

pFFT in FastMaxwell: A Fast Impedance Extraction Solver for 3D Conductor Structures over Substrate

Tarek Moselhy
Research Laboratory in Electronics
Massachusetts Institute of Technology
77 Massachusetts Ave,
Cambridge, MA, 02139
Email: tmoselhy@mit.edu

Xin Hu
Ropes and Gray LLP
Boston, MA
Email: xinhu@alum.mit.edu

Luca Daniel
Research Laboratory in Electronics
Massachusetts Institute of Technology
77 Massachusetts Ave,
Cambridge, MA, 02139
Email: luca@mit.edu

Abstract— In this paper we describe the acceleration algorithm implemented in FastMaxwell, a program for wideband electromagnetic extraction of complicated 3D conductor structures over substrate. FastMaxwell is based on the integral domain mixed potential integral equation (MPIE) formulation, with 3-D full-wave substrate dyadic Green's function kernel. Two dyadic Green's functions are implemented. The pre-corrected Fast Fourier Transform (pFFT) algorithm is generalized and used to accelerate the translational invariant complex domain dyadic kernel. Computational results are given for a variety of structures to validate the accuracy and efficiency of FastMaxwell. $O(N \log N)$ computational complexity is demonstrated by our results in both time and memory.

I. INTRODUCTION

The integration of RF, analog and digital circuitry on a single integrated-circuit substrate, or system-on-chip (SoC), is associated with many challenges. For instance, substrate noise may lead to altered circuit performance. In addition, conductor skin and proximity effects may impact current return paths in a network of closely-spaced conductors, as well as quality factor in RF inductors. Moreover, the occurrence of radiated electromagnetic interferences (EMI) may transmit disturbances across different SoC systems. In order to avoid all the above problems, electromagnetic simulation tools are needed that can accurately and efficiently extract wide-band full-wave conductor impedances in the presence of semi-conducting substrates.

In the last decade, electromagnetic simulators based on integral formulations have been the subject of extensive research due to the development of fast algorithms i.e. $O(N \log N)$ for dense linear systems. These algorithms include Fast Multipole Method (FMM) [1]–[3], precorrected-Fast Fourier Transform [4], [5], and SVD compression [6]. Despite of the large variety of acceleration algorithms and efficient codes, there is no single code that is capable of solving 3-D full-wave Maxwell's equations for arbitrary and very large structures over substrate. For instance, FASTCAP [1], FASTHENRY [2] and ISC3 [6] are based on quasi-static approximations and cannot handle full-wave kernels with substrate effects. FASTIMP [5] solves only the full-wave free space propagation problem and is therefore not suitable for simulating substrate effects. EMX [3] is based on 2.5D formulations with planar metal structures assumed of infinitesimal thickness. This is similar to the formulation in IE3D, though the latter has the ability to model 3-D structures. These approximations are efficient for modeling standard IC technology but not suitable for simulating micromachined structures of complex vertical geometries. Finally, a variety of commercial software based either on Finite Element Methods (FEM) or Finite Difference Time Domain (FDTD) are available and can perform accurate analysis but only on relatively small structures.

In this paper we present a 3D full-wave simulator based on the mixed potential integral equation formulation (MPIE) [7], [8], and accelerated using p-FFT. The solver will be released on public domain starting from December 2006 [9].

The simulator is capable of solving Maxwell's equations for arbitrary 3D multi-port structures over substrate. A variety of acceleration techniques are implemented in the simulator to increase its computational efficiency. The Discrete complex image method (DCIM) [10] is used to approximate the spectral domain Green's function into closed form and thus avoid the need for time consuming Sommerfeld integrals. The set of integration routines [11] to transform 3D integrations into 1D are also implemented. In this paper we illustrate the use of an alternative dyadic Green function, which can be used to describe the potential of arbitrary oriented dipoles without extra computational effort. The alternative Green's function has more symmetrical properties, which can be effectively utilized to enforce the generation by our solver of symmetric positive semidefinite matrices for guaranteed passive model order reduction of distributed structures [12]. In addition, we present the complete p-FFT implementation to handle the dyadic, translational invariant dyadic kernel. In the last section of the paper we give a large number of examples demonstrating the accuracy and computational efficiency of our pFFT implementation in FastMaxwell.

II. BACKGROUND

A. MPIE Formulation

As in [7], [8] the following set of integral equation can be used to compute the unknown volumetric current density vector $\vec{J}(r)$ and surface charge distribution $\rho(r)$:

$$\frac{\vec{J}(r)}{\sigma} + j\omega \frac{\mu}{4\pi} \int_V \vec{\bar{G}}_A(r, r') \vec{J}(r') dr' = -\nabla \phi(r) \quad (1)$$

$$\frac{1}{4\pi\epsilon} \int_S G_\phi(r, r') \rho(r') dr' = \phi(r) \quad (2)$$

$$\nabla \cdot \vec{J}(r) = 0 \quad (3)$$

$$\hat{n} \cdot \vec{J}(r) = j\omega \rho(r), \quad (4)$$

Where $\vec{\bar{G}}_A(r, r')$, $G_\phi(r, r')$ are the dyadic and scalar Green's functions, respectively. V, S are the conductors volume and surface area, respectively, $\phi(r)$ is the scalar potential, ϵ is the complex dielectric constant including dielectric losses, μ magnetic permeability, ω the angular frequency is radians.

