Fast Simulation-Based Verification of RC Power Grids

Mohammad Fawaz
Department of ECE
University of Toronto, ON, Canada
mohammad.fawaz@mail.utoronto.ca

Farid N. Najm
Department of ECE
University of Toronto, ON, Canada
f.najm@utoronto.ca

***Abstract*—To guarantee its safety, the power delivery network (PDN) must undergo a sequence of verification steps throughout the integrated circuit (IC) design flow. This involves checking that the voltage fluctuations in the grid remain within some user-specified safety threshold. Typically, this is done by performing a transient simulation of the grid under certain input current traces. Existing simulation tools require solving a large number of linear systems, making the tools slow for modern power grids containing billions of nodes. We propose a new simulation based approach for RC power grid verification that generates envelope upper bound waveforms on the true voltage drop waveforms. The envelope waveforms capture the peaks of the voltage drops quite accurately while requiring a much smaller number of system solves than traditional tools.**

I. Introduction

Power grid verification has become one of the most critical steps in any modern VLSI design flow. A well designed power grid must be able to provide the required voltage levels to every block/gate in the underlying logic circuitry. Unwanted voltage fluctuations often lead to a decline in the performance of the chip both in terms of functionality and speed. Thus, there is a clear need for efficient power grid verification.

There are two main causes for the variation in the voltage levels: IR drops resulting from the resistive nature of the grid's metallic structure, and Ldi/dt noise which is due to the inductance of the rails as well as the inductance introduced by the grid-package interconnections. Often, the inductance in the grid is ignored, leading to a simpler pure RC mesh which can only exhibit *undershoots* in the node voltages. In order to check the safety of the grid, one must verify that the voltage drop at every node never exceeds a certain user-defined threshold.

In order to do that, modern grid verification tools perform a transient simulation of the grid under some user-defined current traces that model the currents drawn by the underlying logic circuitry. This is typically done by solving the differential equation resulting from Nodal Analysis [1], which models the behavior of the grid. In most cases, solving the differential equation is done by first discretizing time and then solving a linear system of equations at every time-step. State-of-the-art power grid simulation tools suffer from serious performance issues due to the size of modern grids containing billions of nodes. Moreover, for the simulation to be accurate, the number of linear systems that need to be solved is very large because the time-step that is required for an accurate time discretization, has to be very small.

Many algorithms have been proposed for transient simulation by exploiting different tradeoffs. Traditional approaches use the standard LU factorization, Cholesky factorization [2], or the preconditioned conjugate gradient (PCG) method [3]. Other tools use random walks [4] and multigrid techniques [5]. Hierarchical approaches are proposed in [6]. Moreover, and as part of TAU 2012 power grid simulation contest, [7] and [8] have been proposed. Parallelization of forward/backward substitution is proposed in [9]. A solution utilizing the matrix exponential kernel is proposed in [10].

All of the algorithms mentioned above exploit the properties of the power grid model in order to solve every linear system as fast as possible. To preserve the accuracy of the simulation, the total number of linear systems that need to be solved is kept the same. That number is usually determined based on some fixed value of the time-step or based on some dynamic value that keeps changing based on the input currents. However, for verification purposes, the *true* voltage drop waveforms are not necessary. In fact, any set of *upper bound* waveforms capturing the peaks of the voltage drops can be just as useful. In this paper, we propose two new algorithms for power grid verification that generate *envelope* waveforms for the true voltage drop waveforms. These envelope waveforms can be generated much faster than the true waveforms because they only consider the breakpoints in the input current traces. In other words, the total number of linear systems that need to be solved is reduced dramatically. We show that our approaches can have up to 73X speed-up over modern transient simulation algorithms while having a decent accuracy.

The remainder of the paper is organized as follows. In section II, we review some background material on the power grid model and the current waveforms. Section III presents the derivation of the exact solution. In sections IV and V, we explain how the envelope waveforms are found. Section VI explains some implementation details while section VII provides the test results. Finally, we conclude the paper in section VIII

II. Background

A. Power Grid Model

The power grid is a large full-chip structure of connected metal lines, across multiple layers interconnected through vias and connected by C4 bumps to wiring in the package and on the board. Typically, a power grid is modelled as a linear circuit composed of a large number of lumped linear (RLC) elements. In many cases, the inductance in the grid can be ignored leading to a simpler RC model.

Consider an RC grid in which there are three types of nodes: 1) some nodes are connected to ideal current sources to ground, in parallel with a capacitor to ground; 2) some (most) nodes are connected only to resistors to other grid nodes and capacitors to ground; 3) some nodes are connected to resistors to other grid nodes and ideal voltage sources to ground. The current sources (with their parallel C) represent the current drawn by the logic circuits tied to the grid at these nodes. The ideal voltage sources represent the external voltage supply, V_{dd}. Excluding the ground node, let the power grid consist of $n + p$ nodes where nodes $1, 2, \ldots, n$ are the nodes not connected to a voltage source, while the remaining nodes

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

978-1-4673-8721-7/16/$31.00 ©2016 IEEE

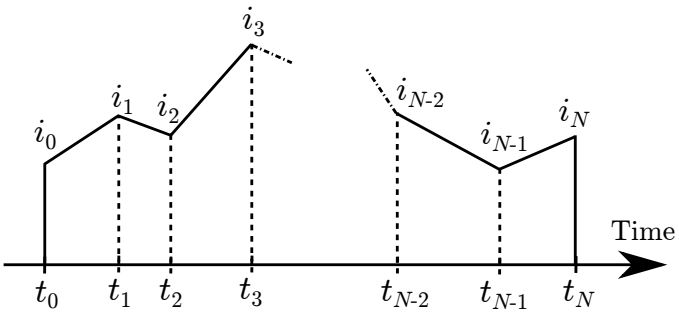

Fig. 1. Piece-wise linear current waveform

$(n+1),(n+2),\ldots,(n+p)$ 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 $1,2,\ldots,m \leq n$. Let H be an $n \times m$ matrix of 0 and 1 entries that identifies which node is connected to which current source.

