Enhanced QMM-BEM Solver for Three-Dimensional Multiple-Dielectric Capacitance Extraction Within the Finite Domain

Wenjian Yu, *Member, IEEE,* and Zeyi Wang, *Member, IEEE*

***Abstract*—The computational time and memory of three-dimensional capacitance extraction have been greatly reduced by using a quasi-multiple medium (QMM) technology, because it enlarges the matrix sparsity produced by the direct boundary element method. In this paper, an approach to automatically determining the QMM cutting pair number and a preconditioning technique are proposed to enhance the QMM-based capacitance extraction. With these two enhancements, the capacitance extraction can achieve much higher speed and adaptability. Experimental results examine the efficiency of two enhancements and show over 10× speed-up and memory saving over the multipole approach with comparable accuracy.**

***Index Terms*—Capacitance extraction, direct boundary element method (BEM), finite domain, multiple dielectrics, preconditioning, quasi-multiple medium (QMM) method.**

I. Introduction

With THE feature size scaled down and work frequency increased, the parasitic parameters of the interconnects have become more and more important for the design of high-performance very large scale integration (VLSI) circuits. Therefore, efficient computation of three-dimensional (3-D) capacitance with multilayered dielectrics has become a focus of current research works.

Many fast capacitance extraction algorithms have been proposed in the literature, such as [1], [2], and [15]. They are based on the boundary element method (BEM) using a so-called total-charge Green's function approach [3], and the matrix solving time is significantly reduced by using the fast multipole approach [1], hierarchical approach [2], or the precorrected fast Fourier transform (FFT) algorithm [15]. Unlike the above BEM, another kind of BEM, the direct BEM [4], is not well known. The direct BEM obtains the direct boundary integral equation (BIE) by adopting Green's identity and using the free-space Green's function as a weighting function [4]. Also, the direct BEM produces a sparse matrix for a multiregion problem, while not a dense matrix. In [5], Fukuda *et al.* apply the direct BEM to the two-dimensional (2-D) capacitance extraction. In [6], Yuan and Banerjee present the parallel formulations for the fast multipole approach and the direct BEM for 3-D capacitance extraction. In [14], Bachtold *et al.* extend the multipole method to handle the "potential boundary integral" (whose kernel is $1/r^3$) in the direct BEM.

In [7], a quasi-multiple medium (QMM) method was proposed to enlarge the matrix sparsity of the direct BEM by making 3-D domain decomposition. With the technology of storing sparse blocked matrix and the efficient iterative equation solver, such as GMRES [11], the QMM-accelerated direct BEM greatly reduced the CPU time and memory usage. Furthermore, a semi-analytical method to handle the nearly singular boundary integrals was proposed to achieve high speed and accuracy [7].

In the existing QMM method, each dielectric layer is cut into $m \times n$ fictitious medium blocks uniformly. The QMM cutting pair (m, n) was manually specified or obtained by an empirical formula [7]. In this paper, two improvements based on [7] are proposed. The first one is an approach to automatically generate an optimal cutting pair so as to resolve the remaining problem in [7]. The number of nonzero matrix entries are calculated for candidates of (m, n), since it is a good indicator of the overall computational time. Finally, the selected cutting pair gives almost the fastest computational speed. The second improvement is that two efficient preconditioners for the GMRES solution are proposed. With the above two modifications, the enhanced QMM-BEM solver can compute the interconnect capacitance with multilayered dielectrics more effectively.

It should be pointed out that the QMM technology is mainly used in a capacitance model of finite domain while not the open-space model widely analyzed by conventional BEMs [1]–[3]. The open-space model, where the electric field is extended to infinity, is ideal for simulating isolated structures. However, for the on-chip application, this condition can hardly be guaranteed because of the influence of neighboring conductors. The finite-domain model has the Neumann boundary surrounding the simulated structures. This boundary condition is also called reflective boundary condition and is introduced as the "magnetic wall" in the dimensional reduction technology [8]. Many published methods have utilized this capacitance model to deal with actual interconnect structures [5], [7]–[10], and the commercial softwares Raphael and SpiceLink consider it as default setting.

The $k \times k$ crossovers embedded in multilayered dielectrics within the finite domain are calculated in the numerical experiments. The results show over a runtime improvement of over ten times and memory saving over the multipole approach (FastCap 2.0) with equal accuracy. The result of Raphael with adequate mesh is considered as a criterion.

Manuscript received April 30, 2003; revised July 15, 2003. This work was supported in part by the China National Foundation for Key Basic Research under Grant G1998030404 and by the National 863 Program for Development of High Technology under Grant 2002AA1Z1460-1.

The authors are with the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: ywj77@computer.org).

