Fast On-chip Inductance Simulation using a Precorrected-FFT Method

Haitian Hu, ECE Department, University of Minnesota, Minneapolis, MN 55455
David T. Blaauw, EECS Department, University of Michigan, Ann Arbor, MI 48104
Vladimir Zolotov, Kaushik Gala, Min Zhao, Rajendran Panda, Motorola, Inc., Austin, TX 78729
Sachin S. Sapatnekar, ECE Department, University of Minnesota, Minneapolis, MN 55455

Abstract

In this paper, a precorrected-FFT approach for fast and highly accurate simulation of circuits with on-chip inductance is proposed. This work is motivated by the fact that circuit analysis and optimization methods based on the partial element equivalent circuit (PEEC) model require the solution of a subproblem in which a dense inductance matrix must be multiplied by a given vector, an operation with a high computational cost. Unlike traditional inductance extraction approaches, the precorrected-FFT method does not attempt to compute the inductance matrix explicitly, but assumes the entries in the given vector to be the fictitious currents in inductors and enables the accurate and quick computation of this matrix-vector product by exploiting the properties of the inductance calculation procedure. The effects of all of the inductors are implicitly considered in the calculation: faraway inductor effects are captured by representing the conductor currents as point currents on a grid, while nearby inductive interactions are modeled through direct calculation. The grid representation enables the use of the discrete Fast Fourier Transform (FFT) for fast magnetic vector potential calculation. The precorrected-FFT method has been applied to accurately simulate large industrial circuits with up to 121,000 inductors and over 7 billion mutual inductive couplings in about 20 minutes. Techniques for trading off CPU time with accuracy using different approximation orders and grid constructions are also illustrated. Comparisons with a block diagonal sparsification method are used to illustrate the accuracy and effectiveness of this method. In terms of accuracy, memory and speed, it is shown that the precorrected-FFT method is an excellent approach for simulating on-chip inductance in a large circuit.

1. Introduction

The fast and accurate simulation of circuits with on-chip inductance is a growing problem, and future trends show that the relative contribution of inductive effects on circuit behavior will continue to increase as technologies shrink further and low-k dielectrics are used to diminish capacitive effects. Inductive effects have become important in determining power supply integrity, timing and noise analysis, especially for global clock networks, signal buses and supply grids for high-performance microprocessors.

One of the major problems in determining inductance has been associated with the fact that wire inductances are defined over current loops, and that the current loops are dependent on the circuit context of the switching wires. This leads to a chicken-and-egg problem where the inductance cannot be extracted until the current return

paths are known, which, in turn, can only be determined after some knowledge of the inductance. Fortunately, an elegant way around this was found using the PEEC model [1], which does not require the current return paths to be predetermined. The PEEC approach introduces the concept of partial inductance of a wire or a wire segment, corresponding to a return path at infinity. The partial self-inductance is defined as the inductance of a wire segment that is in its own magnetic field, while the partial mutual inductance is defined between two wire segments, each of which is in the magnetic field produced by the current in the other. For two wire segments k and m, the partial mutual inductance is given by

$$M_{km} = \frac{1}{I_m a_k}\left(\int_{a_k}\int_{l_k} \vec{A}_{km} \bullet d\vec{l}_k\, da_k\right) \qquad (1)$$

where a_k is the cross section area of segment k, $\vec{l}_k$ is the length vector along segment k and $\vec{A}_{km}$ is the magnetic vector potential along segment k due to the current I_m in segment m, given by:

$$\vec{A}_{km} = \frac{m_0}{4pa_m}\left(\int_{a_m}\int_{l_m} \frac{I_m}{r_{km}}\, d\vec{l}_m\, da_m\right) \qquad (2)$$

Here, r_{km} is the distance between any two points on segment k and m. Simplified closed form formulae for partial self and mutual inductances of typical wire topologies that are useful in current-day integrated circuit environments have been calculated in [2].

One drawback of using the PEEC method directly is that it requires the calculation of nonzero mutual inductances between *every* pair of nonperpendicular wire segments in a layout. This results in a dense inductance matrix that causes a high computational overhead for a simulator. Although many entries in this matrix are small and have negligible effects, zeroing them out may cause the resulting inductance matrix to lose its desirable positive definiteness property [3], which is a necessary condition for the matrix to represent a physically realizable inductor system. Consequently, several efforts have been made to develop algorithms to sparsify the dense inductance matrix while maintaining this property.

The shift and truncate method [4] finds a sparse matrix approximation by assuming that the current return of each wire segment is not at infinity, but is distributed on a shell of finite radius R_0, which must be constant for the analysis of the entire chip. Under this assumption, the inductance formula (1) is altered by subtracting a factor, which is inversely proportional to R_0, from the partial inductance, and setting the value to zero if the result is negative. Although this method succeeds in removing faraway inductive interactions from consideration and maintains the positive definiteness of the matrix, the subtractive factor can cause errors in calculating nearby inductive interactions if the radius is not large enough. Moreover, finding a reliable global value of R_0 is a nontrivial problem: a high accuracy demands a large R_0, which, in turn, can result in low sparsification. Although efforts in the direction of determining R_0 have been made in [3], this is not a solved problem.

An alternative approach that uses return-limited inductances [5] is a shape-based method for sparsifying the inductance matrix using "halo rules". This method is good as a first order approximation. Another approach [6]

introduces a block diagonal method that is a heuristic sparsification technique based on a simple partition of the circuit topology. This approach also maintains the positive definiteness of the matrix, but neglects mutual inductances between partitions. In [7], the circuit element *K*, defined as the inverse of the traditional PEEC inductance matrix, was introduced as an alternative element for representing an inductance system. The *K* matrix is shown in [8] to have better properties than the inductance matrix in that it is symmetric, positive semidefinite and diagonally dominant. Similar to a capacitance matrix, the *K* matrix can be easily sparsified and can obtain a higher accuracy than an inductance matrix for the same sparsification. However, as in the case of the shift and truncate method, the algorithm provided in [7] uses a fixed window size within which local interactions are modeled. Moreover, it requires that existing simulators are extended such that they handle the new circuit element *K*.

The shortcomings common to all of these methods are twofold. First, it is difficult to determine how to set the radius or partition size outside which couplings may be ignored. The principal problem is that it is difficult to definitively demarcate a region such that an aggressor wire segment outside this local interaction region is too weak to have a significant effect on a victim wire segment within it. Some of the heuristics proposed to overcome this limitation entail simulation, which may require large CPU times for large circuits. Second, although the individual couplings that are ignored may be small, it is difficult to determine the cumulative effect of ignoring a larger set of such couplings without detailed knowledge of the current distributions.

FastHenry [9] is a multipole-accelerated method for inductance extraction. However, it works in frequency domain and ignores the effects of capacitance on the estimation of current return path. In order to obtain the time domain simulation, an accurate compact model has to be constructed, which is not an easy procedure.

Recently, a number of methods for circuit and layout analysis and optimization for on-chip inductance have been proposed [10, 11, 12, 13, 14]. However these methods have typically used either RL inductance formulations or analytical models, which have limited accuracy for large circuit structures.

In this paper, we propose a precorrected-FFT method that, instead of entirely dropping long-range couplings, approximates these couplings, thereby overcoming the above two shortcomings. The main idea of this method is to represent the long-range part of the vector potential by point currents on a uniform grid and nearby interactions by direct calculations. The grid representation permits the use of the discrete Fast Fourier Transform (FFT) for fast potential calculations. Because of the decoupling of the short and long-range parts of the potentials, this algorithm can be applied to problems with irregular discretizations. Other techniques similar to the precorrected-FFT method, such as the adaptive integral method (AIM) [15], have been proposed; the advantages of the former over the latter are discussed in [16].

The idea of using a precorrected-FFT approach for accelerated electromagnetic calculations has been used in the past to accelerate the coulomb potential calculation for solving electromagnetic boundary integral equations for three-dimensional geometries. During the capacitance extraction technique introduced in [17, 18], each iteration of the algorithm computes the product of a dense matrix with a charge vector to calculate electrical potential on each

conductor. The basic precorrected-FFT method presented in this paper is inspired by the method in [17, 18] for capacitance extraction, which also demonstrates that for many realistic structures, the precorrected-FFT method is faster and uses less memory compared with the multipole-accelerated method. In our work, the precorrected-FFT method is adapted to the specific requirements of simulation of on-chip inductance. Unlike [17, 18], we do not focus on extracting a matrix describing the parasitics (namely, the inductance matrix M in our case), but rather, directly consider how the inductance matrix is used in fast simulation algorithms. As described in Section 2, many simulators do not require M to be explicitly determined, but instead, require the computation of the product of M with a vector I. The approach developed in this paper accelerates the procedure that is used to directly determine the $M \times I$ product without explicitly finding M. It proceeds by first assuming that the entries in I are fictitious currents in inductors and then transforming the calculation of the $M \times I$ product to the calculation of the integration of the magnetic vector potential $\vec{A}$ over the volume of the inductors, as depicted in equation (5) in Section 2. The long-range magnetic interactions are represented by point currents on a discretized grid, while short-range contributions to the $M \times I$ product are directly calculated. Several considerations are incorporated to make the algorithm efficient and applicable to large circuits and complex layouts. First, since mutually perpendicular segments do not have any inductive interactions, it is possible to apply the precorrected-FFT method to wire segments in the two perpendicular directions separately. This simplification is applicable to inductance systems and not to capacitance systems. Second, since IC chips typically have much larger sizes in the two planar dimensions than in the third (i.e., they tend to be "flat"), we show that a two-dimensional grid may be used instead of a three-dimensional grid.

A comprehensive PEEC model, as described in [6], is used in this paper. We demonstrate the application of the precorrected-FFT method within a simulation flow based on PRIMA [19], on circuits of up to 121,000 inductors and nearly 7 billion mutual inductive couplings. These experiments demonstrate the speed, memory consumption and accuracy of the precorrected-FFT method as compared to the block diagonal method [6]. We also illustrate how tradeoffs may be made in order to obtain higher speed implementations with a small reduction in accuracy.

The remainder of this paper is organized as follows. In Section 2, the motivation for calculating the $M \times I$ product is presented, and a description of the problem formulation is provided. This is followed by a detailed description of the precorrected-FFT algorithm as applied to the inductance problems in Section 3. Experimental results and an analysis of the relation between the accuracy and speed are performed in Section 4, including a comparison with the block diagonal method in terms of speed, memory cost and accuracy. Concluding remarks are presented in Section 5.

2. Motivation and problem formulation

It is well known that a direct application of the PEEC model results in dense inductance matrices. The partial inductances of an n-wire segment system can be written as an $n \times n$ symmetric, positive semidefinite matrix $M \in$

R^{nxn}. Once this inductance matrix has been calculated, it may be incorporated into a circuit model that captures the interactions of R, L, C and active elements in the circuit. If the circuit is linear, it can be solved efficiently using model order reduction techniques such as PRIMA or using a SPICE-like transient simulation flow.

