An Aggregation-based Algebraic Multigrid Method for Power Grid Analysis

Pei-Yu Huang, Huan-Yu Chou and Yu-Min Lee

1 Abstract

This paper develops an aggregation-based algebraic multigrid (AbAMG) method to efficiently analyze the power grids. Different from the conventional algebraic multigrid (AMG) scheme, an innovative constructing method of global inter-grid mapping operator is employed to not only enhance the sparsity of coarse grid operator for reducing the computational complexity but also solve the problem with better convergent rate. The proposed method can solve the circuit with size over two millions in 167.6 CPU seconds (including DC analysis, and transient analysis with 50 time steps), and the maximum error is less than 1%. The significant runtime improvement, over 26X faster than the InductWise [1] and over 1.25X faster than the conventional AMG method, and less memory usage, 40% of the memory usage in [1] are demonstrated.

2 Introduction

With the deep sub-micron technology, several features of chips (such as larger number of transistors and lower supply voltages) have made the quality of power delivery network become a key factor of high performance designs [2]. Generally, the power delivery network contains enormous amounts of circuit elements and efficient analyzers are necessary. Thus, general circuit solvers, such as SPICE/HSPICE, by using direct methods are not practicable for the power delivery analysis. In the past years, various efficient methods have been developed for the power distribution network analysis. The preconditioned conjugate gradient method was applied in [3], and the hierarchical methods were developed in [4] [5]. Multigrid-like methods were developed in [6] [7] to map the original problem to a reduced system by using the geometric property of circuit. However, these geometric multigrid techniques [6] [7] are hard to handle the coupling effects of mutual inductances.

To deal with the coupling effects, an algebraic multigrid (AMG) [9] based approach was developed in [8]. Generally, the mapping operator of AMG method [9] is determined by each row equation of $Ae \approx 0$, where A is the system matrix and e is the error vector of unknown variables. Although this mapping operator doesn't need the geometric information of circuit, its quality strongly counts on the selection of coarse grids and only contains the local information of A. Therefore, it may lose several important error terms, and degrade the convergent rate. To remedy this undesirable behavior, [8] proposed an adaptive coarse grids choosing method. However, its choosing strategy needs to reconstruct the mapping operator at each time step and may boost the CPU time.

To solve above problems, we develop a global mapping operator construction procedure based on aggregation AMG methods [10] [11]. The idea of aggregation originates in economics [12], where products in a large scale system are aggregated to become a small system. This procedure can significantly reduce the problem size, and still maintain the accurate representation of overall behaviors. An algebraic partition is performed to the fine grids and the system matrix is partitioned into several aggregated sub-matrices. The mapping operator is constructed from the global eigen-decomposition property of system. Theoretically, the errors in the directions of eigenvectors associated with larger eigenvalues can be rapidly reduced by relaxation, and the reduction of error in the directions of eigenvectors associated with smaller eigenvalues is stalled as the eigenvalues close to 0 [13]. Hence, the mapping operator P is composed by those eigenvectors associated with smaller eigenvalues. The mapping operator can project the system matrix to a reduced system matrix which is much sparser than the transformed system of conventional AMG method [8] and also achieves better convergent rate.

In Section 5, we will present the *resistance dominant property* of the system matrix of power delivery network as determining aggregations. With this property, the dimension of each aggregated sub-matrix is not greater than 4×4, and, hence, the eigenvectors of sub-matrices can be easily solved by using the QR algorithm [14]. Furthermore, an innovative matrix compensation algorithm with a global error estimation procedure is developed to further improve the quality of mapping operator. Moreover, the mapping operator construction procedure of AbAMG only needs to be performed once for all time steps. With these characteristics, the AbAMG method can construct a much better mapping operator than the traditional AMG method and can achieve better performance.

The remainder of this paper is organized as follows. The problem formulation, and the general AMG method are presented in Section 3, and 4, respectively. Then, the key part of AbAMG method, the global mapping operator construction, is described in Section 5. Finally, the experimental results, and conclusion are given in Section 6, and 7, respectively.

