

The Unified-FFT Grid Totalizing Algorithm for Fast $O(N \log N)$ Method of Moments Electromagnetic Analysis with Accuracy to Machine Precision

Brian Rautio^{1, *}, Vladimir I. Okhmatovski², and Jay K. Lee³

(Invited Paper)

Abstract—While considerable progress has been made in the realm of speed-enhanced electromagnetic (EM) solvers, these fast solvers generally achieve their results through methods that introduce additional error components by way of geometric type approximations, sparse-matrix type approximations, multilevel type decomposition of interactions, and assumptions regarding the stochastic nature of EM problems. This work introduces the $O(N \log N)$ Unified-FFT grid totalizing (UFFT-GT) method, a derivative of method of moments (MoM), which achieves fast analysis with minimal to zero reduction in accuracy relative to direct MoM solution. The method uniquely combines FFT-enhanced Matrix Fill Operations (MFO) that are calculated to machine precision with FFT-enhanced Matrix Solve Operations (MSO) that are also calculated to machine precision, for an expedient solution that does not compromise accuracy.

1. INTRODUCTION

There are many speed-enhanced numerical methods of computational electromagnetics today that represent major achievements in terms of speed and efficiency compared to their predecessors. In the algorithms based on the Moment Method (MoM) solution of the integral equations (IEs), the discretization of Maxwell's equations produces a dense matrix equation. The traditional and direct methods for solving such equations require $O(N^3)$ operations to arrive at a solution for a problem with N degrees of freedom.

However, there are many developed methods which are able to solve the same systems with $O(N \log N)$ operations by making certain approximations. In the class of the FFT-based algorithms such as the Pre-corrected FFT (PFFT) method [1] and CG-FFT algorithm [2], the translational invariance of the IE kernels is used for acceleration of MoM interactions. The methods based on pre-corrections, e.g., PFFT and Adaptive Integral Method (AIM), project an arbitrary geometry onto a uniform FFT grid. This allows for the majority of MoM interactions to be computed quickly using FFT. Only a sparse version of the MoM system pre-correcting the erroneous near field contributions is then directly computed. The other class of algorithms performing rapid MoM solution of the IEs is the Multi-Level-Fast-Multipole-Algorithm (MLFMA) [2]. The latter relies on a hierarchical multipole decomposition of the MoM dense matrix interactions allowing for $O(N \log N)$ solution of the dense matrix equations.

While both the FFT and MLFMA techniques are highly scalable and able to make reliable working-approximations, there are numerous applications, such as high-performance filters, inductors, and capacitors for RF integrated circuits (RFIC) making the FFT-class algorithms as the solution of choice.

Received 2 November 2015, Accepted 28 November 2015, Scheduled 16 December 2015

Invited paper for the Commemorative Collection on the 150-Year Anniversary of Maxwell's Equations.

* Corresponding author: Brian Rautio (brian@sonnetsoftware.com).

¹ Sonnet Software, North Syracuse, NY, USA. ² University of Manitoba, Winnipeg, MB, Canada. ³ Syracuse University, Syracuse, NY, USA.

This is due to their broadband stability and ease in accounting for the multilayered substrates. In this work we propose a novel FFT-based algorithm, which alongside with the above capabilities offers the critical advantage of providing the maximum possible accuracy of the MoM solution for large planar circuits embedded in shielded multilayered media. The Galerkin MoM solution of the Electric Field Integral Equation (EFIE) utilizing roof-top basis functions on a rectangular elements implementation is taken as the reference solution in this work [3].

The new method is termed Unified-FFT-Grid-Totalizing (UFFT-GT) as it unifies the FFT-based matrix fill operations (MFO) and FFT-based matrix solve operations (MSO) both taking advantage of a MoM discretization on a single uniform grid as in Fig. 1. The method is capable of providing the $O(N \log N)$ MoM solution which is accurate to machine precision with respect to the traditional $O(N^3)$ MoM solver, Sonnet [3], with its associated demonstrably high accuracy [4]. The latter takes advantage of the rectangular waveguide Green's function allowing for an efficient way of computing the impedance matrix elements with machine precision [5]. The novelty of the proposed algorithm is in generalization of the previously known CG-FFT methodology [6–8] to the structures situated in a rectangular enclosure, which is further combined with an FFT-enhanced MFO accurate to machine precision for the circuit discretization. The MoM interactions are decomposed into four 2D convolution/correlation terms accounting for the walls of the rectangular enclosure and allowing for the use of FFT for their $O(N \log N)$ evaluation. Analogous formulations have been previously developed only for the FFT-based fast algorithms involving pre-corrections [5, 9]. Such formulations introduce additional error compared to the direct MoM solution making accurate resolution of weak but important interactions inaccurate. Unlike the prior pre-correction based algorithms, including the original version of the UFFT [5] henceforth referred to as UFFT-Precorrected (UFFT-P), the proposed method makes further improvement by eliminating the need for pre-correction due to MoM discretization on the uniform grid. Thus, no additional approximations versus the direct MoM solution (i.e., Sonnet, from which accuracy is only significantly limited by meshing) are made to the circuit geometry or to the dense system of equations solution. Effectively, the proposed FFT-based fast iterative algorithm features only a single error component compared to the direct MoM solution. This component is the error due to finite tolerance in the iterative solution of the matrix equation in GMRES [10] or CG algorithms [6]. However, with adequate preconditioning, it is shown that GMRES can perform very well and allow for convergence near or even equal to that imposed by machine precision.

A brief review of relevant and/or similar work can be found in [5]. In addition to those in [5], there is further noteworthy work. FFTs have been used to increase parallelization of MLFMA with the MLFMA-FFT method [11], and a similar application to GPU clusters [12]. Additionally, speed-enhanced direct solvers have been extended to finite-element analysis in a deterministic fashion [13]. Further, [9] has extended a similar FFT-based approach to cavities for scattering problems.

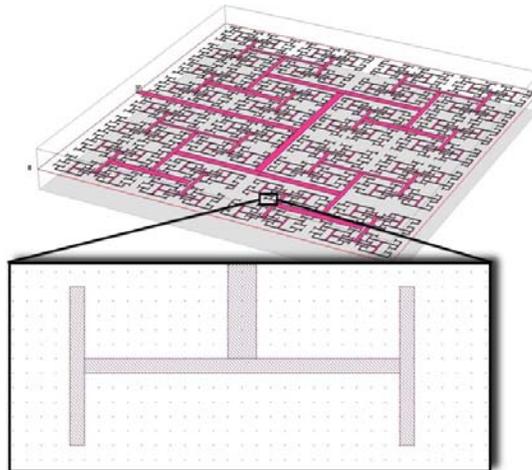


Figure 1. A 3D view of a circuit spanning the waveguide cross-section, featuring a zoomed view of uniform grid as it appears in Sonnet [3]. Note that UFFT-GT 'totalizes' the usage of the grid by using the same grid for both MFO and MSO.