2.1. Model order reduction techniques

To understand how the inductance matrix is used within a model order reduction method, let us use PRIMA as a representative model order reduction engine (the application to AWE is very similar) and consider a circuit that is represented by the modified nodal equation

$$(G + s\,C)\,X = B \qquad (3a)$$

where $(G+sC)$ is the admittance matrix, G is a conductance matrix, C is a matrix that represents the capacitive and inductive elements, X is a vector of unknown node voltages and unknown currents of inductors and voltage sources, and B is a vector of independent time-varying voltage and current sources. Specifically,

$$G = \begin{bmatrix} N & E \\ -E^T & 0 \end{bmatrix} \quad C = \begin{bmatrix} Q & 0 \\ 0 & M \end{bmatrix} \quad x = \begin{bmatrix} v \\ i \end{bmatrix} \qquad (3b)$$

where N, Q and M are, respectively, the submatrices representing conductances, capacitances and inductances in the network. E consists of ones, minus ones and zeros, and N, Q, and M must be symmetric and positive definite to guarantee passivity. The submatrix of capacitances, Q, is typically sparse, while the submatrix M of inductances is dense.

Our objective is to reduce the large conductance and capacitance matrices into smaller reduced matrices, so that the reduced linear system may be simulated exactly in conjunction with nonlinear transistor models in a circuit simulator. The vectors of moments, m_i, of X can be calculated by solving the equations

$$G\,m_0 = b \qquad (4a)$$

$$G\,m_i = -\,C\,m_{i-1} \qquad (4b)$$

These are orthonormalized in each step to obtain an orthonormal X matrix. Note that the right hand side of Equation (4b) involves the multiplication of C by a constant vector. The matrix for the reduced order system can be calculated as follows:

$$\tilde{G} = X^T G X \qquad \tilde{C} = X^T C X$$

Using these reduced matrices and the transistor models, a netlist for the reduced order system may be constructed and simulated using SPICE.

2.2. SPICE-like transient simulation flow

The time-domain modified nodal equation is given by:

$$GX + C\dot{X} = B$$

where the definition and formation of G, C, X and B are the same as in (3). Such equations can be solved using the backward-Euler method under a given time step, h, as follows:

$$GX_{n+1} + C\frac{X_{n+1} - X_n}{h} = B$$

Rearranging the above equation, we obtain:

$$(G + \frac{C}{h})X_{n+1} = B + \frac{C}{h}X_n \text{ or } PX_{n+1} = q$$

Where $P = (G + C/h)$ and $q = (B + C/h\ X_n)$. Given the values of X at the n^{th} time step, we can solve the above equation for X at $(n+1)^{th}$ time step. This equation can be solved by direct methods such as LU factorization, or using an iterative solver such as GMRES [20]. For very large circuits and a dense M matrix in C, LU factorization of $G+C/h$ matrix could become computationally expensive, and therefore the use of iterative methods becomes attractive. The GMRES procedure requires the evaluation of the product of P by a vector, and therefore it is seen that this procedure, as well as the procedure of determining q, require the multiplication of the C matrix by a constant vector.

2.3. Problem formulation

Regardless of whether model order reduction techniques or transient simulations using an iterative solver are employed, we face the problem of the multiplication of C matrix with a constant vector. The product of M with a known vector $I \in R^{nx1}$ for these wire segments can be written as:

$$M \times I = \begin{bmatrix} M_{11} & M_{12} & \cdots & \cdots & \cdots & M_{1n} \\ M_{21} & M_{22} & \cdots & \cdots & \cdots & M_{2n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & M_{km} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ M_{n1} & M_{n2} & \cdots & \cdots & \cdots & M_{nn} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_m \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} \sum_{m=1}^{n} (\frac{1}{a_1} \int \vec{A}_{1m} \bullet d\vec{l}_1 da_1) \\ \sum_{m=1}^{n} (\frac{1}{a_2} \int \vec{A}_{2m} \bullet d\vec{l}_2 da_2) \\ \vdots \\ \sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k) \\ \vdots \\ \sum_{m=1}^{n} (\frac{1}{a_n} \int \vec{A}_{nm} \bullet d\vec{l}_n da_n) \end{bmatrix} \quad (5)$$

Here, we assume that I_m is the fictitious current in wire segment m and $\vec{A}_{km}$ is the magnetic vector potential on wire segment k due to I_m. $\vec{A}_{km}$ is in the same direction as that of I_m and can be determined by the expressions in (2). Each entry M_{km} in matrix M is the partial inductance between wire segment k and m, given by:

$$M_{km} = \frac{m_0}{4pa_k a_m} \int_{a_k} \int_{a_m} \int_{l_k} \int_{l_m} \frac{d\vec{l}_k \bullet d\vec{l}_m}{r_{km}} da_k\, da_m \quad (6)$$

where l_i and a_i (i=k or m) are the length and cross section area of wire segment i. The *kth* entry in the $M \times I$ product, corresponding to the victim wire segment k, is $\sum_{m=1}^{n} M_{km} I_m = \sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k)$. It is the summation of the integration of the magnetic vector potential over wire segment k caused by the current in each aggressor wire segment. The expression (6) can be calculated using approximate formulae available in [2] or using accurate closed form formulae provided in [21].

If the dense inductance matrix M is used, the computational cost for the matrix-vector product is very high: for a system with n variables, this is $O(n^2)$. The larger the circuit, the larger is the number of moments and ports, and the heavier is the overhead of calculating the dense matrix-vector product. Therefore, methods for sparsifying the M matrix have been widely understood as being vital to solving systems with inductances in an efficient manner.

On closer examination, however, we observe that in order to solve the circuit, it is not the dense inductance submatrix M that needs to be determined, but rather, the product of M with a given vector. This is the motivation for this work, and we present a technique that efficiently finds the product of M with a given vector using the precorrected-FFT approach that accelerates the computation of this matrix-vector product.

Therefore, the proposed method is general in that it can be applied whenever the circuit analyzer relies on the computation of the product of the inductance matrix with a given vector, such as PRIMA and SPICE-like transient analysis in the case where an iterative method is used for the equation solution, although it is not especially useful for an LU-factorization method since the latter requires the elements of the M matrix to be listed explicitly. In this work, we use PRIMA as the simulation engine to test the results of the algorithm.

3. Precorrected-FFT method

The precorrected-FFT method presented here provides an efficient method for estimating the dense $M \times I$ matrix-vector product accurately, and is based on dividing the region under analysis into a grid. In the description of this algorithm, we will begin by using a three-dimensional grid, although we will show in the next section that in practice, a two-dimensional grid can also work well in an integrated circuit environment.

Consider the three-dimensional topology of wires that represents the circuit under consideration. After the wires have been cut into wire segments to be represented using the PEEC model, the circuit can be subdivided into a $k \times l \times m$ array of cells, with each cell containing a set of wire segments. The contribution to the values of $\sum_{m=1}^{n} (\frac{1}{a_k} \int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ of wire segments within a cell under consideration (which we will call the "victim cell") that is caused by wires in other cells (referred to as "aggressor cells") can be classified into two categories: long-range interactions and short-range interactions. The central idea of the precorrected-FFT approach is to represent the current distribution in wire segments in the aggressor cell by using a small number of point currents on the grid

that can accurately approximate the vector potential for faraway victim cells. After this, the potential at grid points caused by the grid currents is found by a discrete convolution that can be easily performed using the FFT. Figure 1 shows a schematic diagram of a multiconductor system subdivided into a grid of 3×3×1 cells. The current distributions of wires in each cell are represented by a 2×2×2 grid of point currents, using an approach that will be described later.

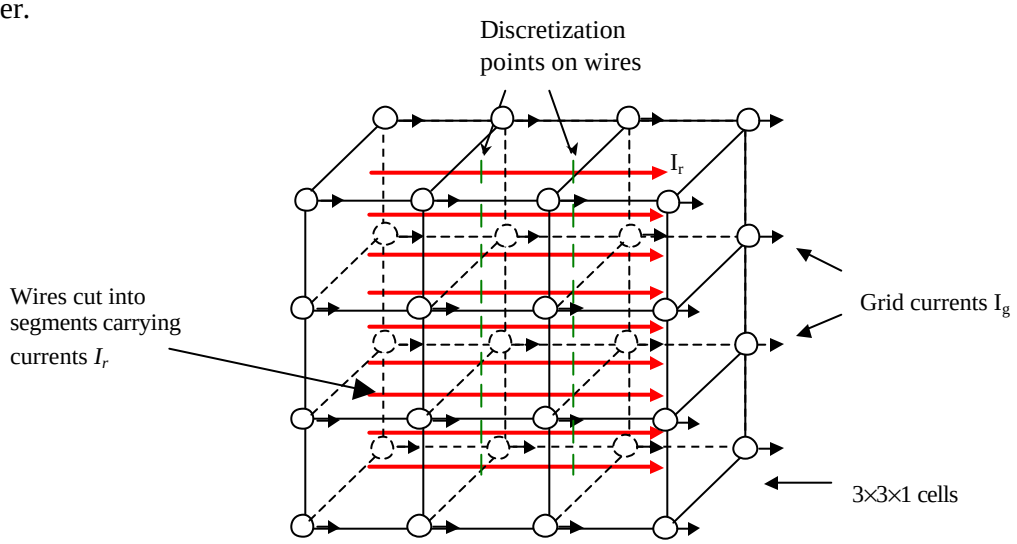


Figure 1: A multiconductor system discretized into wire segments and subdivided into a 3×3×1 cell array with superimposed 2×2×2 grid current representation for each cell. I_g and I_r are currents on grid points and real conductors respectively.

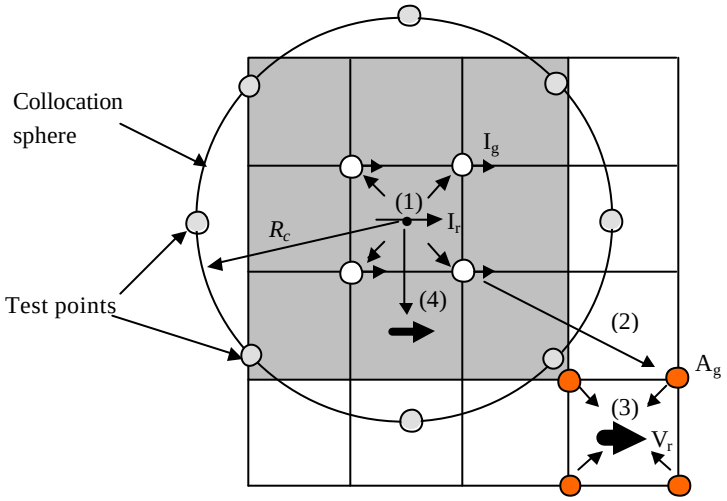


Figure 2: Four steps in precorrected-FFT algorithm. (1) Projection to grid points (2) FFT computation (3) Interpolation within the grid points and (4) Precorrection for accurate computation of nearby interactions. Here, I_g and I_r represent the currents on the grid points and on the real conductors, respectively; A_g and V_r are magnetic vector potential on the grid points, and the values of $\sum_{m=1}^{n}(\frac{1}{a_k}\int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ of real conductors, respectively; R_c is the radius of the collocation sphere, to be defined in section 3.1.

There are four steps in the precorrected-FFT approach to calculate the product of M and I, as illustrated in Figure 2:

1. **Projection:** The currents carried by the wire segments that lie in each cell are projected onto a uniform grid of point currents in the same direction as the currents in the wires. Here, the grid is only required to have a constant grid spacing in each dimension, so that for a three-dimensional grid, the grid spacing can be different in each of the three perpendicular directions. The boundary condition that is maintained during projection is that the vector potentials at a set of test points on a *collocation sphere* surrounding the cell should match the vector potentials due to the actual wires.
2. **FFT:** A multi-dimensional FFT computation is carried out to calculate the grid potentials at the "victim" grid points caused by these "aggressor" grid currents. This computation proceeds by automatically considering all pairs of aggressor-victim combinations within the grid.
3. **Interpolation:** The grid potentials, calculated by the FFT computation, are interpolated onto wire segments in each "victim" cell.
4. **Precorrection:** The projection of wire segments to the uniform grid in step 1 inherently introduces errors into the computation. While these errors are minimal for faraway grid points, they may be more serious in modeling interactions between nearby grid cells. Therefore, the precorrection step directly computes nearby inductive interactions accurately, and "precorrects" to remove the significant errors that could have been introduced as a result of projection.

