

# Overview of Vectorless/Early Power Grid Verification

## Special Session Paper

Farid N. Najm

ECE Department, University of Toronto  
Toronto, Ontario, Canada  
f.najm@utoronto.ca

### ABSTRACT

The power distribution network of an integrated circuit must be checked throughout the design process to ensure that supply voltage fluctuations do not exceed certain critical thresholds. One way of doing this is by simulation, which requires knowledge of the circuit currents that load the grid. These currents are hard to specify. In many cases, and certainly during early power grid design, they may be simply unknown because the circuit itself may not yet be specified. Vectorless verification refers to the class of techniques, developed over the last 12 years, for verifying the grid in the absence of complete information about the circuit currents.

### 1. INTRODUCTION

We will refer to the power supply and ground distribution network of an integrated circuit (IC), comprising the on-die parasitic  $RC$  components due to lines and vias, as well as the package/board parasitic  $RLC$  components, as simply the *power grid*. A well-designed power grid should guarantee correct circuit functionality at the intended design speed, by delivering well-regulated voltages at all grid nodes. Over the last few decades, technology scaling has led to very large power grids, comprising around a billion nodes for very large chips, and reduced supply voltages of around 1 Volt which, coupled with the increase in overall chip power dissipation, has given us chips that draw over 150 Amperes from the supply. The increased overall current, to be distributed over the much larger metal network, with narrower lines, has made power grid verification indispensable to high-performance chip design.

There has been much research in the recent past on speeding up circuit simulation of the grid, employing various innovations such as reduced order modeling, improved simulation algorithms, often using indirect (iterative) simulation methods but also multigrid techniques, parallelization, and even current signature compression [1] (which is a hybrid between vector-based simulation and vectorless methods). Review and discussion of this body of work is beyond the scope of this paper. Instead, we will focus on *vectorless* verification techniques.

In vectorless verification, it is assumed that users do not know the exact circuit currents that load the grid and, dur-

ing early design, that they do not necessarily know what exact circuitry will be implemented under the grid. Instead, we assume that users can specify *incomplete* information about the circuit currents in the form of *current constraints*. We will review the current constraints framework and a variety of vectorless verification techniques.

### 2. CURRENT CONSTRAINTS

While users may not know the loading currents exactly, or may not yet know the underlying circuitry exactly, it is really never the case that they know *nothing* about them. There is always some engineering judgment, and/or expertise from previous design activities, with this or similar technologies, that users can bring to bear. It is precisely *this* that we seek to capture with the concept of current constraints. These constraints are an abstraction which aims to capture design knowledge, and they fall somewhere between knowing *everything* and knowing *nothing* about the currents and circuits that load the grid.

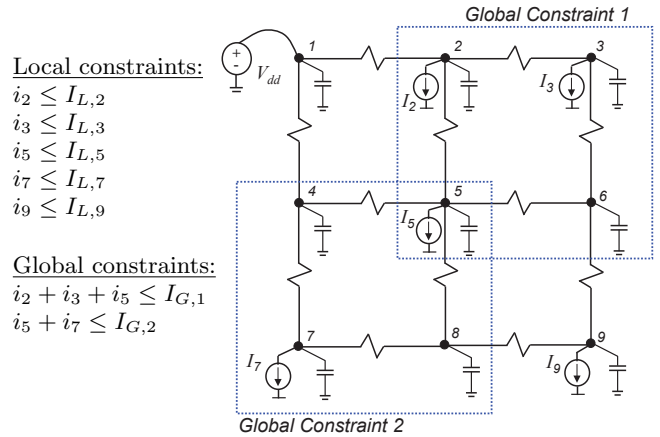


Figure 1: Current constraints

Typically, these constraints are specified as peak current upper-bound limits, or *envelopes*, on every current source that loads the grid. The constraints may be fixed over time (*DC constraints*) or they may be transient waveforms themselves (*transient constraints*). DC constraints are not only easier to work with but are also easier to specify, and so our discussion will focus on the use of DC constraints.

Constraints on single current sources, such as  $i_k(t) \leq I_{L,k}, \forall t \geq 0$ , are referred to as *local constraints* (a current source can represent a single logic gate or cell, but more typically should represent a larger block). If *only* local constraints are provided, the problem becomes much easier, but the results would be overly pessimistic because it is never the case that all chip components simultaneously draw their

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IEEE/ACM International Conference on Computer-Aided Design (ICCAD) 2012, November 5-8, 2012, San Jose, California, USA.

Copyright © 2012 ACM 978-1-4503-1573-9/12/11... \$15.00.

maximum currents, hence the need for what we call *global constraints*. A global constraint is an upper bound on the *sum* of a group of current sources. An example is given in Fig. 1, showing a grid with local and global constraints.

Local and global constraints can be combined into a single matrix inequality,  $Ui(t) \leq I_c$ , where  $i(t)$  is a vector of the current sources that load the grid and  $U$  is a matrix of 0s and 1s, whose top part consists of the identity matrix so that it captures the local constraints, and where  $I_c$  is a vector that contains the (upper-bound) constraint values. We also assume that all currents sources are positive, with reference directions away from the grid, into the circuitry below it. Thus, in general, we will assume that certain local and global constraints are specified, which may be represented succinctly as:

$$0 \leq Ui(t) \leq I_c, \quad \forall t \quad (1)$$

Given such constraints, one can formulate the verification problem as: under all current waveforms that satisfy the constraints, check if the grid voltages satisfy the user specification, i.e., check if the grid is “safe.”

## 2.1 Feasible Space

A current waveform vector  $i(t)$  is said to be *feasible* if it satisfies the current constraints  $0 \leq Ui(t) \leq I_c$  at every time point. We will use the symbol  $\mathcal{F}$  to represent the set of all current vectors  $i$  that satisfy the constraints:

$$\mathcal{F} = \{i : 0 \leq Ui \leq I_c\} \quad (2)$$

We refer to this as the *feasible space* of currents. At the risk of some abuse of notation, we will also write  $i \in \mathcal{F}$  to signify that the transient current waveform vector  $i(t)$  satisfies the current constraints  $0 \leq Ui(t) \leq I_c$  at all points in time:

$$\{i \in \mathcal{F}\} \iff \{0 \leq Ui(t) \leq I_c, \forall t\} \quad (3)$$

## 2.2 Element-wise Operations

It will be helpful to use the following short-hand notation. Let  $f(i) : \mathbb{R}^m \rightarrow \mathbb{R}^n$  be a vector function whose components will be denoted  $f_1(i), f_2(i), \dots, f_n(i)$ , where  $i \in \mathcal{F}$ . Define the vector  $f^{(max)} \in \mathbb{R}^n$  such that, for every  $k$ ,  $f_k^{(max)}$  is the maximum of  $f_k(i)$ , over all  $i \in \mathcal{F}$ . We will denote this vector by the following  $\text{emax}(\cdot)$  operator:

$$f^{(max)} = \text{emax}_{i \in \mathcal{F}}(f(i)) \quad (4)$$

where the “ $\text{emax}(\cdot)$ ” notation is used to denote the fact that this is an element-wise maximization operator. Computationally, this may be found by the application of  $n$  successive optimization steps, as in the following:

$$\begin{aligned} &\text{for}(k = 1, 2, \dots, n) : \\ &\quad \text{Maximize: } f_k(i) \\ &\quad \text{Subject to: } i \in \mathcal{F} \end{aligned} \quad (5)$$

