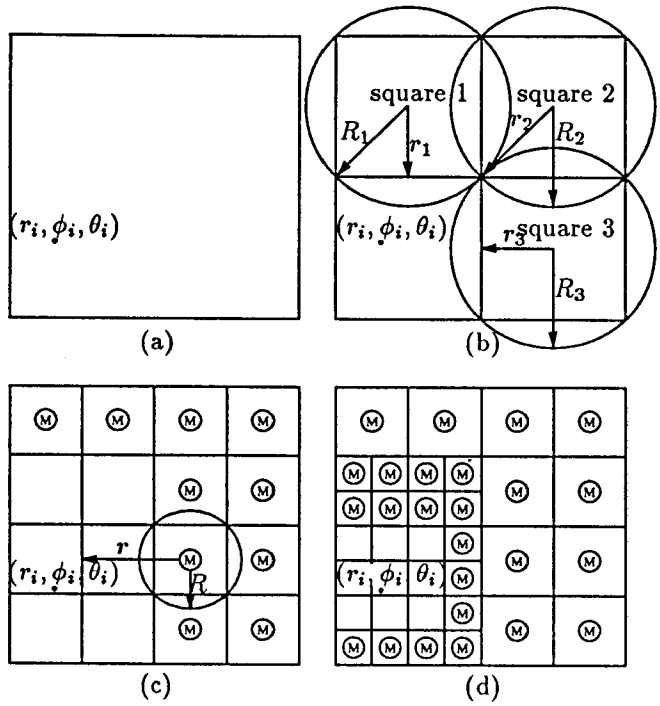

Figure 2: The evaluation of the potential at (r_i, ϕ_i, θ_i).

as in Figure 2c, 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 Figure 2c will be represented with a multipole expansion when evaluating the potential at points in the square containing (r_i, ϕ_i, θ_i).

So that multipole expansions can be used to represent the potential due to panel charges contained in the unmarked squares of Figure 2c, these squares are partitioned further, as in Figure 2d. 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 Figure 2d, 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 Figure 2d, and is computed directly from $P_{ij}q_j$ products. The "far" contribution to p_i is due to distant panels charges and is determined by evaluating the 25 mulitpole expansions indicated in Figure 2d. 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).

The above example suggest that evaluating n potentials requires order $n \log n$ operations. The hierarchical multipole algorithm given in [4], and used in the FASTCAP program described below, is more sophisticated than the above approach 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.

2.3 The Adaptive Algorithm

An adaptive multipole algorithm can be derived from the simplified approach described in Section 2 if the potential due to panel charges in a cube are 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 collection of panels for the problem of interest 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 partitions of panels starting with an 8-way partition and ending with an 8^L-way partition. The number of levels, L, is chosen so that the maximum number of panels in a finest, or L^{th}, 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 lowest-level cube, a multipole list has been constructed using a recursive approach similar to that given in the two-dimensional example above.

Alg. 2: Adaptive Algorithm to Compute $p = Pq$
Comment: Compute potential due to nearby charges.
for each lowest-level cube $i = 1$ to 8^L {
  for each panel j in lowest-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 multipole coefficients from q

```
Comment: order is the expansion order.
for each level j = L to 2 {
  for each j^th-level cube i = 1 to 8^j {
    if cube i has more than (order + 1)^2 panels {
      Compute the multipole coefficients from
      panel charges and/or from coefficients of
      any child cube multipole expansions.
    }
  }
}
Comment: Compute potential due to distant panels.
for each lowest-level cube i = 1 to 8^L {
  for each cube j in cube i's multipole list {
    if cube j has more than (order + 1)^2 panels {
      for each panel k in cube i {
        Evaluate the 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.
        }
      }
    }
  }
}
```

3 Preconditioning

In general, the GCR 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 $\tilde{C}$, in which case preconditioning the GCR algorithm is equivalent to using GCR to solve

$$P\tilde{C}x = \overline{p}. \tag{6}$$

for the unknown vector x, from which the charge density is computed by $q = \tilde{C}x$. Clearly, if $\tilde{C}$ is precisely P^{-1}, then (6) is trivial to solve, but then $\tilde{C}$ will be very expensive to compute.

Below we give 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 cube, as in Algorithm 3 below

```
Alg. 3: Forming C̃.
for For each lowest-level cube i = 1 to 8^L {
  Form P^i, the potential coefficient matrix for the
  reduced problem derived from considering only the
  panels in cube i and cube i's neighbors.
  Compute C̃^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 C̃^i.
    }
  }
}
```

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

$$\sum_{i=1}^{8^L}(\#\ rows\ in\ \tilde{C}^i) = n \tag{7}$$

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 GCR algorithm applied to solving (6), 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.

```
Algorithm 4: Forming q = C̃x.
  for each lowest-level cube i = 1 to 8^L {
    for each panel j in lowest-level cube i {
      for each panel k in cube i or its neighbors {
        Add C̃^i_jk x_k to q_j.
      }
    }
  }
```

4 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. 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 [5]. The structures described 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 GCR convergence tolerance of 0.01.

