

# A Fast Parasitic Extractor Based on Low-Rank Multilevel Matrix Compression for Conductor and Dielectric Modeling in Microelectronics and MEMS

Dipanjan Gope  
dips@u.washington.edu

Swagato Chakraborty  
swagato@u.washington.edu  
Department of Electrical Engineering,  
Box 352500, University of Washington,  
Seattle, WA-98195.  
Telephone: 206-543-2186.

Vikram Jandhyala  
jandhyala@ee.washington.edu

## ABSTRACT

Parasitic parameter extraction is a crucial issue in Integrated Circuit design. Integral equation based solvers, which guarantee high accuracy, suffer from a time and memory bottleneck arising from the dense matrices generated.

In this paper we present a hybrid FMM-QR algorithm that combines the best features of the Fast Multipole Method and the QR based matrix compression method to achieve faster setup and solve time and lower memory requirements. The method is applied to extract parasitic capacitances from the layout of arbitrarily shaped conductors and dielectrics. Examples demonstrating the accuracy and the superior time and memory performances as compared to existing solvers are also presented.

## Categories and Subject Descriptors

J.6 [Computer-Aided Engineering]: Computer-Aided Design

## General Terms

Algorithms, Performance, Design.

## Keywords

Parasitics, Multilevel, Low-rank, conductors and dielectrics

## 1. INTRODUCTION

In deep-submicron technology diminishing wire spacing and higher aspect ratio structures lead to increased parasitic effects [1]. As a consequence, the interconnect delay may significantly increase and eventually dominate over the gate delay causing erroneous clock-timing that leads to a malfunctioning circuit. Hence proper modeling of interconnects is key for meeting timing requirements in a circuit level simulation of VLSI packages. Also in rapidly growing MEMS application electrostatic analysis of large MEMS array has become extremely important.

Analytic models like charge-based capacitance measurement method [2] often suffer from oversimplification of the geometry

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2004, June 7–11, 2004, San Diego, California, USA  
Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

of interconnects. Stochastic models like the floating random walk method [2] suffer from problems of convergence and the effects of multilayered dielectrics. Therefore to achieve a guaranteed high degree of accuracy a numerical 3D electromagnetic solver is necessitated. The direct application of integral equation solvers leads to the generation of a dense matrix with large number of unknowns, the solution of which presents a time and memory bottleneck.

Existing fast solver algorithms aimed at solving the dense system of integral equations through fast matrix-vector product based iterative solutions include the Fast Multipole Method (FMM) [3], the QR-based method (IES<sup>3</sup>) [4] and the Pre-corrected Fast Fourier Transform (FFT) method [5]. The pre-corrected FFT method degrades in efficiency for non-uniform distribution of basis-functions, which is often encountered in circuit problems, due to the inherent necessity of a uniform global grid. The FMM and the IES<sup>3</sup> algorithms on the other hand are ideally suited for application to arbitrarily shaped circuit structures. However they suffer from discrepancies in the setup and solve times and the memory requirements. The FMM method has a faster setup time but a higher memory requirement and a slower solve time. On the other hand the IES<sup>3</sup> method leads to higher compression and consequently a faster solve time at the cost of a slow setup time. Due to these discrepancies, the applicability of the algorithms become problem specific: QR-based methods are ideally suited for large number of nets which involve a single setup but multiple solution phases, and the FMM method is better applicable for a single or small number of excitations.

In the new algorithm presented here, the multilevel oct-tree structure (common to FMM approaches) and the QR compression technique in IES<sup>3</sup> are combined. Further optimizations in the form of a merged interaction list developed here ensure significantly better all-round performance of the hybrid algorithms. The reasons behind the superior performance of this algorithm as compared to FMM and IES<sup>3</sup> are discussed. The method is then applied to extract parasitic capacitance from the layout of conductors and dielectric structures. Accuracy, memory and timing results are compared with FastCap [3], which is an FMM-based open source code and RMBT-QR (Rank-Map based Binary Tree), which is our prototype implementation of IES<sup>3</sup>. In the last example it is demonstrated that the algorithm can also be applied for the fast computation of electrostatic forces on MEMS structures.

