FASTHENRY: A Multipole-Accelerated 3-D Inductance Extraction Program

M. Kamon*, M. J. Tsuk†, and J. White ‡

Abstract

A mesh analysis equation formulation technique combined with a multipole-accelerated Generalized Minimal Residual (GMRES) matrix solution algorithm is used to compute the 3-D frequency dependent inductances and resistances in nearly order n time and memory where n is the number of volume-filaments. The mathematical formulation and numerical solution are discussed, including two types of preconditioners for the GMRES algorithm. Results from examples are given to demonstrate that the multipole acceleration can reduce required computation time and memory by more than an order of magnitude for realistic integrated circuit packaging problems.

1 Introduction

In high performance VLSI integrated circuits and integrated circuit packaging, there are many cases where accurate estimates of the coupling inductances of complicated three-dimensional structures are important for determining final circuit speeds or functionality. The most obvious examples are the pin-connect structures used in advanced packaging. For the past decade, volume-element techniques have been used to compute self and coupling inductances of complex three dimensional geometries, but the techniques were intended for geometries

*M. Kamon and J. White are with the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139

†M. J. Tsuk is with Digital Equipment Corporation, Maynard, MA 01754

‡This work was supported by the Defense Advanced Research Projects Agency contract N00014-91-J-1698, the National Science Foundation contract (MIP-8858764 A02), a National Science Foundation Graduate Fellowship, and grants from Digital Equipment Corporation and IBM.

which could be represented with at most a few hundred volume filaments. However, to be accurately analyzed, the complicated structures currently used in integrated circuit packaging can require up to ten thousand filaments. Existing programs become extraordinarily computationally expensive for such large problems, and new algorithms must be developed whose computational cost and memory use grows more slowly with problem size.

In this paper we describe FastHenry, a program for magnetoquasistatic analysis of complicated three-dimensional packages and interconnect. The program uses a standard discretization of an integral formulation of magnetoquasistatic analysis, but then reformulates the discretized equations using a mesh analysis approach. The mesh formulation leads to a dense system of equations which is solved iteratively using a preconditioned Generalized Minimal Residual (GMRES) algorithm, a rapidly converging Krylov-subspace method. Finally, since the system of equations is dense, the matrix-vector products required for each iteration of GMRES are expensive, and to reduce their cost, a multipole-accelerated algorithm is used. The combination of these techniques yields a packaging analysis program whose computational complexity grows nearly linearly with the number of filaments required to discretize the conductors.

In this paper we describe the above algorithms in some detail. We start, in the next section, with background material on the discretization of the integral equation formulation for magnetoquasistatic analysis of a multiconductor system. In section three, we describe the mesh formulation and iterative solver, and in section four, we describe how to multipole-accelerate the GMRES iterative matrix solution algorithm. Section five describes approaches for accelerating GMRES convergence using several different preconditioners. Experimental results are given in section six, and they demonstrate that the algorithms in FastHenry are accurate and more than an order of magnitude faster and use an order of magnitude less memory than approaches based on direct factorization.

2 Background Formulation

Inductance extraction is the process of computing the complex frequency-dependent impedance matrix of a multiterminal conductor, such as an electrical package, under the magnetoquasistatic approximation [1]. For a system with n terminal pairs, let $Z_r(\omega) \in \mathbf{C}^{n \times n}$ denote this impedance matrix at frequency ω. Then,

$$Z_r(\omega)\tilde{I}_s(\omega) = \tilde{V}_s(\omega), \tag{1}$$

where $\tilde{I}_s, \tilde{V}_s \in \mathbf{C}^n$ are the vectors of terminal current and voltage phasors, respectively [2]. Note that column i of of Z_r can be computed by setting entry i of $\tilde{I}_s$ to one, the rest to zero, and then computing the resulting voltage vector $\tilde{V}_s$. The i^{th} column of Z_r is then given by $\tilde{V}_s$.

2.1 Integral Equation

Several integral equation-based approaches have been used to derive the Z_r associated with a given package or interconnect structure [3, 4]. These integral formulations are derived by assuming sinusoidal steady-state, and then applying the magnetoquasistatic assumption that the displacement current, $\epsilon\omega\mathbf{E}$, is negligible. Given this, the vector potential, $\mathbf{A}$, can be related to the resistive current, $\mathbf{J}$, by

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi} \int_{V'} \frac{\mathbf{J}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} dv' \tag{2}$$

where $\nabla \times \mathbf{A} = \mu\mathbf{H}$, $\nabla \cdot \mathbf{A} = 0$, and where V' is the volume of all conductors.

Also, directly from Faraday's Law and the definition of $\mathbf{A}$, it follows that

$$\mathbf{E} = -j\omega\mathbf{A} - \nabla\Phi \tag{3}$$

where Φ is referred to as the scalar potential.

Assuming the ideal conductor constitutive relation, $\mathbf{J} = \sigma\mathbf{E}$, and combining this relation with (2) and (3) results in

$$\frac{\mathbf{J}(\mathbf{r})}{\sigma} + \frac{jw\mu}{4\pi} \int_{V'} \frac{\mathbf{J}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} dv' = -\nabla\Phi(\mathbf{r}). \tag{4}$$

Then, by simultaneously solving (4) with the current conservation equation,

$$\nabla \cdot \mathbf{J} = 0, \tag{5}$$

conductor current densities and the scalar potential can be computed.

2.2 Discretization

Given the magnetoquasistatic assumption, the current within a long thin conductor can be assumed to flow parallel to its surface, as there is no charge accumulation on the surface. Thus, for long thin structures such as

pins of a package or connector, the conductor can be divided into *filaments* of rectangular cross-section inside which the current is assumed to flow along the length of the filament. In order to properly capture skin and proximity effects in these long, thin conductors, the cross section of the conductor can be divided into a bundle of parallel filaments as shown in Fig. 1. It is also possible to use the filament approach for planar structures, such as ground planes, where the current distribution is two-dimensional. In such cases, a grid of filaments must be used, as in Fig. 2. Once the conductors are discretized into filaments, the interconnection of the current filaments can be represented with a planar graph, where the n nodes in the graph are associated with connection points between filaments, and the b branches in the graph represent the filaments into which each conductor segment is discretized.