That the PAMA algorithm is nearly as accurate as the direct factorization is demonstrated using the 2x2 woven buss structure in Figure 3. The capacitances computed using the two methods are compared in Table 1, using coarse, medium, and fine discretizations of the woven buss structure, also shown in Figure 3. Note that the coupling capacitance C_{12} between conductor one and two, *which is 40-times smaller than*

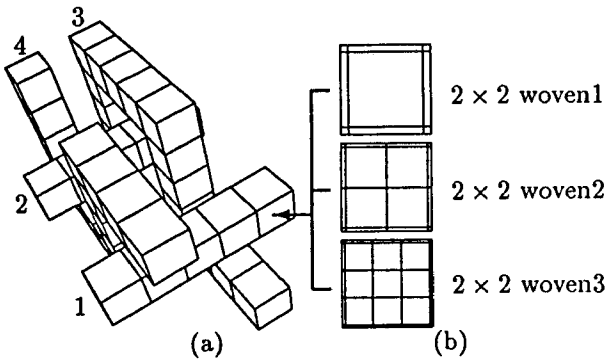

Figure 3: The 2 × 2 woven bus problem: bars have 1m×1m cross sections. The three discretizations are obtained by replacing each square face in (a) with the corresponding set of panels in (b).

Method	Problem					
	Woven1 1584 Panels		Woven2 2816 Panels		Woven3 4400 Panels	
	C_{11}	C_{12}	C_{11}	C_{12}	C_{11}	C_{12}
Direct	251	-6.35	253	-6.45	254	-6.47
PAMA	251	-6.25	253	-6.33	254	-6.38

Table 1: Capacitance values (in pF) illustrating FASTCAP's accuracy for the complicated geometry of Figure 3.

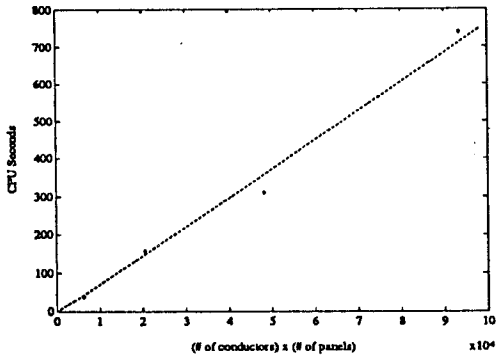

Figure 4: Plot of CPU Time vs mn for the woven-buss example. The dashed line is the best constant-slope fit to the data indicated with *'s.

the self-capacitance, 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 using a parameterized version of the woven bus structure in Figure 3, that is, the structure is extented to make a 3 × 3 buss, a 4 × 4 buss and a 5 × 5 buss. In Figure 4, the CPU

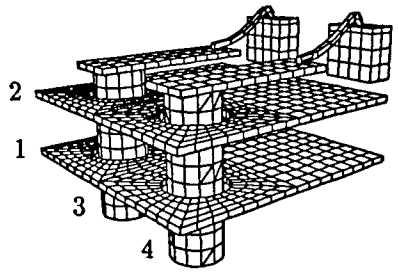

Figure 5: Two signal lines passing through conducting planes; via centers are 2mm appart.

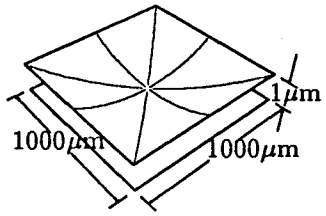

Figure 6: A schematic illustration of the square diaphragm problem. The two plates are 0.02μm appart at the center.

times for computing the capacitances of the woven bus structures are plotted as a function of mn, and as the graph clearly demonstrates the computation time does grow linearly.

To demonstrate the effectiveness of various aspects of the PAMA algorithm on several problems, in Table 2 the CPU times required to compute the capacitances of three different examples using four different methods is given. The example 5 × 5 *Woven buss* is described above; the example *Via*, shown in Figure 5, models a pair of connections between integrated circuit pins and a chip-carrier; and the example *Diaphragm*, show in Figure 6, is a model for a microsensor[6].

From Table 2, it can be seen that using the adaptive multipole algorithm typically reduces the computation time by a factor of two over using the multipole algorithm alone, and that using the preconditioner can reduce the computation time by as much as a factor of three. Also, note that the PAMA algorithm is more than two orders of magnitude faster than direct methods for these large problems.

5 Conclusions

As the results indicate, the multipole-accelerated capacitance extraction algorithm, with the above modifications, is capable of quickly and accurately analyzing very complex structures. Current work is on extending the approach to allow piecewise constant dielectrics.

The authors would like to thank David Ling and Albert Ruehli of the I.B.M. T. J. Watson Research

Method	Via 6185 Panels	Diaphragm 7488 Panels	5x5 Woven Bus 9630 Panels
Direct	(490)	(890)	(1920)
MA	11	11	40
AMA	3.8	9.0	19
PAMA	3.1	3.0	12

Table 2: CPU times in minutes on an IBM RS6000/540 to compute, times in parentheses are extrapolated.

Center for the many discussions that led to the approach presented here, as well as their help along the way, and Brian Johnson for providing several examples. In addition we would like to acknowledge the help of the members of the M.I.T. custom integrated circuits group.

References

[1] K. Nabors and J. White, "Fastcap: A multipole accelerated 3-d capacitance extraction program." To appear in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.*

[2] 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.

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

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

[5] F. Maseeh, R. M. Harris, and S. D. Senturia, "A CAD architecture for microelectromechanical systems," in *IEEE Micro Electro Mechanical Systems*, (Napa Valley, CA), pp. 44–49, 1990.

[6] B. P. Johnson, S. Kim, S. D. Senturia, and J. White, "MEMCAD capacitance calculations for mechanically deformed square diaphragm and beam microstructures," in *Proceedings of Transducers 91*, June 1991.