

國立交通大學

電信工程學系

碩士論文

集合體基礎的代數多重網格方法在晶片  
上功率網路分析上的應用

An Aggregation-Based Algebraic Multigrid Method  
for On-Chip Power Network Analysis

研究生：周桓宇

指導教授：李育民 教授

中華民國九十五年七月

集合體基礎的代數多重網格方法在晶片上功率網路分析上的應用  
An Aggregation-Based Algebraic Multigrid Method  
for On-Chip Power Network Analysis

研究生：周桓宇  
指導教授：李育民

Student : Huan-Yu Chou  
Advisor : Yu-Min Lee



Submitted to Department of Communication Engineering  
College of Electrical Engineering and Computer Science  
National Chiao Tung University  
in partial Fulfillment of the Requirements  
for the Degree of  
Master  
in  
Communication Engineering

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

# 集合體基礎的代數多重網格方法在晶片上功率網路分析上的應用

學生：周桓宇

指導教授：李育民

國立交通大學電信工程學系碩士班

摘 要



隨著次深次微米技術演進到 0.18 微米以下，晶片上功率傳輸網路的分析，已經變成在今日的高效能晶片設計下的一個非常重要而且具挑戰性的一個問題。功率傳輸網路上較低的電源電壓，將會減少電路的雜訊容忍度。除此之外，較高的電路操作頻率，將使得由  $Ldi/dt$  電壓壓降而來的電路雜訊為之增加。這些效應將會增加功率傳輸網路的設計複雜度以及對於有效率的功率傳輸網路分析方法的需求。

在本篇論文當中，對於功率傳輸網路分析，我們提出了一個以集合體基礎的代數多重網格分析方法。首先，我們將原始的功率傳輸網路模擬成許多 RLKC 元件以及片段線性的電流源。然後，利用修飾節點分析方法，我們可以把原始問題轉換成一個  $Ax=b$  的線性代數問題。在此， $A$  是一個  $n \times n$  的矩陣， $x$  和  $b$  是  $n \times 1$  的向量。在對於這個線性代數問題，應用了我們的集合體演算法之後，我們可以把原始的系統矩陣分成許多小的子矩陣，並實行一個代數的切割去簡化問題。

實驗結果顯示出，我們的集合體基礎的代數多重網格方法，在時間和記憶體方面，比較傳統的代數多重網格方法以及現存的改善 Krylov 子系統方法，都得到了更好的結果。

# An Aggregation-Based Algebraic Multigrid Method for On-Chip Power Network Analysis

Student : Huan-Yu Chou

Advisor : Dr. Yu-Min Lee

Department of Communication Engineering  
National Chiao Tung University



As the ultra deep sub-micron technology scales down to 0.18  $\mu\text{m}$ , power distribution network analysis becomes one of the most critical and challenging problems in today's high performance chip design. Lower supply voltage on power distribution network decreases the circuit noise margin and higher circuit operation frequency increases the circuit noise from  $Ldi/dt$  voltage drop. Those effects increase the design complexity of power distribution network and also increase the demand of efficient power distribution network analysis methods.

In this thesis, we present an aggregation-based algebraic multigrid method for power distribution network analysis. First, we model the original power distribution network with RLKC segments and piecewise linear current sources. Then we use modified nodal analysis to transform the problem into an  $Ax=b$  linear algebraic problem where  $A$  is a  $n \times n$  matrix,  $x$  and  $b$  are  $n \times 1$  vectors. By performing an aggregation algorithm, the original system matrix is divided into many small sub-matrices and an algebraic partition is performed to simplify our problem.

Experimental results show our aggregation-based algebraic multigrid method runs faster and spend less memory usage than both traditional algebraic multigrid method and the existing **IEKS** (Improve Krylov Subspace) method.

## 誌 謝

在這篇論文能順利完成的當下，我在此首先最要感謝的就是我的指導教授 李育民博士的指導，老師的專業背景和認真的研究態度，總是能適時的指出學生在研究時所無法發現的盲點，從而做出正確的修正。在碩士生涯所學到的研究態度不只是得到這個學位，相信在之後的人生中，也是學到了應該正確面對事情的態度。

另外，我在此還要感謝實驗室的學弟培育。因為研究領域的相似，在相互討論的過程當中，令我得到了很多非常寶貴的意見，也觸發了很多的想法。

然後，還要感謝同級的震軒、義琅，以及博班學長至鴻在研究以及修課方面的協助和陪伴，讓我的碩士生涯過的更加愉快。

最後我要深深感激的，就是在碩士生涯中支持我的家人，因為有了你們的支持，我才有可能拿到這個學位。

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivations . . . . .	3
1.2	Our Contributions . . . . .	6
1.3	Organization of this Thesis . . . . .	7
<b>2</b>	<b>Preliminaries</b>	<b>8</b>
2.1	Modified Nodal Analysis . . . . .	8
2.2	Direct and Iterative Methods for Solving Linear Equation . . . . .	12
2.3	Multigrid Method . . . . .	16
2.3.1	Traditional Algebraic Multigrid Method . . . . .	18
2.3.2	Aggregation Algebraic Multigrid Method . . . . .	24
<b>3</b>	<b>Aggregation-Based Algebraic Multigrid</b>	<b>26</b>
3.1	Problem Formulation . . . . .	26
3.2	Flowchart of Our Proposed Method . . . . .	27
3.3	Overview of Our Approach and Previous Works . . . . .	28
3.4	Derivation of System Equations . . . . .	29
3.5	Aggregation-Based AMG Cycle Construction . . . . .	32
3.5.1	Aggregation Algorithm . . . . .	32
3.5.2	Practicability of Aggregation AMG Method to Power Network Analysis . . . . .	36
3.5.3	Global Error Estimation . . . . .	37
3.5.4	Matrix Compensation Algorithm . . . . .	38
3.5.5	Aggregation-Based AMG Coarse Grid Construction . . . . .	40
3.6	Aggregation-Based Multilevel Solver . . . . .	40
<b>4</b>	<b>Experimental Results</b>	<b>42</b>
<b>5</b>	<b>Conclusions</b>	<b>48</b>
<b>A</b>	<b>Property of Error Propagation</b>	<b>49</b>

# List of Figures

1.1	An Example of $IR$ Drop Voltage Noise . . . . .	2
1.2	The Structure of Power Network . . . . .	4
2.1	$RLC$ Circuit . . . . .	9
2.2	$RLC$ Circuit After Step 1,2 . . . . .	9
2.3	Decompositions of Matrix $A$ . . . . .	14
2.4	Smoothing of Random Error by Gauss-Seidel Iteration . . . . .	15
2.5	Standard Geometry Coarsening of a Regular Mesh . . . . .	16
2.6	Two-Level Solution Method . . . . .	18
2.7	The Multigrid V-Cycle . . . . .	19
2.8	Example of Color Scheme . . . . .	21
2.9	The Flowchart of Traditional AMG . . . . .	24
3.1	Flowchart of Our Proposed Method . . . . .	27
3.2	Comparison between Our Method and Previous Methods . . . . .	30
3.3	Original System Matrix . . . . .	35
3.4	Example of Aggregation . . . . .	36
3.5	System Matrix After Aggregation . . . . .	37
3.6	Structure of Aggregation of the Size of 3 . . . . .	38
3.7	Coarse Grid Construction . . . . .	40
4.1	Run Time versus Circuit Size . . . . .	44
4.2	Cpu Time versus Circuit Size . . . . .	45
4.3	Non Zero Term versus Circuit Size . . . . .	46
4.4	Total Cycles versus Circuit Size . . . . .	47

# List of Tables

2.1	Algorithm of Color Scheme . . . . .	20
3.1	Algorithm of Aggregation . . . . .	34
3.2	Algorithm of AggreConstruct . . . . .	34
3.3	Algorithm of Matrix Compensation . . . . .	39
4.1	Error percentage of RLKC circuits . . . . .	42
4.2	Runtime of RLKC circuits. “×” denotes this methodology failed. . . . .	43
4.3	Speed up of AbAMG compared to other methods . . . . .	43
4.4	Comparison between AbAMG and standard AMG . . . . .	44





# Chapter 1

## Introduction

This chapter gives an introduction of this thesis. Since our research topic is to develop an efficient analysis tool for on-chip power/ground distribution network. We discuss the basic concepts of on-chip power/ground distribution network in the beginning. The role of an on-chip power/ground distribution network is to supply stable voltage references to the on-chip circuitry and ensuring reliable operation of today's high performance microprocessors. However, as the ultra deep sub-micron technology scales below  $0.18 \mu m$ , circuits with increasingly higher speed are being integrated with increasingly higher density. Higher device densities and faster switching frequencies cause large switching currents to flow in the on-chip power/ground distribution network, and will cause larger voltage fluctuations due to  $IR$  drop and  $Ldi/dt$  noise which degrade the performance and reliability of the circuit. High average currents flowing through the power/ground distribution network may cause the undesirable electromigration effect which will degrade the circuit's reliability. The descriptions of the voltage fluctuations and electromigration effect are shown in the following [1] [2] [3]

- **$IR$  drop voltage noise:** An example of the  $IR$  drop voltage noise is shown in Fig. 1.1.  $IR$  drop mainly results from the resistance of the on-chip power network where  $I$  represents current. If large current flows through the power network, an un-acceptable voltage drop occurs. Large  $IR$  drop results from large current must be handled carefully during design period.

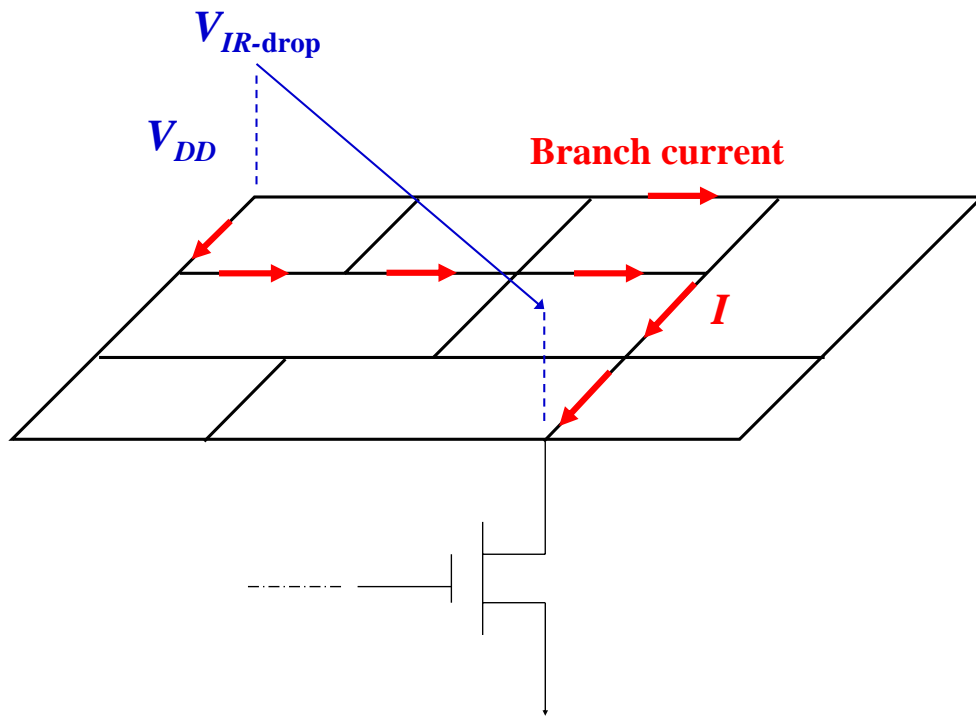


Fig. 1.1: An Example of  $IR$  Drop Voltage Noise

- **$Ldi/dt$  voltage noise:**  $Ldi/dt$  noise occurs from a sudden change of current flowing through a power network. With higher operation frequency of today's high performance IC design,  $Ldi/dt$  voltage noise becomes larger.  $Ldi/dt$  noise also results from mutual inductance coupling effect. Two parallel wires may cause large  $Ldi/dt$  voltage drop noise with each other.
- **Electromigration:** Electromigration effect results from a conductor with too much current flowing through it and hence, the displacement of metal atoms due to electron-flux. This behavior will cause shorts or opens in the metal lines, and degrading the circuit's reliability.

As the supply voltage scaling to control the power dissipation in the circuit [4], the noise margin of the on-chip power/ground distribution network are sensitive to the voltage fluctuations and excessive voltage drops may cause the functional failures of the circuit.

With these reasons, the analysis of on-chip power/ground distribution network has become a critical issue of today's high performance IC design. In order to precisely predict the voltage distribution and correctly simulate the behavior of the on-chip power/ground distribution network, we model the active devices between power and ground distribution networks as time-varying current sources and gate capacitances [14]. By the way, the power distribution network and ground distribution network can be separated for simplicity. We focus on the simulation of power distribution network in this thesis and this method can be extended to the ground distribution network analysis in the same manner. The power distribution network is usually an irregular mesh and is modeled as  $RLKC$  segments where  $R$ ,  $L$ , and  $C$  represent the stamping matrix of resistors, inductances, and capacitances, and  $K$  represents the susceptance matrix [22] [23] which is defined as the inversion of  $L$ . The structure of power distribution network is shown in Fig. 1.2. Since mutual inductance coupling has long range effect [5] which means that the coupling between two parallel wire segments decays very slowly with their separated distance, and generates a dense matrix of  $L$ , for simplicity, the mutual inductances coupling effects are not shown in Fig. 1.2.

The rest of this chapter is organized as following. In Section 1.1, we compare the existing analysis methods for power network analysis and state our research motivation. Our contributions and the organization of this thesis are presented in Section 1.2 and 1.3.