2. IMPLICIT MATRIX SETUP AND STRUCTURE FOR MSO

A uniform grid is first overlain on the circuit geometry situated in the general multilayered medium transverse to the box formed by the side-walls of a rectangular enclosure, as seen in Fig. 1. The total of M_b cells across the height of the grid and N_b cells across the length span the cross-section of the dielectric interface supporting the metallization of the circuit. Note that grid dimensions may be defined arbitrarily in both dimensions, even for cells with wide aspect ratios.

This uniform grid is the same as in the MoM implementation of Sonnet's EM engine [3]. This grid is used for FFT-application to perform MFO to machine precision in $O(N \log N)$ operations as shown in [5]. The MoM impedance matrix fill and storage, however, in the proposed fast algorithm is considerably different from that of the conventional MoM.

2.1. Formulation of the DGF and MFO to Perform MSO

The electric field integral equation (EFIE) for this implementation of UFFT-GT starts in the form

$$\hat{\mathbf{t}} \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{\overline{\mathbf{G}}}(x, x', y, y') \cdot \overline{\overline{\mathbf{J}}}(x', y') dx' dy' = \hat{\mathbf{t}} \cdot \overline{\overline{\mathbf{E}}}_{inc}(x, y), \quad (1)$$

where $\hat{\mathbf{t}}$ is the tangential unit vector to the conductor surface, $x \in [0, a]$, $y \in [0, b]$, and $\overline{\overline{\mathbf{G}}}(x, x', y, y')$ is the dyadic Green's function (DGF) of the multilayered medium situated transversally to the axis of a rectangular waveguide [5]. The Green's function can be represented as a sum of four terms according to the image theory [14] as follows:

$$\begin{aligned} \overline{\overline{\mathbf{G}}}(x, x', y, y') &= \overline{\overline{\mathbf{G}}}^{--}(x - x', y - y') + \overline{\overline{\mathbf{G}}}^{-+}(x - x', y + y') + \overline{\overline{\mathbf{G}}}^{+-}(x + x', y - y') \\ &+ \overline{\overline{\mathbf{G}}}^{++}(x + x', y + y'). \end{aligned} \quad (2)$$

where the dyadic representation of $\overline{\overline{\mathbf{G}}}$ is

$$\overline{\overline{\mathbf{G}}} = \hat{\mathbf{x}}\hat{\mathbf{x}}G_{xx} + \hat{\mathbf{x}}\hat{\mathbf{y}}G_{xy} + \hat{\mathbf{y}}\hat{\mathbf{x}}G_{yx} + \hat{\mathbf{y}}\hat{\mathbf{y}}G_{yy} + \dots \quad (3)$$

noting that there are five terms involving z components not shown as they are unused and thus not calculated or stored. The current discretization is of the form,

$$\overline{\overline{\mathbf{J}}}(x', y') = \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} J_{m'n'}^x \overline{\overline{\mathbf{B}}}^x(x' - x_{m'}, y' - y_{n'}) + \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} J_{m'n'}^y \overline{\overline{\mathbf{B}}}^y(x' - x_{m'}, y' - y_{n'}), \quad (4)$$

with the circuit being discretized over the entire $M \times N$ box, where $J_{m'n'}$ is the weighted residual within MoM. Please note that this carries with it the implication that the discretization consists of elements which are both rectangular and conforming to a uniform grid. Non-rectangular basis functions cannot be processed with this version of UFFT-GT, although there might be potential to expand UFFT-P from [5] for the case of other discretizations and basis functions at the expense of performance and accuracy. While $J_{m'n'}$ is inherently zero at indices where there is no metallization, all possible basis functions are used at this level so as to leverage periodicity. Empty basis functions where there is no current, however, do not produce unknowns with respect to the overall solution. Combining Eq. (4) with Eq. (1), effectively applying our linear operator (EFIE) to the current discretization, yields a discretized form of the EFIE featuring a dyadic Green's function,

$$\hat{\mathbf{t}} \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{\overline{\mathbf{G}}} \cdot \left[\sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} (J_{m'n'}^x \overline{\overline{\mathbf{B}}}^x_{m'n'} + J_{m'n'}^y \overline{\overline{\mathbf{B}}}^y_{m'n'}) \right] dx' dy' = \hat{\mathbf{t}} \cdot \overline{\overline{\mathbf{E}}}_{inc}(x, y). \quad (5)$$

The MoM is implemented as a Galerkin method [15] so as to benefit from electromagnetic reciprocity, and applying Eq. (5) as an inner product with the same function used for basis function

results, for x and y , in

$$\begin{aligned}
& \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^x) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \left[\sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} (J_{m'n'}^x \overline{B}_{m'n'}^x + J_{m'n'}^y \overline{B}_{m'n'}^y) \right] dx' dy' dx dy \\
&= \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^x) \cdot \overline{E}_{inc}(x, y) dx dy \\
& \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^y) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \left[\sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} (J_{m'n'}^x \overline{B}_{m'n'}^x + J_{m'n'}^y \overline{B}_{m'n'}^y) \right] dx' dy' dx dy \\
&= \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^y) \cdot \overline{E}_{inc}(x, y) dx dy,
\end{aligned} \tag{6}$$

where the right-hand-side (RHS) of Eq. (6) reduces simply to the voltage V_{mn}^x and V_{mn}^y , respectively. In order to arrive at a linear operator that may be applied in the fashion of an impedance matrix, Eq. (6) is reordered as

$$\begin{aligned}
& \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} \left(J_{m'n'}^x \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^x) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \overline{B}_{mn}^x dx' dy' dx dy \right) \\
&+ \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} \left(J_{m'n'}^y \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^x) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \overline{B}_{mn}^y dx' dy' dx dy \right) = V_{mn}^x \\
& \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} \left(J_{m'n'}^x \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^y) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \overline{B}_{mn}^x dx' dy' dx dy \right) \\
&+ \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} \left(J_{m'n'}^y \int_{y=0}^b \int_{x=0}^a (\overline{B}_{mn}^y) \cdot \int_{y'=0}^b \int_{x'=0}^a \overline{G} \cdot \overline{B}_{mn}^y dx' dy' dx dy \right) = V_{mn}^y.
\end{aligned} \tag{7}$$