B. Current Waveforms

Generally, specifying the patterns of the currents drawn from the grid is a tedious task. In many cases, these patterns are generated based on the estimated average chip current and average/peak current factor. In other cases, the waveforms are generated by simulating the logic blocks for a certain period of time. The result is usually a set of *piece-wise linear* waveforms generated by probing the output waveforms at particular time points. In this work, we assume that the current patterns are user-defined in the form of a set of pairs as follows:

$$\mathcal{I} = \{(t_0, i_0), (t_1, i_1), \ldots, (t_N, i_N)\} \tag{1}$$

where $0 = t_0 < t_1 < \ldots < t_N$ are the time points at which the current vector $i(t)$ is available (breakpoints), and $i_0, i_1, \ldots, i_N$ are $m \times 1$ vectors containing the current values at $t_0, t_1, \ldots, t_N$ respectively. Figure 1 shows an example of a piece-wise linear current waveform with $N+1$ breakpoints. We also assume that $i(t) = 0$ for all $t < 0$ and $t > t_N$. Moreover, let T be the set of all time points:

$$T = \{t_0, t_1, \ldots, t_N\} \tag{2}$$

and define the intervals $T_k = [t_k, t_{k+1}]$ for $k \in \{0, \ldots, N-1\}$.

III. Nodal Analysis

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 be $u^{(1)}(t) = V_{dd}$, 2) short-circuit all the voltage sources and find the response $u^{(2)}(t)$, in this case clearly $u^{(2)}(t) \leq 0$, and 3) find $u(t) = u^{(1)}(t) + u^{(2)}(t)$. To find $u^{(2)}(t)$, KCL at every node provides, via Nodal Analysis [1], that:

$$Gu^{(2)}(t) + C\dot{u}^{(2)}(t) = -Hi(t) \tag{3}$$

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 or one current source, then G is known to be irreducibly diagonally dominant [1]. With this, it can be shown that G is an $\mathcal{M}$-matrix [11], 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 $v(t) = V_{dd} - u(t) = -u^{(2)}(t) \geq 0$, so that:

$$Gv(t) + C\dot{v}(t) = Hi(t) \tag{4}$$

Note therefore that $v(t)$ can be found directly as the node voltages resulting from an analysis of the circuit in the case where the voltage sources are short-circuited and the current source directions are reversed.

A. Time-Discretization

One approach to solve the dynamic system (4) starts out by discretizing time and using a finite-difference approximation of the derivative, essentially a backward Euler (BE) numerical scheme $\dot{v} \approx \frac{v(t)-v(t-\Delta t)}{\Delta t}$. If that Δt is small enough, then we have:

$$Av(t) \approx Bv(t-\Delta t) + Hi(t) \tag{5}$$

where $B = \frac{C}{\Delta t}$ and $A = G + B$. Assuming then that (5) is accurate, it leads to a recurrence relation that captures the evolution of the system over time, so that the voltage drop at any time t is given by:

$$v(t) = A^{-1}Bv(t-\Delta t) + A^{-1}Hi(t) \tag{6}$$

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

B. Exact Solution

Apply the recursion (6) at $(t-\Delta t)$:

$$v(t-\Delta t) = A^{-1}Bv(t-2\Delta t) + A^{-1}Hi(t-\Delta t) \tag{7}$$

and substituting back for $v(t-\Delta t)$ in (6) gives:

$$\begin{aligned} v(t) = &\left(A^{-1}B\right)^2 v(t-2\Delta t) \\ &+ \left(A^{-1}B\right) A^{-1}Hi(t-\Delta t) + A^{-1}Hi(t) \end{aligned}$$

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

$$v(t) = \left(A^{-1}B\right)^p v(t-p\Delta t) + \sum_{q=0}^{p-1} \left(A^{-1}B\right)^q A^{-1}Hi(t-q\Delta t) \tag{8}$$

Recall that the *spectral radius* of a matrix X, denoted $\rho(X)$, is the magnitude of the largest eigenvalue of X. From [12], we know that:

$$\rho\left(A^{-1}B\right) < 1 \tag{9}$$

This implies that [13]:

$$\lim_{p\to\infty} \left(A^{-1}B\right)^p = 0$$

Therefore, if we let $p \to \infty$ in (8), and because $v(t-p\Delta t)$ is bounded (the grid being a stable system with bounded inputs), then:

$$v(t) = \sum_{q=0}^{\infty} \left(A^{-1}B\right)^q A^{-1}Hi(t-q\Delta t) \tag{10}$$