B. Formulation Discretization

The conductors volume and surface area are discretized using piecewise constant basis functions into filaments and panels (Fig.1) as in [7], [8], [13], respectively. These are then assembled using Galerkin and mesh analysis [7] into a dense linear system of the form:

$$V_{m \times 1} = M_{m \times b} Z_{b \times b} M_{b \times m}^T I_{m \times 1} \quad (5)$$

where b is the number of branches of the equivalent circuit, m is the number of independent meshes, M is the mesh matrix,

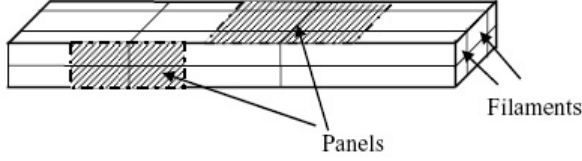


Fig. 1. Conductor volume discretized into filaments supporting the volumetric current density basis functions; conductor surface area discretized into panels supporting the surface charge density basis functions

V is the known excitation vector and I is the unknown current vector. Matrix Z is defined as:

$$Z = \begin{bmatrix} [R] + j\omega[L] & 0 \\ 0 & \frac{[P]}{j\omega} \end{bmatrix} \quad (6)$$

The partial matrices $[R]$, $[L]$ and $[P]$ are generated via Galerkin technique for both volumetric and surface current distributions:

$$R_{ii} = \frac{\ell_i}{\sigma a_i} \quad (7)$$

$$L_{ij} = \frac{1}{a_i a_j} \int_{v_i} \int_{v_j} \bar{G}_A(r, r') \bar{\ell}_i \cdot \bar{\ell}_j dr' dr \quad (8)$$

$$P_{ij} = \frac{1}{S_i S_j} \int_{s_i} \int_{s_j} G_\phi(r, r') dr' dr \quad (9)$$

where a_i , $\bar{\ell}_i$ are the cross sectional area and directional vector of filament i , respectively and S_i is the surface area of panel i .

C. Traditional Dyadic Green's Function

The full-wave substrate effects are captured by using the dyadic Green's function. Different formulations have been previously introduced in literature to satisfy the boundary conditions. In a traditional formulation the horizontal dipole generates horizontal and vertical vector potentials [14]. The scalar potential has two different components corresponding to the horizontal and vertical dipoles.

$$\bar{G}_A(r, r') = \begin{pmatrix} G^{xx} & 0 & 0 \\ 0 & G^{yy} & 0 \\ G^{xz} & G^{yz} & G^{zz} \end{pmatrix}, G_\phi(r, r') = \begin{pmatrix} G^H \\ G^H \\ G^V \end{pmatrix} \quad (10)$$

Detailed derivation of the individual components can be found for instance in [15]. It is noted that closed form expressions are available for the Green's functions in the spectral domain. The two-level discrete complex image method (DCIM) [10] is used to transform the closed form spectral domain Green's functions into the spatial domain counterparts, thus avoiding computing slow converging Sommerfeld integrals. A modified sampling path has been implemented to minimize the number of complex exponential used in the approximation. The final general form of any component of the Green's functions can be written as a sum of real and complex images [11].

$$G_A^{uu}(r, r') = g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k) \quad (11)$$

$$G_A^{vz}(r, r') = \sum_k \frac{\partial}{\partial v} a_k g(x-x', y-y', z+z' + \beta_k) \quad (12)$$

Where $g(x, y, z)$ is the free space Green's function, a_k , β_k are complex constants obtained from the DCIM, $u \in [x, y, z]$ and $v \in [x, y]$.

D. pFFT

The dense linear system in (5) is typically solved using Krylov subspace iterative methods. It should be noted that a good preconditioner is essential for proper operation of the algorithm. A commonly used preconditioner is $M \text{diag}[Z] M^T$. The solver complexity is dominated by a dense matrix vector product, which can be accelerated using the precorrected Fast Fourier Transform (pFFT) algorithm [4], [5]. The acceleration in pFFT is based on sparsifying the dense matrix vector product by separating it into a direct matrix that accounts for nearby interactions, and an approximated matrix that accounts for the far field weak interactions. The resulting direct interactions constitute a sparse matrix because of the relatively few neighbors of a single element and the matrix vector product can therefore be implemented in $O(N)$, where N is the number of unknowns. The weak interactions are approximated via a projection on a regular three-dimensional grid and by making use of the translational invariance of the free space kernel. Interactions on the grid are calculated via fast Fourier transform in $O(N_g \log N_g)$, where N_g is the total number of grid points. The final algorithm complexity is $O(N + N_g \log N_g)$ for the time and $O(N + N_g)$ for the memory. Typically, the number of grid points is comparable to the number of unknowns.