The integrals in Eq. (7) are more conveniently represented as impedance matrices, Z , as in

$$\begin{aligned}
& \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} [Z_{mnm'n'}^{xx} J_{m'n'}^x + Z_{mnm'n'}^{xy} J_{m'n'}^y] = V_{mn}^x \\
& \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} [Z_{mnm'n'}^{yx} J_{m'n'}^x + Z_{mnm'n'}^{yy} J_{m'n'}^y] = V_{mn}^y.
\end{aligned} \tag{8}$$

Due to decomposition of the Green's function according to Eq. (3) each of the xx , xy , yx , and yy blocks the above system of linear algebraic equations (SLAE) can be cast into the form of 2D correlations/convolutions as demonstrated below for the xx part of the SLAE:

$$\begin{aligned}
V_{mn}^{xx} &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m,n,m'n'}^{xx} J_{m'n'}^x = \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m-m',n-n'}^{xx} J_{m'n'}^x + \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m-m',n+n'}^{xx} J_{m'n'}^x \\
&+ \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m+m',n-n'}^{xx} J_{m'n'}^x + \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m+m',n+n'}^{xx} J_{m'n'}^x.
\end{aligned} \tag{9}$$

It is now beneficial to examine the terms for the impedance matrix in detail, from which the periodicity necessary to perform the proposed CG-FFT algorithm hails. As an example, we find elements which couple x -directed sources to x -directed observation points corresponding to the $G_{xx}(x-x', y-y')$ component of the dyadic Green's function,

$$\begin{aligned} Z_{m-m', n-n'}^{xx} = & \int_{y_n-\Delta y}^{y_n+\Delta y} \int_{x_m-\Delta x}^{x_m+\Delta x} (\overline{B}^x(x-x_m, y-y_n)) \\ & \cdot \int_{y_{n'}-\Delta y}^{y_{n'}+\Delta y} \int_{x_{m'}-\Delta x}^{x_{m'}+\Delta x} \overline{G}(x-x', y-y') \cdot \overline{B}^x(x'-x_{m'}, y'-y_{n'}) dx' dy' dx dy. \end{aligned} \quad (10)$$

Letting $\xi = x-x_m$, $\eta = y-y_m$, $\xi' = x'-x_m$, $\eta' = y'-y_m$, and $dx = d\xi$, $dy = d\eta$, $dx' = d\xi'$, $dy' = d\eta'$ (7) can be rewritten as

$$\begin{aligned} Z_{m-m', n-n'}^{xx} = & \int_{-\Delta y}^{\Delta y} \int_{-\Delta x}^{\Delta x} d\xi d\eta \overline{B}^x(\xi, \eta) \int_{-\Delta y}^{\Delta y} \int_{-\Delta x}^{\Delta x} d\xi' d\eta' \\ & \cdot \overline{G}[(\xi+x_m) - (\xi'+x_{m'}), (\eta-y_n) - (\eta'-y_{n'})] \cdot \overline{B}^x(\xi', \eta'), \end{aligned} \quad (11)$$

which is further rewritten so as to produce convolution-correlation pairs as

$$\begin{aligned} Z_{m-m', n-n'}^{xx} = & \int_{-\Delta y}^{\Delta y} \int_{-\Delta x}^{\Delta x} d\xi d\eta \overline{B}^x(\xi, \eta) \int_{-\Delta y}^{\Delta y} \int_{-\Delta x}^{\Delta x} d\xi' d\eta' \\ & \cdot \overline{G}[\xi - \xi' + \Delta x(m-m'), \eta - \eta' + \Delta y(n-n')] \overline{B}^x(\xi', \eta'), \end{aligned} \quad (12)$$

noting that calculation of these elements, the MFO, is accelerated by FFT as in [5]. With a complete $Z_{\mu\nu}^{xx}$ calculated and stored, it is conveniently possible to exploit the periodicity of $\xi = x-x_m$, $\eta = y-y_n$, $\xi' = x'-x_m$, and $\eta' = y'-y_n$ and as a result produce any element of $Z_{m-m', n-n'}^{xx}$ with three arithmetic operations and four memory gathers. Further, this process is only necessary for sparse preconditioner elements that need to be explicitly stored. The dense matrix need not be stored with the CG-FFT algorithm as the periodic storage of the data can further be exploited to accelerate matrix vector products via FFT. Note that this precludes the use of full matrix inverse with this technique; however, it does allow for the use of iterative methods. Note that while MFO includes only those operations before the iterative solver and thus matrix vector products begin, it is further relevant to describe parts of the matrix vector product for a better understanding of the structure of the implicit matrix.

To explain, first it is beneficial to demonstrate full $O(N_b^2 M_b^2)$ matrix vector product which explicitly loads each matrix element,

$$\begin{aligned} Z_{++}^{xx} * J_{\text{book}}^x &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m+m', n+n'}^{xx} J_{m'n'}^x \\ Z_{+-}^{xx} * J_{\text{book}}^x &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m+m', n-n'}^{xx} J_{m'n'}^x \\ Z_{-+}^{xx} * J_{\text{reverse}}^x &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m-m', n+n'}^{xx} J_{m'n'}^x \\ Z_{--}^{xx} * J_{\text{reverse}}^x &= \sum_{m'=0}^{M-1} \sum_{n'=0}^{N-1} Z_{m-m', n-n'}^{xx} J_{m'n'}^x, \end{aligned} \quad (13)$$

where Z_{--}^{xx} , Z_{+-}^{xx} , Z_{-+}^{xx} , and Z_{++}^{xx} are simply 1D re-orderings $Z_{(m-m', n-n')}^{xx}$, $Z_{(m+m', n-n')}^{xx}$, $Z_{(m-m', n+n')}^{xx}$, and $Z_{(m+m', n+n')}^{xx}$, respectively, with appropriate padding so as to preserve the Toeplitz and Hankel