IV. Transient Envelope Waveforms

In this section, we introduce an approach that generates upper bound waveforms on the exact voltage drop waveforms $v(t)$ in (10). We first define the following extreme-value operator:

Definition 1. (emax). Let $f(x) : \mathbb{R} \to \mathbb{R}^n$ be a vector function whose components will be denoted $f_1(x), \ldots, f_n(x)$, and let $\mathcal{A} \subset \mathbb{R}$. We define the operator $\underset{x\in\mathcal{A}}{\text{emax}}[f(x)]$ as one that provides the $n \times 1$ vector $y = \underset{x\in\mathcal{A}}{\text{emax}}[f(x)]$ such that, for every $j \in \{1, 2, \ldots, n\}$:

$$y_j = \max_{x\in\mathcal{A}}[f_j(x)]$$

Thus, $\text{emax}[\cdot]$ performs element-wise maximization over the domain of the one dimensional variable x. The $\text{emax}[\cdot]$ operator can also be used to perform element-wise maximization of a countable set of real valued vectors, i.e. if $\{z_1, z_2, \ldots\}$ is a (finite or countably infinite) set of vectors in $\mathbb{R}^n$, one can write $\text{emax}[z_1, z_2, \ldots]$ to denote the vector containing the element-wise maximum of these vectors.

A. General Case

Let $h' \in \mathbb{R}$ be the time duration that represents the largest significant duration of the impulse response function of any node on the grid, and let h be the smallest multiple of Δt that is larger than h'. The choice of h' (and h) depends on the dynamics of the power grid as a linear system, and will be further discussed in Section VI-A. If h and h' are chosen properly, then one can approximate the exact solution in (10) as follows:

$$v(t) \approx \sum_{q=0}^{h/\Delta t} \left(A^{-1}B\right)^q A^{-1} H i(t - q\Delta t) \triangleq v_h(t) \tag{11}$$

For any $t \in \mathbb{R}$, let $w(t) = A^{-1}Hi(t)$ and:

$$\overline{w}(t) = \underset{\tau \in [t-h,t]}{\operatorname{emax}} [w(\tau)] \tag{12}$$

Clearly, for $q \in \{0, 1, \ldots, h/\Delta t\}$, we have:

$$A^{-1}Hi(t - q\Delta t) \leq \overline{w}(t) \tag{13}$$

Therefore, because $A^{-1}B \geq 0$, we have:

$$\begin{aligned} v_h(t) &\leq \sum_{q=0}^{h/\Delta t} \left(A^{-1}B\right)^q \overline{w}(t) \leq \sum_{q=0}^{\infty} \left(A^{-1}B\right)^q \overline{w}(t) \\ &= \left(\sum_{q=0}^{\infty} \left(A^{-1}B\right)^q\right) \overline{w}(t) \end{aligned} \tag{14}$$

This is valid because the infinite summation $\sum_{q=0}^{\infty} \left(A^{-1}B\right)^q$ is convergent since $\rho\left(A^{-1}B\right) < 1$ [13]. In fact, because $\rho\left(A^{-1}B\right) < 1$, the matrix $I_n - A^{-1}B$ is invertible (where I_n is the $n \times n$ identity matrix), and we have [13]:

$$\sum_{q=0}^{\infty} \left(A^{-1}B\right)^q = \left(I_n - A^{-1}B\right)^{-1} \tag{15}$$

Observe that:

$$\begin{aligned} \left(I_n - A^{-1}B\right)^{-1} &= \left(I_n - A^{-1}(A - G)\right)^{-1} \\ &= \left(A^{-1}G\right)^{-1} = G^{-1}A \end{aligned}$$

so that (14) becomes:

$$v_h(t) \leq G^{-1}A\overline{w}(t) \triangleq \overline{v}(t) \tag{16}$$

The waveform $\overline{v}(t)$ is effectively an *envelope waveform* representing an upper bound on the voltage drop at any time t. In other words, if we solve for the waveform $w(t)$ based on:

$$Aw(t) = Hi(t) \tag{17}$$

then compute the waveform $\overline{w}(t)$ using the $\operatorname{emax}[\cdot]$ operator, we can then solve for the envelope waveform $\overline{v}(t)$ using:

$$G\overline{v}(t) = A\overline{w}(t) \tag{18}$$

B. Case of Piece-Wise Linear Currents

In general, solving (17) is expensive because it requires solving a linear system of equations for every time point t. However, we are assuming that the available current waveforms are piece-wise linear, which means that $w(t)$ must also be piece-wise linear because $w(t)$ is the result of applying a *linear* transformation $A^{-1}H$ to $i(t)$. Accordingly, a simple way of finding $w(t)$ is by finding $w_0 = w(t_0), w_1 = w(t_1), \ldots, w_N = w(t_N)$ using the following linear systems:

$$Aw_k = Hi_k \quad \forall k \in \{0, 1, \ldots, N\} \tag{19}$$

The result would be a piece-wise linear function with the following breakpoints:

$$\mathcal{W} = \{(t_0, w_0), (t_1, w_1), \ldots, (t_N, w_N)\} \tag{20}$$

Notice that for $t < 0$ and $t > t_N$, we have $w(t) = 0$.

