Fast Electromigration Simulation for Chip Power Grids

Bijan Shahriari
ECE Dept, University of Toronto
Toronto, Ontario, Canada
b.shahriari@alum.utoronto.ca

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

***Abstract*—Electromigration (EM) continues to be a serious concern for large chip design. We are focused on EM in the on-chip power grid, because grid lines carry mostly unidirectional currents and because of the very large sizes of modern grids. In the last few years, the capability to simulate EM has become available by simulating the stress in metal lines, which is the main cause of EM-induced failures. In this work, we have improved on the state of the art by developing a new EM simulator that is both faster and has better features than previous work. The work builds on recent results on the equivalence between stress and voltage, and introduces both a model reduction technique that provides up to 4.2X speedup, and a very efficient method for updating the grid currents during the void growth phase.**

***Index Terms*—Electromigration, power grid, stress, equivalent circuit model, void, model order reduction.**

I. Introduction

In a metal line with significant current density, free electrons push and move the metal atoms in the direction of the electron wind; hence the name *electromigration* (EM) for this effect [1]. The resulting atomic flow increases compressive stress at one end of the line (anode) and tensile stress at the other (cathode), which creates a stress gradient that presents an opposing driving force that retards EM [2]. If the level of stress becomes high enough, a void may be created due to high tensile stress near the cathode, or a hillock (metal extrusion through cracks in the dielectric) may form due to high compressive stress near the anode. With the confinement of metal lines in today's technology, voids are much more likely than hillocks and so we are focused on void related failures. A void is created once the stress exceeds a so-called *critical stress*, denoted σ_{crit}. As it grows, the void restricts the line cross-sectional area and increases the line's resistance, up to some high value, even an open-circuit in some cases, which can be manifested as a circuit functional or timing failure.

With continued technology scaling, EM in metal lines has become a major reliability concern for integrated circuits. On-chip metal lines (interconnect) are mostly either signal lines, including intra- and inter-cell connectivity, or power supply and ground lines, whose purpose is to deliver a well-regulated supply voltage across the whole die. We will use the term *power grid* to refer to either the power supply network, the ground network, or both. While signal and clock lines do suffer from EM degradation, these lines carry bidirectional current and so have longer lifetimes due to stress reversal and the healing of micro-voids. In contrast, metal lines in the power grid carry mostly unidirectional current with no benefit of healing and so are more susceptible to EM failure. Thus, in the majority of cases, EM-induced chip failure is due to the failure of the power grid to deliver acceptable voltages to cells in the underlying circuit [3]. Hence the focus of this paper, as with much of the recent literature, is on EM in power grids.

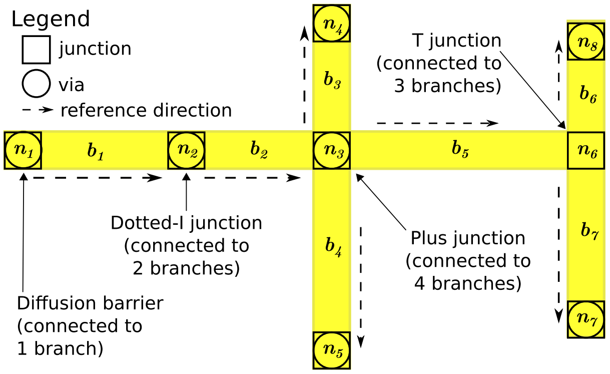
Legend
□ junction
○ via
reference direction
n_1
b_1
n_2
b_2
n_3
b_3
n_4
b_4
n_5
b_5
n_6
b_6
n_8
b_7
n_7
T junction (connected to 3 branches)
Dotted-I junction (connected to 2 branches)
Plus junction (connected to 4 branches)
Diffusion barrier (connected to 1 branch)

Fig. 1. Typical interconnect tree with various junction types [4].

Modern power grids span multiple layers (often all layers) of metallization and they consist of meshes of power and ground lines. Without loss of generality, we will focus on the power lines mesh. Under EM, metal atoms can travel between different connected branches on the same layer. However, they cannot travel through a via to other metal layers above and below, because of the metal liner under every via which acts as a barrier to atomic movement, but allows electron movement and electric current. As a result, EM-induced metal transport within a layer remains within that layer, so that the overall problem is decomposed into sub-problems, one for each connected metal structure in a layer that is physically disconnected from other such structures within that layer. The vast majority of these planar structures turn out to be trees, i.e., they have no cycles. So, it is typical in the field to simply use the term *interconnect trees* to refer to these metal islands; an example is shown in Fig. 1. However, note that our simulation engine can handle general graphs, not just trees. In an interconnect tree, the points where two or more lines meet,

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

979-8-3503-3475-3/23/$31.00 ©2023 IEEE

or where a line terminates, are called *junctions*. Junctions are where voids are most likely to nucleate, and so they are points of interest. Because of the mesh structure of modern power grids, even though the first void may increase the resistance in the grid, it may not actually lead to a failure. A void causes a failure only if it results in a voltage-drop violation, and that may not happen until several voids have nucleated.

Over the last several years, it has become harder to sign off on chip designs using state of the art EM checking tools, as there is very little margin left between the predicted EM stress, using existing tools, and that allowed by EM design rules [5]. This loss of safety margin can be traced back to the inaccurate and oversimplified nature of EM models used in existing tools. Standard practice in the industry is to break up a grid into isolated metal branches, then assess the reliability of each branch separately using Black's model [6]. This approach is inaccurate, especially for power grids, mainly because of the separate treatment of individual lines, which means that the movement of metal atoms from one line to another is not accounted for, a phenomenon that has a big impact on the stress distribution and void nucleation. Modern physics-based EM models do not have this problem, as we will summarize.