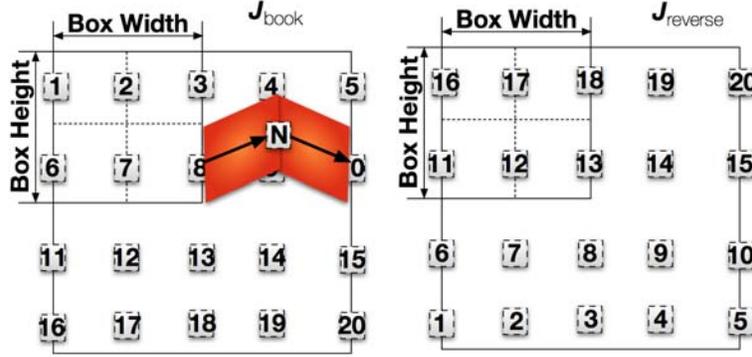


Figure 2. Indexing for the two 1D current arrays. Each number represents a rooftop basis function, which one pictured labeled N .

structures. J_{book}^x and J_{reverse}^x are both 1D re-orderings of $J_{(m'n')}^x$, where the former re-orders basis functions as though they are lines in a book, and the later organizes them as though the lines are reversed down the page, as shown in Fig. 2, which shows an ordering of the basis functions for an extremely coarse 2 by 2 grid with 6 potential basis function locations at indices 1, 2, 3, 6, 7, and 8. Each numbered rectangle represents the center of an x -directed rooftop basis function on the 2D grid, with the number being its associated index in each of the 1D current arrays. Note that the array spans twice the cross-section of the box to ensure Toeplitz and Hankel structure of the implicit matrix.

Matrix vector products are enhanced to $O(N_b M_b \log N_b M_b)$ by performing a series of 1D discrete convolutions through the use of digital signal processing [5], with the x source and x observation coupling producing,

$$V^{xx} = \text{IFFT} \left\{ \text{FFT} \left\{ Z_{++}^{xx} \right\} * \text{FFT} \left\{ J_{\text{book}}^x \right\} \right\} + \text{IFFT} \left\{ \text{FFT} \left\{ Z_{+-}^{xx} \right\} * \text{FFT} \left\{ J_{\text{reverse}}^x \right\} \right\} \\ + \text{IFFT} \left\{ \text{FFT} \left\{ Z_{-+}^{xx} \right\} * \text{FFT} \left\{ J_{\text{reverse}}^x \right\} \right\} + \text{IFFT} \left\{ \text{FFT} \left\{ Z_{--}^{xx} \right\} * \text{FFT} \left\{ J_{\text{book}}^x \right\} \right\}. \quad (14)$$

Using 1D orderings and FFTs is unconventional. Typically 2D orderings are used, which mandate four terms for J^x as opposed to two. The increased time to perform 1D FFTs is negated by the need to do fewer of them, whilst preserving memory. Note that, unlike UFFT-P in [5], the MFO is no longer split between a gridded set of point source basis and testing functions (which require the geometry projection and interpolation to use), and the set of full rectangular rooftop basis functions (necessary for pre-correction due to error from the projection and interpolation). With this work, it is made possible to fill the implicit matrix directly with full rectangular rooftop basis functions, negating the need for projection and interpolation. A near explicitly-defined matrix is still generated for pre-conditioning purposes; however, its elements are reconstructed from the implicit matrix as opposed to an additional fill.

2.2. Software Implementation

Rather than filling the traditional MoM for given circuits, the UFFT-GT method relies on the translational invariance property of the DGF previously noted. It is quite similar to the H -matrix based conventional MFO in Sonnet [5]; however, it is used to store the matrix in an implicit fashion. Indeed, all possible interactions within the totalized grid are represented indirectly, as opposed to just those for where metallization is present. By storing all interactions, (considerably more information than is present in even a full $O(N^2)$ Sonnet matrix), and separately storing each of the four convolution and correlation terms from the translational invariance of the DGF, one arrives at a set of large Toeplitz and Hankel matrices, which are well known to be of the form,

$$A_{(i,j)} = a_{(i-j)} \quad B_{(i,j)} = b_{(i+j)} \quad (15)$$

where A is a Toeplitz matrix and B is a Hankel matrix.

Calculating and storing all elements of these matrices would indeed yield far higher storage and computational requirements vs. a traditional MFO; however, as this abstraction has composed matrices into strictly Toeplitz and Hankel ordered structures, all matrices are both calculated and stored merely as a single row and column.

A total of $N_{\text{arrays}} = 4DP^2$ of these matrices, (stored, calculated, and henceforth referred to as ‘arrays’), are obtained, where D represents the number of tensor elements in the DGF, and P represents the number of planes. Note that for 2D problems, $D = 4$, representing x -directed source and x -directed observation basis interactions, x -directed source and y -directed observation basis interactions, y -directed source and x -directed observation basis interactions and y -directed source and y -directed observation basis interactions. It is also theoretically possible to make the reduction $D = 3$ by leveraging identities in [15]. Unfortunately, while this does reduce memory requirements, due to the way indexing is performed when translating current and voltage data into the Toeplitz/Hankel format (which must be done at every iteration of the simulation), this memory savings greatly increases computational requirements. These arrays are each of length

$$\begin{aligned} x_{\text{length}} &= 2M_b(2N_b + 1) \\ y_{\text{length}} &= 2N_b(2M_b + 1) \\ z_{\text{length}} &= 2M_b2N_b, \end{aligned} \quad (16)$$

where x , y , and z represent the direction of the observation basis function, and the overall lengths being theoretically irrespective of source basis function direction. In practice, the arrays stored are slightly longer and vary with a small amount with respect to source basis function direction; this is so as to allow padding for optimum performance accessing memory during FFTs.

3. ITERATIVE SOLUTION

It is not easily possible to perform a direct inverse of an implicitly stored matrix; however, it is quite convenient to perform matrix-vector multiples. As such, the Generalized Minimum Residual (GMRES) iterative method [4] is used to provide a solution in a fashion most similar to existing CG-FFT implementations, e.g., [16].

Note that although no traditional matrix elements (i.e., impedances) are directly calculated or stored, for pre-conditioning purposes they are available from the summation of four array elements. As such, a pre-conditioner is formed by solving a sparse matrix representation of geometrically close interactions in the circuit.

As the implicitly stored matrix is a set of Toeplitz and Hankel arrays encompassing all interactions possible on the grid, and the current array that is generated by GMRES is dense and represents only interactions between elements where metallization is present, a sparse transform of the current array is made. The Toeplitz and Hankel matrices are then converted to circulant form, which is shown in [17] to allow matrix vector multiply calculations with the use of FFT.