Once $\mathcal{W}$ is found, $\overline{w}(t)$ can be computed using the $\operatorname{emax}[\cdot]$ operator applied at *every* time point $t \in \mathbb{R}$ as in (12). Thus, finding $\overline{w}(t)$ *exactly* is not very practical. One way to overcome this problem is by computing $\overline{w}(t)$ at instants $\{t_0, t_0+\delta t, t_0+2\delta t, \ldots\}$ where δt is small enough so that the result is a good approximation of the actual $\overline{w}(t)$. But this may be expensive and will generate a piece-wise linear waveform that has a larger number of breakpoints than the original current waveforms, which will make the last step of our algorithm significantly more expensive as we will see shortly. Accordingly, we will introduce a new piece-wise linear function $\overline{w}'(t)$, which is much easier to compute than $\overline{w}(t)$, and which will guarantee a true bound on $v_h(t)$ when used instead of $\overline{w}(t)$.

For any time point $t_k \in T$, let $t_{\underline{k}}$ be the largest time point in T smaller than or equal to $t_k - h$ if $t_k - h \geq t_0$, and t_0 otherwise. In other words:

$$t_{\underline{k}} = \begin{cases} t_0 & \text{if } t_k - h < t_0 \\ \max\limits_{\substack{t \in T \\ t \leq t_k - h}} (\ t\ ) & \text{if } t_k - h \geq t_0 \end{cases} \tag{21}$$

Now, let:

$$\overline{w}'_k = \begin{cases} w(t_0) & \text{if } k = 0 \\ \operatorname{emax}\limits_{\tau \in \{t_{\underline{k-1}}, \ldots, t_k\}} [w(\tau)] & \text{if } k \in \{1, \ldots, N\} \end{cases} \tag{22}$$

and let $\overline{w}'(t)$ be piece-wise linear with the following breakpoints:

$$\overline{\mathcal{W}}' = \left\{\left(t_0, \overline{w}'_0\right), \left(t_1, \overline{w}'_1\right), \ldots, \left(t_N, \overline{w}'_N\right)\right\} \tag{23}$$

and such that:

$$\overline{w}'(t) = \begin{cases} 0 & \text{if } t < 0 \\ \overline{w}'_N & \text{if } t > t_N \end{cases} \tag{24}$$

We now present the following Lemma.

Lemma 1. *For every $p, q \in \{0, \ldots, N\}$, $p < q$, we have:*

$$\underset{\tau \in [t_p, t_q]}{\operatorname{emax}} [w(\tau)] = \underset{\tau \in \{t_p, \ldots, t_q\}}{\operatorname{emax}} [w(\tau)] \tag{25}$$

The proof of Lemma 1 is skipped due to lack of space. Lemma 1 is useful to prove Lemma 2 below. It is also be useful for proving the result of section V-B.

Lemma 2. *For every $t \in \mathbb{R}$, $\overline{w}(t) \leq \overline{w}'(t)$.*

The proof of Lemma 2 is also skipped due to lack of space.

Using the result of Lemma 2, and the fact that: $G^{-1}A = G^{-1}\left(G + \frac{C}{\Delta t}\right) = I_n + G^{-1}\frac{C}{\Delta t} \geq 0$ since $G^{-1} \geq 0$ and $C \geq 0$, we can write using (16):

$$v_h(t) \leq G^{-1}A\overline{w}(t) \leq G^{-1}A\overline{w}'(t) \triangleq \overline{v}'(t) \tag{26}$$

which means that $\overline{v}'(t)$ is also an envelope waveform representing an upper bound on the voltage drop at any time t. Because $\overline{w}'(t)$ is piece-wise linear, $\overline{v}'(t)$ must also be piece-wise linear because $\overline{v}'(t)$ is the result of applying a linear transformation $G^{-1}A$ to $\overline{w}'(t)$. Accordingly, one can find $\overline{v}'(t)$ by finding $\overline{v}'_0 = \overline{v}'(t_0)$, $\overline{v}'_1 = \overline{v}'(t_1)$ $\ldots$, $\overline{v}'_N = \overline{v}'(t_N)$ using the following linear systems:

$$G\overline{v}'_k = A\overline{w}'_k \quad \forall k \in \{0, 1, \ldots, N\} \tag{27}$$

so that the result is piece-wise linear with the following breakpoints:

$$\overline{\mathcal{V}}' = \left\{\left(t_0, \overline{v}'_0\right), \left(t_1, \overline{v}'_1\right), \ldots, \left(t_N, \overline{v}'_N\right)\right\} \tag{28}$$

In fact, because $G^{-1}A = I_n + G^{-1}B$, and to avoid the matrix vector multiplication $A\overline{w}'_k$, we first solve the system $G\delta_k = B\overline{w}'_k$ for δ_k,

