

# Short Papers

## Oct-Tree-Based Multilevel Low-Rank Decomposition Algorithm for Rapid 3-D Parasitic Extraction

Dipanjana Gope and Vikram Jandhyala

**Abstract**—Fast parasitic extraction is an integral part of high-speed microelectronic simulation at the package and on-chip level. Integral equation methods and related fast solvers for the iterative solution of the resulting dense matrix systems have enabled linear time complexity and memory usage. However, these methods tend to have large disparities between setup and matrix-vector product times that affect their efficiency when applied to multiple excitation problems, i.e., problems with a large number of nets. For example, FastCap, which is based on the fast multipole method, has a significantly faster setup time than the multilevel QR decomposition-based IES<sup>3</sup>, but relatively slow matrix-vector products. In this paper, we present a novel oct-tree-based QR compression technique for fast iterative solution. The regular cube structure of the fast multipole method and the QR compression scheme for interaction submatrices as in IES<sup>3</sup> are combined to achieve a predetermined compressible matrix-block structure and, consequently, superior memory, setup, and solve time efficiencies.

**Index Terms**—Low-rank-decomposition, oct-tree multilevel hierarchy, parasitic extraction.

### I. INTRODUCTION

In deep submicron technology, as the trace width decreases, the height-to-width ratio is increased to maintain low wire resistance. The larger trace aspect ratio along with the reduction in trace spacing leads to increasing parasitic capacitances [1]–[3], such that the connectivity delay becomes the dominant delay [4]. At the same time, the rise in clock frequencies render the estimation of interconnect delay and, hence, the associated parasitic quantities extremely crucial for modern day circuit design [5], [6]. Due to the complexity of on-chip design structures, numerical techniques that utilize field solutions for parasitic extraction are preferred, to achieve maximum accuracy [7].

Among the existing numerical tools, a surface-based integral equation methodology such as the method of moments (MoM) [8] is ideally suited to address the problem. It leads to a well-conditioned system with reduced size, compared to volumetric methods [9], but the system of equations generated is inherently dense, thereby creating a time and memory bottleneck [10]–[14]. Several fast iterative techniques have been developed to efficiently store and solve an MoM system. All these methods, including QR-based approaches [10]–[12], fast multipole methods (FMMs) [13], and fast Fourier transform (FFT)-based techniques [14] accelerate matrix-vector products and, therefore, expedite the Krylov-subspace iterative solution [15]. The memory requirement and the setup time is reduced from  $O(N^2)$  to  $O(N)$  or  $O(N \log N)$  and the solve time is reduced from  $O(N^3)$  as in Gaussian elimination

or  $O(N^2) \times p \times r$  as in a regular iterative solver to  $O(N) \times p \times r$  or  $O(N \log N) \times p \times r$ , where  $N$  is the number of degrees of freedom, and  $p$  is the number of iterations for convergence per right-hand side (RHS), and  $r$  is the number of RHS vectors.

The QR-based fast iterative solver (IES<sup>3</sup>) is particularly attractive for circuit problems. The approach is based on the utilization of the low-rank property of MoM submatrices under a user-specified tolerance. The rank of a submatrix obtained by such a tolerance-based decomposition process is referred to as the epsilon-rank. This low-epsilon-rank decomposition is achieved by singular-value decomposition (SVD) or the modified gram schmidt method [16]. Unlike fast multipole methods, it is independent of the kernel (Green's function), and can be applied directly to multilayered dielectric cases [17] without increasing the size of the problem [10]. Even in terms of free-space capacitance extraction, IES<sup>3</sup> has been demonstrated as being more efficient in terms of memory and solve time. However, in absolute terms, the setup cost has a high value, owing to complexities in the underlying binary tree decomposition in IES<sup>3</sup>.

In this paper, we present a predetermined interaction list supported oct tree (PILOT)-based QR algorithm that greatly reduces the setup time while maintaining the memory and solve time efficiency of rank-map binary tree QR (RMBT-QR), which is a prototype implementation based on the same principles as IES<sup>3</sup>. PILOT exploits the properties of a multilevel oct-tree implementation used in FMMs to create a predetermined set of interaction lists, thereby reducing the setup time considerably. In short, the regular oct-tree structure of FMM and the compression efficiency of QR are combined together to yield an enhanced efficiency capacitance extraction algorithm. Though not discussed in this paper, the compression scheme is amenable to full-wave multi-layered dielectric kernel solutions for electrically-small structures and is also stable for these problems unlike traditional full-wave FMM techniques.

The paper is organized as follows. Section II introduces the integral equation formulation for an MoM-based parasitic capacitance extraction problem. In Section III, the algorithm of PILOT-QR with its associated components is discussed in detail. Performance comparisons of PILOT, RMBT-QR and FastCap [13] are presented in Section IV. Section V concludes the paper.

