A Solenoidal Basis Method For Efficient Inductance Extraction *

Hemant Mahawar
Department of Computer Science
Texas A&M University
College Station, TX 77843
mahawarh@cs.tamu.edu

Vivek Sarin
Department of Computer Science
Texas A&M University
College Station, TX 77843
sarin@cs.tamu.edu

Weiping Shi
Department of Electrical Engineering
Texas A&M University
College Station, TX 77843
wshi@ee.tamu.edu

ABSTRACT

The ability to compute the parasitic inductance of the interconnect is critical to the timing verification of modern VLSI circuits. A challenging aspect of inductance extraction is the solution of large, dense, complex linear systems of equations via iterative methods. Accelerating the convergence of the iterative method through preconditioning is made difficult due to the non-availability of the system matrix. This paper presents a novel algorithm to solve these linear systems by restricting current to a discrete solenoidal subspace in which Kirchoff's law is obeyed, and solving a reduced system via an iterative method such as GMRES. A preconditioner based on the Green's function is used to achieve near-optimal convergence rates in several cases. Experiments on a number of benchmark problems illustrate the advantages of the proposed method over FastHenry.

Categories and Subject Descriptors

B.7.2 [**Integrating Circuits**]: Design Aids—*placement and routing, simulation, verification*

General Terms

ALGORITHMS

Keywords

Inductance extraction, interconnect, solenoidal basis, iterative methods, preconditioning

1. INTRODUCTION

The effect of inductance is increasingly felt on-chip, mainly due to long interconnect and high operation speed. Therefore fast and accurate inductance extraction is increasing important to the design and verification of VLSI circuits. There are three types of inductance extraction algorithms: *loop inductance*, *partial inductance* and *shape based*. The loop inductance algorithms are the most accurate but slowest, while the shape-based algorithms are the least accurate but the fastest. FastHenry [8] computes the loop inductance. Due to its high accuracy, FastHenry is often used as a reference for all other extraction algorithms. But FastHenry is very slow, and to improve its speed is one of the most challenging problems. Partial inductance was first proposed by Rosa and introduced to circuit design by Ruehli [13]. A number of algorithms have been proposed, such as Krauter [10] and He [5]. Partial inductance algorithms are faster than loop inductance algorithms. However it is shown that partial inductance without current return paths is inaccurate [3]. Shape-based algorithms, such as [9, 15], are fast but inaccurate for complex structures. For multilayer dielectric, a mixed potential integral equation method was proposed by Daniel [2] and a multilayer Green's function method was proposed by Liu [11].

In this paper, we study the extraction of loop inductance of 3D electrical conductors in uniform dielectric. We present a solenoidal basis method that gives a better formulation of the linear system compared to FastHenry, and we also present a novel preconditioning method that works with the solenoidal basis method. FastHenry and other 3D extraction algorithms for loop inductance are slow mainly due to the ill-conditioning of the linear systems. Therefore, our new formulation and preconditioning techniques address the key issue of this problem. Experimental results show that our techniques work well, and the new algorithm has several advantages over FastHenry.

2. MATHEMATICAL PRELIMINARIES

The impedance of an s-conductor geometry can be summarized by an $s \times s$ impedance matrix $\tilde{\mathbf{Z}}$. The l^{th} column of the impedance matrix is determined as follows: a *unit* current is sent through conductor l, and zero current is sent through other conductors. The numerical value of the potential difference between the two ends of conductor k gives values of the matrix element $\tilde{\mathbf{Z}}_{kl}$. The above procedure is repeated s times to compute all columns of $\tilde{\mathbf{Z}}$.

The current density $\mathbf{J}$ at a point r is related to the potential ϕ by the following integral equation [8]

$$\rho\mathbf{J}(\mathbf{r}) + j\omega \int_V \frac{\mu}{4\pi} \frac{\mathbf{J}(\mathbf{r}')}{\|\mathbf{r}-\mathbf{r}'\|} dV' = -\nabla\phi(\mathbf{r}), \tag{1}$$

where μ is the magnetic permeability, ρ is the resistivity, $\mathbf{r}$ is a three-dimensional position vector, ω is the frequency, $\|\mathbf{r}-\mathbf{r}'\|$ is the Euclidean distance between $\mathbf{r}$ and $\mathbf{r}'$, and $j = \sqrt{-1}$. The volume of conductors is denoted by V and incremental volume with respect

*This work has been supported in part by National Science Foundation grants NSF-CCR 9984400 and NSF-CCR 0113668 and the Texas Advanced Technology Program grant 000512-0266-2001.

to $\mathbf{r}'$ is denoted by dV'.

