# Direct Matrix Solution of Linear Complexity for Surface Integral-Equation Based Impedance Extraction of High Bandwidth Interconnects

Wenwen Chai and Dan Jiao

School of Electrical and Computer Engineering, Purdue University, 465 Northwestern Avenue, West Lafayette, IN 47907, USA (phone: 765-494-5240; fax: 765-494-3371; e-mails: {wchai, djiao}@purdue.edu)

## ABSTRACT

A linear-complexity direct matrix solution is developed for the surface-integral based impedance extraction of arbitrarily-shaped 3-D non-ideal conductors embedded in dielectric materials. It outperforms state-of-the-art impedance solvers with fast CPU-time, modest memory-consumption, and without sacrificing accuracy. The inverse of a 2.6-million-unknown matrix arising from the extraction of large-scale 3-D interconnects was obtained in 1.5 GB memory and 1.3 hours on a 3 GHz CPU.

## Categories and Subject Descriptors

B.7.2 **[Integrating Circuits]:** Design Aids - Simulation, Verification

## General Terms

Algorithms

## Keywords

Impedance extraction, interconnect, fast integral equation solvers, direct solvers.

## 1. INTRODUCTION

With the increase in the processing power of the CPU, the memory and system interconnect links connected to a CPU need to have an exponentially increased bandwidth in order to fully utilize the computing power. This leads to higher speed signals on each data line as well as an increase in the number of data lines. It also becomes necessary to move chips closer to each other by revolutionary technologies such as 3-D stacking via TSVs etc. Enabling higher bandwidth brings significant challenges to the analysis and design of interconnects. To address these challenges, a full-wave modeling technology is required that can rapidly characterize the interaction between a large number of I/Os in the face of large problem sizes.

Existing fast full-wave solvers for solving large-scale problems are, in general, iterative solvers [1-4] 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 between I/Os, the number of right hand sides is proportional to the I/O count. Clearly, when the number of I/Os is large, iterative solvers become inefficient since an entire iteration procedure has to be repeated for each port. There has been much recent progress in direct solvers [5-8]. In [6], an $H^2$-matrix based mathematical framework [11-12] was introduced and further developed to reduce the computational complexity of direct matrix solutions. It is shown that a dense matrix of size $N$ can be inverted in $O(N)$ complexity for capacitance extraction involving arbitrary geometry and non-uniform materials. However, the impedance extraction developed in [6] was only for ideal conductors in a uniform material. To the best of our knowledge, the impedance extraction of realistic interconnects involving 3-D non-ideal conductors and non-uniform materials has not been accomplished with a linear-complexity direct solution.

The contribution of this paper is the development of a linear-complexity direct solution for the surface integral equation based impedance extraction involving arbitrarily shaped 3-D non-ideal conductors. A surface integral formulation is attractive for impedance extraction compared to a volume integral formulation since the number of unknowns is greatly reduced. The surface integral formulation used in this work is based on [9]. This formulation is shown to be accurate and robust over a broad band of frequencies. The resultant system matrix is composed of both dense and sparse matrix blocks. Some of these blocks are not square matrices either. Although such a complicated matrix pattern does not create an additional challenge to the fast computation of a matrix-vector multiplication, it does render the fast computation of a matrix inverse or LU particularly challenging. The unknowns solved by the new surface integral formulation in [9] are tangential **E**, tangential **H**, scalar potential, and charge density on the conducting surfaces. The entire system matrix cannot be represented as one $H^2$ matrix. The same is true for its inverse. As a result, the method developed in [6] is not directly applicable to the impedance extraction concerned in this work. In the following sections, we demonstrate how to overcome the numerical challenge of directly solving a highly irregular system that is mixed with both dense and sparse blocks, achieving a linear-complexity direct solution for the impedance extraction of general 3-D structures with lossy conductors under the $H^2$-matrix based mathematical framework.

## 2. BACKGROUND

### 2.1 Surface Integral Formulation

Consider a union of conductors of finite conductivity $\sigma$ immersed in a dielectric material characterized by permittivity $\mu$ and permeability $\varepsilon$, we employ the surface integral-equation based formulation derived in [9] to extract the impedance of the

—This work was supported by a grant from SRC (Task 1292.073), and NSF under award No. 0747578 and No. 0702567.


DAC'11, June 5-10, 2011, San Diego, California, USA

structure in a broad band of frequencies. The formulation comprises the following five equations:

$$-\frac{1}{2}\vec{E}(\vec{r}) = Z_C \cdot \left[ \frac{j}{K_C} \iint_S ds' \, \nabla' \cdot (\hat{n} \times \vec{H}) \nabla G_0 + jK_C \iint_S ds' (\hat{n} \times \vec{H}) G_0 \right] + \iint_S ds' (\hat{n} \times \vec{E}) \times \nabla G_0 + \nabla \varphi(\vec{r}) \quad (1)$$

$$\frac{1}{2}\vec{H}(\vec{r}) = -\left[ \frac{j}{K_C} \iint_S ds' \nabla' \cdot (\hat{n} \times \vec{E}) \nabla G_1 + jK_C \iint_S ds' (\hat{n} \times \vec{E}) G_1 \right] / Z_C + \iint_S ds' (\hat{n} \times \vec{H}) \times \nabla G_1 \quad (2)$$

$$\varphi(\vec{r}) = \iint_S G_0 \rho(\vec{r}') / \varepsilon \, ds' \quad (3)$$

$$\nabla \cdot (n \times H)(\vec{r}_{nc}) = K_C{}^2 \rho(\vec{r}_{nc}) / \sigma\mu \quad (4)$$

$$\varphi(\vec{r}_c) = \psi_C \,, \quad (5)$$

where $\vec{E}$ is electric field intensity, $\vec{H}$ is magnetic field intensity, $\hat{n}$ is the unit vector normal to the conductor surface and pointing away from the conductor, $G_0$ and $G_1$ are full-wave Green's functions in the background material, and the conducting region, respectively. They are given by

$$G_0(\vec{r}, \vec{r}') = e^{jk|\vec{r}-\vec{r}'|} / (4\pi |\vec{r} - \vec{r}'|), \quad G_1(\vec{r}, \vec{r}') = e^{jK_C|\vec{r}-\vec{r}'|} / (4\pi |\vec{r} - \vec{r}'|), \quad (6)$$

