A Fast Layer Elimination Approach for Power Grid Reduction *

Abdul-Amir Yassine
ECE Department
University of Toronto
Toronto, Ontario, Canada
abed.yassine@mail.utoronto.ca

Farid N. Najm
ECE Deptartment
University of Toronto
Toronto, Ontario, Canada
f.najm@utoronto.ca

ABSTRACT

Simulation and verification of the on-die power delivery network (PDN) is one of the key steps in the design of integrated circuits (ICs). With the very large sizes of modern grids, verification of PDNs has become very expensive and a host of techniques for faster simulation and grid model approximation have been proposed. These include topological node elimination, as in TICER and full-blown numerical model order reduction (MOR) as in PRIMA and related methods. However, both of these traditional approaches suffer from certain drawbacks that make them expensive and limit their scalability to very large grids. In this paper, we propose a novel technique for grid reduction that is a hybrid of both approaches–the method is numerical but also factors in grid topology. It works by eliminating whole internal layers of the grid at a time, while aiming to preserve the dynamic behavior of the resulting reduced grid. Effectively, instead of traditional node-by-node topological elimination we provide a numerical layer-by-layer block-matrix approach that is both fast and accurate. Experimental results show that this technique is capable of handling very large power grids and provides a 4.25x speed-up in transient analysis.

Keywords

Power grid, Model order reduction, Graph sparsification

1. INTRODUCTION

On-die power grid integrity checking is one of the key steps in the design process of complex ICs. Integrity checking ensures that the voltage drop at any node in the grid does not exceed a certain threshold, otherwise, the integrated circuit will not perform as intended. This can be done through simulation and/or verification of the power grid. However, as the grid size increases (to around a billion nodes), simulation and verification of such a grid become computationally expensive and almost impossible to implement given the existing resources. Given that all the logic circuitry is connected to the lower metal layers of a power grid, then the safety (integrity) of the nodes in those layers is what matters to ensure a reliable performance. To this end, model order reduction (MOR) has been extensively used in the studies of IC designs for the past two decades. Researchers have been using MOR techniques to reduce the original large power grid systems to much smaller ones, preserving accuracy and timing behavior as much as possible, such that they can be used in further simulations and design verification.

Model order reduction tries to capture the essential features of a system without computing all its details [8]. Several model order reduction techniques have been used or introduced in the past years for the purpose of circuit analysis and power grid reduction. Those techniques can be mainly divided into three different categories: multigrid reduction methods, moment matching techniques based on *Krylov subspaces*, and nodal elimination methods.

Multigrid methods tend to map the problem of a large power grid to some coarse grid (restriction), and then solve that small grid. The solution is then mapped back to the original grid (interpolation). This mapping to smaller grids depends on exploiting the strength of wire connections and the dominant neighbors (strongly connected neighbors) [10]. These methods create significantly smaller grids with relatively sparse system matrices that are easy to solve. However, some multigrid methods require keeping track of the power grid geometry for each multigrid level, which is inefficient for non-uniform grids. Besides, the error in such techniques is usually hard to predict or control.

Moment matching methods are based on explicit *numerical* matching of moments of a system to reduce its order. In other words, assume that the transfer function of the original system is given by:

$$H(s) = M_0 + M_1 s + M_2 s^2 + \dots$$

Then, we try to find a reduced model that matches the first q moments, such that its transfer function is given by:

$$H_q(s) = M_0 + M_1 s + M_2 s^2 + \dots + M_q s^q$$

PRIMA [4], is a well-known algorithm of moment matching, in which the moment space is projected onto an orthonormal subspace called the Krylov subspace. The drawbacks of these methods are that they are not realizable for RC and RLC circuits, and they can produce dense systems. Besides, as the number of port nodes increases, computing higher order moments becomes harder and inefficient.

Nodal elimination techniques are used to *topologically* reduce the number of nodes in the original circuit and approximate the newly added elements in the reduced circuit. One of the well-known nodal elimination methods is the *Time Constant Equilibrium Reduction Scheme*, or TICER [9]. The aim of TICER is to reduce circuits to smaller realizable networks while preserving Elmore delays through RC trees. TICER's key idea is to eliminate a node that has few neighbors and a small time constant, or what the authors call "*quick nodes*". However, the addition of new elements makes the reduced networks denser compared to the original ones. This makes TICER good for tree-like structures of networks observed in circuit timing analysis problems. In other words, TICER cannot be readily applied to mesh-structured chip power grids. The authors in [2] used TICER to eliminate nodes from chip power grids in order to check the safety of those grids. However, the reduction ratio was not very high, and the work couldn't be extended to very large power grids.

*This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

ICCAD'16, November 07-10, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2966986.2966989

Several other works on MOR have been published in the past five years. In [12], the authors geometrically divide the power grid into several blocks, and then they reduce each block to a much smaller one based on *Gaussian Elimination*. Finally, they sparsify each resulting dense block using random sampling of edges and their effective resistances. However, the authors give very little information on their approach for *RC* grids, and most of their results are shown for resistive grids (without capacitances).

In this paper, we propose a novel grid reduction and sparsification technique that exploits the fact that the whole logic circuitry is attached to the bottom metal layer of a power grid. We propose a numerical layer elimination technique for *RC* power grids based on the nodal elimination approach in order to eliminate whole metal layers from a power grid and keep only the important ones. This technique factors in the grid topology when specifying the layers to be eliminated, and when sparsifying the resulting reduced grid while preserving its dynamic behavior.

This paper is organized as follows. In section 2, we present a background on the power grid model used and some concepts and notation used later in the paper. Sections 3 and 4 present our proposed approach and sparsification technique to eliminate layers from resistive and *RC* grids, respectively. In section 5, we propose an incremental approach to eliminate the metal layers. The experimental results are shown in section 6. Finally, section 7 concludes this paper.

2. BACKGROUND AND NOTATION

2.1 Notation

Throughout the rest of this paper, the following notation will be used. Let $(M)_{ij}$ denote the $(i,j)^{th}$ entry of any matrix M, and let $(v)_j$ denote the j^{th} entry of a vector v. We will use the notation $M > 0$ (or $M \geq 0$), for any matrix M, to denote that $(M)_{ij} > 0$ (or $(M)_{ij} \geq 0$), $\forall i,j$.

2.2 Power Grid Model

Consider an *RC* model of a power grid where each metal branch is represented by a resistor and where there exists a capacitor from every node to the ground. Some nodes have ideal current sources (to ground) to represent the current drawn by the underlying circuitry, and some have ideal voltage sources to represent the connections to external power supply. Let the power grid consist of $n+p$ nodes, where nodes $1,2,\ldots,n$ have no voltage sources attached, and the remaining nodes are the nodes where p voltage sources are attached. Let $i(t)$ be the element-wise non-negative vector of all current sources connected to the grid. We assume that $\forall k = 1,\ldots,n$, the entry $(i(t))_k$ is well-defined, so that nodes with no current source attached have $(i(t))_k = 0$. Furthermore, let $\mathcal{N}(i)$ denote the set of all neighboring nodes of a node i, where a node j is considered a neighboring node of i if and only if it is directly connected to node i through a metal branch. Besides, let $g_{(ij)}$ denote a physical conductance between two nodes, i and j, with $g_{(ij)} = 0$ if the two nodes are not directly connected, and $g_{(i)}$ denote the total incident conductance at node i, i.e.:

$$g_{(i)} = \sum_{j \in \mathcal{N}(i)} g_{(ij)}$$

Let $c_{(i)}$ denote a capacitance connected from node i to ground. Finally, Let $u(t)$ be the vector of all nodal voltages, and $v(t) = V_{dd} - u(t)$ be the vector of voltage drops, where V_{dd} is an $n \times 1$ constant vector each entry of which is equal to the ideal supply voltage source value. Then the RC model for the power grid can be written as [3]:

$$Gv(t) + C\dot{v}(t) = i(t) \tag{1}$$

where G is an $n \times n$ conductance matrix, and C is an $n \times n$ diagonal non-singular matrix consisting of all node-to-ground capacitances.