If the current density inside each filament is assumed to be constant, then the approximation to the unknown current distribution can then be written as

$$\mathbf{J}(\mathbf{r}) \approx \sum_{i=1}^{b} I_i w_i(\mathbf{r}) \mathbf{l_i} \tag{6}$$

where I_i is the current inside filament i, $\mathbf{l_i}$ is a unit vector along the length of the filament and $w_i(\mathbf{r})$ is the weighting function which has a value of zero outside filament i, and $1/a_i$ inside, where a_i is the cross sectional area. By defining the inner product of two vector functions, $\mathbf{a}$ and $\mathbf{b}$, by

$$(\mathbf{a}, \mathbf{b}) = \int_V \mathbf{a} \cdot \mathbf{b}\, dv \tag{7}$$

and following the method of moments [5], a system of b equations can be generated by taking the inner product of each of the weighting functions with the vector integral equation, (4). This gives

$$\left(\frac{l_i}{\sigma a_i}\right) I_i + j\omega \sum_{j=1}^{b} \left(\frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\mathbf{r} - \mathbf{r}'|} dV'\, dV \right) I_j = \frac{1}{a_i} \int_{a_i} (\Phi_A - \Phi_B)\, dA \tag{8}$$

where l_i is the length of filament i, a_i is the cross section, Φ_A and Φ_B are the potentials on the filament end faces, and V_i and V_j' are the volumes of filaments i and j, respectively. Note that the right hand side of (8) results from integrating $\nabla\Phi$ along along the length of the filament, and that the right hand side is effectively the average potential on face A minus the average on face B.

In matrix form, (8) becomes

$$(R + j\omega L) I_b = \tilde{\Phi}_A - \tilde{\Phi}_B \tag{9}$$

where $I_b \in \mathbf{C}^b$ is the vector of b filament currents,

$$R_{ii} = \frac{l_i}{\sigma a_i} \tag{10}$$

is the $b \times b$ diagonal matrix of filament dc resistances,

$$L_{ij} = \frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\mathbf{r} - \mathbf{r}'|} dV' \, dV \tag{11}$$

is the $b \times b$ dense matrix of partial inductances, and $\tilde{\Phi}_A$ and $\tilde{\Phi}_B$ are the averages of the potentials over the cross sections of the filament faces [6, 3]. This can also be written as

$$ZI_b = V_b \tag{12}$$

where $Z = R + j\omega L \in \mathbf{C}^{b \times b}$ is called the branch impedance matrix and $V_b = \tilde{\Phi}_A - \tilde{\Phi}_B$ is the vector of branch voltages.

2.3 Nodal Analysis Formulation

Current conservation, (5), must be enforced at each of n nodes where filaments connect. This can be written as

$$AI_b = I_s \tag{13}$$

where $A \in \mathbf{R}^{n \times b}$ is the branch incidence matrix and I_s is the mostly zero vector of source currents. Each row in A corresponds to a filament connection node, and each column to a filament current. Column i in A has two nonzero entries; -1 in the row corresponding to the node from which filament i's current leaves, and $+1$ in the row corresponding to the node to which filament i's current enters.

Since $\nabla^2 \Phi = 0$, the branch voltages, V_b, can be derived from a set of node voltages, denoted $\tilde{\Phi}_n$, as in

$$A^t \tilde{\Phi}_n = V_b. \tag{14}$$

Combining (12), (13), and (14) yields

$$AZ^{-1}A^t \tilde{\Phi}_n = I_s. \tag{15}$$

Notice that column i of Z_r can now be computed by appropriately setting the source currents, I_s, that correspond to $\tilde{I}_{s_i}$ equal to one (unit current through conductor i), and then solving (15) to compute the node

voltages, $\tilde{\Phi}_n$. The difference of appropriate node voltages gives the entries of $\tilde{V}_s$, the vector of voltages across each of the conductors.

In most programs, the dense matrix problem in (15) is solved with some form of Gaussian elimination or direct factorization. These programs avoid forming Z^{-1} explicitly by reformulating (15) into the sparse tableau form,

$$\begin{bmatrix} Z & -A^t \\ A & 0 \end{bmatrix} \begin{bmatrix} I_b \\ \tilde{\Phi}_n \end{bmatrix} = \begin{bmatrix} 0 \\ I_s \end{bmatrix}. \tag{16}$$

Using direct factorization to solve (16) implies that the calculation grows at least as b^3, where again b is the number of current filaments into which the system of conductors is discretized [7]. For complicated packaging structures, b can exceed ten thousand, and solving (16) with direct factorization will take days, even using a high performance scientific workstation.

3 The Mesh-Based Approach

The obvious approach to trying to reduce the cost of solving (16) is to apply iterative methods. However, such methods converge slowly because (16) contains equations of two different types. Another approach is to reformulate the equations using mesh analysis [2], and then apply an iterative method.

3.1 Mesh Analysis

In mesh analysis, a mesh is any loop of branches in the graph which does not enclose any other branches. Also, the currents flowing around any mesh in the network are the unknowns, rather than node voltages. Mesh analysis is easiest to describe if it is assumed that sources generate explicit branches in the graph representing the discretized problem. Kirchoff's voltage law, which implies that the sum of branch voltages around each mesh in the network must be zero, is represented by

$$MV_b = V_s \tag{17}$$

where V_b is the vector of voltages across each branch except for the source branches, V_s is the mostly zero vector of source branch voltages, and $M \in \mathbf{R}^{m \times b}$ is the mesh matrix, where $m = b' - n + 1$ is the number of meshes and b' is the number of current filaments plus the number of source branches.

Algorithm 1 (GMRES Algorithm for $Ax = b$)

```
guess x^0
for k = 0, 1, ... until converged {
  Compute the error, r^k = b - Ax^k
  Find x^{k+1} to minimize r^{k+1}
    based on x^i and r^i, i = 0, ..., k
}
```

The relationship between branch currents and branch voltages given in (12) still holds, and the mesh currents, that is, the currents around each mesh loop, satisfy

$$M^t I_m = I_b, \tag{18}$$

where $I_m \in \mathbf{C}^m$ is the vector of mesh currents. Note that each of the entries in the terminal current vector, $\tilde{I}_s$, will be identically equal to some entry in I_m. And similarly, each of the entries in the terminal voltage vector, $\tilde{V}_s$, will correspond to some entry in V_s. Fig. 3 illustrates the definitions of the above quantities for a single conductor example.

Combining (18) with (17) and (12) yields

$$MZM^t I_m = V_s. \tag{19}$$

The matrix MZM^t is *easily* constructed directly. To compute the i^{th} column of the reduced admittance matrix, $Y_r = Z_r^{-1}$, solve (19) with a V_s whose only nonzero entry corresponds to $\tilde{V}_{s_i}$, and then extract the entries of I_m associated with the source branches.

3.2 Using an iterative solver

The standard approach to solving the complex linear system in (19) is Gaussian elimination, but the cost is m^3 operations. For this reason, inductance extraction of packages requiring more than a few thousand filaments is considered computationally intractable. To improve the situation, consider using a conjugate-residual style iterative method like GMRES [8]. Such methods have the general form given in Algorithm 1.

