INCREMENTAL POWER GRID VERIFICATION

by

Abhishek

A thesis submitted in conformity with the requirements for the degree of Master of Applied Sciences
Graduate Department of Electrical & Computer Engineering
University of Toronto

Copyright © 2012 by Abhishek
Abstract

Incremental Power Grid Verification

Abhishek
Master of Applied Sciences
Graduate Department of Electrical & Computer Engineering
University of Toronto
2012

Verification of the on-die power grid is a key step in the design of complex high-performance integrated circuits. For the very large grids in modern designs, incremental verification is highly desirable, because it allows one to skip the verification of a certain section of the grid (internal nodes) and instead, verify only the rest of the grid (external nodes). The focus of this work is to develop efficient techniques for incremental verification in the context of vectorless constraints-based grid verification, under dynamic conditions. The traditional difficulty is that the dynamic case requires iterative analysis of both the internal and the external sections. A solution in the transient case is provided through two key contributions: 1) a bound on the internal nodes' voltages is developed that eliminates the need for iterative analysis, and 2) a multi-port Norton approach is used to construct a reduced macromodel for the internal section.
Acknowledgements

I would like to gratefully acknowledge the enthusiastic supervision of my supervisor Prof. Farid N. Najm for his continuous guidance and inspiration. Without his efforts, insightful suggestions and constant support, the development of this research work would not have been possible. The weekly meetings with him always gave me much needed analysis of the progress made in my research. I particularly remember a meeting in which he took time in the end to explain to me the reasons as to why my progress was getting hampered. I consider myself lucky to have got this opportunity to work under his supervision. Many thanks professor for my overall development as a professional, a researcher and a person.

I would also like to thank Professors Andreas Veneris, Costas Sarris, and Jason Anderson, from the ECE Department of the University of Toronto for reviewing this work. I would also like to acknowledge the financial support provided by the University of Toronto, Natural Sciences and Engineering Research Council (NSERC) of Canada, and Advanced Micro Devices (AMD) Inc.

I am also thankful to Nahi H. Abdul Ghani and Ankit Goyal for their guidance and support during the formative year of my degree program. They were always there to answer my questions on almost any topic related to my research. I would also like to take this opportunity to thank my colleagues for providing a great and pleasant environment. I am especially grateful to Sari Onaissi, Niyati Shah, Sandeep Chatterjee, Mohammad Fawaz, Jason Luu, Jongsok Choi, Andrew Canis, Li Liu, Braiden Brousseau and Bao Le. I wish them best in their endeavors.

I am also lucky to have the support of my best friend Sampada Bagai. Knowing that she will always be by my side no matter what happens, gave me the patience and the strength to successfully complete my Masters.

My biggest gratitude goes to my parents, Dr. Avadhesh Kumar and Mrs. Chhaya Saxena for always encouraging me and wanting only the best for me. I can never forget those pep talks with mom and dad that always helped me whenever I was struggling.
in my research. They never doubted my ability and invested in my future, at times compromising their present. I would also like to thank my younger brother Avijit for always considering me as a friend and giving me much needed breaks, through his writings and musings.

Lastly, I offer my regards to those who supported me in any respect during the completion of this work.
# Contents

1 Introduction  
  1.1 Motivation .............................................. 1  
  1.2 Objective ............................................... 3  
  1.3 Thesis Organization .................................... 4  

2 Background  
  2.1 Introduction ............................................ 5  
  2.2 Power Grid Model ....................................... 5  
    2.2.1 RC Model ........................................... 5  
    2.2.2 RLC Model ......................................... 8  
  2.3 Power Grid Verification ................................. 11  
    2.3.1 Current Constraints ................................. 11  
    2.3.2 RC Grid Verification ............................... 13  
    2.3.3 RLC Grid Verification ............................... 13  
  2.4 Norton’s Theorem ....................................... 16  
  2.5 Model Order Reduction ................................ 20  
    2.5.1 Concept of Moments ................................. 21  
    2.5.2 Explicit Moment Matching ........................... 22  
    2.5.3 Implicit Moment Matching ........................... 26  
    2.5.4 Truncated Balance Realization ..................... 28
List of Tables

3.1 Legend .................................................................................. 35
3.2 Speed and accuracy after using efficient bounds computation .... 44
3.3 Speed and accuracy after applying macromodeling ................. 45
3.4 Runtime breakdown ............................................................... 48

4.1 Speed and accuracy after reducing the dimensions of the feasible space of currents compared to incremental verification approach presented in chapter 3 ........................................................................... 61
4.2 Runtime breakdown ............................................................... 63
4.3 Speed and accuracy after constructing the CPM using Approach I 69
4.4 Comparison between the three different approaches to macromodeling .. 69
# List of Figures

2.1 An *RC* model of Power Grid ........................................... 6
2.2 Power Grid Model showing Internal, External and Port nodes ........ 7
2.3 *RLC* model of Power Grid [1] ........................................... 8
2.4 Combined Package and On-Chip Power Grid ............................. 10
2.5 Sub-grid and Rest of the Grid ............................................. 16
2.6 Simplified Illustration ..................................................... 17
2.7 Configuration after adding voltage sources ............................. 18
2.8 Rest of the Grid disconnected ............................................. 19
2.9 External Sources set to zero ............................................... 19
2.10 Internal Sources moved to ports ......................................... 20
2.11 Final Configuration ....................................................... 21
2.12 T-Model ................................................................. 25
2.13 Floating Capacitance Model ............................................... 25
2.14 Parallel *RC* model ....................................................... 26

3.1 Relative Error Plot (efficient bounds) ................................ 45
3.2 Relative Error Plot (macromodeling) .................................. 46
3.3 Runtime vs Size of Sub-grid .............................................. 46
3.4 Speed-up vs Size of Sub-grid ............................................ 47
3.5 Runtime and Accuracy vs *κ* for G1 .................................... 49
3.6 % Reduction and Runtime vs *τ*<sub>N</sub> for G1 ....................... 49
4.1 A bar graph showing the contributions of major procedures in runtime for both original (chapter 3) and the modified approaches.

4.2 Relative Error Plot for H6 (modified approach).

4.3 Relative Error Plot for A4 ($v_{ub}(\infty)$).

4.4 Relative Error Plot for A4 ($v_{lb}(\infty)$).
Chapter 1

Introduction

1.1 Motivation

A power grid is an electrical network that is composed of multiple layers of metal and that provides supply voltage connections from the power supply pins on the package to the devices on the chip. There are typically two types of voltage variations: $IR$ drop arising due to resistive elements in the grid, and $Ldi/dt$ noise which is a result of the inductive elements in the grid and is proportional to the rate of change of the current [2, 3]. Such variations in the supply voltage can lead to soft error, increased circuit delays and loss of yield. Therefore, voltage integrity analysis is important for chip design. With technology scaling, a trend of decreasing supply voltages, narrower wires, increasing power densities, and tighter noise margins has been observed [4]. A consequence of these trends is that the reliability of integrated circuits has become increasingly susceptible to the voltage level fluctuations.

Most grid verification techniques use some form of circuit simulation to simulate the grid. Simulation-based approaches require complete knowledge of current waveforms drawn by the underlying logic circuitry, which are used to simulate the grid and determine the grid node voltage drops. However, verifying the grid in this way is prohibitively
expensive because the number of current traces required to cover all the possible circuit behaviors is extremely large. Another disadvantage is that a simulation-based flow does not allow for early grid verification (when changes to the grid can most easily be incorporated) because no current traces may be available at that time.

To overcome these issues, a \textit{vectorless} verification approach based on partial current specification in the form of \textit{current constraints} was proposed in [5], and further developed in subsequent work over the last decade. Grid verification is reduced to a problem of finding the worst-case voltage drop over all possible currents that satisfy certain current constraints. In [6], the authors used an \textit{RC} model of the power grid and gave an upper bound on the worst-case voltage drop using an iterative approach. A closed form expression for the upper bound was later proposed in [7], which involved solving a \textit{linear program} (LP) for every grid node. An efficient way to reduce the size of the LPs based on the sparse approximate inverse technique was proposed in [8].

This previous work is useful for verifying the entire power grid but becomes an overkill when verification of only a part of the grid is required, a scenario that is referred to as \textit{incremental verification}. Incremental verification has become desirable because modern grids can be so large that full verification becomes expensive, and a divide-and-conquer approach becomes a necessity. Alternatively, incremental verification is desirable when design changes are made to a local region of a previously-verified grid, and the local impact of these changes needs to be verified. There are also various other cases, such as in case of IP reuse, where a portion of the grid may not need to be verified.

With increasing number of transistors and high operating frequencies, the \textit{Ldi/dt} noise is becoming increasingly significant [2, 9, 10]. Thus, the effect arising from the inductance of metal lines and the inductance of interconnections between the grid and the package should also be included while performing voltage integrity analysis. Traditionally, IC designers focus on the voltage drop for the on-chip power delivery network (PDN) while the package designers are generally concerned with reducing the off-chip network
impedance. The problem with such a design practice is that the package design does not take into account the resonance which can be caused by the resulting \( RLC \) network (consisting of the package inductance and the \( RC \) on-chip interconnects). Therefore, on-chip interconnects must be considered while designing the chip-package power delivery network [11]. However, this is computationally prohibitive in real designs because of the size of the on-chip PDN. A Chip Power Model [11] is a reduced-order model that captures the electrical behavior of the on-chip power distribution network coupled with the parasitics of the chip-package network. The runtime is thus drastically reduced because the entire on-chip network is now converted into a reduced-order model.

### 1.2 Objective

The goal of this research is to develop efficient techniques to allow incremental verification in the vectorless constraints-based power grid verification context. In [12], a technique is given for incremental verification but only for the case of a resistive grid, under the influence of DC currents. In this work, we propose techniques to efficiently perform incremental power grid verification in the transient case. The difficulty with grid verification in the transient case is that it requires iterative analysis of the complete power grid (including the sections that need not be verified). The proposed incremental verification technique should be such that the verification is only performed for the part of the grid that needs to be verified, even under dynamic conditions.

Power grid macromodeling is an important step in incremental grid verification. The main idea is to abstract the behavior of the parts of the grid on to the interface that connects these parts to the portion of the grid, the user is interested in verifying. Two main steps are involved in macromodeling: 1) moving the current sources to the interface, and 2) reducing the resulting passive \( RC \) parts of the grid that need not be verified. In our case, the current source values are not known instead, we have constraints on the current
sources. We need to develop a technique to perform macromodeling in a constraints-based verification model, such that significant runtime savings can be achieved with negligible loss of accuracy.

In [11], the authors use Norton Equivalent theorem to abstract the behavior of on-chip PDN to the ports connecting the chip to the package. As an extension to incremental verification, we develop an approach to perform vectorless verification of the off-chip interconnects by constructing a Chip Power Model for the constraints-based power grid verification framework.

1.3 Thesis Organization

The thesis is organized as follows: Chapter 2 gives the background information on the power grid model and the vectorless approach to power grid verification. An overview of some of the popular Model Order Reduction (MOR) techniques is also provided in this chapter. In chapter 3, the proposed approach for incremental verification is presented. Chapter 4 provides optimizations to the proposed approach and also extends the concept to the verification of package nodes by constructing a Chip Power Model. Finally, chapter 5 concludes and provides directions for further research.
Chapter 2

Background

2.1 Introduction

In this chapter, we review the background material for this work. In section 2.2, we focus on the power grid model. In the next section, we provide a review of the vectorless constraints-based verification approach, which is then followed by a review of the proof of the multi-port Norton’s theorem [13]. In the last section, we review some popular Model Order Reduction techniques.

2.2 Power Grid Model

2.2.1 RC Model

Consider an RC model of the grid as shown in Fig. 2.1. In such a model, each branch is represented by a resistor and there exists a capacitor from every node to ground. Some nodes have ideal current sources (to ground) to represent the currents drawn by the underlying circuitry, and some have ideal voltage sources to represent the connections to the external power supply. Let the power grid consists of $n + p$ nodes, where nodes 1, 2,\ldots, $n$ have no voltage sources attached, and the remaining nodes are nodes where the $p$ voltage sources are attached. Let $i_{s,k}(t)$ be the value of the current source connected
Figure 2.1: An RC model of Power Grid to node $k$. We assume that $\forall k = 1, \ldots, n$, $i_{s,k}(t)$ is well-defined, so that nodes with no current source attached have $i_{s,k}(t) = 0$. Let $i_s(t)$ be the vector of all current sources $i_{s,k}(t)$ and $u(t)$ be the vector of nodal voltages. Applying Modified Nodal Analysis (MNA) to the grid leads to:

$$G u(t) + C \dot{u}(t) = -i_s(t) + G_0 V_{dd}$$

where $G$ is an $n \times n$ conductance matrix as in the traditional MNA formulation; $G_0$ is another $n \times n$ matrix consisting of conductance elements connected to the $V_{dd}$ sources [5]; $C$ is an $n \times n$ diagonal matrix of node capacitances; and $V_{dd}$ is a constant vector each entry of which is equal to the supply voltage value. The matrix $G$ is known to be a diagonally-dominant, symmetric, positive-definite, and an $M$-matrix, so that $G^{-1} \geq 0$.

Let $v(t) = V_{dd} - u(t)$ be the vector of voltage drops. The $RC$ model for the power grid can then be written as [6]:

$$G v(t) + C \dot{v}(t) = i_s(t)$$

Note that this equation can be obtained directly by writing the MNA system for a modified network in which all voltage sources are shorted (set to 0) and all current sources are reversed. The work in chapter 3 is based on the above $RC$ model. In the incremental verification framework presented in chapter 3, the user identifies a part of
Figure 2.2: Power Grid Model showing Internal, External and Port nodes

the grid that does not need to be verified. This part of the grid is referred to as sub-grid as shown in Fig. 2.2. Verification is required only for grid nodes that are outside the sub-grid, referred to as external nodes while the nodes inside the sub-grid are either internal nodes or port nodes. Nodes inside the sub-grid that are connected to the external nodes are called port nodes, while all remaining sub-grid nodes are referred to as internal nodes. Let $n_{\text{ext}}$, $n_{\text{prt}}$, and $n_{\text{int}}$ be the number of external nodes, port nodes, and internal nodes respectively, such that $n_{\text{ext}} + n_{\text{prt}} + n_{\text{int}} = n$. Because external nodes connect only to the port nodes, the grid equation can be written as:

$$
\begin{bmatrix}
G_{11} & G_{12} & 0 \\
G_{12}^T & G_{22} & G_{23} \\
0 & G_{23}^T & G_{33}
\end{bmatrix}
\begin{bmatrix}
v_{\text{ext}}(t) \\
v_{\text{prt}}(t) \\
v_{\text{int}}(t)
\end{bmatrix}
+ 
\begin{bmatrix}
C_{\text{ext}} & 0 & 0 \\
0 & C_{\text{prt}} & 0 \\
0 & 0 & C_{\text{int}}
\end{bmatrix}
\begin{bmatrix}
\dot{v}_{\text{ext}}(t) \\
\dot{v}_{\text{prt}}(t) \\
\dot{v}_{\text{int}}(t)
\end{bmatrix}
=
\begin{bmatrix}
i_{s,\text{ext}}(t) \\
i_{s,\text{prt}}(t) \\
i_{s,\text{int}}(t)
\end{bmatrix}
$$

(2.3)

where $v_{\text{ext}}$ and $i_{s,\text{ext}}$ are sub-vectors corresponding to voltage drops and current sources at external nodes, $v_{\text{prt}}$ and $i_{s,\text{prt}}$ correspond to voltage drops and current sources at port nodes, and $v_{\text{int}}$ and $i_{s,\text{int}}$ correspond to voltage drops and current sources at internal
nodes. The matrices $G$ and $C$ are partitioned into sub-matrices of appropriate dimensions. Using a finite difference approximation as in [6], the system (2.2) can be written as:

$$A v(t) = \frac{C}{\Delta t} v(t - \Delta t) + i_s(t)$$

(2.4)

where $A = (G + \frac{C}{\Delta t})$ is also a symmetric, positive-definite, and an $\mathcal{M}$-matrix, so that $A^{-1} \geq 0$.

### 2.2.2 RLC Model

The RLC model of the power grid is shown in Fig. 2.3. There are two different types of branches in this model [1]: a branch that is represented by a resistor is referred to as an $r$-branch, while a branch that is represented by a resistor in series with an inductor is referred to as an $rl$-branch. The inductance represents the inductive components of either the grid interconnects or the pad structure that connects the grid to external voltage sources (chip-package interconnections). As in [1, 8], we define two types of nodes: nodes which are internal to the $rl$-branch are referred to as branch nodes while all other nodes are referred to as actual nodes. Some of the actual nodes connect to the supply through a branch node while some have ideal current sources (to ground).
Chapter 2. Background

representing the currents drawn by the underlying logic circuitry. There are capacitors from every actual node to ground.

Let the power grid consist of \( n_b + n_l + p \) nodes where nodes 1, 2, \ldots, \( n_b \) are branch nodes, nodes \( (n_b + 1), (n_b + 2), \ldots, (n_b + n_l) \) are actual nodes with no voltage sources attached and the remaining nodes are the actual nodes where \( p \) voltage sources are attached. Let \( i_{s,k}(t) \) be the current source connected to node \( k \). We assume that \( i_{s,k}(t) \) is defined for all nodes \( k = 1, 2, \ldots, n \), where \( n = n_b + n_l \), so that all nodes with no current sources attached have \( i_{s,k}(t) = 0 \). Let \( i_s(t) \) be the vector of all \( i_{s,k}(t) \) sources and \( u(t) \) be the vector of voltage signals \( u_k(t) \) at every node. Also, let \( i(t) \) be the vector of the inductive branch currents \( i_l(t) \). The time-domain equation for the grid is given by:

\[
G u(t) + C \dot{u}(t) + M i(t) = -i_s(t) + G_0 V_{dd}
\]  

(2.5)

where \( G \) is an \( n \times n \) conductance matrix that is known to be a diagonally-dominant, symmetric, positive definite, \( M \)-matrix; \( G_0 \) is an \( n \times n \) matrix of the conductance elements connected to the \( V_{dd} \) sources [5]; \( C \) is an \( n \times n \) capacitance matrix with the entries corresponding to the branch nodes being equal to zero; and \( M \) is an \( n \times n_b \) incidence matrix whose elements are either \( \pm 1 \) or 0, as in [1]. The term \( \pm 1 \) appears in the location \( m_{ij} \) of the matrix \( M \) when the node \( i \) is connected to the \( j^{th} \) inductor, else a 0 occurs. If the current-direction assignment is away from the node the sign is positive, else it is negative.