3 Problem Formulation

The power delivery network can be modeled as a RLKC (resistance, inductance, susceptance, and capacitance) circuit, and the devices connecting to the power delivery network are viewed as time-varying current sources with gate capacitances [5]. By using the modified nodal analysis (MNA) [15], the circuit equation is

$$\begin{bmatrix} G_n & -A_{l_n}^T \\ A_{l_n} & 0 \end{bmatrix} \begin{bmatrix} v_n(t) \\ i_l(t) \end{bmatrix} + \begin{bmatrix} C_n & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \dot{v}_n(t) \\ \dot{i}_l(t) \end{bmatrix}$$
$$= \tilde{B}\tilde{u}(t) + \begin{bmatrix} G_E \\ L_E \end{bmatrix} v_E(t), \quad (1)$$

The $v_E(t)$, $\tilde{u}(t)$, $v_n(t)$, and $i_l(t)$ are the vectors of independent voltage sources, independent current sources, unknown nodal voltages, and branch currents flowing through inductors, respectively. The G_n is the stamping matrix of resistors not connecting to $v_E(t)$. The C_n, and L are the stamping matrices of capacitors, and inductors, respectively, The A_{l_n} is the coefficient matrix related to inductors not connecting to $v_E(t)$. The $\tilde{B}$, G_E, and 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.

By applying the trapezoidal approximation with time step h, Equation (1) becomes

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

The system matrix in Equation (2) is not symmetric positive definite (SPD). Since AMG methods require the matrix to be SPD [13], we first, split the variable vector into a nodal voltage vector and a branch current vector. After using block matrix operations, the system equations become to

$$\left(2C_n/h + G_n + hA_{l_n}^T L^{-1} A_{l_n}/2\right) v_n(t+h) =$$
$$\left(2C_n/h - G_n - hA_{l_n}^T L^{-1} A_{l_n}/2\right) v_n(t) + 2A_{l_n}^T i_l(t)$$
$$+ \tilde{B}(\tilde{u}(t+h) + \tilde{u}(t)) + hA_{l_n}^T L^{-1} L_E v_E(t) + 2G_E v_E(t) \quad (3)$$

$$i_l(t+h) = i_l(t) - \frac{h}{2}L^{-1}A_{l_n}\left(v_n(t+h) + v_n(t)\right) + hL^{-1}L_E v_E(t). \quad (4)$$

Since G_n, C_n, and $K \equiv L^{-1}$ are SPD, the system matrix of Equation (3) is still SPD [8]. The symmetric property can save 50% of the memory usage, and K [16] is much sparser than L. Equation (3) is equivalent to solve $Ax = b$, where A is equal to $(2C_n/h + G_n + hA_{l_n}^T L^{-1} A_{l_n}/2)$, x is the unknown variable vector $v_n(t+h)$, and b is a known vector. Our main goal is to solve $v_n(t+h)$ under this algebraic equation. After $v_n(t+h)$ being solved, $i_l(t+h)$ can be calculated by Equation (4).

4 Algebraic Multigrid Methods

A two-level AMG method [9] is stated as follows. Given an $Ax = b$ problem, where A is a $N \times N$ matrix, x is a $N \times 1$ unknown variable vector, and b is a $N \times 1$ constant vector, the

relaxation is applied to eliminate the oscillatory error components on those fine nodes (unknown variables) with dimension N and the residual vector is $r = b - Ax$. Then, the residual vector is *restricted* to a few of coarse nodes with a smaller dimension $M < N$ by $r^c = Rr$, and the reduced system matrix is constructed by Galerkin operator $A^c = RAP$. Here, P is a $N \times M$ matrix, and $R = P^T$. On the coarse nodes, the residual equation, $A^c e^c = r^c$, is solved and the error correction vector e^c is *interpolated* back to fine nodes by $e = Pe^c$.

The smooth error components which are not eliminated well by the relaxation on fine nodes can be eliminated by the error correction vector e^c. A complementary two-level solution scheme can be constructed to overcome the stalling behavior of smooth error components in general iterative methods [9]. The corrective solution is obtained by $\hat{x} = x + e$, and a post-relaxation step is applied on fine nodes to ensure that the oscillatory error is not introduced through the coarse-node correction step. The two-level solution method can be easily extended to multilevel, and the efficiency of multigrid method strongly depends on the quality of mapping operator. The constructions of mapping operators of AMG and aggregation AMG are discussed in Section 4.1 and 4.2.