Note that the GMRES algorithm can be directly applied to solving (19), because the matrix MZM^t is easily constructed explicitly. This is not the case for the nodal formulation, (15), as either the Z matrix must first be

inverted or the sparse tableau form in (16) must be used. The sparse tableau form is disadvantageous because it is a much larger system of equations and it is difficult to solve iteratively as it contains two types of equations.

When applying the GMRES algorithm to solving (19), the cost of each iteration of the GMRES algorithm is at least order m^2 operations. This follows from the fact that evaluating $\mathbf{r}^k$ implies computing a matrix-vector product, where in this case the matrix is MZM^t and is dense. Note also that forming MZM^t explicitly requires order m^2 storage.

4 The Multipole Approach

It is possible to approximately compute $MZM^t I_m^k$ in order b operations using a hierarchical multipole algorithm for electrostatic analysis [9]. Such algorithms also avoid explicitly forming MZM^t, and so reduce the memory required to order b.

4.1 The Electrostatic Analogy

To show how a multipole algorithm can be applied to computing $MZM^t I_m^k$, consider expanding the matrix-vector product by separating Z into its real and imaginary parts,

$$MZM^t I_m^k = MRM^t I_m^k + j\omega MLM^t I_m^k. \tag{20}$$

The $MRM^t I_m^k$ term can be computed in order m operations because R is the diagonal matrix derived from the filament resistances, and M is the sparse mesh matrix with order m nonzero elements. Forming $MLM^t I_m^k$ is more expensive, requiring order m^2 operations as L is dense. From (8) and (11) it is clear that entry i of the portion of the product, $LM^t I_m^k$, or equivalently LI_b, is

$$(LI_b)_i = \sum_{j=1}^{b} \left(\frac{\mu}{4\pi a_i a_j} \int_{V_i} \int_{V_j'} \frac{\mathbf{l_i} \cdot \mathbf{l_j}}{|\mathbf{r} - \mathbf{r}'|} dV' \, dV \right) I_j. \tag{21}$$

In terms of the vector potential,

$$(LI_b)_i = \frac{1}{a_i} \int_{V_i} \mathbf{A}(\mathbf{r}) \cdot \mathbf{l_i} dV, \tag{22}$$

since substituting (6) in (2) gives

$$\mathbf{A}(\mathbf{r}) = \frac{\mu}{4\pi} \sum_{j=1}^{b} \left(\int_{V_j'} \frac{\mathbf{l_j}}{|\mathbf{r} - \mathbf{r}'|} dV' \, dV \right) \frac{I_j}{a_j}. \tag{23}$$

The above decomposition shows that LI_b can be evaluated by integrating the vector potential $\mathbf{A}$ over each filament [10]. Also, from (23), each component of the vector potential can be considered a scalar electrostatic potential generated by to a collection of charges. That is, for $p \in \{1,2,3\}$, the p^{th} component of $\mathbf{A}(\mathbf{r})$, denoted $\psi_p(\mathbf{r}) \in \mathbf{C}$, is a scalar potential given by

$$\psi_p(\mathbf{r}) = \frac{\mu}{4\pi} \sum_{j=1}^{b} \left(\int_{V_j'} \frac{(\mathbf{l_j})_p}{|\mathbf{r} - \mathbf{r}'|} dV' \, dV \right) \frac{I_j}{a_j}. \tag{24}$$

and therefore $(I_j/a_j)\,(\mathbf{l_j})_p$ can be interpreted as a charge density due to filament j.

The electrostatic analogy implies that LI_b can be computed by combining the results of evaluating the electrostatic potential along b filaments due to b filament charges *for three separate sets of filament charges.* It is the evaluation of these electrostatic potentials which can be accelerated with the hierarchical multipole algorithm [9]. That is, the electrostatic potential due to b charges can be evaluated at b points in order b operations using the hierarchical multipole algorithm. Therefore, by using the multipole algorithm three times, LI_b can be computed in order b operations.

4.2 The Hierarchical Multipole Algorithm

A complete description of the fast multipole algorithm is quite lengthy, and can be found in [9], or in the context of 3-D capacitance extracation, in [11, 12]. To see roughly what the algorithm exploits to achieve its efficiency consider the two configurations given in Figs. 4 and 5, depicted in 2-D for simplicity. In either figure, the obvious approach to determining the electrostatic potential at the n_1 evaluation points from the n_2 point-charges involves $n_1 * n_2$ operations: at each of the n_1 evaluation points one simply sums the contribution to the potential from n_2 charges.

An accurate approximation for the potentials for the case of Fig. 4 can be computed in far fewer operations using *multipole expansions*, which exploit the fact that $r >> R$ (defined in Fig. 4). That is, the details of the distribution of the charges in the inner circle of radius R in Fig. 4 do not strongly affect the potentials at the evaluation points outside the outer circle of radius r. It is also possible to compute an accurate approximation for the potentials at the evaluation points in the inner circle of Fig. 5 in far fewer than $n_1 * n_2$ operations using *local expansions*, which again exploit the fact that $r >> R$ (as in Fig. 5). In this second case, what can be ignored are

the details of the evaluation point distribution.

This brief description of the hierarchical multipole algorithm is only intended to make clear that the algorithm's efficiency stems from coalescing charges and evaluation points using multipole and local expansions. A few points about the algorithm's application to computing LI_b should be considered however. When filaments are very near each other, a multipole expansion representation would lead to excessive error, so the interaction is evaluated directly using (11). Direct evaluations are also used for small groups of distant filaments when the computation required to build the multipole and local expansions exceeds the direct evaluation cost, thus making the algorithm adaptive. Therefore, when the hierarchical multipole algorithm is used to compute LI_b, the evaluation is typically a combination of three sets of multipole and local expansion evaluations for the three components of the vector potential, along with a single set of nearby-filament direct evaluations.

5 Accelerating Iteration Convergence

In general, the GMRES iterative method applied to solving (19) can be significantly accelerated by *preconditioning* if there is an easily computed good approximation to the inverse of MZM^t. We denote the approximation to $(MZM^t)^{-1}$ by P, in which case preconditioning the GMRES algorithm is equivalent to using GMRES to solve

$$(MZM^t)Px = V_s \tag{25}$$

for the unknown vector x. The mesh currents are then computed with $I_m = Px$. Clearly, if P is precisely $(MZM^t)^{-1}$, then (25) is trivial to solve, but then P will be very expensive to compute.

An easily computed good approximation to $(MZM^t)^{-1}$ can be constructed by noting that the most tightly coupled meshes are ones which are physically close. To exploit this observation, for each mesh i, the submatrix of MZM^t corresponding to all meshes near mesh i is inverted directly. Then, the row of the inverted submatrix associated with mesh i becomes the i^{th} row of P. This is illustrated in Fig. 6, where the submatrix is drawn as a block along the diagonal for illustration. We refer to this preconditioner as a "local-inversion" preconditioner, because it is formed by inverting physically localized problems.