Digital Object Identifier 10.1109/TMTT.2003.821930

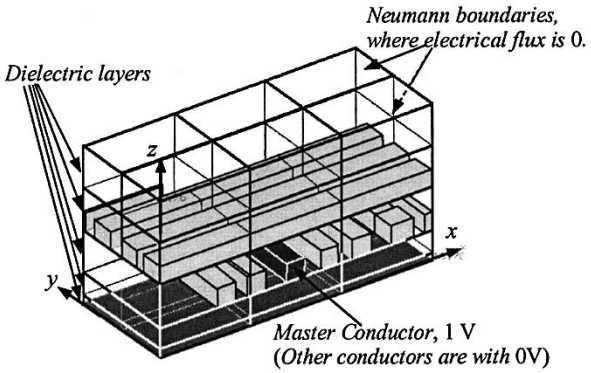

Fig. 1. Typical 3-D interconnect capacitor with five dielectric layers is cut into a 3×2 structure.

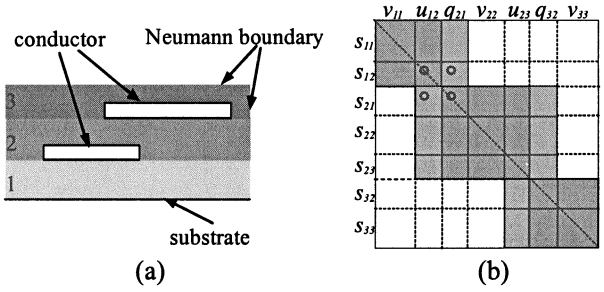

Fig. 2. (a) Problem with three dielectrics. (b) Corresponding coefficient matrix $\boldsymbol{A}$, where the gray blocks stand for nonzero entries and the type of discretized variables are signed beside the corresponding matrix columns.

The remainder of this paper is organized as follows. In Section II, the QMM-based capacitance extraction is briefly reviewed. Sections III and IV present the approach to determine the cutting pair and the preconditioning technique, respectively. Experimental results are reported in Section V, and conclusions are drawn in Section VI.

II. QMM-Based Capacitance Extraction [7]

In direct BEM, the Laplace's equation fulfilled by the electric potential u in each homogenous dielectric region can be transformed into the following direct BIEs [4]–[7]:

$$cu + \int_{\Gamma_i} q^* u d\Gamma = \int_{\Gamma_i} u^* q d\Gamma, \qquad i = 1, \ldots, M \tag{1}$$

where c is a constant depending on the boundary geometry, Γ_i is the boundary of dielectric region i, q is the normal electrical field intensity, and $q = \partial u / \partial \boldsymbol{n}$. For 3-D space, the fundamental solution u^* is $1/4\pi r$. Employing the collocation scheme and constant quadrilateral elements, a group of discretized BIEs are obtained from (1) for each dielectric region. Evaluating two types of boundary integral, we obtain linear equations [7].

Besides, the u and q fulfill the compatibility equations along the interface of two adjacent dielectrics a and b as follows:

$$\begin{cases} \varepsilon_a \cdot \dfrac{\partial u_a}{\partial \boldsymbol{n}_a} = -\varepsilon_b \cdot \dfrac{\partial u_b}{\partial \boldsymbol{n}_b} \\ u_a = u_b \end{cases} \tag{2}$$

where ε_a and ε_b stand for the permittivities of dielectrics a and b, respectively. The discretized BIEs (linear equations) for each dielectric can be coupled utilizing the compatibility equation (2). Then, substituting the boundary conditions (u is known on conductor surfaces as the bias voltage, and q is supposed to be zero on the Neumann boundary as shown in Fig. 1), we get an overall linear system

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{f} \tag{3}$$

where x is a vector comprising all discretized unknowns of u or q. Theoretically, any arrangement of the discretized BIEs in (3) is correct. But, without careful consideration, the population of the coefficient matrix $\boldsymbol{A}$ would be too chaotic to make the equation solution efficient. In [7], Yu *et al.* present an effective arrangement of the unknowns and collocation points, as well as the storage scheme for the resulting sparse matrix $\boldsymbol{A}$. Because the nonzero matrix blocks are distributed very regularly, the additional CPU time for the equation solution is saved remarkably, especially for a problem involving a large number of subregions [7].

