ON-CHIP POWER GRID VERIFICATION WITH REDUCED ORDER MODELING

by

Ankit Goyal

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

© Copyright by Ankit Goyal 2010
ON-CHIP POWER GRID VERIFICATION WITH REDUCED ORDER MODELING

Ankit Goyal

Master of Applied Science, 2010
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

To ensure the robustness of an integrated circuit design, its power distribution network (PDN) must be validated beforehand against any voltage drop on VDD nets. However, due to the increasing size of PDNs, it is becoming difficult to verify them in a reasonable amount of time. Lately, much work has been done to develop Model Order Reduction (MOR) techniques to reduce the size of power grids but their focus is more on simulation. In verification, we are concerned about the safety of nodes, including the ones which have been eliminated in the reduction process. This work proposes a novel approach to systematically reduce the power grid and accurately compute an upper bound on the voltage drops at power grid nodes which are retained. Furthermore, a criterion for the safety of nodes which are removed is established based on the safety of other nearby nodes and a user specified margin.
Acknowledgments

I would like to gratefully acknowledge the enthusiastic supervision of my advisor Professor Farid N. Najm for his continuous guidance, and support. His excellent supervision made the research problem look simpler and easier to approach. Every weekly meeting with him took me a step closer to cracking the problem. His intuition and the way he points one in the proper direction is simply amazing. I admire him for not rushing me at any stage of my study. “Don’t kill yourself for this” is what he said when meeting a deadline would have required me to lose sleep. I didn’t just gained knowledge from him but also learnt personality traits as how to deal with situations tactfully, maintain good work-life balance and much more. It was a privilege to have a supervisor like him. I also appreciate the valuable time and suggestions of my defense committee members: Prof. S. Brown, Prof. A. Veneris and Prof. T. J. Lim.

I would like to thank my fellow graduate students Meric, Mehmet, Khaled and Sari, for all the useful discussions we had, academic or otherwise. Working with all these people was a unique experience in itself and a very memorable one. Specially, I am grateful to Nahi for sharing his knowledge and expertise in the research field. At times, when I need a feedback or stuck badly on something, he is the first person I will turn to. They say “Life is like riding a bicycle. To keep your balance you must keep moving”. Now, when I am departing from this place and its people, I realize the fact: “Moving on is a simple thing, what it leaves behind is hard”.

I would also like to acknowledge financial support provided by the University of Toronto, the Semiconductor Research Corporation (SRC) and Advanced Micro Devices (AMD), Inc.

Finally, my parents, Sh. S. K. Goyal and Smt. Raj Goyal has always encouraged me and guided me to independence, wanting only the best for me. They never doubted my ability and invested in my future, at times, compromising their present. Without their support and wishes, I wouldn’t have reached this far. I am grateful to them for putting their trust in me.
# Contents

## List of Figures

- vi

## List of Tables

- vii

## 1 Introduction

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

## 2 Background

- 2.1 Introduction ......................................... 5
- 2.2 Model Order Reduction .............................. 5
- 2.3 Techniques .......................................... 6
  - 2.3.1 PRIMA: Passive Reduced-Order Interconnect Macromodelling Algorithm ........................................... 7
  - 2.3.2 DeMOR: Decentralized Model Order Reduction .............. 9
  - 2.3.3 TICER: TiMe-Constant Equilibration Reduction ............. 12
  - 2.3.4 Other Methods ..................................... 17
- 2.4 Power Grid Verification ............................. 19
  - 2.4.1 Power Grid Model .................................. 19
  - 2.4.2 Current Constraints ............................... 20
  - 2.4.3 Vectorless Verification ............................ 22

## 3 Power Grid Verification with Reduced Order Modeling

- 3.1 Introduction ........................................... 24
- 3.2 Node Safety Criteria ................................. 24
  - 3.2.1 Moving Current Sources ........................... 25
  - 3.2.2 Moving Capacitances .............................. 31
  - 3.2.3 Characterizing Safety ............................. 33
- 3.3 Grid Reduction ........................................ 34
  - 3.3.1 Computing Auxiliary Drops ........................ 35
  - 3.3.2 Sparsity Considerations ............................ 38
- 3.4 Algorithm ............................................. 39
- 3.5 Experimental Results ............................... 40
4 Extensions
  4.1 Introduction ................................................. 46
  4.2 Macro-model Generation ..................................... 46
  4.3 Incremental Verification .................................... 48

5 Conclusion and Future Directions 51

References 53
## List of Figures

1.1 Design trends .......................................................... 2
1.2 Star-mesh reduction scheme ........................................... 4

2.1 Graphical illustration of MOR ........................................... 6
2.2 RC network before reduction ......................................... 15
2.3 RC network after reduction ........................................... 16
2.4 Original RC power grid model ........................................ 19
2.5 Modified RC power grid model ..................................... 20
2.6 Local and global current constraints ................................ 21

3.1 A power grid node and its neighbors ................................ 26
3.2 Grid section after moving current source .......................... 27
3.3 Grid section after moving capacitor ................................. 31
3.4 Node elimination using star-mesh transformation .................. 33
3.5 Recursive grid reduction to generate safety regions ............... 36
3.6 Dealing with exponential complexity ................................ 37
3.7 Correlation plot between worst-case voltage drops on the retained nodes in the original grid and the corresponding nodes in the reduced grid . 41
3.8 Histogram of error in worst-case voltage drops on eliminated nodes . 42
3.9 Effect of varying conductance threshold, $\kappa$ on runtime and error . 42
3.10 Effect of region size on sparsity and reduction .................... 43
3.11 Reduction and runtime variations with $\tau$ .......................... 44
3.12 Effect of margin, $\gamma$ on reduction ............................... 45
3.13 Impact of $\epsilon$ on time requirements ............................... 45

4.1 Macro-modeled blocks with varying extent of reduction ........... 47
4.2 Region Vs. Macro-model ............................................. 48
4.3 Effect of varying conductance threshold, $\kappa$ on runtime and accuracy of incremental verification ............................... 50
# List of Tables

3.1 Results of verification performed on reduced and original power grid . 40

4.1 Incremental verification results . . . . . . . . . . . . . . . . . . . 49
1 Introduction

The powers of the mind are like the rays of the sun. When they are concentrated, they illumine

Swami Vivekanand

1.1 Motivation

In order to meet the growing demand for computational power, integrated circuits (IC) are getting faster and smaller. The number of transistors on an IC has increased exponentially in the past as per Moore’s law and is expected to grow in a similar fashion. As a result, the size of the network which supply power to millions of on-chip transistors has increased enormously. The problem is further complicated by the use of higher frequencies, which make transient effects more noticeable. Consequently, simple resistive models of the power delivery network, which were used a decade ago, are no longer valid. As a result, it is becoming increasingly difficult to perform analysis on a modern power grid, including all its details.

The reliability of an integrated circuit design depends heavily on its power grid. There is the potential for significant voltage fluctuations on the grid, resulting from high current densities, interconnect resistance and other grid parasitics. Such variations in the supply voltage on an integrated circuit may lead to increased circuit delays and loss of yield. This is crucial especially for modern power-conscious designs which operate on reduced supply voltages. Low supply voltages imply smaller noise margins and make the voltage drop across the power grid very critical. Higher cell placement densities further lead to more devices switching simultaneously, raising the demand for instantaneous current. Figure 1.1 illustrates logic density and supply voltage value projection based on the 2009 International Technology Roadmap for Semiconductors (ITRS) data. The trend indicates that the voltage profile is going...
to get worse in advanced technology nodes. In order to avoid these problems, one can over-design the grid by making the power rails wider or by adding decoupling capacitors, but this will be at the expense of silicon resources. Slowing down the circuit to account for increased timing delay is certainly not a viable option. Instead of adapting these highly pessimistic approaches, we can verify that the grid voltages remain within acceptable limits. For a safe operation, the voltage drop at any node must not exceed a certain threshold.

One way to check the power grid integrity is through simulation. This is typically done by simulating a model of the grid, based on the current loadings that are generated from prior simulations of the underlying transistor circuitry. However, this approach has several drawbacks. It is not feasible to do an exhaustive analysis, including all possible input vector traces and therefore, does not necessarily induce the worst-case behavior. Further, the approach cannot be applied for early power grid verification due to the lack of information about the underlying circuitry as it has not been designed yet.

Another option is to do vectorless verification, which does not need the detailed information of the input stimuli and can be applied in the early stages of design flow. It allows users to specify incomplete information about circuit currents in the form of “current constraints” and computes the worst-case voltage fluctuations at the grid nodes. However, as discussed, it is practically infeasible to perform the verification
on the entire power grid, modeled in full detail. For instance, in the simplest case of a resistive grid, the method in [KN03] requires solving a linear program (LP) for every grid node. For an RC grid, [NND05] formulates an optimization problem which requires solving a LP for every grid node at consecutive time steps. In both cases, the size of each LP is proportional to the grid size. Clearly, the problem complexity grows with the grid size and hence, imposes a limit on the size of grids that can be verified efficiently using such a vectorless approach. However, if we can reduce the grid size by eliminating some nodes, it will result in a reduced optimization problem with fewer and smaller LPs. This can be achieved by using an accurate compact model for the grid. Using this so-called compact model, grids containing millions of nodes can be verified efficiently, both for resistive and RC cases.

### 1.2 Objective

The goal of this research is to develop an efficient power grid verification technique using node elimination. A grid node is considered safe if its worst-case voltage drop is less than a given threshold that guarantees proper circuit operation. Model Order Reduction (MOR) [SVR08] gives a way to replace the original grid with a reduced one, with fewer nodes, while producing a good approximation of the input-output behavior. As the verification is performed on the reduced grid, the computation is relatively more efficient. The aim is to incur minimal error in the modeling process so that if the reduced model is found to be safe, the same can be assumed for the original network. However, this premise is not necessarily true and the original network may still be unsafe even though the reduced network is found to be safe. This is because, although the voltage drop at retained nodes is computed accurately, no definite information is available about the safety of nodes that are eliminated. Therefore, in order to keep the verification conservative, we must judge the safety of removed circuitry implicitly. The concept can be illustrated by the simple $Y-\Delta$ transformation, which is essentially a node reduction scheme. As shown in Figure 1.2, the electrical behavior of nodes 1, 2,..., n is preserved upon reduction and hence, can be verified accurately using the reduced model. However, as node N is eliminated and not verified explicitly, its safety must be concluded in some way, to guarantee the overall circuit reliability.