A detailed description of the four steps is provided in the following subsections.

3.1 Projection

The first step in the precorrected-FFT algorithm is projection, which constructs the grid projection operator W. Using W, the long-range part of the magnetic vector potential due to the current distribution in a given cell can be represented by a small number of currents lying on grid points throughout the volume of the cell. In other words, the current distribution in wire segments can be replaced by a set of grid point currents that are used to calculate the long-range part of the magnetic vector potential. An example of the top view of such a grid representation is shown in Figure 1, where the current distribution in each cell is represented by a $2\times2\times2$ array of grid currents. Since the grid currents are only a substitution for the current distribution in wire segments, the grid can be coarser or finer than the actual problem discretization.

The scheme for representing the current distribution in a cell by a set of grid currents throughout the cell can be illustrated using the first uniqueness theorem in electromagnetic fields [22]. Suppose the current (charge) distribution is contained within some small volume S_0 with radius R_0, as shown in Figure 3, and we are interested in finding the induced magnetic field (electric field) outside of region contained within a surface S. In order to find the magnetic field of a given stationary current distribution (electric field of a given stationary charge distribution) we solve Laplace's equation:

$$\nabla^2 \vec{A} = 0 \quad \text{(For electric field, it is } \nabla^2 V = 0\text{)} \qquad (7)$$

with the boundary condition V_s, which is the known potential distribution on the boundary surface S. The first uniqueness theorem is related to the solution of Laplace's equation, and can be stated as follows:

First uniqueness theorem: The solution of Laplace's equation in some region is uniquely determined if the value of the potential is a specified function on all boundaries of the region.

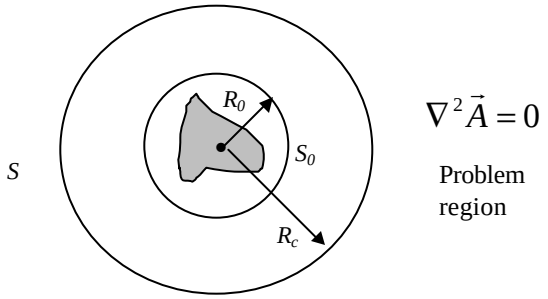


Figure 3: Problem region of Laplace's equation and uniqueness theorem.

This theorem tells us that in order to solve the Laplace's equation, it is not necessary to know the detailed distribution of current sources (charge sources), but that it is sufficient to know the potential distribution on the boundary surface S of the problem region. This suggests a scheme where one current distribution can be replaced by another current distribution provided the two distributions result in the same potential on the boundary surface of the solution of Laplace's equation. For convenience of calculation, we choose the boundary surface as a sphere surface, called the collocation sphere as shown in Figure 2, and point currents lying on a grid as a current distribution that substitutes the original one.

The radius of the current distribution region R_0 is a little larger than the cell size and the small volume S_0 contains the cell. Suppose there are p grid points on each edge of a cell and a $p \times p \times p$ grid of currents is used to represent m currents in wire segments in cell k. A set of N_t test points is chosen on a collocation sphere that has radius $R_c > R_0$, and whose center is coincident with the center of cell k. The problem region is outside of the collocation sphere. Then the potentials on these N_t test points due to the grid currents are forced to match those induced by the current distribution in wire segments by solving the linear equation:

$$P^{gt} I_g(k) = P^{rt} I_r(k) \quad (8)$$

where $I_g(k) \in R^{p^3 \times 1}$ and $I_r(k) \in R^{m \times 1}$ are, respectively, the grid current vector and current vector for wire segments in cell k. $P^{gt} \in R^{N_t \times p^3}$ and $P^{rt} \in R^{N_t \times m}$ represent the mapping between the grid currents to the potential at the test points and currents in wire segments to the potential at the test points, respectively. R_c is chosen according to the accuracy of the projection, as described in Section 3.7. The entry P_{ij}^{gt}, which is the potential at the i^{th} test point induced by the unit point current at the j^{th} grid point, is given by:

$$P_{ij}^{gt} = \frac{m_0}{4p} \frac{1}{\left\| \vec{r}_i^{\,t} - \vec{r}_j^{\,g} \right\|} \quad (9)$$

where $\vec{r}_i^{\,t}$ and $\vec{r}_j^{\,g}$ are the coordinates of i^{th} test point and j^{th} grid point, respectively. The entry P_{il}^{rt} is the potential at the i^{th} test point induced by the unit current in l^{th} wire segment, given by:

$$P_{il}^{rt} = \frac{m_0}{4pa_l} \int \frac{1}{\left\| \vec{r}_i^{\,t} - \vec{r}_l^{\,r} \right\|} d\vec{r}_l^{\,r} \quad (10)$$

where $\vec{r}_l^{\,r}$ is the coordinate of l^{th} real wire segment and a_l is the cross section area of that wire segment. Solving equation (8) gives us the grid current vector $I_g(k)$:

$$I_g(k) = [P^{gt}]^{\uparrow} P^{rt} I_r(k) = W(k) I_r(k) \quad (11)$$

where $[P^{gt}]^{\uparrow}$ is the pseudo-inverse of P^{gt} [23] and can be calculated by singular value decomposition. There are two reasons to use the pseudo-inverse here: first, the number of test points may be larger than the number of grid points for cell k, and second, the possible symmetric positions of the test points on the collocation sphere may cause the P^{gt} matrix to be nearly singular and introduce inaccuracies if the normal matrix inverse is used. This procedure provides us with $W(k)$, the part of the projection operator associated with cell k; the j^{th} column of $W(k)$ is the contribution of the j^{th} wire segment in cell k to the p^3 grid currents. Since P^{gt} is small and is taken to be the same for all of the cells, $[P^{gt}]^{\uparrow}$ can be calculated once in the setup step with a very small computational cost, and is directly used in each step of the precorrected-FFT that requires its value. Note that the grid currents obtained from cell k constitute only a part of the currents on these grid points if they are shared with neighboring cells. The grid current on a grid point shared by multiple cells is calculated as the sum of the contribution from all of the wire segments which reside in those cells.

3.2 Calculation of grid potentials by FFT

Once the currents in wire segments are projected to the grid, the grid potentials due to the grid currents are computed through a multi-dimensional convolution, given by:

$$A_g(i, j, k) = HI_g = \sum_{i', j', k'} H(i, i', j, j', k, k') I_g(i', j', k') \quad (12)$$

where $A_g(i, j, k)$ is the grid potential at the grid point whose index in three dimensions is (i,j,k) and each entry of H is given by:

$$H(i, i', j, j', k, k') = \begin{cases} \dfrac{m_0}{4p \left\| r(i, j, k) - r(i', j', k') \right\|} & \text{if } (i, j, k) \neq (i', j', k') \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

which is the contribution to the grid potential at grid point *(i,j,k)* induced by unit point current at grid point *(i',j',k')*. It can be seen easily that (12) has the form of a convolution operation, and the discrete Fast Fourier Transform (FFT) can be exploited to rapidly implement this convolution. On a practical front, we observe that the matrix *H* needs to be computed only once during this computation. Moreover, the number of grid points in each dimension is best chosen as a power of two, or as a value with only small values of prime factors, so that the implementation of the FFT is efficient. For further efficiency, the sparsity properties of I_g and *H* can be exploited.

3.3 Interpolation

After the grid potential is calculated using the FFT, the values of $\sum_{m=1}^{n}(\frac{1}{a_k}\int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ over victim conductors can be obtained through interpolation of the potentials on grid points throughout the cell that the victim conductor lies in. This step is basically the inverse process of the projection step, and the interpolation operator can be obtained by the following theorem [17, 18]:

Theorem: If $\tilde{V} \in R^{m\times 1}$ is an operator that projects a current onto *m* grid points, $\tilde{V}^T$ may be interpreted as an operator which interpolates potential at *m* grid points onto a current coordinate; conversely, if $\tilde{V}^T \in R^{1\times m}$ is an operator that interpolates the potential at *m* grid points onto a current coordinate, $\tilde{V}$ may be interpreted as an operator that projects a current onto the *m* grid points. In either case, $\tilde{V}$ and $\tilde{V}^T$ have comparable accuracy.

The proof of this theorem is provided in [17, 18]. However, whether the interpolation operator is the transpose of the projection operator or not depends on the discretization scheme used in the discretization of the integral equation [24]. As described in [24], if a Galerkin scheme is used, so that the entries of the dense matrix include the integration about both the aggressor discrete element as well as the victim discrete element, the dense matrix will be symmetric and positive definite. In this case, the transpose of the projection operator can be used as the interpolation operator. If (1) and (2) are applied to calculate inductance values, the inductance matrix, *M*, is just the dense matrix resulting from the discretization under the Galerkin scheme, so that the interpolation operator is W^T.

3.4 Precorrection

The grid representation of the current distribution in a cell is only accurate for potential calculations that correspond to long-range interactions. In practice, nearby interactions have the largest contribution to the total induced potentials, and therefore, they must be treated directly and accurately. Since the nearby interactions have already been included in the potential calculation after the above three steps, it is necessary to subtract this inaccurate part from the result of the interpolation step before the accurate measure of nearby interactions is added in.

This is easily done: the part of the value of $\sum_{m=1}^{n}(\frac{1}{a_k}\int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ of a wire segment in cell k due to the currents in wire segments in cell l is *M(k,l)I(l)*, where *I(l)* is the current vector for cell l and *M(k,l)* is the part of the inductance matrix M corresponding to the mutual inductance terms between the victim wire segments in cell k and the aggressor wire segments in cell l. $V_G(k)$ corresponds to the values of $\sum_{m=1}^{n}(\frac{1}{a_k}\int \vec{A}_{km} \bullet d\vec{l}_k da_k)$ of wire segments in cell k, computed from the projection, FFT and interpolation steps. The part of this calculation related to the currents in cell l is:

$$V_G(k,l) = W(k)^T H(k,l) W(l) I(l) \quad (14)$$

where *W(l)* and *W(k)*^{T} are the projection operator in cell l and interpolation operator in cell k, respectively. *H(k,l)* is the part of the multi-dimensional convolution step that calculates the grid potential throughout cell k due to the grid currents throughout cell l. The precorrection step subtracts $V_G(k,l)$ from $V_G(k)$ and then adds the accurate direct interaction *M(k,l)I(l):*

$$V(k) = V_G(k) - V_G(k,l) + M(k,l) I(l) = V_G(k) + \tilde{M}(k,l) I(l) \quad (15)$$

where $\tilde{M}(k,l)$ is a precorrection operator for cell k corresponding to cell l and is given by:

$$\tilde{M}(k,l) = M(k,l) - W(k)^T H(k,l) W(l) \quad (16)$$

Although the $M \times I$ product may be calculated many times (for example, in the loop of calculating moments in PRIMA), the expense of computing $\tilde{M}(k,l)$ is incurred only once in the initial setup step, and can thence be reused. After precorrection, *V(k)* is a good approximation to the real result of $\sum_l M(k,l) I(l)$, because it includes long-range contribution to the potential through the grid representation and short-range contribution through the direct calculation.