## 2. INTEGRAL EQUATION

The conductor and dielectric interfaces are discretized into panels. Basis functions are defined on the discretized panels and appropriate boundary conditions are enforced on the conductor-to-dielectric and dielectric-to-dielectric interfaces [6]. Let  $S_C$  and  $S_D$  represent the set of discretized panels on the conductor-to-dielectric and dielectric-to-dielectric interfaces. The application of the method of moments formulation leads to the following matrix equation:

$$\begin{pmatrix} \bar{\mathbf{Z}}_{CC} & \bar{\mathbf{Z}}_{CD} \\ \bar{\mathbf{Z}}_{DC} & \bar{\mathbf{Z}}_{DD} \end{pmatrix} \begin{pmatrix} \boldsymbol{\sigma}_C \\ \boldsymbol{\sigma}_D \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ \mathbf{0} \end{pmatrix} \quad (2.1)$$

where,  $\boldsymbol{\sigma}_C$  represents the set of total-charge on  $S_C$ ,  $\boldsymbol{\sigma}_D$  represents the set of total-charge on  $S_D$  and  $\mathbf{V}$  represents the set of potentials on  $S_C$ . The first set of equations ensures equipotentiality on conductor surfaces.

$$\bar{\mathbf{Z}}_{CC/CD}(j,i) = \left\{ \int_{t_j} ds' h_j(\mathbf{r}) \int_{t_i} ds' g(\mathbf{r},\mathbf{r}') f_i(\mathbf{r}') \right\} | t_i \in S_{CD}; t_j \in S_C \quad (2.2)$$

where  $h$  and  $f$  are the testing and basis functions respectively,  $g$  represents the Greens function and  $t$  is the domain of definition of each basis function. The second set of equations enforces the continuity of the normal component of the electric displacement vector  $\mathbf{D}$ , across the dielectric-to-dielectric interfaces:

$$\bar{\mathbf{Z}}_{CD/DD}(j,i) = \left\{ \begin{aligned} & - \int_{t_j} ds h_j(\mathbf{r}) \int_{t_i} \hat{\mathbf{n}} \cdot ds' \nabla g(\mathbf{r},\mathbf{r}') f_i(\mathbf{r}') \\ & + \left[ \frac{\epsilon_r^+ + \epsilon_r^-}{2\epsilon_0(\epsilon_r^+ + \epsilon_r^-)} \right] \delta_{ij} \end{aligned} \right\} | t_i \in S_{CD}; t_j \in S_D \quad (2.3)$$

where  $\delta_{ij}$  is the Kronecker delta function,  $\hat{\mathbf{n}}$  is the outward-pointing unit vector,  $\epsilon_r^+$  and  $\epsilon_r^-$  are the dielectric constants for the 2 sides of the dielectric-to-dielectric interface.

The entries of the capacitance matrix for  $p$  conductors ( $Y_i | i = 1, 2, 3, \dots, p$ ) can be evaluated thereafter by the well-known formulation:

$$\bar{C}_{ij} = \left\{ \int_{t_j} \sigma_f(r') dr' \right\} \quad (2.4)$$

when  $\mathbf{V}_j = 1; \mathbf{V}_{m, m \neq j} = 0 | m = 1, 2, 3, \dots, p$  (2.5)

where  $\sigma_f$  represents the free-charge density on the conductor-dielectric interface and is obtained as:

$$\sigma_f(r) = \epsilon_r(r) \sigma_c(r) \quad (2.6)$$

where  $\epsilon_r$  is the dielectric constant of the dielectric at the conductor-to-dielectric interface.