MOR methods developed so far such as those presented in [YXDCW08], [LCC05+$\ddagger$], and [ZPSB02] approximate the power grid by a smaller grid, which when simulated has
1 Introduction

Figure 1.2: Star-mesh reduction scheme

nearly the same output response as that of the original grid. These methods are not suitable for reducing a verification problem as they do not relate the safety of removed nodes in the original grid to those in the reduced grid. The key difficulty of our problem lies in characterizing the safety of nodes which have been eliminated during the reduction process. Therefore, in the case of verification, besides maintaining electrical equivalence, the method must also be able to infer the safety of the original grid from the verification results of the reduced grid. In this work, we devise a safety criteria for the eliminated nodes based on the worst-case voltage drops at the verified nodes in their proximity. Furthermore, the maximum voltage drop at the nodes which are retained and verified is computed with minimal error.

1.3 Thesis Organization

This thesis is organized as follows: Chapter 2 contains background information on Model Order Reduction (MOR), including an overview of some of the popular techniques. It also provides the background on the power grid model and the vectorless verification technique adapted in subsequent chapters. In Chapter 3, the criteria to establish the safety of an eliminated node is presented. It describes a node elimination process that gives a smaller grid with fewer nodes. It also includes the overall algorithm for the proposed approach along with the experimental results. Chapter 4 extends the proposed concept for macro-model generation and incremental verification. Finally, Chapter 5 concludes and suggests directions for future research.
2 Background

Honest differences are often a healthy sign of progress

Mahatma Gandhi

2.1 Introduction

Compact modeling of passive RC interconnect networks has been an intensive research area in the past decade owing to increasing signal integrity effects in modern system-on-a-chip (SoC) designs. In this chapter, we explain the concept of model order reduction (MOR) and briefly review some of the existing techniques. Specifically, we look at the projection-based methods, methods for linear circuits with multiple ports and the node-reduction based methods. In the later half of the chapter, we provide some background on the power grid verification. We discuss an RC power grid model that accurately captures the on-chip power grid behavior and then, derive from it a modified model to be used in subsequent analysis. In order to perform power grid verification in early stages of the design process, when detailed information about the currents drawn by the transistors is not known, one must use vectorless methods. One such method, which captures circuit uncertainty in the form of current constraints is discussed here. Finally, the constraint-based techniques to compute an upper bound on the worst-case voltage drop at various grid nodes is presented.

2.2 Model Order Reduction

Model order reduction (MOR) is a branch of systems and control theory, which studies the properties of dynamical systems in application for reducing their complexity, while preserving their behavior as much as possible [SVR08]. It transforms a circuit into one
of much smaller size whose behavior is an approximation of the original description. Such transformation on power delivery network (PDN) will allow us to perform its verification within a reasonable amount of time and with good accuracy.

The concept of MOR is illustrated graphically in Figure 2.1 (Graphics credits: “Stanford Bunny”—Harvard University, Microsoft Research). The figure demonstrates that sometimes little information is needed to describe a model. In these pictures, the rabbit can be recognized even with only a few facets.

![Figure 2.1: Graphical illustration of MOR](image)

2.3 Techniques

The origin of MOR dates back to the early eighties of the last century. Since then, much work has been done in this area. Consequently, a number of reduction techniques are available. As a result, it is not possible to include an exhaustive background of all the available techniques. Instead, this section gives an overview of some of the popular techniques. Their limitations and new methods improving upon those are also discussed.
2 Background

2.3.1 PRIMA: Passive Reduced-Order Interconnect Macromodelling Algorithm

Background

According to modified nodal analysis (MNA), a lumped, linear, time-invariant circuit can be described by the following system of first-order differential equations:

\[ C \dot{x}(t) = -Gx(t) + Bu(t) \]  
\[ y(t) = L^T x(t) \]

where the vector \( x(t) \in \mathbb{R}^n \) represents the circuit variables, the matrix \( G \in \mathbb{R}^{n \times n} \) represents the contribution of resistors (conductance matrix), \( C \in \mathbb{R}^{n \times n} \) represents contribution from capacitors and inductors (susceptance matrix), \( L \in \mathbb{R}^{n \times q} \) is the output matrix, \( B \in \mathbb{R}^{n \times p} \) is the input matrix, \( y \in \mathbb{R}^q \) is the output, and \( u \in \mathbb{R}^p \) represents input sources.

Typically, we have \( p \ll n \). Model order reduction algorithms aim to produce a smaller system:

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

where \( \hat{C}, \hat{G} \in \mathbb{R}^{r \times r}, \hat{B} \in \mathbb{R}^{r \times p} \). Order \( r \) is much smaller than the original order \( n \), i.e. \( r \ll n \), but the output \( y(t) \) and \( \hat{y}(t) \) are approximately equal.

This is achieved by constructing a matrix \( V \) whose columns span a useful subspace, and projecting the original system of equations onto the column space of \( V \):

\[ \hat{C} = V^T CV, \hat{G} = V^T GV, \hat{B} = V^T B, \hat{L} = V^T L \]

The above equation (2.5) is sometimes referred to as congruence transforms. The following transfer functions are generally used as a metric for approximation:

\[ H(s) = L^T(sC + G)^{-1}B \]  
\[ \hat{H}(s) = \hat{L}^T(s\hat{C} + \hat{G})^{-1}\hat{B} \]

If \( \|H(s) - \hat{H}(s)\| \leq \epsilon \), in some appropriate norm, for some given allowable error \( \epsilon \) and allowed domain of complex frequency \( s \), the reduced model is accepted as accurate.
This class of reduction methods are commonly known as projection-based methods. Usually, the projection matrix $V$ is constructed so that its columns span a Krylov subspace. For $n \times n$ matrix $A$ and a vector $R$, the Krylov subspace is defined as:

$$K_m(A, R) = \text{span}\{R, AR, A^2R, \cdots, A^{m-1}R\}$$  \hspace{1cm} (2.8)

where $m$ is a given positive integer. Recall that the span of a set of vectors $U = \{u_1, u_2, \cdots, u_n\}$ is the set of all linear combinations of these vectors.

$$\text{span}\{U\} = \text{span}\{u_1, u_2, \cdots, u_n\} = \{x | x = \sum_{i=1}^{n} \alpha_i u_i\}$$  \hspace{1cm} (2.9)

where $\alpha_i$ are real numbers. Then, for an $m^{th}$ order approximation, $V$ is such that:

$$\text{colsp}(V) = K_m(A, R)$$  \hspace{1cm} (2.10)

Once we have this Krylov subspace, we have the transformation matrix $V$ needed for applying the transformations to MNA matrices. Hence, the only question remains unanswered is how to generate matrix $A$ and matrix $R$ to get the Krylov subspace. This is where PRIMA comes in.

**PRIMA**

In PRIMA [OCP98], the transformation matrix $V$ is constructed by using the Arnoldi algorithm [Bol94], thereby spanning a Krylov subspace with:

$$A = (G + s_0C)^{-1}C, R = (G + s_0C)^{-1}B$$  \hspace{1cm} (2.11)

The idea is to project the system onto the Krylov subspace to match dominant block moments. The block moments of $H(s)$ are defined as the coefficients of the Taylor expansion of $H(s)$ around $s_0$ (usually $s_0 = 0$):

$$H(s) = M_0 + M_1(s - s_0) + M_2(s - s_0)^2 + \cdots$$  \hspace{1cm} (2.12)
where $M_i \in \mathbb{R}^{r \times p}$ and can be computed as $M_i = L^T A^i R$. If the order of approximation is $m$, then PRIMA matches at least $\left\lfloor \frac{m}{p} \right\rfloor$ block moments at $p$ ports. Therefore, the reduced transfer function $\hat{H}(s)$ and the original $H(s)$ agree up to the first $k$ derivatives on an expansion around some chosen point in the complex plane ($s_0$).

$$H(s) = \hat{H}(s) + O((s - s_0)^k)$$

(2.13)

where $k = \left\lfloor \frac{m}{p} \right\rfloor$. Also, PRIMA guarantees the passivity of reduced system if the original system is in MNA form with $L = B$ where $B \in \mathbb{R}^{n \times N}$ and $N$ is the number of terminals (as input and output ports).

**Limitations**

[OCP98] does not clearly specify what order of approximation ($m$) to use. This limitation of PRIMA was solved by the same authors after one year [OCP99]. They proposed a convergence criteria based on an error measure for intelligent order selection. Also, PRIMA does not preserve certain important circuit properties such as reciprocity. Another limitation is that for every block-moment increase, PRIMA will generate $N$ new poles in the reduced models, where $N$ is the terminal count. As a result, the model size is proportional to the number of moments matched multiplied by the number of ports. For example, if only two (block) moments are to be matched at each port, and the network has 1000 ports, the resulting reduced model will have 2000 states, and the reduced system matrices will be dense. This makes PRIMA unsuitable for networks with a large number of input/output ports. Due to this problem, much work [LTM07], [FL04] has been done on terminal reduction. Note that terminal reduction is done before PRIMA is applied. Furthermore, PRIMA constructs projection matrix $V$ explicitly. However, in [VYZP08], it is argued that we actually do not need to compute it explicitly thus saving considerable memory and CPU time.

