Direct Finite-Element Solver of Linear Complexity for System-Level Signal and Power Integrity Co-Analysis

Bangda Zhou and Dan Jiao
School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, USA.

***Abstract*—In this paper, we develop a fast direct finite-element solver of linear (optimal) complexity for solving large-scale system-level signal and power integrity problems. The proposed direct solver has successfully analyzed an industry product-level full-package problem and correlated with measurements in time domain. The finite-element matrix of over 15.8 million unknowns resulting from the analysis of the package, including both signal lines and power delivery structures, is directly solved in less than 1.6 hours on a single core running at 3 GHz. Comparisons with the finite element methods that employ the most advanced direct sparse solvers have demonstrated the clear advantages of the proposed high-capacity linear-complexity direct solver.**

I. Introduction

An accurate system-level signal and power integrity analysis for high-speed and high-performance package design would call for a first-principles-based full-wave analysis of global electrical interactions among signal lines, power delivery network, analog/RF components, vias, inhomogeneous dielectrics, etc. in a setting of a complete package. The large scale of the underlying simulation problem, the broad band of the working frequencies, the irregularly shaped geometries, the inhomogeneous materials, and the large number of inputs and outputs, have all conspired to make the full-wave analysis of a full package extremely challenging. Existing fast full-wave solvers for solving large-scale problems are, in general, iterative solvers since traditional direct solvers are computationally expensive. The optimal complexity of an iterative solver is $O(N_{rhs}N_{it}N)$, where N_{rhs} is the number of right hand sides, N_{it} is the number of iterations, and N is the matrix size. To analyze the interaction among a large number of circuit ports and to perform many what-if analyses for an optimal design, the number of right hand sides is proportional to the port count and the number of what-if analyses. When the number of right hand sides is large, iterative solvers become inefficient. In contrast, a direct solver has a potential of achieving $O(N)$ complexity, which is optimal for solving N unknowns.

To analyze system-level signal and power integrity problems with first-principles based accuracy, the finite element method (FEM) is a popular method for choice because of its great capability in handling both complicated materials and geometries. A traditional direct finite element solver is computationally expensive. It is shown in [1] that the optimal operation count of a direct FEM solution in *exact* arithmetic is $O(N^{1.5})$ for 2-D problems, and $O(N^2)$ for 3-D problems. Although there have been successes in speeding up the direct finite element solution with state-of-the-art sparse matrix solvers [2], [3], [4], [5], these solvers have not accomplished $O(N)$ complexity, i. e. optimal complexity, for FEM-based direct solutions of general 3-D circuit problems.

In [6], a direct finite-element solver of linear (optimal) complexity is developed to extract broadband circuit parameters such as S-parameters of arbitrarily shaped 3-D circuits in inhomogeneous dielectrics. Both numerical and experimental results demonstrate a clear advantage of the liner-complexity direct solver as compared with existing solvers that employ a state-of-the-art direct sparse matrix solution [4] as well as a commercial iterative finite-element solver. A finite-element matrix from the analysis of a large-scale 3-D circuit in multiple dielectrics having 5.643 million unknowns is directly factorized in less than 2 hours on a single core running at 2.8 GHz. Linear complexity in both CPU time and memory consumption is achieved with prescribed accuracy satisfied.

In this work, with significant new algorithm developments, we accomplish a high-capacity direct finite-element solver of linear complexity that is capable of analyzing system-level signal and power integrity problems involving over 15 million unknowns on a single core. An IBM product-level package problem [7] having over 15.8 million unknowns is directly factorized and solved in less than 1.6 hours on a single 3 GHz CPU core. The signal integrity of the package interconnects in presence of the surrounding power delivery network is analyzed and correlated with the measurements in time domain. Good agreement is observed. We have also compared the proposed direct solver with a suite of state-of-the-art high-performance direct sparse solvers such as `MUMPS 4.10.0` [2], and `Pardiso` in Intel MKL [5]. It is shown that the proposed direct solver greatly outperforms state-of-the-art sparse solvers in both CPU time and memory consumption with desired accuracy satisfied.