4.1 Mapping Operator Construction of AMG

Since the goal of AMG method is to develop a multilevel scheme to efficiently eliminate all error components, the smooth error components that the relaxation cannot eliminate well must be represented by the mapping operator of AMG. The traditional AMG methods construct the mapping operator with the property of algebraic smoothness, $(Ae)_i \approx 0$, which assumes that the residue is small at each row i after several relaxations. The equation $(Ae)_i \approx 0$ can be expanded as

$$a_{ii}e_i \approx -\sum_{j \neq i} a_{ij}e_j. \quad (5)$$

Using the *Color Scheme* algorithm [9], a coarser discretion is performed to the set of fine grid nodes and the set of coarse nodes can be determined according to the connections of each node i in the matrix graph of A. If $|a_{ij}| \geq \theta|a_{ii}|$, we say that node j strongly influences i and vice-versa. Here, θ is a threshold factor between 0 to 1. The nodes which strongly influence many nodes are defined as coarse nodes since they can approximate other nodes well. However, the above selecting procedure only considers the local connections of each node i, and may result in unsuitable coarse nodes and degrade the convergence rate. To further discuss the construction of mapping operator, Equation (5) can be rewritten as

$$a_{ii}e_i \approx -\sum_{j \in C_i} a_{ij}e_j - \sum_{k \in F_i^s} a_{ik}e_k - \sum_{m \in F_i^w} a_{im}e_m \quad (6)$$

The set of total original nodes is defined as $C \cup F$, where C is the set of coarse nodes and F is the set of remaining fine nodes. The set C_i is equal to $C \cap N_i$, and the set F_i is equal to $F \cap N_i$. Here, N_i is the set of neighboring nodes of node i. The set F_i

can be divided 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.

From Equation (6), an interpolation formula for each fine node error variable, e_i, by its neighboring coarse nodes error variables is defined as $e_i = \sum_{j \in C_i} w_{ij} e_j$ [9], where

$$w_{ij} = -\frac{a_{ij} + \sum_{k \in F_i^s} \dfrac{a_{ik} a_{kj}}{\sum_{l \in C_i} a_{kl}}}{a_{ii} + \sum_{m \in F_i^w} a_{im}}. \tag{7}$$

Here, the coefficient of each node error variable corresponding to the fine node in F_i^s is approximated by a sum of the coefficients of node error variables corresponding to the coarse nodes in C_i, and the coefficient of each node error variable corresponding to the fine node in F_i^w is simply added to the coefficinet a_{ii}. The selection of F_i^s and F_i^w is only determined by the coefficients in Equation (6) and this might lead to the inappropriate choice of F_i^s and F_i^w since those nodes of F_i^w with large errors should be labeled in the set of F_i^s. This unsuitable selection can decrease the convergent rate of AMG.

4.2 Mapping Operator Construction of Aggregation AMG

The idea of the mapping operator construction of aggregation AMG is based on the concept that the smooth error components are in the directions of system's eigenvectors associated with small eigenvalues [11] [13]. Firstly, an algebraic partition is performed to the connection graph of system matrix A, and the nodes which represent the unknown variables with strong influence between them are clustered together. After the aggregating procedure, each aggregated sub-matrix is eigen-decomposed and the eigenvectors related to the small eigenvalues are used to compose the mapping operator P. With accurately calculating the directions of system's smooth error components, the aggregation AMG can achieve better convergence rate than the traditional AMG methods. However, the small connected coefficients between aggregations are usually neglected or added to the diagonal elements of the aggregated sub-matrices. Hence, the convergent rate of aggregation AMG might be declined [10]. To improve the convergent rate, an innovative matrix compensation algorithm with a global error estimation procedure is developed in our AbAMG method and is described in the next section.

5 Global Mapping Operator Construction

In this section, a proposed global mapping operator construction for the AbAMG method is presented. At first, a node-by-node aggregation algorithm is shown in Section 5.1. Then, the global error estimation procedure and matrix compensation algorithm are stated in Section 5.2 and Section 5.3, respectively. After that, the mapping operator construction procedure is summarized in Section 5.4. Finally, the practicability of aggregation AMG method for the power delivery network analysis problem is addressed in Section 5.5.