## 1.1 Motivations

In this section, we compare the existing analysis methods for power network and state our research motivation. With the ultra deep sub-micron technology, several features of chips (higher operating frequency, larger number of transistors, smaller feature sizes of transistors and lower supply voltages) have made the integrity issues of power delivery network become a key issue of high performance designs [6][7][8]. Generally, the power delivery network contains enormous amount of circuit elements and such huge size requires

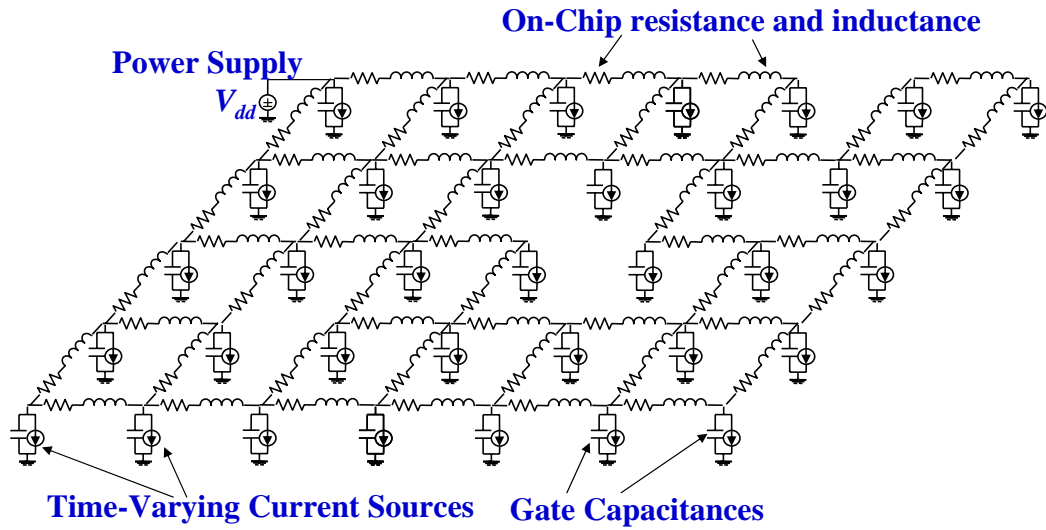


Fig. 1.2: The Structure of Power Network

highly efficient analyzers. Thus, the general circuit solvers such as SPICE by using direct methods are not suitable for the power delivery analysis. In the past years, various efficient methods have been proposed for the power delivery network analysis. The preconditioned conjugate gradient (PCG) method is applied for solving power grid analysis in [9]. The hierarchical methods are developed in [10][11]. The improved extended Krylov subspace (IEKS) method developed in [11] extends the model order reduction technique to deal with time-varying current sources without the moment shifting procedure. Multigrid-like methods are developed in [12][13] to map the original problem to a reduced system with smaller size by using the circuit's geometry properties. However, these frameworks proposed in [12][13] are hard to handle the coupling effects of mutual inductances. Hence, an adaptive algebraic multigrid (AMG) method is used in [14] to analyze the power network. It reformulates the system matrix and views the problem as an algebraic problem which doesn't need the geometry information. With these properties, the AMG based method can handle mutual inductance coupling effects.

The mapping operators of AMG in [14] are determined by locally calculating the

equation,  $Ae \approx 0$ , where  $A$  is the system matrix and  $e$  represents the error vector. The quality of mapping operator strongly depends on the choice of coarse grids and the constructed mapping operators only contain the local information of  $A$ . The mapping operators of AMG may lose a few of important error terms because of the inadequate choice of coarse grids, hence, degrade the convergence rate. Therefore, an adaptive choice method of coarse grids is developed in [14] to improve the above undesirable behavior. However, this method needs to construct the mapping operators at each time step and may boost the CPU time. To solve this problem, our aggregation-based algebraic multigrid (AbAMG) method contains a global mapping operator construction procedure.

The idea of our mapping operator construction is based on the aggregation AMG method used in [25] [26][27]. Aggregation methods originated in economics [28], where similar products are considered together instead of individually. This procedure allows significant reduction in the problem size, and maintaining accurate representation of the overall behaviors. In multigrid terminology, the coarse grid is selected as a collection of subsets of the fine grid. An algebraic partition is performed to the original fine grid and the original system matrix is partitioned into several aggregated sub-matrices. The mapping operators of aggregation AMG method are constructed from the system's global eigen-decomposition property. Generally, the error in the direction of an eigenvector associated with a large eigenvalue is rapidly reduced by relaxation and the error in the direction of an eigenvector associated with a small eigenvalue is reduced by a factor that may approach 1 as the eigenvalue approaches 0 [20]. The eigenvectors associated with small eigenvalues of each sub-matrix are calculated to approximate the smooth error components of the original system matrix and the mapping operator  $P$  is composed by these eigenvectors. With accurate calculation of these eigenvectors, the mapping operator can project the original system to a better transformed system than traditional AMG in [14] and achieving better convergence rate. However, the eigen-decomposition complexity of the aggregated sub-matrix grows rapidly with the matrix size and may boost the CPU time.

In Chapter 3, we will show that the system matrix of the power delivery network analysis problem has resistance dominate property when determining the aggregation. The maximum matrix size of each sub-matrix of the original system is less than 4 and the analysis problem has excellent property for aggregation AMG. The mapping operator construction of our proposed AbAMG method is based on the concept of aggregation AMG and an innovative matrix compensation algorithm with a global error estimation procedure is performed to further improve the quality of the mapping operators. The mapping operator construction procedure of AbAMG is independent of the choice of coarse grids and it only needs to be performed once for all time steps. With these properties, the AbAMG method can construct better mapping operators than the traditional AMG method, and achieving better performance for solving the power delivery network problem.

## 1.2 Our Contributions



This section we discuss our contributions in the following aspects

- **The practicability of aggregation AMG to power network analysis:** In this thesis, we discuss the practicability of the aggregation AMG to power network analysis problem. Although the computation complexities of the eigen-decomposition procedures of the aggregated sub-matrices grow rapidly with the matrix size. We discuss the resistance dominate property when determining the aggregation of the system matrix of the power delivery network. The maximum aggregation size is only of 4 and the aggregation AMG method is efficient to analysis the power delivery network problem. We discovery the practicability of the aggregation AMG to power network analysis and provide a new idea of AMG method to analysis the power delivery network problem.
- **Global error estimation:** Although the real error of the analysis system is unknown, we propose a global error estimation method in this thesis. By applying the

relaxation process to a problem with known solution, we can obtain the information about the troublesome error. The homogeneous equation,  $Ax = 0$ , serves us well for this purpose where  $A$  is the system matrix and  $x$  is the solution vector. Our global error estimation process begins by applying iterative method  $i$  times to the homogeneous equation with a random initial guess  $x^0$ . The resulting solution vector  $x^i$  can provide us the information about global error distribution. The error vector contains information of the algebraic property of system matrix  $A$  and improves the construction of AMG inter-grid mapping operators. The detail discussion of global error estimation and mapping operators construction will be shown in Section 3.4.

- **Global mapping operator construction:** In contrast to AMG, AbAMG constructs the mapping operator from the global information of  $A$ . An aggregation algorithm is performed to partition the original system matrix  $A$  into many sub-systems and the approximated eigenvectors of  $A$  can be calculated from each sub-system locally. Since each sub-system is not totally independent and has weak connections with each other. An algebraic matrix compensation algorithm is performed to catch the weak connection effects. The error vector generated during the global error estimation is used in the matrix compensation algorithm. After the compensation algorithm, we can construct a better mapping operator from the global information of system and get better performance than standard AMG.

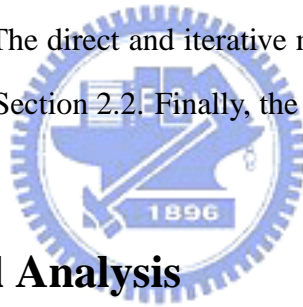
### 1.3 Organization of this Thesis

The rest of this thesis is organized as follows. Chapter 2 introduces the Modified Nodal Analysis method and the mathematical background of multigrid and traditional AMG. Chapter 3 describes our proposed algorithm flow. In Chapter 4, we compare the experimental results of AbAMG, traditional AMG and IEKS methods. Finally, we give a conclusion in Chapter 5.

# Chapter 2

## Preliminaries

This chapter introduces several mathematic background knowledge that will be used in this thesis. The Modified Nodal Analysis (MNA) method [15] is illustrated in Section 2.1. By using MNA, we can get the system equation, and modeling the original problem as a linear algebraic problem. The direct and iterative methods for solving a linear algebraic problem are discussed in Section 2.2. Finally, the theory of the multigrid method is presented in Section 2.3.



### 2.1 Modified Nodal Analysis

MNA is very useful for large circuit analysis and is easier to implement algorithmically on a computer. The analysis principles of it and an example for  $RLC$  circuit are shown below.

- **Principles of MNA:** To apply the MNA to a circuit with  $n$  nodes,  $m$  voltage sources and  $k$  inductances. We apply the following steps.
  - **Step 1:** Name the  $n$  nodes and currents through each current source.
  - **Step 2:** Name the currents through each voltage source and inductance.
  - **Step 3:** Apply Kirchoff's current law to the  $n$  nodes. We take currents out of a node to be positive.
  - **Step 4:** Write an equation for the voltage each voltage source and inductance.



– **Step 5:** Solve the system of  $n + m + k$  unknowns.

**Example:** Consider a *RLC* circuit shown below:

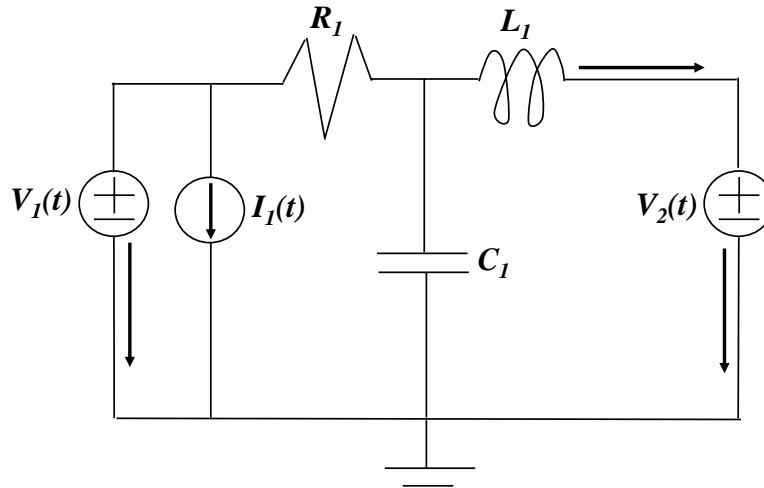


Fig. 2.1: *RLC* Circuit

Apply step 1 and step 2:

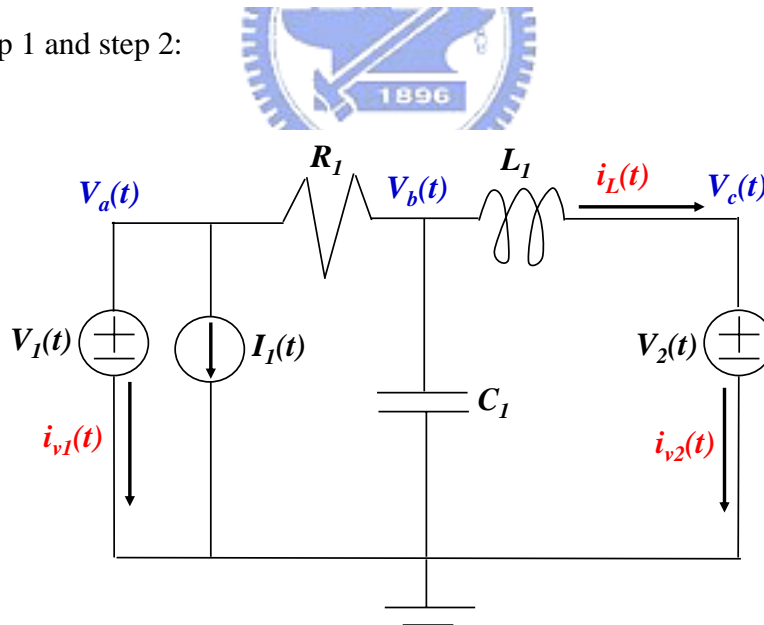


Fig. 2.2: *RLC* Circuit After Step 1,2

Apply step 3:

$$\text{Node } a : i_{v_1}(t) + I_1(t) + \frac{V_a(t) - V_b(t)}{R_1} = 0 \quad (2.1)$$

$$\text{Node } b : \frac{V_b(t) - V_a(t)}{R_1} + C_1 \frac{dV_b(t)}{dt} + i_L(t) = 0 \quad (2.2)$$

$$\text{Node } c : i_{v2}(t) - i_L(t) = 0 \quad (2.3)$$

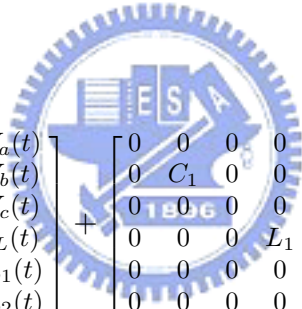
Apply step 4:

$$V_b(t) - V_c(t) = L_1 \frac{di_L(t)}{dt} \quad (2.4)$$

$$V_a(t) = V_1(t) \quad (2.5)$$

$$V_c(t) = V_2(t) \quad (2.6)$$

Apply step 5:



$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 & 1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a(t) \\ V_b(t) \\ V_c(t) \\ i_L(t) \\ i_{v1}(t) \\ i_{v2}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a(t)' \\ V_b(t)' \\ V_c(t)' \\ i_L(t)' \\ i_{v1}(t)' \\ i_{v2}(t)' \end{bmatrix} = \begin{bmatrix} -I_1(t) \\ 0 \\ 0 \\ 0 \\ V_1(t) \\ V_2(t) \end{bmatrix} \quad (2.7)$$

Rearranging Equation (2.7), we can get the following equations:

$$\begin{bmatrix} \frac{1}{R_1} & -\frac{1}{R_1} & 0 & 0 & 1 & 0 \\ -\frac{1}{R_1} & \frac{1}{R_1} & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a(t) \\ V_b(t) \\ V_c(t) \\ i_L(t) \\ i_{v1}(t) \\ i_{v2}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_a(t)' \\ V_b(t)' \\ V_c(t)' \\ i_L(t)' \\ i_{v1}(t)' \\ i_{v2}(t)' \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} I_1(t) \\ V_1(t) \\ V_2(t) \end{bmatrix} \quad (2.8)$$