## 3. MULTILEVEL OCT-TREE BASED QR ALGORITHM

The pre-determined interaction list oct-tree (PILOT) QR algorithm efficiently compresses the method of moments matrix in a multilevel scheme. The algorithm has 4 main constituents:

### 3.1 Oct-tree spatial decomposition in 3D

The basis functions are grouped together by a regular geometric pattern of cells. The best combination, which yields a regular cell pattern, is *loosely bounded, spatially balanced* decomposition into *orthants* (quadrants in 2D and octants in 3D). 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 is associated and the subscript denotes the cube number in that level. Each cell is then recursively decomposed into a maximum of 8 cubes in 3-D, depending on the distribution of basis functions. Thus each cube  $c_i^l$ , which is the  $i^{\text{th}}$  cube at level  $l$  is decomposed by spatially balanced splits along each coordinate,  $x$ ,  $y$  and  $z$ . 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  $c_j^{l+1}$ :

$$P_{c_j^{l+1}} = c_i^l \quad (3.1)$$

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

$$S_{c_i^l} = \{c_k^{l+1} | k | P_{c_i^l} = P_{c_k^{l+1}}\} \quad (3.2)$$

The geometric decomposition is hence exactly similar to that of multilevel FMM and therefore its interaction scheme can be leveraged in the presented algorithm.

### 3.2 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:

$$K_{c_i^l} = \{c_j^l | c_j^l \text{ is in the same level as } c_i^l \text{ and has at least one contact point with } c_i^l\} \quad (3.3)$$

Consequently the interaction list is defined as:

$$I_{c_i^l} = \{c_j^l | P_{c_j^l} \in K_{c_i^l}; c_j^l \notin K_{c_i^l}\} \quad (3.4)$$

The neighbor and the interaction list of a typical cube in level 4 is shown in figure 1:

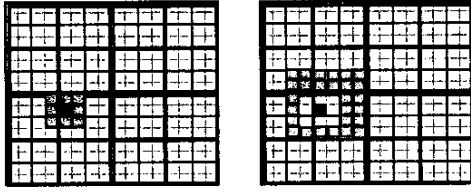


Figure 1: (a) Neighbor list and  
(b) Interaction list for the given cube at level 4.

### 3.3 Merged interaction list

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

$$I_S = \bigcap I_{c_i} \neq \emptyset \quad \forall i | c_i \in S_{c_i} \quad (3.5)$$

The common cubes in the interaction lists of the siblings are denoted by  $I_S$ . For visualization purposes, parts of the 2D common interaction shell is illustrated in figure 2, though our algorithm is designed for 3D geometries.

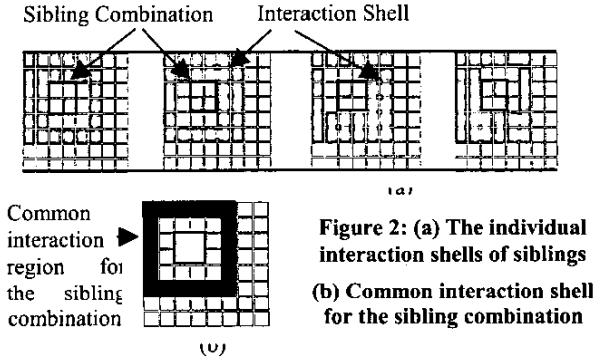


Figure 2: (a) The individual interaction shells of siblings  
(b) Common interaction shell for the sibling combination

