A Linear Time Implicit Congruence Sparsification Technique for BEM Capacitance Extraction

Rong Jiang

Electrical and Computer Engineering
College of Engineering
University of Wisconsin
Madison, WI 53706, USA
jiang@cae.wisc.edu

Charlie Chung-Ping Chen

Electrical and Computer Engineering
College of Engineering
University of Wisconsin
Madison, WI 53706, USA
chen@engr.wisc.edu

Abstract— **This paper presents an efficient methodology for generating sparsified potential coefficient matrices for three-dimensional capacitance extraction. Previous capacitance extraction algorithms based on boundary element method (BEM) formulate the potential coefficient matrix in terms of surface potentials and charges on those most delicate panels (leaf panels). By introducing the concept of *basis charges*, we discover that leaf panel charges compose the worst basis which leads to the densest linear system. Therefore, we propose a linear time basis panel selection algorithm to choose a new basis. It is provable that the $n \times n$ potential coefficient matrix constructed in terms of the new basis contains $O(n)$ non-zero entries and hence the sparse system can be solved much more efficiently by preconditioned Krylov subspace iterative methods. Experimental results demonstrate the superior runtime and memory consumption over previous approaches while achieving similar accuracy.**

I. INTRODUCTION

Boundary element method (BEM) have been adopted as the main approach for solving self and coupling capacitances for interconnects in VLSI design, packaging, and MEMS [1]–[7]. However, BEM yields an extremely dense linear system which not only limits the tractable problem size, but also prevents the usage of efficient matrix solvers.

In this paper, we reveal that the intrinsic reason why the linear system arising from BEM is dense is due to the selection of leaf panel charges as the basis. Therefore, we significantly improve BEM capacitance extraction by providing a linear time basis panel selection algorithm (BPSA) to choose a new basis. The related potential coefficient matrix of the new basis is very sparse and hence the sparse system can be solved by preconditioned Conjugate Gradient (PCG) or GMRES. Experimental results show that the new algorithm is faster and uses less memory than previous algorithms, including FastCap [3], HiCap [5], and PHiCap [7].

II. BACKGROUND

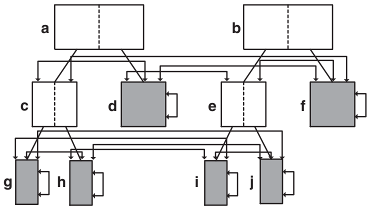
The capacitances between m conductors can be represented by the capacitance matrix $C \in \mathcal{R}^{m \times m}$,

$$C v_m \quad = \quad q_m, \qquad (1)$$

where $v_m \in \mathcal{R}^m$ and $q_m \in \mathcal{R}^m$ are conductor potential and surface charge vectors respectively. To determine the j^{th} column of the capacitance matrix, the surface charge on each conductor is computed by raising the potential on the j^{th} conductor to one while grounding other conductors and C_{ij} is equal to the surface charge on the i^{th} conductor. The procedure is repeated m times to compute all columns of C.

However, due to the uneven distribution of the surface charge, each conductor needs to be divided into smaller panels and surface charges on those more delicate panels are assumed to be uniform.

HiCap [5] and PHiCap [7] propose a hierarchical panel refinement scheme which can be represented by a multiple-tree structure show in Fig. 1. If the estimated potential coefficient between two panels is larger than a threshold value, they are further divided into smaller panels. Otherwise, a link recording the potential coefficient is created between these two panels.



$$J = \begin{array}{c|cccccc|} & d & f & g & h & i & j \\ \hline a & 1 & & 1 & 1 & & \\ b & & 1 & & & 1 & 1 \\ c & & & 1 & 1 & & \\ d & 1 & & & & & \\ e & & & & & 1 & 1 \\ f & & 1 & & & & \\ g & & & 1 & & & \\ h & & & & 1 & & \\ i & & & & & 1 & \\ j & & & & & & 1 \\ \hline \end{array}$$

Fig. 1. BEM capacitance algorithms HiCap and PHiCap.

The charge on a parent panel is the sum of charges on its two child panels. Therefore, an arbitrary panel charge is the sum of charges on all of its underlying leaf panels, i.e., all panel charges can be represented by linear combinations of leaf panel charges:

$$q_N \quad = \quad Jq, \qquad (2)$$

where $q_N \in \mathcal{R}^N$ is the vector of charges on all leaf and non-leaf panels and $q \in \mathcal{R}^n$ is the vector of charges on leaf panels. N denotes the total number of panels in the multiple tree structure and n is the number of leaf panels. $J \in \mathcal{R}^{N \times n}$ is called the structure matrix containing the coefficients of those linear combinations.