Let \( v(t) = V_{dd} - u(t) \), then the \( RLC \) model of the power grid can be written as [1]:

\[
G v(t) + C \dot{v}(t) - M i(t) = i_s(t)
\]  

(2.6)

The inductive branch currents can also be expressed in terms of voltage drops \( v(t) \) as:

\[
M^T v(t) + L \dot{i}(t) = 0
\]  

(2.7)

where \( L \) is an \( n_b \times n_b \) diagonal matrix of inductance values. Equations (2.6) and (2.7) represent the behavior of the \( RLC \) model of the power grid. In the construction of the Chip Power Model discussed in chapter 4, we consider the model as shown in Fig. 2.4.
Chapter 2. Background

We consider an RC model for the on-chip interconnects while the package to supply interconnections are modeled using the RLC model described above. While designing the combined package and on-chip power grids, the on-chip inductance can be safely ignored because the effect of the on-chip inductance is small as compared to the resistance of the lines [2, 14]. The current sources model the chip activity and therefore, are present only on the on-chip interconnects. Let $n_{ext} = n_b + n_{pkg}$ be the number of external nodes that include $n_b$ branch nodes and $n_{pkg}$ actual nodes, $n_{prt}$ and $n_{int}$ be the number of port nodes and internal nodes (on-chip) respectively. The system can thus be partitioned as:

$$
\begin{bmatrix}
G_{11} & G_{12} & 0 \\
G_{12}^T & G_{22} & G_{23} \\
0 & G_{23}^T & G_{33}
\end{bmatrix}
\begin{bmatrix}
v_{ext}(t) \\
v_{prt}(t) \\
v_{int}(t)
\end{bmatrix}
+ 
\begin{bmatrix}
C_{ext} & 0 & 0 \\
0 & C_{prt} & 0 \\
0 & 0 & C_{int}
\end{bmatrix}
\begin{bmatrix}
v_{ext}(t) \\
v_{prt}(t) \\
v_{int}(t)
\end{bmatrix}
- 
\begin{bmatrix}
M_{ext} \\
0 \\
0
\end{bmatrix}
i(t)
= 
\begin{bmatrix}
0 \\
0 \\
i_s,int(t)
\end{bmatrix}
$$

(2.8)

$$
\begin{bmatrix}
M_{ext} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
v_{ext}(t) \\
v_{prt}(t) \\
v_{int}(t)
\end{bmatrix}
+ L\dot{i}(t)
= 0
$$

(2.9)

where $v_{ext}$ is the sub-vector corresponding to voltage drops at the external nodes, $v_{prt}$ correspond to voltage drops at port nodes, and $v_{int}$ and $i_s,int$ correspond to voltage drops

Figure 2.4: Combined Package and On-Chip Power Grid
and current sources at internal nodes. The matrices $G$ and $C$ are partitioned into sub-matrices of appropriate dimensions. Since only the inductances present in the package interconnections are considered, the matrix $M$ is composed of a matrix $M_{ext}$ of size $n_{ext} \times n_b$ and all other entries are equal to zero. As in [1], a discrete time version of the system can be given as:

$$
(G + \frac{C}{\Delta t}) v(t) - Mi(t) = i_s(t) + \frac{C}{\Delta t} v(t - \Delta t)
$$

(2.10)

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

(2.11)

where $(G + \frac{C}{\Delta t})$ is also a symmetric, positive definite, and an $M$-matrix.

### 2.3 Power Grid Verification

In this section, we review the vectorless verification approach based on partial current specifications in the form of current constraints. The grid verification problem is reduced to a problem of finding the worst-case voltage fluctuation over all possible currents that satisfy the current constraints. The next sub-section gives an overview of the current constraints.

#### 2.3.1 Current Constraints

The verification framework used in this work allows for early verification of power grids, when the details of the underlying circuitry may not be known. Current constraints [5] provide a way to capture the uncertainty about circuit behavior and that one is uncertain about the circuit itself early in the design flow. One can specify these constraints from the knowledge of the design specs (area and power budget), and also from engineering judgement (power needs in previous technology, the effect of scaling on those needs etc.).

As in previous work, we use two types of constraints: *local constraints* and *global constraints*. Local constraints are upper bounds on the individual current sources, where
one specifies that the current \( i_{s,k}(t) \) never exceeds a certain fixed level \( i_{L,k} \). We assume that every current source tied to the grid has an upper bound associated with it, so that if a node does not have a current source attached, the upper bound for that current is 0. We can express these constraints as:

\[
0 \leq i_s(t) \leq i_L, \quad \forall t \geq 0
\] (2.12)

If only local constraints are provided, the problem is much simplified but the results become overly pessimistic, because it is never the case that all the chip components are simultaneously drawing their maximum currents. Global constraints are upper bounds on the sums of currents for groups of current sources. They represent the peak total power dissipation of a group of circuit blocks. Assuming that we have a total of \( m \) global constraints, then we can express them in matrix form as:

\[
0 \leq S i_s(t) \leq i_G, \quad \forall t \geq 0
\] (2.13)

where \( S \) is an \( m \times n \) matrix that contains only 0s and 1s, that indicate which current sources are present in each global constraint. Together, the local and the global constraints define a feasible space of currents, denoted by \( F \), such that \( i_s(t) \) lies inside the feasible space \( (i_s(t) \in F) \) if and only if it satisfies (2.12) and (2.13), \( \forall t \geq 0 \). Combining (2.12) and (2.13), we can write:

\[
0 \leq Ui_s(t) \leq u
\] (2.14)

where \( U \) is an \( (m + n) \times n \) matrix consisting of \( I_n \) (the \( n \times n \) identity matrix) and \( S \), and \( u \) is the upper bound vector. Therefore, the feasible space \( F \) can be defined as:

\[
F = \{i_s(t) : 0 \leq Ui_s(t) \leq u, \forall t \geq 0\}
\] (2.15)

Given a power grid, we are interested in finding the worst-case voltage variations at all the nodes, under all the possible (transient) current waveforms \( i_s(t) \) that satisfy the current constraints.
2.3.2 \textit{RC} Grid Verification

For an \textit{RC} grid, the authors in [7] provide an upper bound \(v_{ub}\) on the worst-case voltage drop vector, so that \(v(t) \leq v_{ub}, \forall t\), and this bound is given by:

\[
v_{ub} = \left( I + \frac{G^{-1}C}{\Delta t} \right) v_a
\]  

(2.16)

where \(v_a\) is the worst-case voltage drop vector at \(t = \Delta t\) in the special case when \(i_s(t) = 0, \forall t \leq 0\), and \(I\) is the identity matrix. Since \(v(0) = 0\) in this special case, it follows from (2.4) that:

\[
v(\Delta t) = A^{-1} \left( \frac{C}{\Delta t} v(0) + i_s(\Delta t) \right) = A^{-1}i_s(\Delta t)
\]  

(2.17)

and \(v_a\) can be expressed as:

\[
v_a = \max_{\forall i_s(\Delta t) \in \mathcal{F}} A^{-1}i_s(\Delta t)
\]  

(2.18)

where “emax” is an operator that denotes element-wise maximization of its vector argument, under the given constraints. In other words, \(v_a\) is the result of the for-loop: for \((k = 1, \ldots, n)\) \{maximize the \(k^{th}\) element of the vector \(A^{-1}i_s(\Delta t)\), over all \(i_s(\Delta t) \in \mathcal{F}\}\}. Maximizing each element becomes a linear program (LP). Note that, because the definition of the local and global constraints does not depend on time, then \(v_a\) is independent of \(t\) and we can drop the \(\Delta t\) argument, and write:

\[
v_a = \max_{\forall i_s \in \mathcal{F}} A^{-1}i_s
\]  

(2.19)

where, for the purpose of this optimization, \(i_s\) can be viewed as simply a “dummy variable”, a \(n \times 1\) real vector with units of current. Thus, the problem of finding the worst-case voltage drop is reduced to performing element-wise maximization of \(A^{-1}i_s\), over all \(i_s \in \mathcal{F}\), to find \(v_a\), followed by a standard linear system solve, to find \(v_{ub}\).

2.3.3 \textit{RLC} Grid Verification

In the \textit{RLC} verification framework, we are interested in the maximum and minimum worst-case voltage drops, over all possible currents in \(\mathcal{F}\). An efficient method to compute
the upper bound on the maximum worst-case voltage drop and lower bound on the minimum worst-case voltage drop was proposed in [1]. Let:

\[
A = \begin{bmatrix}
G + \frac{C}{\Delta t} & -M \\
M^T & \frac{L}{\Delta t}
\end{bmatrix}, \quad B = \begin{bmatrix}
\frac{C}{\Delta t} & 0 \\
0 & \frac{L}{\Delta t}
\end{bmatrix}, \quad x(t) = \begin{bmatrix}
v(t) \\
i(t)
\end{bmatrix}, \quad b(t) = \begin{bmatrix}
i_s(t) \\
0
\end{bmatrix}
\]

(2.20)

Combining (2.10) and (2.11) using (2.20), we get a simplified discrete-time expression for the RLC model:

\[
x(t) = A^{-1}Bx(t - \Delta t) + A^{-1}b(t)
\]

(2.21)

Consider the special case where the grid had no stimulus for all \( t \leq 0 \), so that \( x(0) = 0 \). At time \( t = \Delta t \), we get:

\[
x(\Delta t) = A^{-1}b(\Delta t)
\]

(2.22)

Similarly, at \( 2\Delta t \):

\[
x(2\Delta t) = A^{-1}BA^{-1}b(\Delta t) + A^{-1}b(2\Delta t)
\]

(2.23)

Thus, at any future time \( p\Delta t \), we have:

\[
x(p\Delta t) = \sum_{k=0}^{p-1} (A^{-1}B)^k A^{-1}b((p - k)\Delta t)
\]

(2.24)

The general solution to the exact voltage drop maximization and minimization problem is provided in [1] as:

\[
x_{opt}(\tau) = \lim_{p \to \infty} \text{eopt}_{b(t) \in F} \left( \sum_{k=0}^{p-1} (A^{-1}B)^k A^{-1}b((p - k)\Delta t) \right)
\]

(2.25)

where the “eopt” operator denotes the element-wise maximization and minimization of its vector argument, under the given constraints. Since the constraints are DC and do not depend on time, decoupling the components of (2.25) leads to:

\[
x_{opt}(\tau) = \lim_{p \to \infty} \sum_{k=0}^{p-1} \text{eopt}_{b(t) \in F} \left[ (A^{-1}B)^k A^{-1}b((p - k)\Delta t) \right]
\]

(2.26)
Equation (2.26) is of theoretical interest only as it involves a large number of time steps. In [1], the authors proposed bounds on the maximum and minimum worst-case voltage drops by first transforming the RLC grid into a reduced circuit by eliminating the inductive branch currents, without any approximation. The reduced circuit equation can be written as:

\[
\left( G + \frac{C}{\Delta t} + M \frac{L}{\Delta t}^{-1} M^T \right) v(t) = \left( \frac{C}{\Delta t} + M\hat{G} \right) v(t - \Delta t) + i_s(t)
\] (2.27)

where \(\hat{G}\) is an \(n_b \times n\) matrix whose \(k^{th}\) row is that of either \(G\) or \(-G\) depending on the current assignment through the inductive \(rl\)-branch. The same equation can be written in the compact form as:

\[
v(t) = D^{-1}E v(t - \Delta t) + D^{-1}i_s(t)
\] (2.28)

where \(D = \left( G + \frac{C}{\Delta t} + M \frac{L}{\Delta t}^{-1} M^T \right)\) is a sparse symmetric, positive definite, and a banded \(M\)-matrix, and \(E = \left( \frac{C}{\Delta t} + M\hat{G} \right)\). To compute the bounds at infinity, upper and lower bounds on voltage drops for \(r\) time steps ahead in time are computed and are given by:

\[
w_r = \sum_{q=0}^{r-1} eopt \left[ (D^{-1}E)^q D^{-1}i_s \right]
\] (2.29)

\[
= \sum_{q=0}^{r-2} eopt \left[ (D^{-1}E)^q D^{-1}i_s \right] + eopt \left[ (D^{-1}E)^{r-1} D^{-1}i_s \right]
\] (2.30)

\[
= w_{r-1} + eopt \left[ (D^{-1}E)^{r-1} D^{-1}i_s \right]
\] (2.31)

The choice of \(r\) is made in such a way that \(\|N\|_\infty < 1\) and \(\|N\|_1 < 1\), where \(N = (D^{-1}E)^r\). The upper and lower bounds at infinity are now given by:

\[
\begin{bmatrix}
v_{ub}(\infty) \\
v_{lb}(\infty)
\end{bmatrix} = (I - R)^{-1} w_r
\] (2.32)

where \(R\) is a \(2n \times 2n\) matrix defined as:

\[
R = \begin{bmatrix}
N - S & S \\
S & N - S
\end{bmatrix}
\] (2.33)
where $S = \frac{1}{2}(N - Q)$, with $Q$ being the matrix of the element-wise absolute values of the entries in $N$.

2.4 Norton’s Theorem

Norton’s theorem is a fundamental theorem in circuit theory that converts any linear two-terminal network into a simple parallel circuit consisting of an equivalent current source, and an equivalent internal impedance. The equivalent current source value is the current that would flow through a short circuit between the two terminals [15].

In this section, we review a proof for the multi-port Norton’s theorem [13] which can be used to convert a linear $n$-port network into an equivalent circuit with all the current sources internal to the network replaced by Norton current sources at the port nodes. Consider a sub-grid section which is linear and may contain sources, connected to the rest of the grid as shown in Fig. 2.5. For our purpose, the sources are independent current or voltage sources that are always connected between a grid node and ground. But, in general, any configuration of sources is allowed, including controlled sources. It is assumed that no mutual inductance exists between the sub-grid and the rest of the
grid. We can simplify the illustration as shown in Fig. 2.6, where the port nodes are numbered from 1 to \( p \). With reference to Fig. 2.6, let the voltage waveforms at the port nodes be \( v_1(t) \), \( v_2(t) \), \ldots, \( v_p(t) \), and let the port current waveforms be \( i_1(t) \), \( i_2(t) \), \ldots, \( i_p(t) \), as shown. If we connect a new set of independent voltage sources carrying the same port voltage waveforms \( v_1(t) \), \( v_2(t) \), \ldots, \( v_p(t) \), as shown in Fig. 2.7, then clearly these sources will carry no current and the circuit behavior everywhere is not altered. Furthermore, if we now disconnect the rest of the grid, as in Fig. 2.8, the port currents \( i_1(t) \), \( i_2(t) \), \ldots, \( i_p(t) \) and the sub-grid internal signals remain unaffected, because the sub-grid is subjected to the same port voltage waveforms as in the original circuit.

Now, considering Fig. 2.8, we are interested in finding the port currents with the aid of the superposition theorem. The port currents can be found as the sum of two versions of these currents resulting from the following two modified circuits: 1) the circuit of Fig. 2.8, but with the external (voltage) sources \( v_1(t) \), \( v_2(t) \), \ldots, \( v_p(t) \) all set to zero, as shown in Fig. 2.9 and 2) the circuit of Fig. 2.8, with all the internal independent sources of the sub-grid set to zero (voltage sources replaced by short circuits and current sources replaced by open circuits). With reference to Fig. 2.9, we denote the (short-circuit) port current waveforms as \( i_1^{(s)}(t) \), \( i_2^{(s)}(t) \), \ldots, \( i_p^{(s)}(t) \) and note that these currents depend only
on the sub-grid internals and are independent of the rest of the grid. Specifically, they do not depend on the original port voltages $v_1(t), v_2(t), \ldots, v_p(t)$.

It is clear, therefore, that the original port currents $i_1(t), i_2(t), \ldots, i_p(t)$ can be found from the circuit of Fig. 2.10, where the short-circuit currents $i_1^{(s)}(t), i_2^{(s)}(t), \ldots, i_p^{(s)}(t)$ have been introduced as additional independent sources. In this circuit, the sub-grid passive elements remain in place but its internal sources have all been removed – effectively, they have been “moved” to the ports. The grid internal voltages and currents may have been altered as a result of this construction, but its port voltages remain the same and its original port currents are still available as $i_1(t), i_2(t), \ldots, i_p(t)$ in Fig. 2.10.

Finally, we can “put back” the rest of the grid and remove the voltage sources $v_1(t), v_2(t), \ldots, v_p(t)$, reversing the earlier construction, leading to the circuit shown in Fig. 2.11. As a result of this sequence of steps, all independent sources internal to the original sub-grid have been moved to its ports, and these new port current sources depend only on the sub-grid, so that the sub-grid port signals remain unaltered. In summary, the values of current sources at port nodes that replace the internal sources can be found by the following process:

1. The sub-grid is disconnected from the rest of the grid.
Chapter 2. Background

Sub-grid Linear, with sources

Figure 2.8: Rest of the Grid disconnected

Sub-grid Linear, with sources

Figure 2.9: External Sources set to zero
2. Each port node is connected to ground via a short circuit.

3. The currents flowing through these short circuit connections (due to the applied internal current sources) are evaluated.