Note that this equation can be obtained directly by writing the Nodal Analysis (NA) system for a modified network in which all voltage sources are shorted (set to 0) and all current sources are reversed. In the rest of this paper, we are going to assume this modified network topology. Moreover, note that a resistive power grid is modeled in the same way by neglecting all capacitances, and the model can be written as:

$$GV = I \tag{2}$$

where V and I are $n \times 1$ time-independent vectors representing the DC node voltage drops and currents, respectively.

The matrix G is known to be symmetric and diagonally-dominant with positive diagonal entries and non-positive off-diagonal entries, and it is defined as follows, $\forall i,j \in \{1,2,\ldots,n\}$,

$$(G)_{ij} = \begin{cases} g_{(i)} & i = j \\ -g_{(ij)} & i \neq j \end{cases} \tag{3}$$

Assuming the grid is strongly connected and there is at least one external voltage source, then G is known to be irreducibly diagonally dominant. With these properties, G is known to be a so-called $\mathcal{M}$-matrix, so that G^{-1} exists and is non-negative ($G^{-1} \geq 0$).

2.3 Node Elimination

Throughout this paper, we will refer to the topological approach to eliminate nodes from a resistive power grid by *Node Elimination*. This topological approach is based on the well-known $Y - \Delta$ transformation technique for circuit analysis [6]. Suppose that node i is to be eliminated, then the elimination process is as follows:

- Remove all conductances connecting node i to its neighbors.
- For each two nodes $j,k \in \mathcal{N}(i)$, insert a new conductance between them given by:

$$g_{(jk)_{new}} \triangleq g_{(jk)_{old}} + \frac{g_{(ij)}g_{(ik)}}{g_{(i)}} \tag{4}$$

- Remove node i.

This elimination of node i does not change the surrounding nodes' voltages.

3. PROPOSED APPROACH

In [11], the authors use a 2×2 block matrix representation for the NA conductance matrix of their *macromodels* in order to perform macromodeling and get the port admittance matrix of these macromodels using Gaussian Elimination. In this section, we divide the grid into three regions and generalize the work done in [11] to a 3×3 block-matrix Gaussian Elimination. Since the resulting grid is dense, we propose a topological technique for power grid sparsification based on the relation between the effective resistance between two nodes in the grid and spatial locality.

3.1 Resistive Grids and Layer Elimination

Let us divide a resistive grid into 3 blocks; one representing the *top layers* of the grid that should be kept (the topmost layer is connected to the C4 bumps), another representing the layers that are desired to be eliminated (*middle layers*), and finally a block representing the lower (*bottom*) layers that are connected to the current sources.

Let n_l be the total number of nodes in the layers desired to be eliminated (*middle layers*), n_t be the number of nodes in the top layers of the grid, and let n_b be the number of nodes in the bottom layers of the grid. Finally, note that $n = n_t + n_l + n_b$, where n is the total number of nodes in the grid. Consider the following representation of the resistive grid conductance matrix of size $n \times n$:

$$G = \begin{bmatrix} G_{11} & G_{12} & 0 \\ G_{12}^T & G_{22} & G_{23} \\ 0 & G_{23}^T & G_{33} \end{bmatrix} \tag{5}$$

where G_{22} is an $n_l \times n_l$ conductance matrix that represents the middle metal layers, G_{11} is an $n_t \times n_t$ conductance matrix representing the top layers of the grid, and G_{33} is an $n_b \times n_b$ conductance matrix representing the bottom layers of the grid. G_{12}

and G_{23} are non-positive, non-square matrices that represent the vias between the middle layers and each of the top and bottom layers, respectively, where G_{12} is of size $n_t \times n_l$, and G_{23} is of size $n_l \times n_b$. Note that G_{11}, G_{22} and G_{33} are all $\mathcal{M}$-matrices (their inverses are non-negative), since any principal sub-matrix of an $\mathcal{M}$-matrix is also an $\mathcal{M}$-matrix [5].

From (2), one can get the DC voltage drops $V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$ at all nodes in the grid by solving:

$$GV = I \tag{6}$$

where V_1 is of size $n_t \times 1$, V_2 is of size $n_l \times 1$, and V_3 is of size $n_b \times 1$. $I = \begin{bmatrix} 0 \\ 0 \\ I_3 \end{bmatrix}$ is an $n \times 1$ vector representing the DC current values drawn from each node in the grid to ground. Note that I_3 is of size $n_b \times 1$, and the zeros are vectors of sizes $n_t \times 1$ and $n_l \times 1$.
Combining (5) and (6), and solving for V_2, we get:

$$V_2 = -G_{22}^{-1}(G_{12}^T V_1 + G_{23} V_3) \tag{7}$$

From (5) and (6), we also have:

$$G_{11} V_1 + G_{12} V_2 = 0 \tag{8}$$

Substituting (7) into (8), we get:

$$(G_{11} - G_{12} G_{22}^{-1} G_{12}^T) V_1 - G_{12} G_{22}^{-1} G_{23} V_3 = 0 \tag{9}$$

Following the same procedure, we also get:

$$-G_{23}^T G_{22}^{-1} G_{12}^T V_1 + (G_{33} - G_{23}^T G_{22}^{-1} G_{23}) V_3 = I_3 \tag{10}$$

Let:

$$\begin{aligned} \hat{G}_{11} &\triangleq G_{11} - G_{12} G_{22}^{-1} G_{12}^T \\ \hat{G}_{13} &\triangleq -G_{12} G_{22}^{-1} G_{23} \\ \hat{G}_{33} &\triangleq G_{33} - G_{23}^T G_{22}^{-1} G_{23} \end{aligned} \tag{11}$$

Then, the reduced system matrix can be represented by:

$$\hat{G} = \begin{bmatrix} \hat{G}_{11} & \hat{G}_{13} \\ \hat{G}_{13}^T & \hat{G}_{33} \end{bmatrix} \tag{12}$$

where $\hat{G}$ is of size $\hat{n} \times \hat{n}$, and $\hat{n} = n - n_l = n_t + n_b$. Note that computing the inverse of G_{22} is not necessary, as the number of non-zero columns in G_{12}^T and G_{23} is much less than the number of columns of G_{22}. Hence, computing $G_{22}^{-1} G_{12}^T$ and $G_{22}^{-1} G_{23}$ directly is much faster.