The links within the multiple tree structure can be represented by the link matrix $H \in \mathcal{R}^{N \times N}$. If a link exists between panel i and panel j, the corresponding entry in H is non-zero and can be calculated by

$$H_{ij} \quad = \quad \frac{1}{a_j} \int_{panel j} G(x', x_i) da', \qquad (3)$$

where a_j is the surface area of panel j and $G(x, x')$ is the Green's function which has different formulas for uniform dielectric and multiple dielectrics. It has been shown in [5], [7] that the number of links in the multiple tree is $O(n)$.

The potential produced by the link H_{ij} on panel i is equal to $H_{ij}q_j$, so that let $v_N \in \mathcal{R}^N$ be the vector of panel potentials induced by their attached links, then

$$v_N \quad = \quad Hq_N. \tag{4}$$

Since the total potential on a panel is the sum of potentials caused by links on their parent panels and themselves, so that the leaf panel potential vector $v \in \mathcal{R}^n$ is equal to

$$v \quad = \quad J^T v_N. \tag{5}$$

By using Eqs. 2, 4, and 5, the relation between leaf panel potential and surface charge can be formulated as

$$Pq \quad = \quad v, \tag{6}$$

where

$$P \quad = \quad J^T HJ. \tag{7}$$

$P \in \mathcal{R}^{n\times n}$ is the potential coefficient matrix.

Therefore, the capacitance extraction problem becomes solving the leaf panel charges in Eq. 6 when the leaf panel potential distribution v is known.

III. New Algorithm

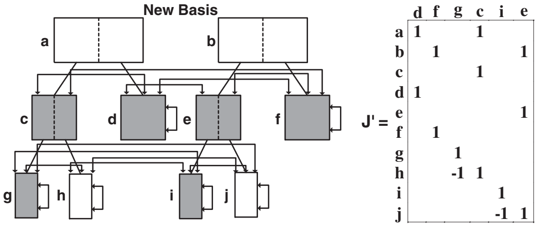
Let S denote the variable space composed of charges on all leaf and non-leaf panels

$$S \quad = \quad \{q_i|\text{surface charge on panel i}, 1 \leq i \leq N\}.$$

If all panel charges in S can be represented by unique linear combinations of charges on a set of n panels, charges on those panels are basis charges and those panels are basis panels.

As shown in Eq. 2, leaf panels compose one set of basis panels. However, since leaf panels interact with each other through links between themselves or their upper-level parent panels, every entry in P is non-zero and hence P containes n^2 fill-ins. Consequently, if we take leaf panel charges as the basis, the corresponding potential coefficient matrix P will be the densest one.

Fortunately, for a given tree structure, there are many possible bases. For example, for the multiple tree structure in Fig. 1, Fig. 2 shows another basis, which includes two non-leaf panels c and e. The related structure matrix J' is also shown in Fig. 2.



$$J' = \begin{array}{c|cccccc} & d & f & g & c & i & e \\ \hline a & 1 & & & 1 & & \\ b & & 1 & & & & 1 \\ c & & & & 1 & & \\ d & 1 & & & & & \\ e & & & & & & 1 \\ f & & 1 & & & & \\ g & & & 1 & & & \\ h & & & -1 & 1 & & \\ i & & & & & 1 & \\ j & & & & & -1 & 1 \end{array}$$

Fig. 2. A new basis and its corresponding structure matrix.

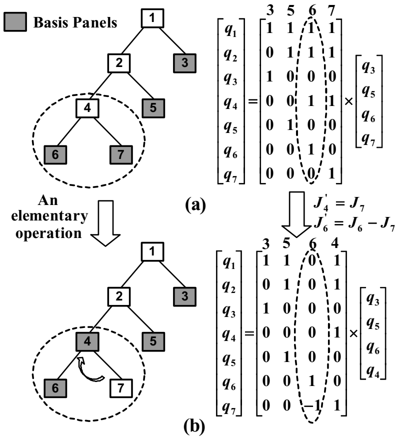
Since each basis has its distinct structure matrices J' such that $q_N = J'q'$, the related potential coefficient matrix $P' = J'^T HJ'$ has different densities. So, the basic idea of our algorithm is to choose a new basis so that the corresponding P' is sparse.

A. New Basis Panels

Our basis panel selection algorithm (BPSA) is based on an important elementary operation to generate a new basis.

For an elementary tree in which two children are included in the current basis, arbitrarily eliminating one of the child panels and adding the parent panel to the basis generates another basis.