This preconditioner works well for pin-connect and other similar structures for which most of the meshes are small and thus what is 'local' is obvious. The fact that most of the meshes are small can be observed from Fig. 3

by noticing that most of the meshes, such as those associated with I_{m3}, I_{m6}, and I_{m9}, are small and consist of only two physically close filaments. Comparatively, there are relatively few large meshes, such as I_{m10}, each which result from the presence of an external source and may include many filaments which span much of the physical problem domain. Therefore, much of the problem can be physically close to these large meshes. For this reason, the large meshes associated with sources cannot be included in the preconditioner, otherwise excessively large subproblems will be inverted directly. Since there are relatively few of these large meshes in a pin-connect structure and they are physically separate (only one per pin), not including the large meshes when forming the preconditioner does not significantly slow convergence.

For ground-plane problems, with possibly hundreds of external sources, the performance of local-inversion is severely degraded. As for pin-connects, many of the meshes are small; in this case, most meshes require four filaments (See Fig. 2). Each external source, however, requires the formation of a large mesh traversing the ground plane between its two contact points as shown in Fig. 7. With hundreds of these meshes, many of them physically close and possibly partially overlapping, local-inversion without large meshes becomes ineffective.

Other approaches to preconditioning which might help account for large mesh interactions involve somehow approximately factoring MZM^t or directly factoring an approximation to MZM^t. Approximately factoring MZM^t using an approach like incomplete LU factorization is ineffective however, because the diagonals of MZM^t are not necessarily greater than the sum of the off-diagonals and therefore ignored terms can become more significant. Another approach is to sparsify MZM^t, possibly based on the sparsity pattern of MRM^t, and then directly factor the sparsified matrix to construct a preconditioner. This is not necessarily effective since by discarding terms, the matrix may become far from positive definite in both the high and low frequency limits. This would make the preconditioned product, $(MZM^t)P$, also far from positive definite in those limits. Instead, we consider sparsifying the branch inductance matrix, L, in some positive-definite manner, and then generate the preconditioner by directly factoring the sparse result, $M(R + j\omega L')M^t$, where L' is the sparsified branch inductance matrix [13]. Note that in the low frequency limit, the whole problem is factored exactly. The simplest choice for L' is the diagonal of L. For the printed circuit board example described below, between the low and high frequency limits, the required GMRES iterations monotonically and asymptotically increase toward the high

frequency limit (See Fig. 8). Other choices for positive-definite sparsifications of L are possible, but care must be taken since with additional terms other than the diagonal of L, the MLM^t matrix quickly becomes more dense and thus more expensive to factor.

To compare the relative merits of the above preconditioners, consider the two industrial examples in Figures 9 and 10. Fig. 9 is thirty-five pins of a 68-pin cerquad package and Fig. 10 is a portion of a printed circuit board (PCB) that would be placed underneath a PGA package. The PCB example consists of two resistive reference planes sandwiching 255 copper lines. Each plane in the PCB has 53 external contacts not shown in the figure. For this experiment, the cerquad package was discretized into 1308 filaments which corresponds to 1125 meshes and each reference plane in the PCB was discretized into a 20×20 grid of meshes giving a total 1099 meshes including the copper lines. The solution error at high frequency as a function of GMRES iteration is plotted in Fig. 11 for the cerquad example, and in Fig. 12 for PCB example. As the plots clearly show, preconditioning substantially accelerates convergence, but the cerquad package example converges slightly more rapidly with the local-inversion preconditioner, and the PCB example converges substantially more rapidly with the sparsified-L preconditioner. Thus, using the sparsified-L preconditioner is generally effective, but for problems without ground planes having many external contacts, local-inversion may be slightly more effective.

6 Results

In this section we demonstrate the accuracy, utility, and computational efficiency of our multipole-accelerated version of FastHenry. For an accuracy comparison, we consider the portion of a 68-pin package, shown previously in Fig. 9. Each pin consists of eight to ten conductor sections. We discretized each section into 2×2 filaments. This generated a problem with 1368 branches for which MZM^t is a 1061×1061 dense matrix. Note, using only four filaments per section is hardly sufficient to model the skin effect, though with the coarse discretization, the problem is small enough to make possible an accuracy comparison between direct factorization, GMRES, and multipole-accelerated GMRES.

For the example package, the mutual inductance between pins 1 and 2 (labeled clockwise from the right) is much larger than the mutual inductance between pins 1 and 18 which are perpendicular to each other except

pin pair	direct	gmres	multipole
1 to 2	5.31870e+00	5.31867e+00	5.31403e+00
1 to 18	3.68292e-02	3.68223e-02	3.71027e-02

Table 1: Comparison of the accuracy of the computed inductance matrix entries between direct factorization, GMRES with explicit matrix-vector products, and the multipole-accelerated GMRES algorithm.

for their vertical sections. To show that the approximations used by the hierarchical multipole algorithm are sufficiently well-controlled that small coupling inductances are computed accurately, consider the results in Table 1. The mutual inductance between pins 1 and 18 is more than two orders of magnitude smaller than the mutual inductance between pins 1 and 2, yet the solution computed using the multipole-accelerated algorithm is still within one percent of the solution computed using direct factorization.

As an example of the utility of frequency dependent inductance extraction possible with FastHenry, consider the two cases of computing the mutual inductance between a pair of PC board traces over a resistive ground plane, as shown in Fig. 13, and the same pair of traces over a divided ground plane, as in Fig. 14. The traces have their return paths through the ground plane. For the divided plane case, the two portions are electrically connected with short resistive 'tethers' toward the outer edges as shown. The traces are 8 mils wide, 1 mil thick, 8 mils above the 1 mil thick ground plane, and their center to center distance is 32 mils.

If a current source is connected to one of the traces, current will flow down the trace and return through the plane. For the solid plane case, the current in the plane with a DC source produces a current distribution pattern which spreads to fill the width of the plane. Similarly in the divided plane case, the current spreads throughout most of the plane, but narrows as it crosses the tethers. The situation at a high frequency is quite different. For the solid plane, the ground plane return current is concentrated directly underneath the trace, but for the divided plane the current leaves the path underneath the trace to cross the tethers (See Figures 15 and 16).

This difference has a marked effect on the mutual inductance between the traces as the frequency rises. For the solid plane, as the frequency rises, the current gathers underneath the trace and the mutual inductance drops by two orders of magnitude, however for the divided plane, little decrease is observed with frequency (See Fig. 17).