II. Vector Finite Element Method and Mathematical Background

A. Vector Finite Element Method

Considering a general physical layout of a package or integrated circuit involving inhomogeneous materials and arbitrarily shaped lossy conductors, the electric field $\mathbf{E}$ satisfies

978-1-4799-5545-9/14/$31.00 ©2014 IEEE

the following second-order vector wave equation

$$\nabla \times (\frac{1}{\mu_r} \times \mathbf{E}) + jk_0\eta_0\sigma\mathbf{E} - k_0^2\varepsilon_r\mathbf{E} = -jk_0Z_0\mathbf{J}, \quad (1)$$

where μ_r is relative permeability, ε_r is relative permittivity, σ is conductivity, k_0 is free-space wave number, Z_0 is free-space wave impedance, and $\mathbf{J}$ is current density. A finite element based solution of (1) subject to pertinent boundary conditions results in the following linear system of equations

$$\mathbf{YX} = \mathbf{B}, \quad (2)$$

where $\mathbf{Y} \in \mathbb{C}^{N\times N}$ is a sparse matrix, and matrix $\mathbf{B}$ is composed of one or multiple right hand side vectors. When the size of (2) is large, its efficient solution relies on fast and large-scale matrix solutions.

B. Mathematical Background

In state-of-the-art direct sparse solvers, multifrontal method [4], [2] is a powerful algorithm. In this algorithm, the overall factorization of a sparse matrix is organized into a sequence of partial factorizations of smaller dense frontal matrices. Various ordering techniques have been adopted to reduce the number of fill-ins introduced during the direct matrix solution process. The computational cost of a multifrontal based solver depends on the number of nonzero elements in the $\mathbf{L}$ and $\mathbf{U}$ factors of the sparse matrix. In general, the complexity of a multifrontal solver is higher than linear (optimal) complexity.

Recently, it is proved in [3] that the sparse matrix resulting from a finite-element based analysis of electromagnetic problems can be represented by an $\mathcal{H}$-matrix [8] without any approximation, and the inverse as well as $\mathbf{L}$ and $\mathbf{U}$ of this sparse matrix has a data-sparse $\mathcal{H}$-matrix approximation with a controlled error. In an $\mathcal{H}$-matrix [8], the entire matrix is partitioned into multilevel admissible blocks and inadmissible blocks. The admissible blocks appear in the off-diagonal blocks, which describe the interaction between two separated sets of unknowns. An inadmissible block keeps its original full matrix form, while an admissible block is represented by a low-rank matrix

$$\tilde{\mathbf{Y}}^{t\times s} = \mathbf{A}_{\#t\times k}\mathbf{B}^T_{\#s\times k}, \quad (3)$$

where k is the rank that is smaller than the row and column dimension of the matrix block, and $\#$ denotes the cardinality of a set. The error of a rank-k approximation can be evaluated as [8]

$$\|\mathbf{Y}^{t\times s} - \tilde{\mathbf{Y}}^{t\times s}\|_2 = \sigma_{k+1}, \quad (4)$$

in which σ_{k+1} is the maximum singular value among truncated singular values. By applying (4), the error of an $\mathcal{H}$-matrix representation can be quantitatively controlled. With a rank-k representation, an $\mathcal{H}$-matrix significantly accelerates matrix operations and reduces storage cost for a prescribed accuracy as compared to a full-matrix based representation. It is shown in [3] that an $\mathcal{H}$-matrix based direct FEM solver has a complexity of $O(NlogN)$ in storage and a complexity of $O(Nlog^2N)$ in CPU time for solving general 3-D circuit problems.

III. Proposed Linear Complexity Direct FEM Solver for Large-Scale Signal and Power Integrity Analysis

A. Proposed Direct Solver

In the proposed solver, we fully take advantage of the zeros in the original FEM matrix, and also maximize the zeros in $\mathbf{L}$ and $\mathbf{U}$ by nested dissection ordering [1]. We store the nonzero blocks in $\mathbf{L}$ and $\mathbf{U}$ with a compact error-controlled $\mathcal{H}$-matrix representation, compute these nonzero blocks efficiently by developing fast $\mathcal{H}$-matrix based algorithms, while removing all the zeros in $\mathbf{L}$ and $\mathbf{U}$ from storage and computation. Moreover, we organize the factorization of the original 3-D finite element matrix into a sequence of factorizations of 2-D dense matrices, and thereby control the rank to follow a 2-D based growth rate, which is much slower than a 3-D based growth rate [9] for analyzing circuits operating at high frequencies. The overall algorithm has six major steps:

1. Build cluster tree $T_{\mathcal{I}}$ based on nested dissection
2. Build elimination tree $E_{\mathcal{I}}$ from $T_{\mathcal{I}}$
3. Obtain the boundary for each node in $E_{\mathcal{I}}$
4. Generate the $\mathcal{H}$-matrix structure for each node
5. Perform numerical factorization guided by $E_{\mathcal{I}}$ by new fast $\mathcal{H}$-matrix-based algorithms
6. Solve for one or multiple right hand sides based on $E_{\mathcal{I}}$