3.5 Complete precorrected-FFT algorithm

Combining the above steps leads to the complete application of precorrected-FFT algorithm on the dense inductance matrix and vector product problem. The final solution of the induced voltages is:

$$V = MI = (\tilde{M} + W^T H W) I \quad (17)$$

where W is the sparse projection operator, of which each nonzero entry W_{ij} is the contribution of *jth* entry in the I vector on to the grid current at the i^{th} grid point. H can also be constructed as a sparse matrix for an efficient implementation of FFT. $\tilde{M}$ is a sparse matrix because the number of cells included in the calculation of nearby interactions is small, and each nonzero entry $\tilde{M}(i,j)$ is the error caused by the grid representation during the

calculation of the value of $M(i, j)I_j$ for wire segment *i* due to the current in wire segment *j* in a nearby cell. The complete algorithm, including the setup step, is illustrated in pseudo code as follows:

Precorrected-FFT approach to compute $M \cdot I$:

1. Setup step:
   - 1.1 Construct $[P^{gt}]^{\uparrow}$
   - 1.2 Construct *W* for the whole circuit
     For each cell k = 1 to K
     {
       Construct $P^{rt}(k)$
       Calculate the projection operator for cell k as
       $$W(k) = [P^{gt}]^{\uparrow} P^{rt}(k)$$
       Accumulate the entries in *W(k)* into *W*
     }
   - 1.3 Construct *H* for all of the grid points, calculate the FFT of *H* and store the results:
     $$\tilde{H} = FFT(H)$$
   - 1.4 Construct $\tilde{M}$ for the whole circuit
     For each cell k = 1 to K
     {
       For each nearby cell l = 1 to *N(k)*
       {
         Calculate $W(k)^T H(k,l) W(l)$
         Calculate the mutual inductance terms associated
           with the aggressor cell l and the victim cell k: *M(k,l)*
         Calculate precorrection operator: $\tilde{M}(k,l) = M(k,l) - W(k)^T H(k,l) W(l)$
           for cells k and l
         Accumulate the entries of $\tilde{M}(k,l)$ to build $\tilde{M}$ for the whole circuit
       }
     }
2. Precorrected-FFT step:
   Given the vector *I*
   - 2.1 Projection
     Calculate grid currents: $I_g = WI$
   - 2.2 Convolution
     Compute $\tilde{I}_g = FFT(I_g)$
     Compute $\tilde{A}_g = \tilde{H}\tilde{I}_g$
     Compute $A_g = FFT^{-1}(\tilde{A}_g)$
   - 2.3 Interpolation
     $$V_G = W^T A_g$$
   - 2.4 Precorrection
     $$V = V_G + \tilde{M}I$$

The concept of the precorrected-FFT method lies in the representation of far away interactions by grid potentials, while the nearby interaction are taken into account by direct calculations. This concept can be used for both the electric field and the magnetic field, and therefore for both capacitance and inductance extraction. The kernel of the calculation of both field potentials is $1/r$, where r is the point-to-point distance or the point-to-origin distance. Although the above description of the precorrected-FFT method is superficially similar to that in [17], the implementation and the application of precorrected-FFT in this paper differs from that for the capacitance extraction in several ways. These differences, which constitute the contributions of this paper, are:

1. The computation of the projection operator W for capacitance extraction involves a two-dimensional integration, while for the magnetic field, it is a much more complicated three-dimensional integration.
2. Our objective is to solve $V=MI$ fast and accurately, rather than calculating M exactly. Here I is treated as the a set of fictitious inductor currents and V is the summation of the integration of the magnetic vector potential over wire segments caused by the current in each aggressor wire segment. The magnetic field induced by I as well as V are not real world quantities, unlike capacitance extraction, where the method is used to solve $V=PQ$ for a real physical electric field. We show how the calculated MI product can be used in various simulation schemes, including PRIMA and in circuit simulation using iterative solvers.

3.6 Computational cost and grid selection

Since VLSI chips are thin and flat, one option is to use only one cell in $\hat{z}$ (thickness) direction. In addition, there are three parameters that need to be determined before the precorrected-FFT algorithm is applied to a circuit: p, q *and* d. Parameter p is the number of grid points on each edge of a cell, so that each cell is approximated by p^3 grid points in three-dimensional grid. For example in Figure 1, there are 2^3 grid points throughout the volume of the three-dimensional cell.

Parameter q is the number of nearby cells that are considered in the precorrection step. For example, if we only consider the first nearest neighbors to each cell (defined as all cells that have a vertex in common with the considered cell, including the cell itself), the value of q is 9. Parameter d is the cell size, defined as the length of a cell's edge in the $\hat{x}$ and $\hat{y}$ directions, which we will take to be equal. For a given chip size, the number of cells N_c is inversely proportional to d^2. We reiterate that in order to implement the FFT efficiently, it is convenient to choose the number of grid points as a power of two, or as a number whose prime factors are small.

As the interpolation operator W^T is only the transpose of the projection operator W, the construction of W^T has virtually no overhead, so that we only consider the projection, the FFT and the precorrection steps in the analysis of the computational cost. The complexity of each of these steps can be analyzed as follows:

- In the projection step, if n wire segments and p^3 test points are used to construct W, then the computational cost is $O(p^3 n) \sim O(n)$.

- The cost of FFT is $O(\hat{n}\log(\hat{n}))$, where $\hat{n}$ is the number of grid points and is related with the number of cells by the relation $\hat{n} \propto p^3 N_c \propto n$.
- In the precorrection step, there are approximately n/N_c wire segments per cell on average, and for each wire segment, qn/N_c mutual inductance terms need to be calculated. Since the on-chip wire segments are in practice nearly homogenously distributed, the value of n/N_c is independent of n. The computational cost in this step is therefore $O(q(n/N_c)^2 N_c) \sim O(N_c) \sim O(n)$ over all cells.

From the above analysis, it is easily seen that the computational complexity of the entire precorrected-FFT procedure is $O(n\log(n))$.

It is clear that increasing the values of p and q will both increase the computational cost of the algorithm and its accuracy. Of the three parameters, if p and q are fixed, then a larger value of the cell size will result in a smaller number of cells, so that the computational cost of precorrection is increased. On the other hand, if the cell size is decreased and the number of cells is increased, the cost of performing the FFT will increase. This suggests that there is an optimal cell size that yields a minimum value of cost. To search for this optimum, it is possible to perform a search that starts with a larger cell size and a smaller number of grid points, and then decreases the cell size until the minimum run time is reached. The worst-case accuracy is a function of q; in most of the experiments in this paper, only the first nearest neighbors are included in the precorrection step and p is chosen as a small value, so that the cell size is easily selected. In this sense, the method for choosing the cell size is somewhat easier and more reliable than the methods used in [4][6] to find the local interaction region, since in the precorrected-FFT approach we only need to look for a minimum value of CPU or memory cost with some consideration of accuracy.

3.7 Accuracy of the projection step

As stated in the earlier description, the precorrected-FFT method uses a grid representation for a current distribution. An analysis of each of the steps for sources of errors is as follows:

- The principal source of errors in the precorrected-FFT method lies in the projection step, where grid currents are used to replace currents in wire segments.
- The interpolation step results in the same theoretical error as the projection step, so that it is not necessary to separately consider this step in the analysis of accuracy.
- The FFT step, which is applied to calculate grid potentials, is an efficient implementation of the discrete convolution and does not introduce any theoretical error.
- The direct calculation of nearby interactions introduces no theoretical error.

Therefore, in order to maintain the accuracy of the precorrected-FFT method, it is critical to ensure the accuracy of the projection step.

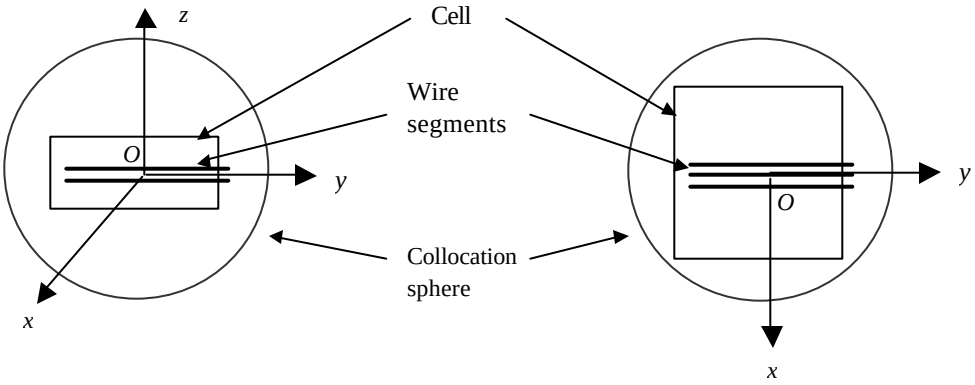
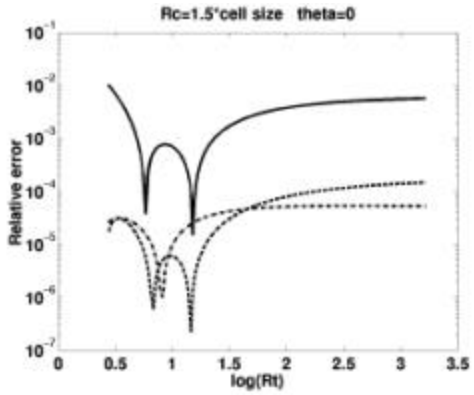


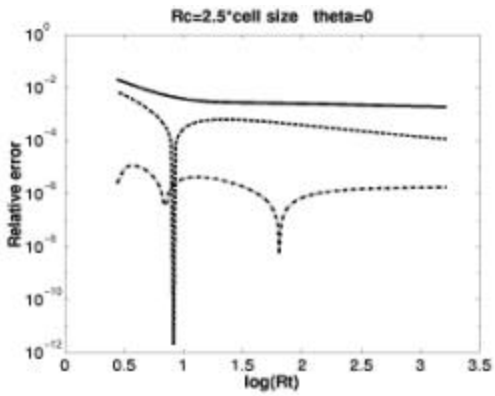
Figure 4: Side view (left) and top view (right) of the experimental setup in the examination of the accuracy of the projection step.

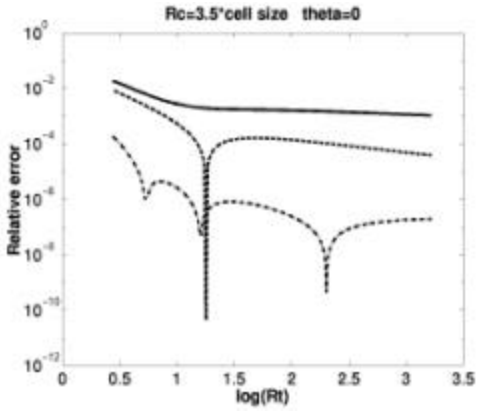
A small circuit, shown in Figure 4, is used to examine the accuracy of the projection step. The setup consists of six 50μm wires lying on two metal layers, with three on the upper layer ($z > 0$) and three on the lower layer ($z < 0$). Each wire is divided into two wire segments of equal length, so that there are 12 segments in all. The width, thickness and spacing of wires are all 1μm. The currents flowing in each wire segment are 100mA in the *y* direction, and the cell size is chosen to be 50μm. The wire segments are off-centered in the *y* direction by 1μm, while the centers of the wire system in *x* and *z* direction are at the origin.