It is possible to merge overlapping interaction lists belonging to siblings in order to compress larger matrices to low epsilon-ranks (numerical rank under a user-defined tolerance) and thereby gain in terms of overall compression. It must be noted that the common interaction list does not directly translate into a low-ranked merged interaction. The common interaction list is decomposed into disjoint parts such that the overall compression is optimized. Each such part denoted by  $\mu$  is an interaction between grouped source cubes and observer cubes and forms an entry of the Merged Interaction List (MIL). A  $\mu$  can be expressed as a combination of multilevel oct-tree cube-to-cube interactions:

$$m_k = \{T_p(c_j^l, c_i^l)\} \quad \forall p | 1 \leq p \leq n_g \quad (3.6)$$

where  $T_p(c_j^l, c_i^l)$  denotes the interaction between the source basis functions in cube  $c_j^l$  and testing functions in cube  $c_i^l$  and  $n_g$  is the number of multilevel oct-tree interactions grouped. Higher compression is achieved since a larger matrix is compressed to a low epsilon-rank under the same tolerance:

$$(m_{\mu_k} + n_{\mu_k})r_{\mu_k} < \sum_{i=1}^{n_g} (m_i + n_i)r_i \quad (3.7)$$

where  $m$ ,  $n$  and  $r$  denote the number of rows, number of columns and the epsilon-rank of a sub-matrix. The subscript  $i$  denotes a regular multilevel interaction list entry that is now a constituent of the MIL. Figure 3 demonstrates the decomposition of the common interaction list of figure 2 into merged interactions.

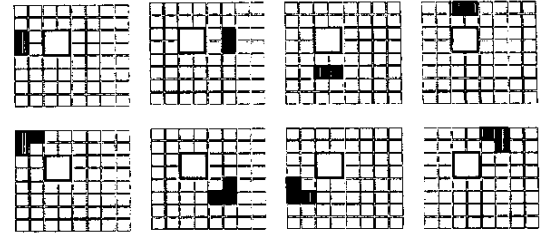


Figure 3: Merged interaction list entries corresponding to the common interaction region of figure 2. Each such entry gives rise to a low-rank matrix block.

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. In PILOT, the interaction lists of 4 siblings in 2D are replaced by 16 merged interaction entries. Each MIL entry constitutes of the grouping information for source cubes and observer cubes and the corresponding expected epsilon-rank. The expected epsilon-rank is the maximum rank observed for sources and observers randomly placed in an MIL setup. The MIL entries along with the expected ranks are setup as a one-time process for a given kernel. Though there are 16 entries in the MIL, there are only 3 different types of interactions to be evaluated and the rest could be derived from symmetry considerations. The same MIL pattern is valid for all sibling pairs across levels. Although for visualization purposes the 2D MIL construction is illustrated in detail, PILOT supports MIL for both 3D and 2D geometries. A similar MIL is derived for 3D geometries with 40 entries, with only 5 unique entries owing to symmetry.

### 3.4 QR compression of MIL entries

MoM sub-matrices pertaining to interactions of the MIL are compressed separately for testing functions belonging to  $S_C$  and  $S_D$  by forming QRs from samples [4]. Consider  $n$  source basis functions  $f_i$  for  $i = 1, 2, \dots, n$  belonging to  $\mu_{k_S}$ , which is the  $k^{\text{th}}$  MIL source group. Similarly, consider  $m$  testing functions belonging to  $S_C$  inside  $\mu_{k_O}$ , which is the observer group of the

same MIL. Let the sub-matrix  $\bar{Z}_{m \times n}^{\text{sub}}$  of the full MoM matrix  $\bar{Z}$  represent the interactions between the basis and the testing functions through the designated Green's function  $g(\mathbf{r}, \mathbf{r}')$ . The column of  $\bar{Z}^{\text{sub}}$  pertaining to the interaction of  $f_i$  with all testing functions is closely related to other columns.

Using the Modified Gram-Schmidt (MGS) process [6] and a user-specified tolerance  $\epsilon$ ,  $\bar{\mathbf{Z}}^{sub}$  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:

$$\frac{\|\bar{\mathbf{Z}}^{sub} - \bar{\mathbf{Q}}\bar{\mathbf{R}}\|}{\|\bar{\mathbf{Z}}^{sub}\|} < \epsilon \quad (3.8)$$

$$\text{where,} \quad \bar{\mathbf{Q}}^H \bar{\mathbf{Q}} = \bar{\mathbf{I}} \quad (3.9)$$

and the matrix norm  $\|\bar{\mathbf{X}}\|$  is defined as the maximum singular value of the matrix  $\bar{\mathbf{X}}$ . Similarly the interactions corresponding to testing functions belonging to  $S_D$  can be compressed for the same MIL.