### II. INTEGRAL EQUATION

Capacitance problems formulated using MoM are solved using the integral form of the Poisson's equation as follows:

$$\nabla^2 \phi(\mathbf{r}) = -\frac{\rho(\mathbf{r})}{\epsilon} \quad (1)$$

relating potential  $\phi$  and charge density  $\rho$ . The discretization of the geometry into  $N$  basis functions results in a matrix system of the form

$$\bar{\mathbf{Z}}\mathbf{I} = \mathbf{B} \quad (2)$$

where the  $N \times N$  MoM matrix  $\bar{\mathbf{Z}}$  is a dense Green's function matrix,  $\mathbf{I}$  represents the unknown coefficients of known basis functions for charge density, and  $\mathbf{B}$  represents the known potential excitations on each basis function. Each element of the MoM matrix denotes the

Manuscript received July 17, 2003; revised December 15, 2003. This work was supported in part by the DARPA/MTO NeoCAD program under Grant N66001-01-1-8920, in part by a National Science Foundation Career Award under Grant ECS-0093102, in part by the National Science Foundation/Semiconductor Research Corporation Joint Initiative on Mixed-Signal Electronic Technologies, and in part by the Ansoft Corporation. This paper was recommended by Associate Editor S. Saxena.

The authors are with the Electrical Engineering Department, University of Washington, Seattle, WA 98195 USA (e-mail: dips@u.washington.edu; jandhyala@ee.washington.edu).

Digital Object Identifier 10.1109/TCAD.2004.836723

interaction between a testing and a basis function and is written as follows:

$$\bar{\mathbf{Z}}(j, i) = \int_{S_j} ds t_j(\mathbf{r}) \int_{S_i} ds' g(\mathbf{r}, \mathbf{r}') f_i(\mathbf{r}') \quad (3)$$

where  $t$  and  $f$  are the testing and basis functions, respectively,  $S$  denotes function domain, and  $g(\mathbf{r}, \mathbf{r}')$  is the relevant Green's function.

In the electrostatic case for  $P$  disconnected conductors, the charge-voltage relation is given by

$$\bar{\mathbf{C}}\mathbf{V} = \mathbf{Q} \quad (4)$$

where  $\bar{\mathbf{C}}$  is the  $P \times P$  capacitance matrix,  $\mathbf{V}$  denotes the potential on each conductor and  $\mathbf{Q}$  is the resultant charge on the conductors. The  $j$ th column of  $\bar{\mathbf{C}}$  is obtained by enforcing a voltage of 1 V on the  $j$ th conductor, 0 V on all other conductors in (2)

$$\begin{aligned} B_i &= 1, & i \in \text{Conductor}_j \\ &= 0, & \text{otherwise.} \end{aligned} \quad (5)$$

Then, the above system is solved, and charge density over each conductor is integrated

$$C_{ij} = \sum_k \mathbf{I}_k \quad k \in \text{Conductor}_i \quad (6)$$

The  $N \times N$  system of (2) is, therefore, solved  $P$  times to obtain the entire capacitance matrix.

### III. OCT-TREE-BASED QR-COMPRESSION ALGORITHM

The presented PILOT QR algorithm develops a predetermined matrix structure for arbitrary 3D geometries that ensures efficient compression. The algorithm has 4 main constituents:

#### A. Oct-Tree Spatial Decomposition in Three-Dimensions (3-D)

The algorithm is based on maintaining a regular geometric pattern of cells. The best combination, which yields a regular cell pattern, is *loosely bounded, spatially balanced* decomposition into *orthants* [18]. Note that, as in FMM, empty cells are ignored.

The starting cell  $c_0^0$  is the smallest cube that encloses the entire geometry. The superscript indicates the level of decomposition to which the cube belongs to and the subscript denotes the cube number in that level. Each cell is then recursively decomposed into a maximum of eight cubes in 3-D, depending on the distribution of basis functions. Thus, each cube  $c_i^l$ , which is the  $i$ th cube at level  $l$  is decomposed by spatially balanced splits along each coordinate,  $x$ ,  $y$ , and  $z$

$$\begin{aligned} \text{split}_x &= \frac{x_{\max} + x_{\min}}{2} \\ \text{split}_y &= \frac{y_{\max} + y_{\min}}{2} \\ \text{split}_z &= \frac{z_{\max} + z_{\min}}{2} \end{aligned} \quad (7)$$

where  $\text{split}_x$ ,  $\text{split}_y$ , and  $\text{split}_z$  are the split positions in the three orthogonal directions and  $x_{\max}$ ,  $x_{\min}$ ,  $y_{\max}$ ,  $y_{\min}$ ,  $z_{\max}$ , and  $z_{\min}$  are the bounding coordinates of the cube. Each cube  $c_j^{l+1}$  resulting from this decomposition is called a child of  $c_i^l$  and the latter is denoted as the parent of  $c_j^{l+1}$

$$P_{c_j^{l+1}} = c_i^l. \quad (8)$$

All the child cubes of  $c_i^l$  are siblings of each other, where a sibling set is defined as follows:

$$S_{c_j^{l+1}} = \left\{ c_k^{l+1} \forall k | P_{c_k^{l+1}} = P_{c_j^{l+1}} \right\}. \quad (9)$$

At each level, the generated cells are identical cubes and the pattern repeats across levels. For 3-D arbitrarily shaped geometries, the cell data structure is in the form of an oct tree. The geometric decomposition is, hence, exactly similar to that of multilevel FMM [13] and, hence, its interaction scheme can be leveraged in the presented algorithm.