Finally, the DC voltage drops at all nodes in the reduced grid can be obtained by solving:

$$\hat{G}\hat{V} = \hat{I} \tag{13}$$

where $\hat{V} = \begin{bmatrix} \hat{V}_1 \\ \hat{V}_3 \end{bmatrix}$ is an $\hat{n} \times 1$ vector that represents the DC voltage drops at all nodes in the reduced grid. $\hat{V}_1$ and $\hat{V}_3$ are of sizes $n_t \times 1$ and $n_b \times 1$, respectively. $\hat{I} = \begin{bmatrix} 0 \\ I_3 \end{bmatrix}$ is also an $\hat{n} \times 1$ vector representing the DC current values drawn from each node in the reduced grid to ground. Note that I_3 is still the same as in (6), since the aim of our proposed approach is to keep the layers that are connected to the underlying logic circuitry, which is represented by the current sources attached.

3.2 Sparsification

As the size of the grid increases, the number of nodes in the middle layers increases, and thus; Gaussian Elimination results in rather dense models that are hard to store in memory, and expensive to use in simulations. To overcome this issue, we use some heuristic techniques in order to reduce the number of connections in the reduced model in (12) topologically without creating too much inaccuracies in both the DC and transient simulations. The method is based on the the *effective* resistance between any two nodes in the original grid.

Several methods for sparsification have been implemented in our work that are based on *random walk* or random *sampling of edges*. However, none of them favored the speed of simulations on the reduced grids. Based on the spatial locality, the closeness to the boundaries and the relation between the top and bottom layers of the grid, we propose a method that captures the important connections in the reduced grid while preserving the electrical properties of the original one as much as possible. Using the concept of *spatial locality*, which states that a supply connection to the grid affects the most the current passing through the nodes closest to it [1], one can say that there is a correlation between the effective resistance between any two nodes and the length of the path between those nodes. In other words, as the nodes become spatially closer to each other, the effective resistance between them becomes smaller. Accordingly, we keep the connections between nodes in the same layer that lie in a small neighborhood defined by the user. Empirical results have shown that nodes that lie within two to three metal lines away from a specific node have low effective resistance among them, and thus, we will use such a neighborhood in our experiments. Clearly, the larger the neighborhood, the denser the final sparsified grid will be, and more accurate of course. Hence, there is a clear trade-off between the sparsity of the grid and simulations speed, and the accuracy of such simulations.

For connections between different metal layers, we tend to keep them, as those connections tend to have less resistance than the branch ones, which means less effective resistance.

4. EXTENSION TO RC GRIDS

In this section, we will extend grid reduction to RC power grids. Our main contribution is to reduce an RC grid numerically to eliminate multiple layers at once, rather than topologically on a node by node basis. We introduce a method to approximate the reduced capacitance matrix such that we have a realizable grid reduction technique. The approximation is based on the work done in [9] and [2]. We give a proof to show that eliminating all the layers at once using our approach and eliminating them on a node by node basis produce the same reduced system.

Let C be the capacitance matrix defined in section 2. Assume that it is partitioned as follows:

$$C = \begin{bmatrix} C_1 & 0 & 0 \\ 0 & C_2 & 0 \\ 0 & 0 & C_3 \end{bmatrix} \tag{14}$$

where C_1, C_2 and C_3 are all diagonal matrices representing the capacitances connected from each node in the top, middle and bottom layers, respectively, to ground. Note that C_1 is of size $n_t \times n_t$, C_2 is of size $n_l \times n_l$ and C_3 is of size $n_b \times n_b$.

In [2], the authors used the TICER algorithm [9] to account for the capacitance connected to a node being eliminated using Node Elimination while approximately preserving the time constant of the circuit. TICER distributes the capacitance among its neighbors in a weighted ratio of conductances. Analytically, this translates to:

$$c'_{(j)} \triangleq c_{(j)} + \frac{c_{(i)} g_{(ij)}}{g_{(i)}} \quad \forall j \in \mathcal{N}(i) \tag{15}$$

where i is the node being eliminated, and $c'_{(j)}$ is the updated value of the capacitance connected from node j to ground. This approximation assumes that the node being eliminated is a so-called *quick node* [9], i.e.:

$$2\pi f_{max} \tau_{(i)} << 1$$

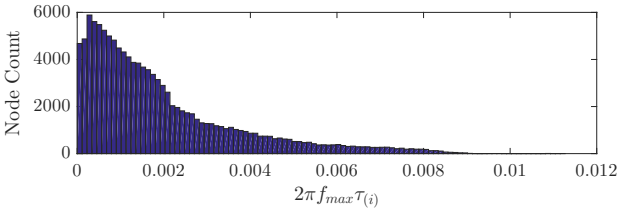

Figure 1: Quick Node property of a power grid of size $152K$ nodes ($f_{max} = 1GHz$).

where f_{max} is the maximum operating frequency of interest, and $\tau_{(i)} = \frac{c_{(i)}}{g_{(i)}}$ is referred to as the time constant of node i. Fortunately, as seen in Fig. 1, the maximum $2\pi f_{max}\tau_{(i)}$ on all the nodes i in metal layers $M2$ to $M8$ in modern chip power grids is very small, less than 0.012. Even if a higher f_{max} is chosen, such as 2 or 3 GHz, it is still clear that the quick nodes approximation is still valid for modern grids. Hence, all the nodes in the layers being eliminated are considered quick nodes.

In this paper, we generalize (15) to eliminate multiple layers at once, as follows. Let c_1 be an $n_t \times 1$ column vector of the capacitors connected from node to ground at every node in the top layers, i.e., c_1 is a column vector of the diagonal entries of C_1. Likewise, let c_2 and c_3 be $n_l \times 1$ and $n_b \times 1$ column vectors of the diagonal entries of C_2 and C_3, respectively.
Let:

$$\hat{c}_1 \triangleq c_1 - G_{12}G_{22}^{-1}c_2 \tag{16a}$$

$$\hat{c}_3 \triangleq c_3 - G_{23}^{T}G_{22}^{-1}c_2 \tag{16b}$$

and let:

$$\hat{C} \triangleq \begin{bmatrix} \hat{C}_1 & 0 \\ 0 & \hat{C}_3 \end{bmatrix} \tag{17}$$

where $\hat{C}_1$ and $\hat{C}_3$ are diagonal matrices consisting of the entries of $\hat{c}_1$ and $\hat{c}_3$, respectively. Claim 1 proves that $\hat{C}$ is the reduced capacitance matrix of the grid obtained by applying (15) sequentially, while eliminating the desired nodes.

Claim 1. *Applying* (16) *when eliminating all q nodes of the middle layers gives the same capacitance values as applying* (15) *sequentially q times for the nodes in the middle layers, while using Node Elimination.*

Proof. We will prove that the new capacitances connected to the top layers are exactly the same in both cases. In other words, we are will prove that (16a) and (15) are equivalent. The proof for the bottom layers' capacitances (16b) follows the same structure, and is skipped due to lack of space. The proof is by induction on the number of nodes being eliminated. Throughout the rest of the proof, the following notation will be used.