with $k = \omega\sqrt{\mu\varepsilon}$, $K_C = \sqrt{\omega^2\mu\varepsilon - j\omega\mu\sigma}$, and $\omega$ is the angular frequency. In (1-5), $Z_C = \omega\mu / K_c$, $\vec{r}$ denotes an observation point, $\vec{r}\,'$ denotes a source point, $S$ is the conducting surface, $\varphi$ is electric scalar potential, $\rho$ is charge density, $\vec{r}_c$ denotes a point on the contact surfaces where the voltage source $\psi_C$ is supplied, and $\vec{r}_{nc}$ denotes a point on the non-contact surface. The first equation is an electric field integral equation (EFIE) that describes the interaction of equivalent electric and magnetic currents on the conductor surfaces via the background material; the second integral equation is a magnetic field integral equation formulated inside each conductor. The unknowns involved in (1-5) are $\hat{n} \times \vec{E}$, $\hat{n} \times \vec{H}$, $\varphi$, and $\rho$ on the conducting surfaces. After $\hat{n} \times \vec{H}$ is solved, we can use it to compute the current and the impedance of the conductor as follows.

$$I = \sigma \iint_{Sc} ds E_n = -\sigma / (\sigma + j\omega\varepsilon_0) \iint_{Sc} ds \nabla \cdot (\hat{n} \times \vec{H}) \text{ , and } Z = V/I. \quad (7)$$

## 2.2 $H^2$-matrix Framework

An $H^2$ matrix is generally associated with a strong admissibility condition [11, pp. 145]. Denoting the full index set of all the basis functions by $\mathcal{I} = \{1, 2, \ldots, N\}$, where $N$ is the total number of basis functions, consider two subsets $t$ and $s$ of the $\mathcal{I}$, the strong admissibility condition is defined as

$$\max\{diam(\Omega_t), diam(\Omega_s)\} \le \eta \, dist(\Omega_t, \Omega_s), \quad (8)$$

where $\Omega_t$ and $\Omega_s$ are the supports of the union of all the basis functions in $t$ and $s$ respectively, $diam(.)$ is the Euclidean diameter of a set, $dist(.\,,.)$ is the Euclidean distance between two sets, and $\eta$ is a positive parameter that can be used to control the admissibility condition. If subsets $t$ and $s$ satisfy (8), they are admissible; otherwise, they are inadmissible. Denoting the matrix block formed by $t$ and $s$ by $\mathbf{Z}^{t,s}$, if all the blocks $\mathbf{Z}^{t,s}$ formed by the admissible $(t, s)$ in matrix $\mathbf{Z}$ can be represented by a factorized form

$$\mathbf{Z}^{t,s} := \mathbf{V}^t \mathbf{S}^{t,s} \mathbf{V}^{s^T}, \ \mathbf{V}^t \in \mathbb{C}^{\#t \times k}, \ \mathbf{S}^{t,s} \in \mathbb{C}^{k \times k}, \ \mathbf{V}^s \in \mathbb{C}^{\#s \times k}, \quad (9)$$

where $\mathbf{V}^t$ is nested, then $\mathbf{Z}$ is an $H^2$ matrix. In (9), $\mathbf{V}^t$ is called a cluster basis, $\mathbf{S}^{t,s}$ is called a coupling matrix, $k$ is the rank of $\mathbf{V}^t$, and "#" denotes the cardinality of a set. The nested property of $\mathbf{V}^t$ enables linear-time arithmetics of $H^2$ matrices. Storage requirements, matrix-vector multiplications, and matrix-matrix multiplications using $H^2$-matrices have been shown to be of complexity $O(N)$ [12]. In [6-7], it is also shown that an $\mathcal{H}^2$-based inverse and LU can be performed in linear complexity.

# 3. SYSTEM MATRIX AND ITS REDUCTION

## 3.1 Formulation of the System Matrix

We discretize the conducting surface into triangular elements to accurately model arbitrarily-shaped conductors. In each triangular element, the equivalent magnetic current $\hat{n} \times \vec{E}$ and the equivalent electric current $\hat{n} \times \vec{H}$ are expanded by using RWG vector basis functions $\vec{J}_n$ [3] as follows

$$\hat{n} \times \vec{H} = \sum_{n=1}^{N} I_n \vec{J}_n \text{ and } \hat{n} \times \vec{E} = \sum_{n=1}^{N} II_n \vec{J}_n \,. \quad (10)$$

The charge density $\rho$ and the potential $\varphi$ are expanded by scalar pulse basis function in each element. We denote the total number of RWG bases by $N$, and the total number of triangular panels by $N_T$. The Galerkin method is applied to test (1) and (2). The centroid collocation method is applied to test (3) and (4). The resultant system of linear equations can be written as

$$Z_c \mathbf{L}_0 [I_n] - \mathbf{K}_0 [II_n] + \boldsymbol{\varphi}_{sp} [\varphi] = 0, \quad (11)$$

$$\mathbf{K}_1 [I_n] + \mathbf{L}_1 / Z_c [II_n] = 0, \quad (12)$$

$$[\varphi] = \mathbf{P}_0 [\rho], \quad (13)$$

$$\mathbf{J}_{sp} [I_n] = [\rho]_{NC}, \quad (14)$$

$$[\varphi]_C = \psi_C, \quad (15)$$

where the subscripts "$C$" and "$NC$" denote the quantities on the contact surfaces and non-contact surfaces, respectively. The $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, $\mathbf{P_0}$ in (11-15) are dense matrices given below

$$\begin{aligned} (\mathbf{L}_i)_{mn} &= jK_C \iint_{S_m} ds \, \vec{J}_m(\vec{r}) \iint_{S_n} ds' \vec{J}_n(\vec{r}\,') G_i \\ &\quad + (1/jK_C) \iint_{S_m} ds \nabla \cdot \vec{J}_m(\vec{r}) \iint_{S_n} ds' \nabla' \cdot \vec{J}_n(\vec{r}\,') G_i \\ (\mathbf{K}_i)_{mn} &= \iint_{S_m} ds \, \vec{J}_m(\vec{r}) \iint_{S_n} ds' \nabla G_i \times \vec{J}_n(r') - 0.5 \iint_{S_m} ds \, (\hat{n} \times \vec{J}_m(\vec{r})) \cdot \vec{J}_n(\vec{r}) \\ \mathbf{P}_0 &= (1/\varepsilon) \iint_S G_0(\vec{r}, \vec{r}\,') \, ds' \end{aligned} \quad (16)$$

