DiMES: Multilevel Fast Direct Solver based on Multipole Expansions for Parasitic Extraction of Massively Coupled 3D Microelectronic Structures

Dipanjan Gope
dips@u.washington.edu

Indranil Chowdhury
burunc@u.washington.edu

Vikram Jandhyala
jandhyala@ee.washington.edu

Department of Electrical Engineering,
Box 352500, University of Washington,
Seattle, WA-98195.
Telephone: 206-543-2186.

ABSTRACT

Boundary element methods are being successfully used for modeling parasitic effects in cutting-edge circuit design. The dense system matrix generated therein presents a time and memory bottleneck. Fast iterative solver techniques, developed to address the problem, suffer from convergence issues which become pronounced for large number of right hand sides as is the case for massively coupled systems. In this paper an iteration free solution scheme is presented. The dense matrix is rendered sparse by applying multilevel multipole expansions, and the resultant sparse matrix is solved by a traditional sparse matrix solver. The accuracy and time and memory requirements for the solver are compared against the regular methods. The advantage of the presented method over the corresponding iterative scheme is also demonstrated.

Categories and Subject Descriptors

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

General Terms: Algorithms, Performance, Design.

Keywords: Parasitics, Multilevel, Multipole, Non-Iterative

1. INTRODUCTION

As the feature size moves steadily down in the nanometer regime parasitic effects play an increasingly dominant role in determining the delay. Therefore proper modeling of these effects is critical for meeting timing requirements. To achieve a guaranteed high degree of accuracy in this process, a numerical 3D electromagnetic solver, such as the method of moments (MoM) integral equation solver, is necessitated. However, the direct application of MoM leads to the generation of a dense matrix with large number of unknowns, the solution of which presents a time and memory bottleneck. Several fast iterative solution based methods have been successfully developed to alleviate the problem, and this field can now be considered a mature area. All these techniques e.g. [1] rely on compression schemes to accelerate the matrix-vector products in a Krylov subspace iterative solution. The overall solution cost for these methods is proportional to the number of right hand sides (RHSs) and the number of iterations per RHS. For many ill-conditioned systems the number of iterations, required for acceptable convergence errors, may be extremely high. The effect is more pronounced in the total solution cost for problems with a large number of RHS vectors.

A regular LU based solution scheme on the other hand is not limited by the speed of convergence. However due to the dense nature of the $N \times N$ MoM matrix, the LU solution scheme is limited by a slow $O(N^3)$ setup and $O(N^2)$ memory and solve time requirements. The multilevel schemes developed for fast matrix-vector products do not inherently lend themselves to generating fast methods for direct LU decomposition or inversion. To accomplish fast direct solution, a compressed LU representation based on QR decomposition of MoM sub-matrices has been proposed [2-3]. However, for arbitrarily shaped structures which do not yield regular banded matrices, the cost of fill-ins even after methodical control [3] leads to inferior algorithm performance in time and memory requirements. On the other hand, neglecting fill-ins, as in incomplete factorization methods [4], may lead to inaccurate solutions if used as a direct method rather than as a good preconditioner.

In this paper, a fast direct solution scheme based on multipole expansions is presented. The FMM based approach is formulated, in a manner different from traditional usage, as a single sparse matrix which is then solved using a traditional sparse LU solver [5]. The unknowns include the charge density of the panels, and the multipole and local expansion coefficients of each cube. Though, this gives rise to a matrix of a larger dimension, the total number of elements in the entire matrix is the same as in an iterative FMM scheme. The resultant matrix has a sparse form, and traditional sparse matrix techniques can be leveraged in its LU factorization. In the results section, the accuracy, and time and memory efficiency of the new approach is compared against the regular LU scheme. The advantage of such a fast direct solution technique over its iterative counterpart is also presented.

2. INTEGRAL EQUATION

The proposed approach is inherently not limited to electrostatic problems. However, for illustration and ease of implementation, the approach is applied to large-scale capacitance simulations.

DAC 2005, June 13–17, 2005, Anaheim, California, USA

Capacitance problems formulated using MoM are solved by the electrostatic equation:

$$\nabla^2\phi(\mathbf{r}) = -\rho(\mathbf{r})/\varepsilon \qquad (2.1)$$

where $\phi_.$ denotes the potential, ρ is the charge-density and ε is the permittivity of the background material. The discretization of the integral equation obtained from (2.1) results in a matrix system of the form:

$$\overline{\mathbf{Z}}\mathbf{X} = \mathbf{V} \qquad (2.2)$$

where N is the number of unknowns, the $N \times N$ MoM matrix $\overline{\mathbf{Z}}$ is a dense Green's function matrix, $\mathbf{X}$ represent the unknown coefficients of known basis functions for charge density, and $\mathbf{V}$ represents the known potential excitation. Each element of the MoM matrix denotes the interaction between a testing and a basis function and is written as follows:

$$\overline{\mathbf{Z}}(\mathrm{i,j}) = \int_{S_i} ds\, t_i(\mathbf{r}) \int_{S_j} ds' g(\mathbf{r},\mathbf{r}') f_j(\mathbf{r}') \qquad (2.3)$$

where t_i is the testing function defined over S_i, f_j is the basis function defined over S_j and $g(\mathbf{r},\mathbf{r}')$ is the relevant Green's function. In the electrostatic case for P disconnected conductors, each column of the required $P \times P$ capacitance matrix is obtained by enforcing a voltage of 1V on the excited conductor, 0V on all other conductors and solving the system (2.2). The $N \times N$ system of equations is therefore solved P times to obtain the entire capacitance matrix.

3. MULTILEVEL DIRECT SOLVER

The direct multipole expansion solver (DiMES) algorithm has 4 main constituents:

A) Oct-tree spatial decomposition 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. Each cube c_j^{l+1} resulting from the the decomposition of c_i^l is called a child of c_i^l and the latter is denoted as the parent of c_j^{l+1}:

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

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

$$S_{c_j^{l+1}} = \{c_k^{l+1} \text{"} k \mid P_{c_k^{l+1}} = P_{c_j^{l+1}}\} \qquad (3.2)$$

B) Basic multilevel interaction list:

Every cube $c_i^l\ \forall i,l \mid 0 \le l \le l_c ; 0 \le i < n_c^l$, 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:

$$K_{c_i^l} = \{c_j^l \mid c_j^l \text{ is in the same level as } c_i^l \text{ and has atleast one contact point with } c_i^l\} \qquad (3.3)$$

Consequently the interaction list is defined as:

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

C) Single sparse matrix representation:

In this sub-section, the representation of the MoM matrix in a sparse form using multilevel multipole expansions is presented. The solution vector $\mathbf{X}$ of (2.2) is now extended to include the multipole and local expansion coefficients of each cube $c_i^l\ \forall i,l \mid 2 \le l \le l_c ; 0 \le i < n_c^l$. The new system matrix therefore is of the form:

$$\overline{\mathbf{Z}}_1\mathbf{X}_1 = \mathbf{V}_1 \qquad (3.5)$$

where: $\mathbf{X}_1 = [\mathbf{X};\mathbf{M}_4;\mathbf{M}_3;\mathbf{M}_2;\mathbf{L}_4;\mathbf{L}_3;\mathbf{L}_2]$ (3.6)

The formulation is explained with a 4 level example in Fig. 1, where, $\mathbf{M}_l$ and $\mathbf{L}_l$ are the multipole and local expansions of all cubes at level l.

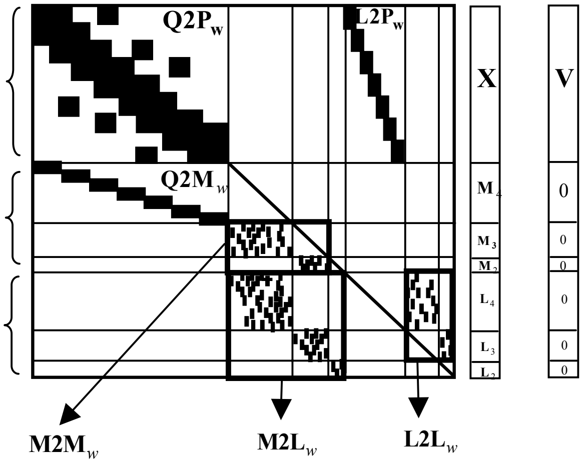

Figure 1: The sparse representation of the MoM matrix using multilevel multipole expansions

The first set of equations represents the computation of potentials in each cube at the lowest level from the charge density of its nearest neighbors and its own local expansion.

$$\mathbf{V} = [\mathbf{Q2P}_w \times \mathbf{X}] + [\mathbf{L2P}_w \times \mathbf{L}_4] \qquad (3.7)$$