Let G be the $n \times n$ conductance matrix as defined in (3). Define $G^{(q)}$ to be a q-partition of G, where a q-partition is a block-matrix representation of G such that the number of nodes to be eliminated is q, and the indexing of the nodes is as follows: all the nodes before the q nodes (*top nodes*) have indices $1, 2, \ldots, \tilde{n}_t$, all the nodes to be eliminated have indices $\tilde{n}_t + 1, \ldots, \tilde{n}_t + q$, and all nodes after the q nodes (*bottom nodes*) have indices $\tilde{n}_t + q + 1, \ldots, n$. Analytically,

$$G^{(q)} = \begin{bmatrix} G_{11}^{(q)} & G_{12}^{(q)} & G_{13}^{(q)} \\ G_{12}^{(q)^T} & G_{22}^{(q)} & G_{23}^{(q)} \\ G_{13}^{(q)^T} & G_{23}^{(q)^T} & G_{33}^{(q)} \end{bmatrix} \tag{18}$$

This means that $G_{22}^{(q)}$ consists of all the q nodes to be eliminated. The sizes of the partitions are as follows: $G_{11}^{(q)}$ is $\tilde{n}_t \times \tilde{n}_t$, $G_{12}^{(q)}$ is $\tilde{n}_t \times q$, $G_{22}^{(q)}$ is $q \times q$, $G_{23}^{(q)}$ is $q \times \tilde{n}_b$, $G_{13}^{(q)}$ is $\tilde{n}_t \times \tilde{n}_b$, and $G_{33}^{(q)}$ is $\tilde{n}_b \times \tilde{n}_b$, where $\tilde{n}_b = n - (\tilde{n}_t + q)$.

Recall that in (5), $G_{13} = 0$. This is due to the fact that in (5), G_{22} consists of a whole metal layer (or multiple layers), and in the structure of chip power grids, each metal layer is connected to only the layers above and below it directly. This means that there are no physical connections between the top and bottom layers, and $G_{13} = 0$. Hence, if the q nodes being eliminated form a whole layer, then $G_{13}^{(q)} = 0$. In fact, if $q = n_l$, then $\tilde{n}_t = n_t, \tilde{n}_b = n_b$, and the block-matrix representations in (5) and (18) are equivalent.

Furthermore, let $\hat{G}^{(q)}$ be the resulting $(\tilde{n}_t + \tilde{n}_b) \times (\tilde{n}_t + \tilde{n}_b)$ conductance matrix after applying Gaussian Elimination on $G_{22}^{(q)}$, i.e., let $\hat{G}_{11}^{(q)}, \hat{G}_{13}^{(q)}$ and $\hat{G}_{33}^{(q)}$ be such that:

$$\begin{aligned} \hat{G}_{11}^{(q)} &= G_{11}^{(q)} - G_{12}^{(q)}G_{22}^{(q)^{-1}}G_{12}^{(q)^T} \\ \hat{G}_{13}^{(q)} &= G_{13}^{(q)} - G_{12}^{(q)}G_{22}^{(q)^{-1}}G_{23}^{(q)} \\ \hat{G}_{33}^{(q)} &= G_{33}^{(q)} - G_{23}^{(q)^T}G_{22}^{(q)^{-1}}G_{23}^{(q)} \end{aligned} \tag{19}$$

Then:

$$\hat{G}^{(q)} = \begin{bmatrix} \hat{G}_{11}^{(q)} & \hat{G}_{13}^{(q)} \\ \hat{G}_{13}^{(q)^T} & \hat{G}_{33}^{(q)} \end{bmatrix}$$

Note that finding the above expression for $\hat{G}_{13}^{(q)}$ follows the same structure of the Gaussian Elimination mentioned in section 3.1, and is skipped due to lack of space.

Finally, let us define $C^{(q)}$, a q-partition of C, such that $c_1^{(q)}, c_2^{(q)}$ and $c_3^{(q)}$ are the $\tilde{n}_t \times 1$, $q \times 1$ and $\tilde{n}_b \times 1$ vectors representing the capacitances from each node to ground in the corresponding partitions of the grid. And, let $\hat{c}_1^{(q)}$ and $\hat{c}_3^{(q)}$ be the resulting $\tilde{n}_t \times 1$ and $\tilde{n}_b \times 1$ vectors after the elimination of q nodes using (16), respectively.

The structure of the proof is as follows:

- Base step: Form a 1-partition of G and C, and prove that (15) and (16a) are equivalent.
- Inductive step: For any integer $q > 1$,
  1. Form a $(q-1)$-partition of G and C, and reduce the system using (19) and (16) to get G' and C'. Assume that (15) and (16a) are equivalent when eliminating $q-1$ nodes.
  2. Eliminate 1 more node using node elimination and (15) from G' and C' to get C'' and its partitions c_1'' and c_3''.
  3. Form a q-partition of G and C using the $(q-1)$ nodes and the additional node used in steps 1 and 2. Apply (16a) to get $\hat{c}_1^{(q)}$.
  4. Prove that $c_1'' = \hat{c}_1^{(q)}$.

Base Step: Let node $k = \tilde{n}_t + 1$ be the node we need to eliminate, and let $G^{(1)}$ and $C^{(1)}$ be the 1-partitions made on G and C to eliminate node k. Then, $G_{22}^{(1)} = g_{(k)}$, $c_2^{(1)} = c_{(k)}$, and:

$$(G_{12}^{(1)})_{jk} = \begin{cases} -g_{(jk)} & \text{if } j \text{ is a neighbor of } k \\ 0 & \text{otherwise} \end{cases}$$

Applying (16a), we get:

$$\hat{c}_1^{(1)} = c_1^{(1)} - G_{12}^{(1)}G_{22}^{(1)^{-1}}c_2^{(1)} = c_1^{(1)} - \frac{c_{(k)}}{g_{(k)}}G_{12}^{(1)}$$

which translates to:

$$(\hat{c}_1^{(1)})_j = \begin{cases} (c_1^{(1)})_j + \frac{c_{(k)}g_{(jk)}}{g_{(k)}} & \text{if } j \text{ is a neighbor of } k \\ (c_1^{(1)})_j & \text{otherwise} \end{cases} \tag{20}$$

On the other hand, applying (15) to get the new capacitances at all top nodes, we get:

$$(\hat{c}_1^{(1)})_j = \begin{cases} (c_1^{(1)})_j + \frac{c_{(k)}g_{(jk)}}{g_{(k)}} & \text{if } j \text{ is a neighbor of } k \\ (c_1^{(1)})_j & \text{otherwise} \end{cases} \tag{21}$$

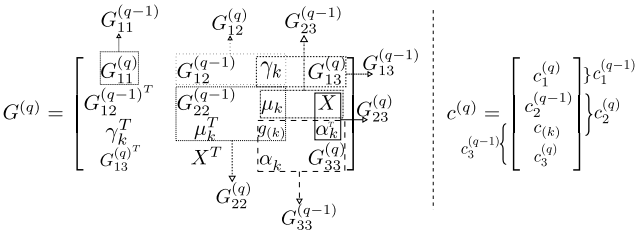

Figure 2: Illustration of $G^{(q)}$ and $c^{(q)}$

We can see that (20) and (21) are the same. This concludes the base step.

Inductive Step: Assume that the claim is true for eliminating $q-1$ nodes, i.e., assume that:

$$\begin{aligned}\hat{c}_1^{(q-1)} &= c_1^{(q-1)} - G_{12}^{(q-1)} G_{22}^{(q-1)} c_2^{(q-1)} \\ \hat{c}_3^{(q-1)} &= c_3^{(q-1)} - {G_{23}^{(q-1)}}^T G_{22}^{(q-1)} c_2^{(q-1)}\end{aligned} \tag{22}$$

are equivalent to applying (15) $q-1$ times sequentially while eliminating $q-1$ nodes using node elimination.