#### B. Basic Multilevel Interaction List

Every cube  $c_i^l \forall i, l | 0 \leq l \leq l_c; 0 \leq i < N_c^l$ , where  $l_c$  is the total number of levels and  $N_c^l$  is the total number of cubes at level  $l$ , has a nearest neighbor list  $K_{c_i^l}$  and an interaction list  $I_{c_i^l}$ . The nearest neighbor list, is defined as

$$K_{c_i^l} = \left\{ c_j^l | c_j^l \text{ is in the same level as } c_i^l \text{ and has at least one contact point with } c_i^l \right\}. \quad (10)$$

Consequently, the interaction list is defined as

$$I_{c_i^l} = \left\{ c_j^l | P_{c_j^{l+1}} \in k_{p_{c_i^l}}; c_j^l \notin K_{c_i^l} \right\}. \quad (11)$$

In multilevel FMM, interactions between the testing functions of cube  $c_i^l$  with basis functions of cubes belonging to  $I_{c_i^l}$  are constructed using multipole expansion. In other words, multipoles are used to construct  $T(c_j^l, c_i^l) \forall j | c_j^l \in I_{c_i^l}$  where  $T(c_j^l, c_i^l)$  denotes the interaction between testing functions of  $c_j^l$  and basis functions of  $c_i^l$ . Furthermore, multipole expansions are formed only once for  $c_i^l$  and used for all its interactions with other cubes. Direct Green's function computations are used only at the finest level to construct  $T(c_j^{l_c}, c_i^{l_c}) \forall j | c_j^{l_c} \in K_{c_i^{l_c}}$ . Since PILOT does not explicitly require cubical regions but simply deals with interaction matrices, there is scope for further compression by combining cubes in  $I_{c_i^l}$  in an a priori manner into a new interaction list called the merged interaction list (MIL).

#### C. MIL

It is observed that the interaction lists of siblings share many common cubes

$$I_S = \bigcap I_{c_i^l} \neq \emptyset \quad \forall i | c_i^l \in S_{c_j^l}. \quad (12)$$

The common cubes in the interaction lists of the siblings are denoted by  $I_S$ . For visualization purposes, the two-dimensional (2-D) common interaction shell is illustrated in Fig. 1, though our algorithm is designed for 3-D geometries.

It is, therefore, possible to group source cubes and observer cubes of different interaction lists in order to compress larger matrices to low epsilon-ranks and thereby gain in terms of overall compressibility. It must be noted that the common interaction list does not directly translate into a merged interaction because the epsilon-rank of such an interaction submatrix will not in general be low. The common interaction list is decomposed into disjointed parts such that the overall compression is optimized. Each such disjointed part is an interaction between grouped source cubes and observer cubes and forms an entry of the MIL denoted as  $\mu$ . A  $\mu$  can be expressed as a combination of multilevel FMM cube-to-cube interactions

$$\mu_k = \left\{ T_p(c_j^l, c_i^l) \right\} \quad \forall p | 1 \leq p \leq n_g \quad (13)$$

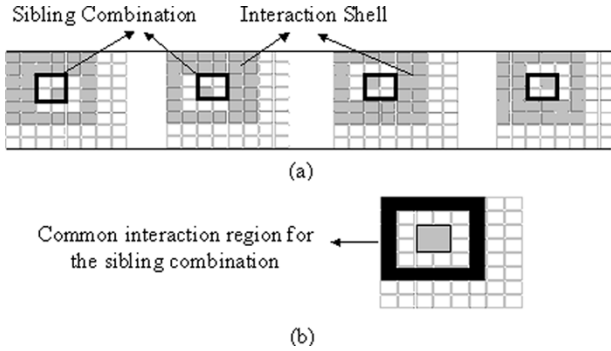


Fig. 1. (a) Individual interaction shells of each cube belonging to the sibling combination. (b) Common interaction shell for the sibling combination formed by the intersection of individual interaction regions of cubes belonging to the sibling combination. For visualization purpose 2-D shells are illustrated. Similar common-interaction regions exist for 3-D geometries.