Thus, it is possible to compute the matrix vector multiples $V = Z_{\text{full}}I$ with implicitly stored matrix elements by calculating and summing

$$V_{(int.)} = \text{FFT}_n^{(-1)} (\text{FFT}_n(Z_{\text{circ}})\text{FFT}_n(J_{\text{circ}})), \quad (17)$$

for each of the $N_{\text{arrays}} = 4DP^2$ arrays, where Z_{circ} are circulant conversions of the matrices $Z_{(++)}$, $Z_{(+-)}$, $Z_{(-+)}$, or $Z_{(--)}$ from (14), and J_{circ} are padded versions of either J_{book} or J_{reverse} from (14). This process is repeated for a new full current vector I at every iteration of GMRES. Once all matrix-vector multiples are completed, the sparse representation of the voltage vector $V_{\text{intermediate}}$ is converted back to the dense representation which is numerically equivalent to V .

4. THE UFFT-GT IMPLEMENTATION

In this work, UFFT-GT is developed largely in MATLAB [18], using a modified prototype MFO from Sonnet, and the SonnetLab interface between MATLAB and Sonnet [19]. This is a similar style to [5] and [20]; however, it is a completely independent code base that this work benefits from both

the increased accuracy of the GT variety of UFFT and eliminating FORTRAN for convenience and portability. Note that this means that benchmarks of the algorithm do not benefit from compiled code in this work, though this is offset to some degree by built-in multithreading support. Additionally, this work benefits from further integration with the Sonnet framework, including integration with the current distribution viewer.

5. RESULTS AND DISCUSSIONS

To demonstrate the speed and accuracy of the UFFT-GT algorithm, a comprehensive suite of four examples is considered. These examples compare speed and accuracy between the developed UFFT-GT solver and one to three other solvers for each example. Note that results from the $O(N^3)$ direct solver Sonnet are considered to be the baseline, as 1) Sonnet makes the fewest approximations of all solvers as it is a direct solver with a DGF evaluated to full numerical precision, 2) Sonnet has seen extensive commercial use over an extended period of time and 3) Considerable work has been done to quantify the accuracy of Sonnet simulation [4].

5.1. Clock-Network Example

The clock-network example from [5], which can be seen in Fig. 1 and Fig. 3, is used to demonstrate the scaling ability of the UFFT-GT algorithm. As in [5], the fractal nature of the clock network is used to increase the complexity of the example. Each step includes either more H -tree elements, or, once a current size H -tree has filled the enclosure width and height, a new level of smaller H -tree is added and the grid and mesh are further refined. This allows for a realistic circuit to be used with increasing complexity in various meshing environments, as opposed to over-meshing of simple common geometries, e.g., spheres, as is commonly done.

It is readily observed from Fig. 4 that the $O(N \log N)$ scaling of UFFT-P is preserved. Indeed, with the removal of the projection and interpolation, performance is even improved vs. UFFT-P for many sizes. Note that data in this chart includes timing for the full solution with convergence to 10^{-12} in the case of UFFT-GT and 10^{-5} in the case of UFFT-P, as opposed to only data for a single matrix vector multiply as is often shown. Memory data includes the entire implicit matrix and full preconditioner before inversion. Up to twice the shown memory may be used in practice due to inefficiencies in GMRES and pre-conditioner banding algorithms; though this is not included in the data as it is a variable inefficiency of the implementation as opposed to the algorithm. The example is scaled up to 250 000 unknowns before reaching the 32 GB memory limitation of the system, which notably exceeds the 125 000 unknown capability of UFFT-P on the same system. The 250 000 unknown simulation took only 2.8 hours on an Intel i5-3570k Quad Core CPU, despite being written in interpreted MATLAB, which demonstrates the high efficiency of the algorithm.

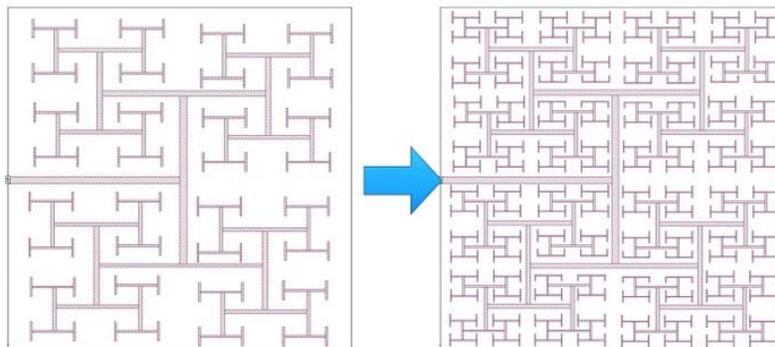


Figure 3. The geometry of the fractal clock network from [5]. It is convenient and realistic to increase the number of unknowns by increasing the number of H -tree elements and/or extending the fractalization.

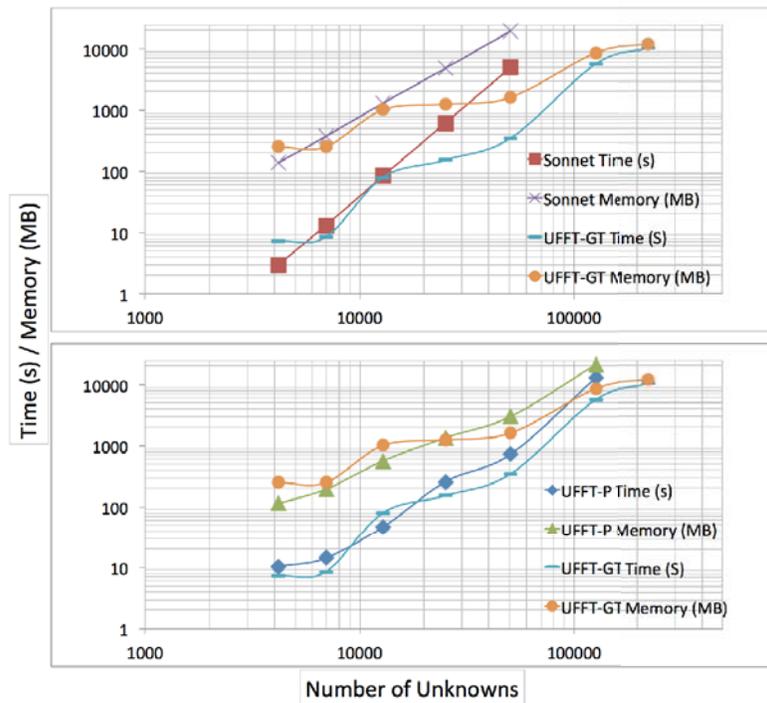


Figure 4. Time and memory scaling data for UFFT-P, UFFT-GT, and conventional Sonnet. Sonnet maxes out on the test system near 50 000, UFFT-P above 125 000, and UFFT-GT above 250 000.