Likewise, we define the operator  $\text{emin}(\cdot)$  to represent element-wise minimization:

$$f^{(min)} = \text{emin}_{i \in \mathcal{F}}(f(i)) \quad (6)$$

Finally, we will use  $\text{eopt}(\cdot)$  to denote an application of both the  $\text{emax}(\cdot)$  and  $\text{emin}(\cdot)$  operators to produce a vector of size  $2n$  that contains the results of both:

$$f^{(opt)} = \text{eopt}_{i \in \mathcal{F}}(f(i)) = \begin{bmatrix} f^{(max)} \\ f^{(min)} \end{bmatrix} = \begin{bmatrix} \text{emax}_{i \in \mathcal{F}}(f(i)) \\ \text{emin}_{i \in \mathcal{F}}(f(i)) \end{bmatrix} \quad (7)$$

## 2.3 Verification by Optimization

We assume that the user has provided voltage thresholds around the nominal supply voltage value that must not be exceeded, for both *voltage drops* (under-shoots) and *voltage over-shoots* (for the inductive case). Therefore, we are interested in the worst-case voltage fluctuations achievable at all grid nodes, over all possible transient current waveforms that satisfy the current constraints.

One way to check node voltage safety is to formulate the problem as a voltage maximization or minimization, thereby looking for *worst-case voltage fluctuations*, which would then be compared against the threshold. The constraints effectively specify a search space of currents for the optimization. If  $v(t)$  is the *voltage drop* vector at time  $t$ , then we are interested in the element-wise worst-case vector:

$$\text{emax}_{i \in \mathcal{F}}(v(t)) \quad (8)$$

Because the constraints are DC, it is clear that the result of the above application of the  $\text{emax}(\cdot)$  operator is *time-independent*. The optimization needs to be performed at *only* one time point, and so we define the time-independent vector  $v^*$ :

$$v^* = \text{emax}_{i \in \mathcal{F}}(v(t)) \quad (9)$$

If  $v^*$  is less than a user-specified threshold, then the grid is safe, otherwise not.

## 2.4 Constraint Specification

How would one obtain/specify these constraints in practice? To be sure, if a logic block is available and small enough to simulate, then one can generate the constraints by an “off-line” process of simulation, which can be viewed as a characterization process. If a block is not yet available or is too large to simulate, then one would need to rely on design expertise with that block (how big it may be, what its power needs were in a previous technology and how scaling would affect those needs, etc.). As a last-resort/fall-back option, if one has some judgment about the power density of the target technology (Watts/ $\mu\text{m}^2$ ) and some knowledge about the target area for this block, then one can multiply the two to get a rough estimate of the current constraint for it.

Another possibility is that the constraints can be used to implement a “spec-based” design flow, as follows. A chip-level designer would simply *specify* the constraints, based on previous design expertise, and the grid is verified under these constraints. The constraints now become *design guidelines* to be obeyed in subsequent design activity. If all design groups follow these guidelines, the final design would have a grid which is *safe by construction*. If there is a need to grant a larger current budget to some group, the chip-level designer would recheck the grid with a new set of constraints and, if necessary, make grid redesign decisions to ensure the increased budget does not make the grid unsafe.

## 3. POWER GRID MODEL

The power grid is a large full-chip structure of connected metal lines, across multiple metal layers, interconnected by vias, and connected by C4 bumps to wiring in the package and on the board, typically terminating at a board-level voltage regulation module (VRM). Excluding the nonlinear (MOSFET) elements of the VRM, this structure is typically modelled as a linear circuit composed of a large number

of lumped linear (*RLC*) elements. It should be mentioned that modern grids include MOSFETs in the on-die portion for purpose of power gating, but such considerations are outside the scope of this study and we will focus on fully passive grid structures. Inductance in the on-die portion of the grid is typically ignored [2], and mutual inductance is also often ignored, but inductance is significant in the package/board portion of the grid and may not be ignored.

At its on-die terminals the grid is loaded by the circuit blocks, where nonlinearities are again encountered due to the circuit MOSFETs. It is practically impossible to jointly simulate or analyze both the full nonlinear circuit and the large grid all at once, and the common practice is to decouple the two. Typically, this means that the circuit blocks are represented by some suitable model, whereby the blocks act as current sources with some nonlinear parasitic network, for purpose of grid simulation or verification. In fact, this nonlinearity is often ignored and the blocks are represented by current sources with linear parasitics. These parasitics are relevant when one considers the impact that grid voltage fluctuation has on the circuit current demands. However, for vectorless verification, because of the larger impact that uncertainty of the currents has on the grid response, this effect is often ignored, and the circuit currents are assumed ideal - and this is what will be assumed for this paper. In any case, it is trivial to extend existing techniques by including *some* current source non-idealities by addition of linear source resistance.

Strictly speaking, the grid is partitioned into two distinct components, the ground network and the power network, so that there is no true “ground” reference node for the grid. It is possible to develop vectorless verification techniques that take this into account [3]. However, in most cases, and because of the near-symmetry of the two portions of the grid, a “virtual ground node” is assumed to represent the half-way point between the two. As a result, almost all published power grid verification methods assume the existence of a global ground node as reference, so that the methods are applicable separately to either the ground network or the supply network. In this framework, all capacitance is from node-to-ground, and coupling capacitance between nodes (within the same network) is ignored.