From Equation (2.8), the MNA circuit equations of a linear  $RLC$  circuit can be represented as following:

$$\hat{G}x(t) + \hat{C} \frac{d}{dt} x(t) = Bu(t) \quad (2.9)$$

where

$$\begin{aligned}\hat{G} &= \begin{bmatrix} G & -A_l^T & -A_{V_E}^T \\ A_l^T & 0 & 0 \\ A_{V_E}^T & 0 & 0 \end{bmatrix} \\ \hat{C} &= \begin{bmatrix} C & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ x(t) &= \begin{bmatrix} v(t) \\ i_l(t) \\ i_{V_E}(t) \end{bmatrix}\end{aligned}$$

$v(t)$  corresponds to the unknown nodal voltages.  $i_l(t)$  and  $i_{V_E}(t)$  correspond to the branch currents flowing through inductors and independent voltage sources.  $G$ ,  $C$  and  $L$  represent the stamping matrices of the resistors, the conductors and the inductors.  $A_l$  and  $A_{V_E}$  correspond to the coefficient matrices related to the inductors and the independent voltage sources.  $u(t)$  is the vector of independent voltage sources and the independent current sources.  $B$  is the coefficient matrix related to  $u(t)$ . Integrating Equation (2.9) from time  $t$  to  $(t + h)$ , we can get the following equation

$$\hat{G} \int_t^{t+h} x(t) dt + \hat{C} \int_t^{t+h} \frac{dx(t)}{dt} dt = B \int_t^{t+h} u(t) dt \quad (2.10)$$

Applying trapezoidal approximation [15] with time step  $h$  to Equation (2.10), we have

$$\hat{G} \left( \frac{x(t+h) + x(t)}{2} \right) h + \hat{C} (x(t+h) - x(t)) = B \left( \frac{u(t+h) + u(t)}{2} \right) \quad (2.11)$$

Reformulating Equation (2.11), we have

$$\left( \hat{G} + \frac{2}{h} \hat{C} \right) x(t+h) = - \left( \hat{G} - \frac{2}{h} \hat{C} \right) x(t) + B(u(t+h) + u(t)) \quad (2.12)$$

Equation (2.12) can be viewed as a linear algebraic problem of  $Ax = b$ , where  $A$  is equal to  $\left( \hat{G} + \frac{2}{h} \hat{C} \right)$ . The solution of  $x$  can be solved iteratively with time step  $h$  and each time step encounters an  $Ax = b$  problem which can be solved by direct or iterative methods.

## 2.2 Direct and Iterative Methods for Solving Linear Equation

### Direct Methods:

When solving a linear algebraic problem

$$Ax = b \quad (2.13)$$

where  $A$  is  $n * n$  matrix,  $x$  and  $b$  are  $n * 1$  vectors. The simplest direct method is to calculate the inverse matrix of  $A$  and the solution of  $x$  is

$$x = A^{-1}b \quad (2.14)$$

Another direct method is the *LU decomposition method* [16]. Substituting the  $A$  matrix into a product of lower- and upper-triangular matrices:

$$A = \begin{bmatrix} \ddots & & 0 \\ L & \ddots & \\ & & \ddots \end{bmatrix} \begin{bmatrix} \ddots & & U \\ 0 & \ddots & \\ & & \ddots \end{bmatrix} \quad (2.15)$$

We have

$$LUx = b \quad (2.16)$$

In order to solve Equation (2.16), we substitute

$$y = Ux \quad (2.17)$$

such that

$$Ly = b \quad (2.18)$$

So we first solve the Equation (2.18) by *Forward Substitution* to obtain  $y$  and then solve the Equation (2.17) by *Back Substitution* to get the final solution  $x$ .

Direct methods can always get the answer of a linear algebraic problem with high computational complexity. Most direct methods have computational complexities in proportion to  $n^3$  [16]. With the tremendous amounts of transistors in today's VLSI design,

direct analysis methods are prohibitive due to computational complexity.

**Iterative Methods:**

In contrast to direct methods, iterative methods solve a linear algebraic problem iteratively [17]. It gets the answer after several iterations. Considering a linear algebraic problem  $Ax = b$ , iterative method splits the matrix  $A$  into the form

$$A = M - N \tag{2.19}$$

where  $M$  and  $N$  are  $n * n$  matrices. So Equation (2.13) becomes

$$(M - N)x = b \tag{2.20}$$

Reformulating Equation (2.20), we have

$$x = M^{-1}Nx + M^{-1}b \tag{2.21}$$

From Equation (2.19), we can get

$$M^{-1}N = I - M^{-1}A \tag{2.22}$$

Substituting Equation (2.22) into Equation (2.21), we can get a standard iterative formula:

$$x^{i+1} = (I - M^{-1}A)x^i + M^{-1}b \tag{2.23}$$

where  $x^i$  is the value of  $x$  after  $i - th$  iterations.

Substituting the real solution  $x$  to the both sides of Equation (2.23), we can get the error propagation equation as the following

$$e^{i+1} = (I - M^{-1}A)e^i \tag{2.24}$$

where  $e^i = x - x^i$

The matrix  $A$  can be decomposed as  $A = D + L + U$  where  $D$ ,  $L$ , and  $U$  are the matrices of the diagonal, lower triangular, and upper triangular elements of  $A$ . The decompositions of  $A$  are shown in Fig. 2.3. For the Jacobi and Gauss-Seidel iterative methods, the  $M$  matrix are substituted by  $D$  and  $D + L$  [17], and the iterative solving scheme of the Jacobi and Gauss-Seidel iterative methods can be showing as following

- **Jacobi Relaxation:**

$$x^{i+1} = x^i + D^{-1}r^i \quad (2.25)$$

- **Gauss-Seidel Relaxation:**

$$x^{i+1} = x^i + (D + L)^{-1}r^i \quad (2.26)$$

where  $r^i$  means the residual after  $i$  times iterations and is equal to  $b - Ax^i$

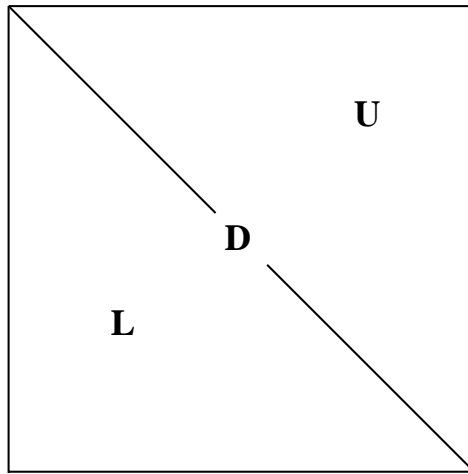


Fig. 2.3: Decompositions of Matrix A

Iterative method converges to the correct answer after several iterations and the computational complexity is often  $n \log n$  per iteration [17]. The efficiency of iterative method depends on how fast it can converge to the correct answer. From Appendix A, we can know that the error in the direction of an eigenvector associated with a large eigenvalue is rapidly reduced by relaxation and the error in the direction of an eigenvector associated with a small eigenvalue is reduced by a factor that may approach 1 as the eigenvalue approaches 0. The smooth error components must be solved by efficient solution methods.

An example of iterative method is shown in Fig. 2.4. We apply the Gauss-Seidel iterative method to a power network analysis problem of dimension 260 with random initial error. The supply voltage of this power network problem is 1V. The initial error of

each node is shown in (a). The error of each node after 5, 20, 50 times of Gauss-Seidel relaxations are shown in (b), (c), (d). The error components that are easy to be eliminated by the iterative methods are defined as the oscillatory errors and the error components that are hard to be eliminated by the iterative methods are defined as the smooth errors. We can find that the Gauss-Seidel relaxations eliminates the error slowly and some nodes have large errors which need 50 times of iterations to eliminate. In order to solve this stalling behavior and make the problem converge faster, the multigrid method is proposed [18].

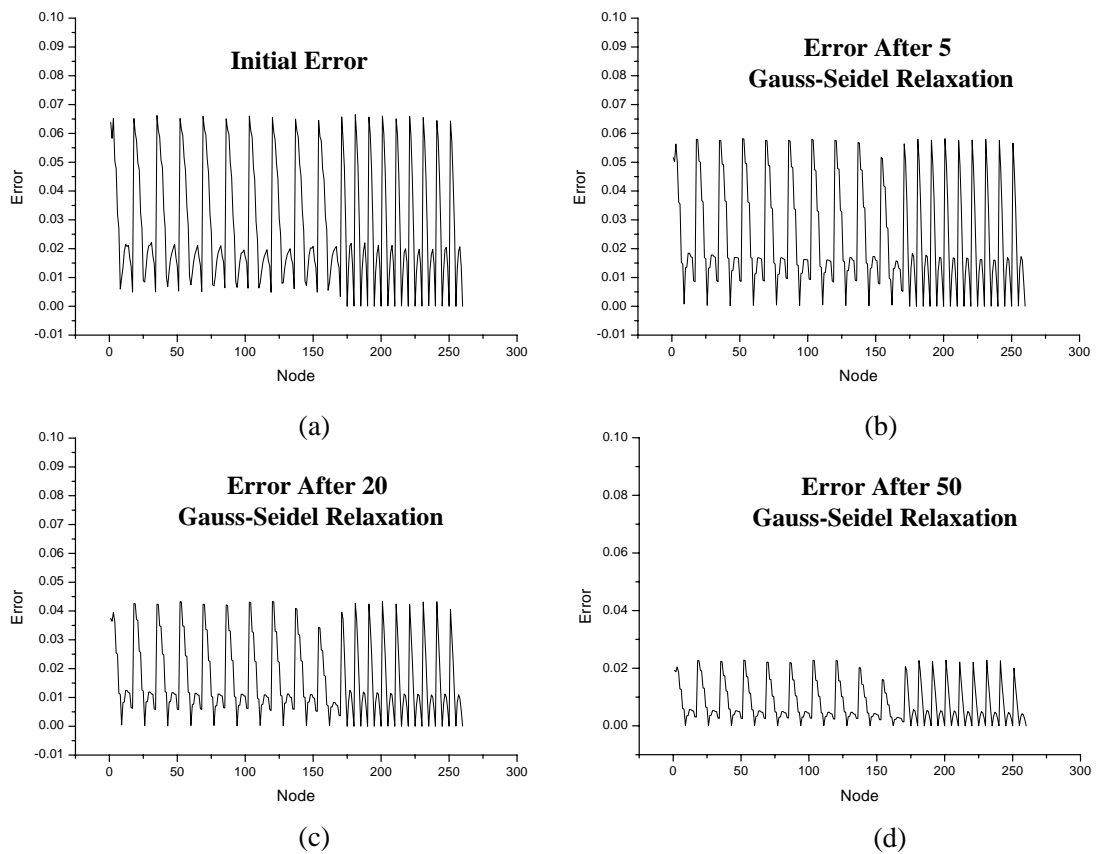


Fig. 2.4: Smoothing of Random Error by Gauss-Seidel Iteration

## 2.3 Multigrid Method

The earliest multigrid method is geometry multigrid (GMG) introduced by Brandt in 1973. It uses the geometry properties of system to construct a complementary multilevel structure to overcome the stalling behavior in general iterative methods. The complementary multilevel structure is developed with two main ideas.

First, we know that the iterative method quickly eliminates oscillatory errors and we must look for a process that can efficiently eliminate smooth errors. Since the global computational complexity is proportional to the problem size, we transfer the original problem from the original domain (fine domain) into a coarser domain such that we can attempt to solve the problem there with cheaper computational cost. This domain transformation is called the coarsening procedure and is determined by the geometry properties of system in GMG. A standard geometry coarsening of a regular mesh is shown in Fig. 2.5. The original analysis system is fine grid of larger dimension and the transformed system is coarse grid of smaller dimension.

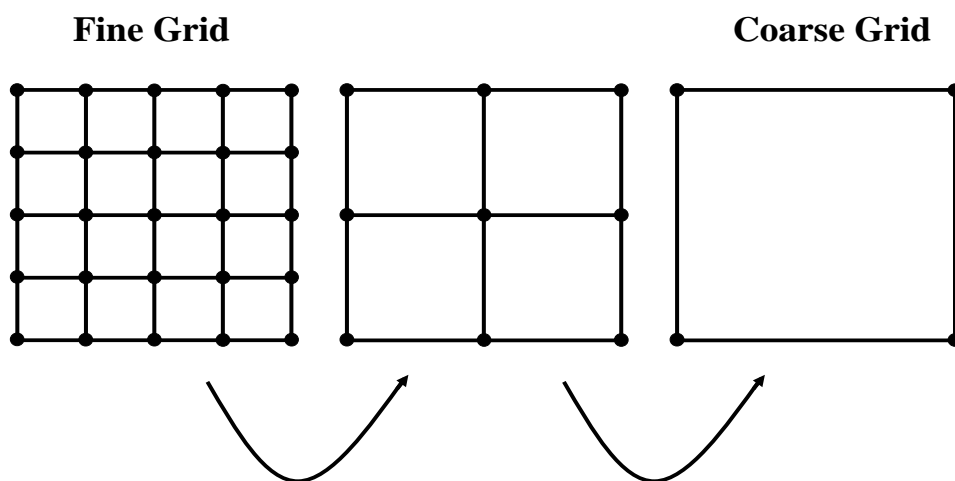


Fig. 2.5: Standard Geometry Coarsening of a Regular Mesh



Second, another difficulty is about how to represent the error on the coarser domain since it is the quantity we do not know. Considering an algebraic equation,  $Ax = b$ , with an approximation,  $\hat{x}$ , the residual is defined as

$$r = b - A\hat{x} = A(x - \hat{x}) = Ae \quad (2.27)$$

So, although the error is unknown, it can be solved by using the residual equation,  $Ae = r$ . If we calculate the residual  $r$  on the original fine domain and project this residual to the transformed coarse domain. The residual equation of the coarse domain can be solved and an error correction term  $e^c$  can be obtained to correct the solution.