III. AN ALTERNATIVE DYADIC GREEN'S FUNCTION

In spite of its popularity, the dyadic Green's formulation in section II-C is associated with some disadvantages, including the need for two scalar Green's functions for the horizontal and vertical dipoles, respectively. This prevents the usage of the formulation for arbitrarily oriented dipoles. In addition, the dyadic kernel in the presence of vertical current is not symmetric, hence it does not allow the generation of symmetric positive semidefinite matrices which would facilitate the passive model order reduction of distributed systems [12].

In order to address these issues we have implemented an alternative formulation for the dyadic kernel [14]. This formulation is associated with a single scalar Green's function provided the source and observation points are in the same dielectric layer. The existence of a unique scalar potential means that this formulation can be used with dipoles of arbitrary orientations. Furthermore, the dyadic kernel of the alternative formulation is symmetric:

$$\bar{G}_A(r, r') = \begin{pmatrix} G^{xx} & G^{xy} & 0 \\ G^{xy} & G^{xx} & 0 \\ 0 & 0 & G^{zz} \end{pmatrix}, G_\phi(r, r') = G^V \quad (13)$$

The closed form spectral domain components of the Green's function are summarized in the appendix of [14]. By applying the discrete complex image method the spectral domain components can be transformed into closed form spatial counterparts. The spatial domain components can be summarized in the following closed form:

$$G_A^{vv}(r, r') = g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k) + \sum_k a_k \frac{\partial^2}{\partial v \partial v'} g(x-x', y-y', z+z' + \beta_k) \quad (14)$$

$$G_A^{zz}(r, r') = g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k)$$

$$G_A^{xy}(r, r') = \sum_k a_k \frac{\partial^2}{\partial x \partial y'} g(x-x', y-y', z+z' + \beta_k) \quad (15)$$

Where $g(x, y, z)$ is the free space Green's function, a_k , β_k are complex constants obtained from the DCIM, and $v \in [x, y]$.

$[x, y]$. We note that the reduction of the number of scalar potentials is compromised by an added term in the components G_A^{vv} . Furthermore, we observe that no first order derivatives are present in the Green's functions components but rather second order derivatives. Fortunately, we can generalize the integration schemes presented in [11], such that they can be used to accelerate the newly introduced second order terms.

$$\begin{aligned} \int_V \int_{V'} \frac{\partial}{\partial h_1} \frac{\partial}{\partial h_2} g(r, r') dV' dV &= \int_V dV \frac{\partial}{\partial h_1} \int_{V'} \nabla' \cdot (\bar{h}_2 g(r, r')) dV' \\ &= \sum_{S'} \int_V dV \nabla \cdot (\bar{h}_1 (\int_{S'} (\bar{h}_2 \cdot d\bar{S}') g(r, r'))) \\ &= \sum_S \sum_{S'} \int_S (\bar{h}_1 \cdot d\bar{S}) (\int_{S'} (\bar{h}_2 \cdot d\bar{S}') g(r, r')) \end{aligned}$$

where $h_1 \in [x, y]$ and $h_2 \in [x, y]$. Each summation is performed over the two surfaces perpendicular to the corresponding integration variable. We therefore have transformed the three dimensional integration with the second order kernel into a sum of four two dimensional integrations. These surface integrals can be accelerated using the same integration technique in [11]. The formulation presented in this paper has the major advantage of handling dipoles with arbitrary orientation, provided they are confined to a single dielectric layer. For conductor structure distributed in different layers either formulation A or formulation C in [14] must be used.

IV. P-FFT FOR FULL-WAVE DYADIC KERNEL

A. pFFT for Traditional Green's Function

In this section we show how the MPIE formulation with the translational invariant traditional Green's function (section II-C) can be accelerated using pFFT. Only the traditional formulation is considered in this subsection, while all necessary modifications required for the alternative symmetric formulation are detailed in the next subsection. Equation (5) needs to be modified such that the unknown vector is the branch currents rather than mesh currents. This is because the branch currents are the physical quantity that are associated with the discretization elements and that are projected on the grid.

$$V_{m \times 1} = M_{m \times b} Z_{b \times b} I_{b \times 1} \quad (16)$$

where $I_{b \times 1}$ is the vector composed of the currents density vector in the filaments $I_f^T = [I_x, I_y, I_z]$ and the current in the panels I_p

$$I_{b \times 1}^T = \begin{pmatrix} I_f^T \\ I_p^T \end{pmatrix} \quad (17)$$

Furthermore, equations (5) and (7)-(10) define the branch voltages $V_{b \times 1} = Z_{b \times b} I_{b \times 1}$:

$$V_{b \times 1} = \begin{pmatrix} [R] I_f \\ 0 \end{pmatrix} + \begin{pmatrix} j\omega [L_{xx}] I_x \\ j\omega [L_{yy}] I_y \\ j\omega ([L_{xz}] I_x + [L_{yz}] I_y + [L_{zz}] I_z) \\ \frac{[P]}{j\omega} I_p \end{pmatrix} \quad (18)$$