The above process belongs to the conventional direct BEM for multiregion problem. A localization character is revealed by (1), where the variables in each BIE are within the same dielectric region. This character of direct BEM results in a blocked sparse coefficient matrix $\boldsymbol{A}$ for a multidielectric problem. For example, a typical capacitor with three dielectrics and the corresponding sparse matrix $\boldsymbol{A}$ are shown in Fig. 2 (the efficient equation arrangement is used; for a more detailed illustration of the notations, please refer to [7]). In the QMM method, every actual dielectric is decomposed into some fictitious medium blocks, whose permittivities are all the same as the original dielectric, to increase the sparsity of matrix $\boldsymbol{A}$. With the storage technique of the sparse blocked matrix and iterative equation solvers such as the GMRES algorithm [11], the computing time and memory usage for the original problem will be greatly reduced. In practice, each layer of an actual multilayered interconnect structure is decomposed into an $m \times n$ fictitious medium blocks, perpendicular to the bottom substrate plane (see Fig. 1). In order to decrease the additional efforts brought by the QMM decomposition, the cutting planes are also dispersed along the X or Y axis uniformly. Thus, the QMM cutting pair (m, n) completely controls the geometry structure of the multiregion problem finally handled. Also, it greatly influences the total computing efficiency of the QMM-based capacitance extraction. Besides, a strategy of nonuniform density partition is adopted to generate fewer boundary elements without loss of accuracy [7].

The existing theoretical analysis has indicated that the number of nonzero entries in matrix $\boldsymbol{A}$ (denoted by Z) is very important to the whole computational efficiency. Generally, fewer nonzero matrix entries mean less memory usage and computing time. Although some additional unknowns are introduced on the additional interfaces between the fictitious mediums when using the QMM technique, the memory usage and CPU time for the overall computation are greatly reduced, since the reduction of Z can overwhelm the disadvantage of adding few unknowns into the majority of unknowns on the conductors' surfaces [7].

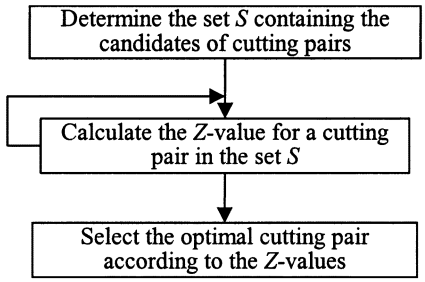

Fig. 3. Flowchart of the SMZ approach to selecting the QMM cutting pair.

III. Determining the QMM Cutting Pair

Based on the importance of Z for the total computational efficiency, an approach called selection with minimal Z value (SMZ) is proposed to select the QMM cutting pair. In the SMZ approach, parameter Z is calculated in advance for a number of candidates of (m, n), and then an optimal cutting pair is selected. Usually, the minimal Z value means the fastest computational speed. Calculating the Z values for different QMM cutting pairs prior to implementing the QMM decomposition and following computations is the basic idea of the SMZ approach. Fig. 3 shows the flowchart of the SMZ approach.

In the first step of the SMZ approach, a set S containing the candidates of QMM cutting pair is determined. Some principles are considered to reduce the candidates of (m, n) in the set S for the sake of saving time. Without loss of generality, the value range of m (which stands for the number of fictitious medium blocks along the X axis) is discussed. First, m should not be too small; otherwise QMM could not result in great acceleration. Second, too large a value of m is also not advisable because the great deal of additional elements on fictitious interfaces would affect the advantage of QMM cutting and lower the computational efficiency, both in CPU time and memory usage. So, a moderate value range should be taken for m, according to the domain geometry. Besides, further limitation for (m, n) could work for the simulated domain with large aspect ratio. With the above considerations, the number of elements in S becomes relatively small. This way, the additional CPU time for the SMZ algorithm does not greatly influence the whole computational efficiency. At the same time, the S is also sufficiently large enough to find an optimal cutting pair.

Next, we analyze how to calculate the Z-value for a giving cutting pair. If the region i involves V_i discretized boundary unknowns (u or q), there are $Z_i = N_i \cdot V_i$ nonzero coefficients in the N_i discretized BIEs, which becomes the nonzero entries of matrix $\boldsymbol{A}$ finally. N_i is the number of the source points in region i whose quantity equals the number of boundary elements in region i. The boundary elements are classified into two kinds: the first kind includes those on the Dirichlet or Neumann boundary, where only one unknown of u or q exists on each element; the second kind includes those on the region interface, where two unknowns of u and q exist on each element. Thus, we have

$$N_i = (a_i + b_i) \text{ and } V_i = (a_i + 2b_i) \tag{4}$$

where a_i stands for the number of first kind of boundary elements, b_i stands for the number of second kind of boundary elements, both in dielectric region i. We then have