It is also noteworthy in Fig. 4 that UFFT-GT does not exhibit linear behavior on the logarithmic scale as does Sonnet and to a lesser degree UFFT-P. This is because Sonnet scales with unknowns, while UFFT-GT scales with a combination of unknowns and density of metallization in the box. A given grid with more metallization will scale better; note that in (13) the summation is over the box as opposed to the unknowns. UFFT-P is slightly more linear in the log scale than UFFT-GT as this is mitigated to some degree by the projection, which scales with unknowns.

5.2. Interdigital Capacitor Example

To compare and contrast the accuracy and performance of UFFT-GT with a wide variety of tools, the interdigital capacitor example from [5] is used, which in turn is derived from [21]. The design is a three-terminal circuit that features two interdigital capacitors. It is shown in Fig. 5, where the blue triangle represents a 1 V delta-gap source being applied. Fig. 6 shows performance, where UFFT-GT and UFFT-P are similarly faster than Sonnet, which in turn is faster than MLFMA due to the more efficient surface meshing of only metallization vs. volume meshing of the entire structure. Moreover, while UFFT-GT is similar in performance to UFFT-P, note that this is while driving GMRES to a minimum residual of 10^{-12} in the GT variant vs. 10^{-5} in the *P* variant (a practical limit when considering the error introduced by projection and interpolation). With similar residual goals, UFFT-GT is approximately twice as fast for MSO with MFO and memory unchanged.

While the same algorithm is used for MFO for both Sonnet and UFFT-GT, UFFT-GT is able to perform MFO faster, despite producing arguably more complete data. UFFT-GT is filling ordered data for the entire uniform grid, and thus produces fewer cache misses (and thus fewer expensive calls to memory) vs. the gather-scatter nature of conventional MFO. Note that for MSO this is a preliminary time that does not benefit from compiled code or advanced pre-conditioners, meaning performance is expected to further improve moving forward. Memory requirements are also reduced due to the lack of projection and interpolation data.

Figure 7 shows *S*-parameter simulations between the four solvers. Note that while MLFMA shows

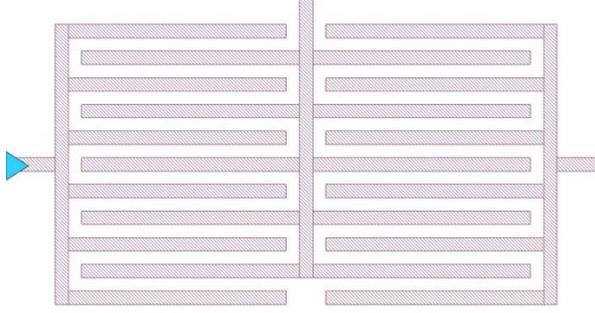


Figure 5. The interdigital filter example. Figure and circuit are taken from [5], which in turn bases the design from [21].

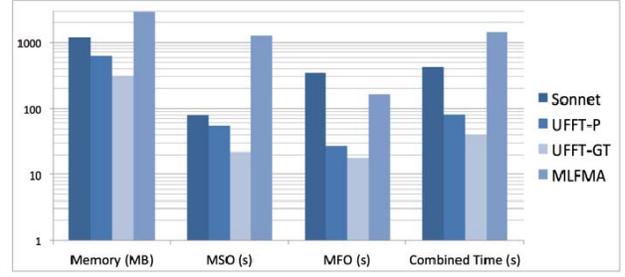


Figure 6. The logarithmic timing data from [5] for MSO, MFO, and overall time, as well as memory usage for Sonnet, UFFT, and MLFMA simulations of the interdigital capacitor geometry. UFFT-GT data has been added for comparison.

Table 1. Detailed S_{11} magnitude comparison.

Freq. (GHz)	UFFT-GT S_{11} (dB)	Sonnet S_{11} (dB)
100	-5.427606067	-5.427606067
105	-4.711806439	-4.711806439
108	-8.422235655	-8.422235655
110	-16.37773803	-16.37773803
112.3	-15.6987829	-15.6987829
117	-6.710118872	-6.710118872
125	-4.303694252	-4.303694252
132.5	-5.570591719	-5.570591719
140	-4.489905667	-4.489905667

Table 2. Detailed S_{11} phase comparison.

Freq. (GHz)	UFFT-GT S_{11} (Degree)	Sonnet S_{11} (Degree)
100	-60.00573797465	-60.00573797460
105	-113.8468205447	-113.8468205447
108	-149.0690098914	-149.0690098914
110	-178.7939190381	-178.7939190381
112.3	-23.91528562615	-23.91528562616
117	-81.98287649010	-81.98287649013
125	-107.2371265149	-107.2371265149
132.5	-110.8248949553	-110.8248949553
140	-155.1378164179	-155.1378164179

a clear difference from Sonnet results, in part due to the unshielded domain, UFFT-P is similar but still imprecise enough to distinguish at this level. UFFT-GT and Sonnet show no difference. Note that Sonnet is calculating a broad, adaptive band sweep (ABS) while all other solvers are calculating discrete points. To confirm this phenomenon, Table 1 and Table 2 show actual S -parameter values of high-precision touchstone files and confirms they are indeed identical to 12 to 13 significant figures, with the interesting exception of round-off differences.

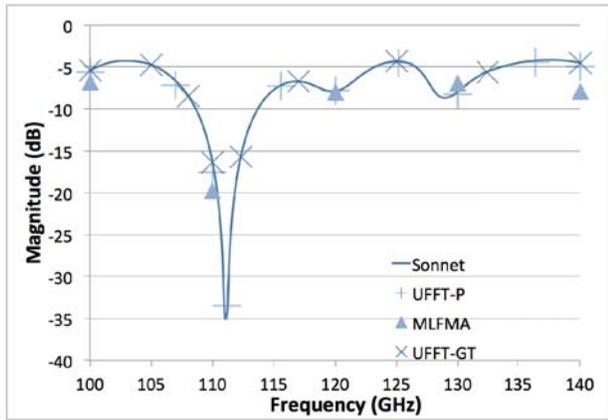


Figure 7. *S*-parameters of the capacitor from four different solvers. UFFT-GT data is added to those from [5]. Note that Sonnet is calculating a broad, adaptive band sweep (ABS) while others are calculating discrete points.