Algorithm 1 generate_transient_envelope

Input: G, C, $\mathcal{I}$, h
Output: Envelope waveform $\overline{\mathcal{V}}'$
1: **for** $k = 0, \ldots, N$ **do**
2: Solve $Aw_k = Hi_k$ for w_k
3: **if** $k = 0$ **then**
4: Set $\overline{w}'_k = w_k$
5: **else**
6: Set $\underline{t_{k-1}} = \begin{cases} t_0 & \text{if } t_k - h < t_0 \\ \max\limits_{\substack{t \in T \\ t \le t_{k-1}-h}} (\ t\ ) & \text{if } t_k - h \ge t_0 \end{cases}$
7: Set $\overline{w}'_k = \underset{\tau \in \{\underline{t_{k-1}}, \ldots, t_k\}}{\text{emax}} [w(\tau)]$
8: **end if**
9: Solve $G\delta_k = B\overline{w}'_k$ for δ_k
10: Set $\overline{v}'_k = \delta_k + \overline{w}'_k$
11: **end for**
12: **return** $\overline{\mathcal{V}}' = \{(t_0, \overline{v}'_0), (t_1, \overline{v}'_1), \ldots, (t_N, \overline{v}'_N)\}$

and then set $\overline{v}'_k = \delta_k + \overline{w}'_k$. This approach requires finding $B\overline{w}'_k$, which is easier than finding $A\overline{w}'_k$ because B is a diagonal matrix.

Notice that for $t < 0$, we have $\overline{v}'(t) = 0$ and for $t > t_N$, we have $\overline{v}'(t) = \overline{v}'(t_N)$.

Algorithm 1 presents a high level description of how to find the envelope waveform $\overline{v}'(t)$. The algorithm requires $2N + 2$ system solves which can be done using a Cholesky factorization [13] of both A and G. Cholesky factorization can be used because both A and G are positive definite matrices.

V. DC Envelope Waveforms

In many cases, simple DC bounds on the true voltage drop waveforms are sufficient for power grid dynamic verification. In this section, we will show how such DC bounds can be generated and used for verification. These bounds turn out to be easier to compute.

A. General Case

Recall that $w(t) = A^{-1}Hi(t)$ and consider the vector:

$$\overline{W} = \underset{\tau \in \mathbb{R}}{\text{emax}} [w(\tau)] \tag{29}$$

Clearly:

$$A^{-1}Hi(t - q\Delta t) \le \overline{W} \quad \forall q \ge 0 \tag{30}$$

Therefore, because $\left(A^{-1}B\right)^q \ge 0, \forall q \ge 0$, we have using (10):

$$\begin{aligned} v(t) &\le \sum_{q=0}^{\infty} \left(A^{-1}B\right)^q \overline{W} = \left(\sum_{q=0}^{\infty} \left(A^{-1}B\right)^q\right) \overline{W} \\ &= \left(I_n - A^{-1}B\right)^{-1} \overline{W} \quad \text{(due to (15))} \\ &= G^{-1}A\overline{W} \triangleq \overline{V} \quad \text{(as in (16))} \end{aligned}$$

Effectively, $\overline{V}$ is a DC envelope waveform representing an upper bound on the voltage drop at any time t. In other words, if we solve for the DC waveform $\overline{W}$ using (29), then we can solve for the DC envelope waveform $\overline{V}$ using:

$$G\overline{V} = A\overline{W} \tag{31}$$

B. Case of Piece-Wise Linear Currents

As in section IV-B, solving for $\overline{W}$ using (29) is expensive because it requires solving an emax operation over $\mathbb{R}$. However, the case of piece-wise linear loading currents is much simpler. After finding $w(t)$ (which is also piece-wise linear) as in (19), we find:

$$\overline{W}' = \underset{\tau \in T}{\text{emax}} [w(\tau)] \tag{32}$$

Algorithm 2 generate_dc_envelope

Input: G, C, $\mathcal{I}$
Output: DC envelope $\overline{V}'$
1: **for** $k = 0, \ldots, N$ **do**
2: Solve $Aw_k = Hi_k$ for w_k
3: **end for**
4: $\overline{W}' = \underset{\tau \in T}{\text{emax}} [w(\tau)]$
5: Solve $G\delta = B\overline{W}'$
6: Set $\overline{V}' = \delta + \overline{W}'$
7: **return** $\overline{V}'$

Using Lemma 1 with $p = 0$ and $q = N$, and because $w(t) = 0$ for $t < t_0$ and $t > t_N$, it should be clear that $\overline{W} = \overline{W}'$. Therefore, a good way of finding a DC bound is by computing $\overline{W}'$ using (32), and then solving:

$$G\overline{V}' = A\overline{W}' \tag{33}$$

for $\overline{V}'$, which is equal to $\overline{V}$ because $\overline{W} = \overline{W}'$. As in section IV-B, and to avoid the matrix vector multiplication $A\overline{W}'$, we first solve $G\delta = B\overline{W}'$ for δ, and then set $\overline{V}' = \delta + \overline{W}'$.

Algorithm 2 presents a high level description of how to find the DC envelope $\overline{V}'$. The algorithm requires $N+1$ system solves which can be done, as before, using a Cholesky factorization of both A and G.