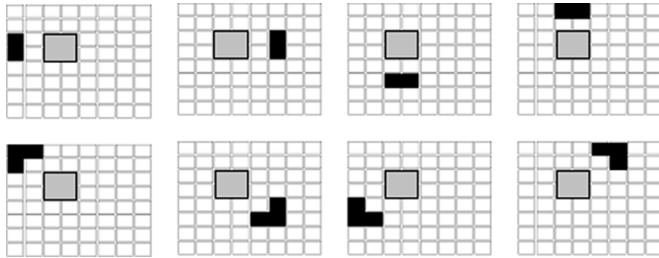


Fig. 2. Merged interaction-list entries corresponding to the common interaction region of Fig. 6(b). Each such entry gives rise to a low epsilon-ranked matrix block.

where  $n_g$  is the number of FMM interactions grouped. Higher compression is achieved since a larger matrix is compressed to a low epsilon-rank under the same tolerance

$$(m_{\mu_k} + n_{\mu_k})r_{\mu_k} < \sum_{i=1}^{n_g} (m_i + n_i)r_i \quad (14)$$

where  $m$ ,  $n$ , and  $r$  denote the number of rows, number of columns, and the epsilon-rank of a submatrix. The subscript  $i$  denotes a regular multilevel interaction list entry that is now a constituent of the MIL. Fig. 2 demonstrates the decomposition of the common interaction list of Fig. 1(b) into merged interactions. Further compression is possible considering common interaction lists for pairs of siblings. Thus, the regular interaction list is replaced by the merged interaction list, which has fewer interactions to consider and larger low epsilon-ranked matrices to compress with the same tolerance.

PILOT supports MIL for both 3-D and 2-D geometries. However, for visualization purposes, the 2-D MIL construction is illustrated in detail. For 2-D geometries, each cube has a maximum of 27 cubes in its interaction list and this pattern repeats for all cubes across levels. In PILOT, the interaction lists of four siblings are replaced by 16 merged interaction entries as demonstrated in Table I. The MIL entries, along with the expected ranks, are setup as a one-time process for a given kernel. The expected epsilon-rank is the maximum rank observed for sources and observers randomly placed in an MIL setup. The expected epsilon-ranks supplied in Table I apply for the free space Green's Function kernel. The cube numbering used in the process is illustrated in Fig. 3.

Though there are 16 entries in the MIL, there are only three different types of interactions to be evaluated and the rest could be derived from symmetry considerations. The same MIL pattern is valid for sibling

TABLE I  
MIL ENTRIES WITH EXPECTED EPSILON-RANKS

MIL No.	Observer Cubes	Source Cubes	Expected Eps-Rank
1	1, 2, 3, 4	5, 6, 8	6
2	1, 2, 3, 4	13, 14, 15	6
3	1, 2, 3, 4	25, 27, 28	6
4	1, 2, 3, 4	34, 35, 36	6
5	1, 2, 3, 4	9, 10	6
6	1, 2, 3, 4	17, 20	6
7	1, 2, 3, 4	22, 23	6
8	1, 2, 3, 4	31, 32	6
9	1, 2	26, 29, 30	7
10	3, 4	11, 12, 16	7
11	2, 3	7, 18, 19	7
12	1, 4	21, 24, 33	7
13	1	16	5
14	2	33	5
15	3	26	5
16	4	7	5

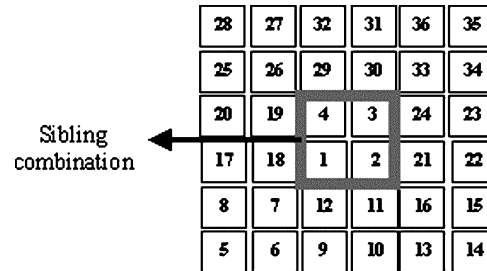


Fig. 3. Siblings, and cubes belonging to their interaction shells are numbered. The MIL entry table (Table I) follows the numbering scheme demonstrated in this figure.

pairs across all levels. The MIL is thus an accurate replacement of the rank-map used in RMBT-QR and leads to a predetermined tree structure. The 2-D MIL shell structure is invariant for multilayered dielectric Green's functions, for structures parallel to the layers and only the expected epsilon-ranks change. A similar MIL has been derived for 3-D geometries with 40 entries, with only five unique entries owing to symmetry.

The construction of the MIL is a one-time process for a given multilayered dielectric environment and due to the regular pattern of the cube structure, only a few interactions need to be considered for epsilon-rank evaluation. The MIL determines the final tree structure for any geometry under consideration.

#### D. QR Compression of MIL Entries

MoM submatrices pertaining to interactions of the MIL are compressed by forming QRs from samples. Consider  $n$  source basis functions  $f_i$  defined over domain  $S_i$  for  $i = 1, 2, \dots, n$ , such that  $S_i \in R_{src}$ , where  $R_{src}$  is the region of space inside an MIL entry source group. Similarly, consider  $m$  testing functions whose domains belong to region  $R_{obs}$ , which is delimited by the MIL entry observer group. Let the sub-matrix  $\bar{Z}_{m \times n}^{sub}$  of the full MoM matrix  $\bar{Z}$  represent the interactions between the basis and the testing functions through the designated Green's function  $g(\mathbf{r}, \mathbf{r}')$ . Green's functions encountered in capacitance extraction problems including those for multilayered dielectrics vary smoothly with distance [10]. Therefore, the column of  $\bar{Z}^{sub}$  pertaining to the interaction of  $f_i$  with all testing functions is closely related to other columns that capture similar interactions for  $f_j \forall j | S_j$  is in the neighborhood of  $S_i$ .