To demonstrate the computational efficiency of FastHenry, we successively refined a coarse discretization of the ground plane of the example shown in Fig. 13. As the discretization of the plane is refined, the size of the

problem will grow quickly, making the memory and CPU time advantage of the multipole-accelerated, local-inversion-preconditioned GMRES algorithm apparent (see Figures 18 and 19). As the graphs clearly indicate, the cost of direct factorization grows like m^3, the cost of explicit GMRES grows as m^2, but the cost of multipole-accelerated GMRES grows only linearly with m. In addition, the memory requirement for multipole-accelerated GMRES algorithm grows linearly with m, but grows like m^2 for either explicit GMRES or direct factorization. In particular, for a $12,802$ mesh problem, the multipole accelerated algorithm is more than two orders of magnitude faster than direct factorization, and uses an order of magnitude less time and memory than explicit GMRES.

A significantly more complex problem and one that uses the sparsified-L preconditioner, is the high frequency analysis of a portion of a PCB described previously and shown in Fig. 10. To properly model the current flow in the two reference planes surrounding the copper lines, the planes must be finely discretized. Here again, as the discretization is refined, the cost of direct factorization grows like m^3, the cost of explicit GMRES grows as m^2, but the cost of multipole-accelerated GMRES grows only linearly with m as shown in Fig. 20. For this PCB example, the associated impedance matrix is 18x18, while the pair of traces over plane example has only a 2x2 impedance matrix. Thus, nine times as many GMRES solutions are required to compute the PCB example's impedance matrix. Even so, for a $10,000$ mesh problem, the multipole-accelerated GMRES algorithm is still over an order of magnitude faster in computation time.

7 Conclusions and Acknowledgements

In this paper, we show that 3-D inductance extraction can be substantially accelerated using the hierarchical multipole algorithm. Our multipole-accelerated inductance extraction program, FastHenry, was shown to be more than two orders of magnitude faster than direct factorization when used to extract the inductance matrix for realistic packaging examples. In addition, the multipole-accelerated algorithm uses an order of magnitude less time and memory than the explicit GMRES algorithm given in [14]. Finally, the sparsified preconditioner insures rapid convergence even for very irregular problems, making FastHenry a robust program. The authors would like to thank Keith Nabors, Songmin Kim, Dr. Sami Ali, and Joel Phillips for their help in understanding inductance. In addition, the authors would like to thank Dr. Albert Ruehli, Prof. Raj Mittra, and Dr. Colin Gordon for their

helpful suggestions.

References

[1] H. A. Haus and J. R. Melcher, *Electromagnetic Fields and Energy.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

[2] C. A. Desoer and E. S. Kuh, *Basic Circuit Theory.* New York: McGraw-Hill, 1969.

[3] A. E. Ruehli, "Survey of computer-aided electrical analysis of integrated circuit interconnections," *IBM Journal of Research and Development*, vol. 23, pp. 626–639, November 1979.

[4] A. C. Cangellaris, J. L. Prince, and L. P. Vakanas, "Frequency-dependent inductance and resistance calculation for three-dimensional structures in high-speed interconnect systems," *IEEE Transactions on Components, Hybrids, and Manufacturing Technology*, vol. 13, pp. 154–159, March 1990.

[5] R. F. Harrington, *Field Computation by Moment Methods.* New York: MacMillan, 1968.

[6] W. T. Weeks, L. L. Wu, M. F. McAllister, and A. Singh, "Resistive and inductive skin effect in rectangular conductors," *IBM Journal of Res. and Develop.*, vol. 23, pp. 652–660, November 1979.

[7] A. E. Ruehli and P. A. Brennan, "Efficient capacitance calculations for three-dimensional multiconductor systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 21, pp. 76–82, February 1973.

[8] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, July 1986.

[9] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems.* Cambridge, Massachusetts: M.I.T. Press, 1988.

[10] M. Kamon, M. J. Tsuk, and J. White, "FastHenry: A multipole-accelerated 3-D inductance extraction program." To appear in *Proceedings of the 30th Design Automation Conference*, June 1993.

[11] 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, pp. 1447–1459, November 1991.

[12] K. Nabors, S. Kim, and J. White, "Fast capacitance extraction of general three-dimensional structures," *IEEE Transactions on Microwave Theory and Techniques*, vol. 40, pp. 1496–1506, November 1992.

[13] M. Kamon and J. R. Phillips, "Preconditioning techniques for constrained vector potential integral equations, with application to 3-D magnetoquasistatic analysis of electronic packages," in *Proceedings of the Colorado Conference on Iterative Methods, Breckenridge, Colorado*, April 1994.

[14] M. Kamon, M. Tsuk, C. Smithhisler, and J. White, "Efficient techniques for inductance extraction of complex 3-d geometries," in *Proceedings of Int. Conf. on Comp. Aided Design*, November 1992.

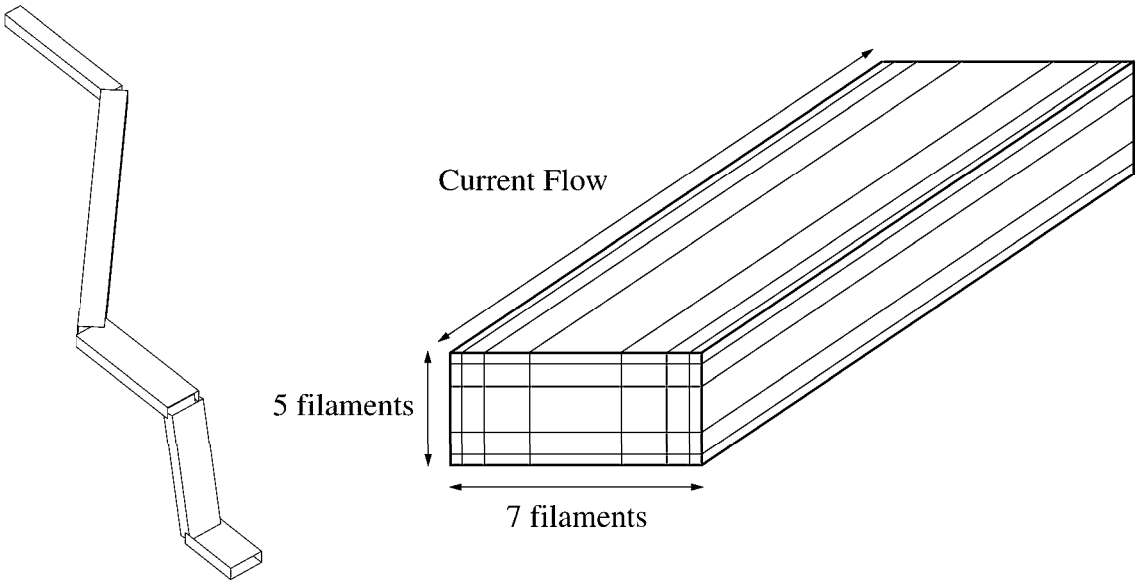

Figure 1: Single pin of a pin-connect divided into 5 sections, each of which is a bundle of 35 filaments.