Matrix $[R]$ is diagonal and therefore the first matrix vector product $[R] I_f$ does not require acceleration. The second term is composed of six matrix vector products, $[L_{uu}] I_u, [L_{vz}] I_v$ and

$[P] I_p$. The elements of the matrices $[L]$ and $[P]$ are:

$$\begin{aligned} L_{uu}^{ij} &= \frac{1}{a_i a_j} \int_{V_i} \int_{V_j} dr' dr (g(x-x', y-y', z-z') \\ &+ \sum_k a_k g(x-x', y-y', z+z' + \beta_k) \bar{\ell}_{iu} \cdot \bar{\ell}_{ju}) \quad (19) \end{aligned}$$

$$\begin{aligned} L_{vz}^{ij} &= \frac{1}{a_i a_j} \int_{V_i} \int_{V_j} dr' dr \\ &\cdot \sum_k a_k \frac{\partial}{\partial v} g(x-x', y-y', z+z' + \beta_k) \bar{\ell}_{iv} \cdot \bar{\ell}_{jz} \quad (20) \end{aligned}$$

$$\begin{aligned} P_{ij} &= \frac{1}{S_i S_j} \int_{S_i} \int_{S_j} dr' dr (g(x-x', y-y', z-z') \\ &+ \sum_k a_k g(x-x', y-y', z+z' + \beta_k)) \quad (21) \end{aligned}$$

Each of the six matrix vector products is accelerated independently using pFFT, because of the different nature of the discretization elements and Green function kernel in each product. Without loss of generality only (19) and (20) are treated in this analysis since (21) is identical to (20) up to the dimension of the integration. From (19) and (20) we observe that three different projection matrices are required; for $g(r)$, $\frac{\partial g(r)}{\partial x}$, and $\frac{\partial g(r)}{\partial y}$ [5]. The first term in (19) is clearly translational invariant and can be handled using traditional pFFT. The second term in (19) and the term in (20), which are a result of the Green's function associated with the complex images, are translational invariant in x , and y but do change with z . The resulting matrix is therefore two-level Toeplitz and one level Hankel. Therefore, these terms cannot be treated with traditional pFFT. The Hankel portion of the matrix must first be transformed into a Toeplitz matrix using an appropriate permutation, multiplied by the vector using FFT and then re-transformed back using the inverse permutation:

$$Hx = P^{-1} P H x = P^{-1} (P H) x = P^{-1} (T) x = P^{-1} (T x)$$

where H is a two level Toeplitz and one level Hankel matrix, T is a three level Toeplitz matrix, x is the input vector, and $P = P_3$ is a permutation matrix defined recursively, such that

$$P_{N+1} = \begin{pmatrix} P_N & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & P_N \end{pmatrix}$$

where P_1 is the anti-diagonal unity matrix with elements $(p_{i, N-i+1} = 1)$, and $P_3^{-1} = P_3$. The size of the matrices P_3 , P_2 , and P_1 is dependent on the number of grid points in the x, y, z directions, respectively. The Hankel matrix is generated by traversing the grid in the sequence z, x, y . The proposed permutation has the effect of reversing the order of the z grid index, which means that the multiplication can be implemented without any extra cost by mapping the z index as $k \leftarrow K - k$, where k is the z index and K is the total number of grid points in the direction. We conclude that the vector multiplication with the Hankel matrix can therefore be implemented with exactly the same computational complexity as the Toeplitz matrix, i.e. in $O(N_g \log N_g)$. After constructing the six matrix vector product the mesh voltages are obtained from the branch current using (16) and (18):

$$V_{m \times 1} = M_{m \times b} V_{b \times 1}$$

Consequently, a standard Krylov subspace iterative technique can be used to solve the dense linear system in $O(N + N_g \log N_g)$.