### 2.3.2 DeMOR: Decentralized Model Order Reduction

**Background**

SVDMOR [Fel04] algorithm was developed to address the reduction of systems with a large number of ports, like power grids. It takes advantage of situations when the matrix-transfer function is numerically close to being low rank and generate for it a
compact and sparse reduced order model. Recall, for an \( m \times n \) matrix \( A \), the singular value decomposition (SVD) of \( A \) is

\[
A = U_{m \times m} \Sigma V_{n \times n}^T
\]

where \( U_{m \times m} \) and \( V_{n \times n} \) are orthogonal matrices, \( U_{m \times m}^T U_{m \times m} = I \) and \( V_{n \times n}^T V_{n \times n} = I \). \( \Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min(m,n)}) \), \( \sigma_i \) are the singular values of \( A \) defined by: \( \sigma_i = \sqrt{\lambda_i(A^T A)} \) and \( \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_{\min(m,n)} \). Let, as in previous section, the circuit transfer function be:

\[
H(s) = L^T(sC + G)^{-1}B
\]

Applying SVD to the system transfer function at DC produces a diagonal matrix \( S \) of the same dimension as \( H \), with nonnegative diagonal elements in decreasing order, and orthogonal matrices \( U \) and \( V \) so that

\[
H_{DC} = L^T G^{-1} B = U S V^T
\]

The terms in \( S \) which are below a certain threshold are dropped to give dominant left and right singular vectors \( U_k \) and \( V_k \), \( k < p \). These singular vectors are then used to find rank-\( k \) approximations for \( L \) and \( B \):

\[
B \approx b_k V_k^T \quad L \approx l_k U_k^T
\]

where \( b_k \in \mathbb{R}^{n \times k} \) and \( l_k \in \mathbb{R}^{n \times k} \) are obtained using the Moore-Penrose pseudo-inverse of \( V_k \).

\[
b_k = B V_k (V_k^T V_k)^{-1}
\]

\[
l_k = L^T U_k (U_k^T U_k)^{-1}
\]

The circuit transfer function now becomes:

\[
H(s) \approx U_k l_k^T (sC + G)^{-1} b_k V_k^T
\]

Since \( H_k(s) \approx l_k^T (sC + G)^{-1} b_k \) represents a terminal-reduced network with \( k(k < p) \) ports, it can be easily reduced by any existing method. But the problem here is, a complete matrix transfer function of a sub-circuit is rarely low rank. However, it may have low rank sub-blocks, which is basically the idea of RecMOR discussed next. SVDMOR does not produce passive models in general and passivity enforcement in
SVDMOR will significantly hamper the terminal-reduction effectiveness.

Later, the same two researchers from IBM proposed RecMOR [FL04], an algorithm for the computation of reduced order models of structured linear circuits with numerous I/O ports. The algorithm exploits certain regularities of the sub-circuit response, so that the normally dense matrix transfer function of the sub-circuit contains sub-blocks that are low rank and uses SVDMOR to reduce them.

DeMOR

In [YZT08], a decentralized model order reduction scheme is proposed, where a whole multi-input-multi-output (MIMO) circuit is decoupled into a number of multi-input-single-output (MISO) circuits based on the input-output interactions and each circuit is reduced individually. The authors argue that the interactions between each input-output pair is not the same and most of them are insignificant. The decoupling process is based on the Relative Gain Array (RGA), which measures the degree of interaction between each input output pair.

Let $H(s) = L^T(Cs + G)^{-1}B$ be the transfer matrix of a linear time invariant (LTI) system. In the ideal decentralized case, $H(s)$ will be diagonal such that no interaction exists and the system is decoupled into a number of independent subsystems. However, it is rarely the case that each output interacts equally with all inputs. RGA gives a measure of these interactions and can be computed using the following method:

$$\Lambda(H) = H(0) o H(0)^{-T}$$

(2.21)

where $o$ denotes element-by-element multiplication (also called Hadamard or Schur product) and $H(0) = L^T G^{-1} B$ is the steady state DC gain. The values in RGA matrix are scaled to fall in the range of $[0, 1]$. If the interaction between $i^{th}$ output and $j^{th}$ input is insignificant, then the scaled RGA value

$$\Lambda_{ij} \approx 0$$

(2.22)

If $\Lambda_{ij} < \epsilon$, then input $j$ can be considered irrelevant for output $i$ and we end up with a decentralized system in terms of the $i^{th}$ output which can be solved using standard projection based methods (PRIMA). Experimental results on a number of interconnect circuits show that most of the input-output interactions are usually insignificant, which can lead to extremely compact models even for systems with
massive number of ports.

**Improvements**

The method in [YXDCW08] exploits locality in the sense that two geometrically far nodes have very small impact on each other. This is because, when the power grid is modeled as an RC network, which is essentially a low pass filter, the input signal is attenuated in proportion to the exponent of the distance. Therefore, the work assumes that the voltage response at a node is dominated by only few inputs which are nearby. These dominant inputs are identified based on Relative Gain Array (RGA) which gives a measure of the interaction between each input and the node we are interested in. Note that the concept of RGA was originally introduced to measure the interaction between an input-output pair in DeMOR. After identifying these inputs, the system is projected on a Krylov space corresponding to the dominant inputs only. Results show good accuracy over PRIMA and better speed. However, the method has two limitations. First, it assumes that the number of input and output ports are approximately equal, which is not necessarily true. Secondly, the implementation is valid if we are interested in observing few nodes, not all; as in that case the computation of RGA becomes very expensive.

### 2.3.3 TICER: TIme-Constant Equilibration Reduction

**Theory**

In 2007, Sheehan [She07] proposed TICER algorithm which reduces circuit complexity by means of node elimination. It can be considered as an approximate Gaussian elimination technique for RC circuits that keeps only the first two moments. The main idea is to remove nodes which have few neighbors and small nodal time constants. The time constant of a node is defined as the ratio of the sum of capacitances and the sum of conductances connected to it. In order to qualify for elimination, the node must have a small time constant, \( \tau \) such that \( |s\tau| \ll 1 \) and if so, the node is said to be a *quick node*. Once a node is removed, the behavior of the circuit is preserved by adjusting the components around this node. Note that the branch and circuit elements connected to the node that is removed, are also removed.
For an RC circuit, we have the following circuit equation in Laplace domain.

\[ Y(s)v = (G + sC)v = J \]  \hspace{1cm} (2.23)

where \( Y(s) \) is \( n \times n \) admittance matrix, \( v \) is the node voltage vector of length \( n \), \( G \) is the \( n \times n \) conductance matrix, \( C \) is the \( n \times n \) capacitance matrix and \( J \) is the \( n \)-dimensional vector of current sources. Suppose that the \( n^{th} \) node is eliminated. Then, the matrix is partitioned in such a way that the \( (n,n) \) entry forms one part, as in the following equation.

\[
\begin{bmatrix}
\hat{Y} & y \\
y^T & G_n + sC_n
\end{bmatrix}
\begin{bmatrix}
\hat{v} \\
v_n
\end{bmatrix} =
\begin{bmatrix}
\hat{J} \\
j_n
\end{bmatrix}
\]

where \( y \) captures the interconnections to the node we plan to eliminate, \( \hat{Y} \) contains the admittance information about the remaining sub-circuit, \( G_n \) is the sum of conductances and \( C_n \) is the sum of capacitances connected to node \( n \). \( v_n \) is the voltage and \( j_n \) is the value of current source at node \( n \). We do not need to know the detailed structure of the matrices involved in the above equation, since they do not participate directly in the elimination process. Then, the variable \( v_n \) is eliminated, which leads to:

\[ (\hat{Y} - E)\hat{v} = (\hat{J} - F) \]  \hspace{1cm} (2.24)

with

\[
E_{ij} = \frac{y_i y_j}{G_n + sC_n} = \frac{(g_{in} + s c_{in})(g_{jn} + s c_{jn})}{G_n + sC_n} = \frac{g_{in} g_{jn}}{G_n + sC_n} + \frac{g_{in} c_{jn} + c_{in} g_{jn}}{G_n + sC_n} s + O(s^2) = \frac{g_{in} g_{jn}}{G_n} \left(1 + \frac{sC_n}{G_n}\right)^{-1} + \frac{g_{in} c_{jn} + c_{in} g_{jn}}{G_n} s \left(1 + \frac{sC_n}{G_n}\right)^{-1} + O(s^2)
\]  \hspace{1cm} (2.25)

After this elimination process, the matrix is not in the form \( G + sC \) anymore, but is a fraction of polynomials in \( s \). To get an RC-circuit representation, an approximation
is needed. Using Taylor expansion and dropping the higher order terms in $s$ gives:

$$E_{ij} \approx \frac{g_{in} g_{jn}}{G_n} (1 - \frac{sC_n}{G_n}) + \frac{g_{in} c_{jn} + c_{in} g_{jn}}{G_n} s (1 - \frac{sC_n}{G_n})$$  \hspace{1cm} (2.26)

According to the quick node approximation made earlier, node $n$ has a small time constant, $\tau_n = \frac{G_n}{C_n}$ and hence, $|s\tau| \ll 1$. Therefore, the $\frac{sC_n}{G_n}$ term in above equation can be neglected in comparison to 1, resulting in:

$$E_{ij} \approx \frac{g_{in} g_{jn}}{G_n} + \frac{g_{in} c_{jn} + c_{in} g_{jn}}{G_n} s$$  \hspace{1cm} (2.27)

Performing similar approximations on $F_i$ will give:

$$F_i = \frac{y_i}{G_n + sC_n} j_n = \frac{g_{in} + sc_{in}}{G_n + sC_n} j_n \approx \frac{g_{in}}{G_n} j_n$$  \hspace{1cm} (2.28)

In summary, the guidelines to remove a quick node $(n)$ and translating into the equivalent modified circuit are:

1. Remove all resistors and capacitors connecting quick node to other nodes

2. Insert new resistors and capacitors between the former neighbors of $n$ according to the following two rules:

   - If nodes $i$ and $j$ had been connected to $n$ through conductances $g_{in}$ and $g_{jn}$, insert a conductance $\frac{g_{in} g_{jn}}{G_n}$ between $i$ and $j$
   - If node $i$ had a capacitor $c_{in}$ to $n$ and node $j$ had a conductance $g_{jn}$ to $n$, insert a capacitor $\frac{c_{in} g_{jn}}{G_n}$ between $i$ and $j$

Note that the algorithm does not consider the case when both nodes $i$ and $j$ connects $n$ through capacitances. But still this is by far considered a very good reduction method as it bridges the gap between circuit level and matrix level reduction techniques. We will try to explain TICER with the help of an example.
Figure 2.2: RC network before reduction

An Example

Figure 2.2 depicts an RC network and we are interested in eliminating node \( n \). As shown, node \( n \) connects to four other nodes through conductances \( (g_i) \) \( \forall i = 1, 2, 3, 4 \) and ground via capacitance \( (c_n) \). Let there be a current source of magnitude \( I_n \) at node \( n \).