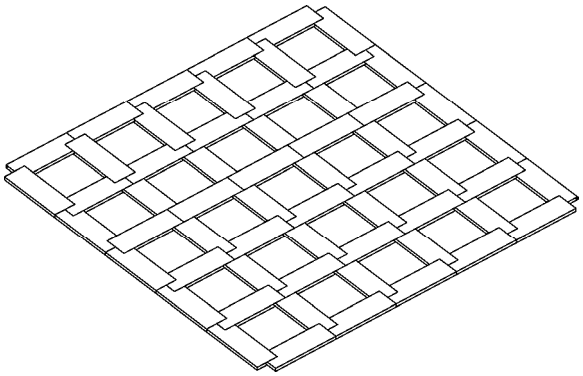

Figure 2: Discretization of a Ground Plane. Segments are one-third actual width.

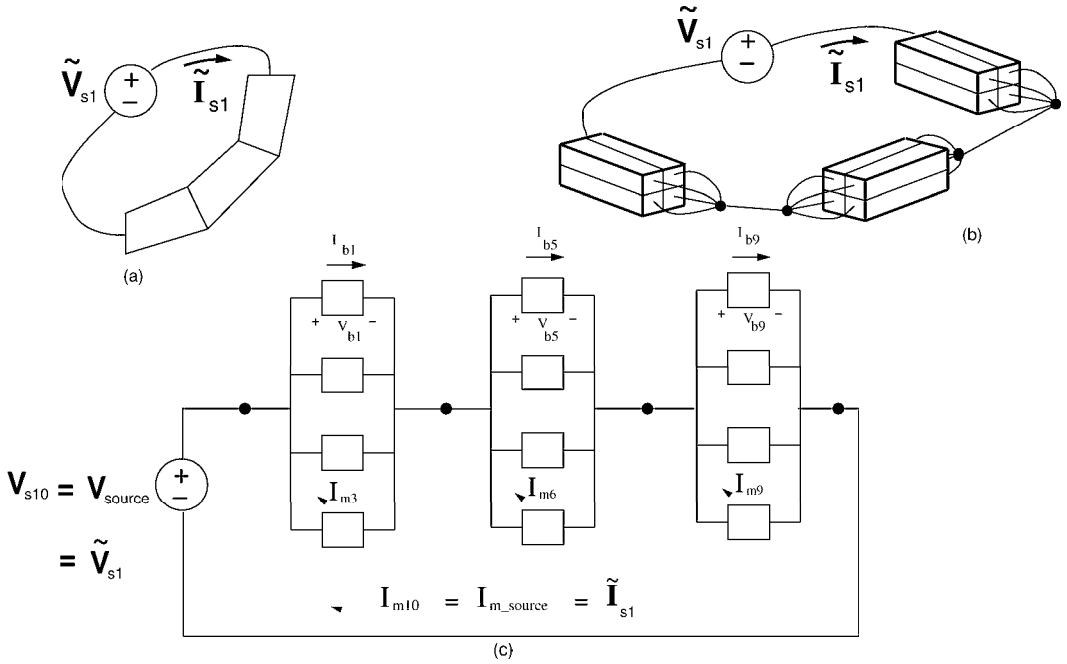

Figure 3: One conductor, (a) as piecewise-straight sections, (b) discretized into filaments, (c) modelled as a circuit.

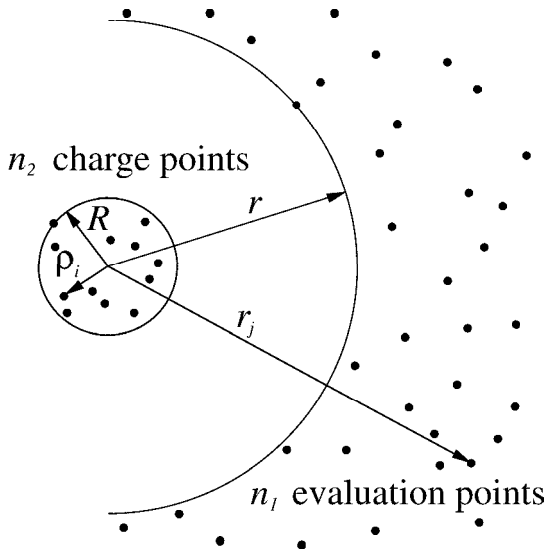

Figure 4: The evaluation point potentials are approximated with a multipole expansion.

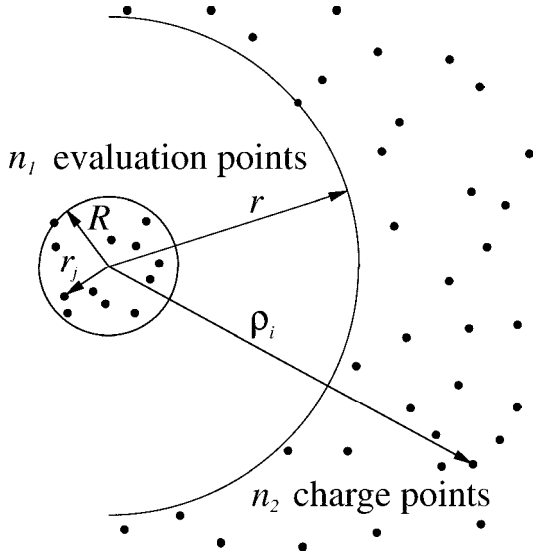

Figure 5: The evaluation point potentials are approximated with a local expansion.

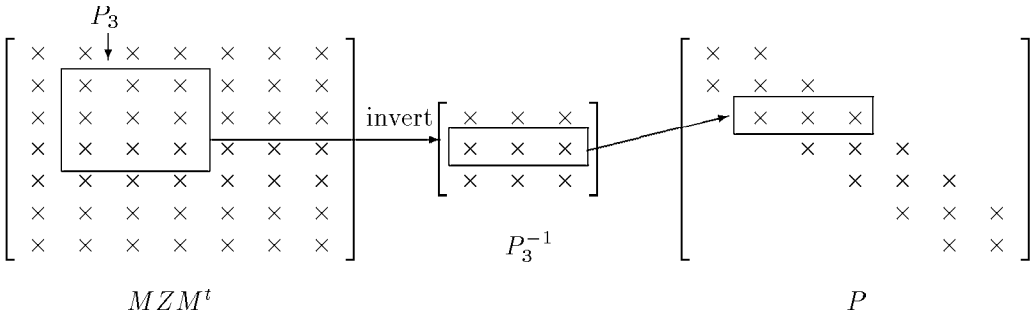

Figure 6: The steps leading to the third row of the preconditioner P ("$\times$" denotes a non-zero element). Note that for illustration, P_3 is drawn as a block along the diagonal of P.