2.5 Model Order Reduction

Compact modeling of the power grid is important because it allows for efficient analysis of large scale designs. Model order reduction (MOR) was developed in the area of systems and control theory, which studies the properties of dynamical systems with the goal of reducing their complexity, while preserving their input-output behavior as much as possible [16]. MOR aims to capture the essential features of a structure, thereby simplifying the model in order to perform analysis or simulation efficiently.

The fundamental methods in MOR were published in the early eighties and nineties of the last century and a lot of work has been done in this area. We provide an overview of some of the popular MOR techniques. The techniques for MOR can be broadly classified into the following:

1. Explicit Moment Matching
2. Implicit Moment Matching

3. Truncated Balance Realization

4. Local Node Elimination

The first two categories are broadly based on the concept of moments of a linear network. In truncated balance realization methods, the “weak” uncontrollable and unobservable state variables are truncated to achieve reduction [17]. These methods produce nearly optimal models but are more computationally expensive than moment-based methods. Local node elimination methods like TICER [18] allow reduction to be applied in a local manner by eliminating circuit nodes that have negligible effect on the input-output behavior of the system.

2.5.1 Concept of Moments

The transfer function of a linear network $H(s)$ is given by:

$$H(s) = \frac{Y(s)}{X(s)}$$  \hspace{1cm} (2.34)

where $Y(s)$ and $X(s)$ are the output and input functions, respectively. If the input is an impulse function $\delta(t)$, then $H(s)$ is also the transient solution of the output impulse
response of the system which can be expanded around \( s = 0 \) by Taylor series expansion as:

\[
H(s) = \sum_{k=0}^{\infty} M_k s^k
\]  
(2.35)

where

\[
M_k = \frac{1}{k!} \times \left. \frac{d^k H(s)}{ds^k} \right|_{s=0}
\]  
(2.36)

is called the \( k^{th} \) order moment. We can represent the MNA formulation in state space as:

\[
G x(t) + C \dot{x}(t) = B u(t)
\]
\[
y(t) = L^T x(t)
\]  
(2.37)

where \( G \) and \( C \) are the \( n \times n \) conductive and storage element matrices, \( B \) is an \( n \times p \) input position matrix, \( L \) is an \( n \times q \) output position matrix, \( x \) represents the circuit variables, \( y \) is the output vector, and \( u \) is the excitation vector. Using the Laplace theorem with the initial condition set to 0, and then expanding at \( s = 0 \) gives:

\[
(G + sC)(x_0 + x_1 s + x_2 s^2 + ...) = B
\]  
(2.38)

From (2.38), the moments for state variable \( x \) are given by a recursive formula as:

\[
x_0 = G^{-1}B; \quad x_1 = -G^{-1}Cx_0
\]

\[
\Rightarrow x_i = -G^{-1}Cx_{i-1}
\]  
(2.39)

The output moment \( M_i = L^T x_i \) is therefore given by:

\[
M_i = L^T(-G^{-1}C)^iG^{-1}B
\]  
(2.40)

### 2.5.2 Explicit Moment Matching

Explicit moment matching methods are based on the direct or explicit matching of the moments in order to reduce the order of the system. Let us assume that the transfer
function of an original model is given by:

\[ H(s) = M_0 + M_1 s + M_2 s^2 + \ldots \]  

(2.41)

We try to build a reduced model that matches the first \( q \) moments and whose transfer function will be given by:

\[ H_q(s) = M_0 + M_1 s + M_2 s^2 + \ldots + M_q s^q \]  

(2.42)

Asymptotic Waveform Evaluation

Asymptotic Waveform Evaluation (AWE) [19] is an efficient frequency-domain analysis approach which combines the moment computation (described in the previous subsection) with Pade Approximation techniques, to match the moments. The main idea of Pade Approximation is to approximate the transfer function \( H(s) \) by an order-limited rational function \( H_q(s) \) of order \( q \). For a single input and single output (SISO) system, \( H_q(s) \) is given as:

\[ H_q(s) = \frac{a_0 + a_1 s + a_2 s^2 + \ldots + a_{q-1} s^{q-1}}{1 + b_1 s + b_2 s^2 + \ldots + b_q s^q} \]  

(2.43)

Next, we match the first \( 2q \) moments of \( H(s) \) and \( H_q(s) \) such that:

\[ H_q(s) = M_0 + M_1 s + M_2 s^2 + \ldots + M_{2q} s^{2q} \]  

(2.44)

Equations (2.43) and (2.44) can be used to solve for the coefficients of the rational function by equating the coefficients of \( s \). The poles, zeros, and residues of the reduced-order model can now be found to perform time-domain or frequency-domain analysis. AWE tends to generate unstable positive poles when higher order moments are computed. The accuracy also suffers with higher order poles. Techniques like Multinode Moment Matching [20] can be used to improve the stability while estimating the higher order poles. In such a method, the poles are estimated from different nodes or different stimuli rather than using a single node and a single stimulus as in AWE.
RC MOR

In RC MOR [21], a large $RC$ interconnect network is partitioned into small subnetworks, that are then approximated with lower order equivalent $RC$ circuits. The partitioning is done using the S-parameter matrix. After performing the partitioning, the admittance matrix looking into the ports for a partition is approximated using the first two order moments as:

$$Y(s) \approx M_0 + M_1 s$$

(2.45)

The admittance to ground of the $i^{th}$ port is given as the sum of the $i^{th}$ row (or column) of $Y(s)$ while the admittance connecting the $i^{th}$ port and the $j^{th}$ port is the negative of the $(i, j)^{th}$ element ($y_{ij}$) in $Y(s)$. The circuit models shown in Fig. 2.12 and 2.13 are used to synthesize the circuit between the pairs of ports by matching the moments $y_{ij}$. The port-to-ground elements are synthesized using a parallel $RC$ model as shown in Fig. 2.14. The T-model is used to synthesize all the off-diagonal elements of $Y(s)$ with $m^{ij}_1 \geq 0$, where $m^{ij}_k$ is the $(i, j)^{th}$ element of the $k^{th}$ order moment matrix. The elements of the T-model are given by:

$$R_{ij1} = \frac{-\sqrt{m^{ij}_1}}{m^{ij}_0 \left(\sqrt{m^{ii}_1} + \sqrt{m^{jj}_1}\right)}$$

$$R_{ij2} = \frac{-\sqrt{m^{ii}_1}}{m^{ij}_0 \left(\sqrt{m^{ii}_1} + \sqrt{m^{jj}_1}\right)}$$

$$C_{ij} = \frac{\left(\sqrt{m^{ii}_1} + \sqrt{m^{jj}_1}\right)^2 m^{ij}_1}{\sqrt{m^{ii}_1 m^{jj}_1}}$$

(2.46)

If the circuit contains floating capacitors, $m^{ij}_1$ can become negative and in that case, we use the floating capacitance model whose elements are given by:

$$R_{ij} = \frac{-1}{m^{ij}_0} \quad C_{ij} = -m^{ij}_1$$

(2.47)
As for the port-to-ground connections, the authors [21] provide:

\[
R_{ii} = \left( m_{i0}^i + \sum_{j=1, i\neq j}^{n_{port}} m_{ij}^i \right)^{-1}
\]

\[
C_{ii} = m_{i1}^i - \sum_{j=1, i\neq j}^{n_{port}} \frac{C_{ij}R_{ij2}^2}{(R_{ij1} + R_{ij2})^2}
\] (2.48)

RC MOR is used as a first step to reduce the sub-grid in the incremental verification approaches presented in chapter 3 and chapter 4. The modeling time required for this method is linear with the number of ports of the original interconnect network. RC MOR preserves the block and sparse structure in the resultant reduced network. It has improved stability over AWE because only lower order moments are required. Another important advantage of this method is that it is completely realizable, as the reduced model is also an \( RC \) circuit. AWE-based methods provide better accuracy because of a high approximation order that results in an increase in the modeling and simulation time. Partitioning allows the use of lower order models with accuracy comparable to reducing a large \( RC \) interconnect using a higher approximation order.

In [22], the authors use hMETIS [23] to perform partitioning. The approach was extended to include \( RLC \) circuits in PartMOR [24]. In PartMOR, moments up to the third order are used to macromodel the \( RLC \) circuit elements. SparseRC [25] employs a fill-in-reducing based ordering scheme to improve sparsity during model reduction. The basic model reduction engine used in SparseRC is similar to PartMOR. The drawback of SparseRC lies in the fact that it can only be used for \( RC \) circuits. In the next sub-section, we cover implicit moment matching-based methods.
2.5.3 Implicit Moment Matching

The main idea of implicit moment matching techniques, also referred to as projection-based MOR, is to project the moment space onto an orthonormal subspace called the Krylov subspace. As always, we are interested in reducing the number of state variables in \( x \) while approximately maintaining the input-output behavior of the system. This can be achieved by finding a transformation matrix \( V \) such that \( x = V \hat{x} \), where \( \hat{x} \) is the state variable vector for the reduced system. The MNA formulation given in (2.37) can also be written as:

\[
\dot{x}(t) = Ax(t) + Ru(t) \\
y(t) = L^T x(t)
\] (2.49)

where \( A = -G^{-1}C \) and \( R = G^{-1}B \). The reduced system can be written as:

\[
\dot{\hat{x}}(t) = \hat{A}\hat{x}(t) + \hat{R}u(t) \\
y(t) = \hat{L}^T \hat{x}(t)
\] (2.50)

where \( \hat{A} = V^T AV \) is \( r \times r \), \( \hat{R} = V^T R \) is \( r \times p \) and \( \hat{L} = V^T L \). The transfer function \( H(s) \) of the original system can be written as:

\[
H(s) = L^T (I - sA)^{-1} R
\] (2.51)
The moment space of the original system for the first \( r \) moments is given by:

\[
\{M_0, M_1, M_2, \ldots, M_{r-1}\} = \{L^T R, L^T AR, L^T A^2 R, \ldots, L^T A^{r-1} R\}
\]

\( = L^T \{R, AR, A^2 R, \ldots, A^{r-1} R\} \) \quad (2.52)

Therefore, the Krylov subspace of order \( r \) for the system described above is:

\[ K_r(R, A) \equiv \text{span}(R, AR, A^2 R, \ldots, A^{N-1} R) \] \quad (2.53)

If the columns \((v_0, v_1, \ldots, v_r)\) of the projection matrix \( V \) are chosen such that:

\[ \text{span}(v_0, v_1, \ldots, v_r) = \text{span}(R, AR, A^2 R, \ldots, A^{N-1} R) \] \quad (2.54)

then the reduced-order model will match the first \( r \) moments of the original system.

**Passive Reduced-Order Interconnect Macromodeling Algorithm**

In PRIMA [26], the projection matrix is constructed using the block-Arnoldi algorithm [17]. It can guarantee the passivity of the reduced system if the original system is in MNA form with \( L = B \) where \( B \) is \( n \times n_{prt} \).

PRIMA does not preserve certain important circuit properties like reciprocity. The resulting reduced matrices are also larger than those obtained from direct pole matching. The size of the matrices (states) depends upon the number of moments to be matched and also on the number of ports. For every block moment order increase, PRIMA will generate \( n_{prt} \) new poles. To overcome this issue, TERMMERG [27] uses a reduced-rank approximation technique to group the terminals with similar timing or delay behavior into one terminal prior to performing reduction using PRIMA. In [18], it has been argued that PRIMA takes in an electrical circuit and outputs a system of poles and zeros which needs to be recast into an electrical circuit. Therefore, one must have a knowledge of the nexus between the matrix formulations and electric circuits to use this approach.
Structure-Preserving Reduced Order Interconnect Modeling

SPRIM [28] is a projection-based method for RLC circuits that can preserve the structure of the original model. It uses a $2 \times 2$ block structure of the MNA matrix:

\[
G = \begin{bmatrix} G' & A^T \\ -A & 0 \end{bmatrix}, \quad C = \begin{bmatrix} C & 0 \\ 0 & L \end{bmatrix}
\]  

(2.55)

where $A$ is a matrix that indicates the branch current flow at the inductor. Similarly, a structured projection matrix $\hat{V}$ is obtained by partitioning $V$ as:

\[
V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} \rightarrow \hat{V} = \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}
\]  

(2.56)

PRIMA can now be used to reduce the system. This reduced matrix will have twice the number of poles but there would be an improvement in accuracy over PRIMA. The system can be further reduced by using Schur's decomposition [29] of the branch-current variables. SPRIM does not completely model the sub-block structure and locality of an interconnect model. Interconnect models or P/G grids are highly local and sparse. In BSMOR [30], the MNA matrices are divided into many blocks, followed by the reduction of every block using PRIMA. BSMOR results in better accuracy and also preserves the block structure of the system.

2.5.4 Truncated Balance Realization

In truncated balance realization methods, the system is mapped onto a basis where the states that are difficult to reach are truncated. These techniques are based on the computation of controllability and observability Gramians [31].
Standard TBR

For the system defined in (2.49), the controllable Gramian $X$ and the observable Gramian $Y$ are unique symmetric, positive definite solutions to the Lyapunov equations [17]:

$$AX + XA^T + RR^T = 0$$
$$A^TY + YA + LL^T = 0$$ (2.57)

Applying a similarity transformation to diagonalize the product $XY$ such that:

$$T^{-1}XYT = \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$$ (2.58)

where the singular values of the system ($\sigma_k$) are arranged in a descending order. The matrices are then partitioned as:

$$\begin{bmatrix} W_1^T \\ W_2^T \end{bmatrix} XY \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix}$$ (2.59)

where $\Sigma_1 = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_r^2)$ is made from the first $r$ largest eigenvalues of $XY$, and $W_1$ and $V_1$ are the corresponding eigenvectors. Thus, the reduced model can now be obtained by using $W_1^T$ and $V_1$ instead of $V$ as in the case of congruence transformations. Therefore, the reduced system equation (2.50) will consist of the following matrices:

$$\hat{A} = W_1^T AV_1 \quad \hat{R} = V_1^T R \quad \hat{L} = V_1^T L$$ (2.60)

Although TBR ensures excellent accuracy, it is computationally very expensive because of the cubic complexity of solving the Lyapunov equations. Poor Man’s TBR [32] is a truncation-based method that uses approximate Gramians to perform reduction. This results in reducing the time complexity of the reduction method. Adaptive sampling-based methods like WBMOR [33] can be used to provide near accurate reduced models over wide frequency bands. These methods are based on the TBR method and compute the approximate Gramians using a Monte-Carlo sampling approach.
2.5.5 Local Node Elimination

The main idea of techniques involving local node elimination is to reduce the number of nodes in the circuit and approximate the newly added elements in the circuit matrix with reduced rational forms [17]. The main advantage of these methods is the ability to perform the reduction in a local manner and the fact that no overall solution of the whole circuit is required, which makes these methods applicable to large interconnection networks.

Time-Constant Equilibrium Reduction

TICER [18] is an elimination-based approach for reducing the interconnect model. It converts a given $RC$ network into a smaller $RC$ network by eliminating nodes that have fewer neighbors and a small value of time constant. These nodes are called quick nodes. The time constant of a node is given by:

$$\tau_N = \frac{C_N}{G_N}$$  \hspace{1cm} (2.61)

where $C_N$ is the sum of all capacitive elements incident on node $N$ and $G_N$ is the sum of all conductive elements. If $|s\tau_N| \ll 1$ where $s$ is the frequency under which the system needs to be operated, the node is called a quick node and can be eliminated from the circuit. In terms of circuit elements, the reduction can be described as the following two-step process:

1. Remove all the resistors and capacitors connecting other nodes to node $N$.

2. Insert new resistors and capacitors between the former neighbors using the following two rules:

   - If nodes $i$ and $j$ were connected to node $N$ by conductances $g_{iN}$ and $g_{jN}$, insert a conductance $g_{iN}g_{jN}/G_N$ between $i$ and $j$. 

Chapter 2. Background

- If node $i$ had a capacitor $c_{iN}$ to $N$ and node $j$ a conductance $g_{jN}$ to $N$, then insert a capacitor $c_{iN}g_{jN}/G_N$ between $i$ and $j$.

In the incremental verification approach, TICER is used to reduce the sub-grid that has already been reduced once using RC MOR. It does not exactly replicate the first order moments of the admittance matrix but it preserves the Elmore delays through $RC$ trees in most cases. It cannot be applied to $RLC$ circuits. As a node is eliminated, it leads to addition of new elements to the neighbors. This addition reduces the sparsity of the matrices. Therefore, care must be taken while selecting the quick nodes. In [34], the authors extend local node elimination to include $RLC$ circuits by matching the DC characteristics and the first two moments at all the nodes. The method is applicable to a simplified $RLC$ model in which the $RLC$ network connecting a pair of nodes has branches, in which one or more of the elements ($R$, $L$, or $C$) is zero. In this scheme, two time constants are associated with each node and the reduction is done based on the dominant time constant. The method introduces a large error if the time constants are comparable to each other.

2.5.6 Divide and Conquer

We have covered some of the most popular techniques to compactly model the power distribution network. There is yet another class of reduction techniques that applies a divide-and-conquer strategy to efficiently reduce the given system. In [35], the power grid is partitioned into global and local grids. The behavior of a local grid is abstracted at the interface (port nodes) with the global grid and, then the global grid is simulated using the macromodels of the local grids. The work presented in chapter 3 adapts this hierarchical approach to the constraints-based verification framework. Block-based partitioning for parallel power grid analysis is presented in [36]. In this work, the power grid is partitioned based on the functional blocks and that most of the block current is drawn from
the C4 bump nearest to the block. The partitioner can then be combined with existing MOR techniques to produce reduced-order models for each partition. The simulation results from individual partitions are then combined to perform full-chip analysis. In [37], a hierarchical matrix representation of the power grid was proposed. The hierarchical matrices are derived from the partitioning of the power grid. This type of representation allows for reduced storage space and improved simulation times. Approaches based on this strategy have fast simulation times because the entire problem is broken down into small sub-problems that can be solved efficiently. In the next chapter, we will be focusing on an incremental verification approach for vectorless constraints-based verification framework.
Chapter 3

Incremental Power Grid Verification

3.1 Introduction

In this chapter, we present an incremental power grid verification approach that extends previous work [12] to the case of transient currents. In the incremental verification framework defined in section 2.2.1, we are interested in finding the worst-case voltage drops at those nodes in the RC model that have been identified as external by the user. A solution for the case when all the currents are DC was given in [12]. Strictly speaking, the extension in the general dynamic case requires an iterative relaxation-based analysis of both the internal and external grid sections, which can be expensive. This difficulty was overcome in [13] for the purpose of circuit simulation (not vectorless verification), through the use of the multi-port Norton theorem.