A numerical solution of (1) can be obtained by discretizing the conductors into n filaments $V_1, V_2, \ldots, V_n$. Assuming current flows along the length of the filament and current density is constant within each filament, a linear system of the following form is obtained:

$$[\mathbf{R} + j\omega\mathbf{L}]\,\mathbf{I} = \mathbf{V},$$

where $\mathbf{R}$ is an $n \times n$ diagonal matrix of filament resistances, $\mathbf{I}$ is the vector of filament currents, and $\mathbf{V}$ is the vector of the difference of potential between ends of each filament. The kth diagonal element of $\mathbf{R}$ is given by $\mathbf{R}_{kk} = \rho l_k / a_k$, where l_k is the length of the kth filament and a_k is the cross-sectional area of the kth filament. Let $\mathbf{u_k}$ denote the unit vector along the kth filament V_k. The elements of the inductance matrix $\mathbf{L}$ are given by

$$\mathbf{L}_{kl} = \frac{\mu}{4\pi} \frac{1}{a_k a_l} \int_{r_k \varepsilon V_k} \int_{r_l \varepsilon V_l} \frac{\mathbf{u}_k \cdot \mathbf{u}_l}{\|\mathbf{r}_k - \mathbf{r}_l\|} dV_k dV_l. \tag{2}$$

The current satisfies Kirchoff's current law at each node according to the constraints

$$\mathbf{B}^T \mathbf{I} = \mathbf{I}_d,$$

where $\mathbf{B}^T$ is the $m \times n$ branch index matrix and $\mathbf{I}_d$ is the vector of external source currents. Here, m and n denote the number of nodes and filaments, respectively, in the mesh. The vector $\mathbf{I}_d$ has non-zero values only for those nodes where external current source is applied. The potential difference across the filaments is expressed as

$$\mathbf{V} = \mathbf{B}\mathbf{V}_d,$$

where $\mathbf{V}_d$ is the node potential vector of length m. To compute the unknown filament currents and node potentials, the following linear system must be solved

$$\begin{bmatrix} \mathbf{R} + j\omega\mathbf{L} & -\mathbf{B} \\ \mathbf{B}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{V}_d \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{I}_d \end{bmatrix}. \tag{3}$$

A straightforward approach to solve this system involves removal of $\mathbf{I}$ by a block-step of Gaussian elimination. The resulting system is defined in terms of the unknowns $\mathbf{V}_d$ only:

$$\mathbf{B}^T [\mathbf{R} + j\omega\mathbf{L}]^{-1} \mathbf{B}\mathbf{V}_d = \mathbf{I}_d.$$

When solving this system by an iterative method, each iteration involves a matrix-vector product with the system matrix. In practice, this matrix is never computed explicitly. Instead, the matrix-vector product is computed as a sequence of three steps: (i) product with $\mathbf{B}$, (ii) solution of the system $[\mathbf{R} + j\omega\mathbf{L}]z = d$, and (iii) product with $\mathbf{B}^T$. The second step may require an inner iterative scheme, resulting in expensive outer iterations. Furthermore, with this kind of a complicated system matrix, it is difficult to find efficient preconditioners for the outer solver.

Alternately, one can compute a subspace for current that satisfies Kirchoff's law at each node. The linear system in (3) is transformed to a reduced system when current is restricted to this subspace, which is then solved by an iterative method. This approach is competitive only when the iterative scheme is preconditioned effectively. In the next section, we discuss how to compute the appropriate subspace and precondition the reduced system without explicitly constructing the matrix. Experiments in Section 4 demonstrate the superior properties of our approach over competing techniques.

3. THE SOLENOIDAL BASIS ALGORITHM

The difficulty in solving the system (3) arises partly due to the constraints imposed by Kirchoff's law on the flow of current at each node. These constraints are identical to divergence-free constraints in incompressible fluids that arise from conservation laws. A *solenoidal* basis is a basis for divergence-free functions that automatically satisfy conservation laws such as Kirchoff's law. The solenoidal basis algorithm constructs a basis for functions that satisfy Kirchoff's law. Expressing current in this basis leads to a reduced system which is solved by a preconditioned iterative method such as GMRES.