Early grid verification work, in both academia and industry, was focused on study of the grid under DC conditions, so that a strictly resistive (*R*) grid was considered, and was then extended to dynamic analysis, including the *RC* case where all inductance is ignored, and the *RLC* case. DC verification is useful to quickly discover gross grid design errors. The *RC* case is suitable when mainly debugging the on-die grid, and *RLC* verification is suitable when mostly concerned with the effect of off-die parasitics. In the following, we will describe the *RC* case problem formulation, then restrict that to the *DC* case, then describe the *RLC* case.

### 3.1 The RC Case

Consider an *RC* grid with a capacitor from every node to ground. Some nodes have ideal current sources, to ground, representing the currents drawn by the logic circuits tied to the grid at these nodes. Some other nodes are connected to ideal voltage sources representing the external voltage supply,  $V_{dd}$ . Excluding the ground node, let the power grid consist of  $n + p$  nodes, where nodes  $1, 2, \dots, n$  are the nodes not connected to a voltage source, while the remaining nodes

$(n + 1), (n + 2), \dots, (n + p)$  are the nodes where the  $p$  voltage sources are connected. Let  $i(t) \geq 0$  be the vector of all the  $m$  current sources connected to the grid, whose positive (reference) direction of current is from node-to-ground, assumed to be connected at nodes  $1, 2, \dots, m \leq n$ . Let  $H$  be an  $n \times m$  matrix of 0 and 1 entries that identifies which node is connected to which current source, and let  $i_s(t) = Hi(t)$ . Fig. 1 shows a representative grid structure.

Let  $u(t)$  be the vector of node voltages, relative to ground. By superposition,  $u(t)$  may be found in three steps: 1) open-circuit all the current sources and find the response, which would obviously be  $u^{(1)}(t) = V_{dd}$  in this case, 2) short-circuit all the voltage sources and find the response  $u^{(2)}(t)$ , in this case  $u^{(2)}(t) \leq 0$ , and 3) find  $u(t) = u^{(1)}(t) + u^{(2)}(t)$ . To find  $u^{(2)}(t)$ , KCL at every node easily provides, via Nodal Analysis [4], that:

$$Gu^{(2)}(t) + C\dot{u}^{(2)}(t) = -i_s(t) \quad (10)$$

where  $C \geq 0$  is an  $n \times n$  *diagonal* non-singular matrix consisting of all node-to-ground capacitances and  $G$  is the  $n \times n$  conductance matrix, which is symmetric and diagonally dominant with positive diagonal entries and non-positive off-diagonal entries. Assuming the grid is connected and has at least one voltage source, then  $G$  is known to be irreducibly diagonally dominant. With this, it can be shown that  $G$  is a so-called *M-matrix*, so that  $G^{-1}$  exists and is non-negative,  $G^{-1} \geq 0$ , and all the eigenvalues of  $G$  are real and positive. We are mainly interested in the *voltage drop*  $v(t) = V_{dd} - u(t) = -u^{(2)}(t) \geq 0$ , rather than  $u(t)$ , so that:

$$Gv(t) + C\dot{v}(t) = i_s(t) \quad (11)$$

Note therefore that  $v(t)$  may be found directly from analysis of the circuit in the case when the voltage sources are short-circuited and the current source directions are reversed.

### 3.2 The DC Case

With all currents fixed over time, only grid resistance is relevant and the voltage drop may be obtained by setting to zero the time-derivative in (11), so that:

$$Gv = i_s \quad (12)$$

And the worst-case voltage drop is:

$$v^* = \max_{i_s \in \mathcal{F}} (G^{-1}i_s) \quad (13)$$

### 3.3 The RLC Case

With the introduction of inductance, Nodal Analysis is not enough and we need Modified Nodal Analysis [4] (MNA). The voltage sources are first eliminated outright, as was done for the *RC* case, by using superposition. So, the voltage drop is  $v(t) = V_{dd} - u(t) = -u^{(2)}(t)$ , where  $u^{(2)}(t)$  is the vector of node voltages when the voltage sources are shorted out. In this special case, MNA provides the required system of equations as the combination of KCL at all the nodes and the branch equations at all the inductors:

$$Gu^{(2)}(t) + C\dot{u}^{(2)}(t) + M\dot{i}_l(t) = -i_s(t) \quad (14)$$

$$M^T u^{(2)}(t) - L\dot{i}_l(t) = 0 \quad (15)$$

where  $L$  is an  $l \times l$  non-singular diagonal matrix consisting of the inductance values of the  $l$  inductors in the circuit (mutual inductance is ignored), and  $M$  is an  $n \times l$  incidence matrix consisting of  $\pm 1$  or 0 elements only, according to:

**Table 1: DC Grid Verification, with  $\delta = 5\text{mV}$ .**

Grid Size (# nodes)	CPU Time
8K	1.1 hr
25K	4.8 hrs
41K	8.3 hrs
50K	10.8 hrs
73K	17.4 hrs
113K	1.14 days

- If node  $j$  is connected to inductor  $k$  and the reference direction for current in  $k$  is away from  $j$ , then  $m_{jk} = 1$ .
- If node  $j$  is connected to inductor  $k$  and the reference direction for current in  $k$  is towards  $j$ , then  $m_{jk} = -1$ .
- Otherwise,  $m_{jk} = 0$ .

Therefore, the voltage drop is the solution of the system:

$$Gv(t) + C\dot{v}(t) - M\dot{i}_l(t) = i_s(t) \quad (16)$$

$$M^T v(t) + L\dot{i}_l(t) = 0 \quad (17)$$

#### 4. VERIFYING DC GRIDS

Given (2), (5), (12), and (13), it is clear that  $v^*$  in the DC case may be found using a *sequence* of linear programs (LP), as first proposed in [5]:

$$\begin{aligned} &\text{for}(k = 1, 2, \dots, n) : \\ &\quad \text{Maximize: } v_k \\ &\quad \text{Subject to: } 0 \leq UGv \leq I_c \end{aligned} \quad (18)$$

Each LP is quite large in size, with  $n$  variables and as many constraints as there are current constraints, both local ( $m$ ) and global, but the fact that  $G$  is highly sparse helps address some of this concern. In addition, it turns out that it is computationally more effective to run the optimization in the current space, rather than voltage space, as follows:

$$\begin{aligned} &\text{for}(k = 1, 2, \dots, n) : \\ &\quad \text{Maximize: } v_k = G^{-1}i_s|_k \\ &\quad \text{Subject to: } 0 \leq U i_s \leq I_c \end{aligned} \quad (19)$$

where the notation  $G^{-1}i_s|_k$  denotes the  $k$ th entry of the vector  $G^{-1}i_s$ . This may be counter-intuitive because computation of a large matrix inverse is usually prohibitive. However, given the convenient properties of  $G$ , it turns out that using a sparse approximate inverse (SPAI) approach (such as will be explained below in reference to *RC* grids) is quite effective, especially that only one row of  $G^{-1}$  is required at a time. The key advantage is that the simplified structure of the constraints  $0 \leq U i_s \leq I_c$  is such that the LPs remain small in size and execute much faster. In spite of this, one obvious concern with this formulation is the fact that  $n$  such LPs have to be solved, one for each power grid node that needs to be verified.