B. pFFT for the Alternative Green's Function

In this section we present the pFFT implementation with the alternative Green's function presented in section III. Equations

(16) and (17) in section IV-A are independent on the kernel, and hence are also valid for this alternative implementation. Equations (18) - (21) are re-written for the alternative kernel:

$$V_{b \times 1} = \begin{pmatrix} [R]I_f \\ 0 \end{pmatrix} + \begin{pmatrix} j\omega([L_{xx}]I_x + [L_{xy}]I_y) \\ j\omega([L_{xy}]I_x + [L_{yy}]I_y) \\ j\omega([L_{zz}]I_z) \\ \frac{[P]}{j\omega}I_p \end{pmatrix} \quad (22)$$

The matrix elements are:

$$L_{vv}^{ij} = \frac{1}{a_i a_j} \int_{V_i} \int_{V_j} dr' dr (\bar{\ell}_{iv} \cdot \bar{\ell}_{jv}) (g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k) + \sum_k \frac{\partial}{\partial v'} \frac{\partial}{\partial v} a_k g(x-x', y-y', z+z' + \beta_k)) \quad (23)$$

$$L_{zz}^{ij} = \frac{1}{a_i a_j} \int_{V_i} \int_{V_j} dr' dr (\bar{\ell}_{iz} \cdot \bar{\ell}_{jz}) (g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k)) \quad (24)$$

$$L_{xy}^{ij} = \frac{1}{a_i a_j} \int_{V_i} \int_{V_j} dr' dr \bar{\ell}_{ix} \cdot \bar{\ell}_{jy} \cdot \sum_k \frac{\partial}{\partial x'} \frac{\partial}{\partial y} a_k g(x-x', y-y', z+z' + \beta_k) \quad (25)$$

$$P_{ij} = \frac{1}{S_i S_j} \int_{S_i} \int_{S_j} dr' dr (g(x-x', y-y', z-z') + \sum_k a_k g(x-x', y-y', z+z' + \beta_k)) \quad (26)$$

From (22) six different matrix vector products are required. By utilizing the symmetry of the new Green's functions the computational effort required for the mutual inductance terms can be significantly reduced. However, we should note that this alternative Green's function requires three interpolation matrices as opposed to only one interpolation matrix required by the traditional Green's function. This is because of the three operators in the outer integral in (23) and (25): $g(r')$, $\frac{\partial g(r')}{\partial x}$ and $\frac{\partial g(r')}{\partial y}$ [5]. The number of projection matrices is still three. Fortunately, the computational effort required for the additional interpolation matrices is insignificant and does not affect the efficiency of the pFFT algorithm, while the symmetry of the Green's function typically results in a net significant cost reduction both in time and memory. However, we further highlight that the main advantage of the alternative Green's function is its ability to handle dipoles in arbitrary orientation.

C. Implementation Optimization

The computational time consumed by the pFFT algorithm is dominated by both the fill time and the solution time. To further increase the computational efficiency of FastMaxwell, the direct and pre-correction matrices fill time is optimized by avoiding redundant calculations:

- 1 Mutual interactions are identical because of the symmetry properties of the scalar Green's function and the used Galerkin technique. $G(r, r') = G(r', r)$
- 2 The regularity of the structures and the translational invariant properties of the Green's function can be effectively used to reduce the time by tabulating computed integrals and reusing them.
- 3 Near interactions in the direct matrix are computed using quasi static approximations, thus accelerating the integration without loss of accuracy.

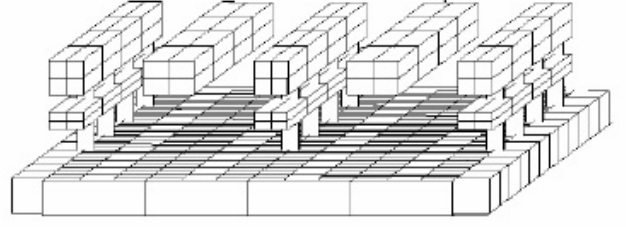


Fig. 2. Microstrip tub geometry; only a small segment of the structure is demonstrated for clarity

In addition, the convolution matrix fill time can be significantly reduced by observing that to perform the FFT, the Toeplitz matrix is embedded into a circulant matrix. Because of the symmetry property of the scalar Green's function, the Toeplitz matrix is symmetric and is completely defined by N_g elements, while the circulant matrix is of size $2N_g$. This means that N_g Toeplitz elements can be reused, corresponding to a reduction factor in time and memory of 2. The computational time and memory required to handle the three-level Toeplitz matrix are thus reduced by a factor of 8 and that of the two level Toeplitz and one level Hankel matrix is reduced by a factor of 4.