3.1 A Solenoidal Basis Approach

The solenoidal basis method is used to solve the problem

$$\begin{bmatrix} \mathbf{Z} & -\mathbf{B} \\ \mathbf{B}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I} \\ \mathbf{V}_d \end{bmatrix} = \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}, \tag{4}$$

where $\mathbf{Z} = \mathbf{R} + j\omega\mathbf{L}$ is the filament impedance matrix, $\mathbf{L}$ is an $n \times n$ dense, symmetric and positive definite matrix, $\mathbf{R}$ is a diagonal matrix, and $\mathbf{B}$ is an $n \times m$ sparse matrix. The filament current vector $\mathbf{I}$ has size n, the node potential vector $\mathbf{V}_d$ has size m, and the right hand side vector $\mathbf{F}$ has size n.

The main difference between (3) and (4) is that the boundary conditions are specified for current in the first system whereas they are specified for potential in the second system. To convert (3) to (4), we let $\mathbf{I} = \mathbf{I}' + \mathbf{I}_p$, where $\mathbf{I}_p$ is a current vector that satisfies the constraints $\mathbf{B}^T \mathbf{I}_p = \mathbf{I}_d$. The following linear system

$$\begin{bmatrix} \mathbf{Z} & -\mathbf{B} \\ \mathbf{B}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}' \\ \mathbf{V}_d \end{bmatrix} = \begin{bmatrix} -\mathbf{Z}\mathbf{I}_p \\ \mathbf{0} \end{bmatrix}$$

is solved for the unknown current $\mathbf{I}'$. The current vector $\mathbf{I}_p$ can easily be found by number of techniques. For instance, when the known branch current has unit magnitude, one can assign a unit current to filaments on an arbitrary path from node with input source current to the node with output source current (see Fig. 1). This approach can be extended to more general boundary conditions in a straightforward manner.

Since a unit current flowing in a closed loop in the mesh satisfies Kirchoff's law, it also satisfies the constraints imposed by $\mathbf{B}^T$. The solenoid basis method for (4) uses these mesh currents to represent the unknown current $\mathbf{I}$. This scheme is similar to the mesh current approach proposed in [6, 7] with the exception of the treatment of source current. In addition, the preconditioning proposed in section 3.2 is more powerful than those suggested in [7].

The null space of $\mathbf{B}^T$ represents a basis for current that obeys Kirchoff's law. Given a full-rank matrix $\mathbf{P} \in R^{n \times (n-m)}$ such that $\mathbf{B}^T \mathbf{P} = \mathbf{0}$, a current vector computed as follows

$$\mathbf{I} = \mathbf{P}x, \qquad x \in R^{(n-m)}$$

will satisfy the constraint $\mathbf{B}^T \mathbf{I} = 0$ for all x. A purely algebraic approach such as QR factorization of $\mathbf{B}$ cannot be used to compute $\mathbf{P}$ due to the prohibitive cost of computation and storage. We define a *unit* current flow in a closed loop as a *local* solenoidal function. Each such mesh current is represented as a vector and the set of these vectors forms the columns of $\mathbf{P}$. The local nature of these mesh currents leads to efficient computation and storage schemes for $\mathbf{P}$.

Since the current vector $\mathbf{I} = \mathbf{P}x$ automatically satisfies the constraint $\mathbf{B}^T \mathbf{I} = 0$, we only need to solve

$$\mathbf{Z}\mathbf{P}x - \mathbf{B}\mathbf{V}_d = \mathbf{F}.$$

After eliminating the branch potential unknowns $\mathbf{V}_d$ by multiplying this equation with $\mathbf{P}^T$, we get