A. Stress-based EM simulation

Modern physics-based models for EM like the Korhonen model [7] have been found to be quite efficient as they are 1-dimensional (1D) and lend themselves to efficient analytical and numerical solutions. According to [8], "*The Korhonen model has been successfully used to explain a wide range of experimental behavior*." Once extended with boundary conditions, this model becomes applicable to multi-line structures like interconnect trees, and this has been done recently, as in the extended Korhonen model (EKM) model [4], [9]. EKM applies to whole interconnect trees, which overcomes the aforementioned inaccuracy of branch-by-branch analysis. Korhonen's equation is discretized in space and time to provide, for every interconnect tree, a linear time-invariant (LTI) system, i.e., $\dot{\sigma} = A\sigma + Bi'$, so it is amenable to efficient numerical simulation for large interconnect trees and power grids. Simulation of this system gives the stress $\sigma(t)$ evolution over time and allows tracking of EM damage, and the voltage drop on the grid, so as to determine the time to failure. However, the method has a couple of limitations. One is the simplifying assumption that void growth happens very quickly, instantaneously in fact, so that the problem of tracking of void growth over time and its impact on the grid resistances and currents is avoided completely. Effectively, the method *does not have a true void model*. Another is that the method discretizes every metal line into a number of smaller segments, so it can become expensive for large interconnect trees.

Others have followed up on the EKM work, including two recent publications in 2018 and 2020. In the first [10], the authors reduce the LTI system by projection onto a suitable Krylov subpace and report on the improved runtime. This is a standard and powerful method in EDA, and has been used with great success in the PRIMA [11] method for reduction of RC networks for circuit delay computation. However, projection is a matrix operation that becomes expensive for large systems and the practical experience with PRIMA reflects this fact. So using projection as part of the simulator to reduce the system for large trees or large power grids can become prohibitively expensive. In the second paper [12], a proper void growth model was added to the simulation engine flow. However, the authors used a void growth model that is based on an integral of the stress, which is really only applicable for single lines, not whole interconnect trees. A void model for multi-line interconnect trees requires the use of a flux based model, not an integral of stress, as described in [13, pg. 316] and [14], and more recently used in [15, pg. 56].

In this paper, we introduce a powerful method for speeding up the EM simulation that has several advantages. First, it's not built directly on the EKM formulation. Instead, we rebuild the numerical system from the ground up, using an alternative junction model, which brings advantages in the sparsity and numerical properties of the system matrices, leading to a more stable and robust simulation engine. Second, we develop and deploy reduced *line models* rather than reduced *tree models*, in contrast to [10], based on a recently discovered equivalence between stress and voltage [16]. In this way, the cost of doing the model order reduction becomes trivial, because we make use of pre-built parameterized reduced models for the metal lines that require no further work during the simulation time, which is a key advantage over projection based reduction methods. Finally, we make use of the proper flux-based void model for multi-line interconnect trees benefiting from the work mentioned above in [13], [14] and [15].

As a result, we will present a fast, accurate, and complete electromigration simulator. Given an input netlist for a chip power grid, this simulator accurately returns the mean time to failure (MTF) of that power grid. In contrast to EKM, the built-in void model allows the engine to track the instantaneous void length of the void in question as the simulation is running. Overall, we achieve about 3–4X speedup over published work for large grids.

II. Background

A. Equivalent circuit model

In recent work [16], an equivalence was established between the stress-flux relations in an interconnect tree and the voltage-current relations in a specially-constructed RC circuit corresponding to the interconnect tree. This circuit is a replica of the interconnect tree, in which every junction in the tree is replaced by a circuit *node* and every branch in the tree is replaced by a series chain of resistors, e.g., as shown in Fig. 2. The internal electrical nodes of this chain are called *internal nodes* while the terminal nodes of the chains, which correspond to junctions in the interconnect tree, are called *junction nodes*. The number of junction nodes is the same as the number of tree junctions m, and the total number of nodes in the circuit will be denoted $n \geq m$. Every node in the equivalent circuit is connected to a capacitor to ground, but there are no resistive connections to ground. This structure is

TABLE I

A NUMBER OF MATERIAL AND PHYSICAL CONSTANTS IN SI UNITS.

Name	Symbol	Value	Units
Boltzmann's Constant	k_b	1.380649E−23	J/K
Elementary Charge	$\lvert e\rvert$	1.602176634E−19	C
Effective Valence	Z	10.00	–
Effective Charge	q^*	1.60E−18	C
Atomic Volume	Ω	1.66E−29	m^3
Temperature	T	400.00	K
Resistivity (Copper)	ρ	3.00E−08 @400K	ohm.m
Activation Energy	E_a	1.00	eV
Diffusivity factor	D_0	5.20E−05	m^2/s
Atomic Diffusivity	D	1.31E−17 @400K	m^2/s
Effective Bulk Modulus	B	3.00E10	Pa

traditionally called an *RC network* in the circuits literature. Every junction node of this circuit is connected to an ideal *current source* to ground, so there are m current sources, captured in the $m \times 1$ vector u. The values of these current sources in u are computed based on the branch currents in the interconnect tree. It has been shown [16], that the voltages in this equivalent circuit are simply a scaled replica of the stresses in the tree, according to $v(t) = \xi\sigma(t)$, where $\sigma(t)$ is the stress at a node and ξ is a conversion factor whose value is $\xi = 1$ V/MPa, where MPa stands for Mega-Pascal, and Pascal is the SI unit of stress. This equivalence allows one to consider any number of circuit reduction and macromodelling techniques from traditional circuit analysis and simulation in order to speed up electromigration simulation.