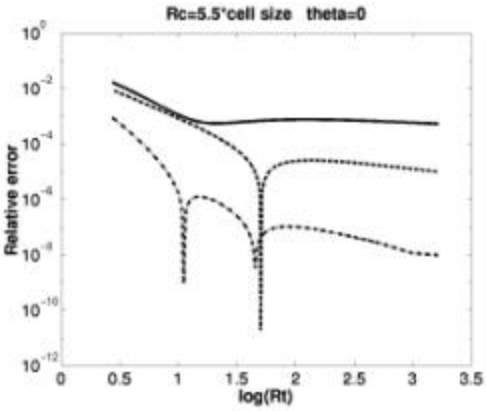
A series of experiments is carried out with different values of the radius R_c of the collocation sphere[1] and of p, where p is set to be 2 or 3 or 4 and R_c is chosen from 1.5, 2.5, 3.5 and 5.5 times the cell size. Theoretically, the grid current representation is only accurate for the magnetic field outside of the collocation sphere. For those neighbor cells that are included in the collocation sphere, the accurate potential calculation should be adjusted by the precorrection step. Since normally at least the nearest neighbor cells are included in the precorrection step, the smallest value of R_c is set to be 1.5 times of the cell size. For a fixed combination of p and R_c, numerous evaluation points (which are different from the evaluation points on the collocation sphere) in three directions are chosen to evaluate the difference between the magnetic potential induced by the current distribution in wire segments and by the p^3 grid currents. A cylindrical coordinate representation is employed so that the coordinates of an evaluation point is expressed as (R_t, q, φ). R_t is the distance of the evaluation point from the origin. The unit of R_t is in terms of the size of a cell. The values of the logarithm of R_t are shown in order to accommodate the large range of R_t values. The directions of evaluation points in cylindrical coordinates, q, are set to be 0°, $45°$ and $90°$ relative to the $+\hat{x}$ direction. For each angle, a set of evaluation points is chosen in the z = 0 plane, such that their distance from the origin varies between 1.5 to 25 times the cell size. Plots of the relative error at these evaluation points are shown in Figure 5.

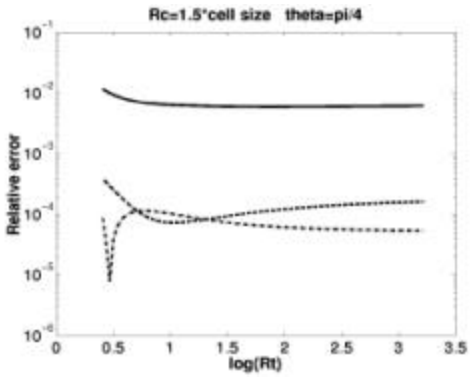
[1] One reason why we choose a spherical collocation surface is that the spherical surface has the minimum surface area, over shapes with the same volume as. For the same number of the collocation points, the smaller surface area of the spherical collocation surface relative to any other surface results in a larger density of the collocation points, which accordingly produces a larger accuracy with the same computational cost.

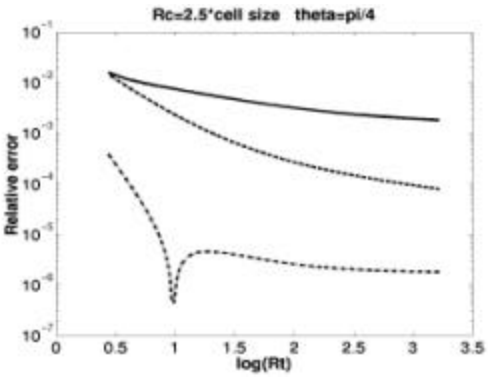

Rc=1.5*cell size theta=0
Relative error
log(Rt)

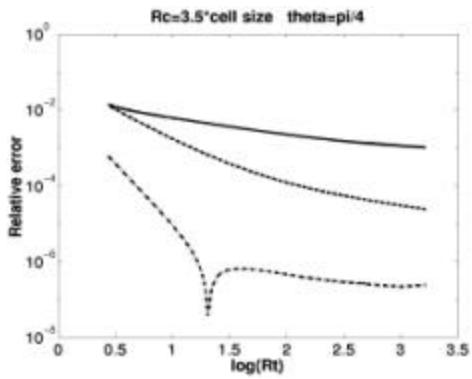

Rc=2.5*cell size theta=0
Relative error
log(Rt)

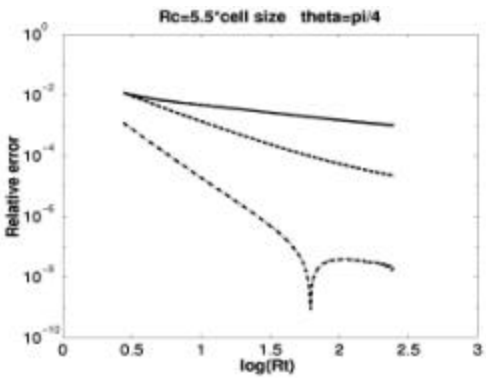

Rc=3.5*cell size theta=0
Relative error
log(Rt)


Rc=5.5*cell size theta=0
Relative error
log(Rt)


Rc=1.5*cell size theta=pi/4
Relative error
log(Rt)


Rc=2.5*cell size theta=pi/4
Relative error
log(Rt)


Rc=3.5*cell size theta=pi/4
Relative error
log(Rt)


Rc=5.5*cell size theta=pi/4
Relative error
log(Rt)

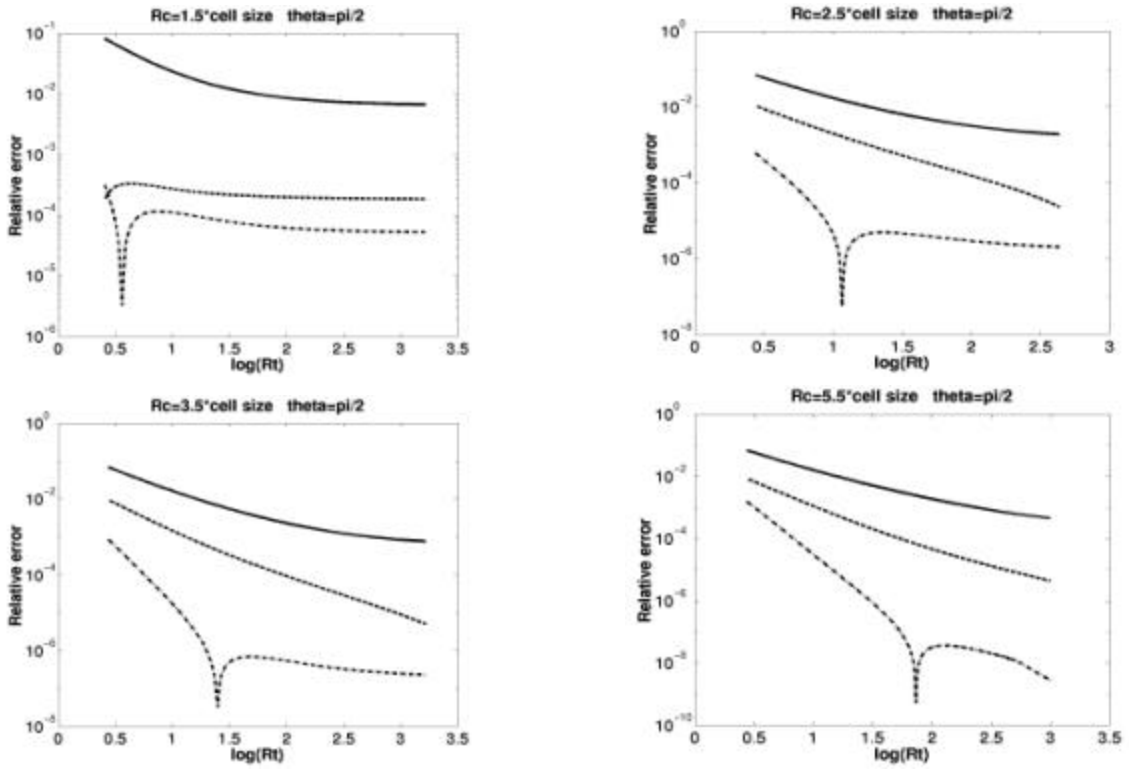
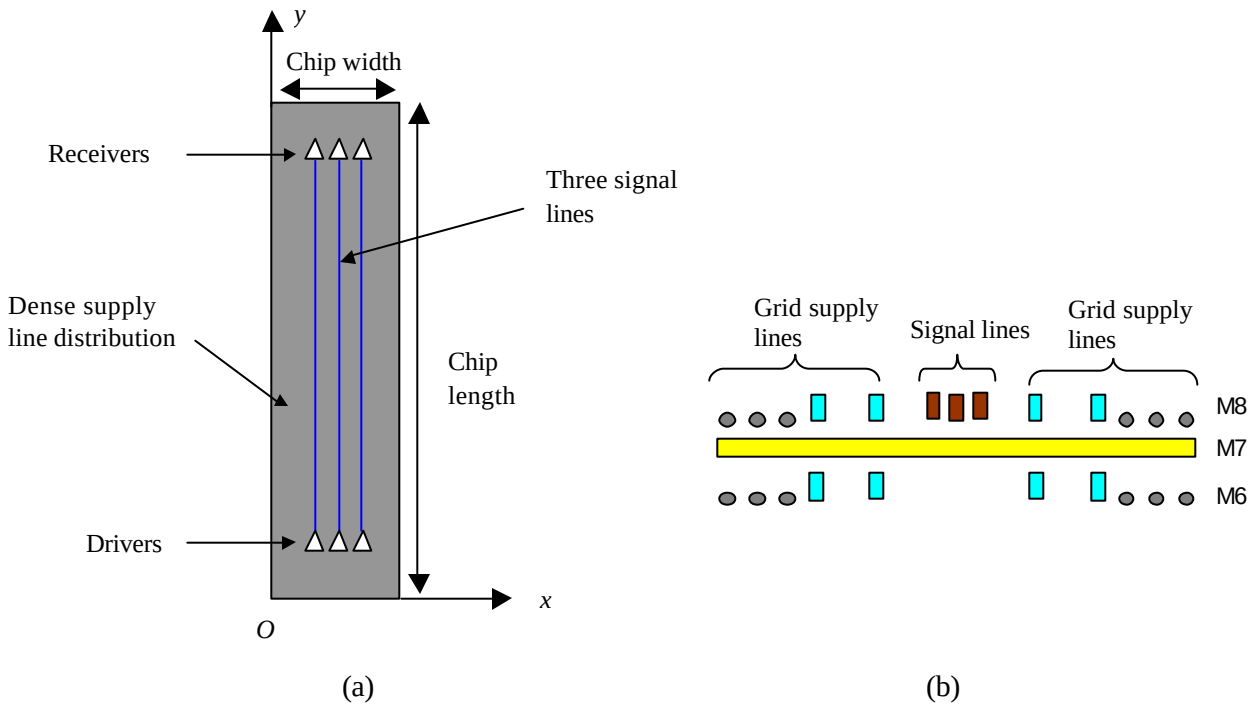


Figure 5: Relative error caused by grid representation with p=2, 3 and 4 and R_c=1.5, 2.5, 3.5, 5.5 times the cell size. Here, theta is the direction of evaluation points, R_c is the radius of the collocation sphere, and R_t is the distance of the evaluation points from the origin in the unit of cell size. The solid line, dashed line and the dash-dot line correspond to p=2, 3 and 4, respectively.