Without loss of generality, we use an example to gain a clear idea of this important operation. Assume the current basis includes all leaf panels and the structure matrix J is shown in Fig. 3.(a). Now, we apply the elementary operation and move one basis panel from panel 7 to its parent panel 4. Apparently this results in another basis since all panels charges still can be represented by charges on those panels. The new structure matrix J' is shown on right hand side in Fig. 3.(b).



$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} 3 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{array} \times \begin{bmatrix} q_3 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix}$$

$J_4' = J_7$
$J_6' = J_6 - J_7$

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \\ q_7 \end{bmatrix} = \begin{array}{c} \begin{array}{cccc} 3 & 5 & 6 & 4 \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \end{array} \times \begin{bmatrix} q_3 \\ q_5 \\ q_6 \\ q_4 \end{bmatrix}$$

Fig. 3. An important operation to generate new basis panels.

The column corresponding to panel 4 in the new structure matrix J' is identical with the column corresponding to panel 7 in the original structure matrix J, since upper level panels originally gathering the charge on panel 7 still collects the charge on panel 4 after the elementary operation.

However, the column corresponding to panel 6 is changed in the new structure matrix, since the charge on panel 4 is the sum of charges on panel 6 and 7, upper level panels now only need to gather the charge on panel 4. So the changed column corresponding to panel 6 in J' is

$$J_6' \quad = \quad J_6 - J_7, \tag{8}$$

where J_i represents the column corresponding to panel i in J. Furthermore, Eq. 8 can be represented in a matrix form as

$$J' \quad = \quad JE. \tag{9}$$

E is an elementary transformation matrix expressed by

$$E = \begin{bmatrix} \ddots & & & \\ & 1 & 0 & \\ & -1 & 1 & \\ & & & \ddots \end{bmatrix} \begin{array}{l} \\ \text{panel 6} \\ \text{panel 7} \\ \\ \end{array} \tag{10}$$

Consequently, by using Eq. 10, the relation between the new potential coefficient matrix P' and P can be written as

$$P' = J'^T HJ' = (JE)^T H(JE) = E^T PE \tag{11}$$

So P' is obtained by a congruence transformation on P.

It is important to notice that this transformation only changes the column and row corresponding to panel 6 which are obtained by subtracting the column and row of panel 7 from the column and row of panel 6 in P. Since a link on an upper level panel above panel 6 and 7 introduces identical fill-ins in columns and rows of panel 6 and 7, the subtraction cancels out identical terms and creates many zeros in P'.

The elementary operation of moving basis panels upward can be executed continuously. As shown in Fig. 4, after moving panel 7 to panel 4, the elementary tree including panel 2, 4, and 5 now has two basis panels (panel 4 and panel 5). So we can eliminate panel 5 (or panel 4) and add it parent panel 2. Also notice that in this step, the column and row corresponding to panel 6 will not be affected and hence zeros created in the previous step are preserved. Similarly, after this step, we can move panel 3 (or panel 2) to panel 1 and again introduce many zero entries.

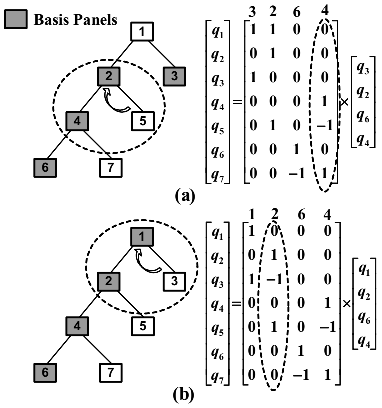


Fig. 4. Successively applying the elementary operation.

So successively moving basis panels upward is equivalent to implicitly apply consecutive congruence transformations on the potential coefficient matrix with the transformation matrix

$$E = E_1 E_2 E_3 \cdots \tag{12}$$

In each step of the transformation, many zeros are created in the new potential coefficient matrix and those already created zero entries will not be destroyed the later steps.

Assume we start from the basis including all leaf panels, and then we apply the elementary operation to consecutively push basis panels from bottom to top. At the end, the result basis will only include root panels and left-hand side (LHS) panels in all internal elementary trees. This process is equivalent to consecutively apply congruence transformations to cancel out duplicated terms introduced by the same link. So in the new potential matrix P', the number of non-zeros is comparable with the total number of links in the multiple-tree structure, which has been proven to be $O(n)$ [5]. This property has also been observed in the experiment as shown in Fig. 5.

Therefore, the new basis includes all root panels and all left-hand side panels will lead to a sparse potential coefficient matrix containing O(n) non-zero entries.

B. Direct Formulation of J'

The corresponding structure matrix J' of the new basis can be constructed using the following method.