III. OUR APPROACH

As mentioned earlier in Section I-A, the EKM method [9] formulates a linear system equation $\dot{\sigma} = A\sigma + Bi'$, which is numerically solved for the stress $\sigma(t)$; note, i' are the branch currents. Instead, and in order to overcome some limitations of EKM, our approach makes use of the equivalent circuit construction described in [16], as the starting point for a new formulation of the problem. For any interconnect tree, we start by considering its equivalent circuit, then use nodal analysis to construct the standard matrix equation for it, as

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

where G is the $n \times n$ conductance matrix of that equivalent circuit, C is the $n \times n$ strictly positive diagonal matrix that holds the circuit capacitance values, H is a $n \times m$ matrix that indicates which junction node is connected to which current source (in the equivalent circuit), and $u(t)$ is the $m \times 1$ vector of these source currents, which can be computed as linear combinations of the branch currents in the interconnect tree.

Based on the equivalence between stress σ (in the tree) and voltage v (in the equivalent circuit), as $v(t) = \xi\sigma(t)$, the values of stress in the interconnect tree in MPa are exactly the same as the voltage solution of (1) in Volts. To be strict about this, one can consider that (1) has been divided by ξ on both sides so that $v(t)$ has been converted to $\sigma(t)$ on the left hand side, while u retains its same original value but is now in different units, which is immaterial. In any case, we will rewrite (1) as

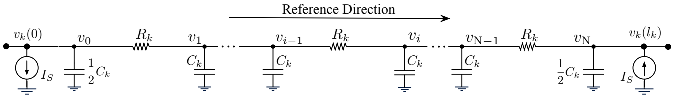

Fig. 2. Contribution of a metal branch to the tree equivalent circuit, in the nucleation phase.

$$Gx(t) + C\dot{x}(t) = Hu(t), \tag{2}$$

where $x(t)$ is a variable that at times will be the node voltage v or the stress σ; the meaning will be clear from the context. Replacing line stress analysis with node voltage analysis for the equivalent circuit allows one to make use of various circuit reduction and macromodelling techniques from traditional circuit analysis, as well as simulation, in order to speed up the stress simulation for electromigration verification. In this way, we have built an entire power grid EM verification simulation flow from the ground up using this new model. Indeed, using the discretized equivalent circuit mentioned above, we apply a model order reduction technique to reduce the dimensions of the system matrices.

So far, we have described the high-level view of this work: First, we build the system (2) for every interconnect tree very efficiently, while parsing the input netlist that describes the power grid, based on element stamps as in typical circuit simulators. Second, we solve these systems for the stress over time and monitor for the time to failure. The next two sections will describe some of the details that lead us to the advances and improvements over the state of the art.

IV. REDUCED LINE MODEL

A. The full model: PIRC20

For any interconnect tree, the equivalent circuit consists of an interconnection of RC chains, each as in Fig. 2, and each of which corresponds to a branch in the interconnect tree. Each chain is the result of discretization of a RC transmission line and consists of 20 RC segments, which has been found to offer a good approximation [16], hence the name PIRC20 for this model. For example, if two lines meet at a dotted-I junction (i.e., a degree 2 junction, as in Fig. 1), their RC chains would meet at the corresponding node in the equivalent circuit, as shown in Fig. 3. To be specific, the RC chain in Fig. 2 corresponds to a line without any voids, i.e., a line in the so-called *nucleation phase*. Once a void has formed (nucleated), we are in the *void growth phase* and a different RC chain model is used to represent the presence of the void, as in Fig. 4. The R_k and C_k values in the nucleation phase model (Fig. 2) are given in [16], as

$$R_k = \frac{k_b T \delta_k}{D_k a_k \psi} \quad \text{and} \quad C_k = \frac{a_k \delta_k \psi}{B\Omega}\ , \tag{3}$$

where the various terms in these equations are material and physical constants as listed in Table I, δ_k is the length of each line segment (length/20) and the ψ term is a scale factor, set

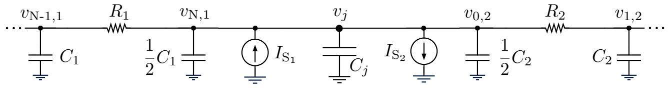

Fig. 3. Degree-2 junction of the discretized equivalent circuit.

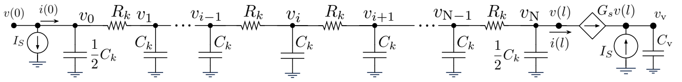

Fig. 4. Discretized equivalent circuit for a line with a void at $x = l$.

at $\psi = 0.01\text{C}^2/\text{m}^3$. The G_s and C_{v} values in the void growth phase model (Fig. 4) are also given, as

$$G_s = \frac{\psi D w h}{k_b T \delta_s} \qquad \text{and} \qquad C_{\text{v}} = \frac{\xi \psi w h}{\Omega \eta} \ , \tag{4}$$

where the term η is another conversion factor $\eta = 1\text{V}/\text{nm}$ and w, h are line width and thickness (height), respectively.

B. The reduced model: PACTN5

The PIRC20 line model entails 20 nodal equations, in as many unknowns. However in practice, interconnect trees may contain many thousands or hundreds of thousands of junctions, and therefore branches. So, the size of the system to be solved can become extremely large. In order to speed up the stress simulation, we have introduced a reduced line model, which we call PACTN5, as shown in Fig. 5. Note, negative capacitance is not uncommon when building reduced models. This reduction is based on a well-known algorithm for reducing RC networks using pole analysis via congruence transformations (PACT), as described in [17]. However, we have developed a way to apply the PACT algorithm to a *parameterized* line model, as in Fig. 2, in such a way that the dependence on the line parameters R_k and C_k remains explicit in the reduced model, as shown in Fig. 5. Briefly, the PACT algorithm works by first transforming the line equations into a form where the eigenvalues are all exposed, then only a few (in our case 5, hence the name) dominant eigenvalues are maintained while the rest are dropped. We then extract the circuit from the reduced system of equations while keeping the dependence on the parameters explicit. The details are omitted for lack of space, but we will show simulation results that validate the accuracy of this model. The key point though is that, with the PACTN5 model being parameterized as it is, computing the reduced model for a metal line in the simulator is as simple as plugging in the corresponding R_k and C_k for that line. As mentioned in Section I-A, it's clear that this reduction of the line models requires almost zero time cost to the simulator. This is in contrast to other reduction algorithms, which use Krylov subspace methods and are more expensive and difficult to accomplish.