In cases where R_c is small, such as 1.5× the cell size, the error decays slowly with the distance of the evaluation point from the current distribution, and falls off sharply when the evaluation points are near the collocation sphere. It can also be seen that the error decays faster if the radius of the collocation sphere is larger. For example, when R_c is 5.5 cell sizes and p equals 4, the error decreases from 10^{-3} at the first evaluation point to 10^{-8} at the evaluation point that is 25 cell sizes away from the current distribution. When R_c is 1.5 cell sizes and p equals 4, the error is nearly level at 10^{-4} after a sharp change at the collocation sphere. For an R_c value of 2.5×, 3.5× or 5.5× the cell size, the worst error is the same. It is also observed that no matter what the radius of the collation sphere is, the accuracy from a higher order approximation is also higher than that of a lower order approximation when the evaluation point is far away from the collocation sphere. These results are seen to be largely consistent for three values of q.



(a) (b)

Figure 6: Top view (a) and cross sectional view (b) of the test chip with three parallel signal lines on M8. M9 is ignored in the cross sectional view for better clarity (not to scale). The dark background (a) represents the dense supply lines' distribution through out the four metal layers. Only a few grid supply lines are shown in (b) and the uniformity of the pattern is indicated by the dots to the far sides of the signal lines.

4. Experimental results

A set of experiments was carried out on a 400MHz Sun UltraSparc-II computer server to test the accuracy of the response from the precorrected-FFT method, and to compare the results with those of the block diagonal method [6] in terms of accuracy, speed and memory cost. The test circuit is a four metal layer conductor structure on layers M6, M7, M8 and M9 of a nine-layer chip, as illustrated in Figure 6, which shows the top view and the cross sectional view of the structure. It lies within an area whose width is 330μm and thickness is 5μm. The circuit consists of three parallel signal wires, each with 0.8μm width, 0.8μm spacing and 0.5μm thickness. The power/ground wires are distributed densely in the four layers and the signal wires are on M8. The width of the test circuit is fixed throughout the experiments and the length changes as the length of the signal is varied in different experiments. The driver sizes for the three signal wires are identical and are altered with the wire length in order to maintain a slope of 40ps at the near end of the signal wires. The drivers are made to switch at the same time so that the inductance effect is maximized and the error incurred by the precorrected-FFT method can be determined for a worst case condition. In the last part of this section, the experiments on a large industrial clock net are carried out the test the efficiency of the precorrected-FFT method in on-chip inductance simulation.

4.1 Accuracy of the precorrected-FFT method

In the accuracy experiments, the value of p is set to 4, and the nearest neighbors and the next nearest neighbors are included in the direct interaction region. The cell sizes in the x and y direction are each chosen to be 15μm, while in the thickness direction, it is set to 7μm, such that the test structure is at the center of the cell. The radius of the collocation sphere is chosen to be 2.5 times the cell size. The cell size is chosen in the following way:

As explained in Section 3, if p and q are fixed, then a larger value of the cell size will result in a smaller number of cells, so that the computational cost of precorrection is increased while that of the FFT will decrease. On the other hand, if the cell size is decreased and the number of cells is increased, the cost of the FFT will increase, but the cost for precorrection will decrease. Of the three matrices W, H and P to be constructed, the run time for constructing W is not affected by the change of cell size. If the cell size is larger, the cost for constructing H and the run time of the FFT will decrease, but the cost for P increases. On the contrary, if the cell size is smaller, the cost for P decreases while that for H and for the FFT computation increase. In all the experiments in Section 4.1 and 4.2, there are 13 ports and the number of moments per port in PRIMA implementation is 5, as in [6], and it has been demonstrated that the response from the reduced order model converges here even if more moments are used in the simulation. Therefore the FFT have to be carried out 65 times because there are multiple ports and moments. The run time for constructing H and P, the run time for all of the FFT calculations, as well as the total run time for the three are listed in Table 1. The cell size is chosen to be the one that results in the lowest run time.

Cell size (μm)	30	22.5	15	7.5
Construction time of P (sec)	750	450	132	44
Construction time of H + 65 computations of the FFT (sec)	160	265	520	2050
Total run time (sec)	910	715	652	2094

Table 1: The change of run time with the cell size with fixed p=4 and q=9.

A simulation for the same circuit is also carried out with the block diagonal approximation [6]. The partition size in the block diagonal approach is 180μm×150μm, which is much larger than the direct interaction region of 75μm×75μm. Figure 7 shows a comparison of the results from the precorrected-FFT and block diagonal methods with the accurate waveforms for 900μm long wires, with waveforms at both the driver and receiver sides of the middle wire being shown. The accurate waveforms are obtained by using the full inductance matrix in PRIMA without any approximation while the approximate waveforms come from the same PRIMA simulator but using the precorrected-FFT or block diagonal method. There are six waveforms in Figure 7, although only four are clearly

visible since the waveforms from the precorrected-FFT almost completely overlap with those from the accurate simulation. The largest error in the response from precorrected-FFT is less than 1mV. With about 100mV oscillation magnitude induced by inductance, the relative error of the oscillation magnitude is 0.1%. The relative error in the 50% delay for the response from precorrected-FFT is even smaller. Although for a victim line segment, more aggressor line segments are considered in the direct interaction region in the block diagonal method than in precorrected-FFT, the error in the response from the block diagonal procedure is still larger than that of precorrected-FFT. The accumulated errors caused by the dropped mutual inductance terms could too large to be ignored if an accurate simulation is desired.

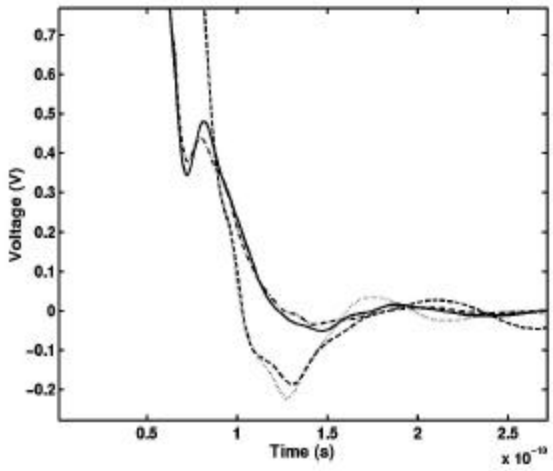


Figure 7: Comparison of waveforms from the precorrected-FFT and the accurate simulation at the driver and receiver sides of the middle wire. Waveforms from the precorrected-FFT and the accurate simulation are indistinguishable.

Because of the high accuracy that can be obtained by the precorrected-FFT method for this example, we observe that we can sacrifice some of the accuracy for higher speed. Different orders of approximation are tested to study the relation between speed, memory requirements and accuracy. The layout tested is similar to the above experiment but the length of the signal wires is extended to 5400μm, which is the largest tested wire length, so as to show the largest reduction in accuracy with the coarsening of the grid. Since there are more than 31,000 inductors in this circuit, including all of the inductors of signal wires and supply wires, a total of nearly one billion mutual inductances is required for accurate simulation. It is therefore impossible to simulate for the accurate waveforms even in PRIMA, let alone in time domain simulation. To simulate the response most accurately, p is set to 4, the cell size is set to be 15μm, and the first, second and third nearest neighbors are considered in the precorrection step. The response obtained from this setup is used as the accurate waveform for comparison purposes.

Other precorrected-FFT simulations are carried out with lower accuracy and a coarser grid, where only the nearest neighbors are considered in the direct interaction region and the cell size is 30μm, which doubles the cell

size in the above experiment. The cell size and the size of the direct interaction region are fixed in these experiments. The grids are variously chosen to be three-dimensional with p=4, p=3, p=2, and two-dimensional with p=4, p=3, p=2. The two-dimensional grid is in the plane that is parallel to the x-y plane and lies at the mid-point of the thickness of the test structure. In the two-dimensional case, the collocation sphere reduces to a collocation circle in the x-y plane, as shown in Figure 8. Reduction of the problem to a two-dimensional grid will increase the efficiency of the computation at some cost in accuracy.

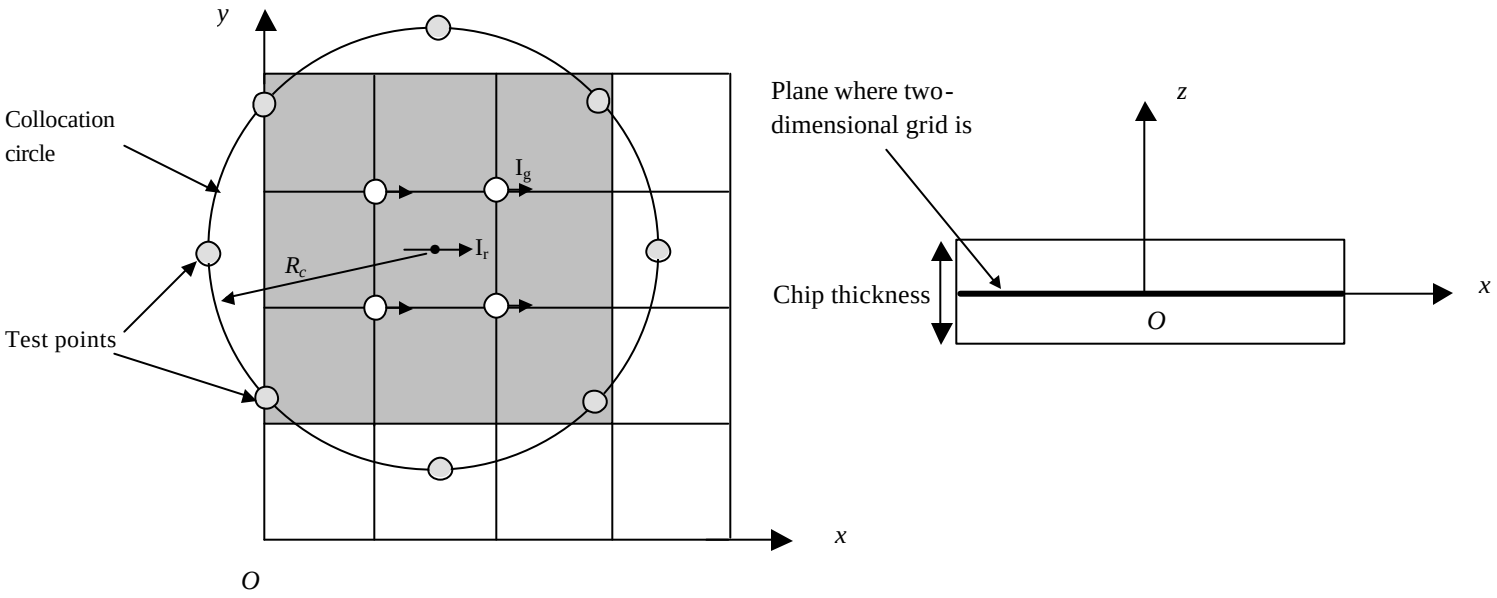


Figure 8: Top view (left) and side view (right) of a two-dimensional grid and the collocation circle.