$$\mathbf{P}^T \mathbf{Z}\mathbf{P}x = \mathbf{P}^T \mathbf{F}. \tag{5}$$

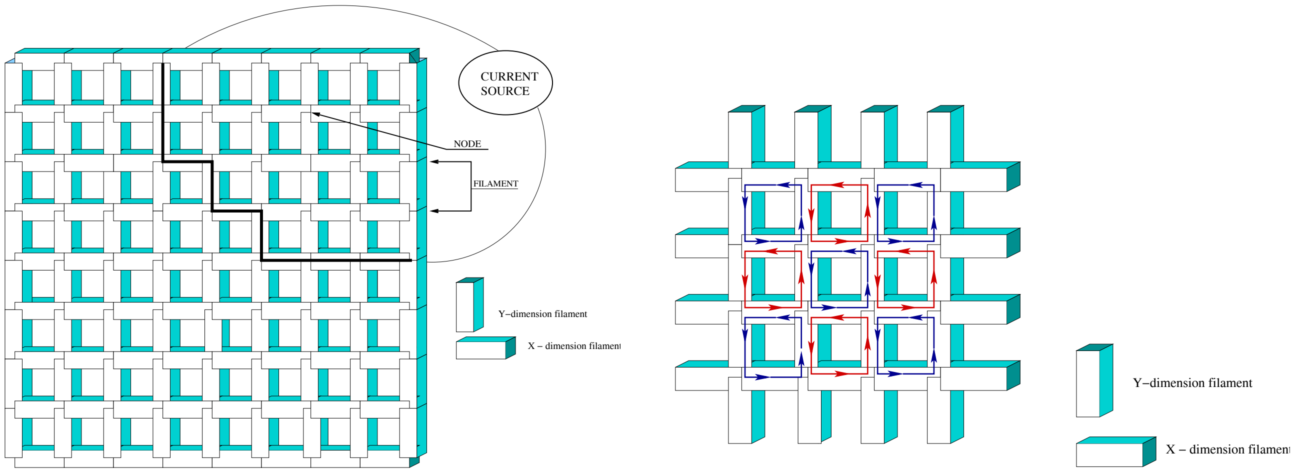

Figure 1: Discretization of a ground plane with a mesh of filaments. Current flowing through the filaments must satisfy Kirchoff's law at each node in the mesh. The bold line indicates a path for current that satisfies boundary conditions. The current is made up of two components: constant current along the bold line shown in the left figure and a linear combination of mesh currents as shown in the partial mesh on the right.

This reduced system can be solved via a suitable iterative scheme such as the GMRES method [14]. Once x is obtained, current is computed as $\mathbf{I} = \mathbf{P}x$. The potential difference across each filament is given by

$$\mathbf{V} = \mathbf{ZI} - \mathbf{F}.$$

The potential difference between two nodes is computed by adding the potential difference across filaments on any path connecting the nodes. This allows computation of impedance between the points where source current is applied.

3.2 Preconditioning the Reduced System

To accelerate the convergence of the iterative method, one must devise robust and effective preconditioning schemes for the system matrix in (5). The task of designing effective preconditioners is made especially challenging due to the unavailability of $\mathbf{L}$. Popular techniques based on incomplete factorizations of the system matrix compute parts of the system matrix selectively, followed by incomplete factorizations. These schemes tend to be expensive and limited in their effectiveness [7].

The success of the solenoidal basis method depends on effective preconditioning of the reduced system (5). A preconditioner is said to be optimal when the condition number of the preconditioned system, computed as the ratio of the largest and the smallest singular values, is bounded by a constant. In such a case, iterative methods may converge to the solution in fixed number of iterations regardless of the frequency ω and mesh width h of the mesh used for discretizing the conductor. The system matrix $\mathbf{P}^T\mathbf{ZP}$ can be analyzed using the observation that the matrices $\mathbf{P}$ and $\mathbf{P}^T$ are equivalent to discrete curl operators. Furthermore, the matrix $\mathbf{P}^T\mathbf{P}$ is equivalent to a discrete Laplace operator in the divergence-free space. We approximate $[\mathbf{P}^T\mathbf{P}]^{-1}$ by the matrix $\tilde{\mathbf{L}}^2$, where $\tilde{\mathbf{L}}$ is defined as follows:

$$\tilde{\mathbf{L}}_{kl} = \frac{\mu}{4\pi}\frac{1}{a_k a_l}\int_{r_k \varepsilon V_k}\int_{r_l \varepsilon V_l}\frac{1}{||\mathbf{r}_k - \mathbf{r}_l||}dV_k dV_l. \tag{6}$$

Here, $\tilde{\mathbf{L}}_{kl}$ gives the mutual inductance between parallel filaments placed at the centers of loop k and l.

The reduced system (5) can be approximated by

$$\mathbf{P}^T\mathbf{ZP} \approx \tilde{\mathbf{L}}^{-1}\left[\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}}\right]\tilde{\mathbf{L}}^{-1}$$

where $\tilde{\mathbf{R}}$ is a diagonal matrix of resistance to mesh currents. The preconditioner for the reduced system (5) is defined as

$$\mathcal{M}^{-1} = \tilde{\mathbf{L}}\left[\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}}\right]^{-1}\tilde{\mathbf{L}}. \tag{7}$$

The matrix $\mathcal{M}^{-1}$ must be multiplied to the reduced system before the iterative method is employed. At each iteration, the preconditioning step consists of the matrix-vector product $z = \mathcal{M}^{-1}r$ which can be computed in the following three steps

$$u = \tilde{\mathbf{L}}r, \qquad v = \left[\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}}\right]^{-1}u, \qquad z = \tilde{\mathbf{L}}v.$$

The matrix-vector products in the first and third steps use approximate hierarchical techniques identical to those used for $\mathbf{L}$. The second step is implemented via an inner iterative solver which is used to solve the system

$$\left[\tilde{\mathbf{R}} + j\omega\tilde{\mathbf{L}}\right]v = u$$

to obtain v.