Representative performance data is given in Table 1, showing that it is possible to verify a grid with 100k nodes in a day. This is obviously not good enough to verify large billion-node grids, but there is a number of considerations that address this concern. Firstly, and especially early in the design flow, one is often dealing with a coarse description of the power grid, in which the number of nodes is not

necessarily in the hundreds of millions and, anyway, it may be that one may not need to verify every node, but only certain critical nodes. Secondly, because each LP is decoupled from all the rest, one easy way to speed things up is to take advantage of parallelism, which is especially relevant today, with the proliferation of multi-core CPUs. Basically, every run of the for-loop above is a separate computational process and they may all be run in parallel. Thirdly, additional research results on macromodeling [6, 7], node elimination [8], and node dominance [9] go a long way to reduce the computational workload.

#### 5. TIME-DISCRETIZATION - RC GRIDS

One approach to solve the problem in the dynamic case (11) starts out by discretizing time and using a finite-difference approximation of the derivative, essentially a backward Euler numerical scheme  $\dot{v}(t) \approx [v(t) - v(t - \Delta t)] / \Delta t$ . Assuming that  $\Delta t$  is small enough to achieve good accuracy, we have:

$$Av(t) = Bv(t - \Delta t) + i_s(t) \quad (20)$$

where  $B = C/\Delta t$  and  $A = G + B$ . This recurrence relation captures the evolution of the system over time, and the voltage drop at any time  $t$  is given by:

$$v(t) = A^{-1}Bv(t - \Delta t) + A^{-1}i_s(t) \quad (21)$$

with the key observation that, like  $G$ , the matrix  $A$  is an  $M$ -matrix, so that  $A^{-1} \geq 0$  exists and  $A^{-1}B \geq 0$ .

##### 5.1 Exact Solution

Applying the recursion (21) at  $(t - \Delta t)$ :

$$v(t - \Delta t) = A^{-1}Bv(t - 2\Delta t) + A^{-1}i_s(t - \Delta t) \quad (22)$$

and substituting back for  $v(t - \Delta t)$  in (21) gives:

$$v(t) = (A^{-1}B)^2 v(t - 2\Delta t) + (A^{-1}B) A^{-1}i_s(t - \Delta t) + A^{-1}i_s(t) \quad (23)$$

and in general, with  $p \geq 1$ , we can write:

$$v(t) = (A^{-1}B)^p v(t - p\Delta t) + \sum_{q=0}^{p-1} (A^{-1}B)^q A^{-1}i_s(t - q\Delta t) \quad (24)$$

Assuming that the values of the source currents  $i_s(\cdot)$  at any two time points are independent variables, we can write:

$$\begin{aligned} v^* = \max_{i_s \in \mathcal{F}} [v(t)] &= \max_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^p v(t - p\Delta t) \right] \\ &+ \sum_{q=0}^{p-1} \max_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^q A^{-1}i_s(t - q\Delta t) \right] \end{aligned} \quad (25)$$

because  $v(t - p\Delta t)$  depends only on values of  $i_s(\tau)$  for  $\tau \leq (t - p\Delta t)$ . The assumption of independence of  $i_s(\cdot)$  across time is clearly conservative because it allows for maximum freedom of the optimization variables. Recall that the *spectral radius* of a matrix  $X$ , denoted  $\rho(X)$ , is the magnitude of the largest eigenvalue of  $X$ . From [10], we know that  $\rho(A^{-1}B) < 1$ . We also know from the *spectral mapping theorem* [11] that, if  $X$  is a square matrix and  $k$  is an integer, then the set of eigenvalues of  $X^k$  consists of all the eigenvalues of  $X$ , each raised to the power  $k$ , so that  $\rho((A^{-1}B)^p) < 1$  as well, which guarantees that  $\lim_{p \rightarrow \infty} (A^{-1}B)^p = 0$ . If we then let  $p \rightarrow \infty$  in (25), and because  $v(t - p\Delta t)$  is bounded

(the grid being a stable system with bounded inputs), then:

$$v^* = \sum_{q=0}^{\infty} \operatorname{emax}_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^q A^{-1}i_s \right] \quad (26)$$

where we use the single variable vector  $i_s$  to denote the current vector at any chosen time-point, again due to the independence of the currents across time. This (26) is the exact solution in the *RC* case. It is an infinite sum whose every term requires the solution of a linear program. Clearly, it is prohibitively expensive to use this for computation. Instead, in the following, we give practical ways to estimate  $v^*$ .

## 5.2 Upper-Bound

Given that the values of the source currents  $i_s(\cdot)$  at any two time points are independent variables, as assumed above, we can use (21) to write:

$$v^* = \operatorname{emax}_{i_s \in \mathcal{F}} [v(t)] = \operatorname{emax}_{i_s \in \mathcal{F}} [(A^{-1}B) v(t - \Delta t)] + \operatorname{emax}_{i_s \in \mathcal{F}} (A^{-1}i_s)$$

which, because  $A^{-1}B \geq 0$ , leads to:

$$v^* \leq (A^{-1}B) \operatorname{emax}_{i_s \in \mathcal{F}} [v(t - \Delta t)] + \operatorname{emax}_{i_s \in \mathcal{F}} (A^{-1}i_s) \quad (27)$$

However, because  $v^*$  is time-independent, then we can write  $\operatorname{emax}_{i_s \in \mathcal{F}} [v(t - \Delta t)] = \operatorname{emax}_{i_s \in \mathcal{F}} [v(t)] = v^*$ , so that:

$$(I - A^{-1}B) v^* \leq \operatorname{emax}_{i_s \in \mathcal{F}} (A^{-1}i_s) \quad (28)$$

