Selective Inversion of Inductance Matrix for Large-Scale Sparse RLC Simulation

Ifigeneia Apostolopoulou
Department of Electrical and Computer Engineering
University of Thessaly
Gklavani 37 str. , Volos, 38221, Greece
ifaposto@inf.uth.gr

Konstantis Daloukas
Department of Electrical and Computer Engineering
University of Thessaly
Gklavani 37 str. , Volos, 38221, Greece
kodalouk@inf.uth.gr

Nestor Evmorfopoulos
Department of Electrical and Computer Engineering
University of Thessaly
Gklavani 37 str. , Volos, 38221, Greece
nestevmo@inf.uth.gr

George Stamoulis
Department of Electrical and Computer Engineering
University of Thessaly
Gklavani 37 str. , Volos, 38221, Greece
georges@inf.uth.gr

ABSTRACT

The inverse of the inductance matrix (reluctance matrix) is amenable to sparsification to a much greater extent than the inductance matrix itself. However, the inversion and subsequent truncation of a large dense inductance matrix to obtain the sparse inverse is very time-consuming, and previously proposed window-based techniques cannot provide adequate accuracy. In this paper we propose a method for selective inversion of the inductance matrix to a prescribed sparsity ratio, which is also amenable to parallelization on modern architectures. Experimental results demonstrate its potential to provide efficient and accurate approximation of the reluctance matrix for simulation of large-scale RLC circuits.

1. INTRODUCTION

The demand for high-frequency large-scale integrated circuits is greater than ever, and necessitates the accurate and efficient modeling of on-chip inductive effects. These effects are ubiquitous in power delivery networks, clock networks, and long and wide bus structures, and can severely affect the signal and power integrity of the chip.

The partial equivalent element circuit (PEEC) method [16] has been widely used to model the circuit inductances whenever the return paths that form the current loops are unknown. However, if the mutual couplings between inductive branches are taken into account, the resulting partial inductance matrix **L** will be a fully dense matrix, which renders its direct use into the subsequent circuit simulation impractical.

A sparsification approach of **L** is mandatory to ensure reasonable simulation times. The simplest approach is to directly truncate (i.e. set to zero) all entries of **L** with magnitude smaller than a certain threshold. However, it is known that the elements of **L** do not decay very fast away from the diagonal, and even more importantly, the simple truncation of **L** can destroy its positive definiteness and the passivity of the circuit, which will lead to unstable simulation [9]. A very important observation, first made in [1], is that the entries of the inverse of the inductance matrix $\mathbf{K} \equiv \mathbf{L}^{-1}$ (called reluctance or susceptance matrix) diminish much faster than those of **L**, which means that the truncated matrix $\widetilde{\mathbf{K}}$ will be sparser than the truncated matrix $\widetilde{\mathbf{L}}$. Also, its sparsification can be done in a stable manner, since **K** can be proven to be diagonally dominant and thus the truncation of any off-diagonal entries will retain the diagonal dominance property, which ensures that $\widetilde{\mathbf{K}}^{-1}$ will still be positive definite.

Although the reluctance matrix offers an efficient and accurate framework for the simulation of RLC circuits, it involves a large and dense matrix inversion which is prohibitively expensive both from a computational and a memory storage perspective. Window-based approaches have been proposed [1]-[4] that result in many smaller matrices (each one corresponding to a window around a conductor segment) to be locally inverted, instead of inversion and truncation of the full inductance matrix. A similar approach called wire-duplication has been proposed in [5], which works directly on the matrix instead of the physical geometry. Related approaches [6]-[7] are very limited in the sense that they only work on buses of parallel lines (in single and multiple layers respectively), which give rise to banded matrices.

The main problem with window-based approaches is that while the reluctance matrix is amenable to sparsification to a greater extent than the inductance matrix, it is by no means as amenable as the capacitance matrix. This has been analyzed in [2], and can be also illustrated in Fig. 1 where we observe that a quite small sparsity ratio (ε = 94.9%, or 3,422,552 nonzeros in a 8192×8192 matrix) is needed to approximate the exact response (ε = 0.0%) of a middle line in a 4-layer 1024-wire bus. The aforementioned fact calls for larger window sizes in practice for reluctance calculation than for capacitance calculation. For example, it has been reported in [2] that a 12X larger window is required for reluctance than for capacitance to attain comparable accuracy. The larger window sizes needed for reluctance calculation - which increase further with the size of the inductance matrix - naturally lead to larger computation times for the (not so small) inverses which even exceed in total the time for full inversion (as we will see in our experimental results).