Let the additional node to be eliminated be the first node in the bottom nodes, which has index $k = \tilde{n}_t + q$. Note that eliminating the nodes in any order does not affect the final reduced system (Claim 2 in the appendix gives a proof of this property). Forming the q-partition of G with this additional node, we get:

$$c_1^{(q)} = c_1^{(q-1)} = c_1 \tag{23a}$$

$$c_2^{(q)} = \begin{bmatrix} c_2^{(q-1)} & c_{(k)} \end{bmatrix}^T \tag{23b}$$

$$c_3^{(q-1)} = \begin{bmatrix} c_{(k)} & c_3^{(q)} \end{bmatrix}^T$$

$$G_{11}^{(q)} = G_{11}^{(q-1)}$$

$$G_{12}^{(q)} = \begin{bmatrix} G_{12}^{(q-1)} & \gamma_k \end{bmatrix} \tag{23c}$$

$$G_{13}^{(q-1)} = \begin{bmatrix} \gamma_k & G_{13}^{(q)} \end{bmatrix}$$

$$G_{22}^{(q)} = \begin{bmatrix} G_{22}^{(q-1)} & \mu_k \\ \mu_k^T & g_{(k)} \end{bmatrix} \tag{23d}$$

$$G_{33}^{(q-1)} = \begin{bmatrix} g_{(k)} & \alpha_k^T \\ \alpha_k & G_{33}^{(q)} \end{bmatrix}$$

where γ_k is an $\tilde{n}_t \times 1$ vector representing the connections between the top nodes and node k, μ_k is a $(q-1)\times 1$ vector representing the connections between the block of the first $(q-1)$ nodes being eliminated and node k, and α_k is an $\tilde{n}_b \times 1$ vector representing the connections between the bottom nodes and node k. Note that: $G_{23}^{(q-1)} = \begin{bmatrix} \mu_k & X \end{bmatrix}$, while $G_{23}^{(q)} = \begin{bmatrix} X \\ \alpha_k^T \end{bmatrix}$, where X is a $(q-1)\times \tilde{n}_b$ non-positive matrix. Fig. 2 illustrates how $G^{(q)}$ and $c^{(q)}$ are composed in terms of $G^{(q-1)}$ and $c^{(q-1)}$, respectively.

In what follows, we will give some results that will be helpful in the rest of the proof. From [7], the inverse of any non-singular 2×2 block-matrix $A = \begin{bmatrix} B & E \\ F & C \end{bmatrix}$, where B is non-singular, can be written as:

$$A^{-1} = \begin{bmatrix} B^{-1} + B^{-1}ES^{-1}FB^{-1} & -B^{-1}ES^{-1} \\ -S^{-1}FB^{-1} & S^{-1} \end{bmatrix} \tag{24}$$

where $S = C - FB^{-1}E$ is the well-known *Schur complement* of C in A.

Let $\varphi_k = {G_{22}^{(q-1)}}^{-1} \mu_k$. Then, using (23d) and (24), we can write:

$${G_{22}^{(q)}}^{-1} = \begin{bmatrix} P & \lambda \\ \lambda^T & r \end{bmatrix} \tag{25}$$

where

$$\begin{aligned} P &= {G_{22}^{(q-1)}}^{-1} + r\varphi_k\varphi_k^T \\ \lambda &= -r\varphi_k \\ r &= (g_{(k)} - \mu_k^T\varphi_k)^{-1} \end{aligned} \tag{26}$$

where r is a scalar, λ is a $(q-1)\times 1$ vector, and P is a $(q-1)\times(q-1)$ matrix.
Let:

$$\begin{aligned}\hat{g}_{(k)}^{(q-1)} &\triangleq r^{-1} \\ &= g_{(k)} - \mu_k^T\varphi_k \\ &= g_{(k)} - \mu_k^T {G_{22}^{(q-1)}}^{-1} \mu_k\end{aligned} \tag{27}$$

Moreover, let:

$$\begin{aligned}\hat{\gamma}_k &\triangleq \gamma_k - G_{12}^{(q-1)}\varphi_k \\ &= \gamma_k - G_{12}^{(q-1)} {G_{22}^{(q-1)}}^{-1} \mu_k \\ \hat{c}_{(k)}^{(q-1)} &\triangleq c_{(k)} - \varphi_k^T c_2^{(q-1)} \\ &= c_{(k)} - \mu_k^T {G_{22}^{(q-1)}}^{-1} c_2^{(q-1)}\end{aligned} \tag{28}$$

Note that $\hat{g}_{(k)}^{(q-1)}$ is the total incident conductance at node k after eliminating the first $q-1$ nodes. This can be seen by evaluating $\hat{G}_{33}^{(q-1)}$ using (19):

$$\hat{G}_{33}^{(q-1)} = G_{33}^{(q-1)} - {G_{23}^{(q-1)}}^T {G_{22}^{(q-1)}}^{-1} G_{23}^{(q-1)}$$

Since $g_{(k)} = \left(G_{33}^{(q-1)}\right)_{11}$, and μ_k is the column of $G_{23}^{(q-1)}$ that corresponds to the connections to node k, then

$$\hat{g}_{(k)}^{(q-1)} = \left(\hat{G}_{33}^{(q-1)}\right)_{11} \tag{29}$$

Similarly, we can show that $\hat{c}_{(k)}^{(q-1)}$ is the capacitance connected to node k after eliminating the first $q-1$ nodes by evaluating $\hat{c}_3^{(q-1)}$ using (22):

$$\hat{c}_3^{(q-1)} = c_3^{(q-1)} - {G_{23}^{(q-1)}}^T {G_{22}^{(q-1)}}^{-1} c_2^{(q-1)}$$

Since $c_{(k)} = \left(c_3^{(q-1)}\right)_1$, and μ_k is the column of $G_{23}^{(q-1)}$ that corresponds to the connections to node k, then

$$\hat{c}_{(k)}^{(q-1)} = \left(\hat{c}_3^{(q-1)}\right)_1 \tag{30}$$

Using the same reasoning, one can prove that $\hat{\gamma}_k$ is an $\tilde{n}_t \times 1$ vector representing the connections between the top nodes and node k after eliminating the first $q-1$ nodes.