The steps of building the cluster tree $T_{\mathcal{I}}$ [8] and the elimination tree $E_{\mathcal{I}}$ [4] by the optimal nested dissection ordering restructure the original matrix and minimize the number of fill-ins introduced during factorization. To build cluster tree $T_{\mathcal{I}}$, we recursively partition a 3-D computational domain into *separators* and *subdomains*. At each level, a domain D is divided into one separator S and two subdomains $D_i (i = 1, 2)$, which is then further divided at next level. Since a separator S completely separates two subdomains D_1 and D_2, the off-diagonal blocks in the FEM matrix corresponding to the interaction between D_1 and D_2, denoted by $\mathbf{Y}_{D_1D_2}$ and $\mathbf{Y}_{D_2D_1}$, are zero. More important, the same zero blocks can be preserved in the $\mathbf{L}$ and $\mathbf{U}$ factors. Hence, $\mathbf{L}_{D_1D_2} = \mathbf{U}_{D_2D_1} = 0$. In the eliminaton tree $E_{\mathcal{I}}$, S becomes the direct parent of D_i. In the cluster tree, S and all D_i are the children of domain D, and S is ordered after all subdomains D_i, since we usually factorize the matrix in a left-to-right manner. As the recursive subdivision goes, the elimination tree as well as the cluster tree are recursively constructed. The process stops when the number of unknowns in each subdomain is no greater than $leafsize$ (a pre-determined constant). The final elimination tree is a multilevel tree with the bottom-level nodes being the subdomains having $leafsize$, and the nodes at upper levels comprising the surface separators of increasing size. The root node is the largest separator.

The LU factorization of the FEM matrix is a bottom-up traversal of the elimination tree as that used in the multifrontal algorithm. Note that the union of all the nodes in $E_{\mathcal{I}}$ is equal to $\mathcal{I}$, which is different from cluster tree $T_{\mathcal{I}}$. For each node s

in the elimination tree, we assemble a *frontal matrix* $\mathbf{F}_s$ from the system matrix $\mathbf{Y}$ and all the *updating matrices* $\mathbf{U}_c$ with $c \in E_s$ being s's children nodes. The F_s can be written as a 2×2 block matrix

$$\mathbf{F}_s = \begin{pmatrix} \mathbf{F}_{s,s} & \mathbf{F}_{s,\Phi_s} \\ \mathbf{F}_{\Phi_s,s} & \mathbf{F}_{\Phi_s,\Phi_s} \end{pmatrix} = \begin{pmatrix} \mathbf{Y}_{s,s} & \mathbf{Y}_{s,\Phi_s} \\ \mathbf{Y}_{\Phi_s,s} & 0 \end{pmatrix} + \bar{\mathbf{U}}_s, \quad (5)$$

in which Φ_s denotes the boundary of s, i.e. the minimum set of unknowns with which $\mathbf{L}_{\Phi_s,s}$ and $\mathbf{U}_{s,\Phi_s}$ are non-zero, and

$$\bar{\mathbf{U}}_s = \sum_{c \in E_s} \mathbf{U}_c. \quad (6)$$

We then apply partial LU factorization to $\mathbf{F}_s$, obtaining

$$\mathbf{F}_s = \begin{pmatrix} \mathbf{L}_{s,s} & 0 \\ \mathbf{L}_{\Phi_s,s} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{U}_{s,s} & \mathbf{U}_{s,\Phi_s} \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathbf{U}_s \end{pmatrix}. \quad (7)$$

Comparing (5) and (7), we can readily obtain the updating matrix $\mathbf{U}_s$

$$\mathbf{U}_s = \mathbf{F}_{\Phi_s,\Phi_s} - \mathbf{L}_{\Phi_s,s}\mathbf{U}_{s,\Phi_s}, \quad (8)$$

which is then used for the LU factorization of the frontal matrices of s's ancestors in $E_{\mathcal{I}}$. The entire LU factorization of matrix $\mathbf{Y}$ is a bottom-up or post-order traversal of $E_{\mathcal{I}}$. For Φ_s, we have an important lemma,

$$\Phi_s \subseteq \bigcup_{c \in E_s} \Phi_c. \quad (9)$$

Therefore, minimizing $\#\{\Phi_s\}$, the size of Φ_s, is critical in avoiding unnecessary operations on zeros. We thus first perform *symbolic factorization* to pre-process the elimination tree to compute the minimum Φ_s for each node in $E_{\mathcal{I}}$ before the real factorization is carried out.