$$Z_i = N_i \cdot V_i = (a_i + b_i)(a_i + 2b_i). \tag{5}$$

Summing up Z_i for all dielectric regions, we obtain

$$Z = \sum_{i=1}^{Q} Z_i = \sum_{i=1}^{Q} (a_i + b_i)(a_i + 2b_i) \tag{6}$$

where Q is the total number of dielectric regions. Without QMM cutting, Q equals the dielectric number M; for the computation with the QMM cutting pair (m, n), Q equals $M \times m \times n$. Therefore, the numbers of two kinds of boundary elements in each region are first calculated, and then the Z value is obtained with (6). The calculation of a_i and b_i is related to the element partition method. For a certain partition method, the number of elements on each boundary surface can be calculated with the surface geometry dimension and then be counted in a_i or b_i; therefore, the $m \times n$ structures need not be actually generated. The algorithm description of the second step in Fig. 3 is shown as follows:

```
For (all (m,n) in the set S)
 Clear the arrays a[ ] and b[ ], which
record the number of first and second kind
of elements, respectively;
 Use arrays CutTrackX[ ] and CutTrackY[ ] to
record the positions of cutting planes;
 Process each dielectric layer to compute
the element numbers of each boundary sur-
face and to modify a[ ] and b[ ];
 Calculate Z[m,n] with (6);
EndFor
```

After getting the Z values for all cutting pairs in S, some cutting pairs with nearly minimal Z value are selected. Among them, the one with the least product of $m \times n$ becomes the optimal cutting pair, in order to balance the memory usage and the expected computational speed. Using this optimal QMM cutting pair, computational accuracy and memory usage of the QMM method are both preserved while achieving higher computational speed.

IV. Efficient GMRES Preconditioners

In the organization of matrix $\boldsymbol{A}$, the order of the source points is consistent with that of the unknowns, so that the diagonal entries of the matrix are obtained by the singular integral, which results in a nonzero entry with large absolute value. For this reason, the Jacobi (or named diagonal) preconditioner can bring quick convergence to the GMRES iterative solution [7]. In this section, we will discuss two easily computed preconditioners which bring a faster convergence rate than the Jacobi preconditioner for the actual 3-D capacitance extraction.

A. Basic Idea

For (3), a preconditioned solution is equivalent to using GMRES to solve $\boldsymbol{APy} = \boldsymbol{f}$ for the unknown vector $\boldsymbol{y}$, from which the original unknown vector $\boldsymbol{x}$ is computed by $\boldsymbol{x} = \boldsymbol{Py}$. This is called the right preconditioning. An ideal preconditioner should first well approximate to $\boldsymbol{A}^{-1}$ so that it can improve the condition of the linear system, and it should also be easily computed and with great sparsity in order not to increase the computation for constructing and using it in the iterations. To certain extent, Vavasis proposed a good idea in [12] to construct preconditioner, which is briefly introduced below.

Each row of the preconditioner $\boldsymbol{P}$ is generated separately. Let the ith column of $\boldsymbol{P}^{\mathrm{T}}$ be denoted by $\boldsymbol{p}_i$, i.e., $\boldsymbol{P}^{\mathrm{T}} = (\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_N)$. Ideally, we would like to have

$$\boldsymbol{PA} = \boldsymbol{I} \Leftrightarrow \boldsymbol{A}^{\mathrm{T}}\boldsymbol{p}_i = \boldsymbol{e}_i \tag{7}$$

where $\boldsymbol{e}_i$ is the ith column of the identity matrix. Note that each column (or row) of matrix $\boldsymbol{A}$ corresponds to a discretized unknown (or source point), and further to a boundary element. Therefore, we use the number of row or column as the index of its corresponding source point, unknown, and element. By some strategy we may determine a small list L of indices drawn from $\{1, 2, \ldots, N\}$, which denotes the unknowns having the most impact on the current unknown i. Then, (7) can be reduced to

$$\overline{\boldsymbol{A}}^{\mathrm{T}}\overline{\boldsymbol{p}}_i = \overline{\boldsymbol{e}}_i \tag{8}$$

where the bars over the variables indicate that all of the rows and columns except for those in L are deleted. After solving (8), we expand $\overline{\boldsymbol{p}}_i$ back to the corresponding entries in row i of $\boldsymbol{P}$. Repeating the above procedure for all rows, we get the whole sparse matrix $\boldsymbol{P}$.

For example, if the L has three indices, and the first one is the current row i, then (8) will be

$$\begin{pmatrix} a_{l_1 l_1} & a_{l_2 l_1} & a_{l_3 l_1} \\ a_{l_1 l_2} & a_{l_2 l_2} & a_{l_3 l_2} \\ a_{l_1 l_3} & a_{l_2 l_3} & a_{l_3 l_3} \end{pmatrix} \begin{pmatrix} p_{l_1} \\ p_{l_2} \\ p_{l_3} \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \tag{9}$$

where a_{ij} means the entry of $\boldsymbol{A}$ on the ith row and jth column, and $l_1 = i$.