VI. Implementation

A. Computing the Duration h'

The duration h' is a parameter that depends on the dynamics of the power grid as a linear system. Observe that one can write the nodal analysis equations (4) in the standard form of a linear system as follows:

$$\dot{v}(t) = -C^{-1}Gv(t) + C^{-1}Hi(t) \tag{34}$$

so that the *system matrix* is $-C^{-1}G$, which is known to have negative real eigenvalues because $C^{-1}G$ has positive real eigenvalues due to the fact that both C and G are symmetric with positive real eigenvalues [11]. The exact solution, at any time t, of the differential equation (34) is:

$$v(t) = e^{-C^{-1}G\alpha}v(t - \alpha) + \int_{t-\alpha}^{t} e^{-C^{-1}G(t-\tau)}C^{-1}Hi(\tau)d\tau \tag{35}$$

for any $\alpha > 0$. We defined the duration h' as the largest significant duration of the impulse response of any node on the grid. This means that h' should be chosen such that if $\alpha = h'$ in (35), then the term $e^{-C^{-1}G\alpha}v(t - \alpha)$ is *negligible*. If this is true, then $v(t)$ can be approximated by the integral term of (35), which only requires information about the input in the duration $[t - h', t]$:

$$v(t) \approx \tilde{v}(t) \triangleq \int_{t-h'}^{t} e^{-C^{-1}G(t-\tau)}C^{-1}Hi(\tau)d\tau \tag{36}$$

The error vector introduced by this approximation is:

$$e(t) = v(t) - \tilde{v}(t) = e^{-C^{-1}Gh'}v(t - h') \tag{37}$$

Let $\eta > 0$ be a user-defined error tolerance on the voltage drop at every node. If h' is chosen such that:

$$\|e(t)\|_\infty \le \eta \tag{38}$$

then this guarantees that every entry of $\tilde{v}$ is at most η away from the corresponding entry of $v(t)$. Below, we show how to determine h' is order to guarantee $\|e(t)\|_\infty \le \eta$. The method requires an upper bound Υ on $\|v(t)\|_2$ for all t.

If the DC envelope waveforms from Section V are available, then one can choose Υ to be either $\|\overline{V}\|_2$ or $\|\overline{V}'\|_2$ (in the case of piece-wise linear currents). If the DC envelope waveforms are not available, then, because the voltage drop at every node is less than V_{dd}, we have:

$$\|v(t)\|_2 \leq \sqrt{\sum_{k=1}^{n} V_{dd}^2} = V_{dd}\sqrt{n}$$

so that Υ can be chosen to be $V_{dd}\sqrt{n}$.

Recall that $C^{-1}G$ has real positive eigenvalues. The Lemma below shows how to choose h' to guarantee $\|e(t)\|_\infty \leq \eta$.

Lemma 3. *Let $\lambda_{\min}$ be the smallest eigenvalue of $C^{-1}G$, and $c_{\max}$ and $c_{\min}$ be the largest and the smallest diagonal elements of C, respectively. If h' is chosen such that:*

$$h' \geq \frac{1}{\lambda_{\min}} \ln\left(\frac{\sqrt{c_{\max}/c_{\min}}}{\eta/\Upsilon}\right) \tag{39}$$

then:

$$\|e(t)\|_\infty \leq \eta \tag{40}$$

The proof of Lemma 3 is skipped due to lack of space. It remains to explain how to compute $\lambda_{\min}$. Notice that:

$$\lambda_{\min} = \frac{1}{\max\{\lambda | \lambda \in \sigma(G^{-1}C)\}}$$

Accordingly, computing $\lambda_{\min}$ requires computing the *dominant* eigenvalue of $G^{-1}C$. The dominant eigenvalue is the eigenvalue with the largest magnitude, which, in this case, is the also the maximum eigenvalue because all the eigenvalues of $G^{-1}C$ are real and positive. Finding the dominant eigenvalue of $G^{-1}C$ can be done using the *power method* [13]. Let x_0 be a non-zero vector in $\mathbb{R}^n$, and let $x_k = (G^{-1}C)^k x_0$. For sufficiently large powers of k, and according to the power method, x_k becomes a good approximation of the dominant eigenvector of $G^{-1}C$. To find the dominant eigenvalue, it remains to write:

$$\lambda_d^* \approx \frac{(G^{-1}Cx_k)^T x_k}{x_k^T x_k} \triangleq \lambda_d^{(k)}$$

where λ_d^* is the actual dominant eigenvalue of $G^{-1}C$, and $\lambda_d^{(k)}$ is the approximate dominant eigenvalue computed using the power method, at iteration k.

Finding $(G^{-1}C)^k x_0$ is done iteratively by computing $x_k = G^{-1}Cx_{k-1}$, which is done by solving the system $Gx_k = Cx_{k-1}$ using a Cholesky factorization of G, and a series of forward/backward solves for $k = 1, 2, \ldots$ As for the stopping criterion, we stop when the ratio $\left|\frac{\lambda_d^{(k)} - \lambda_d^{(k-1)}}{\lambda_d^{(k-1)}}\right|$ becomes smaller than some parameter ϵ.

Once an approximation $\lambda_d^{(k)}$ of λ_d^* is found, we can write $\lambda_{\min} = \frac{1}{\lambda_d^{(k)}}$. This leads to a value of h' that can be computed using (39). The overall procedure for finding h' is outlined in Algorithm 3.

Notice that Lemma 3 is only useful if the lower bound provided on h' is positive, which can be achieved if:

$$\eta \leq \Upsilon \tag{41}$$

This makes the term inside the Logarithm in (39) greater than 1 because $\sqrt{c_{\max}/c_{\min}} \geq 1$. Thus, the lower bound on h' becomes positive.