Based on these ideas, we can construct a two-level solution method as shown in Fig. 2.6. First, we apply the iterative method to the equation  $Ax = b$  to eliminate the oscillatory error components on the fine grid of dimension  $N$ . This step is also called the *relaxation* step and the residual on the fine grids is calculated by  $r = b - Ax$ . Then, the residual is *restricted* to the coarse grids with a smaller dimension  $M$  by  $r^c = Rr$ , and the coarse grid operator is constructed by the Galerkin operator  $A_c = RAP$ . Here,  $R$  is a  $M \times N$ ,  $P$  is a  $N \times M$  matrix and  $R = P^T$ . On the coarse grids, the residual equation,  $A^c e^c = r^c$ , is solved and the error correction term  $e^c$  is *interpolated* to the fine grids by  $e = Pe^c$ . The smooth error components not eliminated well by relaxation on the fine grid can be eliminated by the error correction term  $e^c$ . A complementary two-level solution scheme can be constructed to overcome the stalling behavior of smooth error components in general iterative methods. The correct solution is obtained by  $x = x + e$ , and a post-relaxation step is applied on the fine grids to ensure that the oscillatory error is not introduced through the coarse-grid correction step.

Applying the two-level solution method recursively, a multilevel solution method is constructed and the coarsest residual equation can be solved with cheaper computational cost. The multigrid V-cycle is shown in Fig. 2.7. The fine grid is labeled as level 1 and the coarsest grid is labeled as level L. A relaxation step is first applied in the fine grid of level 1 and the residual is restricted to the next level. These steps are repeated until the

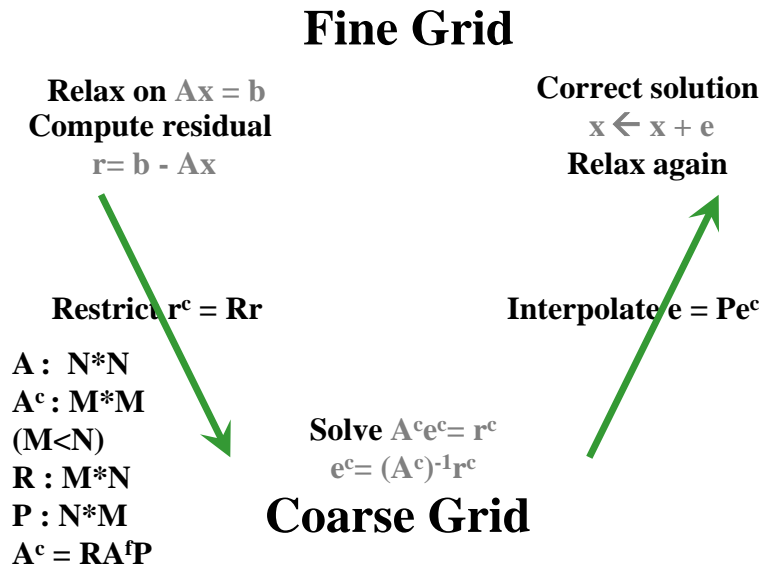


Fig. 2.6: Two-Level Solution Method

coarsest level is reached and the residual equation of level  $L$  is solved to get the correction error term  $e^L$ . Then, the error term  $e^L$  is interpolated to the fine grid of level 1 and the post-relaxation step is performed at each level.

Multigrid method constructs a complementary multilevel structure which can efficiently eliminate all error components. The efficiency of multigrid method depends on how to choose the coarse grid and determine the intergrid mapping operators  $P$  and  $R$ . The mapping mechanism of GMG is easily determined with regular mesh but hard with irregular mesh. In order to develop a more robust solving method, an algebraic multigrid method is proposed in Section 2.3.1.

### 2.3.1 Traditional Algebraic Multigrid Method

AMG method was first introduced by Brandt in [19]. It is developed for solving problems with irregular or unknown geometry properties. In contrast to GMG, AMG uses only information from the system matrix.

In this section, we focus our discussion on the traditional AMG. Considering an alge-

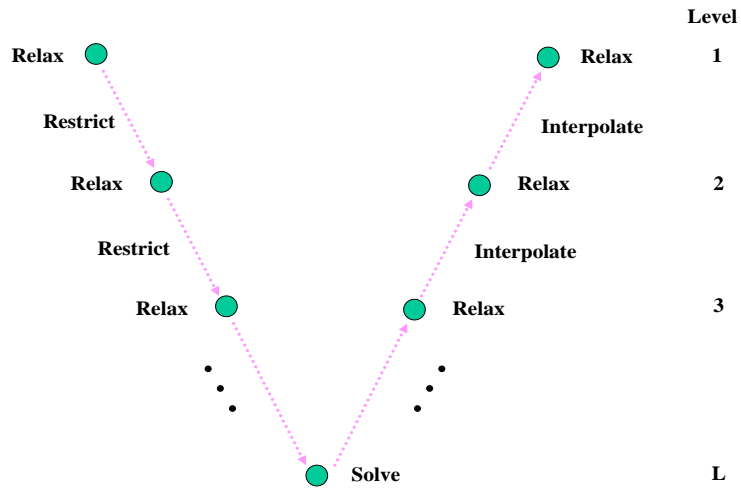


Fig. 2.7: The Multigrid V-Cycle

braic equation,  $Ax = b$ , AMG determines the inter-grid mapping operators, coarse grid from the matrix  $A$  and the graph of it. Each row of the matrix  $A$  can be represented as a node and its connection edge in a graph. The coefficients of the matrix  $A$  represent the connections of the graph. For example, if  $|a_{ij}| = 0$ , there is no edge between node  $i$  and  $j$  in the graph of  $A$ . If  $|a_{ij}| \geq \theta|a_{ii}|$ , we say that node  $j$  strongly influences  $i$ . If  $|a_{ij}| \leq \theta|a_{ii}|$ , node  $j$  weakly influences  $i$ . Here,  $\theta$  is a coefficient from 0 to 1 and is often chosen to be 0.25.

With these definitions, we can construct the matrix graph of  $A$  and determine the coarse grid by the *color scheme* algorithm [18]. This method begins by assigning a measure to each node  $i$  of its potential quality to be a coarse node. The weight of node  $i$  is determined by counting the number of nodes strongly influenced by node  $i$ . Then, we choose the node  $i$  with maximum weight to be the starting coarse grid since it has good potential to approximate other nodes. The nodes strongly influenced by node  $i$  are defined as fine nodes since they can be approximated well by node  $i$ . It's logical that the nodes strongly influence the new fine nodes should be defined as coarse nodes since they can approximate the new fine nodes well. Thus, we increase the weights of the nodes strongly influence the new fine nodes by 1 and repeat the coarse node selection until all nodes of

<b>Algorithm of Color Scheme</b>	
<b>Input:</b> <i>The Graph of System Matrix A of Nodes 1, 2, ..., n and the Related Weights <math>w_1, w_2, \dots, w_n</math> of These Nodes</i>	
<b>Output:</b> <i>The Sets of Coarse and Fine Nodes</i>	
1	<b>Begin</b>
2	NodeCounter=0
3	<b>While</b> NodeCounter!=n
4	MaxWeight=0, StartNode=1
5	<b>For</b> each node $i$
6	<b>If</b> node $i$ is not defined as a coarse or fine node
7	<b>If</b> $w_i > \text{MaxWeight}$
8	<b>Then</b> MaxWeight= $w_i$ , StartNode= $i$
9	<b>EndFor</b>
10	StartNode is defined as a coarse node, NodeCounter++
11	<b>For</b> each node $j$ that is strongly influenced by StartNode
12	<b>If</b> node $j$ is not defined as a coarse or fine node
13	node $j$ is defined as a fine node, NodeCounter++
14	<b>For</b> each node $k$ that strongly influences node $j$
15	<b>If</b> node $k$ is not defined as a coarse or fine node
16	$w_k + +$
17	<b>EndFor</b>
18	<b>EndFor</b>
19	<b>End.</b>

Table 2.1: Algorithm of Color Scheme

the matrix graph of A are defined as coarse or fine nodes. The algorithm of color scheme is shown in Table 2.1 and an example of it is given

Fig. 2.8 shows an example of color scheme. The description of each step is shown below:

- **Example of Color Scheme:**

- **Step a:** A matrix graph of A is given with node number 1 to 14.
- **Step b:** The weight of each node  $i$  is determined.
- **Step c:** Node 3 is defined as the starting coarse node with maximum weight of 4 and node 1, 4, 13, 14 are defined as fine nodes. The weights of node 2, 5, 9, 11 are increased.

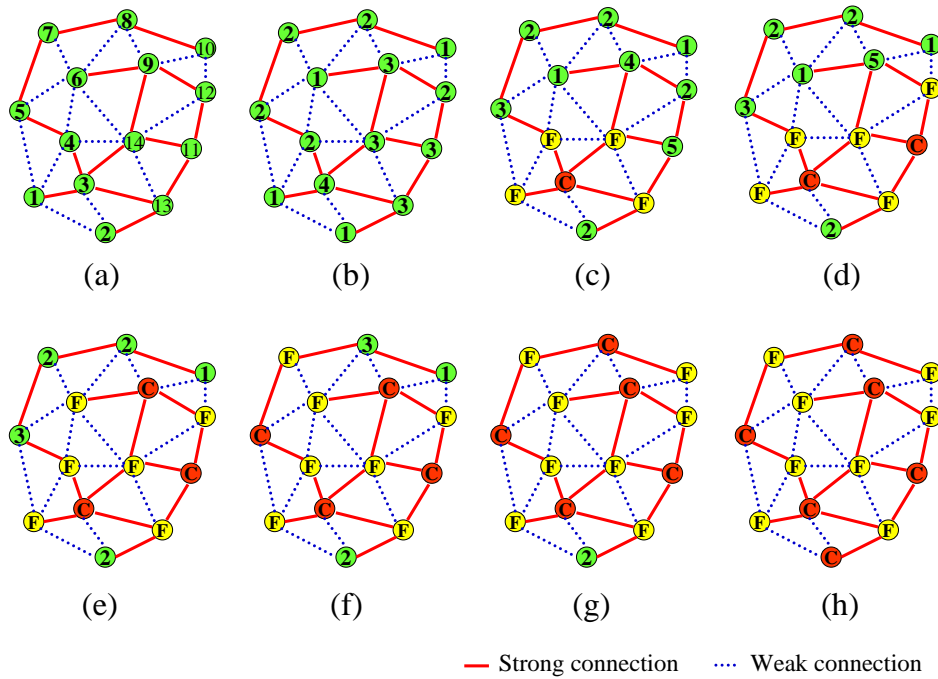



 Fig. 2.8: Example of Color Scheme

- **Step d:** Node 11 is defined as the new coarse node with maximum weight of 5 and node 12 is defined as new fine node. The weight of node 9 is increased by 1.
- **Step e:** Node 9 is defined as the new coarse node with maximum weight of 5 and node 6 is defined as new fine node.
- **Step f:** Node 5 is defined as the new coarse node with maximum weight of 3 and node 7 is defined as new fine node. The weight of node 8 is increased by 1.
- **Step g:** Node 8 is defined as the new coarse node with maximum weight of 3 and node 10 is defined as new fine node.
- **Step h:** Node 2 is defined as the new coarse node with maximum weight of 2. All nodes in the matrix graph of  $A$  are defined and the color scheme finishes.

By using the color scheme algorithm, we can get the coarse grid and every fine node  $i$  can be approximated well by the coarse nodes strongly influence  $i$ . However, the selected coarse grid only considers local connections of each node  $i$  and may choose bad coarse nodes that will decrease the convergence rate. To overcome this defect, we propose a global mapping operator construction to build the global-considering coarse grids.

To further discuss the inter-grid mapping operator, we continue the discussion of inter-grid transfer operator. Since the key to the efficiency of the multigrid method depends on the complementarity of the relaxation and coarse-grid correction steps. We begin the discussion of inter-grid transfer operator with the property of algebraic smoothness,  $(Ae)_i \approx 0$  which means that residual become small after several iterative iterations for each row  $i$ . The equation can be rewritten as

$$a_{ii}e_i \approx - \sum_{j \neq i} a_{ij}e_j \quad (2.28)$$

We define that the DOFs of fine grid is  $C \cup F$ , where  $C$  is the set of coarse-level nodes and  $F$  is the set of remaining fine-level nodes. Rewriting Equation (2.28), we can get

$$a_{ii}e_i \approx - \sum_{j \in C_i} a_{ij}e_j - \sum_{k \in F_i} a_{ik}e_k \quad (2.29)$$

where  $C_i = C \cap N_i$ ,  $F_i = F \cap N_i$ , and  $N_i$  means the neighboring nodes of node  $i$ .

For further discussion, we divide the  $F_i$  into  $F_i^s$  and  $F_i^w$  where  $F_i^s$  is the set of nodes which strongly influence  $i$  in  $F_i$ , and  $F_i^w$  is the set of nodes which weakly influence  $i$  in  $F_i$ . Equation (2.29) can be rewritten as

$$a_{ii}e_i \approx - \sum_{j \in C_i} a_{ij}e_j - \sum_{k \in F_i^s} a_{ik}e_k - \sum_{m \in F_i^w} a_{im}e_m \quad (2.30)$$

From Equation (2.30), we can try to define an interpolation structure since the  $e_i$  for each node is approximated by the neighboring coarse nodes  $C_i$  and fine nodes  $F_i$ . If we can approximate the value of  $F_i$  as a sum of the values of  $C_i$ ,  $e_i$  can be approximated by  $C_i$  only and an interpolation can be defined.

Since the values of  $F_i^s$  nodes are large compared to  $a_{ii}$ , we approximate  $e_k$  by  $C_i$  in the following form

$$e_k \approx \frac{\sum_{j \in C_i} a_{kj} e_j}{\sum_{l \in C_i} a_{kl}} \quad (2.31)$$

Substituting Equation (2.31) into Equation (2.30) and adding the values of  $F_i^w$  points into  $a_{ii}$ , we can get the following equation

$$\left( a_{ii} + \sum_{m \in F_i^w} a_{im} \right) e_i = - \sum_{j \in C_i} \left( a_{ij} + \sum_{k \in F_i^s} \left( \frac{a_{ik} a_{kj}}{\sum_{l \in C_i} a_{kl}} \right) \right) e_j \quad (2.32)$$

From Equation (2.32), an interpolation formula for  $i \in F$ ,  $e_i = \sum_{j \in C_i} w_{ij} e_j$ , can be defined with

$$w_{ij} = \frac{a_{ij} + \sum_{k \in F_i^s} \left( \frac{a_{ik} a_{kj}}{\sum_{l \in C_i} a_{kl}} \right)}{a_{ii} + \sum_{m \in F_i^w} a_{im}} \quad (2.33)$$

The value of  $F_i^s$  is approximated by a sum of the value of  $C_i$ , and the value of  $F_i^w$  is simply added to  $a_{ii}$ . However, the selection of  $F_i^s$  and  $F_i^w$  is fully determined by their coefficients in Equation (2.30), and this would cause the bad choice of  $F_i^s$  and  $F_i^w$ . Some nodes of  $F_i^w$  with large errors should be labeled in the set of  $F_i^s$ . This behavior will decrease the convergence rate of standard AMG. One of the main object of our AbAMG is to overcome this defect.