Because  $B = A - G$ , then  $I - A^{-1}B = I - A^{-1}(A - G) = A^{-1}G$ , and  $I + G^{-1}B = I + G^{-1}(A - G) = G^{-1}A$ , so that:

$$(I - A^{-1}B)^{-1} = I + G^{-1}B \geq 0 \quad (29)$$

which is non-negative because both  $G^{-1} \geq 0$  and  $B \geq 0$ . So, we can multiply the left-hand-side of (28) by  $(I - A^{-1}B)^{-1}$  and the right-hand-side by  $I + G^{-1}B$ , to get:

$$v^* \leq (I + G^{-1}B) \operatorname{emax}_{i_s \in \mathcal{F}} (A^{-1}i_s) \quad (30)$$

This is a very useful upper-bound, which was first derived in [10] based on a power-series argument arising from (26). Its accuracy was investigated [10] and found to be quite good (recent tests show a maximum error of 4mV on a 5K node grid). Computationally, it requires a sequence of  $n$  LPs to evaluate the single  $\operatorname{emax}(\cdot)$  term, exactly like a DC verification (19), followed by the solution of a linear algebraic system based on *LU*-factorization of  $G$ . It reduces the cost of verification under transient currents to about the same as that of verification under DC currents. It provides a *sufficient condition* for grid safety: if the right-hand-side of (30) is less than a user-specified threshold, then the grid is safe.

It remains to deal with the fact that we need to have the explicit inverse  $A^{-1}$  to evaluate (30). Normally, this is prohibitive. However, because the system matrix  $A$  is sparse, symmetric, positive-definite, and a banded  $\mathcal{M}$ -matrix (for a reference on  $\mathcal{M}$ -matrices, consult [12]), then  $A^{-1}$  is non-negative,  $A^{-1} \geq 0$ , and its entries have values that decay exponentially as one moves away from the diagonal [13]. With this, one can use the Sparse Approximate Inverse (SPAI) technique [14] to construct a very good approximation to  $A^{-1}$  that neglects the vast majority of negligible off-diagonal entries, on a row (or column) at a time basis, and subject to a user-specified error tolerance. SPAI was adapted to

**Table 2: RC Grid Verification, with  $\delta = 5\text{mV}$ .**

Grid Size (# nodes)	CPU Time
8K	2.36 mins
21K	13 mins
40K	28 mins
50K	39 mins
73K	1.23 hrs
113K	2.67 hrs
364K	24.6 hrs
1M	9.27 days

power grid verification in [15], to ensure that the resulting matrix inverse guarantees an upper bound on  $v^*$ . With an error tolerance of  $\delta = 5\text{mV}$ , some representative performance data is given in Table 2. Basically, a million-node grid takes just under 10 days. Note that this data is much better than in the DC case. The reason is that the matrix  $A = G + C/\Delta t$  is even more diagonally dominant than  $G$  and so its inverse is more sparse and, overall, the individual LPs turn out to be smaller and less expensive. Finally, SPAI is just as parallelizable as the computation of the  $\operatorname{emax}(\cdot)$  operator, and the only task that is not completely parallelizable is the *LU*-factorization of  $G$ , required for (30), which must be performed only once.

## 5.3 Extended Upper-Bound

The above upper bound can also be extended and generalized for better accuracy if need be, so that multiple (but obviously not infinite), applications of  $\operatorname{emax}(\cdot)$  can be done. From (25), with  $p \geq 1$ , and because  $(A^{-1}B)^p \geq 0$ , then:

$$\begin{aligned} \operatorname{emax}_{i_s \in \mathcal{F}}(v(t)) &\leq (A^{-1}B)^p \operatorname{emax}_{i_s \in \mathcal{F}}(v(t - p\Delta t)) \\ &+ \sum_{q=0}^{p-1} \operatorname{emax}_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^q A^{-1}i_s(t - q\Delta t) \right] \end{aligned} \quad (31)$$

By time-invariance of  $v^*$ , this leads to:

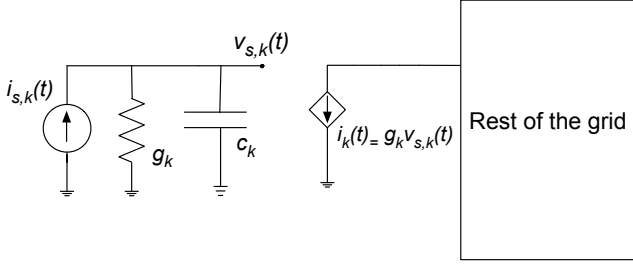
$$\left[ I - (A^{-1}B)^p \right] v^* \leq \sum_{q=0}^{p-1} \operatorname{emax}_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^q A^{-1}i_s \right] \quad (32)$$

Recall that  $\rho((A^{-1}B)^p) < 1$ . Recall also from [12] that, if  $X$  is a non-negative matrix, then  $\rho(X) < 1$  if and only if  $(I - X)$  is non-singular and  $(I - X)^{-1}$  is non-negative. Clearly,  $(A^{-1}B)^p$  is non-negative because  $A^{-1}B \geq 0$ . Then  $[I - (A^{-1}B)^p]$  is non-singular and its inverse is non-negative, so that we can multiply both sides of the above inequality by that inverse, which yields the generalized upper-bound:

$$v^* \leq \left[ I - (A^{-1}B)^p \right]^{-1} \sum_{q=0}^{p-1} \operatorname{emax}_{i_s \in \mathcal{F}} \left[ (A^{-1}B)^q A^{-1}i_s \right] \quad (33)$$

## 5.4 Current Slope Constraints

In the preceding, we have assumed that the currents at any two time-points are independent variables. However, this means that the current source waveform slopes can be quite large, especially for small  $\Delta t$ . To avoid this unrealistic situation without sacrificing computational efficiency, one can use the expedient of *current filters*, which we propose for the case of *RC* grids as follows. First, rewrite the *RC* grid