The frontal matrix is a dense matrix of size $\#node + \#boundary$. Since each non-leaf node in the elimination tree is a 2-D surface separator, the boundary of each node is essentially the union of the unknowns residing on the bounding box of the 2-D separator. Thus, the boundary size is proportional to the node size, which is 2-D. We hence reduce the factorization of the original large 3-D FEM matrix to a sequence of factorizations of 2-D dense matrices shown in (5). We then develop efficient $\mathcal{H}$-matrix based algorithms to accelerate the dense matrix computation. The $\mathcal{H}$-matrix representations are built for $\mathbf{F}_{s,s}$, $\mathbf{F}_{\Phi_s,s}$, and $\mathbf{F}_{s,\Phi_s}$. However, $\mathbf{F}_{\Phi_s,\Phi_s}$ contains reference $\mathcal{H}$-matrices from its parent in the following form

$$\begin{aligned} \mathbf{U}_s &= \mathbf{F}_{\Phi_s,\Phi_s} \\ &= \bigcup_{p,q} \{\mathbf{F}_{p,q} | \exists u \in \Phi_s, \exists v \in \Phi_s, u \in p \text{ and } v \in q\}. \end{aligned} \quad (10)$$

Thus, $\mathbf{F}_{\Phi_s,\Phi_s}$ does not share the same row or column cluster tree with $\mathbf{F}_{\Phi_s,s}$ or $\mathbf{F}_{s,\Phi_s}$. We hence develop new $\mathcal{H}$-matrix arithmetic for incompatible $\mathcal{H}$-matrix structures to efficiently compute the updating matrix in (8). In general, $\#\{\Phi_s\}$ is larger than $\#\{s\}$. Using the conventional $\mathcal{H}$-matrix construction method would generate skewed blocks, which could affect sparsity constant C_{sp} as well as the computational efficiency. We thus develop an adaptive construction scheme that partitions the cluster with larger unknowns first instead of partitioning the row and column clusters at the same time. These modifications ensure the efficient logarithmic $\mathcal{H}$-matrix operations, and ultimately the linear complexity of the proposed direct solver.

B. Complexity and Accuracy Analysis

Considering a 3-D computational domain, the unknown number of which along each direction is n. The total unknown number is thus $N = n^3$. Denoting the depth of the elimination tree by L, we have $L = log_2 n$ and hence, $N = 8^L$. Let the root level be $l = L$. At level l of the elimination tree, there are 8^{L-l} nodes, and the matrix size of each node is $m = O(2^l \times 2^l)$ because a node at level l is either a 2-D surface separator or a leaf domain. Since the operations associated with each node are performed in an $\mathcal{H}$-based fast algorithm, the computational cost of each node of size m is significantly reduced from m^3 to $r_l^2 m log^2 m$ [3], where r_l denotes the rank at the l-th level. As a result, we obtain

$$\text{Time Complexity} = \sum_{l=1}^{L} 8^{(L-l)} r_l^2 (2^l \times 2^l) log_2^2 (2^l \times 2^l). \quad (11)$$

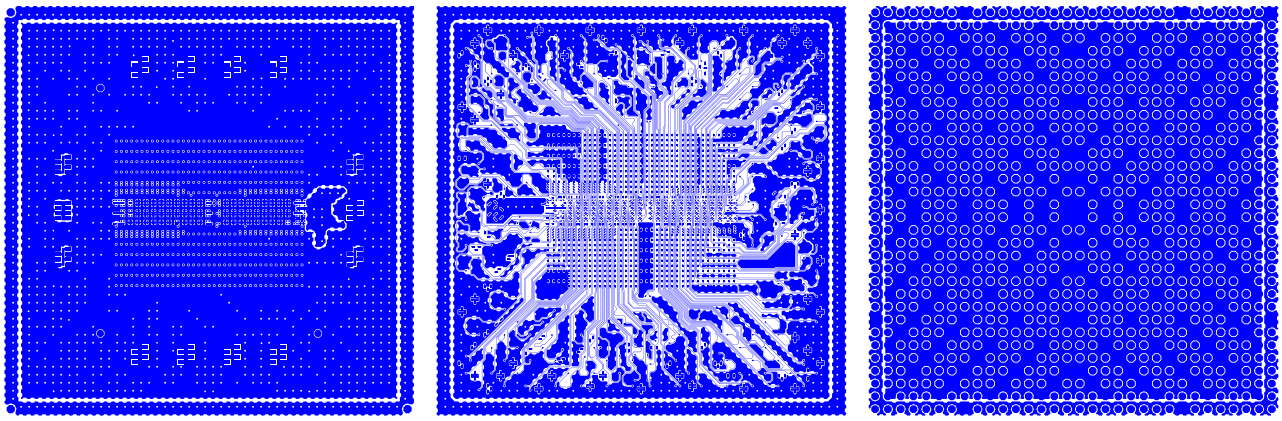
For circuits whose electric size is small, the rank r_l is a bounded constant for achieving any prescribed accuracy. For electrically large problems, i.e. high-frequency problems, the rank for prescribed accuracy is frequency dependent, thus tree-level dependent. Since each node is a 2-D surface separator, the rank r_l follows the 2-D based growth rate, thus it is proportional to the square root of the logarithm of the electrical size of the 2-D surface [9], and hence $r_l = rank_{2D} = O(\sqrt{log 2^l}) = O(\sqrt{l})$. Substituting it into (11), we obtain