In this paper we propose the "Probing Approximate Inverse" algorithm which provides an efficient method for selective inversion of the inductance matrix **L** to a prescribed sparsity ratio. The algorithm attempts to directly calculate selected entries of $\mathbf{K} \equiv \mathbf{L}^{-1}$ from **L** without undergoing the expensive inversion and truncation, via the construction of a small number of suitable

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
DAC '14, June 01 - 05 2014, San Francisco, CA, USA
Copyright 2014 ACM 978-1-4503-2730-5/14/06…$15.00.
http://dx.doi.org/10.1145/2593069.2593213

"probing" vectors. This results in a highly accurate sparse approximation $\widetilde{\mathbf{K}}$, which is much faster to compute than either inversion and truncation or window-based methods for the sparsities needed in practice to supply acceptable accuracy. Moreover, the proposed algorithm offers substantial degree of parallelism, and thus it can harness the computational capabilities of parallel architectures to enable fast simulation of very large designs. Experimental results demonstrate that our methodology can achieve a speedup of 12.3X over the wire duplication method (or any other window-based method), for a sufficiently low sparsity that gives adequate accuracy for lines away from the active wire.

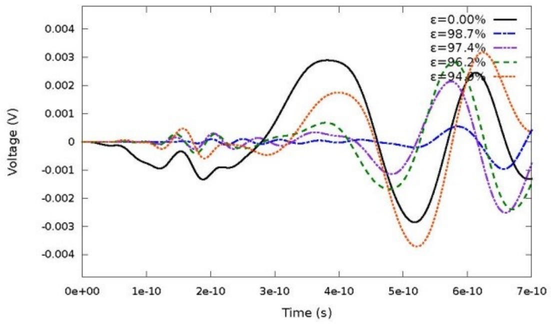

Fig. 1: Voltage response at the far end of line 640 (the middle line of the third layer) of a 4-layer 1024-wire bus, for different sparsity ratios of the reluctance matrix.

The rest of the paper is organized as follows: Section 2 provides some background material regarding RLC simulation, along with the theory behind the selective approximation of matrix elements by probing vectors. The proposed methodology for selective inversion of the inductance matrix is presented in Section 3, and its performance and accuracy is assessed by experimental results in Section 4. Finally, Section 5 concludes the paper.

2. THEORETICAL BACKGROUND

2.1 Overview of RLC simulation

Let the RLC circuit under consideration be composed of *N* nodes and *n* inductive branches, with mutual coupling present between a substantial part of them. By using the Modified Nodal Analysis (MNA) framework in such a linear circuit, we obtain the following system of differential equations:

$$\widetilde{\mathbf{G}}\mathbf{x}(t) + \widetilde{\mathbf{C}}\dot{\mathbf{x}}(t) = \mathbf{b}(t)$$

where

$$\widetilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \mathbf{A}_L \\ -\mathbf{A}_L^T & \mathbf{0} \end{bmatrix}, \quad \widetilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{L} \end{bmatrix}, \quad \mathbf{x}(t) = \begin{bmatrix} \mathbf{v}(t) \\ \mathbf{i}(t) \end{bmatrix}, \quad \mathbf{b}(t) = \begin{bmatrix} \mathbf{e}(t) \\ \mathbf{0} \end{bmatrix}$$

In the above system, **G** and **C** are the $N \times N$ node conductance and node capacitance matrices, **L** is the $n \times n$ dense inductance matrix (with self-inductances as diagonal entries and mutual inductances as off-diagonal entries), and $\mathbf{A}_L$ is the corresponding $N \times n$ node-to-branch incidence matrix. Also, $\mathbf{v}(t)$ and $\mathbf{i}(t)$ are the $N \times 1$ and $n \times 1$ vectors of node voltages and inductive branch currents, while $\mathbf{e}(t)$ is the $N \times 1$ vector of excitations from independent sources at the nodes (with voltage sources being transformed to Norton-equivalent current sources, as described in [4]). Using the Backward-Euler discretization in the above differential system, we obtain the following system of linear algebraic equations:

$$\left(\widetilde{\mathbf{G}} + \frac{1}{h}\widetilde{\mathbf{C}}\right)\mathbf{x}(t_k) = \mathbf{b}(t_k) + \frac{1}{h}\widetilde{\mathbf{C}}\mathbf{x}(t_{k-1}), \quad k = 1,2,\ldots$$