Now, we remove node \( n \) as per TICER rules. This will result in a reduced network as shown in Figure 2.3, where:

\[
\begin{align*}
g_{ij} &= \frac{g_i g_j}{\sum_{k=1}^{4} g_k} \\
c'_i &= c_i + \frac{g_i}{\sum_{k=1}^{4} g_k} c_n
\end{align*}
\]

and:

\[
I'_i = I_i + \frac{g_i}{\sum_{k=1}^{4} g_k} I_n \quad \forall i = 1, 2, 3, 4
\]

Limitations

If we neglect the effect of capacitors and model the power grid as a resistive network, then TICER is exact, in sense that the reduced circuit is electrically equivalent to original one. Note that TICER does not take into account inductive effects and hence,
applies to the reduction of RC circuits only. In case of RC power grids, the TICER approximation is not exact and occasionally leads to large errors as it destroys the moment matching property by dropping the negative capacitances resulting from Taylor expansion. Also, as mentioned earlier, a good reduction technique must maintain the sparsity of system matrices. However, TICER lacks this characteristic and ends up having more non-zero entries in the reduced conductance matrix. When a node with \( c \) connections is eliminated those connections disappear but \( \frac{c(c-1)}{2} \) new connections appear. This is because, whenever we remove a node, we add conductances between every pair of its original neighbors. So, with TICER, the number of connections grow quadratically while the number of nodes decreases linearly.

In TICER, the voltage at an eliminated node is computed by the following expression.

\[
v_n = \frac{\sum_{i=1}^{k} v_i g_i}{\sum_{i=1}^{k} g_i}
\]

(2.32)

where \( v_i \) is the voltage at node \( i \) and \( k \) is the number of immediate neighbors of node \( n \). Although this expression holds for simulation, it can not be used to compute worst-case voltage drop at an eliminated node by replacing the exact voltages with
max voltage drop at its neighbors as:

\[ v'_{ndrop} = \frac{\sum_{i=1}^{k} v'_{idrop} g_i}{\sum_{i=1}^{k} g_i} \]  

(2.33)

where \( v'_{idrop} \) is the worst-case voltage drop at node \( i \). This is because it will result in huge overestimation of the worst-case voltage drop at node \( n \) as all the neighbors are simultaneously held to worst-case drops, which is practically rare.

There is an earlier work by the same author [She99] which is basically a projection based reduction method. But interestingly, the MNA matrices are manipulated in a way such that they become symmetric and positive definite, in which case, efficient Cholesky factorization can be used to solve the reduced circuit. [ACI03] extends TICER to include inductances also. In this work, two time constants are associated with a node. The node is eliminated according to a scheme that depends on the dominant time constant. However, the method is based on the simplification that a node is connected to other nodes through an RLC network with at least one of the three elements (R, L or C) zero. Further, it is stated that if the two time constants are comparable, then the reduction scheme will introduce a large error.

### 2.3.4 Other Methods

So far, we have reviewed a number of important MOR methods that can be used to generate compact PDN models. However, the discussion is certainly not exhaustive and a number of alternative techniques exists (refer [TH07] for a detailed study). For example, a method for the hierarchical analysis of PDNs is presented in [ZPSB02]. It is essentially a “divide and conquer” strategy which involves partitioning of the power grid into local and global grids. The global grid is simulated after augmenting with the macro-models of the local grid. The key idea employed in the partitioning strategy is to identify a subnetwork and an interface boundary such that the number of internal nodes is much larger than the square of the number of nodes at the interface, thus preserving sparsity. A method to reduce large resistor networks to equivalent but smaller resistor networks is proposed in [RS10]. Instead of eliminating all internal nodes, which will lead to a dense matrix, the method identifies strongly connected components in the whole network and simplifies them individually. However, the method relies significantly on the approximate minimum degree (AMD) algorithm and therefore, is not suitable for reducing resistive PDNs which have almost uniform
structure. Furthermore, the method does not take into account capacitive effects and can only be employed in the case of DC analysis.

Another class of reduction methods is based on multi-grid principles. In general, a multi-grid method has two steps: Relaxation, which involves running few iterations of an iterative solver to reduce the high frequency error component. The next step is “coarse grid correction”, in which the grid is reduced to a coarser structure (Restriction), solved and the solution is mapped back to the original grid (Interpolation). However, the method in [KNN01] requires keeping track of the power grid geometry for each multi-grid level and hence, makes it impractical for real designs.

Recently, some new techniques for model order reduction of large scale linear systems has been proposed. The method in [PYPA09] employs heuristic optimization techniques such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), for finding stable reduced order models of single-input-single-output (SISO) large scale linear systems. However, besides being non-applicable to multi-input-multi-output (MIMO) systems, the speed of convergence depends on the choice of parameters. Moreover, these heuristic optimization methods may not find the global optimum.

In this chapter, we have seen a number of techniques including projection based (PRIMA) and Gaussian elimination based (TICER). Although, no technique offers clear advantage over others, TICER technique appears more appropriate for reduction of RC circuits for a number of reasons. One of its strong points is the ability of “on-the-fly” reduction which drastically reduces the amount of circuit data that needs to be stored. Also, in contrast to the projection based methods, TICER does not have problems with networks with many ports. In TICER, realizability is not a problem as it relates the representation of the reduced circuit in the form of another, smaller, circuit rather than in the abstract form of merely one or more matrices that would describe the behavior of reduced model but are not compatible for existing circuit simulators. For these reasons, our approach focuses on TICER as the reduction scheme.
2.4 Power Grid Verification

2.4.1 Power Grid Model

We consider an RC model of the power grid which has resistive connections between nodes and a capacitor from every node to ground, as shown in Figure 2.4. Some of the grid nodes have ideal current sources (to ground) representing the currents drawn by the circuits tied to the grid at those nodes. Further, some nodes have ideal voltage sources (to ground) which represent connections to the external supply voltage. Let the power grid consist of \(n+p\) nodes, where nodes 1, 2, \ldots, \(n\) have no voltage sources attached and the remaining nodes \((n+1), (n+2), \ldots, (n+p)\) are connected to voltage sources. Let \(c_k\) be the capacitance from node \(k\) to ground and \(i_k(t)\) be the value of current source there, such that direction of positive current is assumed to be from node to ground. We assume that \(i_k(t) \geq 0\) is defined for all nodes, \(k = 1, \ldots, n\), so that nodes with no current sources have \(i_k(t) = 0, \forall t\). Let \(u_k(t)\) be the voltage at node \(k, k = 1, \ldots, n\). Let \(i(t)\) be the vector of all \(i_k(t)\) current sources and \(u(t)\) be the vector of all \(u_k(t)\) voltage signals. Applying Modified Nodal Analysis (MNA), based on KCL at nodes 1, 2, \ldots, \(n\), we have:

\[
Gu(t) + C\dot{u}(t) = -i(t) + GV_{dd} \quad (2.34)
\]
2 Background

Figure 2.5: Modified RC power grid model

where \( G \) is the \( n \times n \) conductance matrix of the grid, \( C \) is an \( n \times n \) diagonal matrix containing node capacitances, and \( V_{dd} \) is a vector with all entries equal to the supply voltage, \( v_{dd} \). Let \( v_k(t) = v_{dd} - u_k(t) \) represent the voltage drop at node \( k \), and let \( v(t) \) be the vector of all voltage drops, then equation (2.34) can be re-written as [KN03]:

\[
Gv(t) + C\dot{v}(t) = i(t) \tag{2.35}
\]

This system is, in fact the result of writing standard MNA for a modified grid, which can be readily obtained from the original power grid by removing all its voltage sources and reversing the direction of all current sources. The modified RC power grid model is shown in Figure 2.5. Note that the vector \( v(t) \) represents the node voltages in the modified network. For a purely resistive network, equation (2.35) reduces to:

\[
Gv = i \tag{2.36}
\]

2.4.2 Current Constraints

In order to capture the uncertainty about the circuit details and circuit behavior in early verification of the grid, we will use the concept of current constraints [KN03]. There are two types of constraints, local constraints and global constraints. Local constraints are upper bounds on individual current sources and represent the maximum
2 Background

Figure 2.6: Local and global current constraints

current drawn by individual cells or blocks. They can be represented as:

\[ 0 \leq Li(t) \leq I_L, \quad \forall t \geq 0 \]

where \( I_L \) is a vector of fixed current values and \( L \) is an \( n \times n \) identity matrix. If a grid node does not have a current source attached to it, i.e. if \( i_k(t) = 0, \forall t \geq 0 \), then the corresponding upper bound value is zero, \( I_{L,k} = 0 \).

However, local constraints alone are insufficient, as the chip components do not draw their maximum currents simultaneously. Global constraints are introduced to provide an upper bound on the sum of currents drawn by groups of current sources. They are typically chosen based on some knowledge of the peak power dissipation of a block or a chip, and can be expressed in matrix form as:

\[ 0 \leq Ui(t) \leq I_G, \quad \forall t \geq 0 \]

where \( U \) is a matrix of dimension \( m \times n \) such that \( U_{ij} = 1 \) if the current source at node \( j \) is included in \( i^{th} \) constraint, and 0 otherwise. \( m \) is the number of global constraints.
$I_G$ is a vector of size $m$ and $I_{G,i}$ is the upper bound on the sum of currents included in the $i^{th}$ constraint. The concept is illustrated graphically in Figure 2.6. Note that, although the RC model features dynamic currents and voltages, the values of local and global constraints are static, i.e., $I_L$ and $I_G$ are fixed and do not vary with on time.

In practice, these constraints can be specified depending on the size and availability of the underlying logic block. If it is available and small enough to simulate, then one can generate the constraints by actually simulating the block. If the block is not yet available or is too large to simulate, then one needs to rely on design expertise and engineering judgment. An estimate of the size of the block, its power requirements in previous technology, scaling are few such factors that can guide an engineer. If in the early design stages, no information is available about the block, its projected power budget can be used to generate a rough current constraint. The bottom-line is that something is typically known about a block, which with good design knowledge can be formulated into constraints.

2.4.3 Vectorless Verification

The exact solution to the verification problem is expensive as it involves solving one LP for every node on the grid at multiple time steps [NND05]. Moreover, the number of constraints increases at each time step, leading to a potentially very large optimization problem. Instead, one can use an efficient method presented in [FNK07] to compute upper bounds on the maximum voltage drops and is briefly described here for completeness. Let $A = G + C/\Delta t$. According to [FNK07], a vector of upper bounds, on all node voltages $V_{ub}$ can be written as:

$$V_{ub} = \left( I + \frac{G^{-1}C}{\Delta t} \right) V_a = G^{-1}AV_a$$

where $I$ is the identity matrix, $\Delta t$ is the time step and $V_a$ is obtained by solving the following LP for every grid node, $k = 1$ to $n$.

$$V_{a,k} = \max v_k$$
$$\text{s.t. } SAv \leq I_S$$
$$v \geq 0$$
where $S = [L \ U]^T$ and $I_S = [I_L \ I_G]^T$. Computation of the upper bounds in this way involves solving $n$ LPs followed by a linear system solve. Note that this verification method is equivalent to performing an exact verification on a resistive grid if $A$ is replaced with $G$.