In this work, we provide the first solution in the dynamic case for the purpose of incremental vectorless verification, based on two contributions: 1) upper bounds on the voltage drops at internal nodes are efficiently computed and used in lieu of the worst-case drops, and 2) a macromodel is constructed for the sub-grid based on the movement of the internal current sources by adapting the multi-port Norton theorem proposed in [13] to our verification framework, followed by the reduction of the passive RC circuit by
combining the moment matching-based approach as described in [21] and [22] with the node elimination-based approach of [18]. A version of this work has appeared in [38].

In the next section, we present an efficient way to compute the bounds on the worst-case voltage drops, benefiting from the fact that voltage drops at internal nodes are not required, and the following sections describe our power grid macromodeling approach that then allows for incremental verification.

### 3.2 Efficient Bounds Computation

To compute \( v_{ub} \) using (2.16), we need to have an estimate of the worst-case voltage drop entries of \( V_a \) at both internal and external nodes. From (2.3), we have:

\[
G_{23}^T v_{prt}(t) + G_{33} v_{int}(t) + C_{int} \dot{v}_{int}(t) = i_{s, int}(t)
\]

which, after time-discretization, leads to:

\[
v_{int}(\Delta t) = A_{int}^{-1} i_{s, int}(\Delta t) - A_{int}^{-1} G_{23}^T v_{prt}(\Delta t)
\]

where \( A_{int} = (G_{33} + C_{int} / \Delta t) \) is a symmetric, positive-definite, \( M \)-matrix, so that \( A_{int}^{-1} \geq 0 \). Because \( -G_{23}^T \) and \( A_{int}^{-1} \) are non-negative matrices, then in the special case used earlier to define \( V_a \), we can write:

\[
\max_{\forall i_s \in F} (v_{int}(\Delta t)) \leq \max_{\forall i_s \in F} (i_{s, int}(\Delta t)) + T^T \max_{\forall i_s \in F} (v_{prt}(\Delta t))
\]

where the transformation matrix \( T \) is given by:

\[
T = -G_{23} A_{int}^{-1}
\]

Equation (3.3) gives an upper-bound on the worst-case voltage drops at \( t = \Delta t \) for all internal nodes, so that we can write:

\[
V_a = \max_{\forall i_s \in F} (v(\Delta t)) \leq \begin{bmatrix}
I_{ext} & 0 & 0 \\
0 & I_{prt} & 0 \\
0 & T^T & A_{int}^{-1}
\end{bmatrix}
\begin{bmatrix}
\max_{\forall i_s \in F} (v_{ext}(\Delta t)) \\
\max_{\forall i_s \in F} (v_{prt}(\Delta t)) \\
\max_{\forall i_s \in F} (i_{s, int}(\Delta t))
\end{bmatrix}
\]
Table 3.1: Legend

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v(t)$</td>
<td>voltage drop in original grid</td>
</tr>
<tr>
<td>$v'(t)$</td>
<td>voltage drop at internal nodes when port nodes are shorted</td>
</tr>
<tr>
<td>$\hat{v}(t)$</td>
<td>voltage drop after moving the internal current sources</td>
</tr>
<tr>
<td>$\tilde{v}(t)$</td>
<td>voltage drop after macromodeling</td>
</tr>
</tbody>
</table>

where $I_{ext}$ and $I_{prt}$ are identity matrices of sizes $n_{ext}$ and $n_{prt}$, respectively. From this, and because $G^{-1} \geq 0$, we have from (2.16) that:

$$v_{ub} \leq \left(I + G^{-1} \frac{C}{\Delta t}\right) \begin{bmatrix} I_{ext} & 0 & 0 \\ 0 & I_{prt} & 0 \\ 0 & T^T & A_{int}^{-1} \end{bmatrix} \begin{bmatrix} v_{ext}(\Delta t) \\ v_{prt}(\Delta t) \\ i_{s,int}(\Delta t) \end{bmatrix} \forall i_s \in F$$

(3.5)

Because $\max_{\forall i_s \in F}(i_{s,int}(\Delta t)) = i_{L,int}$ (the vector of local constraint values for current sources internal to the sub-grid), this gives a faster way to compute an upper bound on $v_{ub}$ which involves solving LPs for external and port nodes only, followed by a standard linear solve:

$$v_{ub} \leq \left(I + G^{-1} \frac{C}{\Delta t}\right) \begin{bmatrix} I_{ext} & 0 & 0 \\ 0 & I_{prt} & 0 \\ 0 & T^T & A_{int}^{-1} \end{bmatrix} \begin{bmatrix} \max_{\forall i_s \in F}(v_{ext}(\Delta t)) \\ \max_{\forall i_s \in F}(v_{prt}(\Delta t)) \\ i_{L,int} \end{bmatrix}$$

(3.6)

3.3 Power Grid Reduction

Because the internals of the sub-grid do not need to be verified, significant performance improvement can be obtained by reducing or eliminating much of the sub-grid network. Two steps are involved in this: 1) moving the internal current sources to the port nodes, which benefits from the multi-port Norton theorem reviewed in section 2.4, and 2) reducing the remaining parasitic $RC$ network inside the sub-grid using MOR.
3.3.1 Moving Internal Current Sources

In HiPRIME [13], multi-port Norton equivalent circuits were used to move the current sources internal to a block to the ports. This previous work benefited from the multi-port Norton theorem for simulation purposes. In our work, we adapt this theorem for use in verification, where the current sources are not known, but are instead subject to current constraints.

Norton Equivalent Current Sources

The grid equation of the sub-grid when the port nodes are shorted to ground as in Fig. 2.9 is given by:

\[ G_{33} v_{\text{int}}(t) + C_{\text{int}} i_{\text{int}}(t) = i_{s,\text{int}}(t) \]  (3.7)

where \( v_{\text{int}} \) is the resulting \( n_{\text{int}} \times 1 \) voltage drop vector at internal nodes. Let us call an internal node that connects to a port node \( k \), a neighbor of \( k \). The current through a port node to ground \( i'_k(t) \) is given by:

\[ i'_k(t) = \sum_{\text{neighbors} j \text{ of } k} g_{kj} v'_{\text{int}_j}(t) \]  (3.8)

where \( g_{kj} \) is the conductance through which port node \( k \) is connected to internal node \( j \) and \( v'_{\text{int}_j}(t) \) is the voltage drop for internal node \( j \). In (2.3), \( G_{23} \) is the \( n_{\text{prt}} \times n_{\text{int}} \) matrix consisting of all the conductance links from port nodes to internal nodes. Therefore, using (3.8), the Norton short-circuit current vector \( i'(t) \) can be written as:

\[ i'(t) = -G_{23} v'_{\text{int}}(t) \]  (3.9)

Modified Grid

The grid resulting after the internal current sources of the sub-grid have been removed and replaced by the new port current sources as in Fig. 2.11, will be referred to as the
modified grid. In this modified grid, the voltage drops at the nodes will be denoted by $\hat{v}(t)$, and the system equation becomes:

$$
\mathbf{G} \begin{bmatrix}
\hat{v}_{\text{ext}}(t) \\
\hat{v}_{\text{prt}}(t) \\
\hat{v}_{\text{int}}(t)
\end{bmatrix} + \mathbf{C} \begin{bmatrix}
\dot{\hat{v}}_{\text{ext}}(t) \\
\dot{\hat{v}}_{\text{prt}}(t) \\
\dot{\hat{v}}_{\text{int}}(t)
\end{bmatrix} = \begin{bmatrix}
\mathbf{I}_{\text{ext}} & 0 & 0 \\
0 & \mathbf{I}_{\text{prt}} & 0 \\
0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
i_{s,\text{ext}}(t) \\
i_{s,\text{prt}}(t) \\
i_{s,\text{int}}(t)
\end{bmatrix} + \begin{bmatrix}
0 \\
\mathbf{I}_{\text{prt}} \\
0
\end{bmatrix} \dot{i}(t)
$$

which can also be written as:

$$
\mathbf{G}\hat{v}(t) + \mathbf{C}\dot{\hat{v}}(t) = \mathbf{J}i_{s}(t) + \mathbf{K}\dot{i}(t)
$$

(3.10)

where $\mathbf{J}$ is an $n \times n$ matrix consisting of $\mathbf{I}_{\text{ext}}$ and $\mathbf{I}_{\text{prt}}$, and $\mathbf{K}$ is an $n \times n_{\text{prt}}$ matrix consisting of $\mathbf{I}_{\text{prt}}$. Time-discretizing (3.10) gives:

$$
\mathbf{G}\hat{v}(t) + \frac{\mathbf{C}}{\Delta t}(\hat{v}(t) - \hat{v}(t - \Delta t)) = \mathbf{J}i_{s}(t) + \mathbf{K}\dot{i}(t)
$$

(3.11)

We now return to the special case situation used to define $V_a$ earlier. In that case, the voltage in the modified grid at time $t = \Delta t$ is given by:

$$
\hat{v}(\Delta t) = \mathbf{A}^{-1}(\mathbf{J}i_{s}(\Delta t) + \mathbf{K}\dot{i}(\Delta t))
$$

(3.12)

Likewise, time-discretizing (3.7) and evaluating at $t = \Delta t$, we get:

$$
v'_{\text{int}}(\Delta t) = \mathbf{A}_{\text{int}}^{-1}i'_{s,\text{int}}(\Delta t)
$$

(3.13)

From (3.9) and (3.13),

$$
\dot{i}'(\Delta t) = -\mathbf{G}_{23}v'_{\text{int}}(\Delta t)
$$

$$
= -\mathbf{G}_{23}\mathbf{A}_{\text{int}}^{-1}i'_{s,\text{int}}(\Delta t) \equiv \mathbf{T}i_{s,\text{int}}(\Delta t)
$$

(3.14)

which can also be written as:

$$
\dot{i}'(\Delta t) = \begin{bmatrix} 0 & 0 & \mathbf{T} \end{bmatrix} \begin{bmatrix} i_{s,\text{ext}}(\Delta t) \\
i_{s,\text{prt}}(\Delta t) \\
i_{s,\text{int}}(\Delta t)
\end{bmatrix} = \mathbf{P}i_{s}(\Delta t)
$$

(3.15)
where $P$ is a $n_{prt} \times n$ matrix that contains $T$. Using (3.15) in (3.12), we get:

$$\hat{v}(\Delta t) = A^{-1}(J + KP)i_s(\Delta t) = A^{-1}\hat{J}i_s(\Delta t)$$

(3.16)

where:

$$\hat{J} = \begin{bmatrix} I_{\text{ext}} & 0 & 0 \\ 0 & I_{\text{prt}} & T \\ 0 & 0 & 0 \end{bmatrix}$$

(3.17)

The above results will be used in the following to efficiently verify the external nodes in the modified grid. From Norton’s theorem, the modified grid will exhibit the same voltage response at external and port nodes as the original grid. Therefore, $\forall i_s \in \mathcal{F}$:

$$v_{\text{ext}}(\Delta t) = \hat{v}_{\text{ext}}(\Delta t)$$

$$v_{\text{prt}}(\Delta t) = \hat{v}_{\text{prt}}(\Delta t)$$

so that:

$$\max_{\forall i_s \in \mathcal{F}}(v_{\text{ext}}(\Delta t)) = \max_{\forall i_s \in \mathcal{F}}(\hat{v}_{\text{ext}}(\Delta t))$$

(3.18)

$$\max_{\forall i_s \in \mathcal{F}}(v_{\text{prt}}(\Delta t)) = \max_{\forall i_s \in \mathcal{F}}(\hat{v}_{\text{prt}}(\Delta t))$$

(3.19)

Therefore, any verification that we will do below on the external nodes in the modified grid will also verify the same nodes in the original grid:

$$v_{ub} \leq \left(I + G^{-1}C / \Delta t \right) \begin{bmatrix} I_{\text{ext}} & 0 & 0 \\ 0 & I_{\text{prt}} & 0 \\ 0 & T^T & A_{\text{int}}^{-1} \end{bmatrix} \begin{bmatrix} \max_{\forall i_s \in \mathcal{F}}(\hat{v}_{\text{ext}}(\Delta t)) \\ \max_{\forall i_s \in \mathcal{F}}(\hat{v}_{\text{prt}}(\Delta t)) \\ i_{L,\text{int}} \end{bmatrix}$$

(3.20)

Equation (3.16) is significant in that it is a replacement of (2.17) for the case of the modified grid, and so it provides a way to find $\max(\hat{v}_{\text{ext}}(\Delta t))$ and $\max(\hat{v}_{\text{prt}}(\Delta t))$ using the same user-provided local and global constraints that were given for the original grid.
3.3.2 Sub-grid Reduction

After moving the internal current sources to the port nodes, we are left with a sub-grid consisting only of parasitic \(RC\) elements. Therefore, we can use a passive MOR technique to reduce the internals of the sub-grid. The reduction approach that we have found applicable and beneficial for this work combines the two standard techniques of moment matching and node elimination.

**Moment Matching**

We use a nodal formulation-based method [22] to compute the moments of the system transfer function. The passive sub-grid is first isolated from the rest of the grid by removing (i.e., make into an open circuit) the connections from all port nodes to external nodes. Therefore, the isolated sub-grid can be represented in the \(s\)-domain by:

\[
\begin{pmatrix}
\hat{G}_{22} & G_{23} \\
G_{23}^T & G_{33}
\end{pmatrix} + s
\begin{pmatrix}
C_{\text{prt}} & 0 \\
0 & C_{\text{int}}
\end{pmatrix}
\begin{pmatrix}
v_{\text{prt}}(s) \\
v_{\text{int}}(s)
\end{pmatrix} =
\begin{pmatrix}
i_{s,\text{prt}}(s) + i'(s) \\
0
\end{pmatrix}
\tag{3.21}
\]

where \(\hat{G}_{22}\) is an \(n_{\text{prt}} \times n_{\text{prt}}\) conductance matrix that is derived from \(G_{22}\) by removing the connections from port nodes to external nodes represented by \(G_{12}^T\). The admittance looking into the ports of the sub-grid, a matrix \(Y(s)\), can be approximated as [39]:

\[
Y(s) \approx M_0 + M_1 s
\tag{3.22}
\]

where \(M_0 = \hat{G}_{22} - G_{23}V\) and \(M_1 = C_{\text{prt}} + V^T C_{\text{int}} V\) are the \(n_{\text{prt}} \times n_{\text{prt}}\), zero and first-order moment matrices with \(V = G_{33}^{-1} G_{23}^T\). Because of the quadratic form of \(M_1\), it is clear that it is a non-negative matrix.

For circuits with a non-negative \(M_1\) matrix, a \(2\pi\)-model between pairs of ports was constructed in [21] (RC MOR, reviewed earlier in sub-section 2.5.2) by matching the zero and first-order moments. We will use this approach to reduce the circuit between pairs of ports.
Node Elimination

Note that the T-model given by RC MOR generates an extra node (a new internal node) for each pair of ports. If we have \( n_{\text{prt}} \) port nodes, generating a T-model for every pair of ports can be expensive because it will result in \( n_{\text{prt}}(n_{\text{prt}} - 1)/2 \) new nodes. To overcome this, we can eliminate many of the new internal nodes by using the node elimination-based reduction approach proposed in [18]. For every new internal node, the nodal time constant [18] is given by:

\[
\tau = \frac{C_{ij}R_{ij1}R_{ij2}}{R_{ij1} + R_{ij2}} \quad (3.23)
\]

If \( \tau < \tau_N \) where \( \tau_N \) is a user-specified nodal time constant value, the internal node can be eliminated by adding capacitors \( C_i \) and \( C_j \) to port nodes \( i \) and \( j \) and resistors \( R_{ij} \) between \( i \) and \( j \). The capacitors and resistors are given by:

\[
R_{ij} = R_{ij1} + R_{ij2} \\
C_i = \frac{C_{ij}R_{ij2}}{R_{ij1} + R_{ij2}}, \quad C_j = \frac{C_{ij}R_{ij1}}{R_{ij1} + R_{ij2}} \quad (3.24)
\]

Sparsification

Once the system matrices of the reduced model are formed, it turns out that they often contain a large number of negligible (near zero) entries. As a final step in the reduction, therefore, we have found it useful to apply a sparsification step, where entries whose absolute value is below a small value \( \kappa \) are simply set to 0. We have found the error resulting from this to be insignificant and very much worth the effort.