After introducing the concepts of color scheme and weights calculation, the flowchart of traditional AMG is shown in Fig. 2.9. At first, a cycle construction is performed to construct the multigrid V-cycle. In the fine grid, a color scheme is performed to determine the coarse grid and the weights of intergrid transfer operator can be calculated by Equation (2.33). The coarse grid operator  $A_c$  can be derived from the Galerkin operator  $A_c = RA_fP$ . We apply these steps repeatedly until the coarsest grid operator is coarse enough. After the step of traditional AMG cycle construction, we can derive the multigrid V-cycle and the answer of  $x$  can be solved by the multilevel solver mentioned in section 2.3.

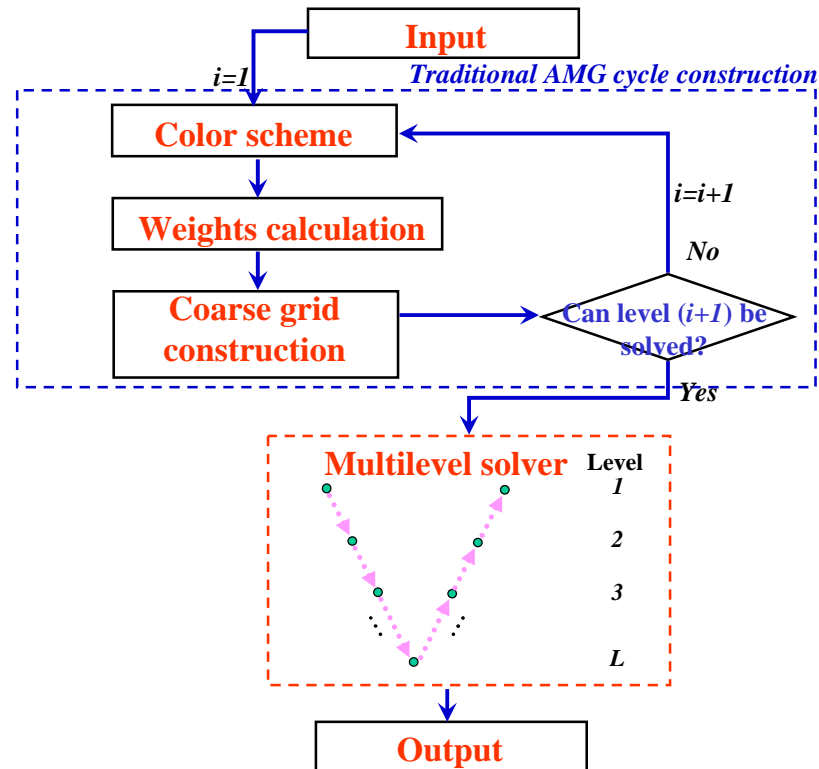


Fig. 2.9: The Flowchart of Traditional AMG

AMG construct the coarse grid and transfer operator from the property of system matrix only. It can be applied to various types of problems without additional geometry information. However, the construction of mapping operator  $R$  and  $P$  strongly depends on the choice of coarse grid and contains only local information of the system. Another algebraic multigrid method using the aggregation concept is introduced in Section 2.3.2.

### 2.3.2 Aggregation Algebraic Multigrid Method

Aggregation methods originated in economics [28], where similar products are considered together instead of individually. This procedure allows significant reduction in the problem size, and maintaining accurate representation of the overall behaviors. In multi-grid terminology, the coarse grid is selected as a collection of subsets of the fine grid. An algebraic partition is performed to the original fine grid and the original system matrix is



partitioned into several aggregated sub-matrices.

The idea of the mapping operator construction of aggregation AMG [25][26][27] is based on the concept that the smooth error components are in the directions of the system's eigenvectors associated with small eigenvalues [27]. An algebraic partition is performed in the aggregation AMG according to the connections of the nodes in the graph of the system matrix  $A$  and the nodes with strong influence between them are clustered together in an aggregation. A node-by-node aggregation algorithm is discussed in Section 3.5.1. After the aggregation procedure, an eigen-decomposition procedure is performed in each aggregated sub-matrix and the eigenvector related to the small eigenvalue is used to compose the mapping operator  $P$ . With accurate calculation of the system's smooth error components, the aggregation AMG can achieve better convergence rate than traditional AMG. However, the weak connected coefficients of small values between aggregations are simply neglected or added to the diagonal elements in the aggregated sub-matrices, and decreasing the convergence rate of aggregation AMG. An innovative matrix compensation algorithm with a global error estimation procedure is proposed in our AbAMG method to improve this defect in Chapter 3.

# Chapter 3

## Aggregation-Based Algebraic Multigrid

In this chapter, we will introduce the algorithm flow of our proposed method. We first state the problem formulation of the research in Section 3.1. In Section 3.2, we show the algorithm flowchart of AbAMG and compare the main differences between it and traditional AMG. The overview of our approach and previous works and the derivation of analysis system equations are discussed in Section 3.3 and Section 3.4. Finally, the cycle construction of our method and multilevel solver are discussed in Section 3.5 and 3.6.

### 3.1 Problem Formulation

The problem formulation of AbAMG for on-chip power network analysis can be formulated as follows.

- **Input:** A RLKC network netlist and the independent voltage sources are given for on-chip power network. The external current sources are modeled as time varying piecewise linear current sources.
- **Output:** The voltage waveform of each node and the current waveform of each wire segment are shown with respect to time.
- **Goal:** To develop an efficient analysis method for on-chip power network.

## 3.2 Flowchart of Our Proposed Method

This section we show the flowchart of our proposed method and point out the main differences between our method and traditional AMG. The flowchart of our proposed method is shown in Fig. 3.1

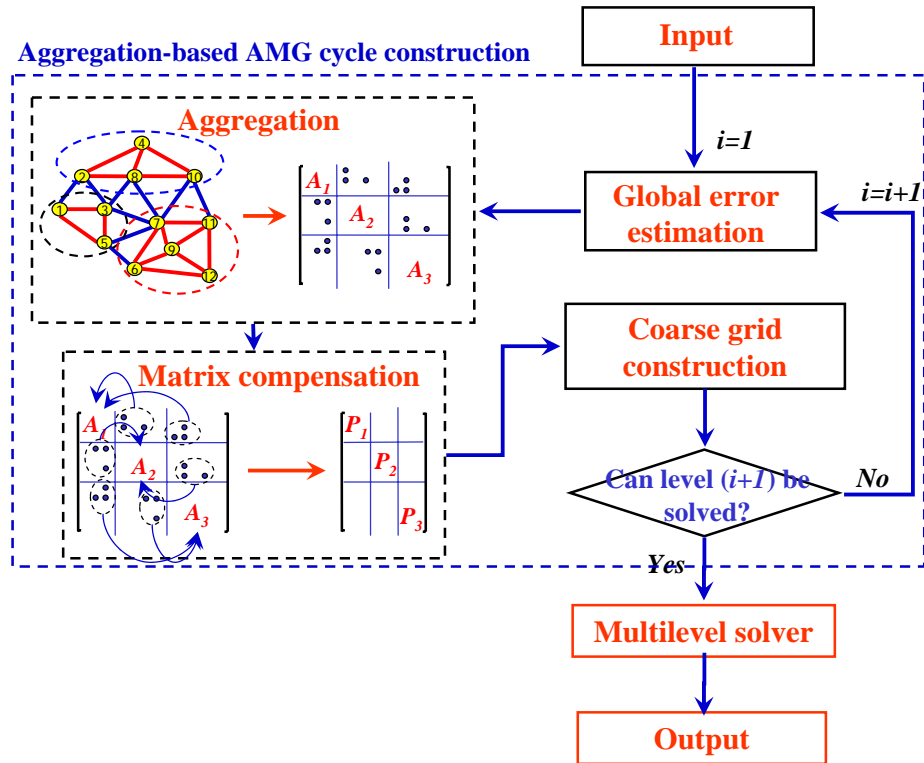


Fig. 3.1: Flowchart of Our Proposed Method

At first, an aggregation-based AMG cycle construction is performed to construct the multigrid V-cycle. In the fine grid, a global error estimation gives an estimation of global null-space error. This information can be used to compensate the following aggregated sub-matrices. Then, an aggregation algorithm is applied to partition the original system matrix into many sub-matrices to localize the problem. After that, a matrix compensation algorithm compensates the sub-matrices and the intergrid transfer operator can be derived from the sub-matrices locally. The coarse grid operator  $A_c$  can be derived from

the Galerkin operator  $A_c = RA_fP$ . These steps are applied recursively to construct the multigrid V-cycle and the answer of  $x$  can be solved by the multilevel solver.

The main difference between our method and traditional AMG is based on the cycle construction step. Traditional AMG selects the coarse grid by the color-scheme algorithm and begins the construction of intergrid transfer operator with the concept of algebraic smoothness  $Ae \approx 0$ . Our proposed method begins the cycle construction step with the aggregation concept. Since most iterative methods quickly eliminate the components of the error in the directions of the eigenvectors of the system matrix associated with the large eigenvalues [20]. If we can get the eigenvectors of the system matrix  $A$ , the interpolation operator can be composed by the eigenvectors associated with small eigenvalues and a multilevel structure that can efficiently eliminate all error components can be constructed.

However, the real eigenvectors of the system matrix can't be calculated directly. We apply an aggregation algorithm to the original system matrix to perform an algebraic partition and try to get the approximated eigenvectors of the system matrix from the aggregated sub-matrices locally. The error estimation and matrix compensation steps are used to improve the quality of the aggregated sub-matrices. The eigenvector associated with the small eigenvalue in each aggregated sub-matrix is used to compose the intergrid transfer operator  $R$  and  $P$ . By using aggregation-based AMG, we can project the original system matrix into another domain to expose the low frequency errors in the original domain, and construct a smaller interpolation operator compared to traditional AMG with the aggregation property.

### 3.3 Overview of Our Approach and Previous Works

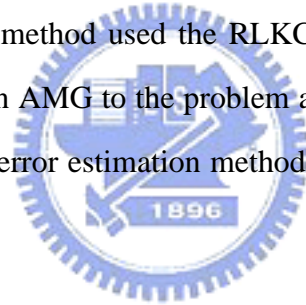
In this section, we compare the existing published multigrid methods on the power network analysis in Fig. 3.2. The multigrid method was first applied to the power network problem by Nassif in [12][13]. It applied the geometry multigrid method and extended the mapping operator constructing scheme to the irregular mesh structure. However, this

method modeled the power network as the RC segments, and had problem with mutual inductance coupling effects.

The concept of algebraic multigrid method was first applied to power network problem by Su. It used a geometry-like mapping operator constructing scheme to solve the problem in short time. However, the model of this method didn't include the mutual inductance and the rough mapping operator constructing scheme lead to large errors.

Another algebraic multigrid solver is developed in [14]. It used the traditional algebraic multigrid method with RLKC model and an adaptive coarsening scheme is applied to improve the convergence rate. However, the coarsening scheme must construct the cycle at every time steps and will increase the CPU time.

Our proposed method developed an aggregation-based AMG method for power network problem. Our proposed method used the RLKC model as [14]. We discuss the practicability of the aggregation AMG to the problem and an additional matrix compensation algorithm with a global error estimation method is applied to further improve the convergence rate.



### 3.4 Derivation of System Equations

By using MNA, the system equation of an irregular power network can be formulated as following

$$\hat{G}x(t) + \hat{C} \frac{d}{dt}x(t) = Bu(t) \tag{3.1}$$

where

$$\hat{G} = \begin{bmatrix} G & -A_l^T & -A_{V_E}^T \\ A_l^T & 0 & 0 \\ A_{V_E}^T & 0 & 0 \end{bmatrix}$$

$$\hat{C} = \begin{bmatrix} C & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$x(t) = \begin{bmatrix} v(t) \\ i_l(t) \\ i_{V_E}(t) \end{bmatrix}$$

	<b>Nassif [DAC00] [ICCAD01]</b>	<b>Su [DAC03]</b>	<b>Zhu [DAC03]</b>	<b>Our proposed method</b>
<b>Method</b>	<b>Geometric multigrid</b>	<b>Algebraic multigrid</b>	<b>Algebraic multigrid</b>	<b>AbAMG (Aggregation- based AMG)</b>
<b>Model</b>	<b>RC</b>	<b>RLC</b>	<b>RLKC</b>	<b>RLKC</b>
<b>Features</b>	<b>Geometry coarsening scheme for irregular power network</b>	<b>Geometry- like mapping operator construction</b>	<b>Adaptive coarsening scheme</b>	<b>Construct the mapping operators from system's eigenvectors</b>

Fig. 3.2: Comparison between Our Method and Previous Methods

The definitions of  $\hat{G}$ ,  $\hat{C}$  and  $x(t)$  are the same as Section 2.1. Since the independent voltage sources are known, it is not necessary to solve the nodes of independent voltage sources and the currents flowing through them. With this idea, the analysis dimension can be reduced and the system equation can be rewritten as following

$$\tilde{G}\tilde{x}(t) + \tilde{C}\frac{d}{dt}\tilde{x}(t) = \tilde{B}\tilde{u}(t) + \tilde{G}_E v_E(t), \quad (3.2)$$

where

$$\tilde{G} = \begin{bmatrix} G_n & -A_{l_n}^T \\ A_{l_n}^T & 0 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} C_n & 0 \\ 0 & L \end{bmatrix},$$

$$\tilde{x}(t) = \begin{bmatrix} v_n(t) \\ i_l(t) \end{bmatrix}, \quad \tilde{G}_E = \begin{bmatrix} G_E \\ L_E \end{bmatrix}.$$

Here,  $v_E(t)$ ,  $v_n(t)$ , and  $i_l(t)$  correspond to the vectors of the independent voltage sources, the unknown nodal voltages, and the branch currents flowing through inductors, respectively,  $G_n$ ,  $C_n$ , and  $L$  represent the stamping matrices of the resistors not connecting to  $v_E(t)$ , the capacitors, and the inductors, respectively,  $A_{l_n}$  corresponds to the

coefficient matrix related to those inductors not connecting to  $v_E(t)$ ,  $\tilde{u}(t)$  is the vector of independent current sources, and  $\tilde{B}$ ,  $G_E$ , and  $A_{l_E}$  are the coefficient matrices related to  $\tilde{u}(t)$ , the stamping of resistors between  $v_n(t)$  and  $v_E(t)$ , and the connecting of  $L$  and  $v_E(t)$ , respectively.