It is expected that larger cell sizes, smaller values of p, and reduction in the size of the direct interaction region will each contribute to a loss in accuracy, but with an accompanying increase in the speed of the computation and a reduction in the memory requirements. The waveforms at the driver and receiver sides of the middle wire are shown with different levels of accuracy, corresponding to p=2, 3 and 4, are virtually indistinguishable. Closer examination reveals that the error in the 50% delay is insignificant for the three cases, but the relative error corresponding to the overshoot/undershoot is discernible, and is listed in the last column of Table 2. This table also lists the accuracy, memory requirements and speed for each level of approximation.

		Total CPU time (s)	Setup time (s)		Memory requirements (Mb)	Relative error of overshoot/undershoot
			Inductance values	$W, H, \tilde{M}$ matrices		
p=2	2D	2917	1060	148	110	13%
	3D	3094	1060	302	113	12%
p=3	2D	3118	1060	312	113	1.3%
	3D	3682	1060	858	156	<1%
p=4	2D	3175	1060	354	117	<1%
	3D	4090	1060	1196	172	<1%

Table 2: A comparison of the accuracy, memory requirements and CPU time for different parameter settings for the precorrected-FFT in the simulation of three 5400μm long signal wires. Here, "2D" and "3D" correspond to the two-dimensional and three-dimensional cases, respectively. The total CPU time corresponds to the time required for the entire simulation, including the time required by the precorrected-FFT computations.

The setup time is the most time-consuming step in the entire algorithm, and is further divided into two parts. The first part corresponds to the calculation of the inductance values needed for the construction of the precorrection matrix, which is equal for each order of approximation, while the second relates to the time required for the calculation of the W, H and $\tilde{M}$ matrices. For p=3, under a three-dimensional grid, the error at the peak is less than 1mV. The relative error in the oscillation magnitude at that point is 1%, while the speed is increased by 45% as compared with the accurate result. If p is further reduced to 2 under a three dimensional grid, the error is 9mV but the speed is improved by an additional 16% compared to the p=3 case. The two-dimensional grid representation with p=2 results in the largest error of about 10mV and a similar relative error, but the speed is increased only by 6% as compared to its three-dimensional counterpart. The reason for this relatively low speed improvement is that in the case that p=2, the precorrected-FFT is rather fast and the time consumed in the calculation of W, H and $\tilde{M}$ matrices is only a small part of the total setup time. Therefore, even a large increase in the speed of calculation of W, H and $\tilde{M}$ matrices will not yield a significant reduction of the total run time. Another reason is that the number of grid points per cell is only reduced by half by going from the three dimensions to two. On the other hand, if we reduce the three-dimensional grid to two dimensions with p=4, the speed can be increased by 22% because the number of grid points per cell is reduced from 4^3=64 to 4^2=16, and the time required for the calculation of W, H and $\tilde{M}$ matrices plays a more important role in the total setup time. In this case, the accuracy is still high even under a two-dimensional grid. The memory requirements show a similar trend as the run time: for p=4 and p=3, the memory requirements are reduced by 27.5% and 32%, respectively, as we go from the three-dimensional grid to a two-dimensional grid.

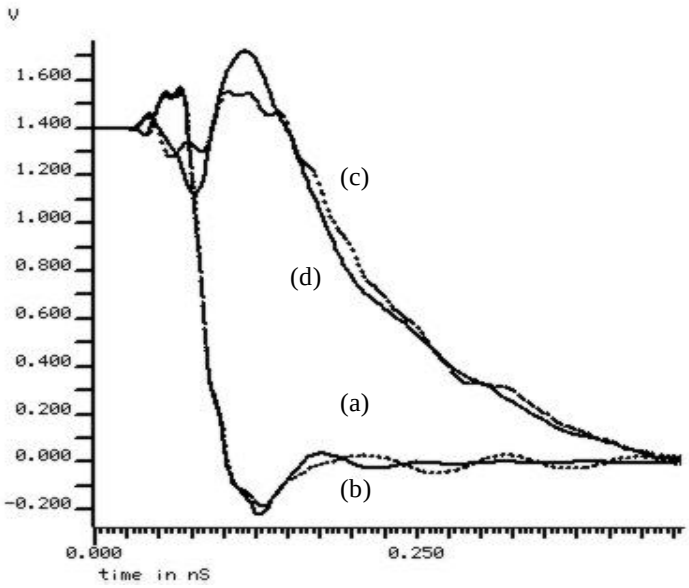


Figure 9: Simulation results at the receiver side of the middle wire from the precorrected-FFT and block diagonal methods for different wire lengths. (a) 900μm, precorrected-FFT (b) 900μm, block diagonal (c) 5400μm, precorrected-FFT (d) 5400μm, block diagonal.

4.2 Comparison of the precorrected-FFT method with the block diagonal method

The comparison in terms of accuracy between the precorrected-FFT and block diagonal methods has been described in Section 4.1. In this section, comparisons in terms of memory consumption and speed between the precorrected-FFT and the block diagonal methods using a Matlab implementation are carried out for structures of different wire lengths. Performance results using an optimized C++ implementation are reported in Section 4.3. The lengths of the signal wires in different experiments are set to 900μm, 1800μm, 3600μm, 4500μm and 5400μm. In the block diagonal method, the partition size is chosen to be 180μm ×150μm (180μm in the x direction and 150μm in the y direction). For the precorrected-FFT method, a two-dimensional grid is imposed with p=2, and the first nearest neighbors are considered for the precorrection step. The cell size is set to 30μm. Figure 9 shows the waveforms computed by the two methods at the receiver end of the middle wire for wire lengths of 900μm and 5400μm. The accuracy, memory requirements and speed for different wire lengths for the block diagonal and precorrected-FFT methods are listed in Table 3. For the wire lengths of 900μm and 1800μm, the results of the precorrected-FFT and block diagonal methods are similar to each other, and the block diagonal method is faster. However, as the wire length increases, the differences in the 50% delay and oscillation magnitude become larger. For example, the 50% delays calculated by the precorrected-FFT and block diagonal methods are 95ps and 100ps respectively for a wire length of 3600μm, which is a difference of about 5%. The difference increases to 8% when the wire length is 4500μm and 12.5% when the wire length is 5400μm. For wire lengths that exceed 1800μm, the precorrected-FFT and block diagonal methods perform their computations at approximately the same speed, but the

former has nearly half the memory requirements as the latter since the partition size for the block diagonal method is much larger than the direct interaction region in the precorrected-FFT, i.e., the number of inductances per wire segment to be calculated by the block diagonal method is much larger than that for the precorrected-FFT approach. Moreover, as the circuit size increases, the setup time and memory consumption are seen to increase at a faster rate for the block diagonal method.

	Total CPU time (s)		Setup time (s)		Memory consumption (Mb)		Relative differences	
	BD	PCFFT	BD	PCFFT	BD	PCFFT	50% delay	Over/Undershoot
900μm	578	683	334	450	66	43	<0.1%	14%
1800μm	1056	1097	571	630	95	56	1%	0.5%
3600μm	1993	1991	1042	1010	153	89	5%	10%
4500μm	2516	2555	1285	1150	184	97	8%	19%
5400μm	3235	2917	1522	1220	210	110	12.5%	>50%

Table 3: A tabulation of the accuracy, memory requirements and CPU time for different circuit sizes using the block diagonal (BD) and precorrected-FFT (PCFFT) methods. The total CPU time corresponds to the time for the entire simulation, including the time required by the block diagonal or precorrected-FFT methods.

Similar trends are seen for the differences in the oscillation magnitude as for 50% delay. For example, if the wire length is 4500μm with a 210mV overshoot, the difference is 40mV. If the wire length is increased to 5400μm, the block diagonal method calculates a larger overshoot of about 300mV, which is about 150mV different from that computed by the precorrected-FFT approach. The precorrected-FFT predicts a more reasonable trend in the overshoot magnitude for different wire lengths: the overshoot increases as the wire length is increased from 900μm to 1800μm, and then decreases gradually as the wires grow longer. When the wire length reaches 5400μm, the output has a smaller overshoot compared with the cases when wires are 4500μm, 3600μm and 1800μm long. However, the trend predicted by the block diagonal method is different: the overshoot magnitude increases from 900μm to 1800μm long wires, and then decreases if the wire length increases from 1800μm to 4500μm, as in the case of the precorrected-FFT method. However, when the wire length increases from 4500μm to the largest tested length of 5400μm, the overshoot is not reduced but is increased in the block diagonal method, which is clearly inconsistent. We observe that the difference between the results from the block diagonal method and those from the precorrected-FFT is larger for longer wires.

Table 4 lists the overshoots and the run time of the responses at the receiver side of the 5400μm wire calculated by the precorrected-FFT and block diagonal methods, with different partition sizes of 30μm×30μm,

180μm×150μm, 330μm×150μm, 330μm×300μm, 330μm×600μm and 330μm×900μm. It is clear that the overshoots given by the block diagonal method do not easy to converge.

	PCFFT	BD					
		30μm× 30μm	180μm× 150μm	330μm× 150μm	330μm× 300μm	330μm× 600μm	330μm× 900μm
Overshoot	151mV	120mV	300mV	120mV	123mV	142mV	161mV
Run time	2917s	681s	3235s	5032s	9700s	6hrs.	12hrs.

Table 4: Overshoots and run times at the receiver side of the middle wire with the length of 5400μm from the precorrected-FFT method (PCFFT) and the block diagonal method (BD) with different partition sizes: 30μm×30μm, 180μm×150μm, 330μm×150μm, 330μm×300μm, 330μm×600μm, 330μm×900μm.

When the partition width increases from 180μm to 330μm, the 300mV bump disappears: the reason may be that more power/ground wires are included in each partition, and the inductance effect is greatly reduced. If the partition length is increased from 150μm to 300μm and then to 600μm and 900μm, with a 330μm partition width, the overshoot increases and nears the result from the precorrected-FFT method. It is impractical to increase the partition size further because the simulation time for 330μm×600μm partition is 6hrs, and includes 26.6M mutual inductances, while the simulation time for 330μm×900μm partition is 12hrs, and uses up about 3Gb memory. On the contrary, the precorrected-FFT method produces a similar overshoot within an hour and only 110Mb memory. We also test the same circuit with a higher level of accuracy in the precorrected-FFT method with the fifth nearest cells included in the precorrection step and the overshoot is only 2mV different. The trends in the overshoots and run time from the precorrected-FFT and block diagonal methods indicate that the precorrected-FFT converges easily, and therefore is a better candidate for fast simulation of large inductive circuits for higher accuracy.

The problem faced here by the block-diagonal method is common to most of the existing algorithms in on-chip inductance extraction. As the circuit size is increased, the local interaction region should be larger to maintain the same accuracy in the simulation. However, it is hard to predict this interaction region *a priori*, and for large circuits, increasing the interaction region gradually is impractical as it could result in very long simulation times. The precorrected-FFT method, on the other hand, overcomes this difficulty by including the calculation of far away inductance interactions using the grid representation.

4.3. Application of precorrected-FFT with optimized implementation on signal lines and a large clock net. In addition to a Matlab-based implementation of the precorrected-FFT method, an optimized version using C++ was also implemented. To demonstrate the efficiency of the precorrected-FFT method, layout structures with

different length of signal wires, as depicted in Section 4.2, and a large global clock net of an industrial giga-hertz microprocessor are simulated using this optimized implementation of the precorrected-FFT method.

For layout structures with different length of signal wires, the number of resistances, capacitances and inductors in circuits and the total CPU times of the simulations in the precorrected-FFT method are listed in Table 5. A three-dimensional grid is imposed with $p=3$, and the first nearest neighbors are considered for the precorrection step. The cell size is set to 30μm. The simulations in the precorrected-FFT method can be very fast. For the circuit with 5400μm signal wire, which includes 32.3K resistances, 64.5K capacitances and 32.3K inductors, the total CPU time is about 6 mins.