The charge to potential $\mathbf{Q2P_w}$ matrix therefore consists of sub-matrices:

$$\mathbf{Q2P}_w = \bigcup \left[\mathbf{Q2P}_{c_i^{l_c}}^{c_j^{l_c}} \right] \qquad (3.8)$$

where each sub-matrix $\mathbf{Q2P}_{n_{c_i^{l_c}} \times n_{c_j^{l_c}}}$ represents the potential produced at the testing functions of an observer cube from the charge density basis functions of a source cube located in its nearest-neighbor list. Sparsity is ensured in the whole $\mathbf{Q2P_w}$ matrix because the interaction of a given lowest level cube is limited to its nearest neighbor list only. However, since an exact reordering procedure does not exist for arbitrary 3D structures, blocks of dense parts are often encountered at a distance from the diagonal. The $\mathbf{L2P_w}$ matrix converts the local expansions about the centers of the cubes $c_i^{l_c}$ at the lowest level to the potential at the testing functions. The $\mathbf{L2P_w}$ matrix consists of sub-matrices:

$$\mathbf{L2P}_w = \bigcup \left[\mathbf{L2P}_{c_i^{l_c}} \right] \qquad (3.9)$$

where the sub-matrices $\mathbf{L2P}_{n_{c_i^{l_c}} \times (p+1)^2}$ represent the local expansion to potential operation [1] for each cube.

The second set of equations represents the formation of multipole expansions about the cube centers from charge basis functions at the lowest level $l = l_c$ through the $\mathbf{Q2M_w}$ matrix and from multipole expansions of children through the $\mathbf{M2M_w}$ matrix at all other levels $l = 2,3,.....l_c - 1$. It should be noted that the $\mathbf{Q2M_w}$ matrix only occurs at the lowest level and is itself a block diagonal matrix since each cube at the lowest level only contributes to multipole expansions of the same cube.

$$\mathbf{Q2M}_w \times \mathbf{X} + \mathbf{M2M}_w \times \mathbf{M} - \mathbf{I} \times \mathbf{M} = 0 \qquad (3.10)$$

where **I** is an identity matrix, $\mathbf{Q2M_w}$ consists of sub-matrices:

$$\mathbf{Q2M}_w = \bigcup \left[\mathbf{Q2M}_{c_i^{l_c}} \right] \qquad (3.11)$$

for all cube $c_i^{l_c}\ \forall i \,|\, 0 \le i < n_c^{l_c}$. Each $\mathbf{Q2M}_{(p+1)^2 \times n_{c_i^{l_c}}}$ sub-matrix pertaining to a cube c_i^l, converts the charge basis functions f_i belonging to c_i^l to multipole expansions about the cube center [1]. Similarly the $\mathbf{M2M_w}$ matrix consists of sub-matrices:

$$\mathbf{M2M}_w = \bigcup \left[\mathbf{M2M}^{c_i^l}_{c^{l-1}_{P_{c_i^l}}} \right] \qquad (3.12)$$

where individual $\mathbf{M2M}_{(p+1)^2 \times (p+1)^2}$ sub-block shifts the multipole expansion of cube $c_i^l\ \forall i,l \,|\; 3 \le l \le l_c\,; 0 \le i < n_c^l$ to the multipole expansion of its parent $P_{c_i^l}^{l-1}$ [1].

The third and final set of equations represents the formation of local expansions for all cubes $c_i^l\ \forall i,l \,|\; 2 \le l \le l_c\,; 0 \le i < n_c^l$ from multipole expansions of cubes in their corresponding interaction lists through the M2L operation and from local expansions of their corresponding parents through the L2L operation:

$$\mathbf{M2L}_w \times \mathbf{M} + \mathbf{L2L}_w \times \mathbf{L} - \mathbf{I} \times \mathbf{L} = 0 \qquad (3.13)$$

The $\mathbf{M2L}_w$ matrix consists of sub-matrices:

$$\mathbf{M2L}_w = \bigcup \left[\mathbf{M2L}^{c_j^l}_{c_i^l} \right] \qquad (3.14)$$

for all $c_i^l\ \forall i,l \,|\; 2 \le l \le l_c\,; 0 \le i < n_c^l$ such that $c_j^l \forall j \,|\, c_j^l \in I_{c_i^l}$. Each $\mathbf{M2L}_{(p+1)^2 \times (p+1)^2}$ sub-matrix represents the local expansion produced in an observer cube due to the multipole expansion of a source cube belonging to its interaction list [1]. Similarly the $\mathbf{L2L}_w$ matrix consists of sub-matrices:

$$\mathbf{L2L}_w = \bigcup \left[\mathbf{L2L}^{P^{l-1}_{c_i^l}}_{c_i^l} \right] \qquad (3.15)$$

where each $\mathbf{L2L}_{(p+1)^2 \times (p+1)^2}$ represents the formation of local expansion in a cube $c_i^l\ \forall i,l \,|\; 3 \le l \le l_c\,; 0 \le i < n_c^l$ due to the local expansion of its parent $P_{c_i^l}^{l-1}$ [1].