Final Reduced Model

After applying the reduction techniques described above, we end up with new conductance (\( \tilde{G}_{\text{sub}} \)) and capacitance (\( \tilde{C}_{\text{sub}} \)) matrices for the isolated sub-grid, given by:

\[
\tilde{G}_{\text{sub}} = \begin{bmatrix} \tilde{G}_{22} & \tilde{G}_{23} \\ \tilde{G}_{23}^T & \tilde{G}_{33} \end{bmatrix}, \quad \tilde{C}_{\text{sub}} = \begin{bmatrix} \tilde{C}_{\text{prt}} & 0 \\ 0 & \tilde{C}_{\text{int}} \end{bmatrix} \quad (3.25)
\]
Algorithm 1 INCR_VERIFY

**Input:** Partitioned power grid matrices in (2.3), $\tau_N, \kappa, \delta_1$ and $\delta_2$

**Output:** Upper bounds on worst-case voltage drops for external nodes

1: Construct subgrid matrices in (3.21)

2: $(T, \tilde{G}_{\text{sub}}, \tilde{C}_{\text{sub}}) = \text{MACRO}(\text{subgrid matrices, } \tau_N, \kappa, \delta_2)$

3: Construct $\tilde{J}, \tilde{G}, \tilde{C}$ and $\tilde{A}$

4: for $(j = 1, \ldots, n_{\text{ext}} + n_{\text{prt}})$ do

5: Compute $j^{th}$ row of $\tilde{A}^{-1}$ using SPAI [8] with $\delta = \delta_1$

6: Multiply that row by the columns of $\tilde{J}$, get row vector $d$

7: Maximize: $d \cdot i_s$, subject to: $i_s \in \mathcal{F}$

8: end for

9: Compute $\tilde{v}_{\text{ub}}$ using (3.29)

where $\tilde{G}_{22}$ and $\tilde{C}_{\text{prt}}$ are the modified port-to-port conductance matrix and capacitance matrix respectively, $\tilde{G}_{33}$ and $\tilde{C}_{\text{int}}$ are $\bar{n}_{\text{int}} \times \bar{n}_{\text{int}}$ conductance and capacitance matrices for the $\bar{n}_{\text{int}}$ remaining (new) internal nodes, and $\tilde{G}_{23}$ and $\tilde{G}_{23}^T$ are matrices consisting of connections between the port nodes and the remaining internal nodes.

### 3.4 Verification after Reduction

After macromodeling, the reduced full grid matrices can be constructed by stitching together the reduced sub-grid and external grid matrices using the connections from external nodes to port nodes:

\[
\tilde{G} = \begin{bmatrix} G_{11} & G_{12} & 0 \\ G_{12}^T & \tilde{G}_{22}' & \tilde{G}_{23} \\ 0 & \tilde{G}_{23}^T & \tilde{G}_{33} \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} C_{\text{ext}} & 0 & 0 \\ 0 & \tilde{C}_{\text{prt}} & 0 \\ 0 & 0 & \tilde{C}_{\text{int}} \end{bmatrix}
\]

where $\tilde{G}_{22}'$ is the updated port-to-port conductance matrix in which the connections from external nodes to port nodes have been added. Since we have used a realizable macromodeling approach, $\tilde{G}$ is also an $\bar{n} \times \bar{n}$ symmetric, positive-definite, $\mathcal{M}$-matrix, where $\bar{n} = n_{\text{ext}} + n_{\text{prt}} + \bar{n}_{\text{int}}$. 
Let \( \tilde{v}(t) \) be the voltage drop vector in the reduced grid, which is partitioned in the usual way into \( \tilde{v}_{\text{ext}}(t), \tilde{v}_{\text{prt}}(t), \) and \( \tilde{v}_{\text{int}}(t) \). It is to be expected that:

\[
\begin{align*}
    v_{\text{ext}}(t) &= \hat{v}_{\text{ext}}(t) \approx \tilde{v}_{\text{ext}}(t) \\
    v_{\text{prt}}(t) &= \hat{v}_{\text{prt}}(t) \approx \tilde{v}_{\text{prt}}(t) \\
\end{align*}
\]  

(3.27)

Applying (3.16) to the reduced grid gives:

\[
\tilde{v}(\Delta t) = \tilde{A}^{-1}\tilde{J}i_{\text{s}}(\Delta t)
\]  

(3.28)

where \( \tilde{A} = \left( \tilde{G} + \frac{\tilde{C}}{\Delta t} \right) \) is an \( \mathcal{M} \)-matrix and \( \tilde{J} \) is an \( \tilde{n} \times n \) matrix with the first \( n_{\text{ext}} + n_{\text{prt}} \) rows equal to the first \( n_{\text{ext}} + n_{\text{prt}} \) rows of \( \hat{J} \) defined in (3.17), and the remaining \( \tilde{n}_{\text{int}} \) rows have all entries equal to 0. The worst-case voltage drop at external and port nodes can be found (approximately) by element-wise maximization of \( \tilde{v}_{\text{ext}} \) and \( \tilde{v}_{\text{prt}} \). To the extent that \( \hat{v}_{\text{ext}} \approx \tilde{v}_{\text{ext}} \) and \( \hat{v}_{\text{prt}} \approx \tilde{v}_{\text{prt}} \), we can approximately restate (3.20) as:

\[
\begin{align*}
    v_{\text{ub}} &\leq \tilde{v}_{\text{ub}} = \left( I + G^{-1}C \frac{1}{\Delta t} \right) \begin{bmatrix} I_{\text{ext}} & 0 & 0 \\ 0 & I_{\text{prt}} & 0 \\ 0 & T^T & \tilde{A}_{\text{int}}^{-1} \end{bmatrix} \begin{bmatrix} \max_{\forall i_s \in F}(\tilde{v}_{\text{ext}}(\Delta t)) \\ \max_{\forall i_s \in F}(\tilde{v}_{\text{prt}}(\Delta t)) \\ i_{L,\text{int}} \end{bmatrix} \\
\end{align*}
\]  

(3.29)

where \( \tilde{v}_{\text{ub}} \) is the \( n \)-vector of upper bounds on worst-case voltage drops with the first \( n_{\text{ext}} \) entries corresponding to upper bounds for external nodes.

### 3.5 Implementation

The overall flow of the proposed incremental verification approach is given in Algorithm 1. We start with a user-specified power grid and sub-grid, along with parameter values for \( \tau_N, \kappa, \) and error tolerance values \( (\delta_1, \delta_2) \) for the sparse approximate inverse (SPAI [8]) engine, which is inherently parallelizable and can compute a single column/row of the approximate inverse. The grid matrices are appropriately partitioned and reduction of the sub-grid is performed. As a result, the size of the original power grid is reduced and the internal current sources are moved to the port nodes. The inverse of the matrix \( \tilde{A} \) is then computed, row by row, using SPAI, and every row is multiplied by \( \tilde{J} \) to account for the effect of the movement of internal current.
Chapter 3. Incremental Power Grid Verification

43

sources. Next, we maximize the voltage drop at external and port nodes, and compute the upper bounds on worst-case voltage drops by using (3.29).

The macromodeling algorithm is presented in Algorithm 2. To avoid the cost of constructing the full matrix $A_{int}^{-1}$, we generate one row of the transformation matrix $T = -G_{23}A_{int}^{-1}$ at a time, using SPAl. For a port node $k$, we first identify the neighbors, and the connections ($g_{kj}$) to the neighbors. Then, we compute the corresponding row of the approximate inverse, multiply the row vector by $g_{kj}$, and add the result to the $k^{th}$ row of $T$. Calculation of moments is also efficiently done by factorizing $G_{33}$, followed by standard system solves.

3.6 Experimental Results

A C++ implementation has been written to test the proposed approach. We use SPAl to compute the approximate inverses, and solve the linear programs using MOSEK [40]. The test grids were generated from user specifications, including grid dimensions, metal layers, pitch and width per layer, and supply voltage sites and current sources distribution. The supply voltages and current sources were randomly placed on the grid. Around 15-20% of the nodes had current sources attached to them while $V_{dd}$ sources were present at about 7% of the nodes. The technology specifications were consistent with 1.1 V 65nm CMOS technology. A global constraint is specified for the sub-grid and other global constraints were specified to cover the entire chip. The sub-grid nodes are also identified by the user. Computations were done using a 2.6 GHz Linux machine with 24 GB of RAM. A SPAl error tolerance value of $\delta_1 = 0.1$mV is used to compute the approximate inverse for the original and modified power grids. A lower value of tolerance $\delta_2 = 0.01$mV is used to construct $T$.

Table 3.2 shows the speed and accuracy of the proposed bounds computation approach (section 3.2). Since we are interested in analyzing only the external nodes, we report maximum error and average percentage error values for the upper bounds on the worst-case voltage drops at external nodes only. The runtime and accuracy are compared with the original approach based on finding the worst-case voltage drop at every node, and then computing the upper bound using (2.16). The results show that we are able to achieve significant runtime savings
Table 3.2: Speed and accuracy after using efficient bounds computation

<table>
<thead>
<tr>
<th>Power Grid Name</th>
<th>Sub-grid</th>
<th>Max Error</th>
<th>Avg. % Error</th>
<th>CPU Time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$</td>
<td>$n_{int}$</td>
<td>$n_{prt}$</td>
<td>(mV)</td>
<td></td>
</tr>
<tr>
<td>G1</td>
<td>8,413</td>
<td>3,891</td>
<td>118</td>
<td>0.08</td>
<td>0.075</td>
</tr>
<tr>
<td>G2</td>
<td>18,678</td>
<td>10,788</td>
<td>176</td>
<td>0.08</td>
<td>0.102</td>
</tr>
<tr>
<td>G3</td>
<td>32,554</td>
<td>15,714</td>
<td>208</td>
<td>0.07</td>
<td>0.038</td>
</tr>
<tr>
<td>G4</td>
<td>50,444</td>
<td>29,458</td>
<td>290</td>
<td>0.07</td>
<td>0.055</td>
</tr>
<tr>
<td>G5</td>
<td>72,692</td>
<td>42,764</td>
<td>348</td>
<td>0.07</td>
<td>0.047</td>
</tr>
<tr>
<td>G6</td>
<td>98,162</td>
<td>68,972</td>
<td>402</td>
<td>0.08</td>
<td>0.067</td>
</tr>
<tr>
<td>G7</td>
<td>128,241</td>
<td>95,294</td>
<td>413</td>
<td>0.08</td>
<td>0.064</td>
</tr>
<tr>
<td>G8</td>
<td>162,087</td>
<td>124,824</td>
<td>518</td>
<td>0.08</td>
<td>0.078</td>
</tr>
</tbody>
</table>

with negligible error values. The power grid macromodeling approach was tested with user-specified values for nodal time constant ($\tau_N = 5\text{ps}$), and conductance threshold ($\kappa = 5 \times 10^{-3}$).

Table 3.3 gives the speed and accuracy obtained after applying power grid macromodeling. The runtime for the reduced grid includes the time taken to perform reduction of the sub-grid, movement of internal current sources, and finding the upper bounds on worst-case voltage drop.

The accuracy is compared with the original approach while speed-up is measured with respect to the efficient bounds computation approach. We also report the total speed-up with respect to the original approach. The results show that we incurred an average error of about 1% for large grids while extracting a total speed-up in the range of 3-8x.

The relative error plots for both the approaches are shown in Fig. 3.1 and Fig. 3.2. These plots give the variation of the relative error $(\tilde{v} - v)/v$ versus the worst-case voltage drop values at the external nodes of the power grid. It can be observed from the plots that we have been able to bound the error to under 1mV for the bounds computation and under 3mV for verification performed after macromodeling the sub-grid.

It is to be expected that, if fewer nodes are to be verified, then corresponding time savings
### Table 3.3: Speed and accuracy after applying macromodeling

<table>
<thead>
<tr>
<th>Power Grid</th>
<th>Max Error (mV)</th>
<th>Avg. % Error</th>
<th>CPU time</th>
<th>Speed-up</th>
<th>Total Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Fast $v_{ub}$</td>
<td>Reduced</td>
<td>Reduced vs Original</td>
</tr>
<tr>
<td>G1</td>
<td>2.4</td>
<td>1.96</td>
<td>10.58 min.</td>
<td>4.73 min.</td>
<td>2.23x</td>
</tr>
<tr>
<td>G2</td>
<td>2.01</td>
<td>1.89</td>
<td>24.97 min.</td>
<td>12.26 min.</td>
<td>2.03x</td>
</tr>
<tr>
<td>G3</td>
<td>1.35</td>
<td>1.05</td>
<td>1.38 h.</td>
<td>1.06 h.</td>
<td>1.3x</td>
</tr>
<tr>
<td>G4</td>
<td>1.92</td>
<td>1.07</td>
<td>1.96 h.</td>
<td>1.44 h.</td>
<td>1.36x</td>
</tr>
<tr>
<td>G5</td>
<td>2.01</td>
<td>0.88</td>
<td>3.11 h.</td>
<td>2.36 h.</td>
<td>1.31x</td>
</tr>
<tr>
<td>G6</td>
<td>2.79</td>
<td>1.14</td>
<td>3.51 h.</td>
<td>2.28 h.</td>
<td>1.53x</td>
</tr>
<tr>
<td>G7</td>
<td>3.13</td>
<td>1.11</td>
<td>4.19 h.</td>
<td>2.64 h.</td>
<td>1.58x</td>
</tr>
<tr>
<td>G8</td>
<td>2.6</td>
<td>1.14</td>
<td>5.27 h.</td>
<td>3.14 h.</td>
<td>1.67x</td>
</tr>
</tbody>
</table>

Figure 3.1: Relative Error Plot (efficient bounds)
Chapter 3. Incremental Power Grid Verification

Figure 3.2: Relative Error Plot (macromodeling)

Figure 3.3: Runtime vs Size of Sub-grid
would be achieved. However, in our case, the speed-ups are much higher than would be obtained based solely on this observation. The graph in Fig. 3.3 shows the variation of runtime for power grid verification after applying macromodeling and verification using the efficient bounds computation approach with respect to the size of the sub-grid. The runtimes are scaled to the time required to perform full grid verification. The theoretical runtime is the runtime which is to be expected considering that we are only verifying a part (certain percentage of total number of nodes) of the grid. Fig. 3.4 gives the same variation for speed-up. It can be observed from these plots that, as the size of the sub-grid increases, gains corresponding to power grid reduction become larger. The power grid macromodeling approach has an overhead associated with the movement of current sources that requires explicit computation of the transformation matrix and moment matching that requires a system solve.

Table 3.4 gives the runtime break down for power grid verification after applying macromodeling. It can be observed that the overheads associated with macromodeling can be significant for small grids (G1 and G2). As the sizes of the grids increase, the majority of the time required is spent in performing verification (INCR_VERIFY), thereby making the overheads less
significant. From a comparison between these runtime results and the results obtained from full grid verification, it can be concluded that power grid macromodeling results in reducing the size of individual problems, which in turn results in significant reductions in the time taken to solve LPs and to compute approximate inverses using SPAI.

We introduced a sparsification step in our reduction approach to reduce the number of fill-ins (so as to maintain the sparsity of the matrices) and hence, make optimization faster. A higher value of conductance threshold ($\kappa$) would mean fewer fill-ins, better runtime, but increased error as shown in Fig. 3.5. The plot shows runtime and error variation for grid G1 with $\kappa$, with $\tau_N$ kept constant at $5 \times 10^{-12}$.

Another important step in the reduction of the power grid was node elimination that was applied to the reduced sub-grid in order to remove some of the remaining (newly formed) internal nodes. The nodal time constant ($\tau_N$) determines the extent of reduction. We performed an analysis of the extent of reduction and runtime variation with $\tau_N$ for G1, with $\kappa$ kept constant at 0.005. It can be seen from Fig. 3.6 that as the value of $\tau_N$ is increased, the number of nodes which become eligible for removal increases, resulting in more reduction and improved runtime.

### Table 3.4: Runtime breakdown

<table>
<thead>
<tr>
<th>Power Grid</th>
<th>Total Time</th>
<th>MACRO</th>
<th>INCR_VERIFY</th>
<th>Bound on $v_{int}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(T)</td>
<td>$\tilde{G}<em>{sub}$ and $\tilde{G}</em>{sub}$</td>
<td>Multiply $\tilde{J}$</td>
<td>LP</td>
</tr>
<tr>
<td>G1</td>
<td>4.73 min.</td>
<td>6.16 s.</td>
<td>1.66 s.</td>
<td>0.3 s.</td>
</tr>
<tr>
<td>G2</td>
<td>12.26 min.</td>
<td>12.17 s.</td>
<td>9.92 s.</td>
<td>0.19 s.</td>
</tr>
<tr>
<td>G3</td>
<td>1.06 h.</td>
<td>14.23 s.</td>
<td>13.58 s.</td>
<td>0.63 s.</td>
</tr>
<tr>
<td>G4</td>
<td>1.44 h.</td>
<td>22.21 s.</td>
<td>45.01 s.</td>
<td>4.96 s.</td>
</tr>
<tr>
<td>G5</td>
<td>2.36 h.</td>
<td>29.68 s.</td>
<td>97.75 s.</td>
<td>12.49 s.</td>
</tr>
<tr>
<td>G6</td>
<td>2.28 h.</td>
<td>50.21 s.</td>
<td>2.49 min.</td>
<td>2.81 s.</td>
</tr>
<tr>
<td>G7</td>
<td>2.64 h.</td>
<td>1.05 min.</td>
<td>3.18 min.</td>
<td>42.1 s.</td>
</tr>
<tr>
<td>G8</td>
<td>3.14 h.</td>
<td>1.47 min.</td>
<td>9.11 min.</td>
<td>25.64 s.</td>
</tr>
</tbody>
</table>
Figure 3.5: Runtime and Accuracy vs $\kappa$ for G1

Figure 3.6: % Reduction and Runtime vs $\tau_N$ for G1
Algorithm 2 MACRO(subgrid matrices, $\tau_N$, $\kappa$, $\delta_2$)

Output: $T$ in (3.3), $G_{sub}$ and $C_{sub}$ in (3.25)

1: Construct $A_{int}$

2: for (every port node $k$) do

3: Find neighbors of $k$ and $g_{kj}$ in (3.8)

4: for (every neighbor $j$ of $k$) do

5: Compute the $j^{th}$ row of $A_{int}^{-1}$ using SPAI [8] with $\delta = \delta_2$

6: Multiply the row entries by $g_{kj}$

7: Add the row entries to the $k^{th}$ row of $T$

8: end for

9: end for

10: Compute $M_0$ and $M_1$

11: for (every pair of ports $i, j$) do

12: Compute $R_{ij1}$, $R_{ij2}$ and $C_{ij}$ using (2.46)

13: end for

14: for (every port node $k$) do

15: Compute $R_{kk}$ and $C_{kk}$ using (2.48)

16: end for

17: for (every new internal node created) do

18: Compute $\tau$ using (3.23)

19: if $\tau < \tau_N$ then

20: Compute $R_{ij}$, $C_i$ and $C_j$ using (3.24)

21: Eliminate the new internal node

22: end if

23: end for

24: Drop insignificant connections with conductance less than $\kappa$

25: Construct $\tilde{G}_{sub}$ and $\tilde{C}_{sub}$
Chapter 4

Dimension Reduction of the Feasible Space of Currents

4.1 Introduction

In the incremental verification framework described in the previous chapter, we did not physically replace the internal current sources by current sources at port nodes. Instead, we captured the effect of the movement of internal current sources by using a transformation matrix \((T)\). Therefore, the feasible space and the size of the LPs in terms of the number of variables remained the same. A typical sub-grid has smaller number of port nodes as compared to the number of internal current sources. If we are able to replace the feasible space \(F\) by another feasible space \(\hat{F}\) defined in terms of the modified current source configuration (external sources and Norton equivalent current sources at port nodes), we can reduce the size of the LPs, thereby achieving additional speed-up. In this chapter, we will be focusing on algorithms to compute this new feasible space \(\hat{F}\) and the application of this concept in the construction of the Chip Power Model (also referred to as CPM in this chapter). The next section describes a modified incremental verification approach that benefits from the use of a dimension reduced feasible space \(\hat{F}\).
4.2 Incremental Verification

In incremental verification, we are interested in verifying external nodes by macromodeling the parts of the grid (sub-grid) that need not be verified. To perform macromodeling of the sub-grid, we need to compute the Norton equivalent current sources at port nodes that will replace the current sources internal to the sub-grid. For a general sub-grid, a global constraint may include current sources internal and external to the sub-grid, which makes it difficult to simulate the sub-grid in isolation. In [12], the authors decoupled the global constraints by replacing the constraints on some external sources (based on locality) with an average value derived from the switching activity of the external circuit. In this work, we use the transformation matrix $\mathbf{T}$ to define a feasible space in terms of the new current source configuration in which the external current sources remain intact while Norton equivalent current sources are added to port nodes.