where *h* is the chosen time-step (which need not be constant during the analysis). By block-matrix operations on the above system (which are equivalent to the companion models presented in [11]-[12]), we obtain the following system of coupled recursive equations:

$$\left(\mathbf{G} + \frac{1}{h}\mathbf{C} + h\mathbf{A}_L\mathbf{L}^{-1}\mathbf{A}_L^T\right)\mathbf{v}(t_k) = \mathbf{e}(t_k) + \frac{1}{h}\mathbf{C}\mathbf{v}(t_{k-1}) - \mathbf{A}_L\mathbf{i}(t_{k-1})$$

$$\mathbf{i}(t_k) = h\mathbf{L}^{-1}\mathbf{A}_L^T\mathbf{v}(t_k) + \mathbf{i}(t_{k-1}), \qquad k = 1,2,\ldots$$

At each time-step t_k, $k = 1,2,\ldots$ we have to solve a $N \times N$ linear system with system matrix $\mathbf{A} \equiv \mathbf{G} + \mathbf{C}/h + h\mathbf{A}_L\mathbf{K}\mathbf{A}_L^T$ (where $\mathbf{K} \equiv \mathbf{L}^{-1}$ is the reluctance matrix) in order to obtain the vector of node voltages $\mathbf{v}(t_k)$, and then find the vector of inductive branch currents $\mathbf{i}(t_k)$ by a simple matrix-vector product. In a typical simulation flow the matrix **K** is substituted by a sparse approximation $\widetilde{\mathbf{K}}$ in order to render the simulation efficient.

2.2 Probing matrix elements by vectors

Let **K** be a $n \times n$ matrix and $\mathbf{v}_1,\ldots,\mathbf{v}_s$ be $s << n$ vectors of size $n \times 1$ each. If these vectors (referred to as "probing" vectors) make up the columns of a $n \times s$ matrix **V**, and if $\mathbf{D} = diag(\mathbf{V}\mathbf{V}^T)$ is the $n \times n$ diagonal matrix composed of the main diagonal of $\mathbf{V}\mathbf{V}^T$, then we can easily verify that the (i,j)-element of the matrix $\mathbf{V}\mathbf{V}^T\mathbf{D}^{-1}$ is equal to $(\mathbf{r}_i^T\mathbf{r}_j)/(\mathbf{r}_j^T\mathbf{r}_j)$, where $\mathbf{r}_i^T$ is the *i*-th row of **V**. The matrix $\mathbf{K}(\mathbf{V}\mathbf{V}^T\mathbf{D}^{-1})$ will then have elements:

$$k_{ij} + \sum_{\substack{l=1 \\ l \neq j}}^{n} k_{il}(\mathbf{r}_l^T\mathbf{r}_j)/(\mathbf{r}_j^T\mathbf{r}_j), \quad i,j = 1,2,\ldots,n$$

It is obvious that when $s = n$ and all rows $\mathbf{r}_i^T$ are orthogonal to each other [i.e. $\mathbf{r}_i^T\mathbf{r}_j = 0$, $\forall i,j = 1,2,\ldots,n \ (i \neq j)$], then $\mathbf{V}\mathbf{V}^T\mathbf{D}^{-1} = \mathbf{I}$ and thus $\mathbf{K}\mathbf{V}\mathbf{V}^T\mathbf{D}^{-1} = \mathbf{K}$. We seek to satisfy this equality in selected entries of **K**.

Assume that $\Im = \{(i,j) : i \in \{1,2,\ldots,n\}, j \in \{1,2,\ldots,n\}\}$ is the set of index pairs denoting the nonzero elements of **K** (or the most significant ones, e.g. those above a certain threshold), and $\wp \subseteq \Im$ is the set of index pairs that are to be kept in the sparse approximation $\widetilde{\mathbf{K}}$ of **K**. Of course it can be selected $\wp = \Im$, but we have observed that the elements of $\widetilde{\mathbf{K}}$ are more accurately estimated by the probing vectors, with only marginal additional cost and the same sparsity in $\widetilde{\mathbf{K}}$, if we introduce extra elements (those with indices in $\Im - \wp$) for their computation. In that case, the condition for $\mathbf{K}\mathbf{V}\mathbf{V}^T\mathbf{D}^{-1}$ to match **K** only in the elements with indices in $\wp$ (i.e. the elements of $\widetilde{\mathbf{K}}$) is:

$\forall (i,j) \in \wp$, and $\forall l \neq j$ such that $(i,l) \in \Im$, it is $\mathbf{r}_l^T\mathbf{r}_j = 0$

2.3 Construction of the probing vectors

In order to construct the probing vectors we first exploit the well-known matrix-graph connection, in which a $n \times n$ sparse matrix corresponds to a graph of *n* vertices and a given set of index pairs (characterizing the nonzeros of the matrix) corresponds to a set of edges in the graph. Here, we form a modified graph associated with both sets $\Im$ and $\wp$, where the aim is to link by an edge every pair of vertices (l,j) for which $(i,j)\in\wp$ and $(i,l)\in\Im$ ($\forall i=1,2,\ldots,n$). This is done by Algorithm 1.In order, now, to find a suitable set of probing vectors with orthogonal rows $\mathbf{r}_l^T\mathbf{r}_j=0$ we only need to perform a standard graph coloring, i.e. to assign an integer color to each vertex of the graph in a way that no vertices linked by an edge have the same color. Ideally, we want to color the graph with the smallest number of colors, but this is known to be an NP-complete problem [8]. A well-known greedy algorithm to find a graph coloring with an acceptably small number of colors is given in Algorithm 2.

```
Inputs: Vertex set $N=\{1,2,\ldots,n\}$
        Edge sets $\Im$ and $\wp$ (preferably in compressed
        row format)
Output: Modified graph $G(N,E)$
$E:=\{\}$
for $i=1,2,\ldots,n$
    for each $j$ such that $(i,j)\in\wp$ do
        for each $l\neq j$ such that $(i,l)\in\Im$ do
            $E:=E\cup(l,j)$
        end
    end
end
```

Algorithm 1: Construction of the modified graph associated with given edge sets $\Im$ and $\wp$.

```
Input: Modified graph $G(N,E)$ associated with $\Im$, $\wp$
Output: Colors $c_1,c_2,\ldots,c_n$ of the graph vertices
Initialize all colors to 0  ($c_1=c_2=\cdots=c_n=0$)
for $j=1,2,\ldots,n$
    $c_j:=\min\{k>0\mid k\neq c_l,\forall(l,j)\in E\}$
end
```

Algorithm 2: Greedy graph coloring.

We remark here that the quality of the coloring (i.e. the number of the resulting colors) is dependent on the ordering of the vertices. Further details on graph coloring, as well as more advanced coloring algorithms, can be found in [8]. After coloring of the modified graph associated with $\Im$ and $\wp$, the probing matrix $\mathbf{V}$ can be constructed as follows:

```
Input: Colors $c_1,c_2,\ldots,c_n$ of the vertices of $G(N,E)$
       Number of colors $s:=\max\{c_1,c_2,\ldots,c_n\}$
Output: Matrix of probing vectors $\mathbf{V}=[\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_s]$
Initialize all probing vectors to $\mathbf{0}$  ($\mathbf{V}=\mathbf{0}$)
for $j=1,2,\ldots,n$
    $v_{j,c_j}:=1$
end
```

Algorithm 3: Construction of the probing matrix.

From the above algorithm it follows that each row of $\mathbf{V}$ consists of exactly one nonzero entry equal to 1, which gives $\mathbf{r}_j^T\mathbf{r}_j=1$, $\forall j=1,2,\ldots,n$, and thus $\mathbf{D}=\mathbf{D}^{-1}=\mathbf{I}$. Furthermore, by construction of the graph (Algorithm 1) and its coloring (Algorithm 2), if $(i,j)\in\wp$ and $(i,l)\in\Im$ ($l\neq j$) then $(l,j)\in E$, and thus rows $\mathbf{r}_l^T$ and $\mathbf{r}_j^T$ will have their only nonzero in different columns (c_l and c_j respectively) which leads to $\mathbf{r}_l^T\mathbf{r}_j=0$.

As an example, let $\mathbf{K}$ be a 8×8 matrix with nonzero pattern $\Im$ and sparse approximation pattern $\wp\subseteq\Im$ shown in Fig. 2(a). Fig. 2(b) depicts the adjacency matrix of the modified graph constructed by Algorithm 1 and the color of each vertex obtained by Algorithm 2. The total number of colors is $s=4$.