D) Sparse LU including all fill-ins:

The matrix $\mathbf{Z}_1$ of (3.5) demonstrates significant sparsity as seen from Fig. 1. The total number of non-zero entries in $\mathbf{Z}_1$ before LU formation is the same as the total number of entries stored for FMM based iterative solutions like Fast-Cap. The key issues pertaining to a fast direct solution of (4.5) are:

i) Number of entries in the LU of $\mathbf{Z}_1$ which determines the memory requirement of the process and also the solve time for each RHS vector. This is affected by fill-ins and can be controlled by proper reordering of the system.

ii) The initial LU setup time, which is controlled by the size of the system as well as the percentage of sparsity in the matrix.

For a given problem, the number of levels controls the sparsity and the size of the system matrix and is optimized to reduce the LU setup time. The reordering and LU formation of the resultant sparse matrix is done by using the Sparse 1.3 package [5].

4. DiMES AGAINST ITERATIVE METHOD

Both DiMES and Fast-Cap are based on multipole expansions and share the same multilevel algorithm. The difference lies in the solution procedure: Fast-Cap employs a Krylov subspace iterative solution whereas DiMES formulates the problem using a single sparse matrix and employs traditional sparse solver techniques [5] to obtain a direct solution.

In the fast iterative solver environment, the matrix vector product operation is expedited to $O(N)$, where N is the number of unknowns. Consequently the total solve-cost for *r* RHS vectors is $O(N) \times p \times r$ where p is the number of iterations required for convergence. The key concern for an iterative solution method is the speed of convergence and for an ill-conditioned problem the value of *p* can be very large. The effect of slow convergence is pronounced when solving for a large number of RHS vectors. A regular MoM dense matrix direct solver, on the other hand, is not limited by the speed of convergence but exhibits inefficient $O(N^2)$ scaling in solve-time and memory and $O(N^3)$ scaling in the one-time LU setup. To alleviate the problem, DiMES employs multilevel multipole expansions to formulate a sparse matrix and therefore reduces the scaling orders for memory, solve and initial setup time. Based on numerical experimentation it is found that DiMES scales as $O(N^{1.2-1.4)}$ in memory and solve time and $O(N^{1.8-2})$ in LU setup time. Hence, for moderately sized problems with large number of RHS vectors DiMES is a more efficient solution scheme, as shown in example 3 of section VI.

However, it should also be noted that due to the high initial LU setup cost, for problems where the ratio of the number of unknowns to the number of right hand sides is very high, iterative techniques like Fast-Cap are preferable.

4. RESULTS

In this section we present simulation results to demonstrate the accuracy and time and memory efficiency of the DiMES algorithm. All experiments are performed on a 1.6GHz processor with 1.5GB available RAM space. For both DiMES and Fast-Cap analytic integrations are used for the Q2P operations in conjunction with the collocation scheme and a multipole order of 2 is employed. A relative residual of 1e-3 is used for Fast-Cap adaptive iterative solution.

Example 1: A test-chip structure consisting of meander lines, coplanar waveguides, pads etc. as shown in Fig. 2 is considered.

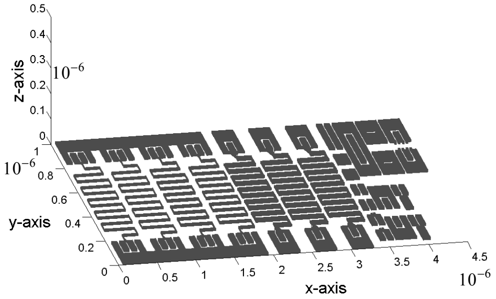

Figure 2: The test chip under consideration consists of meander lines. coplanar wave-guides.

The problem is solved using the regular LU method and DiMES at multiple discretization levels. The memory required for the process using both solvers are plotted against the number of unknowns in Fig. 3a.

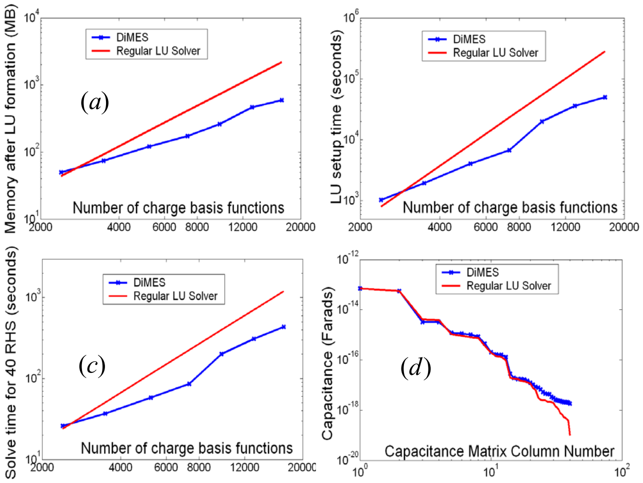

Figure 3: (a) Memory scaling (b) LU set-up scaling (c) LU solve-time scaling (d) First row of capacitance matrix.