Recall that the voltage drop at time $t = \Delta t$ after performing macromodeling was given by (3.28) as:

$$\tilde{v}(\Delta t) = \tilde{A}^{-1}\tilde{J}_i(\Delta t)$$

Let us first define the following two notations:

$$v|_k \triangleq \text{a sub-vector consisting of the first } k \text{ entries of } v$$

$$\mathbf{A}|_k \triangleq \text{a sub-matrix consisting of the first } k \times k \text{ elements of } \mathbf{A}$$

Using the fact that $\tilde{J}$ is an $\tilde{n} \times n$ matrix with the first $n_{ext} + n_{prt}$ rows equal to $\tilde{J}$ defined in (3.17) and the remaining $\tilde{n}_{int}$ rows have all entries equal to 0, the voltage drop for external and port nodes after macromodeling will be given by:

$$\begin{bmatrix}
\tilde{v}_{ext}(\Delta t) \\
\tilde{v}_{prt}(\Delta t)
\end{bmatrix} = \tilde{v}(\Delta t)|_{\eta} = \tilde{A}^{-1}|_{\eta}(\tilde{J}_i(\Delta t))|_{\eta} = \tilde{A}^{-1}|_{\eta}\tilde{i}'_s$$

(4.1)

where $\eta = n_{ext} + n_{prt}$ and $\tilde{i}'_s$ is the $\eta$-vector of current sources consisting of original current sources at port and external nodes and Norton equivalent current sources attached to the port nodes and is given by:

$$\tilde{i}'_s = (\tilde{J}_i)|_{\eta}$$

(4.2)
Using (4.1) in (3.29), we get:

\[
\tilde{v}_{ub} = \left( I + G^{-1} \frac{C}{\Delta t} \right) \begin{bmatrix} I_{ext} & 0 & 0 \\ 0 & I_{prt} & 0 \\ 0 & T^T & A_{int}^{-1} \end{bmatrix} \begin{bmatrix} \text{emax}_{\forall i_s \in F}(\tilde{A}_{i_s})_{|\eta} \\ i_{L,int} \end{bmatrix}
\] (4.3)

From (4.2), we can define a new feasible space in terms of the modified current source vector \( i_s' \) at time \( t = \Delta t \), that is given by:

\[
\hat{F} = \left\{ i_s' : \exists i_s \in F, \text{ for which } i_s' = (\tilde{J}i_s)_{|\eta} \right\}
\] (4.4)

Claim 1. \( \text{emax}_{\forall i_s \in F}(\tilde{A}_{i_s})_{|\eta} \equiv \text{emax}_{\forall i_s' \in \hat{F}}(\tilde{A}_{i_s'})_{|\eta} \), where \( \hat{F} \) and \( i_s' \) are defined in (4.4) and (4.2), respectively.

Proof. Let us assume that:

\[
\text{emax}_{\forall i_s \in F}(\tilde{A}_{i_s})_{|\eta} = \tilde{A}^{-1}_{|\eta} \begin{bmatrix} (\tilde{J}i_s(1))_{|\eta} & (\tilde{J}i_s(2))_{|\eta} & \ldots & (\tilde{J}i_s(\eta))_{|\eta} \end{bmatrix}
\] (4.5)

where \( i_s(1), i_s(2), \ldots, i_s(\eta) \) are values of current sources such that the corresponding rows of \( \tilde{A}^{-1}_{|\eta}(\tilde{J}i_s)_{|\eta} \) are maximized. Using (4.2) in (4.5), we have:

\[
\text{emax}_{\forall i_s \in F}(\tilde{A}_{i_s})_{|\eta} = \tilde{A}^{-1}_{|\eta} \begin{bmatrix} i_s'(1) & i_s'(2) & \ldots & i_s'(\eta) \end{bmatrix}
\] (4.6)

where

\[
i_s'(1) = (\tilde{J}i_s(1))_{|\eta}
i_s'(2) = (\tilde{J}i_s(2))_{|\eta}
\ldots
\ldots
i_s'(\eta) = (\tilde{J}i_s(\eta))_{|\eta}
\] (4.7)

Using (4.4), (4.7) and the fact that \( i_s^{(k)} \in F, \forall k = 1, \ldots, \eta \), we have:

\[
i_s^{(k)} \in \hat{F}, \forall k = 1, \ldots, \eta
\] (4.8)
Thus,

\[
\text{emax}(\tilde{A}^{-1}|_\eta(\tilde{J}_{s}|_\eta) \leq \text{emax}(\tilde{A}^{-1}|_\eta i'_{s})
\]

(4.9)

Similarly, assuming that:

\[
\text{emax}(\tilde{A}^{-1}|_\eta i'_{s}) = \tilde{A}^{-1}|_\eta \begin{bmatrix} i'(1)_{s} \ i'(2)_{s} \ \ldots \ i'(\eta)_{s} \end{bmatrix}
\]

(4.10)

where \(i'(1)_{s}, i'(2)_{s}, \ldots, i'(\eta)_{s}\) are values of current sources such that the corresponding rows of \(\tilde{A}^{-1}|_\eta i'_{s}\) are maximized. Since \(i'(k)_{s} \in \tilde{F}, \forall k = 1, \ldots, \eta,\) we can find \(i(k)_{s} \in F\) such that:

\[
\begin{align*}
i'(1)_{s} &= (\tilde{J}_{s}(1))|_\eta \\
i'(2)_{s} &= (\tilde{J}_{s}(2))|_\eta \\
\vdots & \\
i'(\eta)_{s} &= (\tilde{J}_{s}(\eta))|_\eta
\end{align*}
\]

(4.11)

Using (4.11) in (4.10), we get:

\[
\text{emax}(\tilde{A}^{-1}|_\eta i'_{s}) = \tilde{A}^{-1}|_\eta \begin{bmatrix} (\tilde{J}_{s}(1))|_\eta \ (\tilde{J}_{s}(2))|_\eta \ \ldots \ (\tilde{J}_{s}(\eta))|_\eta \end{bmatrix}
\]

(4.12)

Since \(i(k)_{s} \in F, \forall k = 1, \ldots, \eta,\)

\[
\text{emax}(\tilde{A}^{-1}|_\eta(\tilde{J}_{s}|_\eta) \geq \text{emax}(\tilde{A}^{-1}|_\eta i'_{s})
\]

(4.13)

Therefore, from (4.9) and (4.13):

\[
\text{emax}(\tilde{A}^{-1}|_\eta(\tilde{J}_{s}|_\eta) \equiv \text{emax}(\tilde{A}^{-1}|_\eta i'_{s})
\]

(4.14)

Using (4.14), it can be concluded that (4.3) is equivalent to:

\[
\tilde{v}_{ub} = \left( I + G^{-1} \frac{C}{\Delta t} \right) \begin{bmatrix} I_{ext} & 0 & 0 \\
0 & I_{prt} & 0 \\
0 & T^T & A^{-1}_{int} \end{bmatrix} \begin{bmatrix} \text{emax}(\tilde{A}^{-1}|_\eta i'_{s}) \\
\text{\quad \quad \quad \quad \quad i_{L,int} \end{bmatrix}
\]

(4.15)

To define the problem, we make the following claim:
Claim 2. \( \hat{\mathcal{F}} \) is a convex polytope

Proof. From (2.15), it can be noted that \( \mathcal{F} \) is bounded by an intersection of finitely many halfspaces (given by \( U_i s \leq u \) and \( U_i s \geq 0 \)) and is thus a convex polytope. Let \( i'_1 \) and \( i'_2 \) be two points in \( \hat{\mathcal{F}} \) which can be written as:

\[
i'_1 = (\tilde{\mathbf{J}} i_1)|_\eta, \quad i'_2 = (\tilde{\mathbf{J}} i_2)|_\eta
\] (4.16)

where \( i_1, i_2 \in \mathcal{F} \). Let us take a point \( i'_3 \) such that \( i'_3 \) lies on the line segment joining \( i'_1 \) and \( i'_2 \). Hence,

\[
i'_3 = \alpha i'_1 + (1 - \alpha) i'_2
\] (4.17)

where \( 0 \leq \alpha \leq 1 \). From (4.16) and (4.17), we have:

\[
i'_3 = (\tilde{\mathbf{J}} \alpha i_1)|_\eta + (\tilde{\mathbf{J}} (1 - \alpha) i_2)|_\eta
\]
\[= (\tilde{\mathbf{J}} (\alpha i_1 + (1 - \alpha) i_2))|_\eta
\]
\[= (\tilde{\mathbf{J}} i_3)|_\eta
\] (4.18)

Since \( \mathcal{F} \) is convex and \( i_3 \) lies on the line segment joining \( i_1 \) and \( i_2 \), then \( i_3 \) is also in \( \mathcal{F} \). Therefore, from (4.4), \( i'_3 \in \hat{\mathcal{F}} \) which implies that \( \hat{\mathcal{F}} \) is a convex set. It has been proved in [41] that any linear mapping of a polytope is also a polytope. Since \( \hat{\mathcal{F}} \) is a linear mapping of \( \mathcal{F} \) and \( \mathcal{F} \) is a convex polytope, then \( \hat{\mathcal{F}} \) is also a convex polytope. \( \square \)

Therefore, the convex polytope \( \hat{\mathcal{F}} \) can be expressed as an intersection of finitely many hyperplanes:

\[
\hat{\mathcal{F}} = \{i'_s : l' \leq U' i'_s \leq u' \}
\] (4.19)

In the sub-section that follows, we the hyperplanes that define \( \hat{\mathcal{F}} \).

4.2.1 Defining \( \hat{\mathcal{F}} \)

The linear mapping \( (\tilde{\mathbf{J}}) \) that converts the original current source configuration to the modified current source configuration is non-invertible, implying that there is no one-to-one and onto
relationship between $i'_s$ and $i_s$. In the absence of such a relationship, the convex polytope $\hat{F}$ can be computed as follows [41]:

1. Find the set of extreme points of $\mathcal{F}$, denoted by $E$.

2. Compute $E'$ such that $E' = (\tilde{J}E)|_\eta$, where $E'$ is the superset of extreme points of $\hat{F}$.

3. Compute the convex hull that contains the points in the set $E'$.

The time complexity of the above algorithm is $O(N^{\lfloor \eta/2 \rfloor})$, where $N$ is the number of points in the set $E'$ and $\eta$ is the dimension of the new feasible space $\hat{F}$. Such an exponential time complexity makes the exact solution infeasible in our case. An approximate algorithm for the construction of multidimensional convex hulls was proposed in [42]. The algorithm starts with a set of reference direction vectors represented in matrix form as rows of $\tilde{U}$, such that they are regularly distributed on the unit hypersphere. For each direction vector $n^i$, the inner product $\langle n^i, p^i \rangle$ is maximized and minimized over all $p^i \in E'$ which defines the feasible space (or convex hull) as:

$$\hat{F} = \{ i'_s : \hat{l} \leq \hat{U}i'_s \leq \hat{u} \} \quad (4.20)$$

where $\hat{l}$ and $\hat{u}$ are the results of minimization and maximization of the inner products:

$$\begin{bmatrix} \hat{u} \\ \hat{l} \end{bmatrix} = \max_{\forall i'_s \in E'} (\hat{U}i'_s) \quad (4.21)$$

In this work, we use a modified version of this algorithm in which the set of direction vectors is chosen based on the original constraint matrix $U$ defined in (2.14) and the transformation matrix $T$.

**Direction Vectors**

In the approximate algorithm to compute the convex hull for a set of points [42], the direction vectors are important as they are the normal vectors to the hyperplanes that will describe the convex hull. These hyperplanes circumscribe the exact convex hull meaning that the polytope resulting from the approximate algorithm contains the original or exact convex hull. This will be
proved later in the text. To better approximate the convex hull, we need to find the direction vectors such that the original convex hull is tightly contained by the approximate polytope. Since the direction vectors are spread out uniformly around the unit hypersphere [42], the quality of approximation is directly dependent on the number of these direction vectors. This implies that we need a large number of direction vectors to get a good approximation for the original space, which can be expensive because an “eopt” operation is required for every direction vector.

We observe that the feasible space in our case is irregular meaning that there are certain regions that require more detailing (more direction vectors) than others. The matrix $U$ is such that every row of the matrix corresponds to a direction vector that describes $F$. In our work, the matrix $(\tilde{J})$ transforms the original current source configuration to the modified current source configuration in which the Norton equivalent current sources are added at the port nodes. Geometrically, it means that a point inside the $n$-dimensional space ($F$) is translated into a point inside the $\eta$-dimensional space ($\hat{F}$). Similarly, a direction vector (in $F$) can be projected onto the $\eta$-dimensional space by using the transformation $(\tilde{J})$. Using the above observations, we came up with a heuristic to find the direction vectors.

The first $n$ rows of $U$ are composed of an identity matrix $I_n$ that represents the local constraints on the current sources. Using the same construction, the first $\eta$ rows of the reference direction vector matrix $\tilde{U}$ are composed of an identity matrix $I_\eta$ that represents the local constraints on the current sources present in the modified configuration. In the original constraint matrix, the global constraints are represented by the last $m$ rows of $U$. We use $\tilde{J}$ to project the hyperplanes representing the global constraints into the new $\eta$-dimensional space. Therefore, the last $m$ rows of $\tilde{U}$ denoted by $\tilde{U}_m$ can be given by:

$$\tilde{U}_m = \left[ \begin{array}{c} (\tilde{J} U^T_{(\eta+1)})_{\eta} \\ (\tilde{J} U^T_{(\eta+2)})_{\eta} \\ \vdots \\ (\tilde{J} U^T_{(m+n)})_{\eta} \end{array} \right]^T$$

(4.22)

where $U^T_{(k)}$ is the $k^{th}$ column of $U^T$. Using the above construction, we get an $m + \eta \times \eta$ matrix $\tilde{U}$. The convex polytope $\tilde{F}$ obtained by using the direction vectors given by $\tilde{U}$ will be able to better approximate $\hat{F}$ because the direction vectors are obtained using the same transformation and the same construction which was used to obtain $\hat{F}$ from $F$ and therefore, are better able
to identify the regions that need more detailing.