At very low frequency operation, the preconditioner takes the form

$$\mathcal{M}_{\text{low}}^{-1} = \tilde{\mathbf{L}}\tilde{\mathbf{R}}^{-1}\tilde{\mathbf{L}},$$

in which inverting the diagonal matrix $\tilde{\mathbf{R}}$ is inexpensive. At very high frequency operation, the preconditioner takes the form

$$\mathcal{M}_{\text{high}}^{-1} = -\frac{j}{\omega}\tilde{\mathbf{L}}$$

which is also inexpensive to apply to a vector. In each case, the preconditioning step is relatively cheap since it doesn't involve an inner solve. For intermediate frequencies, however, one must use the preconditioner (7).

There are several advantages of our preconditioning approach. The preconditioning step requires a matrix-vector product which is relatively inexpensive compared to incomplete factorization based preconditioners. The latter involve incomplete factorizations of partially computed $\mathbf{P^T ZP}$ and triangular solves which are expensive, especially on parallel platforms. In addition, experimental evidence suggests that the preconditioner is robust and very effective for the a wide range of frequencies.

3.3 The Algorithm

An outline of the solenoidal basis method to compute $\tilde{\mathbf{Z}}$ is given below.

ALGORITHM 1. *Solenoidal Basis Method.*

1. Generate a uniform mesh for each conductor in the circuit.
2. Compute the solenoidal basis matrix $\mathbf{P}$.
3. For each conductor $l = 1, \ldots, s$,
   (a) Compute the particular solution $\mathbf{I}_p^{(l)}$ for a unit current flow through conductor l and the corresponding induced potential drop vector $\mathbf{F}^{(l)}$.
   (b) Solve the preconditioned system
   $$\mathbf{P}^T\mathbf{Z}\mathbf{P}\mathcal{M}x = \mathbf{P}^T\mathbf{F}^{(l)}, \qquad x^{(l)} = \mathcal{M}x$$
   to determine the mesh current vector $x^{(l)}$ and compute filament current vector
   $$\mathbf{I}^{(l)} = \mathbf{P}x^{(l)} + \mathbf{I}_p^{(l)}.$$
   Use GMRES algorithm with variants of the preconditioner (7). Use approximate hierarchical methods such as FMM or Barnes-Hut to compute matrix-vector products with $\mathbf{L}$ and $\tilde{\mathbf{L}}$ at each iteration.
   (c) For each conductor $k = 1, \ldots, s$, determine the conductor impedance matrix element $\tilde{\mathbf{Z}}_{k,l}$ by computing the potential difference between the two ends of the conductor. This is done by adding the potential drop across all the filaments along a path from one end to the other end of the conductor due to the current $\mathbf{I}^{(l)}$.

An efficient implementation of this algorithm can use a number of optimizations. The matrix $\mathbf{P}$ is never computed explicitly. A matrix-vector product with $\mathbf{P}$ is used to compute filament currents from mesh currents. Since this computation is defined locally, it can be performed by accumulating the contribution of each mesh current to the four filaments that comprise the mesh or loop. Knowledge of the structure of the discretization mesh is sufficient to develop an implementation in which $\mathbf{P}$ is not computed and stored explicitly. Similarly, matrix-vector products with $\mathbf{P}^T$ are used to compute mesh currents from filament currents. These products can also be computed without explicitly computing $\mathbf{P}^T$. This approach leads to significant saving in storage without increase in computation.

The most expensive operations in the algorithm are matrix-vector computations with the system matrix $\mathbf{P}^T\mathbf{Z}\mathbf{P}$ and preconditioning steps involving matrix-vector products with $\tilde{\mathbf{L}}$. A matrix-vector product with $\mathbf{P}^T\mathbf{Z}\mathbf{P}$ is computed as a sequence of three products:

$$u = \mathbf{P}x, \qquad v = [\mathbf{R} + j\omega\mathbf{L}]\,u, \qquad y = \mathbf{P}^T v.$$

Among these, the product with the dense matrix $\mathbf{L}$ is by far the most time-consuming computation. Since the matrix $\tilde{\mathbf{L}}$ used in the preconditioning step is similar to $\mathbf{L}$, improving the computational complexity of this matrix-vector product can reduce the overall computation time significantly. There are several approximate hierarchical methods to compute products with such matrices in which accuracy is traded for a reduction in computational complexity.