It should be noted here that the matrix for the regular LU solver consists of only real numbers and the dimension of the matrix is the number of charge basis functions, whereas for the fast direct solver, the matrix is complex and also of a larger dimension. However, due to its inherent sparsity DiMES exhibits better memory scaling with increasing discretization levels. The LU setup time and solve time for 40 RHS vectors are also plotted for DiMES and a regular LU solver in Fig. 3b and 3c respectively. It can be observed that DiMES exhibits superior scaling with increasing number of unknowns. Finally the first row of the 40×40 capacitance matrix is plotted in Fig. 3d and it can be seen that the results match very closely.

Example 2: In the next example, the relative advantages of a fast direct solution technique over its iterative counterpart for massively coupled problems is demonstrated. The structure under consideration consists of 2500 metal contacts distributed in an area of $1mm \times 1mm$ as shown in Fig. 4. Such structures are frequently encountered for problems like substrate coupling where the capacitance matrix of the entire system is required to be computed to estimate the coupling between the contacts. The entire problem is discretized using 6500 triangular tessellations and the problem is solved using both Fast-Cap and DiMES.

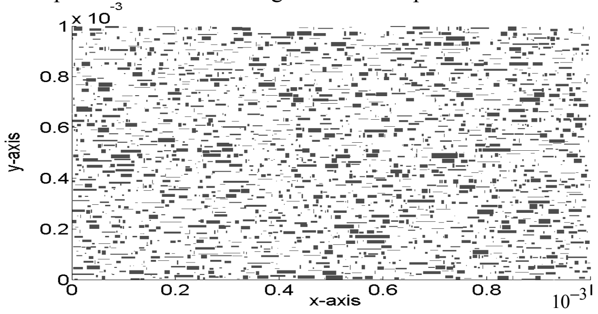

Figure 4: Massively coupled problem consisting of 2500 metal contacts distributed in an area of $1mm \times 1mm$. The entire capacitance matrix of the system is required to be computed.

It can be noticed from Fig. 5a that approximately after 400 RHS vectors the direct solution technique becomes a better alternative as compared to the iterative solution method. The first 100 elements of the first row of the capacitance matrix are plotted for both algorithms in Fig. 5b. In fact the Frobenius norm error between the capacitance matrices obtained by the two algorithms is 0.1%.

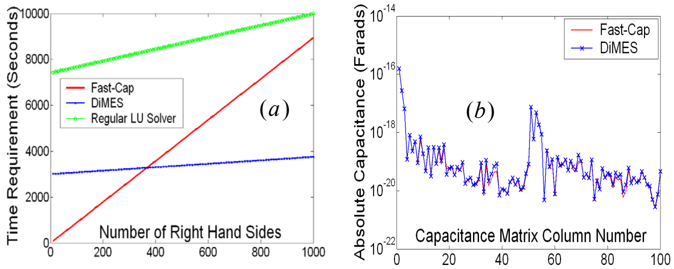

Figure 5: (a) The total setup + solve time required for DiMES and Fast-Cap is plotted against the number of RHS vectors.
(b) The first 10 elements of the first row of the capacitance matrix are plotted for both algorithms and the match is excellent.

5. CONCLUSIONS

In this paper a fast direct solver based on multilevel multipole expansions is presented. The novelty of the proposed scheme is in the representation of the traditional multilevel FMM structure in terms of a slightly larger single sparse matrix which is amenable to fast solution based on existing sparse LU techniques.

REFERENCES

[1] 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.

[2] Francis X. Canning and Kevin Rogovin, “Fast direct solution of standard moment-method matrices” *IEEE Antennas and Propagation Magazine* Vol. 40 No. 3 pp. 15 – 26, June 1998.

[3] D. Gope and V. Jandhyala, “An iteration-free fast multilevel solver for dense method of moment systems”, *IEEE Proc. on Electrical Performance of Electronic Packaging*, pp: 177-180, Oct. 2001.

[4] Shu Yan, Vivek Sarin and Weiping Shi, “Fast capacitance extraction using inexact factorization”, *IEEE Proc. on Electrical Performance of Electronic Packaging*, pp: 285-288, Oct. 2004.

[5] Kenneth Kundert, Sparse Matrix Techniques, in Circuit Analysis, Simulation and Design, Albert Ruehli, (Ed.), North-Holland, 1986.