New Feasible Space

After finding the direction vectors, the authors in [42] compute \( \hat{l} \) and \( \hat{u} \) using (4.21). Using the same technique, the feasible space \( \tilde{F} \) can now be defined as:

\[
\tilde{F} = \{ i'_s : \tilde{l} \leq \tilde{U} i'_s \leq \tilde{u} \}
\]  

(4.23)

where

\[
\begin{bmatrix}
\tilde{u} \\
\tilde{l}
\end{bmatrix}
= \text{eopt} \begin{bmatrix}
\tilde{l}'_s \\
\tilde{u}'_s
\end{bmatrix}_{\forall i'_s \in E'}
\]  

(4.24)

Using (4.2) and the fact that \( E' = (\hat{J} E)|_{\eta} \), (4.24) can also be written as:

\[
\begin{bmatrix}
\tilde{u} \\
\tilde{l}
\end{bmatrix}
= \text{eopt} \begin{bmatrix}
\tilde{u}(\hat{J} i_s) |_{\eta} \\
\tilde{l}(\hat{J} i_s) |_{\eta}
\end{bmatrix}_{\forall i_s \in E}
\]  

(4.25)

Since \( E \) is a set of the extreme points of \( F \) and solution to an LP is found at an extreme point of the feasible space [41], we have:

\[
\begin{bmatrix}
\tilde{u} \\
\tilde{l}
\end{bmatrix}
= \text{eopt} \begin{bmatrix}
\tilde{u}(\hat{J} i_s) |_{\eta} \\
\tilde{l}(\hat{J} i_s) |_{\eta}
\end{bmatrix}_{\forall i_s \in F}
\]  

(4.26)

The matrix \( \hat{J} \) is composed of an identity matrix and the transformation matrix (\( T \)). Since \( T \) is a non-negative matrix (proved in section 3.2), \( J \) is also a non-negative matrix. The constraint matrix \( U \) is composed of 0s and 1s and is thus non-negative. Therefore, the new constraint matrix \( \tilde{U} \) in (4.22) is also a non-negative matrix. From (2.15) and the arguments above, the result of minimization operation in (4.26) is a zero-vector which implies that \( \tilde{l} = 0 \). Therefore, (4.26) gets reduced to:

\[
\tilde{u} = \text{emax} (\tilde{U}(\hat{J} i_s) |_{\eta})_{\forall i_s \in F}
\]  

(4.27)

This implies that we need \( \eta + m \) optimizations to compute \( \tilde{u} \). The new local constraints can be separated out from \( \tilde{u} \) as:

\[
i'_L = \text{emax} (I_{\eta}(\hat{J} i_s) |_{\eta})_{\forall i_s \in F}
\]  

(4.28)
Algorithm 3 compute\_new\_space(\(U, \tilde{J}\))

**Input:** \(U\) in 2.15 and \(\tilde{J}\)

**Output:** New feasible space \(\tilde{F}'\) in (4.33)

1. Initialize \(\tilde{U}\) to a zero matrix \((\eta + m \times \eta)\)
2. for \((j = 1, \ldots, n_{ext} + n_{prt})\) do
   3. \(\tilde{U}[j][j] = 1\)
3. end for
4. Compute \(\tilde{U}_m\) using (4.22)
5. Construct \(\tilde{U}\)
6. Compute \(i''_L\) using (4.30)
7. for \((j = 1, \ldots, m)\) do
   8. Multiply the \(j^{th}\) row of \(\tilde{U}_m\) by the first \(\eta\) rows of \(\tilde{J}\) to get a row vector \(e\)
   9. Maximize: \(e \cdot i_s\), subject to: \(i_s \in F\)
10. end for
11. Construct \(\tilde{u}'\) using (4.32)
12. Construct \(\tilde{u}'\) using (4.32)

while the new global constraints will be given by:

\[
i'_G = \max_{i_s \in \tilde{F}} (\tilde{U}_m (\tilde{J}i_s)) |_{\eta} \quad (4.29)
\]

Further reduction in the number of LP solves can be made by using the original local constraints to compute \(i''_L\) instead of maximizing over the feasible space \(\tilde{F}\). The new local constraints can then be computed by:

\[
i''_L = \tilde{J}i_L \quad (4.30)
\]

The results of maximization using the original local constraints instead of the original feasible space are pessimistic because it can never be the case that all the chip components are simultaneously drawing maximum currents, implying that:

\[
i''_L \geq i'_L \quad (4.31)
\]
Algorithm 4 NEW_INCR_VERIFY

Input: Partitioned power grid matrices in (2.3), \(\tau_N, \kappa, \delta_1\) and \(\delta_2\)

Output: Upper bounds on worst-case voltage drops for external nodes

1: Construct subgrid matrices in (3.21)
2: \((T, \hat{G}_{\text{sub}}, \hat{C}_{\text{sub}}) = \text{MACRO}(\text{subgrid matrices, } \tau_N, \kappa, \delta_2)\)
3: Construct \(\hat{J}, \hat{G}, \hat{C}\) and \(\hat{A}\)
4: \(\hat{F}' = \text{compute\_new\_space}(U, \hat{J})\)
5: for \((j = 1, \ldots, \eta)\) do
6: Compute the \(j\)th row of \(\hat{A}^{-1}\) using SPAI [8] with \(\delta = \delta_1\)
7: Use the first \(\eta\) elements of the row to get row vector \(d\)
8: Maximize: \(d \cdot i'_s\), subject to: \(i'_s \in \hat{F}'\)
9: end for
10: Compute \(\tilde{v}_{ub}\) using (4.34)

Combining (4.29) and (4.30), we get:

\[
\tilde{u}' = \begin{bmatrix} i'_L \\ i'_G \end{bmatrix}
\] (4.32)

so that the new feasible space is given by:

\[
\hat{F}' = \{i'_s : 0 \leq \tilde{U}i'_s \leq \tilde{u}'\}
\] (4.33)

From (4.31), \(\tilde{u}' \geq \tilde{u}\), implying that \(\tilde{F} \subseteq \hat{F}'\). The approach to compute \(\hat{F}'\) is presented in Algorithm 3. Equation (4.15) can now be restated as:

\[
\tilde{v}_{ub} = \left( I + G^{-1} \frac{C}{\Delta t} \right) \begin{bmatrix} I_{\text{ext}} & 0 & 0 \\ 0 & I_{\text{prt}} & 0 \\ 0 & T^T & A^{-1}_{\text{int}} \end{bmatrix} \begin{cases} \text{max} & \forall i'_s \in \hat{F}' \left( \hat{A}^{-1}_{\text{int}} |_{\eta i'_s} \right) \\ i_{L,\text{int}} \end{cases}
\] (4.34)

The incremental verification approach using the new feasible space \((\hat{F}')\) is given in Algorithm 4. To comment on the quality of results obtained by using the approach described above, we make the following claim:
Table 4.1: Speed and accuracy after reducing the dimensions of the feasible space of currents compared to incremental verification approach presented in chapter 3

<table>
<thead>
<tr>
<th>Name</th>
<th>Power Grid</th>
<th>Sub-grid</th>
<th>Max Error (mV)</th>
<th>Avg. Error</th>
<th>CPU time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n$</td>
<td>$n_{int}$</td>
<td>$n_{prt}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td>23,378</td>
<td>11,484</td>
<td>118</td>
<td>0.11</td>
<td>0.39 h.</td>
<td>0.28 h.</td>
</tr>
<tr>
<td>H2</td>
<td>36,277</td>
<td>18,616</td>
<td>148</td>
<td>0.06</td>
<td>0.5 h.</td>
<td>0.43 h.</td>
</tr>
<tr>
<td>H3</td>
<td>52,359</td>
<td>31,098</td>
<td>176</td>
<td>0.01</td>
<td>0.61 h.</td>
<td>0.52 h.</td>
</tr>
<tr>
<td>H4</td>
<td>72,692</td>
<td>47,450</td>
<td>348</td>
<td>0.05</td>
<td>0.69 h.</td>
<td>0.63 h.</td>
</tr>
<tr>
<td>H5</td>
<td>98,162</td>
<td>68,972</td>
<td>402</td>
<td>0.02</td>
<td>0.77 h.</td>
<td>0.68 h.</td>
</tr>
<tr>
<td>H6</td>
<td>143,561</td>
<td>85,600</td>
<td>440</td>
<td>0.03</td>
<td>0.97 h.</td>
<td>0.81 h.</td>
</tr>
<tr>
<td>H7</td>
<td>162,087</td>
<td>124,824</td>
<td>518</td>
<td>0.05</td>
<td>1.25 h.</td>
<td>1.1 h.</td>
</tr>
</tbody>
</table>

Claim 3. The feasible space of voltages generated by the modified current constraints described by $\tilde{F}'$ contains the original feasible space of voltages generated by the original current constraints.

Proof. Since $\tilde{F} \subseteq \tilde{F}'$, we have:

$$\max_{\forall i'_s \in \tilde{F}'} (\tilde{A}^{-1}|_{\eta_{i'_s}}) \geq \max_{\forall i'_s \in \tilde{F}} (\tilde{A}^{-1}|_{\eta_{i'_s}})$$

(4.35)

Recall that $E'$ is a superset of the extreme points of $\tilde{F}$ and the solution to an LP is found at an extreme point which implies that (4.24) can be written as:

$$\begin{bmatrix} \tilde{u} \\ \tilde{l} \end{bmatrix} = \text{eopt}(\tilde{U}_{i'_s}) \quad \forall i'_s \in \tilde{F}$$

(4.36)

From (4.23) and (4.36),

$$\forall i'_s \in \tilde{F} \Rightarrow \{ \tilde{l} \leq \tilde{U}_{i'_s} \leq \tilde{u} \}$$

$$\Rightarrow i'_s \in \tilde{F}$$

$$\Rightarrow \tilde{F} \subseteq \tilde{F}'$$

$$\Rightarrow \max_{\forall i'_s \in \tilde{F}'} (\tilde{A}^{-1}|_{\eta_{i'_s}}) \geq \max_{\forall i'_s \in \tilde{F}} (\tilde{A}^{-1}|_{\eta_{i'_s}})$$

(4.37)
From (4.35), (4.37) and that $G^{-1} \geq 0$, we can conclude that the vector of upper bounds given by (4.34) is element-wise greater than or equal to the vector given by (4.15). Also, from the equivalence relation between (4.3) and (4.15), it can be deduced that the feasible space of voltages generated by the modified current constraints described by $\tilde{\mathcal{F}}'$ contains the original feasible space of voltages generated by the original current constraints.

4.2.2 Experimental Results

To test this approach, we implemented Algorithm 3 and Algorithm 4 in C++. The grids generated were consistent with 1.1 V 65nm CMOS technology and the sub-grid nodes were identified by the user. Computations were done using a 2.6 GHz Linux machine with 24 GB of RAM. A SPAI tolerance value of $\delta_1 = 5\text{mV}$ was used to compute approximate inverse of $\tilde{\mathbf{A}}$. A lower value of tolerance $\delta_2 = 0.1\text{mV}$, conductance threshold value $\kappa = 5 \times 10^{-3}$, and time constant value $\tau_N = 5\text{ps}$ were used to macromodel the sub-grid using Algorithm 2.

A comparison between the speed and accuracy obtained after reducing the dimensions of the feasible space of currents (referred to as modified approach) and incremental verification approach [38] (referred to as original approach) is presented in Table 4.1. We report the maximum absolute error (in mV) and the average percentage error incurred while performing verification of external nodes using the modified approach as compared to verification using the original approach. The results show that we have been able to achieve additional speed-up (1.18x on average) while incurring negligible loss of accuracy. The speed-up obtained from the modified approach is due to the fact that the size of LPs has decreased because the internal current sources have been physically replaced by Norton equivalent current sources at port nodes.

On average, the size of LPs in terms of the number of variables was reduced by 78.8 % as compared to the original approach. Although the size of LPs got significantly reduced, we were not able to get speed-ups to the same scale because the contribution of LP solves in the CPU time for the original approach is just 18.3 %. Since the modified approach targets this 18.3 % chunk of runtime, the speed-up is not comparable to the amount of reduction in the size of the LPs. The contribution further gets reduced to only 9 % in the modified approach.
Table 4.2: Runtime breakdown

<table>
<thead>
<tr>
<th>Power Grid</th>
<th>Total Time</th>
<th>MACRO $\tilde{T}$</th>
<th>$\tilde{G}<em>{sub}$ and $\tilde{G}</em>{sub}$</th>
<th>$\tilde{\mathcal{F}}$</th>
<th>INCR_VERIFY $v_{int}$ Bound on</th>
<th>Bound on $v_{int}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\tilde{G}_{sub}$</td>
<td>$\tilde{G}_{sub}$</td>
<td></td>
<td>LP</td>
<td>SPAI</td>
</tr>
<tr>
<td>H1</td>
<td>0.28 h.</td>
<td>12.17 s.</td>
<td>0.36 min.</td>
<td>0.65 s.</td>
<td>1.83 min.</td>
<td>16.48 min.</td>
</tr>
<tr>
<td>H2</td>
<td>0.43 h.</td>
<td>14.23 s.</td>
<td>0.71 min.</td>
<td>0.63 s.</td>
<td>2.57 min.</td>
<td>22.7 min.</td>
</tr>
<tr>
<td>H3</td>
<td>0.52 h.</td>
<td>27.9 s.</td>
<td>1.17 min.</td>
<td>0.85 s.</td>
<td>3.24 min.</td>
<td>27.3 min.</td>
</tr>
<tr>
<td>H4</td>
<td>0.63 h.</td>
<td>36.7 s.</td>
<td>1.43 min.</td>
<td>0.83 s.</td>
<td>3.76 min.</td>
<td>31.2 min.</td>
</tr>
<tr>
<td>H5</td>
<td>0.68 h.</td>
<td>43.9 s</td>
<td>2.19 min.</td>
<td>0.93 s.</td>
<td>3.93 min.</td>
<td>34.3 min.</td>
</tr>
<tr>
<td>H6</td>
<td>0.81 h.</td>
<td>1.05 min.</td>
<td>3.18 min.</td>
<td>0.89 s.</td>
<td>4.35 min.</td>
<td>39.4 min.</td>
</tr>
<tr>
<td>H7</td>
<td>1.1 h.</td>
<td>1.44 min.</td>
<td>9.11 min.</td>
<td>1.23 s.</td>
<td>5.02 min.</td>
<td>49.3 min.</td>
</tr>
</tbody>
</table>

Figure 4.1: A bar graph showing the contributions of major procedures in runtime for both original (chapter 3) and the modified approaches
Figure 4.2: Relative Error Plot for H6 (modified approach)

as shown in Fig. 4.1. A similar analysis when done for the traditional full-chip verification flow showed that the LPs constitute only about 12% of the runtime. With the modified approach, we were able to obtain a total speed-up (full-chip vs modified incremental verification) of 10.2x on average. The runtime breakdown for the modified approach is presented in Table 4.2. It can be noted from the data presented in Table 4.2 that majority of the CPU time is consumed in the computation of approximate inverse using SPAI. The relative error plot for H6 grid in Fig 4.2 shows that the we have been able to bound the error to under 0.5mV for the modified approach.

4.3 Chip Power Model

In this section, we propose a framework for performing power integrity analysis for the off-chip interconnections in a vectorless verification context by adapting the problem of constructing the CPM, to the incremental verification framework. The model that we are going to use for this work is described in section 2.2.2. Verification of the package (external) nodes can be efficiently performed by macromodeling the on-chip network (sub-grid). The sub-grid in this case is special because all the current sources in the model are present inside (on internal nodes
of the sub-grid, meaning that there are no global constraints in which the current sources from inside the sub-grid are related to the current sources outside the sub-grid. We adapt our algorithm to allow for verification of external nodes in the RLC verification context.

### 4.3.1 Adapting to RLC

In the RLC case, upper and lower bounds on voltage drops for \( r \) time steps ahead of time are required. For \( r = 1 \) or \( t = \Delta t \), the upper and lower bounds on voltage drops can be computed from (2.29) as:

\[
&w_1 = \text{eopt} (D^{-1}i_s) \\
&(\forall i_s \in \mathcal{F}) \tag{4.38}
\]

Using (2.8), the grid equation of the RC sub-grid when the port nodes are shorted to ground is given by:

\[
G_{33}v'_{\text{int}}(t) + C_{\text{int}}i'_{\text{int}}(t) = i_{s,\text{int}}(t) \tag{4.39}
\]

The Norton equivalent current source vector at port nodes will be given by:

\[
i'(\Delta t) = T_{\Delta t}i_{s,\text{int}}(\Delta t) \tag{4.40}
\]

where \( T_{\Delta t} = -G_{23}A_{\text{int}}^{-1} \) is the transformation matrix that transforms the internal current sources to the Norton equivalent current sources at \( t = \Delta t \) as in (3.4). We can construct a new feasible space \( \tilde{F}_{\Delta t} \) at \( t = \Delta t \) using Algorithm 3, where the transformation that maps the original current source configuration to the modified configuration is given by:

\[
i'_s(\Delta t) = \left( \begin{array}{cc} \mathbf{I}_{\text{ext}} & 0 \\ 0 & \mathbf{I}_{\text{prt}} \\ 0 & T_{\Delta t} \end{array} \right) \left( \begin{array}{c} i_s(\Delta t) \end{array} \right) \bigg|_\eta
\]

\[
= (J_{\Delta t}i_s)|_\eta \tag{4.41}
\]

Thus, the upper and lower bounds on voltage drops at \( r = 1 \) for external and port nodes are given by:

\[
\begin{bmatrix}
\tilde{w}_{1,\text{ext}} \\
\tilde{w}_{1,\text{prt}}
\end{bmatrix} = \text{eopt} \begin{bmatrix}
\tilde{v}_{\text{ext}}(\Delta t) \\
\tilde{v}_{\text{prt}}(\Delta t)
\end{bmatrix} = \text{eopt} (\tilde{D}^{-1}|_{\eta'_s}) \tag{4.42}
\]

\[
\begin{bmatrix}
\tilde{w}_{1,\text{ext}} \\
\tilde{w}_{1,\text{prt}}
\end{bmatrix} = \text{eopt} \begin{bmatrix}
\tilde{v}_{\text{ext}}(\Delta t) \\
\tilde{v}_{\text{prt}}(\Delta t)
\end{bmatrix} = \text{eopt} (\tilde{D}^{-1}|_{\eta'_s}) \tag{4.42}
\]
where \( \tilde{D} \) is an \( \tilde{n} \times \tilde{n} \) reduced system matrix obtained after performing reduction of the on-chip \( RC \) interconnection network. To find an estimate of the upper and lower bounds on voltage drops for internal nodes, we use the approach similar to the efficient bounds computation approach described in section 3.2. From (3.2), we have:

\[
v_{int}(\Delta t) = A_{int}^{-1}i_{s,int}(\Delta t) - A_{int}^{-1}G_{23}^T v_{prt}(\Delta t)
\]

\[
\Rightarrow \begin{cases} 
\max_{\forall i_s \in F}(v_{int}(\Delta t)) \leq A_{int}^{-1}\max_{\forall i_s, i_{s,int} \in F}(i_{s,int}(\Delta t)) + T_{\Delta t}^T \max_{\forall i'_s \in F_{\Delta t}}(v_{prt}(\Delta t)) \\
\min_{\forall i_s \in F}(v_{int}(\Delta t)) \geq A_{int}^{-1}\min_{\forall i_s, i_{s,int} \in F}(i_{s,int}(\Delta t)) + T_{\Delta t}^T \min_{\forall i'_s \in F_{\Delta t}}(v_{prt}(\Delta t)) 
\end{cases} \tag{4.43}
\]