**Figure 2: Controlled source filter.**

system (11), recalling that  $i_s(t) = Hi(t)$ , as:

$$Gv(t) + C\dot{v}(t) = Hi(t) \quad (34)$$

Next, we assume that every current source  $i_k(t)$  is in fact derived from another independent current source  $i_{s,k}(t)$  by means of the simple relationship:

$$i_k(t) = g_k v_{s,k}(t) \quad (35)$$

where  $v_{s,k}(t)$  is the voltage at the terminal of  $i_{s,k}(t)$  that is connected to ground through the parallel combination of a resistor with conductance  $g_k$  and a capacitor  $c_k$ , while the other terminal of  $i_{s,k}(t)$  is connected directly to ground, as in Fig. 2. Assuming the positive (reference) direction of current  $i_{s,k}(t)$  is as shown in Fig. 2, then:

$$g_k v_{s,k}(t) + c_k \dot{v}_{s,k}(t) = i_{s,k}(t) \quad (36)$$

With this, the source  $i_{s,k}(t)$  can take values that are independent variables at any two-time points, while the sources  $i_k(t)$  applied to the grid are slope-constrained by the single time-constant arising from the parallel combination of  $g_k$  and  $c_k$ . The matrix equation for all these filtered sources becomes:

$$G_s v_s(t) + C_s \dot{v}_s(t) = i_s(t) \quad (37)$$

where  $G_s$  and  $C_s$  are non-singular diagonal matrices with positive diagonals. Making use of  $i(t) = G_s v_s(t)$ , arising from (35), and combining (34) and (37) leads to:

$$\begin{bmatrix} G & -HG_s \\ 0 & G_s \end{bmatrix} \begin{bmatrix} v(t) \\ v_s(t) \end{bmatrix} + \begin{bmatrix} C & 0 \\ 0 & C_s \end{bmatrix} \begin{bmatrix} \dot{v}(t) \\ \dot{v}_s(t) \end{bmatrix} = \begin{bmatrix} 0 \\ i_s(t) \end{bmatrix} \quad (38)$$

or:

$$\tilde{G}\tilde{v}(t) + \tilde{C}\dot{\tilde{v}}(t) = \tilde{i}_s(t) \quad (39)$$

Note that the larger capacitance matrix  $\tilde{C}$  remains diagonal, non-negative, and non-singular with a positive diagonal. Note also that  $\tilde{G}$  is non-singular because its inverse is:

$$\tilde{G}^{-1} = \begin{bmatrix} G^{-1} & G^{-1}H \\ 0 & G_s^{-1} \end{bmatrix} \quad (40)$$

which, crucially for our work, is non-negative.

Using a backward Euler finite difference approximation,  $\dot{\tilde{v}} \approx (\tilde{v}(t) - \tilde{v}(t - \Delta t))/\Delta t$ , a discrete-time version of the system equation (39) can be written, as before, as:

$$\left( \tilde{G} + \frac{\tilde{C}}{\Delta t} \right) \tilde{v}(t) = \frac{\tilde{C}}{\Delta t} \tilde{v}(t - \Delta t) + \tilde{i}_s(t) \quad (41)$$

or, simply:

$$\tilde{A}\tilde{v}(t) = \tilde{B}\tilde{v}(t - \Delta t) + \tilde{i}_s(t) \quad (42)$$

where  $\tilde{B} = \tilde{C}/\Delta t$  is a non-singular diagonal matrix, with a positive diagonal, and  $\tilde{A} = (\tilde{G} + \tilde{B})$  can be shown to be non-singular with a non-negative inverse, as follows. Note:

$$\tilde{A} = \begin{bmatrix} G + \frac{C}{\Delta t} & -HG_s \\ 0 & G_s + \frac{C_s}{\Delta t} \end{bmatrix} \quad (43)$$

Considering the two block matrices on the diagonal, notice first that  $G + C/\Delta t$  is our original  $A$  matrix (from (20)) which is an  $\mathcal{M}$ -matrix, and so has a non-negative inverse. Notice also that  $G_s + C_s/\Delta t$  is a diagonal matrix with a positive diagonal, so that it is non-singular and its inverse is also diagonal with a positive diagonal. Therefore, the following matrix is well-defined:

$$\begin{bmatrix} (G + \frac{C}{\Delta t})^{-1} & (G + \frac{C}{\Delta t})^{-1} HG_s (G_s + \frac{C_s}{\Delta t})^{-1} \\ 0 & (G_s + \frac{C_s}{\Delta t})^{-1} \end{bmatrix} \quad (44)$$

and is non-negative, and it is trivial to verify that this matrix is in fact  $\tilde{A}^{-1}$ . Therefore,  $\tilde{A}^{-1}\tilde{B} \geq 0$ , which is of interest because of the relation:

$$\tilde{v}(t) = \tilde{A}^{-1}\tilde{B}\tilde{v}(t - \Delta t) + \tilde{A}^{-1}\tilde{i}_s(t) \quad (45)$$

Recall again from [12] that, for a non-negative matrix  $X$ ,  $\rho(X) < 1$  if and only if  $(I - X)$  is non-singular and its inverse is non-negative. For our work, let  $X = \tilde{A}^{-1}\tilde{B}$ , then  $X = \tilde{A}^{-1}(\tilde{A} - \tilde{G}) = I - \tilde{A}^{-1}\tilde{G}$ , so that  $I - X = \tilde{A}^{-1}\tilde{G}$ , whose inverse  $\tilde{G}^{-1}\tilde{A} = \tilde{G}^{-1}(\tilde{G} + \tilde{B}) = I + \tilde{G}^{-1}\tilde{B}$  exists and is non-negative. Therefore,  $\rho(\tilde{A}^{-1}\tilde{B}) < 1$ .

With  $\tilde{A}^{-1}\tilde{B} \geq 0$  and  $\rho(\tilde{A}^{-1}\tilde{B}) < 1$ , we now have the same key properties that enabled the derivation of the exact solution and the upper bounds in the previous sections. Thus, the same algorithms given above for  $RC$  grids can be applied here, thereby incorporating source waveform slope constraints. Empirical data shows that this has some, but not a great impact on the results, probably due to the fact that grid parasitics impose their own “filter effect”, so that slope constraints probably affect only a small fraction of the grid nodes, mainly those that are close to the current sources.

## 6. TIME-DISCRETIZATION - RLC GRIDS

As in the  $RC$  case, backward Euler discretization can be applied to the  $RLC$  dynamic equations (16), (17), leading to:

$$\left( G + \frac{C}{\Delta t} \right) v(t) - Mi_l(t) = \frac{C}{\Delta t} v(t - \Delta t) + i_s(t) \quad (46)$$

$$M^T v(t) + \frac{L}{\Delta t} i_l(t) = \frac{L}{\Delta t} i_l(t - \Delta t) \quad (47)$$

Multiplying (47) by  $L^{-1}\Delta t$  to get an expression for  $i_l(t)$  and substituting that in (46), similar to what was done for the case of trapezoidal integration in [16], gives:

$$Dv(t) = \frac{C}{\Delta t} v(t - \Delta t) + i_s(t) + Mi_l(t - \Delta t) \quad (48)$$

where:

$$D \triangleq G + \frac{C}{\Delta t} + M \left( \frac{L}{\Delta t} \right)^{-1} M^T \quad (49)$$

The next step is to find a way to remove the  $i_l(t - \Delta t)$  term from (48). The details are given in [17] and require a mild assumption about the circuit topology (every inductance must be in series with a resistor), from which it follows

that  $i_i(t-\Delta t) = \hat{G}v(t-\Delta t)$ , where  $\hat{G}$  is a simple variation on  $G$  that is very easily constructed based on circuit topology. This is important because it leads to the following single recurrence equation that governs the system dynamics:

$$Dv(t) = Ev(t - \Delta t) + i_s(t) \quad (50)$$

where  $E = M\hat{G} + C/\Delta t$ , and it can be proven that the system matrix  $D$  is sparse, symmetric positive-definite, and a banded  $\mathcal{M}$ -matrix. It follows that  $D^{-1} \geq 0$  exists and its entries decay exponentially as we move away from the diagonal [13].