B. Extended Jacobi (EJ) and $\mathrm{MN}(n)$ Preconditioner

Two strategies for selecting the set L are proposed to construct our preconditioners. The first one is called EJ preconditioner. Actually, the Jacobi preconditioner uses the $L = \{i\}$ for each row. However it does not consider all effects of the singular integrals for a multiregion BEM analysis. For the boundary element Γ_{I} on the interface of medium region i and j, the two unknowns on it are denoted by $u_{ij}(\Gamma_{\mathrm{I}})$ and $q_{ji}(\Gamma_{\mathrm{I}})$. Note that the source point on element Γ_{I} presents twice in the matrix $\boldsymbol{A}$, for the discrete BIE of region i and region j, respectively. Therefore, the singular integral on element Γ_{I} has four positions in matrix $\boldsymbol{A}$ [shown as the small circles in Fig. 2(b)]. Two of them are not on the main diagonal. The EJ preconditioner is based on the above observation and let L contain two indices of the row itself and the other occurrence of the source point on the same element, for the row corresponding to an interface element. Otherwise, L only contains the index of the current row. The EJ preconditioner is a little more complex than the Jacobi (for some rows, a 2×2 equation is solved), but it accelerates the convergence remarkably.

In the EJ preconditioner, no "neighbor" boundary element is considered. To bring faster convergence to the GMRES iteration, a mesh neighbor, i.e., $\mathrm{MN}(n)$, preconditioner is proposed, where n stands for the number of neighbor elements. The geometry distance of two elements does not need to be calculated, since the matrix $\boldsymbol{A}$ is stored explicitly and its entry value can be used to judge the neighborhood. For each pair of elements, the maximum absolute value of matrix entries (more than one, if the interface element is involved) representing the interactions between them is called "gravitation" here. Comparing each nonzero entries on row i, the n elements that have the maximun "gravitations" to the current element can be selected. These n elements are then considered as the most neighboring to row i's source element, and the indices of their variables are added to L. Because the index for the current element must be selected and one element may contain two variables, the L has the maximum degree of $2(n+1)$ in the $\mathrm{MN}(n)$ preconditioner.

The difference between our $\mathrm{MN}(n)$ preconditioner and other MN-like preconditioners (such as that in [1]) is that we use the explicitly stored matrix entry to judge the neighborhood to avoid the relative complex calculation of the 3-D distance between elements. So, our method has less computational consumption for a little n and is adapted to the 3-D finite-domain capacitance extraction with multiple dielectrics very well.

Since the GRMES iterations are fewer in 3-D capacitance extraction (the relative error norm of 10^{-2} or 10^{-3} is usually used) and the coefficient matrix involved is much sparser because of using the QMM technology, the simplicity of preconditioning is very important. Therefore, some traditional preconditioners such as that using the incomplete *LU* decomposition (ILUD) and blocked diagonal preconditioner are forbidden in our consideration. This demonstrates the significance of easily computed preconditioners, such as those proposed above.

More than 100 structures of VLSI interconnects are computed, and we find that the EJ and $\mathrm{MN}(1)$ both have high efficiency. To compare with the GMRES solver using the Jacobi preconditioner, the new solvers using these two preconditioners both can reduce the computational time by about 30% or more on an average. For the problem with larger order (10^4 or more), the $\mathrm{MN}(1)$ preconditioner seems to yield a better performance.

V. Numerical Results and Discussion

The enhanced QMM-BEM solver (QBEM) is compared with FastCap 2.0 and Raphael (version 2000.2). All computations are done on a SUN Ultra Enterprise 450 with UltraSparc II processors at 248 MHz. FastCap with default expansion order 2 is denoted by FastCap(2), and a faster program FastCap(1) has the expansion order 1. Raphael is a widely used commercial software, with a finite difference solver (RC3) with advanced nonuniform meshing scheme. The result of Raphael under very dense mesh is often used as a criterion by the industry.

The test examples are $k \times k$ bus crossing conductors embedded in five layered dielectrics ($k = 2$–5). The 2×2 bus example is shown in Fig. 4. Each bus in the $k \times k$ example is scaled to $1 \times 1 \times (2k + 5)$ (unit in micrometers). The distance

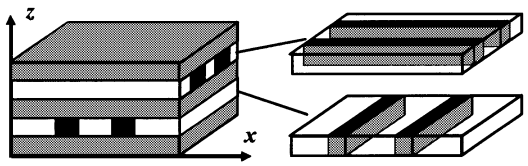

Fig. 4. 2×2 crossover embedded in five dielectric layers.

between the conductors in the same layer is 1, and the distance between the Neumann boundary and its neighboring conductor is 3. The thickness and relative permittivity of every layer is 1 and 3.9, respectively. All crossovers have a ground plane at the very bottom. The conductors are numbered from one side to the other side $1, 2, \ldots, k$ (bottom layer) and then $k+1, \ldots, 2k$ (top layer). The total capacitance of conductor 1 and its couplings with the other conductors are computed; they form a column of the whole capacitance matrix, which is denoted by vector $\boldsymbol{C}_1$ here. These finite-domain and multidielectric problems can be easily handled by Raphael and QBEM. In the input of FastCap, 0 is assigned to the permittivity of outer space to handle the finite-domain model, and each interface between the dielectric layers is specified to make the comparisons equitable. Also, to make FastCap only compute the capacitances related with conductor 1, the "-rs" option is used to remove other conductors from solving [13].