$$\text{Time Complexity} = 8^L \sum_{l=1}^{L} l(2l)^2 / 2^l = O(N), \quad (12)$$

which is linear. The last equality holds true because the denominator grows with l much faster than the numerator. Similarly, the storage complexity of the proposed solver can be proved to be $O(N)$. In the proposed solver, we represent the intermediate dense matrices associated with each node (a leaf domain or a separator) in the elimination tree by an $\mathcal{H}$-matrix. The accuracy of such an $\mathcal{H}$-matrix representation can be proved from the fact that the original FEM matrix has an exact $\mathcal{H}$-representation and its inverse has an error-bounded $\mathcal{H}$-representation [3].

IV. Simulation of System-Level Signal and Power Integrity Problems

An IBM product-level package [7] is simulated from 100 MHz to 50 GHz to examine the capability of the proposed direct finite element solver in co-analyzing system-level signal and power integrity. The package involves 92 k unique elements such as lines, pins, and shapes. There are eight metal layers and seven dielectric layers. The layout of selected layers



(a) Layer 0 (b) Layer 2 (c) Layer 14

Fig. 1. Layout of a product-level package in different layers (Courtesy of Dr. Jason Morsey).

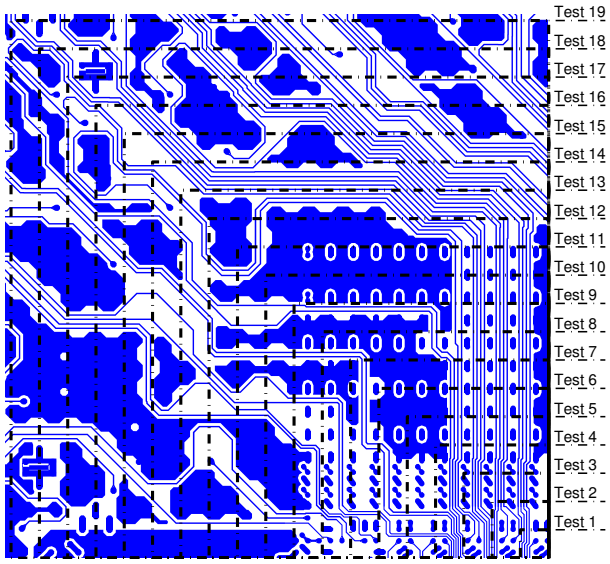


Fig. 2. Nineteen structures generated for solver performance verification.

can be seen from Fig. 1, in which the blue region stands for the metal region whereas the white region is occupied by the dielectric materials. The layer with an even number is a metal layer, while the odd numbered layers only contain dielectrics and pins to connect different metal layers.

We first develop a module to interpret the board file of the full IBM package into the geometrical and material data that can be recognized by the proposed solver. We then mesh the entire layout of the package into triangular prism elements. Arches in the package design are transformed into polygons by predefined angle parameters.