## 6.1 Exact Solution

In the *RLC* case, because of the possibility of getting over-shoots, we are interested in both the maximum voltage drop  $\text{emax}[v(t)]$  and the minimum voltage drop  $\text{emin}[v(t)]$ , which gives the over-shoot. We combine these in the single  $2n \times 1$  vector  $v^{(opt)}$  using the  $\text{eopt}(\cdot)$  operator (7):

$$v^{(opt)} = \text{eopt}(v(t)) = \begin{bmatrix} \text{emax}_{i_s \in \mathcal{F}}(v(t)) \\ \text{emin}_{i_s \in \mathcal{F}}(v(t)) \end{bmatrix} \quad (51)$$

Starting with (50) and proceeding as in the *RC* case, skipping the details, one can find a similar expression for the exact solution, based on the argument [17] that  $\rho(D^{-1}E) \leq 1$  (due to the fact that backward Euler is absolutely stable and the grid is a stable system), so that:

$$v^{(opt)} = \sum_{q=0}^{\infty} \text{eopt} \left[ (D^{-1}E)^q D^{-1}i_s \right] \quad (52)$$

Here too, this is too expensive to compute and more practical estimates have been found, as follows.

## 6.2 Upper and Lower Bounds

Unlike in the *RC* case where  $A^{-1}B$  was non-negative, we do not in this case have a guarantee that  $D^{-1}E$  is non-negative. This necessitates a more complex approach [17], whose end result is as follows. We can produce an upper-bound on the maximum voltage drop, call this  $v_{ub}$ :

$$\text{emax}_{i_s \in \mathcal{F}}(v(t)) \leq v_{ub} \quad (53)$$

and a lower-bound on the minimum voltage drop,  $v_{lb}$ :

$$\text{emin}_{i_s \in \mathcal{F}}(v(t)) \geq v_{lb} \quad (54)$$

and so the desired bounding solution in lieu of  $v^{(opt)}$  is:

$$\begin{bmatrix} v_{ub} \\ v_{lb} \end{bmatrix} \quad (55)$$

Let  $J = (D^{-1}E)^r$  where  $r \geq 1$  is an integer, and let  $T = |J|$  consist of the element-wise absolute values of  $J$ . We can prove that there exists a value of  $r$  for which  $\rho(T) < 1$  and we have devised a simple test by which this condition can be checked for a given  $r$ . The first phase is to increment  $r$  from 1, and identify the first  $r$  for which  $\rho(T) < 1$ . This becomes the fixed value of  $r$  for the rest of the algorithm. We then show that the desired bounds are given by:

$$\begin{bmatrix} v_{ub} \\ v_{lb} \end{bmatrix} = \left\{ I - \frac{1}{2} \begin{bmatrix} J+T & J-T \\ J-T & J+T \end{bmatrix} \right\}^{-1} \times \sum_{q=0}^{r-1} \text{eopt} \left[ (D^{-1}E)^q D^{-1}i_s \right] \quad (56)$$

**Table 3: RLC Grid Verification, with  $\delta = 5\text{mV}$ .**

Grid Size (# nodes)	$r$	CPU Time
9K	3	41 mins
14K	1	1.26 hrs
20K	3	3 hrs
43K	4	4.17 hrs
76K	4	9.8 hrs
119K	7	22.32 hrs
170K	5	1.5 days

In our experience, the values of  $r$  are very small, in the range 1-7 or so. SPAI is again used to speed up the  $\text{eopt}(\cdot)$  operations and the overall runtime is demonstrated by some representative data in Table 3. The quality of the bounds is quite good, in the range of 1-9mV for grids where a comparison to the exact solution is possible [17].

## 7. CONTINUOUS TIME - WAVELETS

A drastically different approach has been provided in [18], where time is not discretized but, instead, current source waveforms are expressed as an expansion on a set of basis functions, and this is applicable to both *RC* and *RLC* grids. A finite set of wavelets  $\psi_{m,n}(t)$  is used as the basis, where  $m = 1, 2, \dots, M$ , where  $M = \lceil \log_2(2f_{max}/f_{min}) \rceil$  enumerates the spectrum of frequencies in the basis, up to a maximum that is determined by the signal frequency bandwidth of interest to the user  $[f_{min}, f_{max}]$ , and  $n = 1, 2, \dots$  up to a maximum that depends on the impulse response duration of the grid. With this, the current waveform of the  $j$ th current source is expressed as:

$$i_j(t) = i_{dc,j} + \sum_{m=1}^M \sum_{n=0}^{n_{m,j}} T_{m,n,j} \psi_{m,n}(t) \quad (57)$$

where  $i_{dc,j}$  and  $T_{m,n,j}$  are coefficients that are determined as a result of the power grid verification. Once the grid has been verified, we would not only know the worst-case under-shoot and over-shoot but also the combination of current stimulus waveforms at the current sources that is responsible for the worst-case. This provision of diagnostic information, about the current combination that causes the worst case, is a special and key feature of this approach.