Going back to the main part of the proof, we have from (16a):

$$\hat{c}_1^{(q)} = c_1^{(q)} - G_{12}^{(q)} {G_{22}^{(q)}}^{-1} c_2^{(q)} \tag{31}$$

Plugging (23a)-(23d) and (25)-(28) into (31), we get:

$$\begin{aligned}\hat{c}_1^{(q)} &= c_1 - \begin{bmatrix} G_{12}^{(q-1)}\gamma_k \end{bmatrix} \begin{bmatrix} P & \lambda \\ \lambda^T & r \end{bmatrix} \begin{bmatrix} c_2^{(q-1)} \\ c_{(k)} \end{bmatrix} \\ &= c_1 - G_{12}^{(q-1)}(Pc_2^{(q-1)} + c_{(k)}\lambda) - \gamma_k(\lambda^T c_2^{(q-1)} + rc_{(k)}) \\ &= c_1 - G_{12}^{(q-1)}\left(({G_{22}^{(q-1)}}^{-1} + r\varphi_k\varphi_k^T)c_2^{(q-1)} - c_{(k)}r\varphi_k\right) \\ &\quad - \gamma_k(-r\varphi_k^T c_2^{(q-1)} + rc_{(k)}) \\ &= c_1 - G_{12}^{(q-1)}\big({G_{22}^{(q-1)}}^{-1} c_2^{(q-1)} + \tfrac{\varphi_k\varphi_k^T c_2^{(q-1)}}{\hat{g}_{(k)}^{(q-1)}} - \tfrac{c_{(k)}\varphi_k}{\hat{g}_{(k)}^{(q-1)}}\big) \\ &\quad - \tfrac{-\varphi_k^T c_2^{(q-1)} + c_{(k)}}{\hat{g}_{(k)}^{(q-1)}}\gamma_k \\ &= c_1 - G_{12}^{(q-1)} {G_{22}^{(q-1)}}^{-1} c_2^{(q-1)} + G_{12}^{(q-1)}\varphi_k \tfrac{c_{(k)} - \varphi_k^T c_2^{(q-1)}}{\hat{g}_{(k)}^{(q-1)}} \\ &\quad - \tfrac{c_{(k)} - \varphi_k^T c_2^{(q-1)}}{\hat{g}_{(k)}^{(q-1)}}\gamma_k \\ &= \hat{c}_1^{(q-1)} - \tfrac{c_{(k)} - \varphi_k^T c_2^{(q-1)}}{\hat{g}_{(k)}^{(q-1)}}(\gamma_k - G_{12}^{(q-1)}\varphi_k) \\ &= \hat{c}_1^{(q-1)} - \tfrac{\hat{c}_{(k)}^{(q-1)}}{\hat{g}_{(k)}^{(q-1)}}\hat{\gamma}_k\end{aligned} \tag{32}$$

Algorithm 1 `Incremental_Elimination`

Inputs: G, C, h, l, δ
Outputs: $\hat{G}, \hat{C}$
1: **for** $k = h-1$ **down to** $l+1$ **do**
2: layer_nodes := get nodes of layer k
3: Form partitions of G and C using (5) and (14) based on layer_nodes
4: Compute $\hat{G}$ based on (11) using G
5: Compute $\hat{C}$ based on (17) using C
6: $C := \hat{C}$
7: Use `SPER` to sparsify $\hat{G}$, and put the result in G
8: **end for**
9: $\hat{G} := G$
10: $\hat{C} := C$
11: **return** $\hat{G}, \hat{C}$

which translates to:

$$(\hat{c}_1^{(q)})_j = \begin{cases} (\hat{c}_1^{(q-1)})_j + \dfrac{\hat{c}_{(k)}^{(q-1)} \, |(\hat{\gamma}_k)_j|}{\hat{g}_{(k)}^{(q-1)}} & \text{if } j \text{ is a neighbor of } k \\ (\hat{c}_1^{(q-1)})_j & \text{otherwise} \end{cases} \tag{33}$$

On the other hand, let $G' = \hat{G}^{(q-1)}$ and $C' = \hat{C}^{(q-1)}$, applying node elimination then (15) to eliminate the same node k, and get the new capacitances at all nodes in the top nodes c_1'', we get:

$$(c_1'')_j = \begin{cases} (c_1')_j + \dfrac{c'_{(k)} g'_{(kj)}}{g'_{(k)}} & \text{if } j \text{ is a neighbor of } k \\ (c_1')_j & \text{otherwise} \end{cases} \tag{34}$$

Using the fact that $G' = \hat{G}^{(q-1)}$ and $C' = \hat{C}^{(q-1)}$, we can see from (29) that $g'_{(k)} = \hat{g}_{(k)}^{(q-1)}$. Similarly, from (30), we can see that $c'_{(k)} = \hat{c}_{(k)}^{(q-1)}$. It can also be shown using a similar reasoning that $g'_{(kj)} = |(\hat{\gamma}_k)_j|, \forall j$. Hence, we can see that $c_1'' = \hat{c}_1^{(q)}$, and this completes the proof. □

5. INCREMENTAL ELIMINATION

Looking at the reduced conductance matrix in (11) and the approximate reduced capacitance matrix in (16), we can see that they involve computing G_{22}^{-1}, which represents all the nodes in the layers to be eliminated. As the size of the grid and number of layers increase, the number of nodes to be eliminated becomes larger. Clearly, computing the inverse will take more CPU time, and the result may not fit in memory anymore. To this end, we propose an incremental layer-by-layer approach to the elimination process. The idea is to eliminate one layer at a time and sparsify the reduced grid along the way until we eliminate all the desired layers. This should reduce the time needed for elimination and the memory needed in the process. Note that the quick node property has been checked after each elimination process, and it holds. In section 6, we will compare both approaches, the direct and incremental ones. We will see that the incremental approach takes much less time, and it allows us to use very large sizes of grids that we couldn't test with the direct approach. **Algorithm** 1 summarizes our incremental elimination approach to eliminate k layers. The algorithm takes as input the original grid conductance matrix G, the original grid capacitance matrix C, the index of the lowest top layer h and the index of the topmost bottom layer l such that $\forall k \in \{l+1, \ldots, h-1\}$, layer k is eliminated. It also requires δ, the sparsification lower threshold on conductances to keep. The output of the algorithm is $\hat{G}$ and $\hat{C}$ which are the reduced grid conductance and capacitance matrices, respectively. $SPER$ is the sparsification method described in section 3.2.

Table 1: Comparison between Direct and Incremental Elimination. Grid Size: $150K$ nodes

Measure	Direct Elimination	Incremental Elimination
Sparsity	0.279%	0.1833%
ET (min)	11.11	0.7
SPT (sec)	11.09	33.974
Speed-up in DC solve	1.57x	1.71x
Average Error (mV)	1.907	1.943
Maximum Error (mV)	15.44	16.17

Table 2: Numerical Elimination v.s. Topological Elimination

Original Grid Size	Numerical Elimination			Topological Elimination		
	Reduced Size	*ET + SPT*	*Reduction Ratio(%)*	*Reduced Size*	*RT*	*Reduction Ratio (%)*
$152K$	$33K$	$1.27m$	78.3	$44.7K$	$3.4m$	70.6
$342K$	$74K$	$6.54m$	78.4	$100.4K$	$16.3m$	70.6
$609K$	$133K$	$25.2m$	78.2	$179K$	$50.7m$	70.6

6. EXPERIMENTAL RESULTS

In this section, we will present some results from tests run on the proposed Layer Elimination method with sparsification. First, we will compare the direct and incremental approaches for layer elimination by comparing their DC responses and elimination time. Next, we will test the reduced RC grid by comparing its transient voltage drops to those computed on the original grid.

A `C++` implementation was written to perform the above tests. Our test grids were generated based on user specifications such as the grid dimensions, number of layers, layers' geometrical specifications and current source distributions. The generated grids are consistent with a $45nm$ technology. All tests were run on a Linux machine with 3.4 GHz Intel core i7-4770 processor with 8 cores and 32GB of RAM. In all tests, the number of metal layers in the grids was set to 8. The number of layers eliminated was 6, keeping only layers M8 that is attached to the C4 bumps (top layer) and M1 that is attached to the current sources (bottom layer). The model assumes that there is a current source attached to each non-boundary node in M1.