Applying trapezoidal approximation with time step  $h$ , Equation (3.2) can be reformulated as following

$$\begin{aligned} \begin{bmatrix} \frac{2C_n}{h} + G_n & -A_{l_n}^T \\ A_{l_n} & \frac{2L}{h} \end{bmatrix} \begin{bmatrix} v_n(t+h) \\ i_l(t+h) \end{bmatrix} &= 2 \begin{bmatrix} G_E \\ A_{l_E} \end{bmatrix} v_E(t) \\ + \begin{bmatrix} \frac{2C_n}{h} - G & A_{l_n}^T \\ -A_{l_n} & \frac{2L}{h} \end{bmatrix} \begin{bmatrix} v_n(t) \\ i_l(t) \end{bmatrix} + \tilde{B} \begin{bmatrix} \tilde{u}(t+h) + \tilde{u}(t) \\ 0 \end{bmatrix}. \end{aligned} \quad (3.3)$$

After the time domain discretion, we can observe that the transient analysis system matrix in Equation (3.3) is not symmetric and positive definite due to the introduction of current variables.

Since the Multigrid method requires the matrix to be symmetric positive definite, some extra processing is needed to reformulate the system matrix [21]. Similar to the method used in [9], we split the variable vector into nodal voltage vector and branch current vector. By using block matrix operations, we can decompose Equation (3.3) into two iteration formulas for nodal voltages and branch currents. The system equations are reformulated as following

$$\begin{aligned} \left( \frac{2C_n}{h} + G_n + \frac{h}{2} A_{l_n}^T L^{-1} A_{l_n} \right) v_n(t+h) &= \\ \left( \frac{2C_n}{h} - G_n - \frac{h}{2} A_{l_n}^T L^{-1} A_{l_n} \right) v_n(t) + 2A_{l_n}^T i_l(t) & \\ + \tilde{B}(\tilde{u}(t+h) + \tilde{u}(t)) + hA_{l_n}^T L^{-1} A_{l_E} v_E(t) + 2G_E v_E(t) & \end{aligned} \quad (3.4)$$

$$i_l(t+h) = i_l(t) - \frac{h}{2} L^{-1} A_l (v_n(t+h) + v_n(t)) + hL^{-1} A_{l_E} v_E(t) \quad (3.5)$$

Since the matrices  $G_n$ ,  $C_n$ , and  $L^{-1}$  (or  $K$ ) are SPD, we can prove that the system matrix of Equation (3.4) is still SPD. The  $L^{-1}$  ( $K$ ) [22] is sparser than the original  $L$  matrix, and the above symmetric property can save 50% of the memory usage. Equation (3.4) is equivalent to solve an  $Ax = b$  problem, where  $A$  is equal to  $(\frac{2C_n}{h} + G_n + \frac{h}{2} A_{l_n}^T L^{-1} A_{l_n})$ ,

and this problem can be solved by the two-level solution method, and the solutions of Equation (3.4) and (3.5) are solved iteratively.

## 3.5 Aggregation-Based AMG Cycle Construction

In this section, a global mapping operator construction of our AbAMG is presented. At first, a node-by-node aggregation algorithm is shown in Section 3.5.1. The practicability of the aggregation AMG method to the power network analysis problem is discussed in Section 3.5.2. The global error estimation procedure and matrix compensation algorithm are stated in Section 3.5.3 and Section 3.5.4. Finally, the mapping operator construction procedure is stated in Section 3.5.5.

### 3.5.1 Aggregation Algorithm

The purpose of the aggregation method in AMG is to reformulate the original system matrix such that the smooth error components of the system can be calculated from the modified system easily. Different from the difficulty of performing a geometry partition with the complex mutual inductance coupling effects on the circuit topology, the aggregation method provides an easy approach to partition the problem in the algebraic manner, and simplifying the problem.

A node-by-node aggregation algorithm is discussed in this section. The definition of strong connection between nodes  $i$  and  $j$  provides a good measurement when determining aggregations. Nodes  $i$  and  $j$  are defined to have a strong connection if there is any strongly influence relation between them. If there is no strongly influence relation between node  $i$  and node  $j$ , we say that they have a weak connection. Node with maximum number of strong connections acts a good candidate to be the starting node in the aggregation algorithm and nodes with strong connections between them must be labeled in the same aggregation since the value of  $a_{ij}$  is large with respect to  $a_{ii}$ . The nodes with weak connections between them should be labeled in different aggregations and each node can



only be included in an aggregation. The rules of aggregation can be concluded in the following

- **Aggregation Rules:**

- Select the node with maximum number of strong connections in the graph as the starting node when determining an aggregation
- Every node must be included in an aggregation
- Each node can not be labeled to different aggregation
- Let the nodes which have strong connections between them be labeled in same aggregation
- Let the nodes which have weak connections between them be labeled in different aggregations

Considering a system equation,  $Ax = b$ , the aggregation algorithm is shown in Table 3.1 and 3.2.

An example of aggregation algorithm is shown in the following. Considering a system equation  $Ax = b$ , the original system matrix  $A$  and the graph of it are shown in Fig. 3.3.

Fig. 3.4 shows an example of aggregation. The description of each step is shown below:

- **Example of Aggregation:**

- **Step a:** A matrix graph of  $A$  is given with node number 1 to 12.
- **Step b:** The weight of each node  $i$  is determined by counting the number of strong connections.
- **Step c:** Node 9 is defined as the starting aggregated node with maximum weight of 4 and labeled to aggregation a.

---

---

**Algorithm of Aggregation**

---

---

**Input:** *The Graph of System Matrix  $A$  of Nodes  $1, 2, \dots, n$  and the Related Weights  $w_1, w_2, \dots, w_n$  of These Nodes*

---

---

**Output:** *Aggregations  $1, 2, \dots, m$*

---

---

```
1  Begin
2    NodeCounter=0, AggCounter=0
3    While NodeCounter!= $n$ 
4      MaxWeight=0, StartNode=1
5      For each node  $i$ 
6        If node  $i$  is not in an aggregation
7          If  $w_i > \text{MaxWeight}$ 
8            Then MaxWeight= $w_i$ , StartNode= $i$ 
9        EndFor
10     AggCounter++
11      $j = \text{AggCounter}$ , StartNode is labeled in aggregation  $j$ 
12     NodeCounter++
13     AggreConstruct(StartNode)
14  End.
```

---

Table 3.1: Algorithm of Aggregation

---

---

**Algorithm of AggreConstruct**

---

---

**Input:** *The Node  $i$  and It's Strongly Connected Nodes  $n_1, n_2, \dots, n_s$*

---

---

```
1  Begin
2    For Each strongly connected node  $k$  of node  $i$ 
3      If node  $k$  is not in an aggregation
4        Node  $k$  is labeled in aggregation  $j$ 
5        NodeCounter++
6        AggreConstruct( $k$ )
7    EndFor
8  End.
```

---

Table 3.2: Algorithm of AggreConstruct



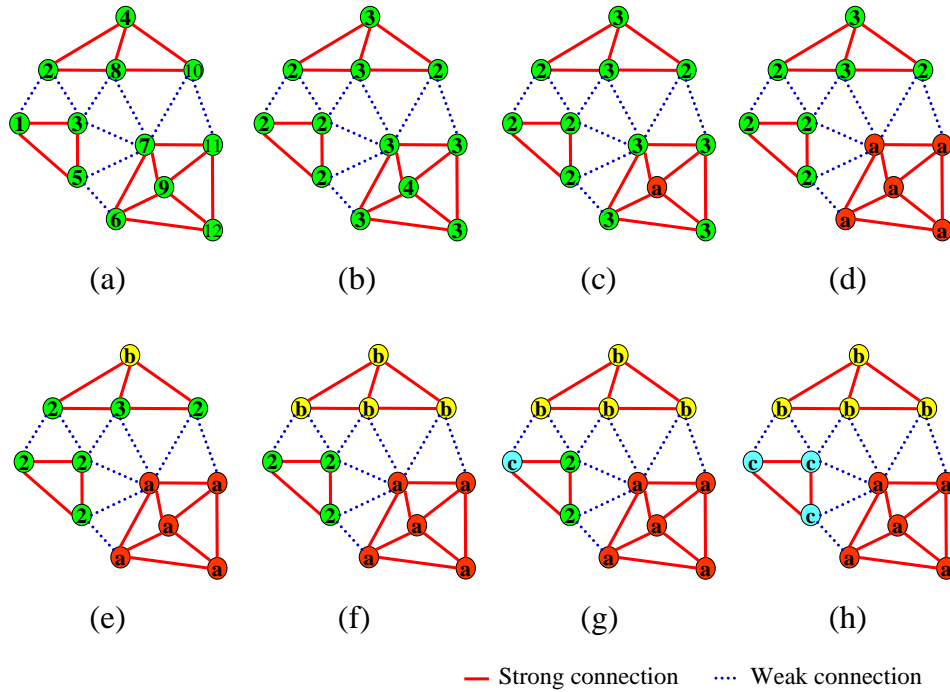


Fig. 3.4: Example of Aggregation

by the eigen-decomposition analysis of these sub-matrices. The weakly connected coefficients between aggregations are simply added to the diagonal elements of the aggregations or neglected since their value is small compared to the diagonal elements. However, some nodes with weakly connected coefficient may have large errors in the global view, and decreasing the convergence rate. A global error estimation procedure is presented in Section 3.5.3 to give an estimation of these troublesome error components and the related matrix compensation algorithm is shown in Section 3.5.4 to improve this defeat.

### 3.5.2 Practicability of Aggregation AMG Method to Power Network Analysis

In this subsection, we discuss the practicability of the aggregation AMG method to power network analysis problem. The aggregation AMG method is applied to the  $Ax = b$  problem in equation (3.4), where  $A$  is equal to  $(\frac{2C_n}{h} + G_n + \frac{h}{2}A_{l_n}^T L^{-1} A_{l_n})$ . From the predictive

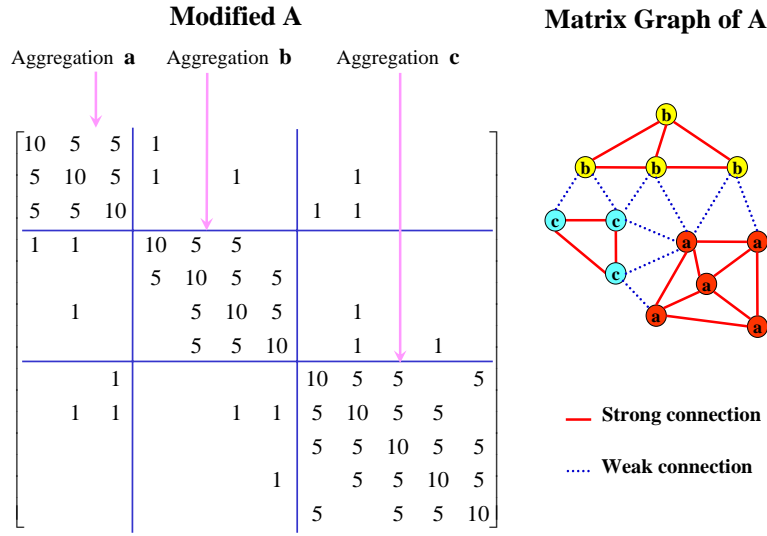


Fig. 3.5: System Matrix After Aggregation

technology model (PTM) developed by the Berkeley university (<http://www.eas.asu.edu/ptm/>), the dimensions of  $R, L, C$  in the  $0.13\mu m$  technology are shown as following, where  $R = 0.046ohm/\mu m$ ,  $L = 1.69pH/\mu m$ ,  $C = 0.13011fF/\mu m$ , length of each wire segments is of  $100\mu m$ . The contributions of the values of  $\frac{2C_n}{h}$ ,  $G_n$ , and  $\frac{h}{2}A_{l_n}^T L^{-1} A_{l_n}$  vary from  $5E-3$  to  $8E-3$ ,  $2E-1$  to  $4E-1$  and  $2E-2$  to  $5E-2$ . The contribution of the value of the  $G_n$  term is often 10 times larger than other terms of  $A$ . From the aggregation algorithm discussed in Section 3.5.1, we can know that the determination of the aggregation of  $A$  is dominated by the effects of the resistances of  $G_n$ . Since the on-chip power delivery network is of mesh structure, most aggregations are of the size of 3 as shown in Fig. 3.6 and the maximum size of the aggregation is less than 4 with via connected to the structure of Fig. 3.6.