V. Branch Current Update During Void Growth

Electromigration is typically simulated under given *effective currents* that represent the underlying circuit activity. These currents are "effective" in the sense that they are typically time averages of typical current waveforms that may result from transient circuit activity. This is standard practice in the field because EM is a very slow phenomenon, so that the impact of fast underlying circuit transients is averaged out over time. These effective currents are modelled as fixed (DC) current sources that load the grid. The corresponding currents in the grid branches are easily found by solving the linear system $Gv = i$, where G is the grid conductance matrix, i the grid current source vector, and v the vector of the grid node voltages. Then, a simple application of Ohm's law provides all the via and branch currents. This is done once at the beginning of the simulation, and may need to be recomputed if/once voids start forming.

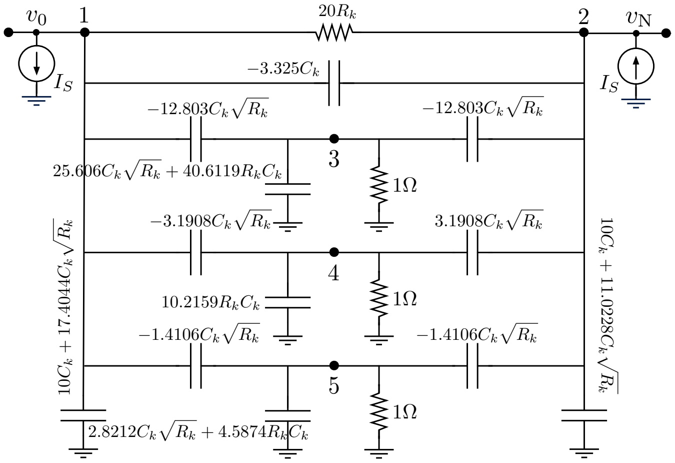

Fig. 5. PACTN5 reduced order model.

Once a void nucleates, the simulator can track its growth by monitoring the void length, which is in fact proportional to the voltage of the capacitor C_{v} in the voided line model in Fig. 4, specifically $l_{vd} = v_{\text{v}}/\eta$ [16]. As the void length grows over time, it increases the resistance of the branch that contains that void in a way that is easy to compute, thus altering the voltage of that junction node as well as potentially many other grid node voltages, and therefore also altering the branch and via currents. This is a complication that does not arise if one does not have a true void model, as was the case with EKM. It would be very expensive to recompute all branch currents by rebuilding and solving $Gv = i$ at every time point in the simulation process. Standard practice in the field is to only update (recompute) the currents once the resistance of the line in question has increased by some pre-set threshold relative amount, e.g., 20%. In a sense, this approach makes the simplifying assumption that the line resistance increases in a staircase fashion, rather than a continuous function as it actually is. Even then, re-doing a large number of LU factorizations to solve $Gv = i$ during the void growth phase can be prohibitively large.

To address this issue, we have developed an algorithm that takes advantage of the 'rank k update' that the power grid conductance matrix G undergoes if the grid has k voids that have just changed in size, or just appeared. A rank k update of a matrix A is when we add to it a matrix B of rank k. We will show how one can update all grid voltages and currents by doing a minimal amount of numerical work that requires LU factorization of a matrix of size only $k \times k$, as in (19), rather than the size of the whole grid, as follows.

Suppose we have a power grid with n nodes. When a void forms in a metal line, there is a change in the conductance of that line. Let $g_j(t_0) = 1/R$ be the conductance of line j pre-void at time t_0, and $g_j(t_1) = 1/(R + \Delta R)$, where $\Delta R > 0$ is the change in resistance of that line, from time t_0 to time $t_1 > t_0$. We define the *conductance correction* $\Delta g_j(t_1)$ to be

$$\begin{aligned}\Delta g_j(t_1) = g_j(t_1) - g_j(t_0) &= \frac{1}{R+\Delta R} - \frac{1}{R} \\ = -\frac{\Delta R}{R\,(R+\Delta R)} &= -g_j(t_0)\frac{\Delta R}{R+\Delta R}\,,\end{aligned} \tag{5}$$

where $\Delta g_j(t)$ is with respect to $g_j(t_0)$ where t_0 is the the start of the simulation time; a time when the grid was undamaged. This means that $\Delta g_j(t) \leq 0$, since the conductance can only decrease with respect to the undamaged line, as can be seen from the last expression in (5). We define a square matrix N of dimension $k \times k$, with k as the number of currently voided lines in the power grid, such that $N = -I_k$, where I_k is the identity matrix of size k and notice that $N^{-1} = N$. Let g_j be the conductance of a resistor connected between nodes p and q, with $p < q$. We then define the *difference conductance matrix* $\Delta G_j(t) \in \mathbb{R}^{n \times n}$, as the difference between $G_j(t)$ and G_0 where $G_j(t)$ is the (new) conductance matrix for the case where the line j is the only place with a void in the grid, and G_0 is the original conductance matrix of the undamaged power grid. As a result, we have $\Delta G_j(t) = u_j(t) N u_j(t)^T$ with $u_j(t)$ given by

$$u_j(t) = \sqrt{|\Delta g_j(t)|}(e_p - e_q), \tag{6}$$

where e_p and e_q are elementary column vectors with a 1 at the p^{th} and q^{th} rows respectively. Note that $\Delta G_j(t)$ takes the opposite sign of $u_j(t)u_j(t)^T$.