In the above, optimization problem is formulated in voltage domain. Alternatively, it can also be formulated in current domain, as shown below:

$$
\begin{align*}
V_{a,k} &= \max f_k(i) \\
\text{s.t. } &S_i \leq I_S \\
i &\geq 0
\end{align*}
$$

where $f_k(i) : \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function of $i$ and depends on the $k^{th}$ row of the inverse of system matrix, $A$. It is well known that the inverse of a sparse matrix that results from a mesh structure is almost full. However, since $A$ is sparse, symmetric positive definite (SPD) and banded, the entries in its inverse decay exponentially as the distance from diagonal increases and can be dropped. This fact is exploited in [GN09], where a Sparse Approximate Inverse (SPAI) technique [GH97] is used for power grid verification in current domain.

In spite of the overhead incurred due to the computation of approximate inverse, verification in current domain is still advantageous. This is because the size of the linear programs in optimization is significantly less than the size of linear programs in the voltage domain. Note that the proposed grid reduction scheme is independent of verification methodology and can be applied equally to all. However, the experimental results are generated by verifying the test grids in the voltage domain.
3 Power Grid Verification with Reduced Order Modeling

You miss 100% of the shots you don’t take

Wayne Gretzky

3.1 Introduction

In the last chapter, we reviewed a number of MOR techniques. The primary focus of these and all other available methods is to reduce the circuit size for their efficient simulation. They do not take into account the safety of eliminated nodes and hence, do not perform a conservative analysis. In this chapter, we first develop a criterion that can be used to infer the safety of eliminated nodes using the available information. This gives a way to characterize the safety of an eliminated node when all of its immediate neighbors are retained and verified. Later, the idea is extended to overcome the requirement of retaining all immediate neighbors and allow one to eliminate multiple nodes in the immediate neighborhood.

3.2 Node Safety Criteria

In case of a purely resistive grid, the safety of a node is implied by the safety of all its immediate neighbors. This comes from Kirchoff’s current law (KCL), according to which there must be at least one branch carrying current away from the node under consideration. Since this is a path of increasing voltage drop, at least one of the neighbors will have larger voltage drop than the given node. And therefore, the safety of all neighboring nodes implies the safety of a node. However, in an RC power grid, a node can be connected to capacitors and/or a current source besides resistive
connections to other nodes. If, somehow we can move these capacitors and sources to leave the node with resistive connections only, we no longer need to worry about its safety and it can be eliminated. This is the motivation behind our proposed approach. However, the removal of current sources and capacitors from a node is not trivial and it results in additional voltage drop at that node. We now explain how we deal with these challenges.

3.2.1 Moving Current Sources

The idea of moving a current source is not new and has appeared in the literature from time to time. In [LCC+05], a grid is partitioned into blocks and all current sources internal to these blocks are transformed to Norton current sources at the ports. Elimination of a node in [She07] effectively results in its current source being moved to its neighbors. However, this section explains the effect of moving a current source in the context of verification, emphasizing on its impact on node voltages and on the local and global constraints.

Prior Work

First, we describe how a current source present at a node can be moved to the neighboring nodes while preserving the electrical behavior of the power grid. Fig. 3.1 shows a section of an RC power grid. We are interested in removing the current source at node 0, to get the corresponding circuit in Fig. 3.2. Note that, in doing so, no change to the rest of the circuit is made and only the current sources at immediate neighbors of node 0 are added. The analysis shown below considers a node with four immediate neighbors but can easily be extended to any number of neighbors.

The following development parallels the work in [She07] and is presented for clarity. Writing KCL at node 0 in Fig. 3.1 and transforming it to the Laplace domain gives:

\[
\sum_{i=1}^{4} (v_0 - v_i) g_i + v_0 c_0 s = I_0
\]

where \(v_i\) denotes the voltage at node \(i\), \(g_i\) is the conductance between node 0 and node \(i\), \(c_0\) is capacitance to ground and \(I_0\) represents the current source at node 0. Note that although \(v_i\), \(v_0\) and \(I_0\) are functions of \(s\), they are written like this for brevity. Likewise, writing KCL at node 0 in the modified circuit shown in Fig. 3.2 gives:
Figure 3.1: A power grid node and its neighbors

\[
\sum_{i=1}^{4} (v'_0 - v_i)g_i + v'_0c_0s = 0 \tag{3.2}
\]

We require the voltages at nodes 1, 2, 3 and 4 to be the same in both circuits. However, the node 0 voltage will change due to the removal of the current source and its new value is denoted by \(v'_0\). Subtracting (3.2) from (3.1), we have:

\[
v_0 - v'_0 = \frac{I_0}{G_0(1 + \frac{sC_0}{G_0})} \tag{3.3}
\]

where \(C_0 = c_0\) and \(G_0 = \sum_{i=1}^{4} g_i\). Assume that \(\frac{C_0}{C_0}\) is less than \(\tau\), with \(|s|\tau \ll 1\), referred to the quick node approximation in [She07]. As we will see later on, only the nodes satisfying this property will be considered for elimination. Note that the validity of this approximation depends on the operating frequency. The selection of \(\tau\) can be made [She07] based on maximum operating frequency, \(f_{\text{max}}\) and a small tolerance, \(\epsilon \ll 1\) such that \(\tau = \frac{\epsilon}{2\pi f_{\text{max}}}\). Under this assumption, equation (3.3) can be simplified to:

\[
v_0 - v'_0 \approx \frac{I_0}{C_0} \tag{3.4}
\]

As the electrical behavior of boundary nodes is preserved, the net current leaving node 1 to external circuitry (shown by dashed lines) remains the same in Fig. 3.1 and Fig. 3.2. Therefore, using KCL, we have:
Equations (3.4) and (3.5) lead to \( I_1 \approx \frac{I_{0i}}{G_0} \). Similarly for nodes 2, 3 and 4; we get:

\[ I_i \approx \frac{I_{0i} g_i}{G_0} \quad \forall i = 2, 3, 4 \]

Hence, the current source at node 0 is distributed among its neighbors in weighted ratio of conductances connecting them. This is the same result as in equation (45) of [She07], obtained there in a different context of simulation, rather than verification. The rest of this section is our original contribution.

![Figure 3.2: Grid section after moving current source](image)

**Application to Verification**

Because the values of the current sources are not known explicitly, they cannot be distributed directly as suggested above. Only the peak values of these sources are specified, in terms of local and global constraints. Therefore, we have to express the above current source distribution scheme in terms of current constraints, in order to formulate the verification problem on a reduced grid.

**Local current constraints** Let \( I_j \) be a feasible current assignment for some neighbor \( j \) of node \( i \). Then, according to the local constraints, we have:

\[ I_j \leq I_{L,j} \]
The removal of the current source from node \( i \) will introduce a new current source at node \( j \). Node \( j \) will now have this new current source in addition to any current source it may have had previously. Therefore, the modified feasible current assignment for node \( j \), \( I'_j \) can be written as:

\[
I'_j \approx I_j + \frac{I_i g_j}{G_i}
\]

where \( I_i \) is the feasible current assignment for node \( i \), \( g_j \) is the conductance connecting node \( i \) and node \( j \) and \( G_i \) is the sum of conductances connected to node \( i \). From the above two equations, we have:

\[
I'_j \leq I_{L,j} + \frac{I_i g_j}{G_i}
\]

Using \( I_i \leq I_{L,i} \) in the above equation gives:

\[
I'_j \leq I'_{L,j} = I_{L,j} + \frac{I_{L,i} g_j}{G_i}
\]

where \( I'_{L,j} \) is the new value of local constraint at node \( j \). Therefore, the modified local constraint is obtained by adding to its original value, the local constraint value of its neighbor scaled by conductances.

As discussed in section 2.4.2, we have the following overall local constraints.

\[
LAv \leq I_L \quad (3.6)
\]

The voltage vector, \( v \) and the vector of upper bounds on current sources, \( I_L \) for the small grid section shown in Fig. 3.1 can be written as: \( v = [v_0 \ v_1 \ v_2 \ v_3 \ v_4]^T \) and \( I_L = [I_{L,0} \ 0 \ 0 \ 0 \ 0]^T \). After moving the current source from node 0, we have the following modified current constraints which are obtained by taking the value of current source at node 0 to be \( I_{L,0} \) and then distributing it among its neighbors:

\[
LAv' \leq I'_L \quad (3.7)
\]

where \( v' = [v'_0 \ v_1 \ v_2 \ v_3 \ v_4]^T \) and \( I'_L = \left[ 0 \ \frac{I_{L,0} g_1}{G_0} \ \frac{I_{L,0} g_2}{G_0} \ \frac{I_{L,0} g_3}{G_0} \ \frac{I_{L,0} g_4}{G_0} \right]^T \).

If we assume that there is no approximation involved, then the systems of constraints represented by equations (3.6) and (3.7) are equivalent and represent the same feasible space of voltages. The maximum voltage drop at boundary nodes will be equal in both cases, and the node 0 maximum voltage drop can be computed using
equation (3.4) as:

$$\max(v_0) = \max(v'_0) + \frac{I_0}{G_0}$$

In fact, the quick node assumption is a good approximation and its validity will be clear from experimental results. In conclusion, if a neighbor $j$ has its own current source, its modified local constraint will be written as:

$$I'_{L,j} = I_{L,j} + \frac{I_{L,0g_j}}{G_0} \quad j \in \{1, 2, 3, 4\}$$

**Global current constraints**  Let there be $m$ global constraints, represented as:

$$UAv \leq I_G$$

If we remove a current source from node $i$, all entries in the $i^{th}$ column of $U$ should then be set to zero. Besides this, if the $k^{th}$ constraint contains node $i$, then $U_{kj}$ is set to 1 for every neighbor $j$ of node $i$. This is because new current sources at the neighbors must collectively contribute in same amount as was done by the source at node $i$ alone. Further, if neighbor $j$ was not initially included in the $k^{th}$ constraint and has its own current source, then $I_{G,k}$ must be updated by adding to it the local constraint value corresponding to node $j$, $I_{L,j}$. It can be easily seen that $I_{G,k}$ remains unchanged if the neighbor of node $i$ has its own current source and is already included in constraint $k$. The modified global constraints may be represented as:

$$U'Av' \leq I'_G$$

Consider the small grid section in Fig. 3.1. Suppose, we have current sources at all the neighbors of node 0, except node 1, and have the following global constraint on them.

$$I_0 + I_2 + I_4 \leq I_{G,1}$$ \hspace{1cm} (3.8)

In order to get the modified global constraint, when $I_0$ is removed, we replace $I_0$ with new current sources created at the neighbors, leading to:

$$\sum_{i=1}^{4} \frac{I_{0g_i}}{G_0} + I_2 + I_4 \leq I_{G,1}$$
which can be re-written as:

\[ \frac{I_0 g_3}{G_0} + I'_1 + I'_2 + I'_4 \leq I_{G,1} \]

with \( I'_1 = \frac{I_0 g_1}{G_0} \), \( I'_2 = I_2 + \frac{I_0 g_2}{G_0} \) and \( I'_4 = I_4 + \frac{I_0 g_4}{G_0} \). We can do this substitution because the neighbor 1, 2, and 4 were either already included in the constraint or did not have any current sources of their own. However, node 3 has its own current source and is not already included in constraint, in which case, \( I_{G,1} \) needs to be updated by adding \( I_3 \) to the left hand side and \( I_{L,3} \) to the right hand side, so that the modified constraint becomes:

\[ I'_1 + I'_2 + I'_3 + I'_4 \leq I'_{G,1} \]

(3.9)

where \( I'_3 = I_3 + \frac{I_0 g_3}{G_0} \) and \( I'_{G,1} = I_{G,1} + I_{L,3} \).