To the extent that \( \max_{\forall i_s \in F}(v_{prt}(\Delta t)) \approx \max_{\forall i'_s \in F_{\Delta t}}(\tilde{v}_{prt}(\Delta t)) \), (4.43) can also be approximately re-stated as:

\[
\Rightarrow \begin{cases} 
\max_{\forall i_s \in F}(v_{int}(\Delta t)) \leq A_{int}^{-1}\max_{\forall i_s, i_{s,int} \in F}(i_{s,int}(\Delta t)) + T_{\Delta t}^T \max_{\forall i'_s \in F_{\Delta t}}(\tilde{v}_{prt}(\Delta t)) \\
\min_{\forall i_s \in F}(v_{int}(\Delta t)) \geq A_{int}^{-1}\min_{\forall i_s, i_{s,int} \in F}(i_{s,int}(\Delta t)) + T_{\Delta t}^T \min_{\forall i'_s \in F_{\Delta t}}(\tilde{v}_{prt}(\Delta t)) 
\end{cases} \tag{4.44}
\]

Thus, the lower bounds and upper bounds on voltage drops at \( t = \Delta t \) can be computed by using (4.42) and (4.44). For any \( r \) or \( t = r \Delta t \), the bounds were expressed in (2.31) as:

\[
w_r = w_{r-1} + \text{eopt} \left[ (D^{-1}E)^{r-1}D^{-1}i_s \right] \\
= \text{eopt} \left[ v((r - 1)\Delta t) \right] + \text{eopt} \left[ (D^{-1}E)^{r-1}D^{-1}i_s \right] \tag{4.45}
\]

where

\[
v(r\Delta t) = \sum_{k=0}^{r-1}(D^{-1}E)^kD^{-1}i_s \tag{4.46}
\]

\[
v((r - 1)\Delta t) + (D^{-1}E)^{r-1}D^{-1}i_s \tag{4.47}
\]

This implies that to find the bounds for any \( r \), we need to iteratively perform an optimization operation on voltage drops (given by (4.46)) at \( t = \Delta t, \ldots, r \Delta t \) such that the optimization function is given by:

\[
(D^{-1}E)^{r-1}D^{-1}i_s \text{ at } t = r \Delta t \tag{4.48}
\]
Similarly, for the $RC$ system when the port nodes are shorted to ground (described by (4.39)), the voltage at time $t = r\Delta t$ is given in [8] as:

$$v'_{\text{int}}(r\Delta t) = v'_{\text{int}}((r - 1)\Delta t) + \left( A_{\text{int}}^{-1} C_{\text{int}} \right)^{r-1} (A_{\text{int}}^{-1}s_{\text{int}})$$

(4.49)

From the arguments above and using (3.9), the Norton equivalent current vector at port nodes at $t = r\Delta t$ is given by:

$$i'(r\Delta t) = -G_{23}v'_{\text{int}}(r\Delta t)$$

$$\Rightarrow T_{r\Delta t} = -G_{23} \left( A_{\text{int}}^{-1} C_{\text{int}} \right)^{r-1} A_{\text{int}}$$

(4.50)

The new feasible space ($\tilde{F}'_{r\Delta t}$) at $t = r\Delta t$ can now be defined using Algorithm 5, with the transformation matrix $J_{r\Delta t}$ being composed of $T_{r\Delta t}$. Using an approach similar to that was used to determine $\tilde{w}_1$ after macromodeling, the lower and upper bounds on voltage drops at time $t = r\Delta t$ after macromodeling can be expressed as:

$$\begin{bmatrix} \tilde{w}_{r,\text{ext}} \\ \tilde{w}_{r,\text{prt}} \end{bmatrix} = \begin{bmatrix} \tilde{w}_{r-1,\text{ext}} \\ \tilde{w}_{r-1,\text{prt}} \end{bmatrix} + \text{eopt } \left[ (\tilde{D}^{-1}\tilde{E})^{r-1}\tilde{D}^{-1} \right]_{\eta \epsilon \tilde{F}'_{r\Delta t}}$$

$$\Rightarrow \tilde{w}_{r,\text{int, max}} = \left( A_{\text{int}}^{-1} C_{\text{int}} \right)^{r-1} A_{\text{int}}^{-1} i_{L,\text{int}} + T_{r\Delta t}^{T} \tilde{w}_{r,\text{prt, max}}$$

$$\tilde{w}_{r,\text{int, min}} = T_{r\Delta t}^{T} \tilde{w}_{r,\text{prt, min}}$$

(4.51)

(4.52)

The choice of $r$ is made in such a way that $||\tilde{N}||_{\infty} < 1$ and $||\tilde{N}||_1 < 1$, where $\tilde{N} = (\tilde{D}^{-1}\tilde{E})^r$.

Recall that in (2.32), the bounds at infinity were given by:

$$\begin{bmatrix} v_{ub}(\infty) \\ v_{lb}(\infty) \end{bmatrix} = (I - R)^{-1} w_r$$

where $R$ is a $2n \times 2n$ matrix defined as:

$$R = \begin{bmatrix} N - S & S \\ S & N - S \end{bmatrix}$$

where $S = \frac{1}{2}(N - Q)$, with $Q$ being the matrix of the element-wise absolute values of the entries in $N$. We also have an $\tilde{n} \times \tilde{n}$ reduced order matrix $\tilde{N}$ such that:

$$\tilde{N}_{\eta} \approx N_{\eta}$$

(4.53)
Using the above approximation, we create an $\tilde{n} \times n$ matrix $N'$ such that:

$$N' = \tilde{N}J_{r\Delta t}$$  \hspace{1cm} (4.54)

and the bounds at infinity can now be computed as:

$$\begin{bmatrix} \tilde{v}_{ub}(\infty) \\ \tilde{v}_{lb}(\infty) \end{bmatrix} = (I - \tilde{R})^{-1}\tilde{w}_r$$  \hspace{1cm} (4.55)

where $\tilde{v}_{ub}(\infty)$ and $\tilde{v}_{lb}(\infty)$ are respectively, the $\tilde{n}$-upper and lower bounds on the worst-case voltage drops in the RLC case with the first $n_{ext}$ entries corresponding to the bounds on worst-case voltage drops on external nodes and $\tilde{R}$ is constructed in the same way using $N'$ instead of $N$.

It can be noted from (4.50) that computation of $T_{r\Delta t}$ for any $r$ requires one to compute $A_{int}^{-1}$. Although $T_{r\Delta t}$ can be efficiently computed by constructing the full matrix inverse using SPAI, we found it worth the effort to alleviate the need to explicitly compute the inverse by using an iterative approach. From (4.50), we have:

$$T_{(r-1)\Delta t} = -G_{23} \left( A_{int}^{-1} \frac{C_{int}}{\Delta t} \right)^{r-2} A_{int}^{-1}$$

$$\Rightarrow T_{(r-1)\Delta t} A_{int} = -G_{23} \left( A_{int}^{-1} \frac{C_{int}}{\Delta t} \right)^{r-2}$$  \hspace{1cm} (4.56)

Also, (4.50) can be written as:

$$T_{r\Delta t} = -G_{23} \left( A_{int}^{-1} \frac{C_{int}}{\Delta t} \right)^{r-2} \left( A_{int}^{-1} \frac{C_{int}}{\Delta t} \right) A_{int}^{-1}$$  \hspace{1cm} (4.57)

Using (4.56) in (4.57),

$$T_{r\Delta t} = T_{(r-1)\Delta t} A_{int} \left( A_{int}^{-1} \frac{C_{int}}{\Delta t} \right) A_{int}^{-1}$$

$$= T_{(r-1)\Delta t} \frac{C_{int}}{\Delta t} A_{int}^{-1}$$

$$\Rightarrow T_{r\Delta t} A_{int} = T_{(r-1)\Delta t} \frac{C_{int}}{\Delta t}$$  \hspace{1cm} (4.58)

Taking transpose on both sides of (4.58) and using the fact that $A_{int}$ is a symmetric matrix, we have:

$$A_{int} T_{r\Delta t}^T = \frac{C_{int}^T}{\Delta t} T_{(r-1)\Delta t}$$  \hspace{1cm} (4.59)
### Table 4.3: Speed and accuracy after constructing the CPM using Approach I

<table>
<thead>
<tr>
<th>Name</th>
<th>$n$</th>
<th>$n_{int}$</th>
<th>$n_{prt}$</th>
<th>$v_{ub}$</th>
<th>$v_{lb}$</th>
<th>$v_{ub}$</th>
<th>$v_{lb}$</th>
<th>CPU time</th>
<th>Speed</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>13,905</td>
<td>12,577</td>
<td>664</td>
<td>4</td>
<td>2.6</td>
<td>8.8</td>
<td>1.2</td>
<td>47.21 min</td>
<td>2.6 min</td>
</tr>
<tr>
<td>A2</td>
<td>24,548</td>
<td>22,208</td>
<td>1,170</td>
<td>5.5</td>
<td>4</td>
<td>10.2</td>
<td>5.1</td>
<td>2.41 h</td>
<td>10.8 min</td>
</tr>
<tr>
<td>A3</td>
<td>34,183</td>
<td>30,925</td>
<td>1,629</td>
<td>10.9</td>
<td>3.2</td>
<td>12.8</td>
<td>4.3</td>
<td>3.36 h</td>
<td>18.4 min</td>
</tr>
<tr>
<td>A4</td>
<td>52,968</td>
<td>47,920</td>
<td>2,524</td>
<td>11.19</td>
<td>3.2</td>
<td>12.9</td>
<td>2.7</td>
<td>6.27 h</td>
<td>31.8 min</td>
</tr>
</tbody>
</table>

### Table 4.4: Comparison between the three different approaches to macromodeling

<table>
<thead>
<tr>
<th>Power Grid</th>
<th>CPU time</th>
<th>Speed-up</th>
<th>Max Error I vs III (mV)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Approach I</td>
<td>Approach II</td>
<td>Approach III</td>
</tr>
<tr>
<td>A1</td>
<td>2.6 min</td>
<td>9.3 min</td>
<td>4.13 min</td>
</tr>
<tr>
<td>A2</td>
<td>10.8 min</td>
<td>41.2 min</td>
<td>12.2 min</td>
</tr>
<tr>
<td>A3</td>
<td>18.4 min</td>
<td>56.4 min</td>
<td>20.3 min</td>
</tr>
<tr>
<td>A4</td>
<td>31.8 min</td>
<td>51.3 min</td>
<td>35.1 min</td>
</tr>
</tbody>
</table>

A method to compute $T_{\Delta t}$ without constructing the full matrix $A_{int}^{-1}$ is presented in Algorithm 2. In this method, we first identify for every port node $k$, the neighbors and the connections ($g_{kj}$) to the neighbors. Then, we compute the corresponding row of the approximate inverse, multiply the row vector by $g_{kj}$, and add the result to the $k^{th}$ row of $T_{\Delta t}$. To compute $T_{2\Delta t}$, solve (4.59) for $T_{2\Delta t}$ using $T_{\Delta t}$ and then re-transpose to get $T_{2\Delta t}$. The same steps can be repeated iteratively to find $T_{r\Delta t}$. The complete algorithm to perform verification of the off-chip interconnects after constructing the CPM is presented in Algorithm 5.

### 4.3.2 Experimental Results

A C++ implementation was written to test the proposed approach. The grids generated were consistent with 1.1 V 65nm CMOS technology and the chip-to-package interconnections were modeled as tank circuits [3]. Inductance values were specified based on the technology specifi-
Chapter 4. Dimension Reduction of the Feasible Space of Currents

A SPAI error tolerance value $\delta_1 = 5\text{mV}$ was used to compute the approximate inverse ($\tilde{D}^{-1}$). To macromodel the on-chip $RC$ grid, a SPAI error tolerance value $\delta_2 = 0.1\text{mV}$, conductance threshold value $\kappa = 5 \times 10^{-3}$ and a nodal time constant value $\tau_N = 5\text{ps}$ were used. Three different approaches to macromodel the on-chip interconnects were implemented: 1) CPM is constructed using the dimension reduced feasible space and computation of $T_{r\Delta t}$ is done using (4.59), referred to as Approach I, 2) CPM is constructed using reduced dimensions while $T_{r\Delta t}$ is evaluated by constructing the full matrix $A^{-1}_{int}$ (Approach II), and 3) the original incremental verification approach presented in chapter 3 is adapted to the $RLC$ case to construct CPM and $T_{r\Delta t}$ is computed using the iterative approach (4.59) in Approach III.

Table 4.3 gives the speed and accuracy achieved by performing verification of the off-chip interconnects after constructing the CPM for the on-chip grid using Approach I. It can be observed that use of CPM results in significant speed-up, because verification can now be performed only on the nodes that need to be verified and optimizations are performed on a reduced problem (as a result of macromodeling). The error incurred is primarily due to the approximation involved with performing reduction of the sub-grid. This error can be reduced by tweaking the values of $\kappa$ and $\tau_N$ with some loss in the speed of verification. The relative error plots for the error incurred in the upper bound ($v_{ub}(\infty)$) and lower bound ($v_{lb}(\infty)$) for
A4 grid are presented in Fig. 4.3 and Fig. 4.4 respectively.

Table 4.4 provides representative data for comparison between the three approaches to construct CPM. It can be observed that Approach I leads to significant speed-up (3.01x on average) over Approach II in which the full inverse matrix was constructed. We were also able to achieve speed-up over Approach III without incurring significant error. The speed-up obtained was a result of the reduction in the size of the LPs in terms of the number of variables. The data reiterates the fact that using an iterative approach to compute $T_{r\Delta t}$ results in runtime savings and less memory penalty because the full inverse matrix need not be stored.
Algorithm 5 CPM_VERIFY

**Input:** Partitioned power grid matrices in (2.8), $\tau_N$, $\kappa$, $\delta_1$ and $\delta_2$

**Output:** Upper bound and lower bound estimates at infinity for external nodes

1: Construct subgrid matrices in (3.21)
2: $(T_{\Delta t}, \hat{G}_{sub}, \hat{C}_{sub}) = \text{MACRO}(\text{subgrid matrices}, \tau_N, \kappa, \delta_2)$
3: Construct $\hat{E}$ and $\hat{D}$
4: Compute $\hat{D}^{-1}$ using SPAI with $\delta = \delta_1$
5: $r = 1$
6: Construct $\hat{J}_{\Delta t}$ using (4.41)
7: $\tilde{F}_{\Delta t} = \text{compute\_new\_space}(U, \hat{J}_{\Delta t})$
8: Compute $\tilde{w}_1$ using (4.42) & (4.44)
9: $N = \hat{D}^{-1}\hat{E}$
10: while $(\min(\|N\|_\infty, \|N\|_1) \geq 1)$ do
11: \hspace{1em} $r = r + 1$
12: \hspace{1em} Compute $T_{r\Delta t}^T$ using (4.59)
13: \hspace{1em} $T_{r\Delta t} = (T_{r\Delta t}^T)^T$
14: \hspace{1em} $\tilde{F}_{r\Delta t} = \text{compute\_new\_space}(U, \hat{J}_{r\Delta t})$
15: \hspace{1em} Compute $\tilde{w}_r$ using (4.51) & (4.52)
16: \hspace{1em} $N = N \times (\hat{D}^{-1}\hat{E})$
17: end while
18: Compute upper and lower bounds using (4.55)
Chapter 5

Conclusions & Future Work

With technology scaling, voltage integrity analysis of power grids is becoming increasingly important. The main problem with simulation-based verification approaches is that the number of possible input vectors is very large that makes grid simulation not practical. Another drawback is that it does not allow the designer to verify the grid before the complete circuit has been designed that renders it difficult to make modifications in the power grid. As a result, an early vectorless verification approach based on the notion of current constraints is adopted in this work. The current constraints capture the uncertainty about the circuit behavior. Grid verification is thus reduced to a problem of finding the worst-case voltage drop over all possible currents that satisfy the constraints, implying that a solution in traditional vectorless verification requires solving a LP for every node.

Another trend that has emerged out of technology scaling is that the size of the power grids is becoming large that makes traditional flat (full-chip) verification an overkill. The need of the moment is to come up with efficient divide-and-conquer based approaches that can scale well to the increasing size of the power grids. One such approach is incremental verification that allows for efficient verification of only the sections of the grid that the user is interested in verifying. It also creates an avenue to analyze the local impact of changes made to the design of the power grid. In this work, we describe an early incremental verification approach for RC grids under a constraints-based power grid verification framework, in which only the nodes that
are external to a sub-grid region are to be verified. Our approach gives a fast and accurate way to compute the upper bounds on worst-case voltage drops at external nodes, based on two contributions: 1) an upper-bound method that eliminates the need to perform multiple iterations, and 2) a macromodeling method that drastically reduces the internals of the sub-grid. As a result, 3-8x speed-ups are obtained, with negligible 1-2% error. With this proposed approach, it becomes practical to perform early incremental design verification of the on-die power grid under dynamic conditions.

We have also extended this approach to efficiently perform vectorless constraints-based verification of off-chip $RLC$ interconnects by constructing the Chip Power Model for the on-chip interconnects. An optimized version of the incremental verification approach which involves construction of a new dimension reduced feasible space of currents is used while constructing the CPM. The results show that we have been able to achieve significant speed-ups without considerable loss of accuracy. Another point to be noted is that the loss of accuracy can be reduced by tweaking the parameters that guide the order reduction of the passive $RC$ sub-grid with some reduction in the speed-up.

Modern power grids are made up of many metal layers such that the higher metal layers provide connections to supply while lower metal layers provide supply to the underlying logic circuitry. As a future work, the incremental approach can be used to macromodel the higher metal layers to a point that makes it computationally effective to verify the lower metal layers without incurring much loss of accuracy. Similarly, incremental verification can be used to guide the placement of current sources in the power grid model that contribute to the worst-case voltage drop. Usually, the current sources are distributed uniformly over the entire block. The problem with such an approach is that it does not guarantee worst-case switching activity for the grid. Therefore, macromodeling can be used to abstract of the current sources to the block level, such that worst-case voltage behavior is ensured.
Bibliography


[40] The MOSEK optimization software (www.mosek.com).