Suppose only line j in the power grid has voided between junctions p and q, which means $k = 1$. We then get

$$\Delta G_j(t) = \begin{bmatrix} \ddots & \vdots & \dots & \vdots & \vdots \\ \dots & \Delta g_j(t) & \dots & -\Delta g_j(t) & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \dots & -\Delta g_j(t) & \dots & \Delta g_j(t) & \dots \\ \vdots & \vdots & \dots & \vdots & \ddots \end{bmatrix}. \tag{7}$$

Therefore, $\Delta G_j(t)$ contains the opposite conductance stamps of $\Delta g_j(t)$. Furthermore, $\Delta G_j(t)$ is of rank 1 since it only contains two non-zero columns, which are negative of each other. We then extend this to the rank k case, as follows.

Let U be a matrix such that after k voided interconnects, we have $U(t) = [u_1(t), \dots, u_k(t)]$. Taking the negative outer product of this matrix as above, we get

$$U(t)NU(t)^T = \begin{bmatrix} -u_1(t) & \dots & -u_k(t) \end{bmatrix} \begin{bmatrix} u_1(t)^T \\ \vdots \\ u_k(t)^T \end{bmatrix} \tag{8}$$

$$= -u_1(t)u_1(t)^T + \cdots + -u_k(t)u_k(t)^T \tag{9}$$

$$= \Delta G_1(t) + \Delta G_2(t) + \cdots + G_k(t) \tag{10}$$

$$= \sum_{j=1}^{k} \Delta G_j(t). \tag{11}$$

We then define the *cumulative difference conductance matrix* (CDCM) $\sum_{j=1}^{k} \Delta G_j(t) = U(t)NU(t)^T$. This matrix contains the stamps of all the corrections $\Delta g_j(t)$ of every voided line. A large grid will typically develop a voltage drop violation after a relatively small number of voids have occurred; so voids are rare, compared to the number of branches. Therefore, as a simplifying assumption, we assume that a line can have no more than one void at a time, so there is only one $u_j(t)$ vector for each voided line. Furthermore, every stamp in this CDCM for a voided line j is composed of two linearly dependent column vectors (see (7)), which means every voided line increases the rank of the CDCM by 1. Therefore, $\sum_{j=1}^{k} \Delta G_j(t)$ is a matrix of rank k. With this, the new conductance matrix of this damaged power grid would have the form

$$G(t) = G_0 + \sum_{j=1}^{k} \Delta G_j(t) \tag{12}$$

Finally, we have an expression for the node voltages $v_k(t)$ of every junction in the power grid after k branch failures

$$\begin{aligned} v_k(t) = G(t)^{-1} i &= \left(G_0 + \sum_{j=1}^{k} \Delta G_j(t) \right)^{-1} i \\ &= \left(G_0 + U(t)NU(t)^T \right)^{-1} i. \end{aligned} \tag{13}$$

Using the Woodbury formula, defined in [18, pg. 19] as

$$(A + XRY)^{-1} = A^{-1} - A^{-1}X \left(R^{-1} + YA^{-1}X \right)^{-1} YA^{-1} \tag{14}$$

where A, X, R, and Y are matrices, we have

$$\begin{aligned} &\left(G_0 + U(t)NU(t)^T \right)^{-1} = \\ &G_0^{-1} - \left[G_0^{-1}U(t) \left(N^{-1} + U(t)^T G_0^{-1} U(t) \right)^{-1} U(t)^T G_0^{-1} \right]. \end{aligned} \tag{15}$$

Plugging (15) in (13) and using $N^{-1} = N$, we have

$$\begin{aligned} v_k(t) &= G_0^{-1} i \\ &- \left[G_0^{-1}U(t) \left(N + U(t)^T G_0^{-1} U(t) \right)^{-1} U(t)^T G_0^{-1} \right] i. \end{aligned} \tag{16}$$

Notice that the original grid node voltages are $v_0 = G_0^{-1} i$. Next, define

$$Z(t) = G_0^{-1} U(t) = [G_0^{-1} u_1(t), \dots, G_0^{-1} u_k(t)]. \tag{17}$$

It's clear that both v_0 and $Z(t)$ can be found using the same LU factorization, which would have been done at the beginning of the simulation. Also, v_0 remains constant through the entire simulation, therefore only needs to be calculated once. Therefore, to find $Z(t)$, we only need to perform k forward/backward substitutions, with k as the number of voided interconnects in the power grid. Finally, we define $W_k(t) = \left(N + U(t)^T Z(t)\right)$ and $y_k(t) = U(t)^T v_0$, so that

$$\begin{aligned} v_k(t) &= v_0 - Z(t)\left(N + U(t)^T Z\right)^{-1} U(t)^T v_0 && (18) \\ &= v_0 - Z(t) W_k(t)^{-1} y_k(t). && (19) \end{aligned}$$

This result is the "big deal," as it shows that all grid node voltages can be found using the small $k \times k$ matrix $W_k(t)$. It only remains to efficiently compute the vector $W_k(t)^{-1} y_k(t)$. Because the number of voided lines in a grid is relatively small (68 is the most we've seen; most cases are under 40), the dimensions of $W_k(t)$ will be small. As a result, even if this matrix is dense, it does not greatly affect the memory usage, nor the runtime. In fact, we can find $W_k(t)^{-1} y_k(t)$ using standard LU factorization on the system $W_k(t)x = y_k(t)$.

VI. Simulation results

This simulation flow has been implemented in C++ and has proven to be both accurate and fast. We will start with test cases of individual trees, including a one-line tree test case which has a known analytical solution, as well as a four-line test case. Then, we test the simulator on a number of full power grids, with the largest test case being 4.1M nodes in size. Similarly to the individual tree tests, we compare the full model to its reduced counterpart.