### 3.5.3 Global Error Estimation

In this subsection, we introduce a global error estimation step. In this thesis, we can know that the efficiency of multigrid method depends on the complementarity between relaxation and coarse-grid correction. The error components not efficiently reduced by relaxation must be represented in the range of interpolation. However, these error com-

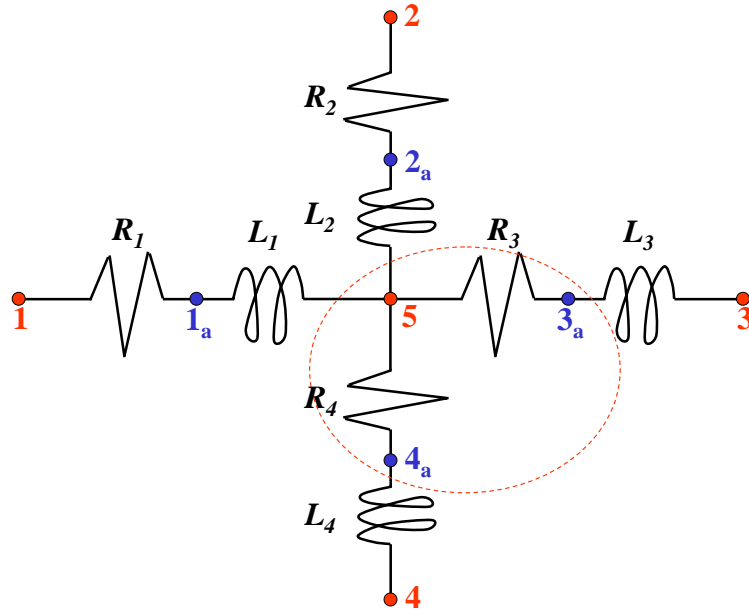


Fig. 3.6: Structure of Aggregation of the Size of 3

ponents are the quantity we do not know. A simple method to gain information about the errors that relaxation does not efficiently reduce is applying the relaxation scheme to a problem with known solution.

The homogeneous equation,  $Ax = 0$ , serves us well for this purpose. The real error of  $Ax = 0$  can be known since the exact solution of this equation is zero. By applying relaxation several times to this equation with a random initial guess, we can get a error vector,  $e_G$ , which can represent the error component that the relaxation can not eliminate well. This candidate error vector can provide information about troublesome error and be used to compensate the aggregated matrices in our method. The compensation algorithm will be introduced in Section 3.5.4.

### 3.5.4 Matrix Compensation Algorithm

In this subsection, we introduce a matrix compensation algorithm in this thesis. Considering a linear algebraic system equation  $Ae_G = 0$  of dimension  $N$ . We begin the discussion of this algorithm with the concept of global error vector  $e_G$  mentioned in Section 3.5.3.

<b>Algorithm of Matrix Compensation</b>	
<b>Input:</b> Original System Matrix $A$ and Aggregations $1, 2, \dots, n$	
<b>Output:</b> Aggregated Sub-matrices $A_1, A_2, \dots, A_n$	
1	<b>Begin</b>
2	<b>For</b> each aggregation $m$
3	<b>For</b> each node $i$ in aggregation $m$ , sweep the $i$ -th row of $A$
4	<b>If</b> node $j$ is within aggregation $m$
5	<b>Then</b> $A_{m_{ij}} = A_{ij}$
6	<b>Else</b>
7	<b>If</b> Node $j$ is a strong node, sweep the $j$ -th row of $A$
8	Total = 0
9	<b>For</b> each column $k$ in row $j$
10	<b>If</b> node $k$ is within aggregation $m$
11	<b>Then</b> total+ = $A_{jk}$
12	<b>EndFor</b>
13	<b>For</b> each column $k$ in row $j$
14	<b>If</b> node $k$ is within aggregation $m$
15	<b>Then</b> $A_{m_{ik}} + = A_{ij} \times A_{jk} / Total$
16	<b>EndFor</b>
17	<b>Else</b>
18	<b>EndFor</b>
19	<b>EndFor</b>
20	<b>End.</b>



Table 3.3: Algorithm of Matrix Compensation

Node  $i$  is defined as strong node if  $e_{G_i} > \lambda \max(e_{G_j})_{j=1}^n$  and defined as weak node if  $e_{G_i} < \lambda \max(e_{G_j})_{j=1}^n$ . Here,  $\lambda$  is a coefficient from 0 to 1 and is chosen to be 0.25 in our algorithm.

With this concept, the matrix compensation algorithm is shown in Table 3.3. The weak connected coefficients related to the strong nodes are approximated with the aggregated nodes and the effects related to the weak nodes are simply neglected in our algorithm. By using this algorithm, we can construct a better system for analysis and make each aggregated sub-matrix independent from other sub-matrix. The compensation procedure is exactly matched the weight calculation step in traditional AMG. The modified sub-matrices can be used for coarse grid construction and will be introduced in next section.

### 3.5.5 Aggregation-Based AMG Coarse Grid Construction

In this subsection, we introduce the coarse grid construction of our Aggregation-based AMG method. After the step of matrix compensation, we can get the independent sub-matrices from the original system. An example of coarse grid construction is shown in Fig. 3.7, an eigenvalue decomposition procedure is performed in each aggregated sub-matrix. The eigenvector related to the smallest eigenvalue is used to compose the inter-grid transfer operator  $P$  and the coarse grid operator can be constructed by the Galerkin operator  $A_c = RAP$ .

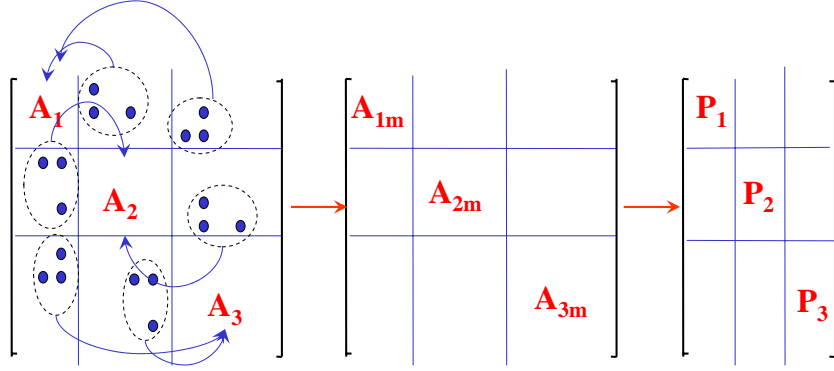


Fig. 3.7: Coarse Grid Construction

## 3.6 Aggregation-Based Multilevel Solver

This section we introduce the overall solver of our proposed method. The system equation derived in Section 3.4 is shown below

$$\begin{aligned}
 \left( \frac{2C_n}{h} + G_n + \frac{h}{2} A_{l_n}^T L^{-1} A_{l_n} \right) v_n(t+h) &= \left( \frac{2C_n}{h} - G_n - \frac{h}{2} A_{l_n}^T L^{-1} A_{l_n} \right) v_n(t) \\
 &+ \tilde{B}(\tilde{u}(t+h) + \tilde{u}(t)) + 2A_{l_n}^T i_l(t) \\
 &+ hA_{l_n}^T L^{-1} L_E V_E + 2G_E V_E \quad (3.6)
 \end{aligned}$$



$$\begin{aligned}
i_l(t+h) &= i_l(t) - \frac{h}{2}L^{-1}A_l(v_n(t+h) + v_n(t)) \\
&\quad + hL^{-1}L_E V_E
\end{aligned} \tag{3.7}$$

At first, we apply the aggregation-based AMG cycle construction procedure to the system matrix of  $\left(\frac{2C_n}{h} + G_n + \frac{h}{2}A_{l_n}^T L^{-1}A_{l_n}\right)$  in Equation 3.5. Then, we apply AbAMG solver to calculate the value of  $v_n(t+h)$ . With the value of  $v_n(t+h)$ , we can get the value of  $i_l(t+h)$  from Equation 3.6. Recursive calculating Equation 3.5 and 3.6, we can solve the power network problem and get the voltage waveform of the analysis voltage nodes. The mapping operator construction of AbAMG is determined from the global information of system and only needs to be performed once for all time step calculation. The experimental results are shown in Chapter 4.



# Chapter 4

## Experimental Results

This chapter demonstrates the speed and accuracy of our proposed AbAMG solver and compares our results with other methods. The power delivery networks are randomly generated as mesh networks which consist of lumped RLKC segments and many current sources. This work is implemented in C++ language and test on a Pentium IV 3.4-GHz machine with 3 GB memory.

First, an efficient and accurate time domain solver InductWise [24] is used to demonstrate the accuracy of our method. The accuracy of RLKC circuits is shown in Table 4.1. In Table 4.1, Min V means the minimum voltage drop with respect to each test circuit. The minimum voltage drop of each test circuit is above 0.832V. The maximum error is within 0.973% for each RLC test circuit and the average error is less than 0.067% for AbAMG with compensation. These results demonstrate the excellent accuracy of our algorithm.

To show the efficiency of our AbAMG solver, the analysis of DC and 50 transient time steps are executed and the results are compared with three state-of-the-art methods, IEKS [11], InductWise [24] and standard AMG. The comparison results are shown in

Circuit Size	Min V	Standard AMG		AbAMG without compensation		AbAMG with compensation	
		Max Error (%)	Avg Error (%)	Max Error (%)	Avg Error (%)	Max Error (%)	Avg Error (%)
49.6K	0.85	1.173	0.077	1.165	0.077	0.973	0.067
199.2K	0.832	1.107	0.066	1.059	0.065	0.885	0.058
448.8K	0.835	1.17	0.065	1.072	0.064	0.955	0.056
798.4K	0.842	1.13	0.067	1.081	0.067	0.967	0.059

Table 4.1: Error percentage of RLKC circuits

Circuit Size	InductWise [24]		IEKS [11]		Standard AMG		Ours Result		
	RT(s)	Mem(MB)	RT(s)	Mem(MB)	RT(s)	Mem(MB)	RT*(s)	RT**(s)	Mem(MB)
49.6K	78.34	111	6.25	68	3.953	46	3.593	2.972	40
199.2K	391.7	424	29.76	308	15.875	182	14.235	12.719	156
448.8K	1576	994	82.56	747	38.187	407	32.594	28.312	351
798.4K	2903	1547	131.31	1230	68.219	721	59.328	51.812	624
1.248M	×	>3000	×	×	105.59	1130	93.75	83.156	974
1.7976M	×	>3000	×	×	152.36	1627	137.312	119.36	1401
2.4472M	×	>3000	×	×	×	×	196.531	167.6	1907

Table 4.2: Runtime of RLKC circuits. “×” denotes this methodology failed.

Circuit Size	Speed up			
	$S_{In}$	$S_{IEKS}$	$S_{AMG}$	$S_{No}$
49.6K	26.36	2.1	1.33	1.21
199.2K	30.8	2.34	1.25	1.12
448.8K	55.7	2.92	1.35	1.15
798.4K	56	2.53	1.32	1.15

Table 4.3: Speed up of AbAMG compared to other methods

Table 4.2 for different RLKC circuits. In Table 4.2, RT is the CPU run time and Mem means the memory usage. RT\* means the run time of AbAMG without compensation and RT\*\* represents the run time of AbAMG with compensation. The speedup of our method for each test circuit case is shown in Table 4.3. In Table 4.3,  $S_{In}$ ,  $S_{IEKS}$ ,  $S_{AMG}$  and  $S_{No}$  are the speedup of AbAMG with compensation respect to InductWise, IEKS, standard AMG and AbAMG without compensation. The significant speed improvement, 26 times faster than the InductWise [24], 2 times faster than IEKS [11] and 1.27 times faster than standard AMG, and less memory usage, two fifth of the memory usage in [24] and half of the memory usage in [11], are observed.

A plot of CPU time versus circuit size for each method is shown in Fig. 4.1, we can observe that the CPU time of AMG-based methods are proportional to circuit size and the AbAMG method without compensation has best performance. The memory usage versus circuit size of each method is plotted in Fig. 4.2. The memory usage of AMG-based methods are proportional to the circuit size and AbAMG method spends least memory.

The proposed AbAMG solver can solve the DC and transient nodal voltages of a circuit with the circuit size being 2.4472M in 167.6 CPU seconds, and this indicates that the proposed simulator is very efficient in solving power delivery networks and capable of handling more than two million circuit size.

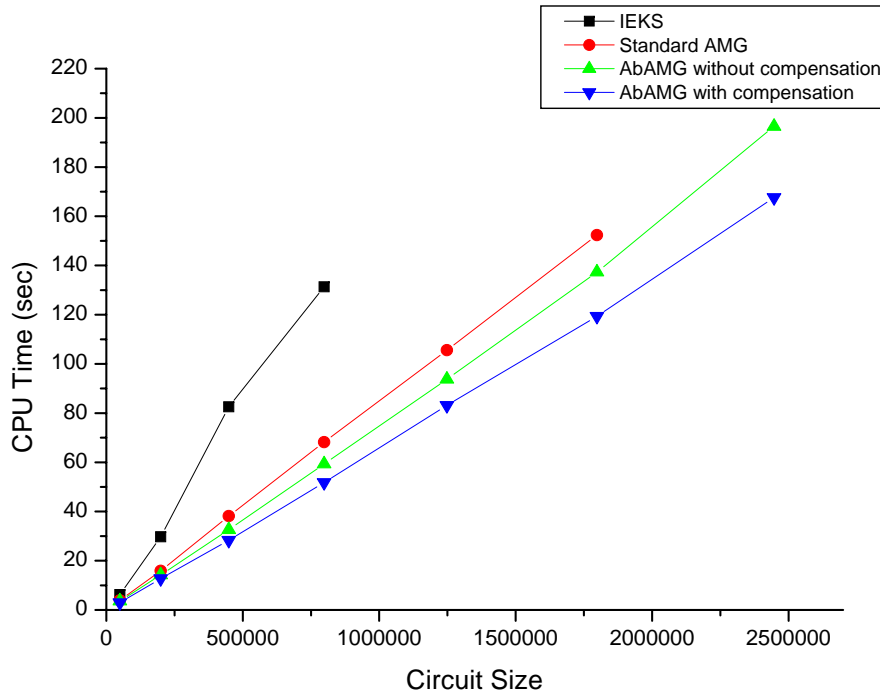


Fig. 4.1: Run Time versus Circuit Size

Circuit Size	Standard AMG			AbAMG		
	Fine Grid NZ	Coarse Grid NZ	Cycle	Coarse NZ	Cycle*	Cycle**
49.6K	119.5K	92K	118	33.8K	127	101
199.2K	480K	370K	113	135.6K	131	100
448.8K	1082K	834K	111	305.4K	131	101
798.4K	1925K	1484K	114	543K	135	103
1.248M	3009K	2320K	109	849K	131	102
1.7976M	4335K	3342K	108	1223K	132	100

Table 4.4: Comparison between AbAMG and standard AMG

A comparison between standard AMG and AbAMG is shown in Table 4.4, Fine Grid NZ , Coarse Grid Nz, Cycle\* and Cycle\*\* are the non-zero terms of original fine grid, non-zero terms of coarse grid, total number of multilevel cycle of AbAMG without compensation and total number of multilevel cycle of AbAMG with compensation. The plot of non-zero terms versus circuit size and total multilevel cycles versus circuit size are shown in Fig. 4.3 and Fig. 4.4. The coarse grid Nz of AbAMG is only one third of standard AMG and the number of cycle of AbAMG with compensation is smaller than standard AMG.