Notice that the set of constraints in equations (3.8) and (3.9) are not equivalent. In fact, the feasibility space defined by (3.8) is a subset of that defined by (3.9), as is evident from the equation below:

\[
\begin{cases}
    I_3 \leq I_{L,3} \\
    \frac{I_0 g_3}{G_0} + I'_1 + I'_2 + I'_4 \leq I_{G,1}
\end{cases} \Rightarrow \{ I'_1 + I'_2 + I'_3 + I'_4 \leq I'_{G,1} \}
\]

Consequently, in the absence of any approximations, the analysis performed using the modified global constraints guarantees a conservative estimate. However, as we will see later, the experimental results are not overly pessimistic and the error is quite small.

**Implications for node safety**

As noticed from equation (3.3), the movement of a current source from a node affects its voltage. From equation (3.3), we have:

\[ v_0 \approx v'_0 + \frac{I_0}{G_0} \]

(3.10)

Note that the voltage at node 0 is reduced in the modified network by a quantity approximately equal to \( \frac{I_0}{G_0} \), referred to as the auxiliary voltage drop or simply the auxiliary drop of node 0. Therefore, whenever a current source is moved from a node, its auxiliary drop must be added to the node voltage in the modified network in order to obtain the actual node voltage.
3.2.2 Moving Capacitances

Consider again the generic node situation shown in Fig. 3.1. After moving the current source, we are left with a circuit as seen in Fig. 3.2. Next, we want to move the capacitor from node 0 to its immediate neighbors. The target modified circuit, with capacitor $c_i$ added to neighbor $i$, is shown in Fig. 3.3.

![Figure 3.3: Grid section after moving capacitor](image)

Prior Work

The required analysis follows from the node elimination process in [She07], which results in a similar distribution of the capacitor connected to an eliminated node. Writing KCL at node 0 for the modified circuit in Fig. 3.3 and transforming it to the Laplace domain gives:

$$\sum_{i=1}^{4} (v_0'' - v_i)g_i = 0 \quad (3.11)$$

where $v_0''$ is the new voltage at node 0. Subtracting (3.2) from (3.11), we have:

$$v_0'' = v_0' \left( 1 + \frac{c_0s}{\sum_{i=1}^{4} g_i} \right) \quad (3.12)$$

For the same reasons discussed above, by KCL this gives:

$$v_1c_1s + (v_1 - v_0'')g_1 - I_1 = (v_1 - v_0')g_1 - I_1$$
Substituting the value of $v_0'$ from (3.12), the above equation becomes:

$$v_1 c_1 s + (v_1 - v_0'')g_1 = \left(v_1 - \frac{v_0''}{1 + \frac{c_0 s}{\sum_{i=1}^{4} g_i}}\right) g_1 \quad (3.13)$$

Expanding the $\left(1 + \frac{c_0 s}{\sum_{i=1}^{4} g_i}\right)^{-1}$ term in equation (3.13) in increasing powers of $s$ and retaining terms up to first order, we have:

$$v_1 c_1 s + (v_1 - v_0'')g_1 \approx (v_1 - v_0'') \left(1 - \frac{c_0 s}{\sum_{i=1}^{4} g_i}\right) g_1 + \frac{v_1 g_1 c_0 s}{\sum_{i=1}^{4} g_i}$$

If $\frac{c_0 s}{\sum_{i=1}^{4} g_i} \ll 1$, as according to the quick node assumption [She07], we have:

$$v_1 c_1 s + (v_1 - v_0'')g_1 \approx \frac{v_1 g_1 c_0 s}{\sum_{i=1}^{4} g_i} + (v_1 - v_0'')g_1$$

Therefore,

$$c_1 \approx \frac{g_1 c_0}{G_0}$$

Likewise, for nodes 2, 3 and 4, we have:

$$c_i \approx \frac{c_0 g_i}{G_0} \quad \forall i = 2, 3, 4$$

Thus, similar to the current source case, a capacitor is also distributed among its neighbors in a weighted ratio of conductances.

**Implications for node safety**

Unlike the previous case of moving a current source, moving a capacitance away from a node will have negligible impact on its voltage. This is because $\frac{|s| c_0}{\sum_{i=1}^{4} g_i}$, by assumption, is very small and utilizing this in equation (3.12) gives:

$$v_0'' \approx v_0' \quad (3.14)$$

This means that the voltage at node 0 in the modified network is approximately equal to that in the original network.
3.2.3 Characterizing Safety

After removing a current source and/or capacitance from node 0, it is connected to its neighboring nodes only via conductances as seen in Fig. 3.3. As pointed out earlier, in this case, the safety of all neighboring nodes would guarantee the safety of node 0. In terms of voltage drops:

\[ v_0'' \leq \max(v_1, v_2, v_3, v_4) \]  

(3.15)

Therefore, if all of the neighbors are known to be safe, node 0 must be safe as well. As we no longer need to directly verify node 0 for safety, it can be eliminated using an exact star-mesh transformation. The equivalent circuit shown in Fig. 3.4 is obtained by removing node 0 and adding conductance \( g_{ij} \) between neighbors \( i \) and \( j \), such that:

\[ g_{ij} = \frac{g_i g_j}{G_0} \]

where \( g_i \) is the conductance between node 0 and neighbor \( i \).

![Node elimination using star-mesh transformation](image)

Figure 3.4: Node elimination using star-mesh transformation

However, besides ensuring the safety of the neighbors, we must also account for the auxiliary drop incurred due to current source removal. From section 3.2.1, we know that removal of current source from node 0 results in an auxiliary drop of \( I_0/G_0 \). Also, recall that removal of a capacitor does not cause any additional auxiliary drop. Therefore, two factors define the safety of a node in the original grid: safety of its immediate neighbors and its auxiliary drop. From equations (3.10), (3.14) and (3.15), the criterion for safety of node 0 becomes:
\[ \frac{I_0}{G_0} + \max(v_1, v_2, v_3, v_4) \leq V_{\text{allowed}} \]

where \( V_{\text{allowed}} \) is the maximum voltage drop allowed on any grid node.

### 3.3 Grid Reduction

In the previous sections, we described a procedure to eliminate a node and establish its safety. We are more interested in eliminating a group of nodes, rather than a single node. We want to repeatedly apply the above node elimination procedure in order to eliminate whole sections of the grid. However, the above procedure cannot be applied to all grid nodes, but only to the ones that have a small \( C_0/G_0 \) ratio, as we have made this approximation at several instances. Therefore, the first step will be to identify regions containing these special nodes. For doing this, the grid nodes are traversed topologically, starting from an arbitrary node, and all the adjacent nodes which have small \( C_0/G_0 \) are grouped together in the same region.

We must also keep track of nodes which will later be used to characterize the safety of nodes that have been removed. Define the safety set, \( X \), of a region to be the collection of all those nodes which will be included in the safety criteria of any internal (removed) node \( i \) in that region.

\[ \frac{I_i}{G_i} + \max(v_s \mid \forall \ s \in X) \leq V_{\text{allowed}} \]

As we expand the region, we keep updating the safety set by adding to it the neighbors of the node under consideration and removing from it, if present, the current node. As the region grows, the safety criteria may become pessimistic as a result of increased auxiliary drops and, possibly, due to maximization over a bigger set. Therefore, a user-specified limit on the size of regions is imposed. Once a region is created, its safety set must contain all its external boundary nodes.

These regions will initially include nodes having current sources as well as capacitors. Once we move these elements from internal nodes to the boundary, after accounting auxiliary drops introduced, we would get regions having node to node conductances only. In case of such purely resistive regions, the voltage drop at any internal node must be less than some boundary node according to the same KCL argument given earlier. Therefore, the safety of an internal node is governed by the
safety of boundary nodes in addition to its auxiliary drop.

### 3.3.1 Computing Auxiliary Drops

We have seen how a current source can be moved from a node to its immediate neighbors. The result had been the introduction of auxiliary drop at the node, along with secondary current sources at its neighbors. Because our aim is to move current sources all the way to the boundaries, the secondary sources created at the neighbors must be moved further away, one by one, until the boundary is encountered. The movement of secondary sources will introduce auxiliary drops at these neighbors as well. Since the safety of a node is computed with respect to its neighbors, and if these neighbors themselves have auxiliary drops, then the auxiliary drop at the node must be updated by adding to it the maximum auxiliary drop among its neighbors. Moreover, as the exact values of current sources are not known, their maximum values specified in the local constraint vector, $I_L$, are used to compute auxiliary drops. The `computeAuxDrop` function, presented in Procedure 1, performs this process recursively.