V. RESULTS

A. Accuracy Validation

In this section we establish the accuracy of our solver by validating its results in examples small and simple enough, such that they can be solved also by available commercial software. **Shorted transmission line.** The first example (Fig.2) is a microstrip tub [16] consisting of two quarter wavelength copper microstrip lines embedded in a half space of dielectric constant 4.4 and isolated by a microstrip tub. The lines are 10um above a ground plane and have dimensions: width=5um height=4um length=400um, and center to center separation 20um. Three identical parallel shields are used to isolate the lines. Each shield consists of two parallel lines of width 4um, thickness 1um and 4um and distance to ground 5um and 10um, respectively. A set of 50 vertical square vias connect the lines and the ground. The signal carrying microstrip lines are terminated by 50Ω lumped element resistances for matching. The isolation coefficient S_{12} is simulated and compared against results obtained from IE3D (Fig.3). **Microstrip line.** The second example is a single microstrip line terminated by a short circuit and placed in a half space of dielectric constant 10.2. The line is surrounded by two shields as described in the last example. In Fig.4 FastMaxwell results are compared against IE3D. Very good matching is observed. **Array of RF inductors.** The third example we use to establish the accuracy of our solver consists of an array of 16 element array of square inductors (Fig.5). All the inductors are identical and are separated by 200um. Each inductor area is $0.01mm^2$. Copper wires are 2um thick and 15um wide. The separation distance between each of the 3 turns is 5um. An underpass is situated 1um below the inductor to contact the center of the spiral. The isolation coefficient S_{12} between one inductor in the corner of the array and all the inductor in the diagonal of the array including the element itself are calculated and compared against IE3D (Fig. 6). The small discrepancies are mainly due to the excitation mechanism and the planar approximation in IE3D.

B. P-FFT Validation

The following examples have been used to demonstrate the computational efficiency of the pFFT implementation:

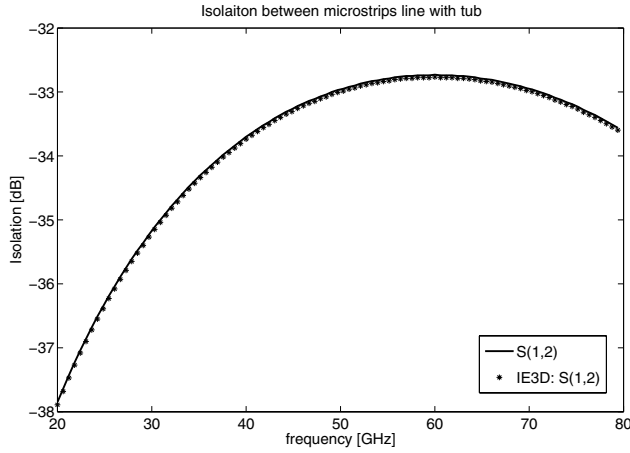


Fig. 3. Isolation coefficient S_{12} between two microstrip lines in the microstrip tub. FastMaxwell (continuous line) compares very closely to IE3D (dotted)

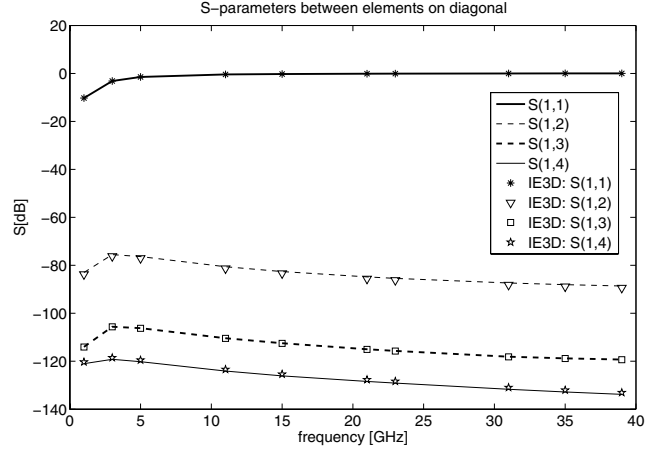


Fig. 6. Comparison of isolation coefficients simulated from FastMaxwell (continuous line) and IE3D (dotted shapes). All four curves compare very closely

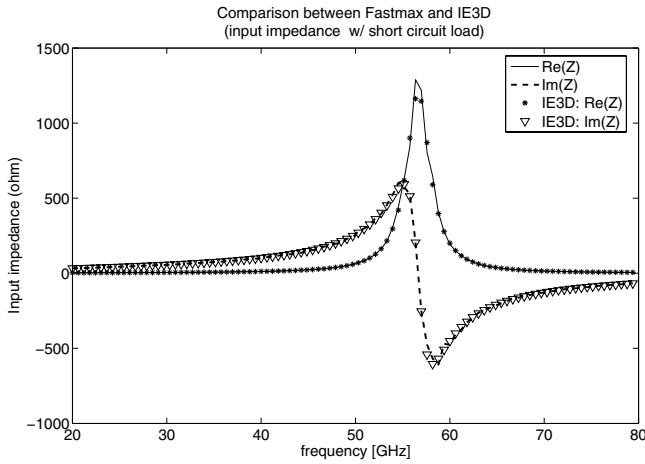


Fig. 4. Input impedance of a single microstrip line in microstrip tub. FastMaxwell (continuous line) compares very closely to IE3D (dotted)