with $i$ = 0, 1. The $\boldsymbol{\varphi}_{sp}$ in (11) is a sparse matrix of dimension $N \times N_T$. For each $(\boldsymbol{\varphi}_{sp})_{mn}$, $m$ is the index of the RWG basis function, the degree of freedom of which is assigned to each edge, and $n$ is the index of the pulse basis function, the degree of freedom of which is assigned to each triangular panel. The $\mathbf{J}_{sp}$ has a dimension of $NC \times N$, which is also sparse. The elements of $\boldsymbol{\varphi}_{sp}$ and $\mathbf{J}_{sp}$ are given by:

$$(\boldsymbol{\varphi}_{sp})_{mn} = -\iint_{S_m} ds \, \nabla \cdot \vec{J}_m(\vec{r}_n) \,, \quad (\mathbf{J}_{sp})_{mn} = \sigma\mu / K_C{}^2 \nabla \cdot \vec{J}_n(\vec{r}_m) \,. \quad (17)$$

The total number of unknowns in the system (11-15) is $2N+2N_T$, where $2N$ unknowns are associated with RWG basis functions, and $2N_T$ unknowns are associated with pulse basis functions.

As can be seen from (16), $\mathbf{L}_0$ and $\mathbf{L}_1$ are very similar to those resulting from the discretization of the conventional EFIE. The EFIE suffers from the well-known low-frequency break-down

problem [3]. However, this problem is significantly alleviated in $\mathbf{L}_0$ and $\mathbf{L}_1$. This is because although the first term and the second term in $\mathbf{L}_0$ and $\mathbf{L}_1$ scale with frequency differently, the matrix norm of the first term over that of the second term is in the order of $K_C^2 l^2$, where $l$ is the average edge length used in the discretization. Compared to the ratio of $k^2 l^2$ present in a conventional EFIE, $K_C^2 l^2$ is orders of magnitude larger since $K_C$ is $O(\sqrt{\omega\mu\sigma})$, and hence the breakdown frequency experienced by $\mathbf{L}_0$ and $\mathbf{L}_1$ is orders of magnitude smaller. Based on our quantitative analysis, for on-chip circuits the geometrical dimension of which is at μm level, one has to go to a frequency as low as $10^{-5}$ Hz to observe the breakdown of $\mathbf{L}_0$ and $\mathbf{L}_1$ in double precision computing. For package problems, this frequency is even lower. Therefore, one can safely use the formulation presented in this work for any low frequency of practical interest. However, special care still needs to be taken for the numerical integration of the near-field interaction in $\mathbf{L}_0$, $\mathbf{L}_1$, and $\mathbf{P}_0$ in the frequency range where the integrand varies very rapidly, as shown in [1].

## 3.2 Matrix Reduction

The matrix system shown in (11-15) consists of both sparse and dense blocks, which cannot be represented as an $H^2$ matrix as a whole. To develop an $H^2$-based direct solution of (11-15), our strategy is to eliminate all the $[II_n]$, $[\rho]$, and $[\varphi]$ unknowns from (11-15) to reduce the system to a small one that only involves $[I_n]$. Using (12) to eliminate $[II_n]$, (11) becomes

$$Z_c(\mathbf{L}_0 + \mathbf{K}_0 \cdot \mathbf{L}_1^{-1} \cdot \mathbf{K}_1)[I_n] + \boldsymbol{\varphi}_{sp}[\varphi] = 0 \Rightarrow \mathbf{H}_{\mathbf{LK}}[I_n] + \boldsymbol{\varphi}_{sp}[\varphi] = 0$$

$$\text{with } \mathbf{H}_{\mathbf{LK}} = Z_c(\mathbf{L}_0 + \mathbf{K}_0 \cdot \mathbf{L}_1^{-1} \cdot \mathbf{K}_1). \quad (18)$$

Based on (13), the charge $[\rho]$ on the non-contact surface can be represented by $[\varphi]$ as

$$[\rho]_{NC} = (\mathbf{P}_0^{-1})_{NC\times C}[\psi]_C + (\mathbf{P}_0^{-1})_{NC\times NC}[\varphi]_{NC} \quad . \quad (19)$$

Using (14) and (19), we can directly relate $[\varphi]$ to $[I_n]$ by

$$[\varphi]_{NC} = (\mathbf{P}_0)_{NC\times NC} \cdot \left(\mathbf{J}_{sp}[I_n] - (\mathbf{P}_0^{-1})_{NC\times C}\psi_C\right) \quad . \quad (20)$$

We can write (18) as

$$\mathbf{H}_{\mathbf{LK}}[I_n] + (\boldsymbol{\varphi}_{sp})_{N\times C} \cdot \psi_C + (\boldsymbol{\varphi}_{sp})_{N\times NC} \cdot [\varphi]_{NC} = 0 \quad . \quad (21)$$

Substituting (20) into (21), we obtain

$$\mathbf{Z}[I_n] = \mathbf{rhs}, \quad (22)$$

where,

$$\mathbf{Z} = \mathbf{H}_{\mathbf{LK}} + \mathbf{H}_{\mathbf{SP}}, \text{ with } \mathbf{H}_{\mathbf{LK}} = Z_c(\mathbf{L}_0 + \mathbf{K}_0 \cdot \mathbf{L}_1^{-1} \cdot \mathbf{K}_1)$$
$$\text{and } \mathbf{H}_{\mathbf{SP}} = (\boldsymbol{\varphi}_{sp})_{N\times NC} \cdot (\mathbf{P}_0)_{NC\times NC} \cdot \mathbf{J}_{sp} \quad (23)$$
$$\mathbf{rhs} = \left((\boldsymbol{\varphi}_{sp})_{N\times NC} \cdot (\mathbf{P}_0)_{NC\times NC} \cdot (\mathbf{P}_0^{-1})_{NC\times C} - (\boldsymbol{\varphi}_{sp})_{N\times C}\right) \cdot \psi_C$$

The number of unknowns involved in (22) is *N*, which is the total number of RWG bases used in the discretization.

In general, the unknown elimination has a cube complexity. In this work, we make sure that the elimination of $[II_n]$, $[\rho]$, and $[\varphi]$ is performed in linear complexity, and also the reduced $[I_n]$-system is solved in linear complexity, holding the complexity of the entire solution to linear. The linear-complexity computation is presented in the following Section 5.