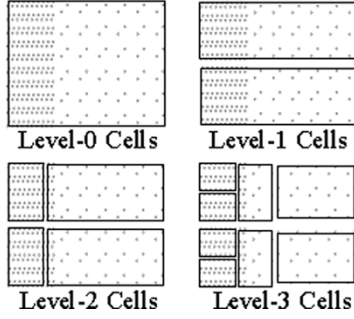


Fig. 4. Levels of tight bound  $k$ - $d$  tree decomposition. As can be seen, rectangular (parallelepipeds in 3-D) cells obtained from subdivision can have arbitrary dimensions depending on the distribution of basis functions.

Using the MGS process and a user-specified tolerance  $\varepsilon$ ,  $\bar{\mathbf{Z}}^{\text{sub}}$  can be decomposed into a unitary matrix  $\bar{\mathbf{Q}}_{m \times r}$  and an upper triangular matrix  $\bar{\mathbf{R}}_{r \times n}$  such that

$$\frac{\|\bar{\mathbf{Z}}^{\text{sub}} - \bar{\mathbf{Q}}\bar{\mathbf{R}}\|}{\|\bar{\mathbf{Z}}^{\text{sub}}\|} < \varepsilon \quad (15)$$

where

$$\bar{\mathbf{Q}}^H \bar{\mathbf{Q}} = \bar{\mathbf{I}} \quad (16)$$

and the matrix norm  $\|\bar{\mathbf{X}}\|$  is defined as the maximum singular value of the matrix  $\bar{\mathbf{X}}$ .

The QR decomposition of  $\bar{\mathbf{Z}}^{\text{sub}}$ , as shown above, requires the construction of the entire submatrix. With such a scheme the setup time for an  $N \times N$  MoM matrix will be  $O(N^2)$ . However it is possible to obtain the compressed form of the MoM matrix in subquadratic time [11].

#### IV. ADVANTAGES OVER EXISTING QR ALGORITHM

The absolute setup time for RMBT-QR is large owing to the following reasons. The setup cost of the algorithm [11] is largely controlled by the accuracy of rank map predictions. An accurate and exhaustive rank map would preclude the necessity for unnecessary merges and splits and the optimum tree structure would be achieved without any backtracking or refinement within the tree structure. However, a foolproof rank-map is difficult if not impossible to construct owing to the fact that the algorithm can lead to cells with any shape and size as can be seen in Fig. 4. It is unfeasible to cover the infinite combinations of parameters, thus introducing a scope of error in the rank map. A conservative rank map will require more merges, whereas a liberal rank map will induce wastage of time by constructing unacceptable QRs, which are then discarded. Thus, the setup time is largely increased by the variability of the tree structure and the resulting backtracking and refinement.

In our work, we significantly reduce the setup time without compromising on memory or solve-time compression. The new algorithm exploits the regularity of cell size, shape, and location of a spatially balanced oct tree as in a multilevel FMM algorithm. By recourse to the FMM interaction list and by adding a few additional features to maximize compression, a regular and compressed interaction pattern is generated. The number of different interactions to be evaluated is finite and small and, therefore, an exhaustive and accurate *a priori* epsilon-rank estimation is possible. PILOT therefore incorporates the best features of the regular cube structure of multilevel FMM and the kernel-independent low-epsilon-rank compression of IES<sup>3</sup>.

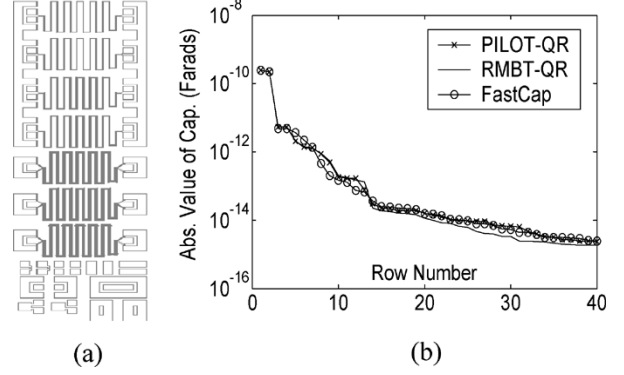


Fig. 5. (a) Structure under simulation with meander lines, coplanar waveguides, pads, etc. (b) The absolute values of the first column of the capacitance matrix for the structure in Fig. 5(a) are plotted for all three algorithms. Even for relatively small off-diagonal terms the results agree remarkably well.

TABLE II  
ACCURACY, MEMORY, AND TIME

Algorithm	$\bar{\epsilon}_{11}$	$\bar{\epsilon}_{1r}$	Memory	Time
PILOT	2.413e-10	-5.259e-12	365MB	19min
RMBT-QR	2.413e-10	-5.261e-12	380MB	27min
FastCap	2.412e-10	-5.257e-12	500MB	30min

#### V. SIMULATION RESULTS

In this section, simulation results are presented to demonstrate the accuracy and time and memory efficiency of the PILOT algorithm. For a comparative analysis, results obtained from RMBT-QR, which is based on IES<sup>3</sup>, and FastCap are presented side-by-side. Analytic integration scheme [20] is used for near-field integrals and discretization scheme is collocation. A QR decomposition tolerance of 1e-3 is used for both PILOT and RMBT-QR, whereas for FastCap the adaptive algorithm with multipole order of two is employed. An absolute residual of 1e-3 is used for the Krylov subspace iterative solution. Diagonal preconditioners are used and the number of iterations required is observed to be the same for the same problem with all the iterative algorithms. All tests were run on a processor with 4-GB RAM and 1.6-GHz CPU speed.