A. *One-line test case*

For the one-line test case, we assume the stress is initially zero all along the line. This test case has an analytical solution, described in [7] and [19], and gives the exact solution for the stress at time t at any location x along the line, as

$$\sigma(x,t) = \sigma_0 - \frac{q^* \rho j l}{\Omega} \left(\frac{1}{2} - \frac{x}{l} - 4 \sum_{n=0}^{\infty} \frac{1}{s_n^2} r_{x,t}(n) \right), \quad (20)$$

where $s_n = (2n+1)\pi$ and

$$r_{x,t}(n) = \cos\left(s_n \frac{x}{l}\right) \exp\left(-s_n^2 \frac{\kappa t}{l^2}\right), \quad (21)$$

with $\kappa = DB\Omega/k_b T$, and l the length of the line. We will use this exact solution to compare our simulation results from the discretized equivalent circuit (PIRC20) and the PACTN5 reduced order model. We set the line length $l = 250$ μm, height $h = 1$ μm, width $w = 1$ μm, temperature $T = 400$ K, and current density $j = 1 \times 10^9$ A/m^2. The resulting simulation plots are shown in Fig. 6. We observe that both the PIRC20 and PACTN5 models match the exact solution nearly perfectly.

For the case of one line with a void, there also exists an exact solution for the void length evolution, given in [14], as

$$\frac{l_{vd}}{l_{vd,sat}} = 1 + 4 \sum_{n=1}^{\infty} \frac{(-1)^n}{c_n^3} \exp\left(-c_n^2 \frac{\kappa t}{l^2}\right), \quad (22)$$

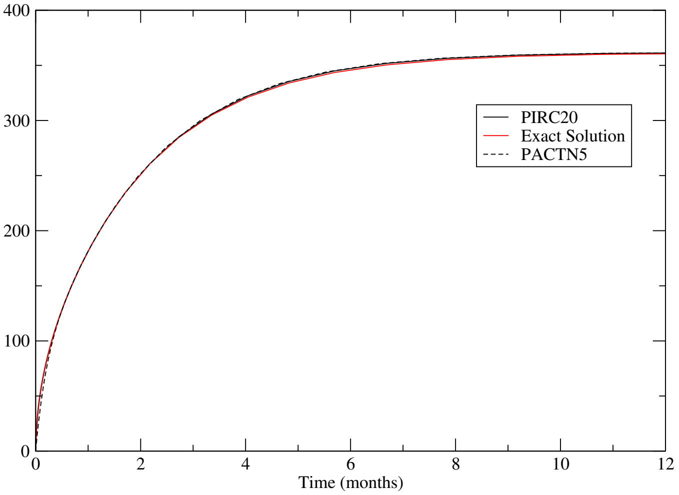

Fig. 6. Exact solution vs. PIRC20 vs. PACTN5 for the evolution of stress at the cathode of a single line.

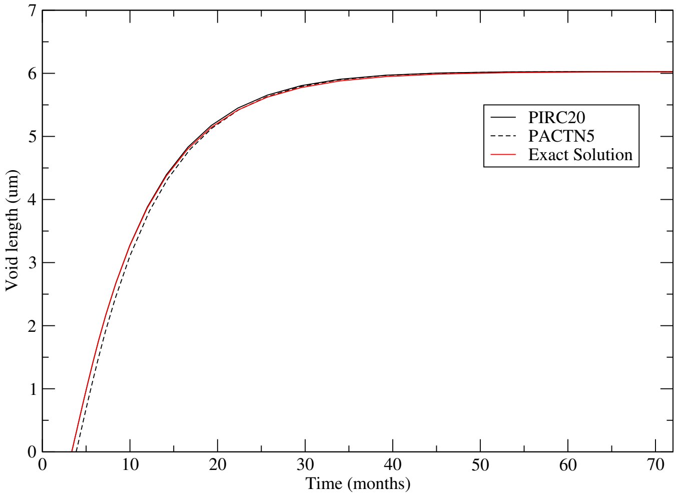

Fig. 7. Evolution of the void length at the cathode for the one line case for the PIRC20 and PACTN5 models.

where $c_n = (2n-1)\pi/2$ and $l_{vd,sat} = q^* \rho |j| l^2 / (2B\Omega)$. We set the critical stress to be $\sigma_{crit} = 600$ MPa and the line current density to be $j = 2 \times 10^9$ A/m^2. The void length evolution for the PIRC20, PACTN5, and exact solution are plotted in Fig. 7. We observe very tiny discrepancies between the PIRC20 and PACTN5 models at the beginning, but both models achieve the same void length steady state value. We also observe excellent agreement between PIRC20 and the exact solution. It should be noted that for this test case, we set all node stresses to be zero at the time of nucleation in order to get a fair comparison with the above exact solution, which assumes the same. Finally, we observe that the void reaches steady state after about 40 months. This shows the importance of having a true void model. EKM, which assumes the steady state is reached instantaneously, would be highly pessimistic in this case, under-estimating the failure time by about 3 years.

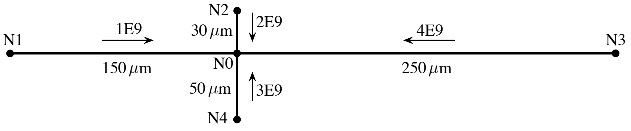

Fig. 8. Layout for a four-line configuration at a plus-junction.

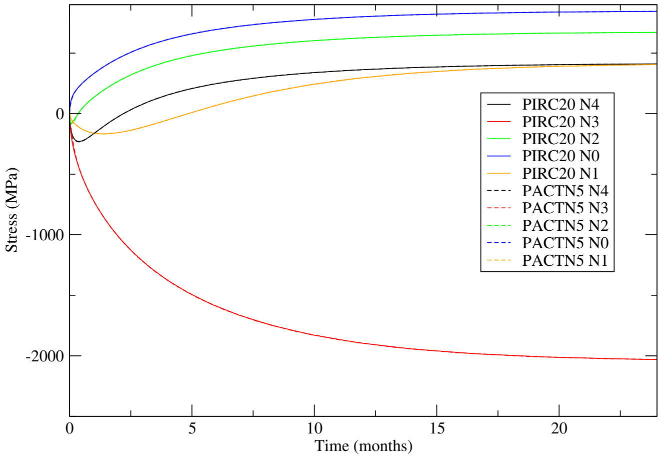

Fig. 9. PIRC20 vs. PACTN5 solution for the evolution of stress at the junctions for a four line plus-junction arrangement.