The above crossover problems are computed by Raphael with 0.25×10^6 grid points and 10^6 grid points, FastCap(1), FastCap(2), and QBEM. Our GMRES solver reduces the two-norm of the residual to 1% of the initial residual, which is the same condition used in FastCap. The number of panels per edge for each conductor is specified individually, so as to make FastCap compute a similar system of nonuniform panels like that in QBEM. Using the result $\boldsymbol{C}_1$ of Raphael with 1M grids as the standard, the error of capacitance vector $\boldsymbol{C}_1'$ computed by another program is estimated in the two-norm: $\|\boldsymbol{C}_1' - \boldsymbol{C}_1\|/\|\boldsymbol{C}_1\|$.

Table I compares the QMM-BEM solver, FastCap, and Raphael. The following is a summary of the comparison.

1) Using the Raphael's result under 1M grids as criterion, the errors of FastCap(1), FastCap(2), and QBEM are all within 3%. The error of FastCap(1) seems larger, while our method has less error.
2) FastCap uses almost the same (even fewer) number of panels as our QBEM and uses the nonuniform partition (dense near the master). So, under the same-scale discretization, the speed-up of QBEM to FastCap(2) is from 12 to 16 and 6 to 9 to FastCap(1).
3) The QBEM uses 1/18 to 1/11 of the memory used by FastCap(2). Compared with FastCap(1), the QBEM's memory usage is 1/11 to 1/7 that of FastCap (1).
4) Compared with Raphael of 0.25M grids, QBEM has over a runtime improvement of over $55\times$ and $17\times$ memory saving.

In the above computations with the enhanced QMM-BEM solver, the EJ preconditioner is used, and the optimal QMM cutting pairs generated by the SMZ approach are (4, 4), (5, 5), (3, 3), and (3, 3), respectively, for the four crossover problems. The detailed results for the 4×4 crossover are shown in Table II. To demonstrate the validity of the SMZ approach, we use the QMM-BEM solver to compute the 4×4 crossover problem with manually specifying the cutting pairs from set $S = \{(m, n) | 2 \le m, n \le 5, |m - n| \le 2\}$. The corresponding Z values and total CPU times are depicted in Fig. 5. The nearly linear relationship between the computing time and the Z value is demonstrated, which is the base of the SMZ approach. From Fig. 5, we can see that (3, 3) corresponds to nearly minimal Z value and has a little product of $m \times n$. So, it is selected to be the optimal cutting pair and results in very short computational time (Note that the time in Table II is about 0.1 s longer than that in Fig. 5, because the SMZ approach costs some CPU time). Compared with the (4, 4) generated by an empirical formula, the optimal cutting pair makes the QMM-based capacitance extraction about 20% faster.

For the 4×4 crossover problem, different preconditioners discussed in Section IV are used for comparison. Related data are listed in Table III. From it we can see that $\mathrm{MN}(1)$ and $\mathrm{MN}(2)$ consume a great deal of time in constructing and using the preconditioner for the problem (with 2435 variables), so the reduction of the iterative number does not efectively speed up solution of the equation. The EJ preconditioner is a little more complex than the Jacobi, but has six fewer steps in iteration, and therefore it achieves the least computational time for solving the equation, which is about 30% less than that using the Jacobi preconditioner. More experiments have also shown that the iteration number decreases gradually for preconditioners in this order: Jacobi, EJ, $\mathrm{MN}(1)$, and $\mathrm{MN}(2)$. Also, the EJ or $\mathrm{MN}(1)$ has the best overall performance, achieving much faster equation solution than the Jacobi.

TABLE I
COMPARISONS OF FASTCAP, RAPHAEL, AND QBEM FOR THE CROSSOVER PROBLEM

	Test problems			
	2×2	3×3	4×4	5×5
	FastCap(1)			
Time(s)	7.9	9.2	10.0	12.5
Memory(MB)	17.9	17.9	19.1	23.7
Panel	1080	1284	1487	1804
Error	1.6%	2.1%	3.4%	2.9%
	FastCap(2)			
Time(s)	11.5	15.1	17.5	24.3
Memory(MB)	26.4	28.4	30.7	38.5
Panel	1080	1284	1487	1804
Error	2.1%	2.3%	2.6%	3.0%
	Raphael (0.25M grids)			
Time(s)	78.8	67.1	88.9	81.9
Memory(MB)	47	45	48	48
Error	0.3%	0.4%	0.5%	0.8%
	QBEM			
Time(s)	1.0	1.3	1.6	1.5
Memory(MB)	1.7	2.7	2.1	2.1
Panel [a]	1184	1431	1502	1558
Error	2.7%	2.5%	1.0%	1.2%
Sp. to FC1 [b]	8	9	6	8
Mem_R to FC1 [b]	11	7	9	11
Sp. to FC2 [b]	12	12	11	16
Mem_R to FC2 [b]	16	11	15	18