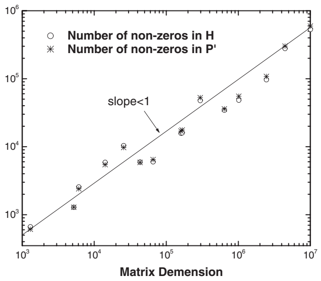


Fig. 5. Comparison of non-zero entries in H and P'.

In the column J'_i corresponding to a basis panel i, each entry J_{ij} is 1 if panel i contains the right-hand side panel j. If panel i is not a root panel, then each entry J_{ij} is -1 if the parent of panel i contains the right-hand side panel j.

The above method can be illustrated by an example in Fig. 6. Panel 2 is a LHS panel and has been included in the new basis. Panel 5 and 7 are its underlying RHS panels and hence the related entries in J' are filled by 1. The parent of panel 2 contains RHS panel 3, so that the related entry in J' is -1.

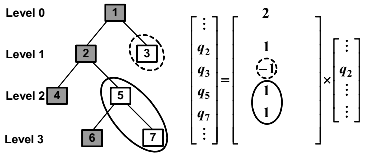


Fig. 6. Efficient construction of the new structure matrix J'.

C. Extracting E from J'

We have shown that the new potential coefficient matrix P' is obtained by applying congruence transformations on the original P matrix. Since $P'q' = v'$, by substituting $P' = E^T P E$, we get $E^T P E q' = v'$. Also we know that $Pq = v$. So these two equations can be satisfied by setting $v' = E^T v$ and $q = Eq'$.

From $q = Eq'$, we can see that E is the coefficient matrix when representing leaf panel charges using charges on those new basis panels. Since we have known that all panel charges can be represented by $q_N = J'q'$, so that E has already been included in the J' matrix and hence can be obtained directly.

D. Solving $P'q' = v'$

For the problem with uniform media, the sparse linear system $P'q' = v'$ is symmetric. In this scenario, we use incomplete Cholesky factorization with no fill to compute the preconditioner. Preconditioned Conjugate Gradients method is used to solve the system. For the problem with multiple dielectrics, the sparse linear system is unsymmetric. The preconditioner is computed from incomplete LU factorization of the coefficient matrix. No fill is allowed during factorization. We use right preconditioned GMRES method to solve the system. After we get q' which is the charges on optimal basis panels, the charges on leaf panels can be easily obtained by $q = Eq'$.

E. Complexity Analysis

The outline of our new algorithm is summarized in Table. I. The first step of selecting the new basis can be done by scanning all $N = 2n - 1$ panels to determine which are roots and LHS panels and hence takes O(n) time. Constructing J' is equivalent to insert $O(n)$ non-zeros in J' and hence is also $O(n)$. For the second step, E is contained in J' and does not require extra time. H has been proved to contain $O(n)$ non-zeros [5], so that the construction of $P' = J'^T H J'$ can also be done in $O(n)$.

TABLE I

OUTLINE OF OUR NEW ALGORITHM.

```
BEGIN
  1. Select one optimal set of panels and generate the new
     structure matrix $J'$.
  2. Extract the transformation matrix $E$ from $J'$. Given a leaf
     panel voltage vector $v$, $v' = E^T v$.
  2. Calculate the new potential matrix $P' = J'^T H J'$ and solve
     $P'q' = v'$ using PCG or GMRES.
  4. Obtain root panel charges directly in $q'$.
END
```

IV. EXPERIMENTAL RESULTS

We implemented the new algorithm in $C++$ language and Matlab. All experiments are executed on Sun-Blade 2500 with two 1.28-GHz UltraSPARC IIIi processors, 8G RAM and Sorlaris 9. The test examples are $k \times k$ bus crossing conductors for $k = 2$ to 16, generated by busgen in FastCap released package [3].

The density of the new potential coefficient matrix P' related to the new basis is plotted in Fig. 7. The density is defined as the total number of non-zeros in P' divided by its dimension. As shown in Fig. 7, as the number of leaf panels goes over one thousand, P' is very sparse and the density of P' becomes well below 10%.

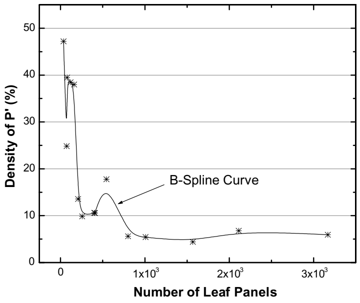


Fig. 7. Density of the new potential coefficient matrix P'.