Accurate matrix-vector products with the dense matrices $\mathbf{L}$ and $\tilde{\mathbf{L}}$ require $O(n^2)$ operations where n is the size of these matrices. A number of techniques have been developed to exploit the rapid decay of the kernel in (2) and (6) with distance to compute approximate matrix-vector products in $O(n\log n)$ or $O(n)$ operations. These include hierarchical techniques such as Barnes-Hut [1] and Fast Multipole Method (FMM) [4, 12] which use a truncated series approximation of filament currents within a localized region to estimate impact on well-separated sets of filaments. The method of Barnes and Hut relies only on filament-cluster interactions to achieve an $O(n\log n)$ computational bound for uniform filament distributions. The Fast Multipole Method uses both filament-cluster and filament-filament interactions to achieve an $O(n)$ bound for uniform distributions. In each case, reduction in computational complexity is associated with decrease in accuracy of the matrix-vector product.

We use a variant of the Barnes-Hut algorithm developed specifically for the inductance extraction problem. A hierarchical quadtree is used to partition the filaments in the mesh. Internal nodes of the tree represent hypothetical filaments carrying current equal to the sum of currents in that subtree. The coordinates of these hypothetical filaments are computed as a weighted sum of coordinates of the subtree filaments in which the weights correspond to the magnitude of current flowing in those filaments. This strategy exploits the fact that the combined inductance effect of a cluster of filaments can be approximated by the inductance effect of a single filament placed at the "center of current" of that group. This approximation is used to compute inductance on a filament which is well-separated from the cluster. This is analogous to the center of mass concept in Barnes-Hut algorithm. The leaf nodes of the quadtree are filaments of the mesh. Inductance on a leaf node is computed by traversing the tree in a top down manner. A subtree is traversed only if the corresponding box is not well-separated from the leaf node. Well-separatedness is established by using a distance metric to determines if the leaf node is sufficiently "far off" such that the error in the associated truncated expansion is below a threshold. The inductance due to filaments in a well-separated subtree is computed by direct interaction with the corresponding hypothetical filament. Remaining interactions consist of leaf-leaf interactions which consist of mutual inductance computation between pairs of filaments. Care should be taken to ensure that the algorithm doesn't check the well-separatedness criteria for the box containing the leaf node itself.

4. EXPERIMENTS

4.1 Ground Plane

We consider the problem of computing impedance of a 1cm$\times$1cm ground plane with unit inflow current from the left edge and unit outflow current at the right edge (see Fig. 1). This benchmark problem allows discretization by uniform two-dimensional meshes with varying mesh width h. This problem exhibits the dependence of the condition number of the system matrices on h via the growth in iterations of the unpreconditioned GMRES algorithm. Such convergence behavior is ideal for testing the effectiveness of the preconditioning approach for different levels of refinement as well as operating frequencies.

Table 1 shows the number of iterations needed by the preconditioned GMRES to solve the linear system. The width of each filament is one-third of its length, and thickness is 2^{-13}cm. A tolerance of 10^{-3} was specified on the relative residual norm. The table shows that the rate of convergence is independent of the frequency ω and the mesh width h indicating that the preconditioner is optimal for the benchmark problem .

Table 1: Ground plane: Iterations for convergence of preconditioned GMRES.

Mesh Size	Filament Length (cm)	Frequency (in GHz)			
		1	10	100	1000
33×33	2^{-5}	6	5	5	5
65×65	2^{-6}	6	6	5	5
129×129	2^{-7}	8	7	7	6
257×257	2^{-8}	11	9	8	8

4.2 Spiral Inductor

This is a more complicated problem consisting of a conductor in the shape of a coil (see Fig. 2) which is discretized by a two-dimensional mesh. The coil is contained within a square region of size 1 cm$\times$1 cm. Table 2 shows the number of iterations needed by the preconditioned GMRES to solve the linear system with a tolerance of 10^{-3} on the relative residual norm. Filament width and thickness are identical to the previous experiment. The number of iterations required by the solver are nearly independent of the mesh width for the range of frequencies considered here.

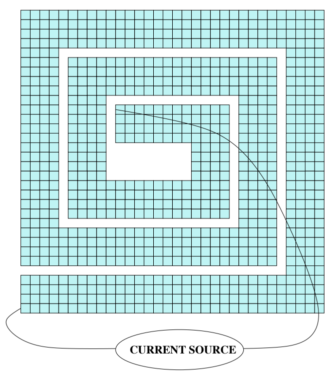

Figure 2: Spiral inductor.