# 4. NEW $H^2$ PARTITION

An $H^2$-based partition is to separate a matrix into admissible blocks and inadmissible blocks. This is generally done by a cardinality based splitting method [11]. Although this method is general, it is not efficient in the context of the surface integral based impedance extraction. In this work, we propose a new partition method that significantly improves the efficiency of the resultant $H^2$-based computation for wideband impedance extraction. We first show the proposed scheme for constructing a cluster tree, from which we build an $H^2$-partition. Assuming ($t_1$, $t_2$, $t_3$) to be a coordinate system, the arbitrarily-shaped conductors can be orientated in any direction in the 3-D space. We first find out the direction along which the structure being simulated has the maximal size, we then split the entire system into two sub-systems along this direction. We repeat the process, during which each conductor is treated as the smallest splitting-unit, and we do not split any single conductor. We continue to split in this way until one conductor is left in each sub-system. After that, we switch to another strategy to split a single conductor. We separate the single conductor into three groups that respectively contain the panels on $t_1t_2$-, $t_2t_3$-, $t_1t_3$- surfaces, and place each group as a top cluster of the conductor. We then use the conventional splitting method to construct the descendant clusters of the three top clusters. We keep such a splitting until the number of bases involved in each cluster is less than or equal to *leafsize*, which is a parameter to control the tree depth. The major advantage of the aforementioned scheme for constructing a cluster tree is that each cluster is made two dimensional, and hence fully taking advantage of the surface-based formulation to speed up the $H^2$-based computation of 3-D problems.

Based on the cluster tree and the admissibility condition (8), we construct an $H^2$ partition. There are two different basis functions used for discretization: one is RWG basis and the other is triangular-panel-based pulse basis. Therefore, we should construct a cluster tree-R and a cluster tree-T for RWG bases and pulse bases, respectively. The cluster tree-R is used to construct an $H^2$ partition of $\mathbf{L}_0$, $\mathbf{K}_0$, $\mathbf{L}_1$, $\mathbf{K}_1$, and the cluster tree-T is used to construct the $H^2$ partition of $\mathbf{P}_0$. Fig. 1(a) shows one example of such an $H^2$ partition. Although $\mathbf{L}_0$, $\mathbf{K}_0$, $\mathbf{L}_1$, $\mathbf{K}_1$ share the same $H^2$ partition, the $H^2$ partition of $\mathbf{L}_1$ and $\mathbf{K}_1$ is different from that of $\mathbf{L}_0$ and $\mathbf{K}_0$. This difference lies in the fact that Eqn. (2) is only satisfied in each conductor, and hence the matrix block formed for one conductor does not couple with that formed for another conductor that is physically disconnected. Thus, only the diagonal blocks that are formed by the same conductor cluster are nonzero and all the other blocks are zero as shown in Fig. 1(b).

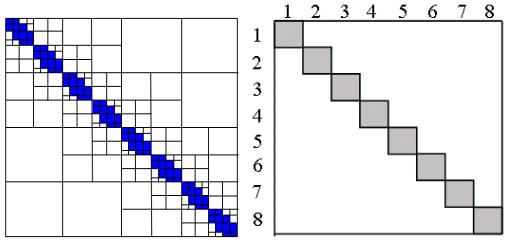


Fig. 1. An $\mathcal{H}^2$-matrix partition: (a) Partition for $\mathbf{L}_0$, $\mathbf{K}_0$ ( ■ full matrix block, □ admissible block). (b) Diagonal partition for $\mathbf{L}_1$, $\mathbf{K}_1$: grey blocks represent non-zero blocks, white blocks represent zero blocks.

# 5. LINEAR-COMPLEXITY DIRECT SOLUTION

In this section, we present a linear-complexity direct solution to (11-15) for full-wave based impedance extraction.

## 5.1 $H^2$ Representation of $L_0$, $K_0$, $L_1$, $K_1$, $P_0$, $P_0^{-1}$, and $L_1^{-1}$

First, we obtain an $H^2$ representation of all dense matrices $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ involved in the computation of (11-15). These matrices are given in (16). For an admissible block ($t$, $s$), we replace the $G_1$ and $G_0$ in (16) by a degenerate approximation

$$\tilde{G}^{t,s}(\vec{r},\vec{r}') = \sum_{\nu\in K^t}\sum_{\mu\in K^s} G(\xi_\nu^t,\xi_\mu^s) L_\nu^t(\vec{r}) L_\mu^s(\vec{r}') \tag{24}$$

where $(\xi_\nu^t)_{\nu\in K^t}$ and $(\xi_\mu^s)_{\mu\in K^s}$ are interpolation points chosen respectively in $t$ and $s$; and $(L_\nu^t)_{\nu\in K^t}$ and $(L_\nu^t)_{\nu\in K^t}$ are the corresponding Lagrange polynomials. The number of interpolation points in each cluster is $p^d$ , where $d$ is reduced to 2 by the proposed method although the problem to be solved is 3-D in nature. This is because the underlying integral equation formulation is surface based and, also, each cluster is made two dimensional by the new partition scheme. The accuracy of (24) can be controlled to any order as proved in [6] and [10].

By substituting (24) into (16), the double integrals in (16) can be separated into two single integrals, from which we obtain an $H^2$ – representation of $\mathbf{L_0}$, $\mathbf{K_0}$, $\mathbf{L_1}$, $\mathbf{K_1}$, and $\mathbf{P_0}$ as

$$\tilde{\mathbf{L}}_{0,1}^{t,s} = \mathbf{V}_{\mathbf{L}}^t \mathbf{S}_{\mathbf{L}_{0,1}}^{t,s} \mathbf{V}_{\mathbf{L}}^{s\mathrm{T}},\ \tilde{\mathbf{K}}_{0,1}^{t,s} = \mathbf{V}_{\mathbf{K}}^t \mathbf{S}_{\mathbf{K}_{0,1}}^{t,s} \times \mathbf{V}_{\mathbf{K}}^{s\mathrm{T}},\ \tilde{\mathbf{P}}_0^{t,s} = (\mathbf{V}_{p1}^T)^t \mathbf{S}_{\mathbf{P}_0}^{t,s} (\mathbf{V}_{p2}^T)^{s\mathrm{T}}, \tag{25}$$

where the cluster basis in a given cluster $t$ satisfies:

$$(\mathbf{V}_{\mathbf{K}}^t)_{i\nu} = \iint_{S_i} \vec{J}_i(\vec{r}) L_\nu^t(\vec{r}) ds,\ (\mathbf{V}_S^t)_{i\nu} = \iint_{S_i} (\nabla\cdot\vec{J}_i(\vec{r})) L_\nu^t(\vec{r}) ds$$

$$\mathbf{V}_{\mathbf{L}}^t = [\mathbf{V}_{\mathbf{K}}^t \quad \mathbf{V}_S^t],\ (\mathbf{V}_{p1}^T)_{i\nu}^t = L_\nu^t(\vec{r}_i),\ (\mathbf{V}_{p2}^T)_{i\nu}^t = \iint_{S_i} L_\nu^t(\vec{r}) ds. \tag{26}$$