Whenever we compare two things by looking at the error, we mean the error at the nodes of layer $M1$ (that are attached to current sources) in the reduced grid relative to the exact values of those in the original grid. The comparison may be based on the DC voltage drops or the transient voltage drops in an RC grid.

6.1 Direct vs. Incremental Elimination

In this section, we present a comparison between the direct and the incremental elimination processes discussed in sections 3.1 and 5, respectively. Table 1 presents such a comparison on a grid of size $150K$ nodes. The size of the reduced system is $33K$ nodes. For each approach, the table shows the sparsity of the reduced resistive systems (the sparsity of the original system is 0.00261%), the elimination time (ET), the sparsification time (SPT) using SPER, the speed-up in the DC solve of the systems compared to the original system, and the average error of the DC voltage drops at layer $M1$. The sparsity of any $n \times m$ matrix A is calculated as follows:

$$Sparsity(A) = \frac{nnz(A)}{nm} \times 100\%$$

where $nnz(A)$ is the number of non-zeros in the matrix A. We can see that the elimination time of the direct approach is around 15 times that of the incremental approach. This is due to the fact that in the direct approach, the size of the matrix being inverted is much larger than those being inverted in the gradual process. However, the sparsification process in the incremental elimination is 3 times slower than the direct one. This is because in the incremental approach, SPER is called as many times as there are layers being eliminated, and in each time, it goes over all the connections in the reduced system. The error in the incremental process is insignificantly larger, however, the speed-up in the DC

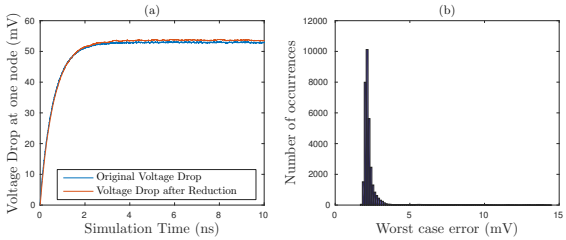

Figure 3: (a): Voltage Drop Waveforms at one node in the grid (b): Histogram of the error over all nodes

solve is better. Finally, It is worth mentioning that the direct simulation stopped working on grids of size beyond $300K$ nodes due to lack of memory, while with the incremental simulation, we were able to go up to sizes of the order of $3M$ nodes.

Table 2 compares the reduction time and ratio of our incremental numerical approach with the topological one implemented in [2], which is based on TICER [9]. A lower threshold of 10^{-5} was used on the new conductances to be added to the reduced grids in both cases. The table presents, along with the size of the original grid, the reduction time ($ET + SPT$) in our incremental approach and RT in the topological one. In addition, the table presents the reduction ratio in both approaches, which is the ratio of the size of the reduced grid to that of the original one. From the table, we can see that our incremental approach is almost 2.2 times faster than the topological one. Besides, the reduction ratio in our technique is higher, which is better as we aim to eliminate complete layers and keep only the important nodes. Note that the topological approach broke down after a grid size of $609K$ due to lack of memory, while our approach handled much larger grid sizes.

6.2 Transient Simulations

To find the transient voltage drops on all the nodes in an RC power grid, we created a simple simulation engine for the ODE in (1) based on a standard backward-Euler discretization. All current waveforms were generated randomly. When comparing the voltage drops on the original and reduced grids, we computed the worst-case voltage drop error at each node over the whole simulation time, and then took the average and maximum error over all the nodes.

Fig. 3a shows the voltage drop waveform before and after reduction at one node in a grid of size $152K$. The node lies in the center of layer $M1$, and it is representative because most nodes have low worst-case error associated with them, as can be seen from the histogram of the worst-case errors on all $M1$ nodes in Fig. 3b. We can see that our capacitance approximation is good, since the rate of convergence to steady state is almost the same. In other words, the time constant of both grids is almost the same.

Table 3 shows the transient simulation results for different grid sizes. The length of the simulation was taken to be $120ns$. For each grid, the table presents the transient simulation time (TT) before and after reduction, the elimination and sparsification times (ET and SPT), the speed-up gained after reduction, and the average and maximum worst-case errors on the voltage drops. The speed-up is computed as follows:

$$Speedup = \frac{TT_{original}}{TT_{reduced}}$$

As can be seen in the table, the speed-up gain is around 4.25x. This speed-up is obtained due to our topological sparsification procedure, which keeps only the important connections in the reduced grid. Accordingly, the time taken in one linear solve of the reduced grid is around 4.25 times smaller that of the original grid, and hence, the 4.25x speed-up. Fig. 5 shows that as the simulation time increases, the effect of the reduction overhead diminishes and becomes negligible.

As mentioned before, there is a clear trade-off between the simulation speed-up and the accuracy of the simulations. To see this trade-off, Fig. 4 shows the relative worst-case error rate at the $M1$ nodes versus the actual voltage drop for two cases. In

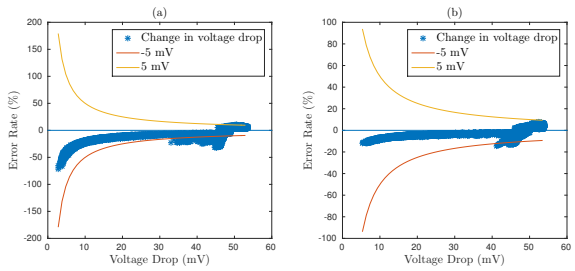

Figure 4: (a): Error rate with layer $M2$ eliminated (b): Error rate while keeping layer $M2$

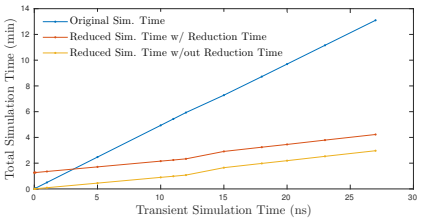

Figure 5: Total Simulation Time before and after reduction

the first case (a), we eliminate 6 out of 8 layers, while in the second case (b), we eliminate only 5 layers by keeping layer $M2$. From the figures, we can see how keeping layer $M2$ affects the errors by restricting them below the $5mV$ hyperbolas. In fact, the average worst-case error while keeping layer $M2$ is $1.2mV$. However, it is worth mentioning that the speed-up in the first case simulations is 4.2x, while in the second one, it is only 2.34x. Moreover, the elimination ratio in the first case is around 80%, while in the second, it is only 46%. This gives an idea on how there is a trade-off with more layers being eliminated.

7. CONCLUSION

Model order reduction (MOR) is one of the common techniques to study the behavior of modern on-die power grids. Traditional methods suffer from many drawbacks that limit their scalability to very large grids. We propose a novel numerical layer-by-layer grid reduction technique that is fast and accurate. We prove using empirical tests that it is scalable to modern grids. Results show that we are able to achieve a 4.25x speed-up in the transient analysis and a $2.4mV$ average worst-case error at the bottom layer of the grid.

8. REFERENCES

[1] E. Chiprout. Fast flip-chip power grid analysis via locality and grid shells. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 485–488, Nov 2004.

[2] A. Goyal and F. Najm. Efficient RC power grid verification using node elimination. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, pages 1–4, March 2011.