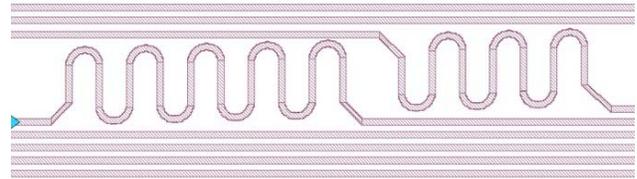


Figure 8. The meander line example geometry. A 1 V port excitation is applied at the blue triangle.

5.3. Meander Line Example

While *S*-parameters as presented in the previous example are a good benchmark for accuracy, essentially they represent accuracy on only the port subsection. It is further beneficial to examine current distributions, which show the accuracy on all subsections. Further, the previous examples have not featured curved geometries, which can prove difficult for improperly developed rectangular rooftop-basis solvers. As such, the meander line example, again from [5], is used, and is pictured in Fig. 8.

Entire current distributions are calculated for the circuit with both solvers, as is shown in Fig. 9. As they are visually indistinguishable, it is necessary to compare the current distributions numerically. The current distributions are loaded into MATLAB and the difference between each element is explicitly calculated. The maximum difference is found to be 7.456×10^{-13} , while the mean of all differences is calculated as 2.955×10^{-18} . Note that while general CG-FFT implementations are similar in terms of matrix vector multiply error, it has not previously been combined with FFT-enhanced MFO, nor has it been applied to the shielded domain for planar circuits. As the FFT-enhanced DGF and MFO are also calculated to machine precision, this type of fast and accurate convergence to machine precision

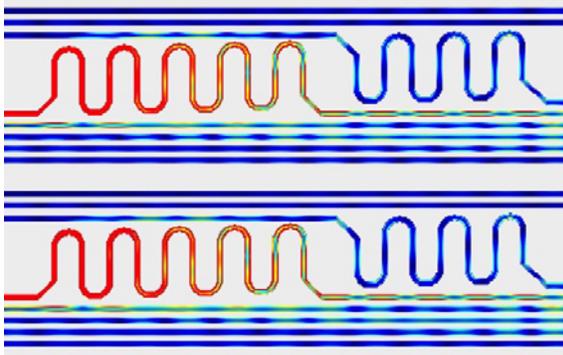


Figure 9. Meander current comparison, Sonnet (above), UFFT-GT (below). They are visually indistinguishable.

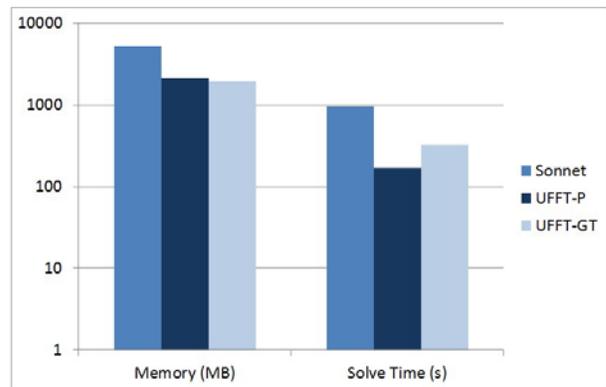


Figure 10. Logarithmic scale performance data of the meander circuit. Data from [5] with addition of UFFT-GT.

is unique. The above two extensions to CG-FFT and the resulting overall machine precision of the $O(N \log N)$ solution with respect to the direct MoM solution constitute the novel contributions of this work. In Fig. 10, performance data can be seen. Interestingly, UFFT-GT is more memory efficient than UFFT-P; however, its simulation time is longer. This is due to the interpreted code base of UFFT-GT vs. the compiled code of UFFT-P in combination with the low metallization content in a large box. Both fast solvers are markedly faster than Sonnet, despite the relatively low unknown count of 26 286.

5.4. Motherboard Memory Bus Example

Between the capacity to handle a large number of unknowns and the meshing efficiency of the MoM for shielded planar structures, UFFT-GT can handle circuit board geometries of unprecedented complexity. As there are few existing examples capable of pushing the limits of the solver without resorting to unnecessarily dense meshes, a new methodology is developed to reverse-engineer existing circuit geometries. Starting with the code-base developed in [22], a photograph of a computer motherboard is used to reverse-engineer the geometry of the memory to CPU bus — one of the highest performance parts of the board. The resulting geometry is shown in Fig. 11. Digital signal processing (DSP) is used in conjunction with SonnetLab to extract and draw polygons from the photograph.

The circuit covers approximately $4 \text{ cm} \times 4 \text{ cm}$ ($1.57 \text{ in} \times 1.57 \text{ in}$) and is simulated at 100 GHz, which represents the top end of a broadband sweep which would be necessary for transient analysis. It features 300 385 unknowns, and MSO is completed in an average time of 11.4 minutes per port in an average of 36 iterations per port with convergence to 10^{-12} or better, while consuming 6.8 GB of memory before solving the preconditioner. The calculated current distribution is shown in Fig. 12. At left shows the current with several lines excited, at right shows with only a single line excited.

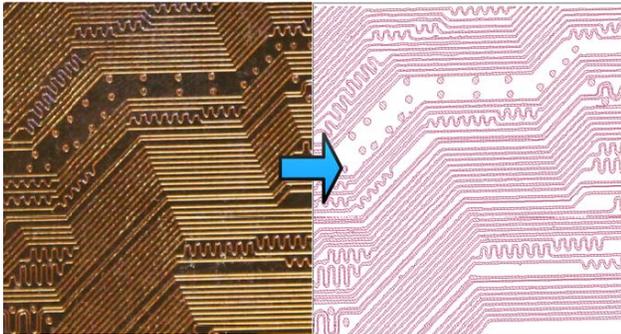


Figure 11. The motherboard circuit geometry as reverse engineered from a photograph.

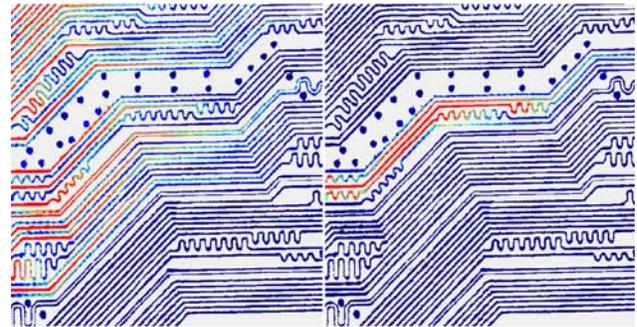


Figure 12. Current distribution of the motherboard memory bus example, with (a) several lines excited and (b) only a single line excited.