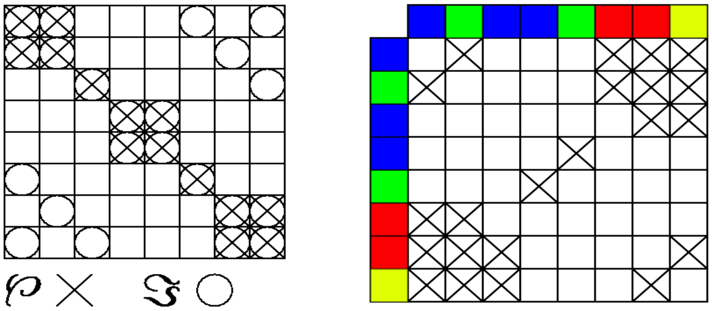

Fig. 2: (a) Example edge patterns $\Im$ and $\wp$ (O and × respectively), and (b) modified graph and its coloring.

The probing matrix $\mathbf{V}$ that results from Algorithm 3 is:

$$\mathbf{V}=\begin{bmatrix}1&0&0&0\\0&1&0&0\\1&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$$

while the matrix $\mathbf{KVV}^T\mathbf{D}^{-1}\equiv\mathbf{KVV}^T$ is:

$$\mathbf{KVV}^T=\begin{bmatrix}k_{11}&k_{12}&*&*&*&*&*&*\\k_{21}&k_{22}&*&*&*&*&*&*\\*&*&k_{33}&*&*&*&*&*\\*&*&*&k_{44}&k_{45}&*&*&*\\*&*&*&k_{54}&k_{55}&*&*&*\\*&*&*&*&*&k_{66}&*&*\\*&*&*&*&*&*&k_{77}&k_{78}\\*&*&*&*&*&*&k_{87}&k_{88}\end{bmatrix}$$

We can clearly see that the entries with indices in $\wp$ coincide with the corresponding entries in $\mathbf{K}$ (the rest of the entries are irrelevant).

3. METHODOLOGY FOR SELECTIVE INVERSION OF INDUCTANCE MATRIX

In our particular problem, we have the matrix **L** of self and mutual inductances readily available, and we want to estimate selected entries of its inverse $\mathbf{K} = \mathbf{L}^{-1}$ (given the sets of index pairs $\Im$ and $\wp$ for **K**). In this case, after constructing the $n \times s$ probing matrix **V**, the product $\mathbf{X} \equiv \mathbf{KV}$ can be computed by solving $\mathbf{LX} = \mathbf{V}$ which entails solving *s* linear systems $\mathbf{Lx}_i = \mathbf{v}_i$, $i = 1,2,\ldots,s$, of size $n \times n$ each. If $s << n$ then the resulting complexity will be much smaller than solving *n* linear systems which are required for full matrix inversion. After the solution of $\mathbf{LX} = \mathbf{V}$, the product $\mathbf{XV}^T \equiv \mathbf{KVV}^T$ can be computed directly from **X** since its (i, j) - element is equal to x_{i,c_j}. The whole procedure for selective inversion of **L** is given in the following Algorithm 4:

```
Input: Dense inductance matrix $\mathbf{L}$
       Nonzero pattern $\Im$ and sparse approximation
       pattern $\wp$ of $\mathbf{K} = \mathbf{L}^{-1}$
Output: Sparse approximation $\widetilde{\mathbf{K}}$ of $\mathbf{K}$
Construct modified graph $G(N, E)$ associated with $\Im$ and $\wp$ from Algorithm 1
Color $G(N, E)$ according to Algorithm 2
Construct the matrix of probing vectors $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_s]$ from Algorithm 3
for $i = 1,2,\ldots,s$
    Solve $\mathbf{Lx}_i = \mathbf{v}_i$
end
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_s]$
for $i = 1,2,\ldots,n$
    for each j such that $(i, j) \in \wp$ do
        $k_{ij} = x_{i,c_j}$
    end
end
```

Algorithm 4: Selective inversion of inductance matrix.

Of course **K** is not actually sparse so $\mathbf{KVV}^T$ matches only approximately the entries of **K** in $\wp$. However, since **K** is diagonally dominant it has many entries which are negligible, leading to a very good approximation (especially for large matrices). The number *m* of elements to be kept in the sparse approximation $\widetilde{\mathbf{K}}$ is a user-defined parameter for any truncation or window-based method, and represents a compromise between accuracy and sparsity of $\widetilde{\mathbf{K}}$ for the subsequent simulation. Usually, a target sparsity ratio $\varepsilon = (n \times n - m)/(n \times n)$ is specified up-front from which the number *m* follows. In our proposed method we introduce extra nonzeros $\Im$ to improve the accuracy in the estimation of entries in $\wp$. A sound number for those extra entries is 2*m*, which has been found to significantly improve the estimation accuracy while only marginally increase the number of colors in the modified graph (and subsequent systems to be solved).