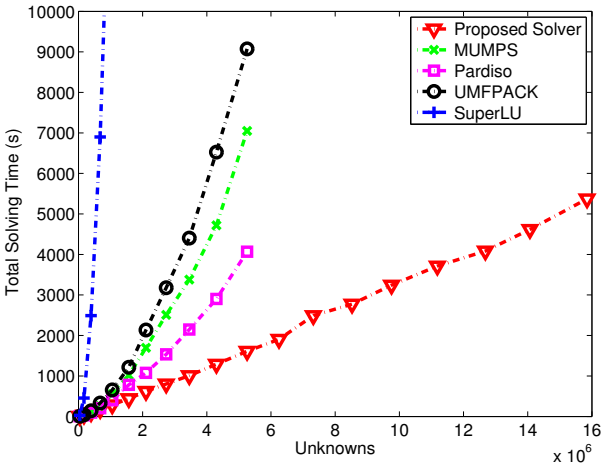
A. Complexity and Performance Verification

To examine the accuracy and efficiency of the proposed direct solver, we first simulate a suite of nineteen substructures of the full package. These substructures are illustrated in Fig. 2. The smallest structure occupies a package area of 500 μm in width, 500 μm in length, and 4 layer in thickness, whereas the largest structure occupies a package area of 9500 μm by 9500 μm. The resultant number of unknowns ranges from 31,276 to 15,850,600. Two ports at the topmost layer are excited. The important simulation parameters used in the proposed direct solver are $leafsize = 8$ and truncation error $\epsilon = 10^{-6}$. The computer used has a single core running at 3 GHz with 64 GB memory.

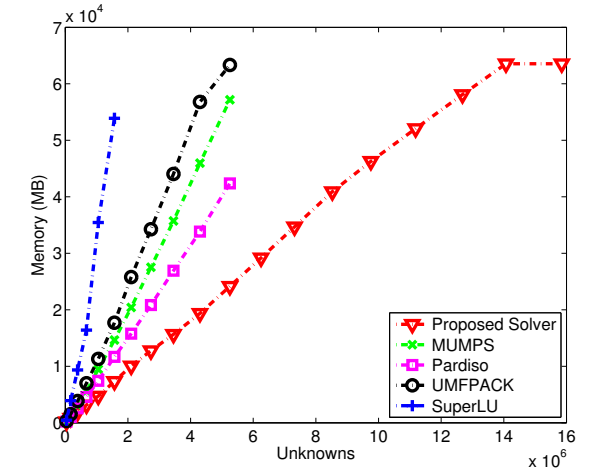
The CPU time and memory cost of the proposed solver with respect to unknown number N are shown in Fig. 3 in comparison with those of the direct finite element solver that employs the most advanced direct sparse solvers provided by `SuperLU 4.3`, `UMFPACK 5.6.2` [4], `MUMPS 4.10.0` [2], and `Pardiso` in Intel MKL [5]. The largest number of unknowns associated with each curve in Fig. 3 is the largest one that can be simulated by the specific solver on the given computing platform. It is evident that the proposed direct finite element solver greatly outperforms the other state-of-the-art direct solvers in both CPU time and memory consumption. More important, the proposed direct solver demonstrates a clear linear complexity in both time and memory across the entire unknown range, whereas the complexity of the other direct solvers is much higher. With the optimal linear complexity achieved, the proposed direct solver is able to solve the extremely large 15.8 million unknown case using less than 1.6 hours on a single core. The solution error of the proposed direct solver, measured in relative residual $||\mathbf{YX}-\mathbf{B}||/||\mathbf{B}||$, is plotted in Fig. 3(c) for all the testing cases. Excellent accuracy is observed across the entire unknown range. The relative residual of the other direct solvers is not plotted because they are based on exact arithmetic without any approximation.

B. Signal and Power Integrity Analysis and Correlation with Measurements

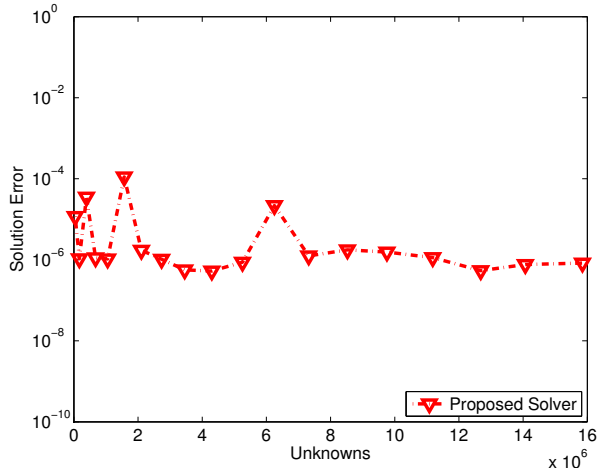
With the accuracy and efficiency of the proposed direct solver validated, next, we analyze the signal and power integrity of the IBM product-level package and correlate our analysis with measured data. First, the frequency-domain S-