[a] The panels on the interface between the fictitious medium blocks are not counted.
[b] Sp. to FC1/2 means the speed-up ratio to FastCap(1/2); Mem_R to FC1/2 means the memory reduction compared with FastCap(1/2).

TABLE II
COMPARISON OF 4 × 4 CROSSOVER PROBLEM (CAPACITANCE IN PICOFARADS)

	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	Cut pair	Panel	Ele_N	Var_N	Z-val	Iter	Mem(MB)	Tgen(s)	Tsol(s)	Time(s)
QBEM	2422	-620	-8.7	-2.0	-224	-143	-143	-224	(3, 3)	1502	1896	2435	0.24M	11	2.1	1.02	0.29	1.6
FastCap(2)	2352	-573	-4.1	-3.2	-214	-143	-143	-214	-	1487	1487	1487	-	9	30.7	13.4	4.0	17.5
FastCap(1)	2345	-547	5.6	2.3	-220	-142	-145	-219	-	1487	1487	1487	-	13	19.1	6.9	2.9	10.0
Raphael(1M)	2408	-601	-9.2	-1.9	-224	-150	-150	-224	-	-	-	-	-	-	-	-	-	-

Panel means the number of "true" elements not including that on the fictitious interfaces; Ele_N means the number of all elements including those on the fictitious interfaces; Var_N means the number of unknowns; Z-val means the number of non-zero matrix entries in QMM-BEM solver (Z value); Iter is the number of iterative steps; Tgen is the time for forming the linear equation system; Tsol is the time for equation solution; Time is the total time of capacitance extraction.

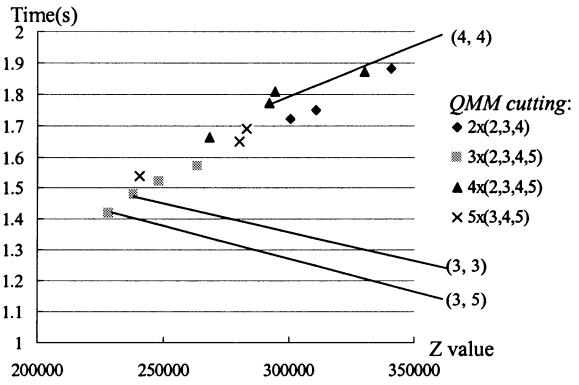

Fig. 5. Time of QMM-based capacitance extraction versus the Z value for different QMM cutting pairs imposed on the 4 × 4 crossover.

TABLE III
COMPARISON OF FOUR PRECONDITIONERS FOR THE 4 × 4 CROSSOVER PROBLEM

Jacobi		EJ		MN(1)		MN(2)	
Iter	Tsol(s)	Iter	Tsol(s)	Iter	Tsol(s)	Iter	Tsol(s)
17	0.42	11	0.29	11	0.44	10	0.52

The enhanced QMM-BEM solver and FastCap are all of the boundary integral method. So, for the same finite-domain multidielectric problem, N boundary elements not including that on the fictitious medium interfaces will guarantee the same accuracy for both methods. The FastCap utilizes the BEM of the total-charge Green's function, which produces a dense matrix with N^2 nonzero entries. With the multipole approach, not all matrix entries need to be computed, and the matrix-vector multiplication is accelerated. The direct BEM used in the QMM-BEM solver has the character of resulting in a sparse matrix for a multiregion problem. With the QMM method, the degree of the matrix is a little more than N (adding elements on the fictitious interfaces), but the sparsity is greatly enlarged. So, the nonzero matrix entries are much less than N^2, and the matrix-vector multiplication is also accelerated. Since both methods have almost the same number of iterative steps, the QMM method has shown the same or better efficiency than the multipole approach on matrix sparsification. Furthermore, the careful processing of the integrals and the characters of direct BEM make the equation forming fast and matrix-vector multiplication more convenient than FastCap, which has a large auxiliary cost on the cube partition and multipole expansion (see Table II). With the above analysis and experiment results, we can see that the QMM-BEM solver is superior to the multipole accelerated BEM for the actual finite-domain capacitance extraction.