For the patterns $\Im$ and $\wp$ of nonzeros in **K** it is very reasonable to match them with the positions of the 2*m* and *m* largest elements in the inductance matrix **L** itself. It can be proven [13] that for any diagonally dominant matrix the entries of its inverse decay (row-wise and column-wise) away from the diagonal, leading to a very similar structure between **K** and **L** (differing only in the rates of decay). This fact has been employed previously in [14] to select a sparsity pattern of a diagonally dominant matrix when its inverse is available. In case that we have additional information on the physical geometry of the problem, we can exploit it to select a better sparsity pattern for **K** (e.g. the elements along some important diagonals, as is described in [7]).

The computationally most demanding step in Algorithm 4 is the solution of the *s* linear systems $\mathbf{Lx}_i = \mathbf{v}_i$, since the rest are operations of complexity $O(n)$ or linear in the number of edges in $\Im$ and $\wp$. The sequence of linear systems can be solved by a standard direct or iterative method. In addition, they are completely independent to each other and can be solved in parallel. The fact that their number is much smaller than their dimension renders the procedure attractive for mapping onto many-core systems or distributed-memory multiprocessors where enough powerful processors are available.

Other parallelization opportunities include the construction of probing vectors in **V** (Algorithm 3) and the computation of the product $\mathbf{XV}^T$ (second loop of Algorithm 4). Also, the outer loop in the construction of the modified graph (Algorithm 1) can be parallelized if the index pairs $\Im$ and $\wp$ are stored in compressed row format, since every row index *i* can be treated independently from the others and all sub-graphs be concatenated in the end. The greedy graph coloring procedure of Algorithm 2 is not inherently parallel, but various parallel graph coloring algorithms have been developed and are presented in [15].

The internal memory requirements of the proposed method are also very reasonable, and can be made even smaller by allocating storage only for the *n* nonzero entries of **V** and only for the entries of **X** that are involved in the computation of $\mathbf{XV}^T$.

Finally, it should be mentioned that a small adjustment of $\widetilde{\mathbf{K}}$ might be required to ensure that it remains diagonally dominant (e.g. by increasing - wherever needed - the diagonal entry to exceed the corresponding row sum), in order to guarantee that $\widetilde{\mathbf{K}}^{-1}$ is positive definite, but has not been proven necessary in any of our experiments.

4. EXPERIMENTAL RESULTS

For the experimental evaluation of the proposed methodology we consider a bus structure of 3D geometry (nevertheless, our methodology is completely general and applicable to any circuit model with mutual inductances). The bus consists of 4 layers with two 128-wire blocks per layer (giving a total of 1024 wires), and each wire is divided into 8 segments yielding an inductance matrix of size 8192×8192. The wires have length 1mm and cross-section $1\mu\mathrm{m}^2$. The separation between successive layers or between blocks within a layer is 2μm, and the separation between adjacent wires within the same block is 1μm. The inductance matrix **L** for this structure was calculated using FastHenry [10] for the conductivity of aluminum, $\sigma = 3.77 \times 10^7 \ (\Omega \cdot \mathrm{m})^{-1}$. For the RLC simulation of the structure, the driver resistance was 30Ω, the load capacitance was 20fF, and the total wire capacitance was 40fF. A 1V 20ps ramp voltage source was applied to the first line of the lowest layer (bus line 0) while the rest of the lines were inactive. The bus was simulated for a period of 700ps with a fixed time step of 1ps. We applied the proposed Probing Approximate Inverse (PB) algorithm to calculate the sparse reluctance matrix $\tilde{\mathbf{K}}$ used in the RLC simulation (Section 2.1), and compared the results with the Inversion and Truncation (IT) of $\mathbf{L}$ and the method of Wire Duplication (WD). Since the latter is a window-based method, we do not report results for other window-based methods as they are very similar.

Fig. 3 and Fig. 4 depict the voltage response at the far end of the middle line in the third layer (line 640) and the most distant line of the bus (line 1023) for a sparsity ratio ε = 94.9% (low enough to provide acceptable accuracy in the simulation of this structure). Fig. 5 depicts the voltage response at the far end of line 640 for a sparsity ratio ε = 97.4% (at which the simulation accuracy suffers). Table 1 reports the execution time for constructing the sparse reluctance matrix $\tilde{\mathbf{K}}$ by the three methods under comparison for various sparsity ratios. The actual simulation times are not reported for the three methods, since they are obviously identical for a given sparsity ratio in $\tilde{\mathbf{K}}$.