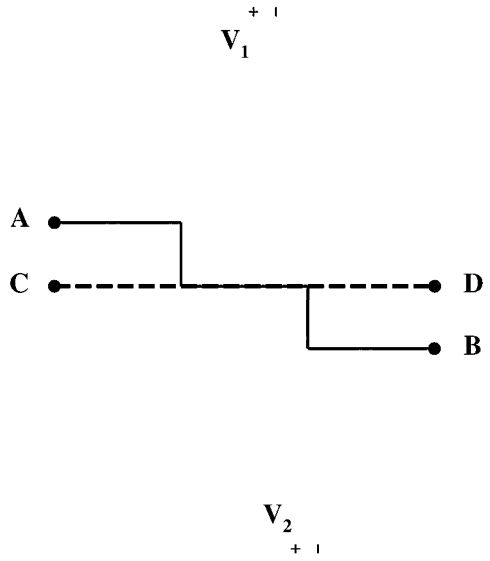

Figure 7: Two ground plane meshes due to external sources. One mesh includes the filaments along the path from point A to B and the other from C to D. The filaments that make up the plane are drawn one-third their actual width for illustration.

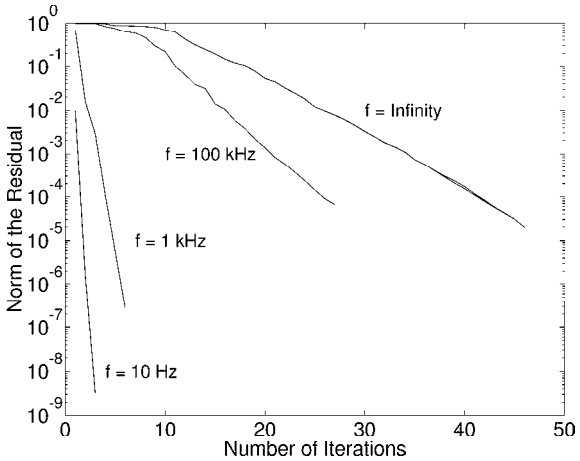

Figure 8: Convergence of GMRES using sparsified-L preconditioner on printed circuit board package at various frequencies

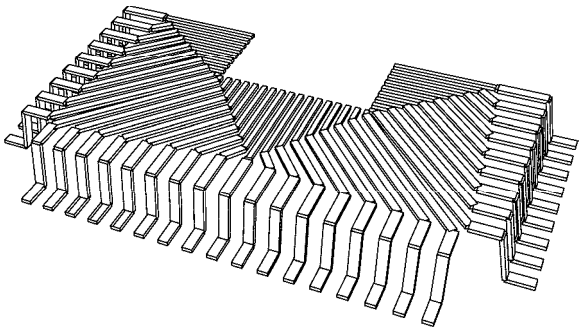

Figure 9: Half of a cerquad package. Thirty-five pins shown.

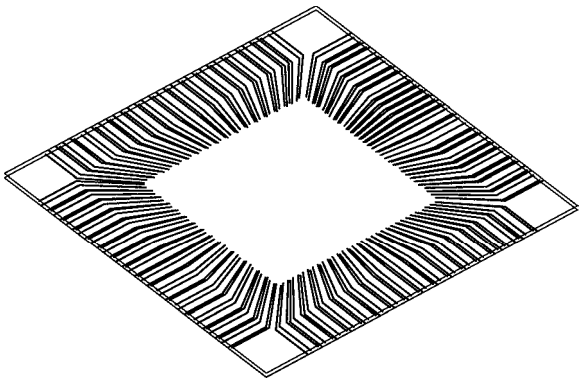

Figure 10: A portion of a printed circuit board directly underneath a PGA package. Two resistive reference planes sandwiching 255 copper lines. Only the outline of the planes is drawn.

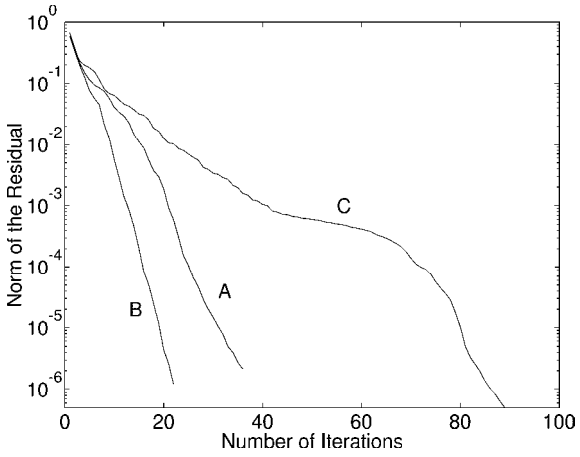

Figure 11: Convergence of GMRES applied to the cerquad example with sparsified-L preconditioning (A), local inversion preconditioning (B), and no preconditioning (C).

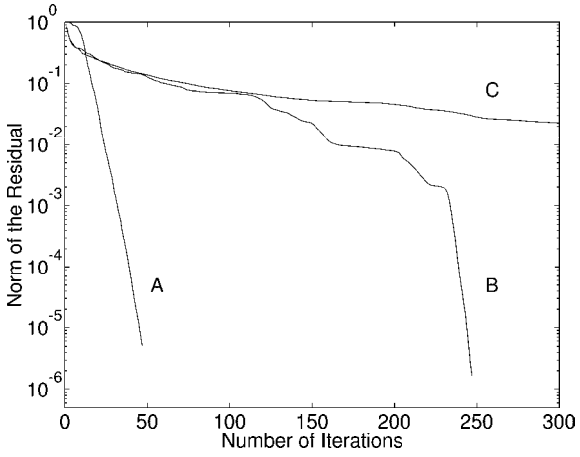

Figure 12: Convergence of GMRES applied to the PCB example with sparsified-L preconditioning (A), local inversion preconditioning (B), and no preconditioning (C).