In the first example, the capacitance matrix of a structure consisting of meander lines, coplanar waveguides and pads as in Fig. 5(a) is simulated. The surface of the structure is then meshed into triangular patches. Piecewise constant basis functions are defined on each triangular patch. The total number of patches for this particular problem is 0.113 million and the number of nets involved is 40. The absolute values of the 40 entries of the first column as obtained from all the three algorithms are plotted in Fig. 5(b). The elements are found to closely match the ones obtained by solving the system with RMBT-QR or FastCap even for relatively small off-diagonal terms. In Table II, selected values of the capacitance matrix as obtained with all three algorithms are presented as well as the comparative time and memory required for the process. The total time includes the entire setup and iteration time for six excitations.

The next example is that of a three-layer interconnect structure embedded in a  $10 \times 4 \times 1$ - $\mu\text{m}$  space. Each interconnect is  $0.1 \times 0.1 \mu\text{m}$  in cross section. The separations between the layers are  $0.4 \mu\text{m}$  and the minimum separation between traces on the same layer is  $0.15 \mu\text{m}$ . The number of triangular patches is varied from 2000 to 0.6 million and the problem is solved with ten of the interconnects as active nets and the rest as floating conductors. The time and memory requirements

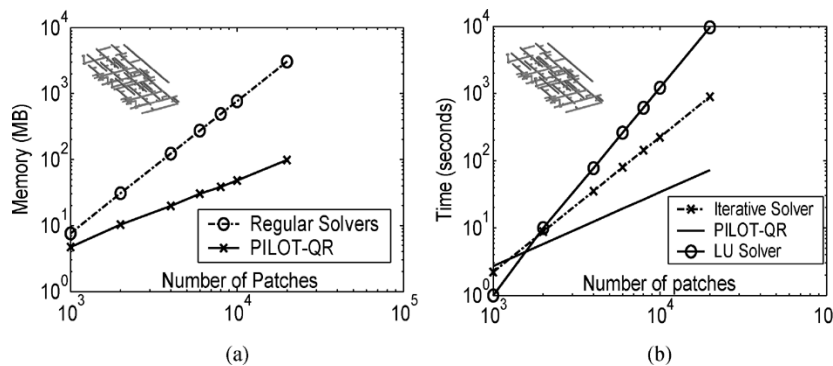


Fig. 6. Performance comparison of PILOT with regular direct and iterative solvers for an interconnect bus structure. (a) Memory requirement in MB and (b) Time requirement in minutes for setup and 10 RHS solutions.

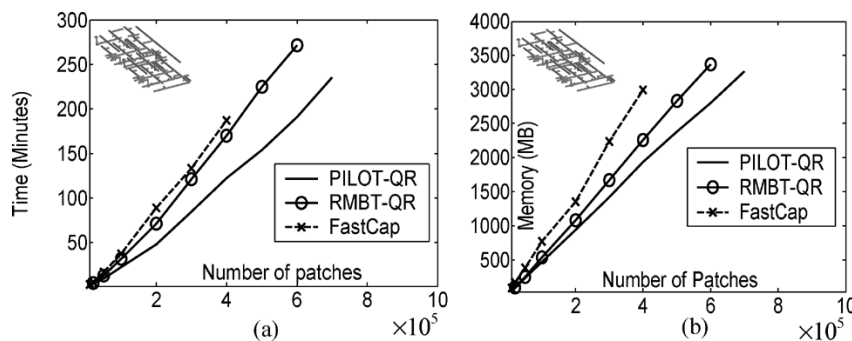


Fig. 7. Performance comparison of PILOT with existing fast solvers for an interconnect bus structure. (a) Memory requirement in MB and (b) time requirement in minutes for setup and 10 RHS solutions.

of PILOT are first compared to those of regular direct and iterative solvers in Fig. 6. It is observed that performance of PILOT is orders of magnitude superior for large number of patches. The efficiency of the algorithm is then compared to those of RMBT-QR and FastCap in Fig. 7. Due to limitations of RAM space, FastCap results are available for up to 0.3 million patches. As expected the time required by PILOT is least among the algorithms: it beats FastCap in faster matrix vector products and RMBT-QR in faster setup. Memory-wise PILOT is superior to FastCap due to greater compression. Also it is observed that PILOT, on occasions, requires less memory compared to RMBT-QR. This is because of the fact that the latter, inspite of multiple splits and merges often fails to attain the optimal tree structure compared to the physics-based interaction list in PILOT, leading to less compression than expected.