(a) Factorization time complexity

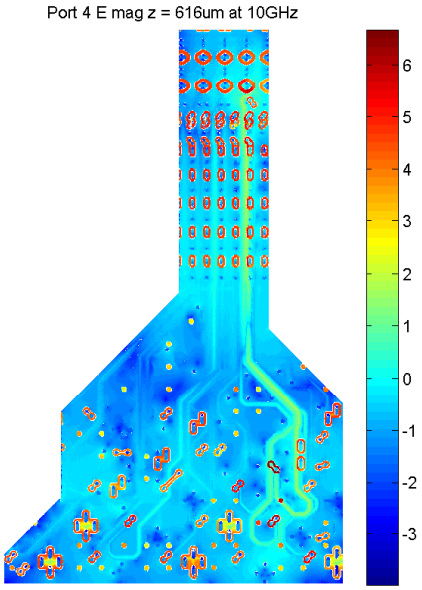


(b) Memory complexity

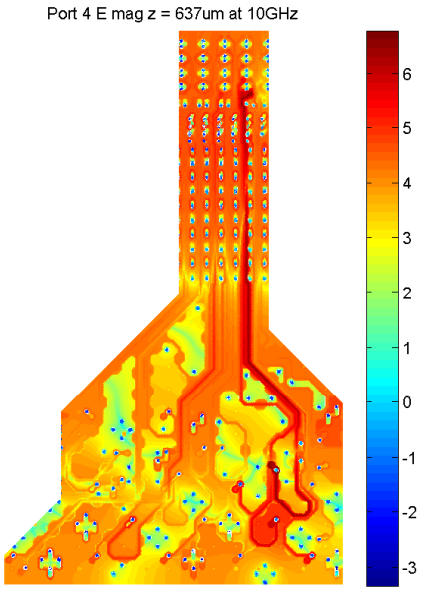


(c) Solution error

Fig. 3. Complexity and performance verification of the proposed direct solver. (a) Time Complexity. (b) Memory Complexity. (c) Solution Error (defined as relative residual $||\mathbf{YX} - \mathbf{B}||/||\mathbf{B}||$).



(a) Layer 1

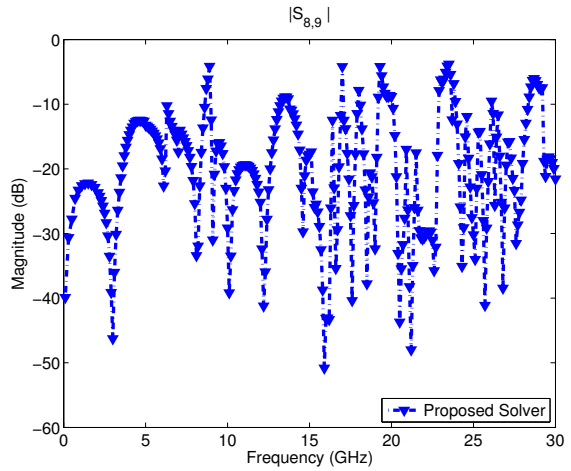


(b) Layer 2

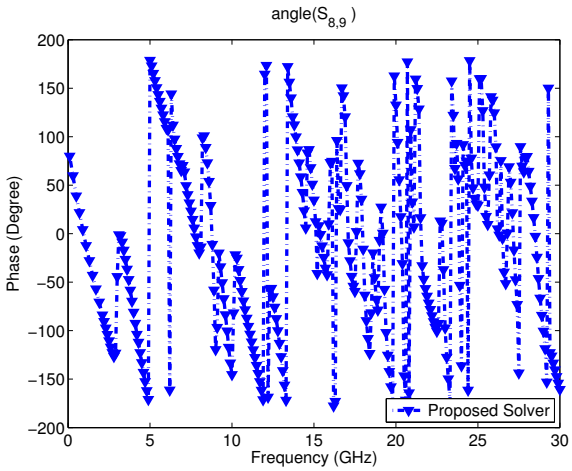
Fig. 4. Electric field distribution in log scale with port 4 excited at 10 GHz in layer 2.(a) Layer 1. (b) Layer 2.

parameters from 100 MHz to 30 GHz were generated for 16 ports assigned to the 4 interconnects located in the full package. The near-end ports of the interconnects are placed on the topmost layer (chip side), while the far-end ports are at the bottom-most layer (BGA side). The entire stack of 8 metal layers and 7 inter-layer dielectrics are simulated. The resultant number of unknowns is 3,149,880.

The proposed direct solver only takes less than 3.3 hours and 29 GB peak memory at each frequency on a single core



(a) Magnitude



(b) Phase

Fig. 5. S-parameters measured at the input and the output ports versus frequency. (a) Magnitude. (b) Phase.