**Procedure 1 computeAuxDrops**

**Inputs:** regionSize, node, totalAuxDrops, incAuxDrops, violate  
**Outputs:** incAuxDrops, violate  

1. $\text{incAuxDrops[node]} += \frac{I_{node}}{G_{node}}$  
2. if $(\text{incAuxDrops[node]} < \epsilon)$ then  
3. prune = true  
4. if $(\text{incAuxDrops[node]} + \text{totalAuxDrops[node]} > \gamma)$ then  
5. violate = true and return 0  
6. if $(\text{prune} \neq \text{true} \text{ and regionSize} \neq 1)$ then  
7. Distribute $I_{node}$ among neighbours and eliminate node  
8. for (each neighbor of node inside region) do  
9. if (violate $\neq$ true) then  
10. neighbourDrop = computeAuxDrops(regionSize-1, neighbor, totalAuxDrops, incAuxDrops, violate)  
11. if (maxNeighborDrop < neighbourDrop) then  
12. maxNeighborDrop = neighbourDrop  
13. incAuxDrops[node] += maxNeighborDrop  
14. if $(\text{incAuxDrops[node]} + \text{totalAuxDrops[node]} > \gamma)$ then  
15. violate = true and return 0  
16. return incAuxDrops[node]

The function moves a current source at a node to create secondary sources at its
immediate neighbors while updating the auxiliary drop at the node. The function is then called again, recursively, for each of these secondary sources and the recursion is continued until the boundary is encountered or some condition is violated. Once all the inner recursions are executed, the auxiliary drop at the node is incremented by adding to it the maximum auxiliary drop among all its neighbors. However, we do not want these auxiliary drops to become too large because then the nodes may be falsely classified as unsafe. Therefore, if moving a current source causes the auxiliary drop at any node to exceed a user-specified threshold, $\gamma$, then the node containing that current source is excluded from the region and added to the safety set, and the violate flag is set. The safety set, $X$ of a region now consists of all such nodes in addition to its boundary nodes as shown in Figure 3.5. In the figure, solid square inside a region represents a node that is retained to prevent its current source from causing excess auxiliary drops. These nodes along with the boundary nodes, represented by same color solid squares along the periphery of a region, constitute the safety set. Thus, the safety of nodes in the safety set will essentially guarantee the safety of all internal nodes within a margin, which is equal to the threshold value, $\gamma$ itself.

![Figure 3.5: Recursive grid reduction to generate safety regions](image)

Note that the node must be eliminated after moving its current source, otherwise a source will reappear there when the secondary sources on its neighbors are moved. However, this is not the final elimination and is done for the computation of auxiliary drops only. We start with a fresh copy of the conductance matrix, $G$, for each current source. This is done in order to consider the mutual effect of two current sources on neighboring nodes. If one of them is eliminated before the other, the effect of the
neighboring current source will not appear in its auxiliary drop unless we start with a new copy of G for the second node, with the first node intact in its place. Actual node elimination is performed once auxiliary drops due to all internal current sources have been computed.

Typically, there can be several current sources in a region. Instead of taking up current sources in random order, they are considered in increasing order of magnitude. This is done to prevent substantial increase in auxiliary drop of nodes in the initial stages. If a large current source is moved first, it may create a large auxiliary drop on some nodes and when, later, smaller current sources are moved, the threshold limit on some of these nodes may be exceeded.

Note that the number of secondary sources increases exponentially as we repeatedly move current sources. This is because whenever a node is removed, after moving its current source, all its neighbors get interconnected, thus increasing their connectivity at each level of elimination. Conceptually, a graph corresponding to a region can be considered, as shown in Fig. 3.6, where child nodes are the neighbors of a parent node in the grid. Whenever a node is eliminated, all its child nodes become connected. If we are not careful, this continuous graph restructuring will result in the exponential complexity of computeAuxDrop procedure presented above.

![Figure 3.6: Dealing with exponential complexity](image)

However, we can simplify our algorithm significantly by pruning, similar to what is done in many decision problems. Notice that the incremental auxiliary drops keeps on decreasing as we traverse down the graph. This is related to the concept of grid locality, which states that the effect of a current source diminishes as we move away from it [Chi04]. The factors by which the drops are decremented are quite significant and are, for example: 1, 3.75, 14, 42, 100 ... for a uniform grid of degree four with
uniform conductance values. So, practically, these drops become so small after a few levels down the graph that there is no point in computing them. Hence, we can define a threshold, ε, on the auxiliary drops and prune everything below it, thus bringing the algorithm to near constant time complexity.

Once we have computed auxiliary drops due to all current sources, the next step is to eliminate nodes from a region. To remove a node, its current source and/or capacitance is moved to its neighbors, followed by exact star-mesh transformation as discussed earlier.

### 3.3.2 Sparsity Considerations

In the field of numerical analysis, a sparse matrix is a matrix populated primarily with zeros [SB02]. The star-mesh transformation results in the addition of new connections to each neighbor of the eliminated node. If there is a region containing n nodes, the removal of these nodes will increase the number of non-zero terms in the conductance matrix by \(O(nm)\), where \(m\) is the number of boundary nodes of this region. Therefore, the total number of fill-ins created, across the whole grid, would be of the order of

\[
\sum_{i=0}^{r} n_i m_i
\]

where \(r\) is the total number of regions, \(n_i\) is the number of internal nodes in the \(i^{th}\) region, and \(m_i\) is the number of boundary nodes of the \(i^{th}\) region. A large number of fill-ins results in a denser constraint matrix, which may lead to slower optimization. The runtime may actually increase because the denser constraints may dominate over the reduced problem size. This is another reason to limit the size of regions. This, in a sense, will reduce both \(n_i\) and \(m_i\) but with an increased \(r\). Nevertheless, the reduction in the quadratic \(n_i m_i\) is significant, compared to an increase in the linear effect of \(r\).

Moreover, as we remove nodes from a region, many of the new connections added are insignificant in terms of magnitude and can be dropped without introducing much error. For instance, consider two grid nodes located far from each other and let all the nodes between them be eliminated, resulting in a new connection between these two nodes. Since removal of each intermediate node will cause the conductance to decrease, the ultimate conductance connecting these two nodes will be negligible. Therefore, in order to maintain sparsity, any new connections which are below a cer-
tain threshold, $\kappa$, are dropped. This is similar to the approximation step of sparsification, which is often employed when working with large matrices. Another advantage of doing this is to avoid propagating insignificant connections as we remove nodes, thus making the reduction process faster.

### 3.4 Algorithm

**Algorithm 1** Overall flow

1: identify regions based on $\tau$
2: $\text{totalAuxDrops} = 0$
3: for (each region) do
4:   regionSize = number of nodes inside this region
5:   identify nodes with current sources and sort them
6:   for (each of these nodes in given order) do
7:     violate = false
8:     $\text{incAuxDrops} = 0$
9:     $\text{computeAuxDrops}(\text{regionSize}, \text{node}, \text{incAuxDrops}, \text{totalAuxDrops}, \text{violate})$
10:    if (violate is true) then
11:       exclude this node from region & include in safetySet
12:       discard $\text{incAuxDrops}$
13:    else
14:       update totalAuxDrops by adding $\text{incAuxDrops}$
15:    eliminate internal nodes of all regions while dropping insignificant connections less than $\kappa$
16: verify the reduced grid

A high-level description of the overall process is given in Algorithm 1. The algorithm starts with the identification of regions based on the quick node criterion, in which adjacent nodes having $C_0/G_0$ ratios smaller than a threshold $\tau$ are grouped together as a region. Next, in steps 3-9, we compute incremental auxiliary drops due to each current source inside a region, as described in Procedure 1. While doing this, if the total auxiliary drop at any internal node exceeds the user-specified threshold, then the node containing that current source is excluded from the region and added to the safety set. This action is performed in steps 10-12. Otherwise, we execute steps 13-14 in which the incremental auxiliary drops caused due to this current source are recorded in the total drops. As discussed in the previous section, step 5 ensures that the current sources are considered in increasing order of magnitude. Once auxiliary
3 Power Grid Verification with Reduced Order Modeling

Table 3.1: Results of verification performed on reduced and original power grid

<table>
<thead>
<tr>
<th>#Nodes</th>
<th>% Reduction</th>
<th>Runtime</th>
<th>% Runtime Savings</th>
<th>Max Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reduced</td>
<td>Original</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4,050</td>
<td>64.12%</td>
<td>7.18 mins</td>
<td>24.5 mins</td>
<td>70.7%</td>
</tr>
<tr>
<td>8,722</td>
<td>65.58%</td>
<td>60.2 mins</td>
<td>139 mins</td>
<td>56.7%</td>
</tr>
<tr>
<td>15,313</td>
<td>66.61%</td>
<td>2.84 hours</td>
<td>7.44 hours</td>
<td>61.88%</td>
</tr>
<tr>
<td>24,100</td>
<td>64.65%</td>
<td>9.78 hours</td>
<td>23.37 hours</td>
<td>58.14%</td>
</tr>
<tr>
<td>34,522</td>
<td>64.96%</td>
<td>20.32 hours</td>
<td>48.73 hours</td>
<td>58.3%</td>
</tr>
<tr>
<td>46,605</td>
<td>65.22%</td>
<td>39.05 hours</td>
<td>100.58 hours (est.)</td>
<td>61.2% (est.)</td>
</tr>
</tbody>
</table>

drops are computed, all the internal nodes of a region are eliminated with star-mesh transformations after moving any capacitances and/or current sources connected to them using the procedure described in section 3.2, thereby resulting in a smaller grid in step 15. Finally, verification is performed on the reduced grid and an upper bound on the worst case voltage drops is computed as in section 2.4.3. The safety of the internal (eliminated) nodes of a region is guaranteed by those in its safety set, taking into account the auxiliary drops.

3.5 Experimental Results

A C++ implementation is written to evaluate the proposed approach. The grid verification problem is formulated as a linear program (LP) and solved using the interior-point based optimizer included in MOSEK [MOS]. Our test grids were generated according to user specifications, which include grid dimensions, the number of metal layers, C4 and current source distributions, among others. All experiments were performed on a 64-bit, 2.6GHz, dual core AMD Opteron™ processor with 8GB of memory.

The results are presented in Table 3.1 and are obtained for $\epsilon = 0.05mV$, $\gamma = 10mV$, $\kappa = 1e^{-3}\Omega$ and $\tau = 5e^{-12}s$. The size of a region is limited to 1% of total grid size in order to maintain sparsity. The percentage of nodes eliminated is shown in the second column. The extent of reduction can be specified explicitly by the user or it can be automatic, depending on the criterion used for classifying quick nodes ($C_0/G_0 < \tau$). The CPU times for verifying the reduced and the original grids are shown. The runtime for the reduced grid includes the time consumed by the algorithm to eliminate internal nodes and to compute auxiliary drops. Note that
Figure 3.7: Correlation plot between worst-case voltage drops on the retained nodes in the original grid and the corresponding nodes in the reduced grid

The safety of eliminated nodes can be established using the criteria developed earlier in this paper. Moreover, if all the verified nodes have voltage drops less than the allowed limit by an amount equal to $\gamma$, then the safety check on the eliminated nodes can be skipped and the grid may be considered safe. A histogram of the differences between the worst-case voltage drops on eliminated nodes computed from safety criteria (i.e., maximum of the voltage drop on nodes in the safety set, plus the auxiliary drop) and the actual values from the full grid is shown in Fig. 3.8. The plot confirms that the values provided by the safety criteria exceed actual values in all cases and hence, is a conservative estimate.