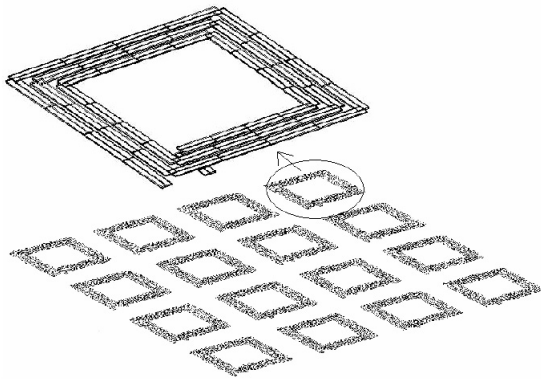


Fig. 5. discretized three- turn square RF-inductor and sixteen element inductor array

- 1 Industry provided three dimensional interconnect structure composed of two parallel microstrip lines surrounded by a set of horizontal wires and connected to the ground plane via six thick vias (Fig. 7).
- 2 two microstrip lines isolated by a microstrip tub and terminated by 50 ohm lumber resistor.
- 3 three-turn square RF inductor placed 4.8um above a patterned finite ground plane.
- 4 Array of 8 inductors on top of semi-conducting substrate
- 5 Array of 16 inductors in free space
- 6 Power and ground distribution grid with 400 ports and two metal layers
- 7 Power and ground distribution grid with 900-ports and of two metal layers.

Note that fullwave analysis is necessary in testcases 4 to 7 since the overall array dimension exceeds one wavelength. All the simulations have been performed on a desktop computer with a Pentium IV microprocessor with 2GHz clock frequency, 256MB of cache, 1GB of RAM, and operating under Redhat Enterprise Linux, kernel 2.4.21-37.EL. The parameters of the pFFT algorithm such as grid size, stencil sizes and iterative algorithm convergence criteria are tuned to obtain accuracy of 5% compared to the exact vector matrix product. If exact results cannot be obtained, then pFFT is repeated with higher accuracy until change in results does not exceed 2%. Table. I summarizes the overall time and memory performance of the complete solver. In the last two examples the exact results cannot be obtained and numbers are estimated based on complexity asymptotes. It can be observed that the time computational complexity of FastMaxwell is $O(N \log N)$ and that of memory is $O(N)$, as opposed to $O(N^2)$ and $O(N^2)$ for the exact vector matrix product.

C. Validation of pFFT Optimization

The implementation optimization techniques suggested in section IV-C are validated by simulating the array of identical inductors with different number of inductors and with different discretizations. In Fig. 8 pFFT overall time is decomposed into its two main components, matrix fill time and solution time. Matrix fill time is shown for the full-wave substrate Green's function with integrations calculated with and without the optimization techniques. It is observed that the matrix fill time dominates for small sized and medium sized problems up to a few tens of thousands of unknowns, because of the complexity of the integrated kernel and the lack of closed form analytical

TABLE I
COMPARISON OF FULL WAVE OVERALL TIME AND MEMORY
REQUIREMENTS FOR BOTH pFFT AND EXACT VECTOR MATRIX PRODUCT
SOLVERS ON PENTIUM IV AT 200GHZ AND 1GB OF RAM

Structure	Num. unknowns	pFFT		Exact Method	
		Memory in (MB)	Time in minutes	Memory in (MB)	Time in minutes
3D interconnect	1232	8.7	8	10	11
Microstrip Tub	1428	9	8	10	13
Patterned Ground Plane	3295	18.4	21	98.3	98
8 Inductor Substrate	4594	24.7	28	150	133
16 Inductor Array	25342	119	100	4034	estimated 22 hour
20 × 20 Power Grid	43486	150	196	estimated 64140	estimated 4 days
30 × 30 Power Grid	99342	423	500	estimated 413000	estimated 16 days

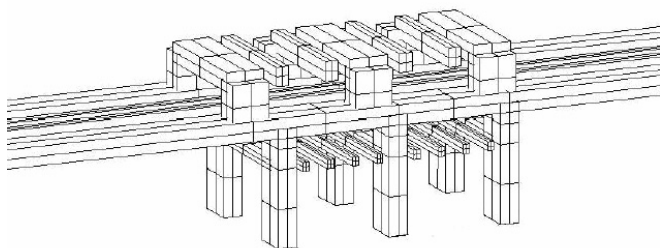


Fig. 7. Portion of the industry provided three-dimensional interconnect structure

formula. This time is reduced by an average factor of 8 if the proposed optimization techniques are used. On the other hand, the solution time dominates in large scale examples and behaves as $O(N \log N)$. The only way to reduce the solution time is to use a better preconditioner and therefore reduce the number of iterations required for convergence. Analogous conclusions are valid for the memory requirements of small and large examples: in small examples the memory consumed is a function of N and the number of direct neighbors of each element. Asymptotically the number of neighbors is a constant determined by the pFFT parameters and the memory computational complexity grows as $O(N)$.