As we can observe from Fig. 3, the PB method is able to achieve slightly better accuracy than the WD method for a middle line, but with much smaller construction time as seen in Table 1. In addition, as depicted in Fig. 4, the PB method offers much higher accuracy than WD for a faraway line, while still being substantially faster. For the sparser reluctance matrices $\tilde{\mathbf{K}}$ with ratio ε = 97.4%, where the construction time of WD becomes comparable to the construction time of PB, we can observe from Fig. 5 that the accuracy of WD is significantly deteriorated even for a middle line.

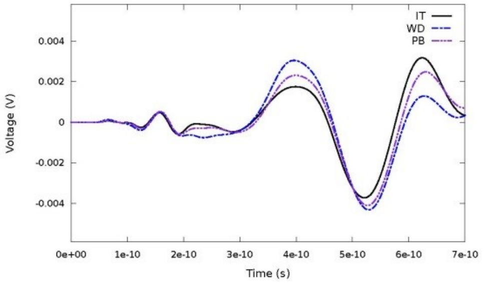

Fig. 3: Voltage response at the far end of line 640 (the middle line of the third layer) for a sparsity ratio ε = 94.9%.

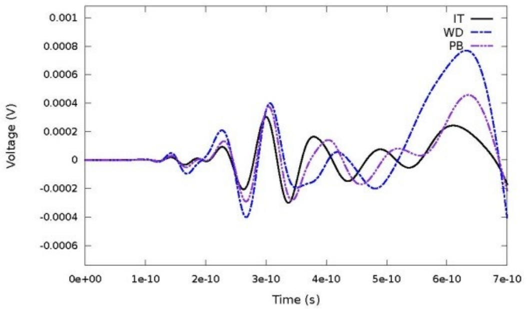

Fig. 4: Voltage response at the far end of line 1023 (the most faraway line from the active line) for a sparsity ratio ε=94.9%.

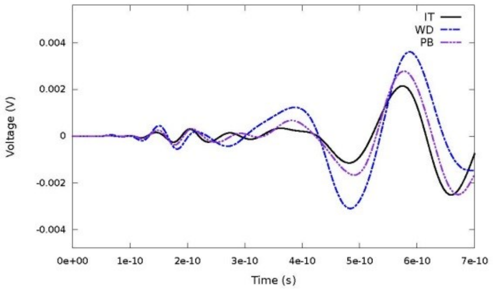

Fig. 5: Voltage response at the far end of line 640 (the middle line of the third layer) for a sparsity ratio ε = 97.4%.

As is also observed from Table 1, the WD method exhibits a super-linear increase in construction time as the sparsity ratio of $\tilde{\mathbf{K}}$ decreases, and for the lowest ratio of ε = 94.9% the construction time of WD even exceeds the construction time of the IT method. On the other hand, the PB method exhibits only a linear increase in construction time with the decrease of the sparsity ratio, since it evidently approaches the construction time of the IT method as ε approaches 0.0%.

A further demonstration of the efficiency of the proposed methodology is presented in Table 2, which reports the number of colors that result from the graph coloring phase of the PB algorithm. As remarked in Section 3, the number of colors equals the number of linear systems that are be solved, while for the full matrix inversion inherent in the IT method the number of linear systems equals the dimension of the inductance matrix (8192 for the design under consideration). It is seen from Table 2 that the PB method is able to achieve a reduction factor ranging between 6.94X and 33.16X in the number of linear systems, requiring the solution of only 1180 linear systems for the lowest (and most accurate) sparsity ratio that was considered. This reduction, which is expected to become even greater as the matrix size increases, is a testament for the low computational and memory requirements of the proposed methodology. The total number of colors of the PB method can be further reduced if more advanced graph coloring algorithms are employed, which can provide fewer colors and thus fewer linear systems to be solved.

Table 1: Execution time for constructing the sparse approximation of the reluctance matrix in the three methods under comparison (PB, WD, and IT). Spd_{WD} and Spd_{IT} denote the speedup of PB over WD and IT respectively.