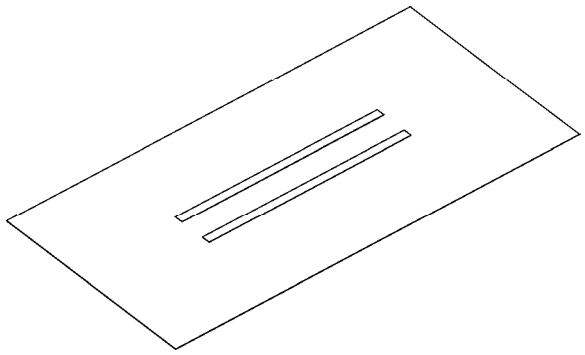

Figure 13: Two Traces over a Solid Ground Plane. The return path for the traces is through the plane. Traces are widened for illustration

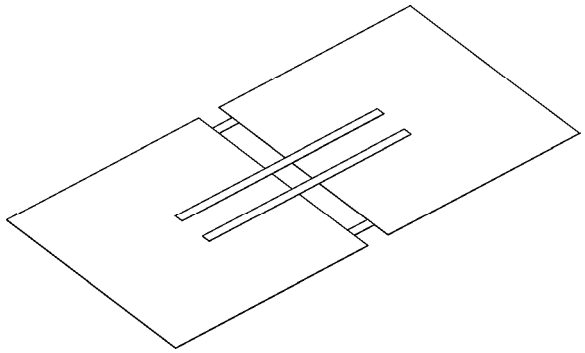

Figure 14: Two Traces over a Divided Ground Plane. The return path for the traces is through the plane. The divided portions are connected together toward the edges as shown. Traces are widened for illustration

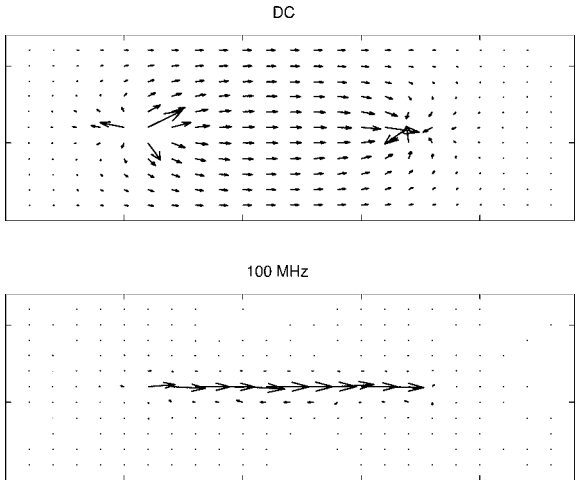

Figure 15: Current Distribution in Solid Ground Plane at DC and high frequency

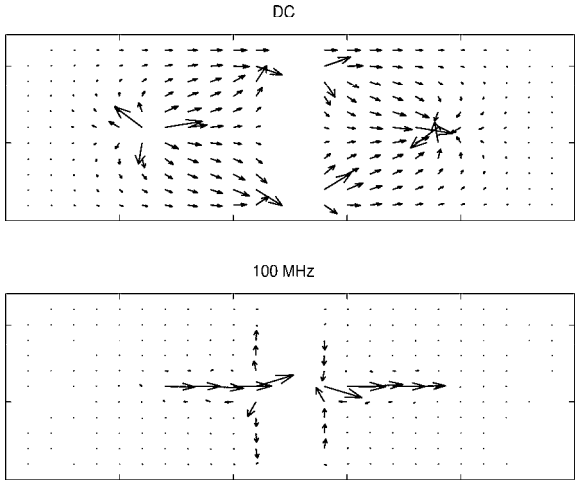

Figure 16: Current Distribution in Divided Ground Plane at DC and high frequency

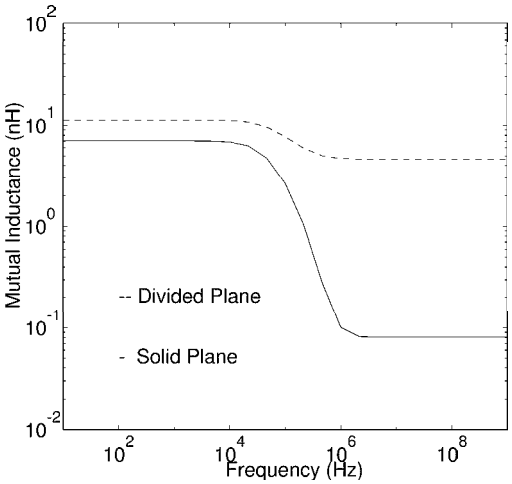

Figure 17: Mutual Inductance Between Traces

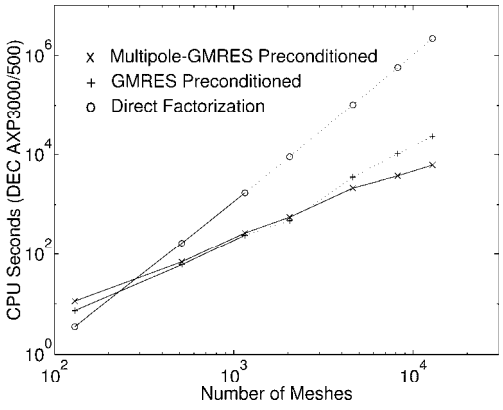

Figure 18: Comparison of the CPU time to compute the reduced inductance matrix for two traces over a solid plane using direct factorization, GMRES, and GMRES with multipole acceleration.

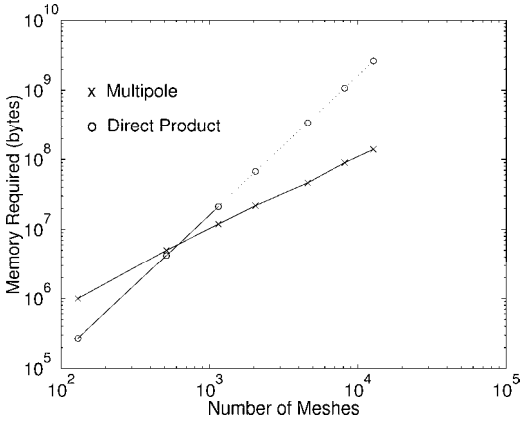

Figure 19: Comparison of the memory required using explicit matrix-vector products and using the multipole algorithm.

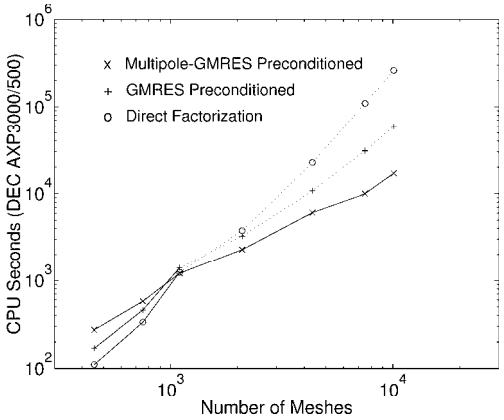

Figure 20: Comparison of the CPU time to compute the reduced inductance matrix for the PCB package using direct factorization, GMRES, and GMRES with multipole acceleration.

List of Figures