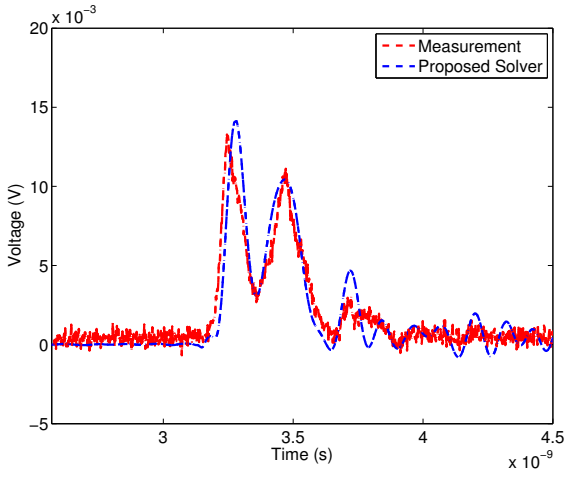


Fig. 6. Time-domain correlation with full-package measurements.

running at 2.33 GHz to extract the 16 by 16 S-parameter matrix, i.e. solutions for 16 right hand sides. The electric field distributions at 10 GHz in the region close to the 20 lines in two layers, with port 4 excited and other ports left open, are displayed in Fig. 4. The interaction between signal lines and power/ground planes can be clearly seen. In the measurement setup, the near end of line 6 located on the chip side is excited by a step function, while the timed-domain voltage is measured at the far end of line 2 located at the bottom BGA side. To correlate with the measurements, we perform the time-domain simulation based on the same setup. The frequency-domain crosstalk between the near end of line 6 (port 8) and the far end of line 2 (port 9) with all the other ports left open is plotted in Fig. 5 from 100 MHz to 30 GHz. No measured S-parameters are available for comparison since the measurements are only performed in time domain [7]. However, we are able to correlate the time-domain result obtained from the computed frequency-domain crosstalk with the measured data. As shown in Fig. 6, very good agreement is observed.

V. Conclusion

In this paper, we develop a high-capacity linear-complexity direct finite element solver to co-analyze system-level signal and power integrity problems. It successfully simulated a large-scale IBM product-level package problem involving over 15.8 million unknowns in less than 1.6 hours on a single core running at 3 GHz. Comparisons with measurements and the state-of-the-art direct sparse solvers have demonstrated the superior performance of the proposed direct solver in accuracy, efficiency, and capacity.

VI. Acknowledgments

This work was supported by a grant from NSF under award No. 0747578 and a grant from SRC (Task 1292.073). The authors would like to thank Dr. Jason Morsey at IBM Corporation for providing package structures and measured data.

References

[1] A. George, "Nested dissection of a regular finite element mesh," *SIAM Journal on Numerical Analysis*, vol. 10, no. 2, pp. 345–363, 1973.
[2] P. R. Amestoy, I. S. Duff, and J.-Y. L'Excellent, "Multifrontal parallel distributed symmetric and unsymmetric solvers," *Computer methods in applied mechanics and engineering*, vol. 184, no. 2, pp. 501–520, 2000.
[3] H. Liu and D. Jiao, "Existence of $\mathcal{H}$-matrix representations of the inverse finite-element matrix of electrodynamic problems and-based fast direct finite-element solvers," *IEEE Trans. MTT*, vol. 58, no. 12, p. 3697, 2010.
[4] T. A. Davis, "Algorithm 832: Umfpack v4. 3—an unsymmetric-pattern multifrontal method," *ACM Transactions on Mathematical Software (TOMS)*, vol. 30, no. 2, pp. 196–199, 2004.
[5] Intel, "Intel math kernel library 12.0.0," 2011.
[6] B. Zhou, H. Liu, and D. Jiao, "A direct finite element solver of linear complexity for large-scale 3-d circuit extraction in multiple dielectrics," in *ACM/IEEE 50th Design Automation Conference (DAC)*, 2013, p. 140.
[7] J. Morsey, *IBM Packaging Structure*, 2013.
[8] S. Börm, L. Grasedyck, and W. Hackbusch, "Hierarchical matrices," *Lecture notes*, vol. 21, 2003.
[9] H. Liu and D. Jiao, "A theoretical study on the rank's dependence with electric size of the inverse finite element matrix for large-scale electrodynamic analysis," *IEEE Int. Symp. Ant. Propagat.*, 2012.