Let  $h_{m,n,j,z}(t)$  be the voltage drop waveform at node  $z$ , due to a single wavelet stimulus  $\psi_{m,n}(t)$  applied at source  $i_j(t)$ , with all other current sources turned off (open-circuit). Let  $k_{j,z}$  be the dc gain from current source  $j$  to node  $z$ . Then, by superposition, the voltage response at node  $z$  when only current source  $i_j(t)$  is active is given by:

$$v_{j,z}(t) = k_{j,z} i_{dc,j} + \sum_{m=1}^M \sum_{n=0}^{n_{m,j}} h_{m,0,j,z} ((n_{m,j} - n)2^m u) T_{m,n,j} \quad (58)$$

where  $u$  is the time unit. If the grid is driven by  $q$  current sources then the complete voltage response at node  $z$  is:

$$\begin{aligned} v_z(t) &= \sum_{j=1}^q k_{j,z} i_{dc,j} \\ &+ \sum_{j=1}^q \sum_{m=1}^M \sum_{n=0}^{n_{m,j}} h_{m,0,j,z} ((n_{m,j} - n)2^m u) T_{m,n,j} \end{aligned} \quad (59)$$

which is a linear combination of known time functions with a number of unknown coefficients. Current constraints (including frequency domain constraints in this case) applied to the currents (57) translate to a set of linear constraints on the coefficients. The node voltages (59) are then maximized or minimized subject to these constraints, resulting in a linear program implementation for power grid verification.

In spite of the ominous mention of impulse response and dc gain, the method is quite efficient to set up because it is based on the Haar wavelet family. The main source of computational difficulty is actually the size of the wavelets basis and so the number of coefficients. Overall, the method can be used to verify a 36K node grid in about 11 hours. Given the diagnostic ability of the approach, it is certainly applicable to a high-level power grid model in which, for example, the resonance with the package is to be studied.

## 8. ADDITIONAL FEATURES

There have been many variations on the above techniques. Verification where both the power and ground grids were considered jointly is given in [3]. Macromodeling of parts of the grid that do not need to be verified, in the interest of an incremental or hierarchical grid verification have been studied in [6], [8], and more recently in [7]. The very powerful technique of node dominance has been given in [9], whereby many nodes are identified that do not need to be verified because their safety is implied by the safety of other nodes. Dual LP based techniques have been proposed in [19], whereby the constraints remain linear but the objective function for verification becomes nonlinear but convex, allowing one to benefit from powerful modern convex optimization methods. An interesting technique is given in [20] for fast matrix inversion whereby the values of some columns of the matrix are inferred from other columns. An extension of the verification problem to the case of transient constraints has been given in [21]. Finally, in [22] and recently in [23] (corrected in [24]), a restriction of the problem to *hierarchical constraints* is considered, whereby global constraints are required to be wholly contained in other global constraints.

## 9. REFERENCES

- [1] R. Chaudhry, D. Blaauw, R. Panda, and T. Edwards, "Current signature compression for IR-drop analysis," in *Design Automation Conference*, Los Angeles, CA, June 5-9 2000, pp. 162–167.
- [2] S. Pant and E. Chiprout, "Power grid physics and implications for CAD," in *ACM/IEEE 43rd Design Automation Conference (DAC-06)*, San Francisco, CA, July 24-28 2006, pp. 199–204.
- [3] M. Avci and F. N. Najm, "Early P/G grid voltage integrity verification," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 7-11 2010, pp. 816–823.
- [4] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: John Wiley & Sons, Inc., 2010.
- [5] D. Kouroussis and F. N. Najm, "A static pattern-independent technique for power grid voltage integrity verification," in *ACM/IEEE 40th Design Automation Conference (DAC-03)*, Anaheim, CA, June 2-6 2003, pp. 99–104.
- [6] D. Kouroussis, I. A. Ferzli, and F. N. Najm, "Incremental partitioning-based vectorless power grid verification," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD-05)*, San Jose, CA, November 6-10 2005, pp. 358–364.
- [7] Abhishek and F. N. Najm, "Incremental power grid verification," in *ACM/IEEE 49th Design Automation Conference (DAC-2012)*, San Francisco, CA, June 3-7 2012, pp. 151–156.
- [8] A. Goyal and F. N. Najm, "Efficient RC power grid verification using node elimination," *Design, Automation and Test in Europe (DATE-11)*, pp. 257–260, March 14-18 2011.
- [9] N. Abdul Ghani and F. N. Najm, "Power grid verification using node and branch dominance," in *ACM/IEEE 48th Design Automation Conference (DAC-2011)*, San Diego, CA, June 5-9 2011, pp. 682–687.
- [10] I. A. Ferzli, F. N. Najm, and L. Kruse, "A geometric approach for early power grid verification using current constraints," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 5-8 2007, pp. 40–47.
- [11] S. Lang, *Algebra*. Menlo Park, CA: Addison-Wesley, 1993.
- [12] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA: SIAM, 2003.
- [13] S. Demko, W. F. Moss, and P. W. Smith, "Decay rates for inverses of band matrices," *Mathematics of computation*, vol. 43, no. 168, pp. 491–499, Oct. 1984.
- [14] M. J. Grote and T. Huckle, "Parallel preconditioning with sparse approximate inverses," *SIAM Journal on Scientific Computing*, vol. 18, no. 3, pp. 838–853, May 1997.
- [15] N. H. Abdul Ghani and F. N. Najm, "Fast vectorless power grid verification using an approximate inverse technique," in *ACM/IEEE 46th Design Automation Conference (DAC-09)*, San Francisco, CA, July 26-31 2009, pp. 184–189.
- [16] T.-H. Chen, C. Luk, and C. C.-P. Chen, "INDUCTWISE: inductance-wise interconnect simulator and extractor," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 7, pp. 884–894, Jul. 2003.
- [17] N. H. Abdul Ghani and F. N. Najm, "Fast vectorless power grid verification under an RLC model," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 5, pp. 691–703, May 2011.
- [18] I. A. Ferzli, E. Chiprout, and F. N. Najm, "Verification and codesign of the package and die power delivery system using wavelets," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 1, pp. 92–102, Jan. 2010.
- [19] X. Xiong and J. Wang, "Dual algorithms for vectorless power grid verification under linear current constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 10, pp. 1469–1482, Oct. 2011.
- [20] —, "A hierarchical matrix inversion algorithm for vectorless power grid verification," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 7-11 2010, pp. 543–550.
- [21] —, "Vectorless verification of RLC power grids with transient current constraints," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, November 7-10 2011, pp. 548–554.
- [22] M. Avci, "Early dual grid voltage integrity verification," Master's thesis, University of Toronto, Toronto, Ontario, Canada, 2010.
- [23] Y. Wang, X. Hu, C.-K. Cheng, G.-K.-H. Pang, and N. Wong, "A realistic early-stage power grid verification algorithm based on hierarchical constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 1, pp. 109–120, Jan. 2012.
- [24] —, "Corrigendum to 'A realistic early-stage power grid verification algorithm based on hierarchical constraints'," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 3, pp. 452–452, Mar. 2012.