The coupling matrices shown in (25) satisfy:

$$\begin{cases} \mathbf{S}_{\mathbf{L}_{0,1}}^{t,s} = \begin{bmatrix} jK_C \mathbf{S}_{G_{0,1}} & \\ & (1/jK_C)\mathbf{S}_{G_{0,1}} \end{bmatrix},\ (\mathbf{S}_{G_{0,1}})_{\nu\mu} = G_{0,1}(\xi_\nu^t,\xi_\mu^s) \\ (\mathbf{S}_{\mathbf{K}_{0,1}}^{t,s})_{\nu\mu} = \nabla G_{0,1}(\xi_\nu^t,\xi_\mu^s),\ (\mathbf{S}_{\mathbf{P}0}^{t,s})_{\nu\mu} = G_0(\xi_\nu^t,\xi_\mu^s)/\varepsilon \end{cases} \tag{27}$$

with $\mathbf{V}^t \in \mathbb{R}^{\#p\times\#K^t}$, $\mathbf{S}^{t,s} \in \mathbb{C}^{\#K^t\times\#K^s}$, and $\#K^{t(s)} << \#t$ ($\#s$) is the rank in the cluster $t$ or $s$ determined by the number of interpolation points. $\mathbf{V}_{\mathbf{L,K,S}}$ denotes cluster basis formed by the cluster tree-R, and $(\mathbf{V}^{\mathbf{T}})_{p1,p2}$ denotes cluster basis formed by the cluster tree-T. Cluster bases $\mathbf{V}_{\mathbf{L,K,S}}$ and $(\mathbf{V}^{\mathbf{T}})_{p1,p2}$ are all nested cluster bases.

After the $H^2$ representations of $\mathbf{P}_0$ and $\mathbf{L}_1$ have been constructed, we compute their inverses $\mathbf{P}_0^{-1}$ and $\mathbf{L}_1^{-1}$ to facilitate efficient computation of (22). As can be seen from (16), the $\mathbf{P}_0$ is similar to the dense system obtained from capacitance extraction problem; and the $\mathbf{L}_1$ is similar to the dense system formulated for an EFIE. Therefore, their inverses can be computed directly in linear complexity by the inverse algorithm in [6], and stored in $H^2$ matrices. It is worth mentioning that due to the block diagonal nature of $\mathbf{L}_1$, as shown in Fig. 1(b), to compute $\mathbf{L}_1^{-1}$, we only need to compute the inverse of each diagonal block.

From (26) and (27), it can be seen that $\mathbf{L}_0$ and $\mathbf{L}_1$ share the same cluster basis $\mathbf{V}_{\mathbf{L}}$, and they are only different in coupling matrix $\mathbf{S}$; similarly, $\mathbf{K}_0$ and $\mathbf{K}_1$ share the same cluster basis $\mathbf{V}_{\mathbf{K}}$ but with different coupling matrices $\mathbf{S}$. However, $\mathbf{V}_{\mathbf{L}}$ is different from $\mathbf{V}_{\mathbf{K}}$. The $\mathbf{P}_0$ even has different row and column cluster bases. Therefore, for efficient computation of (22), in what follows, we propose an algorithm to convert different cluster bases into the same set of cluster basis with nested property preserved.

## 5.2 Unifying Cluster Bases in Linear Time

We propose to use orthogonalization to unify the cluster bases. Take $\mathbf{V}_{\mathbf{L}}$ shown in (26) as an example, it is composed of a vector-based $\mathbf{V}_{\mathbf{K}}$ and a scalar-based $\mathbf{V}_S$. Given a cluster $t$, we first expand the vector-based $\mathbf{V}_{\mathbf{L}}^t$ into a scalar based form, i.e., $\mathbf{V}_{\mathbf{L}\text{expand}}^t = \left[\mathbf{V}_{\mathbf{K}x}^t \ \mathbf{V}_{\mathbf{K}y}^t \ \mathbf{V}_{\mathbf{K}z}^t \ \mathbf{V}_s^t\right]$. We then orthogonalize $\mathbf{V}_{\mathbf{L}\text{expand}}^t$, which results in an orthogonal cluster $\tilde{\mathbf{V}}_{\mathbf{L}}^t$ that contains all the independent column vectors of the original $\mathbf{V}_{\mathbf{L}}^t$. We use the orthogonalization algorithm in [14] to perform this task, which has a linear complexity. With $\tilde{\mathbf{V}}_{\mathbf{L}}^t$ obtained, the cluster bases of $\tilde{\mathbf{L}}_0$, $\tilde{\mathbf{L}}_1$, $\tilde{\mathbf{K}}_0$, and $\tilde{\mathbf{K}}_1$ can be accurately unified by $\tilde{\mathbf{V}}_{\mathbf{L}}^t$. For example, based on $\tilde{\mathbf{V}}_{\mathbf{L}}^t$, $\tilde{\mathbf{L}}_0$ can be updated based on the following equation:

$$\tilde{\mathbf{L}}_0^{t,s} = (\tilde{\mathbf{V}}_{\mathbf{L}}^t \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \cdot \mathbf{V}_{\mathbf{L}}^t)\cdot \mathbf{S}_{\mathbf{L}_0}^{t,s} \cdot (\tilde{\mathbf{V}}_{\mathbf{L}}^s \tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} \cdot \mathbf{V}_{\mathbf{L}}^s)^{\mathrm{T}} = \tilde{\mathbf{V}}_{\mathbf{L}}^t (\tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_{\mathbf{L}}^t \mathbf{S}_{\mathbf{L}_0}^{t,s} \mathbf{V}_{\mathbf{L}}^{s\mathrm{T}} \tilde{\mathbf{V}}_{\mathbf{L}}^s) \tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}}$$