Condition (41) is an easy condition to achieve because η is a small error tolerance (no more than few μV) that guarantees that $v(t)$ and $\tilde{v}(t)$ are almost identical for every t. If $\Upsilon = V_{dd}\sqrt{n}$, then clearly $\eta \leq \Upsilon$ because V_{dd} is usually in the range of several hundreds of millivolts. If $\Upsilon = \|\overline{V}\|_2$ (or $\|\overline{V}'\|_2$), then $\Upsilon \geq \|\overline{V}\|_\infty$ (or $\|\overline{V}'\|_\infty$) which is an upper bound on the largest peak voltage drop achieved. This is usually in the range of 50-60 mV for modern power grids. Therefore, $\eta \leq \Upsilon$ is also easily achieved.

Algorithm 3 find_h'

Input: G, C, η, ϵ
Output: h'

$x_0 = [1\ 1\ \ldots\ 1] \in \mathbb{R}^n$, $k = 0$
Solve $Gx_1 = Cx_0$ for x_1
$\lambda_d^{(1)} = \frac{x_1^T x_0}{x_0^T x_0}$
while $\left(\left|\frac{\lambda_d^{(k)} - \lambda_d^{(k-1)}}{\lambda_d^{(k-1)}}\right| \geq \epsilon\right)$ **do**
  $k = k + 1$
  Solve $Gx_{k+1} = Cx_k$ for x_{k+1}
  $\lambda_d^{(k)} = \frac{x_{k+1}^T x_k}{x_k^T x_k}$
end while
$\lambda_{\min} = \frac{1}{\lambda_d^{(k)}}$
Find $c_{\min}$, $c_{\max}$, and Υ.
return $\frac{1}{\lambda_{\min}} \ln\left(\frac{\sqrt{c_{\max}/c_{\min}}}{\eta/\Upsilon}\right)$

B. Time-Step

Even though our approach only considers the breakpoints in the input current waveforms, a time-step Δt is still needed in order to compute the matrix A. We choose Δt based on the algorithm proposed in [14]. The algorithm requires computing the dominant eigenvalues of $G^{-1}C$, which we are already doing as part of the algorithm that computes h' (Algorithm 3). The resulting time-step is simply:

$$\Delta t = \lambda_d^{(k)} \tag{42}$$

This choice of Δt was proven to be useful in the case of *vectorless* RC verification in [14] where the authors have used a similar upper bound to the one we are using in this paper. As we will see in the results section, the same choice of Δt turns out to be adequate for our approach as well.

VII. Experimental Results

A C++ implementation has been written to test Algorithms 1, 2, and 3. All the linear systems are solved using Cholmod [15] from SuiteSparse [16]. The tested grids were generated based on user specifications, including grid dimension, metal layers, pitch and width per layer, and C4 pad and current sources distribution. The technology specifications were consistent with 1V 45nm CMOS technology. The input current waveforms are piece-wise linear with breakpoints that are generated based on a certain average value per current source. The duration between two consecutive breakpoints is randomly chosen inside a user-defined range. Computations were done using a 3.4 GHz Linux machine with 32 GB of RAM.

In order to verify the accuracy of our approaches, we compare the resulting envelopes with the exact voltage drop waveforms resulting from an HSPICE simulation of the grid under the same set of input current waveforms. We report the maximum and average overestimation error of the DC envelopes and the peaks of the transient envelopes as compared to the peaks of the exact voltage drop waveforms. We observe that the overestimation error is no more than 3.91 mV with an average that is smaller than 1.92 mV, for the first 4 grids tested. The overestimation errors are not available for grids G5, G6, G7, and G8 because HSPICE takes a long time to finish the simulation for these grids. This shows that our proposed simulation-based approaches are very accurate, and can be useful for the purpose of checking the safety of the power grids because they provide conservative bounds on the voltage drops. Figure 2 shows the exact voltage drop waveform and the resulting envelope waveforms for a particular node in a certain power grid. We can observe how well the transient envelope waveform follows the peaks of the exact waveform, making it very useful for verification purposes. Using the

TABLE I
SPEED AND ACCURACY OF THE PROPOSED APPROACHES

Grid			PowerRush	DC envelope				Transient envelope			
Name	Nodes	h' (ns)	Runtime* (sec)	Maximum Error (mV)	Average Error (mV)	Runtime (sec)	Speed-up	Maximum Error (mV)	Average Error (mV)	Runtime (sec)	Speed-up
G1	13,736	5.70	5.72	3.91	1.91	0.46	**12.43X**	3.35	1.36	0.93	**6.15X**
G2	31,038	6.04	20.81	3.39	1.92	1.08	**19.27X**	3.37	1.39	2.20	**9.46X**
G3	55,118	21.83	65.72	2.90	1.13	2.24	**29.34X**	2.81	1.00	4.68	**14.04X**
G4	126,908	21.40	222.17	2.48	1.09	5.04	**44.08X**	2.31	0.96	10.55	**21.06X**
G5	863,436	21.44	2542.37	-	-	40.49	**62.79X**	-	-	83.16	**30.57X**
G6	951,676	21.60	3310.77	-	-	45.52	**72.73X**	-	-	93.29	**35.49X**
G7	1,071,148	21.76	2030.32	-	-	52.32	**38.81X**	-	-	107.10	**18.96X**
G8	1,668,874	24.06	3897.40	-	-	83.72	**46.55X**	-	-	171.66	**22.70X**

$*$ The runtime of PowerRush for the grids with no HSPICE or PowerRush data available was found by linear interpolation