VI. CONCLUSION

Two improvements are added in the QMM-based capacitance extraction: one selects an optimal cutting pair to bring the fastest computational speed, and the other proposes the EJ and $\mathrm{MN}(1)$ preconditioners to solve the equation 30% faster than the original Jacobi prconditioner. The numerical results show the enhanced QMM-BEM solver has a speed-up of over $10\times$ and memory saving over FastCap with comparable accuracy.

ACKNOWLEDGMENT

The authors would like to thank H. Wei, Dr. Z. Yang, and Prof. X. Hong, Tsinghua University, Beijing, China, for many helpful discussions.

REFERENCES

[1] K. Nabors and J. K. White, "Multipole-accelerated capacitance extraction algorithms for 3-D structures with multiple dielectrics," *IEEE Trans. Circuits Syst. I*, vol. 39, pp. 946–954, Nov. 1992.
[2] W. Shi, J. Liu, N. Kakani, and T. Yu, "A fast hierarchical algorithm for three-dimensional capacitance extraction," *IEEE Trans. Computer-Aided Design*, vol. 21, pp. 330–336, Mar. 2002.
[3] C. Wei *et al.*, "Multiconductor transmission lines in multilayered dielectric media," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-32, pp. 439–450, Apr. 1984.
[4] C. A. Brebbia, *The Boundary Element Method for Engineers*. London: Pentech Press, 1978.
[5] S. Fukuda, N. Shigyo, K. Kato, and S. Nakamura, "A ULSI 2-D capacitance simulator for complex structures based on actual processes," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 39–47, Jan. 1990.
[6] Y. Yuan and P. Banerjee, "Comparative study of parallel algorithms for 3-D capacitance extraction on distributed memory multiprocessors," in *Proc. IEEE Int. Conf. Computer Design*, Sept. 2000, pp. 133–138.
[7] W. Yu, Z. Wang, and J. Gu, "Fast capacitance extraction of actual 3-D VLSI interconnects using quasi-multiple medium accelerated BEM," *IEEE Trans. Microwave Theory Tech.*, vol. 51, pp. 109–119, Jan. 2003.
[8] W. Hong *et al.*, "A novel dimension-reduction technique for the capacitance extraction of 3-D VLSI interconnects," *IEEE Trans. Microwave Theory Tech.*, vol. 46, pp. 1037–1043, Aug. 1998.
[9] Z. Zhu, H. Ji, and W. Hong, "An efficient algorithm for the parameter extraction of 3-D interconnect structures in the VLSI circuits: Domain-decomposition method," *IEEE Trans. Microwave Theory Tech.*, vol. 45, pp. 1179–1184, Aug. 1997.
[10] E. A. Dengi and R. A. Rohrer, "Boundary element method macromodels for 2-D hierarchical capacitance extraction," in *Proc. ACM/IEEE Design Automation Conf.*, 1998, pp. 218–223.
[11] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 856–869, 1986.

[12] S. A. Vavasis, "Preconditioning for boundary integral equations," *SIAM J. Matrix Anal. Appl.*, vol. 13, no. 3, pp. 905–925, 1992.
[13] K. Nabors, S. Kim, J. White, and S. Senturia, *FastCap User's Guide*. Cambridge, MA: MIT Press, 1992.
[14] M. Bachtold, J. G. Korvink, and H. Baltes, "Enhanced multipole acceleration technique for the solution of large poisson computations," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 1541–1546, Dec. 1996.
[15] J. R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 1059–1072, Oct. 1997.

Wenjian Yu (S'01–M'04) was born in Nanchang, China, in 1977. He received the B.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 1999 and 2003, respectively, both with the highest honors.

Since August 2003, he has been a Research Assistant with the Department of Computer Science and Technology, Tsinghua University. His research focused on the fast computation of three-dimensional interconnect capacitance and resistance in very large scale integration (VLSI) circuits, using the BEM. His current research interests include the parasitic parameter extraction of interconnects in VLSI circuits, the direct boundary element analysis of electromagnetic field, and modeling and simulation of VLSI interconnects. He has authored or coauthored over 20 technical papers in refereed journals or conferences.

Dr. Yu is a member of the IEEE Computer Society. He has served as a reviewer for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He was the recipient of the Microsoft Fellowship Award in 2002.

Zeyi Wang (M'94) received the Computational Mathematics degree from the Xian Jiaotong University, Xian, China, in 1965.

Since 1965, he has been with Tsinghua University, Beijing, China, where he is currently a Professor with the Department of Computer Science and Technology. From 1987 to 1988, he was a Visiting Scholar with Stanford University, where he was involved with three-dimensional device simulation on a parallel computer. His main research interests are the applications and research of the numerical methods, including the parallel computations in the areas of very large scale integration computer-aided design such as circuit analysis, device simulation, and parasitic interconnect parameter extraction.