$$= \tilde{\mathbf{V}}_{\mathbf{L}}^t \begin{pmatrix} \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_{\mathbf{K}x}^t (jK_C \mathbf{S}_{G_0}^{t,s}) (\tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} \mathbf{V}_{\mathbf{K}x}^s)^{\mathrm{T}} + \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_{\mathbf{K}y}^t (jK_C \mathbf{S}_{G_0}^{t,s}) (\tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} \mathbf{V}_{\mathbf{K}y}^s)^{\mathrm{T}} \\ + \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_{\mathbf{K}z}^t (jK_C \mathbf{S}_{G_0}^{t,s}) (\tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} \mathbf{V}_{\mathbf{K}z}^s)^{\mathrm{T}} + \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_S^t (\mathbf{S}_{G_0}/jK_C) (\tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} \mathbf{V}_S^s)^{\mathrm{T}} \end{pmatrix} \tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}} =$$

$$\tilde{\mathbf{V}}_{\mathbf{L}}^t \left( jK_C \left( \mathbf{W}_x^t \mathbf{S}_{G_0}^{t,s} \mathbf{W}_x^{s\mathrm{T}} + \mathbf{W}_y^t \mathbf{S}_{G_0}^{t,s} \mathbf{W}_y^{s\mathrm{T}} + \mathbf{W}_z^t \mathbf{S}_{G_0}^{t,s} \mathbf{W}_z^{s\mathrm{T}} \right) + \mathbf{W}_S^t \mathbf{S}_{G_0}^{t,s} \mathbf{W}_S^{s\mathrm{T}} / jK_C \right) \tilde{\mathbf{V}}_{\mathbf{L}}^{s\mathrm{T}}$$

with $\mathbf{W}_{x,y,z}^t = \tilde{\mathbf{V}}_{\mathbf{L}}^{t\mathrm{T}} \mathbf{V}_{\mathbf{K}x,y,z}^t$. $\tilde{\mathbf{K}}_0$ can be updated in a similar way.

## 5.3 $H^2$ Representation of $\mathbf{H_{LK}}$ and $\mathbf{H_{SP}}$

After we obtain the $H^2$ representation of $\tilde{\mathbf{L}}_0$, $\tilde{\mathbf{L}}_1$, $\tilde{\mathbf{K}}_0$, $\tilde{\mathbf{K}}_1$ as well as $\tilde{\mathbf{L}}_1^{-1}$ based on the above procedure, the $H^2$ representation of $\mathbf{H_{LK}}$ shown in (23) can be constructed based on an $H^2$-based matrix-matrix multiplication algorithm given in [12]. To be specific, by performing an $H^2$-based matrix-matrix multiplication in the diagonal blocks, we can obtain an $H^2$ based representation of $\tilde{\mathbf{L}}_1^{-1}\cdot\tilde{\mathbf{K}}_1$. By performing $\tilde{\mathbf{K}}_0 \cdot (\tilde{\mathbf{L}}_1^{-1}\tilde{\mathbf{K}}_1)$ and adding it upon $\tilde{\mathbf{L}}_0$, we can obtain an $H^2$ based representation of $\mathbf{H_{LK}}$. All these operations have a linear complexity since an $H^2$-based matrix-matrix multiplication as well as addition has a linear complexity.

In order to represent $\mathbf{H_{SP}}$ in (23) as an $H^2$ matrix, we need to compute an $H^2$-based product based on a sparse matrix $\boldsymbol{\varphi}_{\text{sp}}$ (or $\mathbf{J}_{\text{sp}}$) and an $H^2$ matrix $\mathbf{P_0}$. In order to make use of the linear-time matrix-matrix multiplication algorithm, we represent both sparse matrices $\boldsymbol{\varphi}_{\text{sp}}$ and $\mathbf{J}_{\text{sp}}$ as an $H^2$ matrix. To do so, we first construct an $H^2$ partition for $\boldsymbol{\varphi}_{\text{sp}}$. As can be seen from (17), the row cluster of $\boldsymbol{\varphi}_{\text{sp}}$ is formed by RWG bases, and the column cluster of $\boldsymbol{\varphi}_{\text{sp}}$ is formed by triangular panel based pulse basis. Hence, we choose

the cluster tree-R as its row cluster tree and cluster tree-T as its column cluster tree, and build an $H^2$ partition based on the two cluster trees and the same admissibility condition (8). We then fill in the matrix entry in each $H^2$-block. Based on (17), it can be seen that for an RWG edge *i,* only $(\boldsymbol{\varphi}_{sp})_{i,t1}$ and $(\boldsymbol{\varphi}_{sp})_{i,t2}$ are non-zero, and all the other $(\boldsymbol{\varphi}_{sp})_{i,*} = 0$. In other words, only adjacent RWG edges and triangular panels can have interactions in $\boldsymbol{\varphi}_{sp}$. Therefore, all the admissible blocks become zero, only a few inadmissible blocks are nonzero. Furthermore, in each nonzero block, only several elements are nonzero. Thus, we only need to record all the nonzero elements and their locations in the $H^2$ partition. The $H^2$ representation of $\mathbf{J}_{sp}$ can be formed similarly. After obtaining the $H^2$ form of $\boldsymbol{\varphi}_{sp}$ and $\mathbf{J}_{sp}$, we use them to compute $(\boldsymbol{\varphi}_{sp})_{N\times NC} \cdot (\mathbf{P}_0)_{NC\times NC} \cdot \mathbf{J}_{sp\,NC\times N}$ in (23), where all the operations are only associated with the nonzero elements in the $H^2$-based sparse matrix. The computation of such a matrix-matrix multiplication is equivalent to a few matrix-vector multiplications. When computing an $H^2$-based matrix-matrix multiplication, we choose the partition formed by the first matrix's row cluster and the second matrix's column cluster as the product's partition.

## 5.4 Direct Solution of Z

With the steps described in previous sections completed, we obtain an $H^2$ matrix $\tilde{\mathbf{Z}}$ to represent the $\mathbf{Z}$ matrix in (22). We then employ the $O(N)$ $H^2$-inverse algorithm [6] to compute $\tilde{\mathbf{Z}}^{-1}$ and employ the $O(N)$ $H^2$-based matrix-vector or matrix-matrix multiplication [12] to compute **rhs** as well as $[I_n] = \tilde{\mathbf{Z}}^{-1} \cdot \mathbf{rhs}$. Based on $[I_n]$, the current and impedance of the system can be extracted from (7). It should be noted that an $H^2$-based inverse is different from an $H^2$-based matrix-matrix multiplication. In a matrix-matrix multiplication, the computation is always performed based on the original matrix; whereas, in a matrix inverse, the computation at each step has to be performed based on updated matrices obtained from previous steps**.** Therefore, the linear-complexity algorithm of an $H^2$-based matrix-matrix multiplication cannot be directly applied to an $H^2$-based inverse.