Table 2: Spiral inductor: Iterations for convergence of preconditioned GMRES.

Mesh Size	Filament Length (cm)	Frequency (in GHz)			
		1	10	100	1000
33×33	2^{-5}	7	6	6	6
65×65	2^{-6}	8	7	7	7
129×129	2^{-7}	10	9	9	9
257×257	2^{-8}	16	12	11	11

4.3 Wire Over Ground Plane

We consider a 3D problem of determining impedance of a horizontal wire of length 0.25cm suspended at a distance of 0.1cm over a 1cm$\times$1cm ground plane (see Fig. 3). The wire is discretized along its length. The ground plane is discretized by a two-dimensional mesh similar to the one in the ground plane problem discussed earlier. Filament width and thickness are identical to previous experiments. Table 3 shows the number of iterations needed by the preconditioned GMRES. Solver parameters were identical to earlier experiments. These experiments also indicate that the number of iterations required by the solver are nearly independent of the mesh width for the range of frequencies considered here.

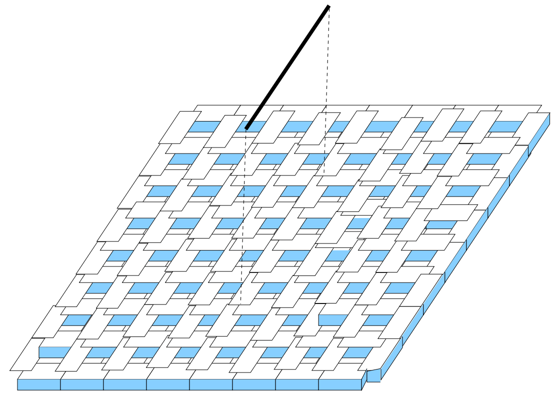

Figure 3: Wire over ground plane.

Table 3: Wire over ground plane: Iterations for convergence of preconditioned GMRES.

Mesh Size	Filament Length	Frequency (in GHz)			
		1	10	100	1000
33×33	2^{-5}	5	4	4	4
65×65	2^{-6}	6	5	5	5
129×129	2^{-7}	8	6	6	6
257×257	2^{-8}	12	8	8	7

4.4 Comparison with FastHenry

The performance of the solenoidal basis method was compared to FastHenry on two problems: the 2D ground plane problem and the 3D problem of a wire over the ground plane. FastHenry uses mesh currents to generate a reduced system which is then solved by the preconditioned GMRES method. Fast Multipole Method is used to compute matrix-vector products with dense matrices. Preconditioners are derived from incomplete factorizations of approximations to the reduced system. These approximations are computed by various techniques to sparsify the inductance matrix $\mathbf{L}$. The software allows preconditioners such as CUBE and SHELL, in which the off-diagonal non-zeros in each column of $\mathbf{L}$ are restricted to those resulting from mutual inductance between filaments in the same box at a specific level in the quad tree and mutual inductance between filaments within a specific radius, respectively. The DIAG preconditioner corresponds to the case when all mutual inductances are ignored.

For the solenoidal basis method, the Barnes-Hut variant described in Section 3 was used to compute products with the system matrix $\mathbf{L}$ and preconditioner $\tilde{\mathbf{L}}$ directly without explicitly computing these matrices. The resulting implementation is a matrix-free code in which neither the system matrix nor the preconditioner matrix is ever computed. This reduces the storage requirement considerably, thereby allowing larger problems to be solved.

Ground Plane

Table 4 shows the number of iterations needed by preconditioned GMRES for 10GHz frequency. A tolerance of 10^{-3} was specified on the relative residual norm of both the methods. FastHenry was allowed to use default values for all other parameters. In these experiments, the inductance computed by the solenoidal basis method was within 3% of that obtained by FastHenry.

FastHenry requires significant amount of memory to construct the preconditioner matrix and compute its LU factorization. The entries marked "–" indicate the inability of FastHenry to solve the

Table 4: Ground plane problem.

Mesh	FastHenry-DIAG			FastHenry-CUBE			Solenoidal Method		
	Iter.	Time (s)	Mem. (MB)	Iter.	Time (s)	Mem. (MB)	Iter.	Time (s)	Mem. (MB)
33×33	13	2.04	10	13	2.36	10	5	2.03	1
65×65	16	12.70	42	17	16.59	42	6	11.91	5
129×129	21	95.30	177	19	142.42	177	7	67.61	17
257×257	26	835.66	734	28	1364.19	734	9	409.21	69
513×513	–	–	–	–	–	–	14	2924.58	298

Table 5: Wire over ground plane problem.