#### 4. ADVANTAGES OF PILOT

The main idea behind PILOT is to maintain the superior compression efficiency inherent to QR-based methods compared to FMM based methods, while expediting the process of matrix compression. In other words the setup time is reduced while preserving the solve time and memory efficiency.

The absolute setup time for RMBT-QR (our prototype implementation of IES<sup>3</sup>) 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 figure 4.

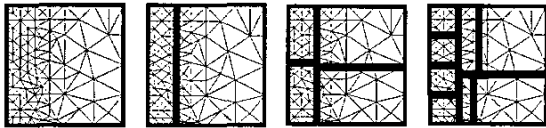


Figure 4: Multilevel tight bound  $k$ -d tree used in RMBT-QR (IES<sup>3</sup>)

It is infeasible 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 QR factorizations which are then discarded. Thus the setup time is considerably influenced by the variability of the tree structure and the resulting backtracking and refinement. In PILOT, the setup time is significantly reduced 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 exhaustive and accurate *a priori* epsilon-rank estimation is possible. PILOT therefore

incorporates the best features of the regular tree structure of multilevel FMM and the kernel-independent low-epsilon-rank compression of IES<sup>3</sup>.

#### 5. RESULTS

In this section we present simulation results to demonstrate the accuracy and time and memory efficiency of the PILOT-QR algorithm as compared to other existing solvers. All experiments are performed on a 1.6GHz processor with 4GB available RAM space. For PILOT and RMBT-QR analytic integrations are used for the near field terms in conjunction with the collocation scheme. For off-diagonal blocks, 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 2 is employed. A relative residual of  $1e-3$  is used for the GMRES iterative solution. The number of iterations required is observed to be similar for all the methods.

In the first example we consider a 3 level interconnect structure in a 10mm x 4mm x 1mm space. Each interconnect is 0.1mm x 0.1mm in cross-section. The separations between the levels are 0.4mm and the minimum separation between traces on the same level is 0.15mm.

The problem is solved with 10 interconnects as active nets and the rest as floating conductors. The  $\text{norm}_2$  errors observed for all the algorithms in the capacitance matrix so extracted are around  $1e-3$ .

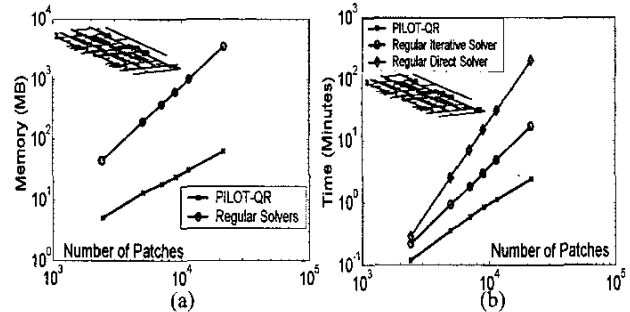


Figure 5: Performance comparison of PILOT with regular direct (dense-LU) and iterative solvers

In figures 5a and 5b the performance of PILOT is compared with that of regular direct (LU) and iterative solvers for the 3 level VLSI interconnect structure as discussed above. It can be observed that both the time and memory requirement scales linearly for PILOT as compared to the quadratic memory and time scaling for the regular iterative solver and the quadratic memory and cubic time scaling for the regular direct solver. It must be mentioned that for very small number of unknowns the regular approaches (direct/iterative) could be more profitable owing to the computational overhead of the fast solvers.

In figures 6a and 6b, the linear scaling coefficients of PILOT are compared with those of other existing fast solvers, FastCap and RMBT-QR. It is observed that PILOT maintains the compression efficiency of RMBT-QR while improving on the time requirements owing to a faster set-up time, thereby achieving an all-round superior performance over the existing fast solvers.