## 5.5 Complexity Analysis

The storage of an $H^2$-matrix is $O(N)$. The total computational cost for solving the impedance system (22) includes two parts: one is the $H^2$-based construction of $\mathbf{P}_0^{-1}$, $\mathbf{L}_1^{-1}$, $\mathbf{H}_{SP}$ and $\mathbf{H}_{LK}$, which is the cost of eliminating unknowns $[II_n]$, $[\rho]$, and $[\varphi]$ from (11-15), and the other is the direct solution of (22) by computing $\mathbf{Z}^{-1}$, each of which is performed in linear complexity as analyzed in the subsections above. In addition, since $\mathbf{P}_0$ and $\mathbf{Z}$ are both symmetric, we only need to store and compute half of them.

# 6. NUMERICAL RESULTS

For all the examples simulated in this work, we choose *leafsize* = 8 and $\eta = 1$ for the construction of $H^2$ partition; and we use $p = 3$ for the interpolation along each dimension.

## 6.1 A Straight Conductor Wire

We first use a simple conductor wire to test the accuracy of the proposed method for wideband impedance extraction. The wire has a cross section of 1 mm by 1 mm and a length of 25 mm. The conductivity of the wire is 5.8e+7 S/m. A single discretisation resulting in 630 triangular panels is used across the entire frequency band from 1 Hz to $10^9$ Hz. Both real part and imaginary part of the impedance computed by the proposed method are compared with those extracted by FastHenry [15], which is based on a volume integral equation method accelerated by fast multipole algorithm. The inductances generated by both methods

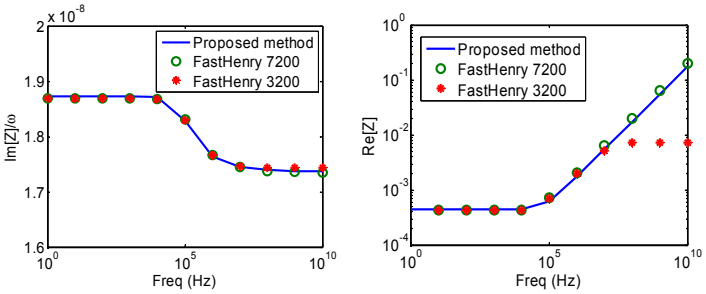


Fig. 2. Impedance of a straight conductor wire from 1 Hz to 1 GHz. (a) Imaginary part. (b) Real part.

agree with each other very well as can be seen from Fig. 2(a). For the real part of the impedance as shown in Fig. 2(b), FastHenry has to use a denser mesh with 7200 filaments to capture the skin effect in the higher frequency band, while the proposed surface-based method can still capture the frequency dependency with the same discretization.

## 6.2 Large-scale Spiral Inductor Array

In this example, we consider an inductor array composed of rectangular spirals, each of which has 4 full turns. The width, thickness, and spacing of the rectangular spiral are 1, 1, and 1$\mu m$, respectively, and the inner diameter of the rectangular spiral is 10 $\mu m$. The array includes 2×2, 2×4, 2×8, 2×16 rectangular spirals, which respectively results in 72692, 145384, 290768, 581536 number of unknowns. For a fair comparison, a similar discretisation is used in FastImp [13] and the residual error of GMRES used by FastImp is set as $10^{-2}$. The frequency is 1 GHz. Fig. 3(b) and (c) show the memory and the total solution time for both methods. As can be clearly seen, the total solution time of

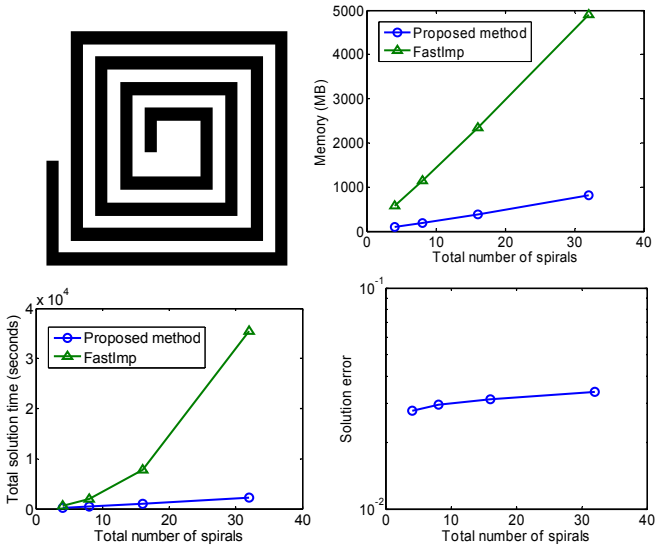


Fig. 3. Simulation of an array composed of rectangular spirals. (a) One rectangular spiral. (b) Memory. (c) Total solution time. (d) Solution error.

the proposed method scales linearly with the number of unknowns, whereas the total solution time of FastImp is affected by the number of iterations and the number of right hand sides, and hence FastImp failed to show linear complexity although its matrix-vector multiplication achieved almost linear complexity. In Fig. 3(d), we plot the solution error measured by $\|\tilde{\mathbf{Z}}I - \mathbf{rhs}\|/\|\mathbf{rhs}\|$ of the proposed method. Good accuracy is observed in the entire unknown range.