	PB	WD	IT	Spd_{WD}	Spd_{IT}
98.7%	543.93	43.16	2395.33	0.08X	4.40X
97.4%	606.45	358.91	2395.33	0.59X	3.95X
96.2%	700.06	1124.94	2395.33	1.60X	3.42X
94.9%	794.52	9778.83	2395.33	12.30X	3.01X

Table 2: Number of linear systems required by the PB and IT methods. Red_{IT} denotes the reduction factor of PB over IT.

	PB	IT	Red_{IT}
98.7%	247	8192	33.16X
97.4%	512	8192	16.00X
96.2%	860	8192	9.52X
94.9%	1180	8192	6.94X

5. CONCLUSION

We have presented an efficient and accurate methodology for constructing a sparse approximation of the inverse of the inductance matrix for use in RLC simulation of large-scale linear circuits. The methodology is centered on the proposed "Probing Approximate Inverse" algorithm, which can perform fast and accurate selective inversion of a fully dense matrix to a prescribed sparsity ratio, and also offers substantial degree of parallelism for further acceleration on parallel architectures. Evaluation of our methodology on a bus structure with 1024 wires showed that it is much faster and accurate than the wire duplication method (and window-based methods in general), for the relatively low sparsity ratios needed in practice to achieve the necessary accuracy.

REFERENCES

[1] A. Devgan, H. Ji, and W. Dai, "How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element *K*", *IEEE/ACM Inf. Conf. Computer-Aided Design* (*ICCAD*), 2000.

[2] M. Beattie and L. Pileggi, "Modeling Magnetic Coupling for On-Chip interconnect", *IEEE/ACM Design Automation Conf.* (*DAC*), 2001.

[3] H. Zheng, B. Krauter, M. Beattie, and L. Pileggi, "Window-Based Susceptance Models for Large-Scale RLC Circuit Analyses", *IEEE/ACM Design, Automation and Test in Europe* (*DATE*), 2002.

[4] T. Chen, C. Luk, and C. Chen, "Inductwise: Inductance-Wise Interconnect Simulator and Extractor", *IEEE Trans. Computer-Aided Design*, vol. 22, no. 7, pp. 884-894, 2003.

[5] G. Zhong, C. Koh, and K. Roy, "On-Chip Interconnect Modeling by Wire Duplication", *IEEE Trans. Computer-Aided Design*, vol. 22, no. 11, pp. 1521-1532, 2003.

[6] H. Li, V. Balakrishnan, C. Koh, and G. Zhong, "Compact and Stable Modeling of Partial Inductance and Reluctance Matrices", *IEEE/ACM Asia and South Pacific Design Automation Conf.* (*ASP-DAC*), 2005.

[7] H. Li, V. Balakrishnan, and C. Koh, "Stable and Compact Inductance Modeling of 3-D Interconnect Structures", *IEEE/ACM Inf. Conf. Computer-Aided Design* (*ICCAD*), 2006.

[8] M. Kubale, *Graph Colorings*, American Mathematical Society, 2004.

[9] Z. He, M. Celik, and L. T. Pileggi, "SPIE: Sparse Partial Inductance Extraction", *IEEE/ACM Design Automation Conf.* (*DAC*), 1997.

[10] M. Kamon, M.J. Tsuk, and J. K. White, "FastHenry: A Multipole-Accelerated 3-D Inductance Extraction Program", *IEEE/ACM Design Automation Conf.* (*DAC*), 1993.

[11] R. Jiang and C. Chen, "SCORE: SPICE Compatible Reluctance Extraction", *IEEE/ACM Design, Automation and Test in Europe* (*DATE*), 2004

[12] H. Ji, Q. Yu, and W. Dai, "SPICE Compatible Circuit Models for Partial Reluctance K", *IEEE/ACM Asia and South Pacific Design Automation Conf.* (*ASP-DAC*), 2004

[13] R. Horn and C. Johnson, *Topics in matrix analysis*, Cambridge Univ. Press, 1991.

[14] G. Alleon, M. Benzi, and L. Giraud, "Sparse approximate inverse preconditioning for dense linear systems arising in computational electromagnetics", *Numerical Algorithms*, vol.16, pp. 1-15, 1997.

[15] A. Gebremedhin, F. Manne, and A. Pothen, "What color is your Jacobian? Graph coloring for computing derivatives", *SIAM Review*, vol. 47, pp. 629-705, 2005.

[16] A. Ruehli, "Equivalent circuit models for three dimensional multi-conductor systems", *IEEE Trans. on Microwave Theory and Techniques*, vol. 22, pp. 216-221, 1974