Mesh	FastHenry-DIAG			FastHenry-CUBE			Solenoidal Method		
	Iter.	Time (s)	Mem. (MB)	Iter.	Time (s)	Mem. (MB)	Iter.	Time (s)	Mem. (MB)
33×33	13	2.03	10	11	2.20	10	4	1.43	1
65×65	13	11.60	42	14	16.01	42	5	9.20	4
129×129	13	79.06	178	12	123.96	178	6	55.12	15
257×257	3	718.93	735	3	2732.73	735	8	351.19	61
513×513	–	–	–	–	–	–	12	2426.73	260

problem within the available system memory. These experiments were conducted on a 1.5GHz Pentium Dell Workstation with 1GB of memory. Furthermore, a growth in the number of iterations with mesh size indicates a sub-optimal preconditioning scheme which contributes an additional factor towards the increase in cost of solving these systems as the mesh size is increased. In contrast, the solenoidal basis method is able to solve the systems in almost fixed number of iterations.

Wire Over Ground Plane

Table 5 shows the number of iterations needed by preconditioned GMRES for 10GHz frequency. Again, a stopping tolerance of 10^{-3} was used for the relative residual norm of both the methods. FastHenry was allowed to use default values for all other parameters. These results also demonstrate the comparative advantage of the solenoidal basis method. The performance is similar to the ground plane problem.

Discussion

The modest performance of the preconditioners in FastHenry come with a significant cost of computing the preconditioners themselves as well as storing them. While these storage requirements can be reduced by computing incomplete factorizations, often this results in a weak preconditioner. The slower convergence rates associated with ineffective preconditioning may lead to overall higher computational cost. The comparative advantage of the solenoidal method is expected to grow with larger mesh sizes.

5. CONCLUSIONS

This paper presents a solenoidal basis method for inductance extraction of VLSI circuits. The proposed approach solves a linear system of equations to compute the mutual inductance effect on a filament mesh using eddy currents defined on the mesh loops. The resulting reduced system is solved iteratively by the preconditioned GMRES method. The preconditioner suggested for the system matrix exhibits convergence in a relatively few iterations irrespective of the mesh refinement and operational frequency. Experimental results indicate that the algorithm has several advantages over FastHenry. Since our algorithms is kernel independent, it can be applied to multi-layer dielectric by using multi-layer Green's function.

6. REFERENCES

[1] J. Barnes and P. Hut. A hierarchical O(n log n) force calculation algorithm. *Nature*, Vol. 324, 1986.

[2] L. Daniel, A. Sangiovanni-Vincentelli, and J. White. Techniques for including dielectrics when extracting passive low-order models of high speed interconnect. ICCAD 01.

[3] K. Gala, et. al. On-chip inductance modeling and analysis. DAC 2000.

[4] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. The MIT Press, Cambridge, Massachusetts, 1988.

[5] Z. He, M. Celik and L. Pileggi. SPIE: Sparse partial inductance extraction. DAC 1997.

[6] M. Kamon. *Fast Parasitic Extraction and Simulation of Three-dimensional Interconnect via Quasistatic Analysis*. PhD thesis, EE & CS Dept., MIT, 1998.

[7] M. Kamon and J. R. Philips. 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, 1994.

[8] M. Kamon, M. J. Tsuk, and J. White. FASTHENRY: A multipole-accelerated 3D inductance extraction program. *IEEE Trans. MTT*, Vol. 42:1750–1758, 1994.

[9] B. Krauter and S. Mehrotra. Layout based frequency dependent inductance and resistance extraction for on-chip interconnect timing analysis. DAC 1998.

[10] B. Krauter and L. T. Pileggi. Generating sparse partial inductance matrices with guaranteed stability. ICCAD 95.

[11] M. Liu, T. Yu and W. W.-M. Dai. Fast 3D inductance extraction in lossy multi-layer substrate. ICCAD 01.

[12] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *Journal of Computational Physics*, Vol. 60:187–207, 1985.

[13] A. E. Ruehli. Inductance calculation in a complex integrated circuit environment. *IBM J. Res. Dev.*, Sept. 1972.

[14] Y. Saad. Iterative methods for sparse linear systems. PWS publishing company, 1996.

[15] K. L. Shepard and Y. Zhang. Return-limited inductances: A practical approach to on-chip inductance extraction. *IEEE Trans. CAD*, 19(4):425–436, 2000.