The third example demonstrates the relative advantage of QR methods as compared to FMM-based algorithms for higher number of nets and consequently larger number of matrix vector products. A package structure with 14 leads as illustrated in Fig. 8(a) is considered. The surface is meshed with 0.101 million patches and then solved for increasing number of right hand sides (1–14). The time requirements are plotted in Fig. 8(b). The constant offset between the plots of PILOT and RMBT-QR is due to the superior one-time setup cost. The memory required for the process by PILOT is 441 MB, by RMBT-QR is 445 MB and by FastCap is 700 MB.

The last example demonstrates the computing efficiency of the PILOT algorithm for very large-scale problems. The structure of example 1 is repeated in a  $10 \times 3$  array, and shown in Fig. 9. The entire structure is meshed with 0.913 million patches. The problem is setup and solved for three different excitations. In the 4 GB of memory available, only PILOT was able to fit and run this example, and required 3.3 GB, 48 min for setup, and 90 min for solution with three RHS.

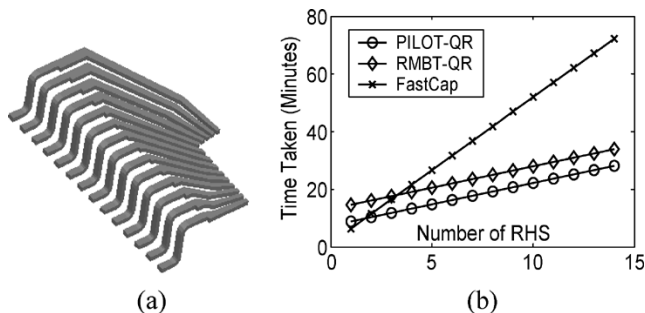


Fig. 8. Performance comparison showing efficiency of QR-based algorithms for increasing number of RHS (a) multipin structure considered for simulation consisting of 14 pins (b) setup and solve time for increasing number of RHS.

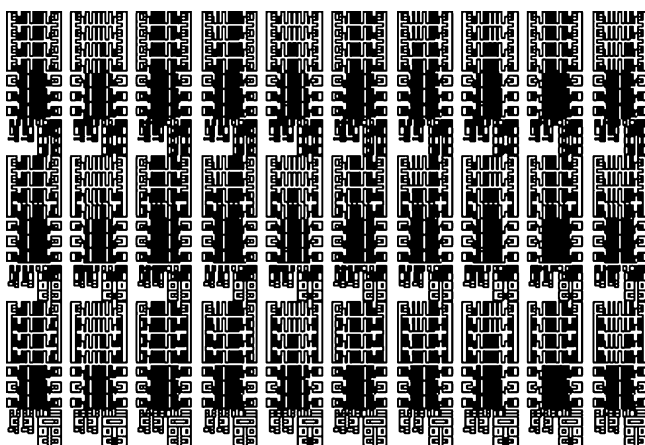


Fig. 9. Illustrated structure is generated by placing the geometry of Fig. 5(a) in a  $10 \times 3$  array. The surface is meshed with 0.913 triangular patches.

## VI. CONCLUSION

In this paper, we presented PILOT, an improved algorithm for QR-compression-based fast iterative solver and apply it to parasitic capacitance extraction problems modeled on surface-based method of moments. The regular geometry decomposition scheme of FMM and improved compression capability of IES<sup>3</sup> are combined together to yield an algorithm with superior efficiency. From the IES<sup>3</sup> perspective, the concept of rank-map and fine-tuning through merges and splits is replaced by the *a priori* merged interaction list, enabled through exploitation of the regular oct-tree structure in FMM. As a result, accurate prediction of predetermined low epsilon-rank blocks is possible and this, in turn, reduces the setup time of the process. Compared to the FMM interaction list, greater compression is achieved through merging source sibling cubes and observer cubes in their interaction list to form the merged interaction list. The resultant blocks in the list are then QR-compressed. The merged interaction list, like the rank-map of IES<sup>3</sup> is created only once for a given Green's function. However, due to the regular pattern of cubes, far fewer epsilon-rank evaluations are required to construct the list compared to the original binary-tree rank map.

The simulation results presented demonstrate the relative efficiency of the PILOT algorithm compared to existing QR methods and FastCap, in terms of setup time, memory, and matrix-vector products for large number of excitations. While we have discussed PILOT only in the context to parasitic capacitance extraction, continuing work focuses on its application to full-wave kernels in multilayered media for electrically small structures where classical FMM techniques break down.

## ACKNOWLEDGMENT

The authors thank Prof. M. Gu, Department of Mathematics, University of California, Berkeley, for stimulating discussions on this topic, and the reviewers for their suggestions for improvement.