[3] M. Nizam, F. N. Najm, and A. Devgan. Power grid voltage integrity verification. In *Proceedings of the 2005 International Symposium on Low Power Electronics and Design, 2005. ISLPED'05.*, pages 239–244. IEEE, 2005.

[4] A. Odabasioglu, M. Celik, and L. T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. In *Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design*, pages 58–65. IEEE Computer Society, 1997.

[5] R. Plemmons and A. Berman. Nonnegative matrices in the mathematical sciences. *Ch*, 6:137, 1979.

[6] R. Powell. *Introduction to electric circuits*. Butterworth-Heinemann, 1995.

Table 3: Power grids reduction time and Transient simulations run time

Original Grid		**Reduced Grid**				*Speedup*	**Average Error in** (mV)	**Worst Case Error in** (mV)
Size	*Transient Time (TT)*	*Size*	*Elimination Time (ET)*	*Sparsification Time (SPT)*	*Transient Time (TT)*			
$152K$	$1.17hr$	$33K$	$43.68s$	$36.92s$	$16.68m$	4.2x	2.4	14.5
$342K$	$2.51hr$	$74K$	$3.71m$	$2.94m$	$35.49m$	4.24x	2.5	19.1
$609K$	$4.34hr$	$133K$	$15.8m$	$9.7m$	$1.01hr$	4.3x	2.4	16.2
$950K$	$8.05hr$	$208K$	$47.25m$	$21.06m$	$1.91hr$	4.21x	2.3	14.3
$1.36M$	$11.22hr$	$299K$	$112.95m$	$43.41m$	$2.64hr$	4.25x	2.3	17.7
$1.86M$	$19.18hr$	$407K$	$4.24hr$	$1.34hr$	$4.25hr$	4.51x	2.3	15.4
$2.43M$	$23.65hr$	$532K$	$7.42hr$	$2.39hr$	$5.11hr$	4.63x	2.4	16.0
$3.08M$	$26.59hr$	$676K$	$9.06hr$	$3.49hr$	$5.73hr$	4.64x	2.39	16.9

[7] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[8] W. H. Schilders, H. A. Van der Vorst, and J. Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer, 2008.

[9] B. N. Sheehan. Realizable reduction of RC networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 26(8):1393–1407, 2007.

[10] H. Su, E. Acar, and S. R. Nassif. Power grid reduction based on algebraic multigrid principles. In *Proceedings of the 40th annual Design Automation Conference*, pages 109–112. ACM, 2003.

[11] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw. Hierarchical analysis of power distribution networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):159–168, Feb. 2002.

[12] X. Zhao, Z. Feng, and C. Zhuo. An efficient spectral graph sparsification approach to scalable reduction of large flip-chip power grids. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2014*, pages 218–223, Nov 2014.

APPENDIX

In this appendix, we prove a result that is used in the proof of Claim 1.

CLAIM 2. *Eliminating q nodes in any order using* (11) *and* (16) *from an RC power grid produces the same reduced RC grid.*

PROOF. Let G_{22} be an $n_l \times n_l$ conductance matrix that represents the connections between the n_l nodes to be eliminated in a specific order. Changing the order of the nodes to be eliminated is achieved by permuting the order of the rows and columns of G_{22}, using $PG_{22}P^T$, where P is a permutation matrix. A permutation matrix P is a matrix obtained by permuting the rows or columns of an identity matrix according to some permutation. Moreover, $P^{-1} = P^T$, because permutation matrices are *unitary* [7].

Let P be any $n_l \times n_l$ arbitrary permutation matrix, and let Q be another permutation matrix of size $n \times n$ given by:

$$Q = \begin{bmatrix} I_{n_t} & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & I_{n_b} \end{bmatrix} \tag{35}$$

where $n = n_t + n_l + n_b$, and I_{n_t} and I_{n_b} are identity matrices of sizes $n_t \times n_t$ and $n_b \times n_b$, respectively. Let G' be a reordered version of G in (5) given by:

$$\begin{aligned} G' &= QGQ^T \\ &= \begin{bmatrix} I_{n_t} & 0 & 0 \\ 0 & P & 0 \\ 0 & 0 & I_{n_b} \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} & 0 \\ G_{12}^T & G_{22} & G_{23} \\ 0 & G_{23}^T & G_{33} \end{bmatrix} \begin{bmatrix} I_{n_t} & 0 & 0 \\ 0 & P^T & 0 \\ 0 & 0 & I_{n_b} \end{bmatrix} \\ &= \begin{bmatrix} G_{11} & G_{12}P^T & 0 \\ PG_{12}^T & PG_{22}P^T & PG_{23} \\ 0 & G_{23}^T P^T & G_{33} \end{bmatrix} \end{aligned} \tag{36}$$

From (36), we have:

$$\begin{aligned} G'_{11} &= G_{11} \\ G'_{12} &= G_{12}P^T \\ G'_{22} &= PG_{22}P^T \\ G'_{23} &= PG_{23} \\ G'_{33} &= G_{33} \end{aligned} \tag{37}$$

To find the reduced grid of G' after elimination, we apply (11) on the partitions in (37). Hence, we get:

$$\begin{aligned} \hat{G}'_{11} &= G'_{11} - G'_{12}G'^{-1}_{22}G'^T_{12} \\ &= G_{11} - G_{12}P^T PG_{22}^{-1}P^T PG_{12}^T \\ &= G_{11} - G_{12}G_{22}^{-1}G_{12}^T \\ &= \hat{G}_{11} \\ \hat{G}'_{13} &= -G'_{12}G'^{-1}_{22}G'_{23} \\ &= -G_{12}P^T PG_{22}^{-1}P^T PG_{23} \\ &= -G_{12}G_{22}^{-1}G_{23} \\ &= \hat{G}_{13} \\ \hat{G}'_{33} &= G'_{33} - G'^T_{23}G'^{-1}_{22}G_{23} \\ &= G_{33} - G_{23}^T P^T PG_{22}^{-1}P^T PG_{23} \\ &= G_{33} - G_{23}^T G_{22}^{-1}G_{23} \\ &= \hat{G}_{33} \end{aligned} \tag{38}$$

where $G'^{-1}_{22} = (PG_{22}P^T)^{-1} = PG_{22}^{-1}P^T$, since P is unitary. Form (38), we can see that $\hat{G} = \hat{G}'$.

Similarly, reordering the matrix C using Q gives the following reordered versions of the vectors c_1, c_2 and c_3:

$$\begin{aligned} c'_1 &= c_1 \\ c'_2 &= Pc_2 \\ c'_3 &= c_3 \end{aligned} \tag{39}$$

Using (37) and (39) in (16), we get:

$$\begin{aligned} \hat{c}'_1 &= c'_1 - G'_{12}G'^{-1}_{22}c'_2 \\ &= c_1 - G_{12}P^T PG_{22}^1 P^T Pc_2 \\ &= \hat{c}_1 \\ \hat{c}'_3 &= c'_3 - G'^T_{23}G'^{-1}_{22}c'_2 \\ &= c_3 - G_{23}^T P^T PG_{22}^1 P^T Pc_2 \\ &= \hat{c}_3 \end{aligned} \tag{40}$$

From (40), we can see that $\hat{C}' = \hat{C}$, and this completes the proof. □