6. CONCLUSION

A new version of the fast UFFT solver, Grid Totalizing (GT), is developed and presented. It is tested and shown to provide machine precision accuracy with respect to the high-accuracy $O(N^3)$ tool Sonnet, while still performing analysis that scales with $O(N \log N)$. It is a unique methodology as it is the first to combine FFT-enhanced MFO which are calculated to machine precision with FFT-enhanced MSO that are also calculated to machine precision, allowing fast analysis that is essentially limited in accuracy only by the circuit discretization. The algorithm in essence is the CG-FFT fast algorithm with two novel modifications. The first modification is handling of MoM interactions with a machine-precision accurate rectangular waveguide Green's function through their decomposition into four convolution/correlation terms. Contribution from each of the four terms into the matrix-vector product allows for fast evaluation using FFTs. The second modification is fast evaluation of MoM implicit impedance matrix elements using FFT, which is made possible through grid conformal MoM discretization.

This speed and accuracy is demonstrated with a suite of four examples, as well as comparisons to two commercial solvers and a prototype UFFT-P solver. Future implementations of UFFT-GT should increase the performance by taking advantage of acceleration techniques such as compiled code and more sophisticated and robust pre-conditioning techniques.

ACKNOWLEDGMENT

This work was done as graduate research at Syracuse University. The authors would like to thank Sonnet Software, North Syracuse, NY for funding provided.

REFERENCES

1. Bleszynski, E., M. Bleszynski, and T. Jaroszewicz, "AIM: Adaptive integral method for solving largescale electromagnetic scattering and radiation problems," *Radio Science*, Vol. 31, No. 5, 1225–1251, Sep.–Oct. 1996.
2. Coifman, R., V. Rokhlin, and S. Wandzura, "The fast multipole method for the wave equation: A pedestrian prescription," *IEEE Antennas Propagat. Mag.*, Vol. 35, No. 3, 712, Jun. 1993.
3. Sonnet emSuite Version 14, Sonnet Software, North Syracuse, NY, 2013.
4. Rautio, J. C., "An ultra-high precision benchmark for validation of planar electromagnetic analyses," *IEEE Trans. on Microw. Theory Tech.*, Vol. 42, No. 11, 2046–2050, Nov. 1994.
5. Rautio, B. J., V. I. Okhmatovski, A. C. Cangellaris, J. C. Rautio, and J. K. Lee, "The unified-FFT algorithm for fast electromagnetic analysis of planar integrated circuits printed on layered media inside a rectangular enclosure," *IEEE Trans. on Microw. Theory Tech.*, Vol. 62, No. 5, 1112–1121, May 2014.
6. Catedra, M. F., R. F. Torres, J. Basterrechea, and E. Gago, *The CG-FFT Method — Application of Signal Processing Techniques to Electromagnetics*, Artech House, Norwood, MA, 1995.
7. Morsey, J. D., "Integral equation methodologies for the signal integrity analysis of PCB and interconnect structures in layered media from DC to multi-GHz frequencies," Ph.D. Dissertation, Clemson Univ., Clemson, SC., 2003.
8. Morsey, J. D., V. I. Okhmatovski, and A. C. Cangellaris, "Finite-thickness conductor models for full-wave analysis of interconnects with a fast integral equation method," *IEEE Adv. Packag.*, Vol. 27, No. 1, 24–33, Feb. 2004.
9. Yang, K. and A. E. Yilmaz, "An FFT-accelerated integral-equation solver for analyzing scattering in rectangular cavities," *IEEE Trans. on Microw. Theory Tech.*, Vol. 62, No. 9, 1930–1942, Sep. 2014.
10. Saad, Y., *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, 2003.
11. Taboada, J. M., M. G. Araujo, F. O. Basterio, J. L. Rodriguez, and L. Landesa, "MLFMA-FFT parallel algorithm for the solution of extremely large problems in electromagnetics," *Proc. of the IEEE*, Vol. 101, No. 2, 350–363, Jan. 2013.
12. Dang, V., Q. M. Nguyen, and O. Kilic, "GPU cluster implementation of FMM-FFT for large-scale electromagnetic problems," *IEEE Antennas Wireless Propag. Lett.*, Vol. 13, 1259–1262, Jun. 2014.
13. Sheng, F. and D. Jiao, "A deterministic-solution based fast eigenvalue solver with guaranteed convergence for finite-element based 3-D electromagnetic analysis," *IEEE Trans. Antennas Propag.*, Vol. 61, No. 7, 3701–3711, Jul. 2013.
14. Chew, W. C., J.-M. Jin, E. Michielssen, and J. Song, *Fast and Efficient Algorithms in Computational Electromagnetics*, Artech House, Norwood, MA, 2001.
15. Harrington, R. F., *Time-harmonic Electromagnetic Fields*, McGraw-Hill Co., New York, NY, 1961.
16. Smith, C. F., A. F. Peterson, and R. Mittra, "The biconjugate gradient method for electromagnetic scattering," *IEEE Trans. Antennas Propag.*, Vol. 38, No. 6, 938–940, Jun. 1990.
17. Golub, G. H. and C. F. van Loan, *Toeplitz and Related Systems, Matrix Computations*, 3rd Edition, Ch. 4, Sec. 7, 193–205, Johns Hopkins Univ. Press, Baltimore, MD, 1996.

18. MATLAB, MathWorks, Natick, MA, 2012.
19. SonnetLab Toolbox, Sonnet Software, North Syracuse, NY, 2013.
20. Rautio, B. J., V. I. Okhmatovski, and J. K. Lee, "Fast 3D planar electromagnetic analysis via unified-FFT method," *IEEE MTT-S Int. Symp. Dig.*, 1–3, Seattle, WA, Jun. 2–7, 2013.
21. Miyazawa, N., "Device having interdigital capacitor," U.S. Patent 6 949 811, Sep. 27, 2005.
22. Rautio, B. J., Q. Long, A. Agrawal, and M. A. El Sabbagh, "Simulation geometry rasterization for applications toward graphene interconnect characterization," *Proc. IEEE Intl. Symp. on Electromag. Compat*, 406–410, Pittsburgh, PA, Aug. 6–10, 2012.