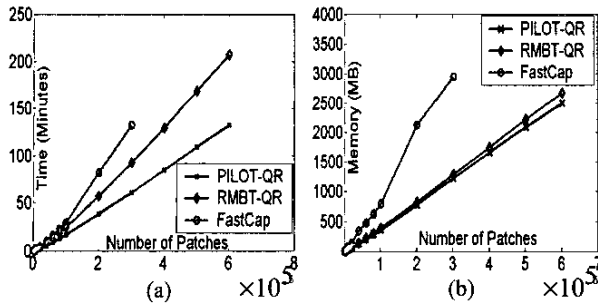


Figure 6: Performance comparison of PILOT with existing fast solvers, FastCap (FMM based) and IES<sup>3</sup> (QR-based)

In the next example we consider a package structure with 56 conducting leads as shown in figure 7:

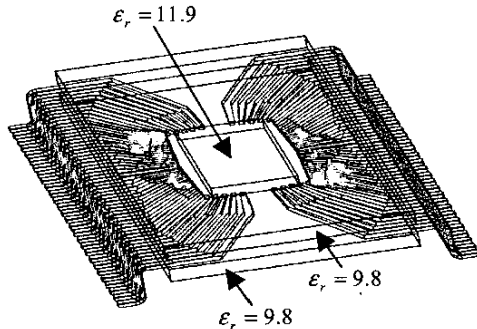


Figure 7: Package structure 280 $\mu\text{m}\times 680\mu\text{m}\times 56\mu\text{m}$  with 56 leads. The package is sandwiched by a top and a bottom Al<sub>2</sub>O<sub>3</sub> ceramic layer ( $\epsilon_r=9.8$ ) of thickness 20 $\mu\text{m}$ . A dielectric slab of  $\epsilon_r=11.9$  and thickness of 10 $\mu\text{m}$  is placed in the space between the leads (96 $\mu\text{m}\times 220\mu\text{m}$ ).

The first row of the capacitance matrix obtained for the 2 cases, with and without dielectrics, are plotted in figure 8 using PILOT-QR and FastCap. It can be observed that the results demonstrate excellent match between the values obtained from the 2 different algorithms. It can also be noted that due to the presence of the dielectrics the coupling-capacitances between the leads increase.

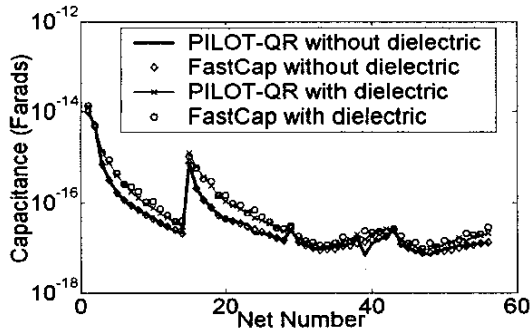


Figure 8: Comparison of results of PILOT with FastCap with or without dielectrics compared with/without dielectrics for the package structure in figure 8.

The required memory is compared with FastCap in figure 9a. In figure 9b, the advantage of PILOT is demonstrated over RMBT-QR for multiple nets for the package structure shown in Fig. 7, without the dielectrics. Time taken by RMBT-QR is always more compared to the time taken by PILOT, by an offset amount, though they demonstrate to have the same slope. This observation supports an improved set-up time performance of PILOT compared to RMBT-QR which is emphasized for a multiple net problem.