Length of signal wires	No. of resistances	No. of capacitances	No. of inductors	Total CPU time (s)
900μm	7.3K	14.7K	7.3K	~ 60
1800μm	12.3K	24.7K	12.3K	137
3600μm	22.3K	44.6K	22.3K	261
4500μm	27.3K	54.6K	27.3K	306
5400μm	32.3K	64.5K	32.3K	358

Table 5: Circuit parameters and run times for layouts with different length of signal wires from the precorrected-FFT method.

The layout of the clock net is shown in Figure 10 and has 4 ports, 12 sinks and 121,065 inductors, which corresponds to 7.3G inductance terms. With the optimized implementation of the precorrected-FFT algorithm, the run time for generating the reduced order model was 21 minutes, using a three-dimensional grid. The signal responses from simulation of the RC model, the ROM generated using precorrected-FFT and block diagonal methods are shown in Figure 11 and the layout and experimental parameters are listed in Table 6. On-chip inductance has a strong effect on the clock net responses. The 50% interconnect delay with the precorrected-FFT method is 130ps, compared with 86ps for the response with the RC model. Relative to the 0.5 Vdd point at the far end response under an RC-only model, the corresponding point with the precorrected-FFT model has a shift of 17ps, while the shift with the block diagonal method is only 6ps. In addition, the 10%-90% transition time at the near and far with the precorrected-FFT model differ from those with the RC-only model by 53ps and 70ps, respectively, while the block diagonal model results in a difference of 20ps and 90ps. In this example, the block diagonal method therefore underestimates the inductance effect on the slope at the near end by 62% and overestimates the effect on the slope at the far end by 28.5%. The partition size in the block diagonal method and the direct interaction region in the precorrected-FFT procedure are both 150μm×150μm. The errors in the responses calculated by the block diagonal method arise from elimination of a large number of far away mutual inductance terms.

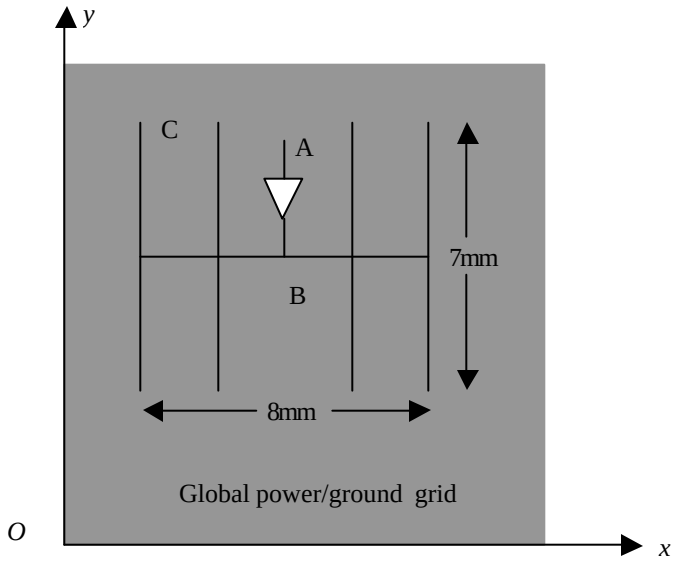


Figure 10: Top view of the layout structure of a global clock net.
(A: driver input, B: driver output, C: receiver input)

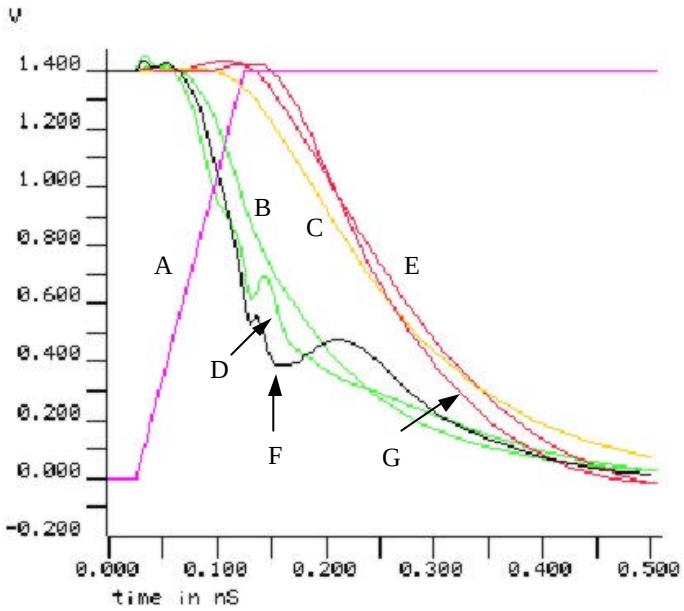


Figure 11: Responses from simulation with an RC-only model, the precorrected-FFT method and the block diagonal method for the near and far ends. A: driver input waveform, B and C: driver output and receiver input, waveform, respectively, under an RC-only model, D and E: driver output and receiver input waveform, respectively, calculated using the precorrected-FFT method, F and G: driver output and receiver input waveform, respectively, calculated by the block diagonal method.

No. of sinks	12
No. of ports	4
No. of inductors	121K
No. of resistances	160K
No. of capacitances	400K
No. of mutual inductance terms	7.3G
Run time	21mins
No. of nodes	245,780
No. of moments per port	10
X/Y/Z dimension (μm)	4798/4768/4.14
Cell size in X/Y/Z	74.97/74.50/4.968
No.of grid points in X/Y/Z per cell	3/3/2
Grid pitch in X/Y/Z	37.485/37.25/4.968
No. of grid points in X/Y/Z	129/129/2
Relative radius of collocation sphere	1.2
No. of collocation points	144
No. of cells in direct interaction region	9

Table 6: Layout and experimental parameters (X, Y, Z: x, y and z directions in Figure 10)

5. Conclusions

A precorrected-FFT algorithm for fast and accurate simulation of inductive systems is proposed in this paper, in which long-range components of the magnetic vector potential are approximated by grid currents, while nearby interactions are calculated directly. All inductance interactions are considered in computing the product of the inductance matrix with a given vector, so that the waveforms at the nodes of interest are calculated accurately. A comparison with the block diagonal algorithm showed that the precorrected-FFT method results in more accurate waveforms and less run time with much smaller memory consumption. Different approximations in the precorrected-FFT method, including using a two-dimensional grid structure, were tested and showed that the lower order of approximation greatly increases the speed and reduces the memory consumption without much loss in accuracy. Experiments carried out on large industrial circuits demonstrate that the precorrected-FFT method is a fast and highly accurate approach for on-chip inductance simulation in large circuits.

Acknowledgements

The authors would like to thank Jacob White for some helpful discussions.

References

[1] A. E. Ruehli, "Inductance Calculations in a Complex Integrated Circuit Environment," *IBM Journal of Research and Development*, pp. 470-481, vol. 16, No. 5, September 1972.

[2] F. W. Grover, *Inductance calculations: Working Formulas and Tables*, Dover Publications, New York, NY, 1946.

[3] Z. He, M. Celik and L. T. Pileggi, "SPIE: Sparse Partial Inductance Extraction," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 137-140, June 1997.

[4] B. Krauter and L. T. Pileggi, "Generating Sparse Inductance Matrices with Guaranteed Stability," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 45-52, November 1995.

[5] K. L. Shepard and Z. Tan, "Return-Limited Inductances: A Practical Approach to On-Chip Inductance Extraction," *Proc. of the IEEE Custom Integrated Circuits Conference*, pp. 453-456, May 1999.

[6] K. Gala, V. Zolotov, R. Panda, B. Young, J. Wang and D. Blaauw, "On-Chip Inductance Modeling and Analysis," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 63-68, June 2000.

[7] A. Devgan, H. Ji and W. Dai, "How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element K," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 150-155, November 2000.

[8] H. Ji, A. Devgan and W. Dai, "KSPICE: Efficient and Stable RKC Simulation for Capturing On-Chip Inductance Effect," Technical Report UCSC-CRL-00-10, University of California Santa Cruz, Santa Cruz, CA, 2000. Available at http://ftp.cse.ucsc.edu/pub/tr/ucsc-csl-00-10.ps.Z.

[9] M. Kamon, M. J. Tsuk and J. White, "FastHenry: A Multipole-Accelerated 3-D Inductance Extraction Program," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 678-683, June 1993.

[10] K. Banerjee and A. Mehrotra, "Analysis of On-Chip Inductance Effects Using a Novel Performance Optimization Methodology for Distributed RLC Interconnects," *Proc. of the ACM/IEEE Design Automation Conference*, pp. 137-140, June 2001.

[11] Y. I. Ismail, E. G. Friedman and J. L. Neves, "Repeater Insertion in Tree Structured Inductive Interconnect," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, pp. 471-481, vol. 48, No. 5, May 2001.

[12] S. C. Chan and K. L. Shepard, "Practical Consideration in RLCK Crosstalk Analysis for Digital Integrated Circuits," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 598-604, November 2001.

[13] L. He and K. M. Lepak, "Simultaneous Shield Insertion and Net Ordering for Capacitive and Inductive Coupling Minimization," *Proc. Of the International Symposium on Physical Design*, pp. 55-60, April 2000.

[14] G. Zhong, G-K. Koh and K. Roy, "A Twisted-Bundle Layout Structure for Minimizing Inductive Coupling Noise," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 406-411, November 2000.

[15] E. Bleszynski, M. Bleszynski and T. Jaroszewicz, "A Fast Integral Equation Solver for Electromagnetic Scattering Problems," *IEEE APS International Symposium Dig.*, pp. 416-419, vol. 1, 1994.

[16] X. Nie, L.-W. Li and J. K. White, "Fast Analysis of Scattering by Arbitrarily Shaped Three-Dimensional Objects Using the Precorrected-FFT Method," *Symposium on High Performance Computation for Engineered Systems*, 2002.
Available at http://web.mit.edu/sma/About/SpecialEvents/Symposium/HPCES/Spore/NieXC-3D.pdf

[17] J. R. Philips and J. K. White, "A Precorrected-FFT Method for Capacitance Extraction of Complicated 3-D Structures," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 268-271, November 1994.

[18] J. R. Philips and J. K. White, "A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, pp. 1059-1072, vol. 16, No. 10, October 1997.

[19] A. Odabasioglu, M. Celik and L. T. Pileggi, "PRIMA: Passive Reduced-Order Interconnect Macromodeling Algorithm," *Proc. of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 58-65, November 1997.

[20] Y. Saad, *Iterative Methods for Sparse Systems*, PWS Publishing Company, Boston, MA, 1996.

[21] C. Hoer and C. Love, "Exact Inductance Equations for Rectangular Conductors with Applications to More Complicated Geometries," *J. Res. Nat. Bureau of Standards*, pp. 127-137, vol. 69C, No. 2, April-June 1965.

[22] D. J. Griffiths, *Introduction to Electrodynamics*, 2nd edition, Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[23] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, NY, 1973.

[24] K. Nabors, F. T. Korsmeyer, F. T. Leighton and J. K. White, "Precorrection, Adaptive, Multipole-Accerelated Interactive Methods for Three-Dimensional Potential Integral Equations of the First Kind," Available at http://rle-vlsi.mit.edu/~white/pubs/siammulti.ps