## 6.3 Large-scale 3-D Buses in Multiple Layers

The third example is a crossover bus structure in two layers and each layer has $m$ conductors. Each conductor has a size of $1\ \mu m \times 1\ \mu m \times (2m+1)\ \mu m$ and a conductivity of 5.8e+7 S/m. The $m$ is chosen as 1, 2, 4, 8, 16, 32, 64, which respectively results in 800, 4480, 14080, 48640, 179200, 686080, 2,682,880 number of unknowns. The impedance is extracted at 1 GHz. We test the efficiency of the proposed direct solver and compare its performance with FastImp. For a fair comparison, a similar discretization is used in both methods. The relative residual in GMRES used by FastImp is set as $10^{-3}$. The FastImp is only used to simulate the bus structure with $m$ up to 32 since the advantage of the proposed solver is already obvious. Fig. 4(a) shows the memory consumption by both methods. It can be seen that the memory required by the proposed method is 8 times less than that required by FastImp. The time for one matrix-vector multiplication is plotted in Fig. 4(b), which shows that the proposed scheme is about 18 times faster than FastImp. Fig. 4(c) shows the total solution time of the proposed method that includes the matrix reduction, the construction of the $H^2$ representation, and the direction solution of (22). As can be seen clearly, the total solution time of the proposed method scales linearly with the number of unknowns. It is also much faster than FastImp even though the proposed method computes the entire inverse, whereas FastImp only solves for $2m$ right hand sides. In addition, we test the accuracy of the extracted impedance. One column of the impedance matrix is extracted. We test the solution accuracy of both the proposed method and FastImp based on $\|\mathbf{Z}_1 - \mathbf{Z}'_1\| / \|\mathbf{Z}_1\|$, where $\mathbf{Z}_1$ is a vector of impedances between port 1 and other ports computed by a *full-matrix* based direct solver, while $\mathbf{Z}'_1$ is computed by either the proposed method or FastImp. Since the full-matrix-based direct solver is very expensive, we only use this approach to assess error for small bus structures. It is shown that the accuracy of the proposed method is 0.32%, and 0.47% respectively for the bus structure with $m$=1 and $m$=2. For the same two structures, the accuracy of FastImp is 0.70%, and 0.89% respectively. Since both methods achieve very good accuracy, we use FastImp to benchmark the accuracy of the proposed method for larger buses based on $\|\mathbf{Z}_{1,this} - \mathbf{Z}_{1,fastimp}\| / \|\mathbf{Z}_{1,fastimp}\|$, where $\mathbf{Z}_{1,this}$ is computed by the proposed method, and $\mathbf{Z}_{1,fastimp}$ is computed by FastImp. As can be seen from Fig. 4(d), an excellent accuracy is observed in the entire unknown range.

# 7. CONCLUSIONS

A direct solver of linear complexity is developed for the surface integral based impedance extraction of arbitrarily-shaped non-ideal 3-D conductors embedded in materials. Numerical results demonstrate its superior performance.

# 8. REFERENCES


[1] Zhenhai Zhu, Ben Song and Jacob White, "Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3-D geometries," *DAC 2003*, pp. 712-717.

[2] S. Kapur and D.E. Long, "IES$^3$: A fast integral equation solver for efficient 3-dimensional extraction," *ICCAD*, pp. 448-455, 1997.

[3] "Fast and Efficient Algorithms in Computational Electromagnetics," edited by W. C. Chew, J. M. Jin, E. Michielssen, and J. M. Song. Norwood, MA: Artech House, 2001.

[4] Y. Yi, P. Li, V. Sarin, W. Shi, "A Preconditioned Hierarchical Algorithm for Impedance Extraction of Three-Dimensional Structures With Multiple Dielectrics," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 27(11): 1918-1927 (2008).

[5] D. Gope, I. Chowdhury, and V. Jandhyala, "DiMES: Multilevel fast direct solver based on multipole expansions for parasitic extraction of massively coupled 3D microelectronic structures," *DAC* 2005.

[6] W. Chai, D. Jiao, and C. C. Koh, "A Direct Integral-Equation Solver of Linear Complexity for Large-Scale 3D Capacitance and Impedance Extraction," *DAC*, pp. 752-757, July, 2009.

[7] W. Chai and D. Jiao, "An LU Decomposition Based Direct Integral Equation Solver of Linear Complexity and Higher-Order Accuracy for Large-Scale Interconnect Extraction," *IEEE Trans. Advanced Packaging*, vol. 33, no. 4, pp. 794-803, 2010.

[8] J. Shaeffer, "Direct Solve of Electrically Large Integral Equations for Problem Sizes to 1 M unknowns," *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2306–2313, Aug. 2008.

[9] Ben Song, Zhenhai Zhu, John Rockway and Jacob White, "A new surface integral formulation for wideband impedance extraction of 3-D structures," *ICCAD*, pp. 843-847, 2003.

[10] W. Chai and D. Jiao, "An H$^2$-Matrix-Based Integral-Equation Solver of Reduced Complexity and Controlled Accuracy for Solving Electrodynamic Problems," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 3147–3159, Oct. 2009.

[11] S. Borm, L. Grasedyck, and W. Hackbusch, "Hierarchical matrices," Lecture note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

[12] S. Börm, "$\mathcal{H}^2$-matrix arithmetics in linear complexity," *Computing*, 77: 1-28, 2006.

[13] http://www.mit.edu/people/zhzhu/fastImp.html

[14] Börm, S., "Approximation of integral operators by H$^2$-matrices with adaptive bases," *Computing* 74 (2005), 249-271.

[15] http://www.rle.mit.edu/cpg/research_codes.htm


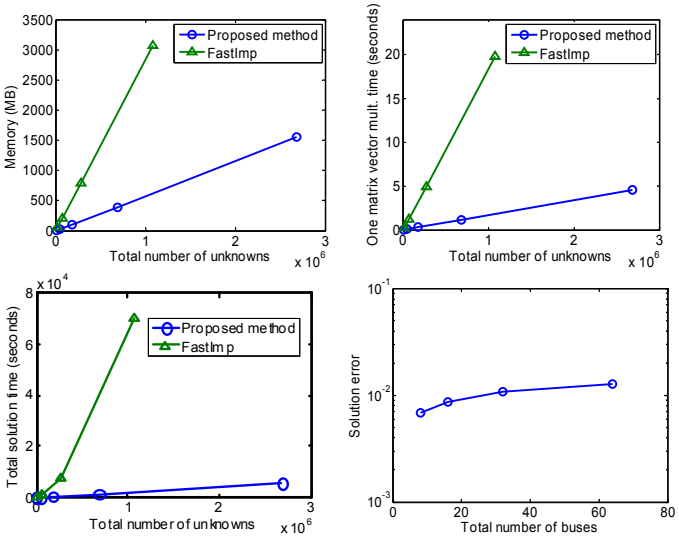


Fig. 4. Simulation of a 3-D bus. (a) Memory. (b) Time for one matrix-vector product. (c) Total solution time. (d) Impedance error compared to FastImp.