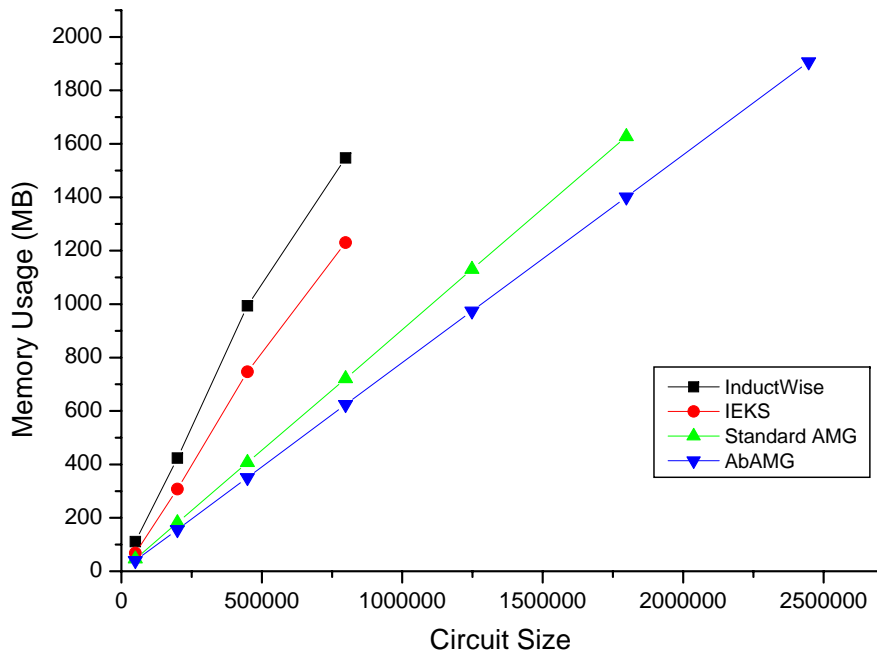


Fig. 4.2: Cpu Time versus Circuit Size

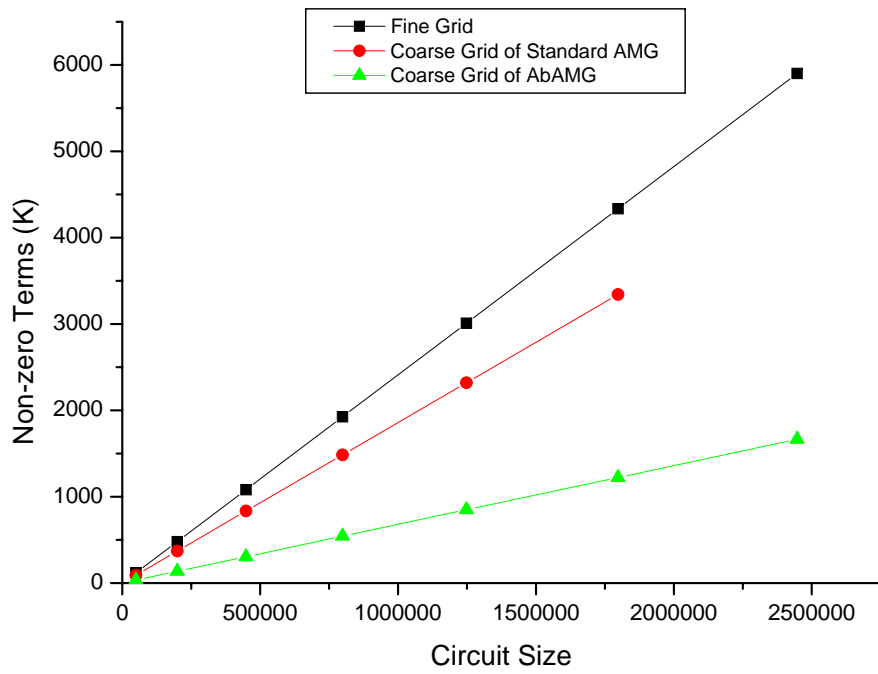


Fig. 4.3: Non Zero Term versus Circuit Size

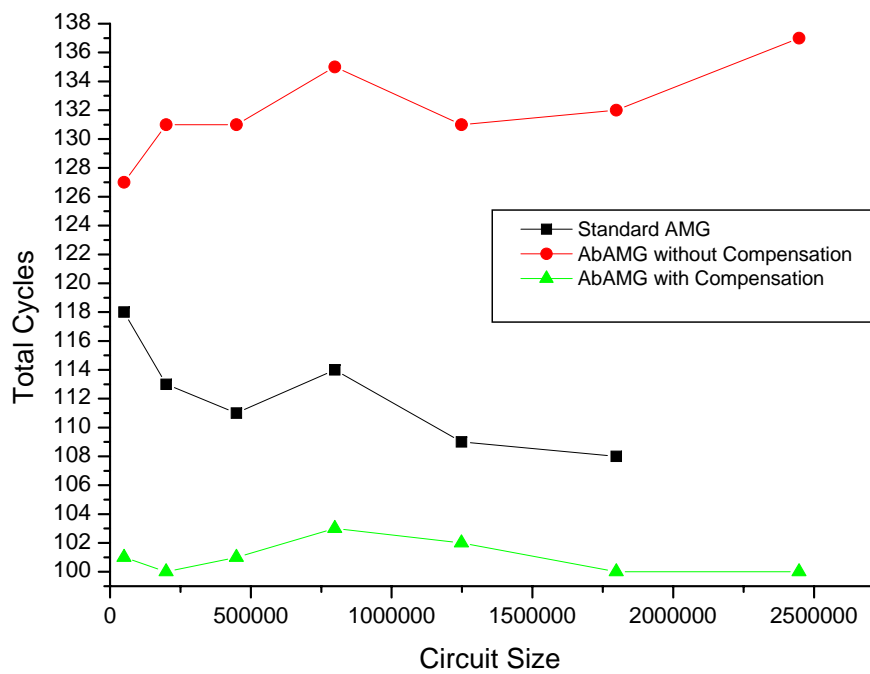


Fig. 4.4: Total Cycles versus Circuit Size

# Chapter 5

## Conclusions

In this thesis, we present an aggregation-based algebraic multigrid solver for the power/ground distribution network analysis. Different from the traditional algebraic multigrid solver, our AbAMG solver constructs the inter-grid mapping operators from the global information of the original system matrix. By performing an aggregation algorithm, we can perform an algebraic partition to the original system. With the matrix compensation algorithm and the global error estimation process, we can get the modified sub-matrices from the original system and calculating the approximated eigenvector to constructed the global-considering inter-grid mapping operators.

Experimental results show that the proposed methodology can handle circuit size more than two million in 167.6 CPU seconds. The maximum error of each RLKC test circuit is less than 1%. The significant speed improvement and the less memory usage show our AbAMG methodology is very suitable for analyzing the power delivery network. The global construction of mapping operator improves the performance of AbAMG, constructs smaller coarse grid and converges with smaller cycles than standard AMG.



# Appendix A

## Property of Error Propagation

The property of error propagation of the basic iterative methods and an related prove of it is shown in the following

- **Error in the direction of an eigenvector associated with a large eigenvalue is rapidly reduced by relaxation and the error in the direction of an eigenvector associated with a small eigenvalue is reduced by a factor that may approach 1 as the eigenvalue approaches 0**

It's not too difficult to show that standard relaxation methods, such as Richardson, Jacobi, or Gauss-Seidel, satisfy the property. To see the related prove for Richardson iteration and assuming that the system matrix  $A$  is SPD so that  $\|A\|_2 = \lambda_m$ , where  $\lambda_m$  is the largest eigenvalue of  $A$ . The iterative formula of Richardson iteration with relaxation parameter  $\omega \in (0, 2)$  is shown in the following

$$x^{i+1} = x^i + \frac{\omega}{\|A\|_2} r^i \quad (\text{A.1})$$

Subtracting the exact solution  $x$  to the both sides of Equation A.1, we can get the error propagation equation as following

$$e^{i+1} = (I - \frac{\omega}{\lambda_m} A) e^i \quad (\text{A.2})$$

If  $e^i$  is an eigenvector of  $A$  and  $\lambda$  is its related eigenvalue, then

$$e^{i+1} = (1 - \omega \frac{\lambda}{\lambda_m}) e^i \quad (\text{A.3})$$

We can find that if  $\lambda \ll \lambda_m$ , then the factor is very close to 1 and  $e^{i+1}$  is almost the same size as  $e^i$ . In the contract, if  $\lambda \approx \lambda_m$ , then this factor is approximately  $1 - \omega$ , so substantial error reduction occurs so long as  $\omega$  is not too far from 1. Thus, with proper choice of relaxation parameter  $\omega$ , Richardson iteration satisfies the property and the related proves of other iterative methods can be derived in similar ways.



# Bibliography

- [1] Shen Lin and Norman Chang, "Challenges in Power-Ground Integrity" *IEEE/ACM International Conference on Computer Aided Design*, 2001.
- [2] Black, J. R., "Electromigration failure modes in aluminum metallization for semiconductor devices" *Proc. IEEE*, pp. 1587-1594, Sept 1969.
- [3] S. Bobba, T. Thorp, K. Aingaran, D. Liu, "IC power distribution challenges" *IEEE/ACM International Conference on Computer Aided Design*, 2001.
- [4] T. Burd and R. Brodersen, "Design Issue for Dynamic Voltage Scaling" *Int. Symp. on Low Power Electronics and Design*, pp. 9-14, 2001.
- [5] C. K. Cheng, J. Lillis, Shen Lin, and Norman Chang, "Interconnect Analysis and Synthesis" *John Wiley & Sons*, 1999.
- [6] M. K. Gowan, L. L. Biro and D. B. Jackson, "Power consideration in the design of the Alpha 21264 microprocessor", in *Proc. of Design Automation Conf.*, pp. 726-731, 1998.
- [7] Hui Zheng and Lawrence T. Pileggi, "Modeling and Analysis of Regular Symmetrically Structured Power/Ground Distribution Networks", *DAC*, June 10-14, 2002.
- [8] Abhijit Dharchoudhury, Rajendran Panda, David Blaauw, Ravi Vaidyanathan, "Design and Analysis of Power Distribution Networks in PowerPC Microprocessors", *DAC*, June 15-19, 1998.

- [9] T.-H. Chen and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods", in *Proc. of Design Automation Conference*, CA, Nov. 2001, pp. 259-62.
- [10] M. Zhao, R. V. Panda, S. S. Sapantnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks", in *IEEE Trans. on computer-aided design of integrated circuits and systems*, vol. 21, no. 2, IEEE, Feb. 2002. pp. 159-68
- [11] Yahong Cao, Yu-Min Lee, Tsung-Hao Chen and Charlie Chung-Ping Chen, "HiPRIME: Hierarchical and Passivity Reserved Interconnect Macromodeling Engine for RLKC Power Delivery" *Design Automation Conference*, 2002.
- [12] S. R. Nassif et al, "Fast Power Grid Simulation" *IEEE/ACM DAC*, 2000.
- [13] J. N. Kozhaya, S. R. Nassif, F. N. Najm, "Multigrid-like Technique for Power Grid Analysis" *IEEE/ACM International Conference on Computer Aided Design*, 2001.
- [14] Zhengyong Zhu, Bo Yao, Chung-Kuan Cheng, "Power network analysis using an adaptive algebraic multigrid approach" *IEEE/ACM DAC*, 2003.
- [15] L.T. Pillage, R.A. Rohrer and C. Visweswariah, "Electronic Circuit and System Simulation Methods" *McGRAW-HILL Book Co.*
- [16] Gene H. Golub and Charles F. Van Loan "Matrix Computations," third edition, *Johns Hopkins*.
- [17] Yousef Saad, "Iterative Methods for Sparse Linear Systems" *Society for Industrial and Applied Mathematics*.
- [18] W.L. Briggs, "A Multigrid Tutorial" *SIAM 2000*.
- [19] A. Brandt, S. F. McCormick, J. W. Ruge, "Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations" *tech. rep., Institute for Computational Studies, Colorado State University, 1982*.

- [20] A. Brandt, "Algebraic multigrid theory: the symmetric case" *Applied Mathematics and Computation*, 1986
- [21] "Design of high-performance microprocessor circuits" *IEEE Press*, 2001
- [22] A. Devgan, "How to Efficiently Capture On-Chip Inductance Effects: Introducing a New Circuit Element K" *IEEE/ACM International Conference on Computer Aided*, pp 150-155, Nov 2000
- [23] M. Beattie and L. Pileggi, "Efficient Inductance Extraction via Windowing" *DATE*, pp 430-436, March 2001
- [24] T.-H. Chen, C. Luk, H. Kim, and C. C.-P. Chen, "INDUCTWISE: Inductance-wise interconnect simulator and extractor", in *Proc. Int. Conf. Computer-Aided Design*, CA, pp. 215-220, Nov. 2002.
- [25] P. Vanek, "Acceleration of Convergence of a Two-Level Algorithm by Smoothing Transfer Operator" in *Application of Mathematics.*, 1992, Vol.37, pp. 265-274.
- [26] J. Fish and V. Belsky, "Generalized Aggregation Multilevel Solver" in *International Journal for Numerical Methods in Engineering.*, 40 (1997), pp. 4341-4361.
- [27] Thomas E. Giddings and Jacob Fish, "An Algebraic Two-Level Preconditioner for Asymmetric Positive-Definite Systems" in *International Journal for Numerical Methods in Engineering.*, 52 (2001), pp. 1443-1463.
- [28] W. Leontief, "The Structure of the American Economy" *Oxford U.P., NY*, 1951.