B. Four-line test case

A pseudo-layout for the four-line test case is shown in Fig. 8. All lines have a temperature T = 400 K, height $h = 1$ μm, and width $w = 1$ μm. The results of the simulations are displayed in Fig. 9, showing near perfect agreement between the discretized PIRC20 and reduced PACTN5 models.

C. Large tree simulations

We then randomly generated seven interconnect trees of increasing sizes, from 100 nodes to 4.1 million nodes, as shown in Table II. We simulated each of these trees for 12 years, using both the PIRC20 and the PACTN5 reduced models, with the void model turned off, so to speak, in order to test the raw speed-up in the simulator due to the reduced model. The results are shown in the table, where we notice that the reduced model provides *significant speed-up*, over 4X for very large trees.

TABLE II
RUNTIMES FOR PIRC20 AND PACTN5 MODELS FOR DIFFERENT SIZED INTERCONNECT TREES.

	CPU Time		
Tree Junction Size	PIRC20	PACTN5	Speedup
100	158 msec	114 msec	1.35 X
1000	414 msec	160 msec	2.6 X
100,000	43 sec	9 sec	4.7 X
1,000,000	8 min	2 min	4 X
2,000,000	15 min 30 sec	3 min 30 sec	4.4 X
3,000,000	23 min 20 sec	5 min 30 sec	4.2 X
4,100,000	39 min	9 min 30 sec	4.1 X

D. Power grid simulations

We performed tests on various power grids from the IBM benchmarks [20], as well as our own custom internally generated power grids. In Table III, “Size” refers to the number of junctions in the power grid. We performed multiple complete simulations of every power grids inside a Monte Carlo loop, in order to determine their MTFs. In this Monte Carlo loop, we considered the line diffusivity D_0 and the junction critical stress threshold σ_{crit} as the underlying statistical manufacturing variations. For the Monte Carlo randomization, the lognormal mean of the diffusivity was set to $\mu_{D_0} = 5 \times 10^{-11}$ m^2/s and its standard deviation was set to $\sigma_{D_0} = 0.05 \times \mu_{D_0}$. As for the critical stress of each junction, we set the mean to be $\mu_{\sigma_{\text{crit}}} = 600$MPa and the standard deviation $\sigma_{\sigma_{\text{crit}}} = 0.05 \times \mu_{\sigma_{\text{crit}}}$. Finally, we set the confidence and relative error for the Monte Carlo analysis to be $1 - \alpha = 95\%$ and $\epsilon = 5\%$, respectively. To properly compare the PIRC20 and PACTN5 models, as well as to have reproducible test cases, we seeded the pseudo-random number generator with the current iteration number of the Monte Carlo loop. Since the seeding is the same for both models, we can accurately compare them. The results for these simulations are shown in Table III, where we once again notice that the PACTN5 model is roughly as accurate, and much faster than the PIRC20 model for all test cases.

E. Comparison with published work

In this section, we compare the runtimes with the published runtime data for EKM [9]. We compare only the runtimes, between EKM and our two models, for the case when a single TTF (time-to-failure) is being computed. Because of the absence of a true void model in EKM, it makes no sense to compare their TTFs, or their MTFs for that matter. EKM would give much shorter lifetimes because it assumes the void growth size is zero. For these reasons, we only compare runtimes between the two engines.

We see from Table IV that our engine, with both the PIRC20 and PACTN5 models, is much faster than EKM. This is due in part to the reduced model in PACTN5, but also to our multi-rate parallel simulation scheme, whereby every interconnect tree is simulated with its own sequence of time-steps and time-points, a feature which EKM does not have. EKM however has a *filtering* mode, which drastically reduces its overall CPU runtime. This filtering scheme essentially determines which trees in the power grid are deemed to be *immortal* and removes them from the simulation altogether. This permits the simulator to simulate a significantly reduced number of trees. However, this type of scheme does not work in the context of a true void model because the change of currents during void growth may make a previously immortal tree become suddenly mortal, so it would be unfair to compare our engine to EKM in the filtering mode.

VII. CONCLUSION

In this work we have developed a complete EM simulation engine for chip power grid verification, from the ground up. This engine simulates the power grid interconnect tree

TABLE III

Comparison of runtime and MTF between the PIRC20 model and the PACTN5 (reduced) model.

Power Grid		MC Iteration count		CPU Time			MTF (yrs)		
Name	Size	PIRC20	PACTN5	PIRC20	PACTN5	Speedup	PIRC20	PACTN5	Error (%)
ibmpg1	11,572	22	29	5 min 20 sec	2 min	2.5X	11.14	11.15	0.09
internalpg1	36,863	30	30	32 min	11 min	2.9 X	20.11	19.97	0.7
ibmpg2	61,797	19	13	14 min 40 sec	6 min 40 sec	2.17X	9.43	9.87	4.61
internalpg2	146,113	10	16	2 hrs 50 min	2 hrs 30 min	1.14X	15.67	15.16	3.25
ibmpg3	410,505	11	10	3 hrs 20 min	47 min	4.2X	5.01	5.05	0.8
ibmpg4	474,836	6	5	3 hrs 6 min	45 min	4.1X	5.98	6.04	1
internalpg4	1,232,261	5	8	7 hrs 40 min	5 hrs 36 min	1.35X	10.07	10.29	2.18
internalpg6	1,643,815	5	5	3 hrs	56 min	3X	5.02	5.12	2
internalpg5	1,703,447	13	14	14 hrs	5 hrs 38 min	2.5X	11.08	11.07	0.09
internalpg7	2,629,449	7	6	16 hrs	5 hrs	3.2X	12.78	12.85	0.55
internalpg8	3,235,095	14	15	25 hrs	8 hrs 30 min	2.9X	11.08	11.22	1.26
internalpg9	4,094,705	30	30	2 days	14 hrs	3.4X	11.02	10.94	0.73

TABLE IV

Comparison of runtimes between EKM, PIRC20 and PACTN5.