In the following, all the plots are for the 8722-node grid with all parameters,
Figure 3.8: Histogram of error in worst-case voltage drops on eliminated nodes

Figure 3.9: Effect of varying conductance threshold, $\kappa$ on runtime and error
except the one under study (if any), kept fixed at their values used in Table 1, and the maximum voltage drop on this grid is 60.3mV. In order to ensure good speed-ups, recall, it is important to maintain sparsity of the conductance matrix. Our approach deals with this issue, explicitly by introducing a threshold on conductance values, $\kappa$ and implicitly by limiting the size of the regions. A higher value of $\kappa$ would mean fewer fill-ins and hence, faster optimization, but at the cost of an increased error as shown in Fig. 3.9.

![Graph showing impact of region size on sparsity and reduction]

**Figure 3.10: Effect of region size on sparsity and reduction**

As for region size, the tradeoff is between sparsity and the percentage reduction, as shown in Fig. 3.10, where sparsity is defined as the percentage of zero elements in a matrix. The smaller the maximum allowable size of the regions, the more sparse the matrix is but with lesser reduction as an increased number of nodes are boundary nodes and cannot be removed.

As mentioned earlier, the extent of grid reduction is controlled by the C/G threshold, $\tau$. Fig. 3.11 shows the variation of $\tau$ and its effect on runtime and reduction. Initially, as $\tau$ is increased, more nodes are eliminated accompanied by a gain in runtime due to the increased reduction. Afterwards, the graph saturates as the remaining grid nodes are not suitable for elimination because they are either boundary nodes, C4s or those with current sources which exceeded the margin, $\gamma$. 
Another parameter that dictates the extent of reduction is the auxiliary drop margin, $\gamma$. The variation of reduction with this margin is shown in Fig. 3.12. If the margin is low, a large number of current sources may violate the criteria. As the nodes containing these current sources can no longer be eliminated, less reduction is achieved. The reduction increases with margin up to a certain level after which the graph saturates. At this stage, the margin is large enough and almost all the current sources can be removed without exceeding the margin. Any further increase in margin will result in a minimal increase in grid reduction.

The time required to compute auxiliary drops is another contributing factor to the overall runtime. In order to minimize runtime, a pruning strategy is employed which involves the threshold, $\epsilon$. Fig. 3.13 shows a plot of the time taken to compute auxiliary drops for different values of $\epsilon$. In the absence of any pruning mechanism, computation of auxiliary drops becomes practically infeasible as is evident from the rising curve as $\epsilon$ approaches zero. Also, as observed from the graph, even a small value of $\epsilon$ can result in significant runtime savings.

![Figure 3.11: Reduction and runtime variations with $\tau$](image)
Figure 3.12: Effect of margin, $\gamma$ on reduction

Figure 3.13: Impact of $\epsilon$ on time requirements
4 Extensions

Logic will get you from A to B.
Imagination will take you everywhere

Albert Einstein

4.1 Introduction

This chapter extends the concept developed previously in two other domains. In situations, when multiple verification passes of the same portion of a power grid is required in different settings, its macro-modeled instance can be used instead. Furthermore, a way to isolate the macro-models will allow different portions of the grid to have different level of abstraction. Another usage of the proposed idea can be found in incremental verification to obviate the need of re-verifying the whole grid, in case local changes are made.

4.2 Macro-model Generation

The current reduction scheme does not allow to have different levels of abstraction for different parts of the grid. The reduction is guided by a global quick node criteria and hence, uniform across the grid. However, in some cases, grid is known to have minimal loading in certain areas. Such areas can be reduced more aggressively to a higher abstraction level in anticipation of minimal voltage drops. Therefore, macro-models with different level of abstraction can be created for different parts of the grid as illustrated in Figure 4.1. Furthermore, if a section of the grid is repeated or intended to be used in a different setting, it can be replaced by its macro-model. This would obviate the need to verify all of its nodes repeatedly and only those included in its macro-model can be verified. The use of a macro-model will simplify the system while allowing us to correlate the safety of the original system to its macro-model.
47

Figure 4.1: Macro-modeled blocks with varying extent of reduction
The formulation of a macro-model is similar to that of a region discussed previously except two changes. One, the boundary of a macro-model is specified by the user, as opposed to the region which is determined automatically. Secondly, unlike the case of a region, macro-model may contain nodes which do not satisfy quick node criteria i.e. with $C_0/G_0 > \tau$. As these nodes must be retained for an accurate analysis, the safety set for a macro-model will also include these nodes. Note that there is no need to compute auxiliary drops due to the current sources present at these nodes as they are not eliminated.

Safety set of a macromodel, $X_m = \{\ast, \bullet, \circ\}$

Safety set of a region, $X_r = \{\bullet, \circ\}$

where $\bullet$ represents boundary nodes, $\circ$ represents internal nodes with current sources which are retained in order to keep auxiliary drops within limits and $\ast$ are the nodes which do not satisfy quick node assumption. The concept is shown graphically in Figure 4.2.

![Figure 4.2: Region Vs. Macro-model](image)

### 4.3 Incremental Verification

Often, it is the case that few faulty nodes are detected after the initial verification run. The worst-case voltage drop at these nodes exceeds the limit set for safe operation. In order to fix this, a designer needs to do some grid modifications such as change in the metal width, pitch etc. Once all the changes are made, the grid needs to be re-verified again to make sure that faulty nodes are corrected and no other node (classified safe earlier) has become faulty due to the modifications made. This process
is applied repeatedly until all the nodes are found to be safe. This, however, may require several runs of full grid verification, which is highly impractical.

In this scenario, we can benefit from the fact that changes made to the grid are local and hence, verification of the complete grid can be avoided. This can be done by just verifying the modified area, referred to as area under verification (AUV), in full and simplifying the rest of the grid. Note that since we need not to consider the safety of nodes outside AUV, these nodes can be eliminated by the procedure discussed earlier, without the need of computing auxiliary drops.

<table>
<thead>
<tr>
<th>#Nodes</th>
<th>% Reduction</th>
<th>Time(incremental)</th>
<th>Time(full)</th>
<th>Speedup</th>
<th>Max Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,050</td>
<td>58.29%</td>
<td>89.7s</td>
<td>146.41s</td>
<td>1.632×</td>
<td>4.09 mV</td>
</tr>
<tr>
<td>8,722</td>
<td>60.05%</td>
<td>509.06s</td>
<td>798.25s</td>
<td>1.57×</td>
<td>7.91 mV</td>
</tr>
<tr>
<td>15,313</td>
<td>59.84%</td>
<td>27.42m</td>
<td>46.18m</td>
<td>1.68×</td>
<td>1.66 mV</td>
</tr>
</tbody>
</table>

Table 4.1: Incremental verification results

Experiments are performed to test the accuracy and runtime efficiency in case only the AUV is verified, while macro-modeling the rest of the grid. The results for three different grids with 16% of M1 nodes as AUV, conductance threshold, \( \kappa = 0.05 \) mho and region size limited to 1% of total grid nodes are shown in Table 4.1. Note that it is possible to gain speed-ups at the cost of accuracy by increasing the conductance threshold, \( \kappa \) as demonstrated for the 15,313-node grid in Figure 4.3. Clearly, one can achieve significant runtime savings with minimal loss of accuracy by employing incremental verification.
Figure 4.3: Effect of varying conductance threshold, $\kappa$ on runtime and accuracy of incremental verification
5 Conclusion and Future Directions

Success is not final, failure is not fatal: It is the courage to continue that counts

---

Winston Churchill

In integrated circuits, power is supplied from pad locations to the various components of the chip over a power distribution network (PDN) or simply the power grid. Decreasing voltage levels, faster switching frequencies and increasing power consumption in deep sub-micron technologies cause large switching currents to flow in the PDN which degrade performance and reliability. A robust PDN is essential to ensure safe operation of the circuits on the chip. As a result, power grid integrity verification is a critical concern in high-performance designs.

With the growing size of power delivery networks, it is becoming difficult to verify them with traditional methods. The verification process is becoming increasingly costly in terms of time and computation resources, calling for the need of model order reduction (MOR)-like techniques. MOR is used widely to abstract large systems for their efficient analysis. However, existing MOR techniques are not suitable for power grid verification. In this work, we propose a technique to selectively eliminate certain nodes of the power grid, while verifying the rest. The safety of the eliminated nodes is inferred from the safety of those which are verified within a predefined margin. Experimental results show excellent accuracy, along with significant runtime savings.

Although, the worst-case voltage drop computed on retained nodes in the reduced grid is very close to the corresponding values in the original grid, it may not be a conservative estimate. This means that the voltage drop at a node in the reduced configuration can be slightly underestimated. This is due to the involvement of approximations, which are necessary to maintain the efficiency and feasibility of this work. However, in the future, one can develop a reduced model verification procedure which always guarantees a tight conservative estimate. In this case, depending
on the maximum allowable overestimation, it should be possible to guide the extent of approximation or choice of parameters automatically.

The safety criteria developed here on the basis of KCL is no longer valid, in case, node has inductive connections. Also, the proposed technique capitalizes, to some extent, on the low-pass behavior of an RC grid. Although, in a grid, lower metal layers are mainly RC, upper metal layers might have considerable inductive effects due to their interaction with the package. Therefore, an accurate model of the power grid must also include inductances.

However, in presence of inductances, the low-pass filter property is lost and currents can travel long distances across the grid. This essentially destroys the property of grid locality exploited in the proposed approach. Furthermore, in the case of an RLC grid model, besides verifying for voltage drops, one must also verify the grid for voltage overshoots to ensure that the voltage level at a given node does not exceed the nominal supply voltage. Consequently, the node safety criteria applicable to RLC grids must test the nodes against overshoots as well. In the future, work can be done to extend this work to include inductances also.
References