Table II compares the performance of three algorithms : FastCap [3] with expansion order 2, HiCap [5], and the new algorithm. The convergence tolerance is set to 0.01, and error is calculated with respect to FastCap (-o2). Iteration is the average number of iterations per conductor. Our new algorithm is the fastest one in these three algorithms. Compared with FastCap, it is $30 - 40$ times faster and with much less memory. Compared with HiCap, for the bus 12×12 benchmark, the new algorithm exhibits nearly 10 times speedup.

HiCap represents P as a block matrix instead of implementing it directly, and hence the real storage of P is $O(n)$. All H, J, and P' of our new algorithm contain $O(n)$ non-zeros, so that the memory consumptions of the new algorithm and HiCap are in the same order. The actual accuracy and memory consumption depend on the refinement parameters. When the number of leaf panels is roughly the same, HiCap and the new algorithm have comparable accuracy.

We do not have access to PHiCap [7] and cannot compare with it explicitly. Published results show PHiCap is $2 - 3$ times faster than HiCap for benchmarks in Table 2. Based on the comparison with HiCap, we can expect our new algorithm is faster than PHiCap as well. Also for testing cases in Table 2, normally the new algorithm converges in less than 2 iterations while PHiCap needs about 3 iterations. Also, the main disadvantage of PHiCap is its memory consumption due to the explicit formulation of transformation matrix while the new algorithm directly formulates the sparse matrix P'. Also [7] shows that PHiCap has lower accuracy than HiCap. So the new algorithm can be superior to PHiCap in terms of memory and accuracy.

TABLE II

SIMULATION RESULTS COMPARISON.

4×4 Bus, Unit List: Time(Sec), Memory(MB)

Algorithm	Time	Iteration	Memory	Error	Panels
FastCap	8.03	18.63	26.27	–	2736
HiCap	0.77	8.7	0.99	0.72%	2176
New Alg	0.39	1.12	0.581	0.76%	2112

6×6 Bus, Unit List: Time(Sec), Memory(MB)

Algorithm	Time	Iteration	Memory	Error	Panels
FastCap	35.55	14.4	65.19	–	5832
HiCap	3.19	14.5	1.85	1.42%	3168
New Alg	0.7	1.08	1.54	1.50%	3168

8×8 Bus, Unit List: Time(Sec), Memory(MB)

Algorithm	Time	Iteration	Memory	Error	Panels
FastCap	67.4	12	114.5	–	10080
HiCap	14.64	13.4	5.03	1.63%	8448
New Alg	2.84	1.43	3.58	1.91%	8320

12×12 Bus, Unit List: Time(Sec), Memory(MB)

Algorithm	Time	Iteration	Memory	Error	Panels
FastCap	357.99	18.1	297.8	–	22032
HiCap	76.53	15.1	12.72	1.08%	12864
New Alg	7.21	1.41	11.87	1.18%	12480

V. CONCLUSION

This paper presents a 3D capacitance extraction algorithm. The new algorithm exhibits significant improvement of previous best algorithms by proposing a novel but simple method to obtain a sparse potential coefficient matrix. Thus preconditioners can be easily constructed and hence greatly speedup iterative matrix solvers. Detailed analysis shows that the new algorithm is faster and consumes less memory than all previous algorithms.

VI. ACKNOWLEDGMENT

This work was partially funded by SpringSoft, National Science Foundation under grants CCR-0093309 & CCR-0204468 and National Science Council of Taiwan, R.O.C. under grant NSC 92-2218-E-002-030.

REFERENCES

[1] S. Kapur and D. E. Long. Ies3: A fast integral equation solver for efficient 3-dimensional extraction. *Proc. ICCAD*, pages 448–455, 1997.

[2] B. Krauter, X. Yu, A. Dengi, and L. Pileggi. A sparse image method for bem capacitance extraction. *Proc. DAC*, pages 357–362, June 1996.

[3] K. Nabors and J. White. Fastcap: a multipole accelerated 3-d capacitance extraction program. *IEEE Trans. on CAD*, pages 1447–1459, 1991.

[4] J. R. Phillips and J. White. A precorrected fft method for capacitance extraction of complicated 3-d structures. *IEEE Trans. CAD*, pages 1059–1072, 1997.

[5] W. Shi, J. Liu, N. Kakani, and T. Yu. A fast hierarchical algorithm for 3-d capacitance extraction. *IEEE Trans. on CAD*, pages 330–336, 2002.

[6] J. Tausch and J. White. A multiscale method for fast capacitance extraction. *Proc. DAC*, pages 537–542, 1999.

[7] S. Yan, V. Sarin, and W. Shi. Sparse transformations and preconditioners for hierarchical 3-d capacitance extraction with multiple dielectrics. *Proc. DAC*, pages 788–793, 2004.