Power Grid	**CPU Time**		
Name	**EKM**	**PIRC20**	**PACTN5**
ibmpg1	2 min 18 sec	13 sec	6 sec
internalpg1	2 min	1 min 5 sec	25 sec
ibmpg2	3 min 12 sec	44 sec	36 sec
internalpg2	39 min	17 min	8 min
ibmpg3	31 min	15 min	10 min
ibmpg4	1 hr	35 min 40 sec	13 min 20 sec
internalpg3	1 hr 42 min	19 min	3 min 50 sec
internalpg4	7 hr 50 min	1 hr 28 min	45 min

stress dynamics based on a recently described equivalent circuit model for electromigration. Furthermore, the simulator contains an accurate void growth model, which allows reliable tracking of void lengths and therefore the line resistances over time. Our novel contribution to this part of the flow is a method for updating the circuit currents once a resistance change occurs with a computational cost that depends on the number of voids rather than the grid size. This results in a faster and more accurate estimate of the impact of a void on the power grid lifetime. We also developed a reduced-order line model to speed up the simulation, which stems from the RC transmission line properties in the equivalent circuit formulation. We compared the runtime and accuracy of the full model and the reduced model for full power grid MTF simulation. The reduced model proves to be as accurate, and up to 3.4X faster, than the full model for a 4.1 million node power grid. The largest observed speedup was 4.2X on a grid with 410,000 nodes.

References

[1] P. S. Ho, C.-K. Hu, M. Gall, and V. Sukharev, *Electromigration in Metals*. Cambridge, UK: Cambridge University Press, 2022.

[2] I. A. Blech, "Electromigration in thin aluminum films on Titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, Apr. 1976.

[3] V. Sukharev, "Beyond Black's equation: Full-chip EM/SM assessment in 3D IC stack," *Microelectronics Engineering*, vol. 120, pp. 99–105, May 2014.

[4] S. Chatterjee, V. Sukharev, and F. N. Najm, "Fast physics-based electromigration checking for on-die power grids," in *IEEE/ACM International Conference on Computer-Aided Design*, Austin, TX, Nov. 2016.

[5] A. S. Oates, "Interconnect reliability challenges for technology scaling: A circuit focus," in *IEEE International Interconnect Technology Conference / Advanced Metallization Conference (IITC/AMC)*, San Jose, CA, May 2016, pp. 59–59.

[6] J. R. Black, "Electromigration failure modes in aluminum metallization for semiconductor devices," *Proceedings of the IEEE*, pp. 1587–1594, Sep. 1969.

[7] M. A. Korhonen, P. Borgesen, K. N. Tu, and C.-Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, Apr. 1993.

[8] Z.-S. Choi, J. Lee, M. K. Lim, C. L. Gan, and C. V. Thompson, "Void dynamics in Copper-based interconnects," *Journal of Applied Physics*, vol. 110, no. 03, Aug. 2011.

[9] S. Chatterjee, V. Sukharev, and F. N. Najm, "Power grid electromigration checking using physics-based models," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 7, pp. 1317–1330, Jul. 2018.

[10] C. Cook, Z. Sun, E. Demircan, M. D. Shroff, and S. X.-D. Tan, "Fast electromigration stress evolution analysis for interconnect trees using Krylov Subspace method," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 5, pp. 969–980, May 2018.

[11] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Transactions on Computer-Aided Design*, vol. 17, no. 8, pp. 645–654, Aug. 1998.

[12] Z. Sun, S. Yu, H. Zhou, Y. Liu, and S. X.-D. Tan, "EMSpice: Physics-based electromigration check using coupled electronic and stress simulation," *IEEE Transactions on Device and Materials Reliability*, vol. 20, no. 2, pp. 376–389, Jun. 2020.

[13] Z. Suo, "8.08 - Reliability of Metal Interconnects," in *Comprehensive Structural Integrity*, I. Milne, R. Ritchie, and B. Karihaloo, Eds. Oxford: Pergamon, Jan. 2003, pp. 265–324. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0080437494081258

[14] J. He and Z. Suo, "Statistics of electromigration lifetime analyzed using a deterministic transient model," *AIP Conference Proceedings*, vol. 741, no. 1, pp. 15–26, Dec. 2004.

[15] V. Sukharev, A. Kteyan, and X. Huang, "Postvoiding stress evolution in confined metal lines," *IEEE Transactions on Device and Materials Reliability*, vol. 16, no. 1, pp. 50–60, Mar. 2016.

[16] F. N. Najm, "Equivalent circuits for electromigration," *Microelectronics Reliability*, vol. 123, pp. 114 200:1–16, Aug. 2021.

[17] K. J. Kerns and A. T. Yang, "Stable and efficient reduction of large, multiport RC networks by pole analysis via congruence transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 734–744, 1997.

[18] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[19] H.-B. Chen, S.-X.-D. Tan, X. Huang, T. Kim, and V. Sukharev, "Analytical modeling and characterization of electromigration effects for multibranch interconnect trees," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 11, pp. 1811–1824, Nov. 2016.

[20] S. R. Nassif, "Power grid analysis benchmarks," in *13th Asia and South Pacific Design Automation Conference (ASPDAC-08)*, Seoul, Korea, Jan. 2008, pp. 376–381.