VI. CONCLUSION

In this paper we have presented a complete pFFT implementation of a fullwave electromagnetic field solver based on the MPIE formulation with dyadic substrate Green function. The alternative Green function implemented in the code can lead to computational cost reduction in several examples. Most importantly it allows the use of dipoles (i.e. filaments) in arbitrary orientations, and it helps providing symmetric positive semidefinite matrices, hence facilitating the passive model order reduction of distributed systems. The pFFT algorithm has been extended and optimized to accelerate the fullwave substrate Green's function. The accuracy and efficiency of FastMaxwell and the implemented acceleration algorithms has been established by a variety of examples as large as 100,000 unknowns. The pFFT computational complexity of $O(N \log N)$ in overall solver time and $O(N)$ in memory usage has been observed from our results.

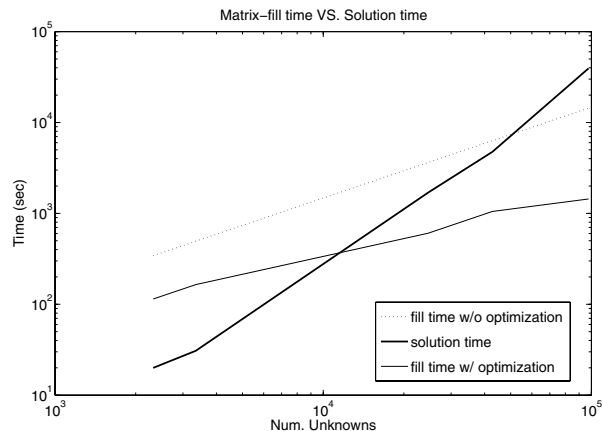


Fig. 8. Comparison between solution time and matrix fill time with and without implementation optimization. The effect of the implementation optimization is to reduce the average matrix filling time by a factor of 8.

ACKNOWLEDGMENT

The authors would like to acknowledge Ibrahim Elfadel from IBM T.J. Watson Research Center for providing industrial examples used for the validation of the performance of FastMaxwell. This work was supported in part by Semiconductor Research Corporation, MARCO, and by the National Science Foundation.

REFERENCES

- [1] K. Nabors and J. White, "FASTCAP A multipole-accelerated 3-D capacitance extraction program," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 10, pp. 1447-1459, November 1991.
- [2] M. Kamon, M. J. Tsuk, and J.K. White, "FastHenry: A multipole-accelerated 3-D inductance extraction program," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 9, pp. 1750-1758, September 1994.
- [3] S. Kapur and D.E. Long, "Large Scale Full-Wave Simulation," In *Proc. 41st Design Automation Conf.*, pages 744-749, 2004.
- [4] Joel R. Phillips and J. K. White, "A precorrected-FFT method for electrostatic analysis of complicated 3D structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1059-1072, 1997.
- [5] Z. Zhu, B. Song, and J. K. White, "Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3-D geometries," In *Proc. 40th Design Automation Conf.*, pages 712-717, 2003.
- [6] S. Kapur and D.E. Long, "IES3: A fast integral equation solver for efficient 3-dimensional extraction," *International Conference on Computer Aided-Design*, pp. 448-455, 1997.
- [7] M. Kamon, N. Marques and J. K. White, "Generating compact guaranteed passive reduced-order models of 3-D RLC interconnects" *IEEE Transaction on Advanced Packaging*, Vol.27, Nov. 2004.
- [8] A. E. Ruehli, "Equivalent Circuit Models for Three Dimensional Multiconductor Systems," *IEEE Trans. Microwave Theory Tech.* Vol. 22, March 1974.
- [9] "http://www.rle.mit.edu/cpg/fastmaxwell.htm"
- [10] M.I. Aksun, "A Robust Approach for the Derivation of Closed-Form Green's Functions," *IEEE Trans. Microwave Theory and Tech.*, Vol. 44, No. 5, May 1996.
- [11] Xin Hu, J.K. White, Jung Hoon Lee, L. Daniel, "Analysis of full-wave conductor system impedance over substrate using novel integration techniques", In *Proc. 42nd Design Automation Conf.*, pages 147-152, 2005.
- [12] L. Daniel, J. Phillips, "Model Order Reduction for Strictly Passive and Causal Distributed Systems", *IEEE/ACM 39th Design Automation Conference*, New Orleans, Jun 2002.
- [13] Luca Daniel, *Simulation and Modeling Techniques for Signal Integrity and Electromagnetic Interference on High Frequency Electronic Systems*, Ph.D. thesis UCB EECS Department, 2003.
- [14] K.A. Michalski, D. Zheng, "Electromagnetic Scattering and Radiation by Surfaces of Arbitrary Shape in Layered Media, Part I: Theory," *IEEE Trans. Antennas Propagat.*, Vol.1.38.No. 3, March 1990.
- [15] Xin Hu, *Full-wave analysis of large conductor systems over substrates*, Ph.D. thesis MIT EECS Department, 2006.
- [16] A. Komijani, A. Hajimiri, "A wideband 77GHz, 17.5dBm power amplifier in silicon," *Proc. of the IEEE Custom Integrated Circuits Conference*, pages 571-574, Sept. 2005.