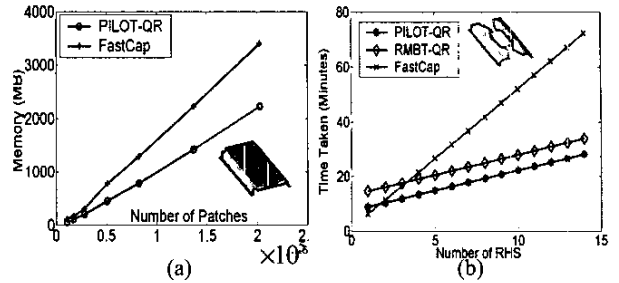


Figure 9: a) Memory comparison of PILOT with FastCap in the presence of dielectrics b) The total time required by the different algorithms for multiple right hand side solutions for the package structure in figure 8.

In the last example we demonstrate the applicability of PILOT in the electrostatic force analysis for MEMS array structures where large number of unknowns necessitates fast solution. The structure under consideration is a prototype of an array of micro-mirrors, which are used as electro-statically controlled optical switches. One array element is shown in figure 10.

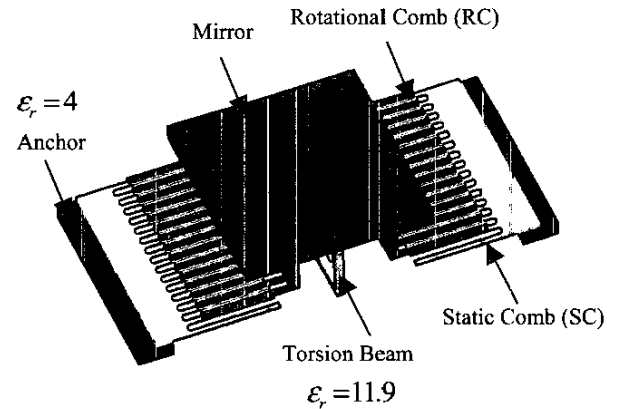


Figure 10: Micro-mirror array element. Two static combs are anchored to the base by 3 $\mu\text{m}$  thick SiO<sub>2</sub> slabs. Each static comb has 16 teeth each with dimensions of 20 $\mu\text{m}\times 1\mu\text{m}$  teeth, separated by 3 $\mu\text{m}$ . Each mirror is 45 $\mu\text{m}\times 32\mu\text{m}\times 2\mu\text{m}$  and can be tilted about the axis of the supporting torsion beam. Two comb structures are attached one on each side of the mirror, each containing 15 teeth each with dimensions of 20 $\mu\text{m}\times 1\mu\text{m}$  teeth.

Application of voltage on any of the static combs (SC) can rotate the micro-mirror about its supporting torsion beam. The PILOT-QR algorithm is applied to compute the electrostatic fields

and forces on the structures at various mirror angles. Since in an array of micro-mirrors the combs are placed close to each other there is electrostatic cross-coupling between the elements. This necessitates the simulation of the entire array for accurate prediction of the pull-down voltage of every mirror.

In this example the pressure distribution on a  $10 \times 10$  array of micro-mirrors due to electrostatic actuation is simulated. The results are demonstrated in figure 11 on a part of the array.

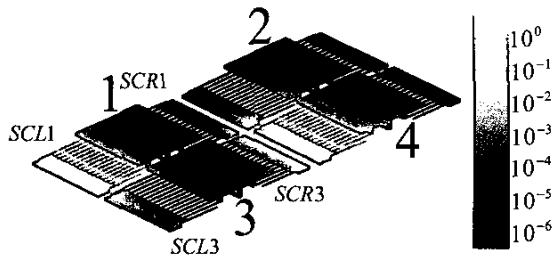


Figure 11: Electrostatic pressure ( $N/m^2$ ) distribution obtained on a 4 element part of a  $10 \times 10$  array of MEMS micro-mirror structure, by placing 1Volt on SCL1 and SCL4

In figure 12, the cross-coupling effects between the elements of the micro-mirror array structure are demonstrated. The static right comb of element 1 (SCR1) is excited by applying 1Volt. The electrostatic pressures on points at the edge of the rotational comb to the left of element 1 (RCL1) are plotted. Maintaining the excitation on the SCR1, additional excitations are applied in succession on static combs intended to rotate other elements. However, due to cross coupling, the electrostatic pressure on the