## REFERENCES

- [1] G. Serval and D. Deschacht, "On-chip crosstalk evaluation between adjacent interconnections," in *Proc. 7th IEEE Int. Conf. Electron., Circuits, Syst.*, vol. 2, 2000, pp. 827–834.
- [2] Y. Im and K. Roy, "A novel high-performance predictable circuit architecture for the deep submicron era," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2000, pp. 503–506.
- [3] M. Kuhlmann, S. S. Sapatnekar, and K. K. Parhi, "Efficient crosstalk estimation," in *Proc. Int. Conf. Comput. Design*, 1999, pp. 266–272.
- [4] E. A. Dengi and R. A. Rohrer, "On-chip interconnect modeling technologies," in *Proc. IEEE 6th Topical Meeting Elect. Perform. Electron. Packag.*, 1997, p. 41.
- [5] K. Kundert, H. Chang, D. Jefferies, G. Lamant, E. Malavasi, and F. Sendig, "Design of mixed-signal systems-on-a-chip," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 1561–1571, Dec. 2000.
- [6] N. K. Verghese, T. J. Schmerbech, and D. J. Allstot, *Simulation Techniques and Solutions for Mixed-Signal Coupling in Integrated Circuits*. Norwell, MA: Kluwer, 1995, p. 160.
- [7] A. Husain, "Models for interconnect capacitance extraction," in *Proc. Int. Symp. Quality Electron. Design*, 2001, pp. 167–172.
- [8] R. F. Harrington, *Field Computation by Moment Methods*. New York: IEEE Press, 1991.
- [9] T.-Y. Chou and Z. J. Cendes, "Capacitance calculation of IC packages using the finite element method and planes of symmetry," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 1159–1166, Sept. 1994.
- [10] S. Kapur and D. E. Long, "IES<sup>3</sup>: efficient electrostatic and electromagnetic solution," *IEEE Comput. Sci. Eng.*, vol. 5, pp. 60–67, Oct./Dec. 1998.
- [11] S. Kapur and D. Long, "IES<sup>3</sup>: A fast integral equation solver for efficient 3-dimensional extraction," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 448–455.

- [12] S. Kapur, D. Long, and J. Zhao, "Efficient fullwave simulation in layered lossy medium," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1998, pp. 211–214.
- [13] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 1447–1459, Nov. 1991.
- [14] J. R. Phillips and J. 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.
- [15] H. A. van der Vorst, "Krylov subspace iteration," *Comput. Sci. Eng.*, vol. 2, no. 1, pp. 32–37, 2000.
- [16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 2nd ed. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
- [17] A. Cangelaris and Y. Ling, "Rapid calculation of electrostatic green's functions in layered dielectrics," *IEEE Trans. Magn.*, vol. 37, pp. 3133–3136, Sept. 2001.
- [18] R. J. Anderson, "Tree data-structures for N-body simulation," *SIAM J. Comput.*, vol. 28, no. 6, pp. 1923–1940.
- [19] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Trans. Microwave Theory Tech.*, vol. 29, pp. 76–82, Feb. 1973.
- [20] H. Wilton, S. Rao, A. Glisson, D. Schaubert, O. Al-Bundak, and C. Butler, "Potential integrals for uniform and linear source distributions on polygonal and polyhedral domains," *IEEE Trans. Antennas Propagat.*, vol. 32, pp. 276–281, Mar. 1984.

## Accurate and Efficient Modeling of SOI MOSFET With Technology Independent Neural Networks

S. Hatami, M. Y. Azizi, H. R. Bahrami, D. Motavalizadeh, and A. Afzali-Kusha

**Abstract**—This paper presents neural network (NN) approaches for modeling the  $I$ - $V$  characteristics of silicon-on-insulator MOSFETs. The modeling approach is technology independent, fast, and accurate, which makes it suitable for circuit simulators. In the model, two different NN architectures, namely, multilayer perceptron and generalized radial basis function, are used and compared. To increase the training efficiency of the NN, both modular and region partitioning methods have been proposed and utilized. In addition, two approaches for obtaining the transconductance and output conductance of the device are discussed. The first approach makes use of an NN for the conductances, while the second uses the numerical differentiation of the  $I$ - $V$  results. To confirm the accuracy of the model, the drain-current characteristics as well as conductances obtained by the model are compared to the simulation data for the points where the NNs are not trained. The comparison shows excellent agreements with relative errors of around 1% over a wide range of drain and gate voltages as well as channel lengths and widths.

**Index Terms**—Circuit simulation, fully depleted (FD),  $I$ - $V$  characteristic, neural network (NN) modeling, partially depleted (PD), silicon-on-insulator (SOI) modeling, technology independent modeling, unified modeling.

## I. INTRODUCTION

MOSFET devices in silicon-on-insulator (SOI) technology have many advantages over bulk counterparts, such as lower parasitic capacitance and radiation hardness. The silicon layer on the oxide

Manuscript received March 18, 2003; revised December 19, 2003. This paper was recommended by Associate Editor C.-J. R. Shi.

The authors are with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Tehran, Tehran, Iran (e-mail: hatami\_safar@yahoo.com; y.azizi@ece.ut.ac.ir; hrbahrami@yahoo.com; motavalizadeh@yahoo.com; afzali5@gmail.com).

Digital Object Identifier 10.1109/TCAD.2004.836725