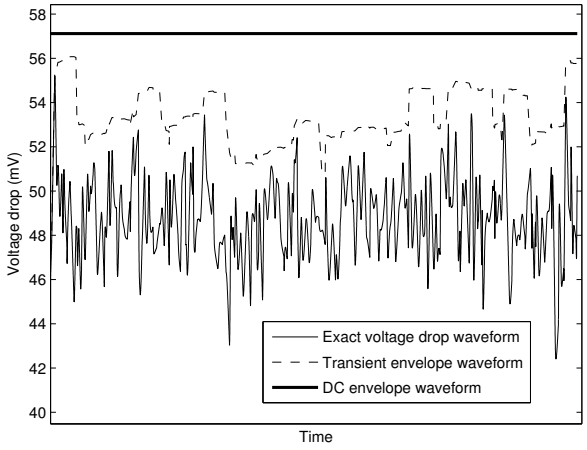

Fig. 2. Envelope waveforms at a certain node in a 1596-nodes grid

transient envelope, one can figure out if the node is safe or not. If the node is not safe, the envelope waveform provides the user an idea of the time intervals at which the node is becoming unsafe. This could potentially be very useful for debugging purposes.

In terms of speed, we compare our approaches to PowerRush [7]. To do that, we compute the runtime required by one iteration of PowerRush from the table reported in [7], for power grids that have almost the same size of ours, and multiply the result by the number of iterations required by HSPICE to simulate the grid subject to the input current waveforms that we are choosing. Essentially, the result is approximately the time required by PowerRush to simulate our grids if it were to use the same time-step control algorithm as HSPICE. The resulting durations are reported in the fourth column of Table I. The runtime of our algorithms is also reported along with the corresponding speed-ups over PowerRush. The values reported include the runtime required to find Δt and h'. We notice that computing the DC envelopes can be up to 72.73X faster than running PowerRush while computing the transient envelopes can be up to 35.49X times faster. This shows that our approaches are quiet practical and present significant performance advantages over the existing tools since they require solving a much smaller number of linear systems.

VIII. CONCLUSION

In this paper, we introduced a fast and efficient simulation-based approach for checking RC power grids. The algorithms we proposed generate envelope upper bound waveforms on the exact voltage drop waveforms. The overestimation of the envelopes was shown to be minimal and the speedup over modern simulation tools was shown to be dramatic. The generated envelopes can be very useful for checking the safety of the grid, as well as for debugging purposes in the cases where particular nodes were found to be unsafe.

REFERENCES

[1] F. N. Najm, *Circuit Simulation*. Hoboken, NJ: John Wiley & Sons, Inc, 2010.

[2] Z. Zeng, T. Xu, Z. Feng, and P. Li, "Fast static analysis of power grids: Algorithms and implementations," in *ACM/IEEE ICCAD*, San Jose, CA, Nov. 7-10 2011.

[3] C.-H. Chou, N.-Y. Tsai, H. Yu, C.-R. Lee, Y. Shi, and S.-C. Chang, "On the preconditioner of conjugate gradient method - A power grid simulation perspective," in *ACM/IEEE ICCAD*, San Jose, CA, Nov. 7-10 2011.

[4] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Power grid analysis using random walks," *IEEE TCAD*, vol. 24, no. 8, August 2005.

[5] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "A multigrid-like technique for power grid analysis," *IEEE TCAD*, vol. 21, no. 10, Oct. 2002.

[6] M. Zhao, R. V. Panda, S. S. Sapatnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks," *IEEE TCAD*, vol. 21, no. 2, Feb. 2002.

[7] J. Yang, Z. Li, Y. Cai, and Q. Zhou, "PowerRush: Efficient transient simulation for power grid analysis," in *ACM/IEEE ICCAD*, San Jose, CA, Nov. 5-8 2012.

[8] T. Yu and M. D. F. Wong, "PGT_SOLVER: An efficient solver for power grid transient analysis," in *ACM/IEEE ICCAD*, San Jose, CA, Nov. 5-8 2012.

[9] X. Xiong and J. Wang, "Parallel forward and back substitution for efficient power grid simulation," in *ACM/IEEE ICCAD*, San Jose, CA, Nov. 5-8 2012.

[10] H. Zhuang, S.-H. Weng, J.-H. Lin, and C.-K. Cheng, "MATEX: A distributed framework for transient simulation of power distribution networks," in *IEEE DAC*, San Francisco, CA, June 1-5 2014.

[11] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. Cambridge University Press, 2012.

[12] I. A. Ferzli, F. N. Najm, and L. Kruse, "A geometric approach for early power grid verification using current uncertainties," in *ACM/IEEE International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, Novermber 5-8 2007.

[13] Y. Saad, *Iterative methods for sparse linear systems*, 2nd ed. SIAM, 2003.

[14] M. Fawaz and F. N. Najm, "Accurate verification of RC power grids," in *ACM/IEEE DATE*, Dresden, Germany, March 14-18 2016.

[15] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam, "Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate," *ACM Transactions on Mathematical Software*, vol. 35, no. 3, pp. 22:1–22:14, 2008.

[16] T. A. Davis. Suitesparse 4.4.6. [Online]. Available: http://faculty.cse.tamu.edu/davis/suitesparse.html