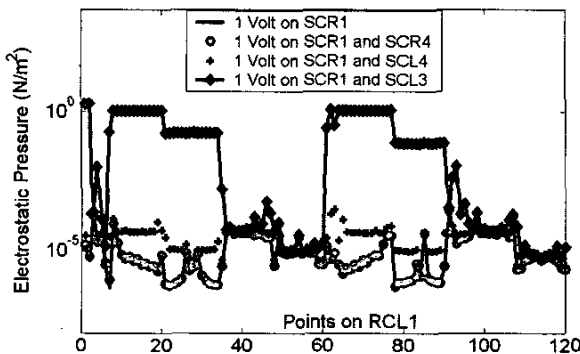


Figure 12: The cross-coupling effects in a micro-mirror array are demonstrated by plotting the electrostatic pressure distribution at points on rotational-comb-left due to excitations at various static-combs.

RCL1 change and the effects are demonstrated in figure 9. It is observed that as we move the second excitation closer to the RCL1 (observation comb), the coupling effect becomes pronounced. The simulation results confirm that the excitation of neighboring elements could change the electrostatic torque on a mirror, which in turn affects its rotational angle. The number of patches used for the simulation of the entire array is 0.342

million. The PILOT solved the problem with 3.5GB memory, 54 minutes of one-time setup and 16 minutes of solve-time per right hand side. FastCap could not fit the problem in the given RAM resources.

## 6. CONCLUSIONS

In this paper an improved multilevel scheme for QR-based matrix compression and subsequent fast iterative solution is presented. The method is based on exploiting the regular geometry structure of oct-tree decomposition along with merged-interaction optimizations to achieve a pre-determined block-matrix structure and consequently a faster setup time for QR-based compression. The memory and solve-time efficiency common to QR based approaches is preserved and hence overall superior performance is achieved.

The method is applied to parasitic capacitance extraction for structures on boards, packages and chips. The superior time and memory performance of the algorithm is demonstrated as compared to existing fast solvers. The method is also applied to predict the electrostatic pressure distribution on a MEMS array structure.

The algorithm is kernel independent and hence can be applied to solve multi-layered dielectric problems. It is also amenable to fast solution of full-wave kernels for electrically small structures.

## 7. REFERENCES

- [1] M. Kamon, S. McCormick and K. Shepard, "Interconnect parasitic extraction in the digital IC design methodology" *IEEE/ACM International Conference on Computer-Aided Design*, pp. 223 -230 Nov. 1999.
- [2] A. Brambilla, P. Maffezzoni, L. Bortesi and L. Vendrame, "Measurements and extractions of parasitic capacitances in ulsi layouts", *IEEE Transactions on Electron Devices*, vol. 50, Issue: 11, Nov. 2003, pp. 2236 -2247.
- [3] K. Nabors and J. White, "FastCap: a multipole accelerated 3-D capacitance extraction program", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10 issue 11 pp.1447-1459, Nov. 1991.
- [4] S. Kapur and D. Long, "IES<sup>3</sup>: A fast integral equation solver for efficient 3-dimensional extraction", *IEEE/ACM International Conference on Computer-Aided Design*, 1997, pp. 448 -455, Nov. 1997.
- [5] J.R. Phillips and J. White, "A precorrected-FFT method for electrostatic analysis of complicated 3-D structures", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16 issue 10 pp.1059-1072, Oct. 1997.
- [6] S. Rao, T.K. Sarkar and R.F. Harrington, "The Electrostatic Field of Conducting Bodies in Multiple Dielectric Media", *IEEE Transaction on Microwave Theory and Techniques*, vol. 32, issue 11, Nov 1984 pp. 1441-1448.
- [7] G.H. Golub and C.F. Van Loan, *Matrix computations* 2<sup>nd</sup> Ed. The Johns Hopkins University Press, Baltimore, 1989.