5.1 Aggregation Algorithm

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

The node i and node j are defined to have a strong connection if any strong influence relation exists between them. Please refer to Section 4.1 for the meaning of strong influence. Otherwise, we say that they have a weak connection. The above definition of strong connection provides a reasonable measure when determining aggregations. A node having the maximum number of strong connections is a good candidate to be the starting node of aggregation algorithm, and those nodes with strong connections between them must be labeled in the same aggregation since their values might be highly correlated. On the other hand, the nodes with weak connections between them should be classed to different aggregations. Each node can only be included in an aggregation. The aggregation algorithm is shown in Fig. 1 and 2. The notation w_i of node i in Fig. 1 is the number of nodes with strong connection between i.

An example of aggregation algorithm is shown in Fig. 3. Given a system equation $Ax = b$, the system matrix A and its matrix connection graph are shown at the left hand side of Fig. 3. The modified system matrix A after aggregating and rearranging is shown at the top of the right hand side of Fig. 3. After that, this modified system matrix can be further simplified into three isolated aggregated sub-matrices a, b, and c, and the smooth error components of original system can be derived by the eigen-decomposition analysis of these sub-matrices. Because the nodes with weakly connected coefficients might have large errors in the global view, the way of discarding the coefficients or adding them into to the diagonal elements of aggregations [10] is not a suitable simplification strategy. Therefore, a global error estimation procedure is developed in Section 5.2 to estimate these troublesome error components, and the related matrix compensation algorithm is shown in Section 5.3 to remedy this defect.

5.2 Global Error Estimation

An intuitive way to get the information of errors which the relaxation cannot efficiently eliminate is applying the relaxation scheme to an auxiliary problem with a known solution. The system homogeneous problem, $Ax = 0$, provides us the proper information for this purpose. The error of the approximated solution of $Ax = 0$ can be known since its solution is a zero vector. By applying the relaxation several times to this problem with a random initial guess, we can get an error vector, e^g, which consists of the error components that the relaxation can not eliminate well. The above global error estimation is equivalent to find the error vector of the equation, $(Ae) \approx 0$, which is stated in section 4.1. This error vector can deliver the information of those
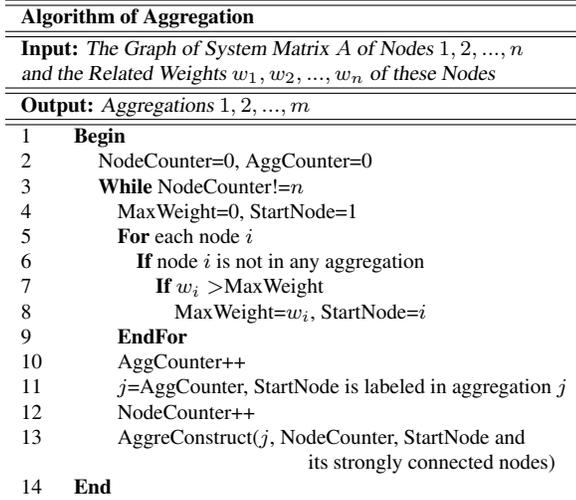
Algorithm of Aggregation
Input: *The Graph of System Matrix A of Nodes* $1, 2, ..., n$ *and the Related Weights* $w_1, w_2, ..., w_n$ *of these Nodes*
Output: *Aggregations* $1, 2, ..., m$

```
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 any aggregation
7                If w_i >MaxWeight
8                   MaxWeight=w_i, StartNode=i
9          EndFor
10         AggCounter++
11         j=AggCounter, StartNode is labeled in aggregation j
12         NodeCounter++
13         AggreConstruct(j, NodeCounter, StartNode and
                            its strongly connected nodes)
14   End
```

Figure 1. Algorithm of Aggregation

Algorithm of AggreConstruct
Input: *AggCounter* j, NodeCounter, *The StartNode* i *and it's Strongly Connected Nodes* $n_1, n_2, ..., n_s$

```
1    Begin
2       For each strongly connected node k of node i
3          If the node k is not in any aggregation
4             the node k is labeled in aggregation j
5             NodeCounter++
6             AggreConstruct(j, NodeCounter, node k and
                               its strongly connected nodes)
7       EndFor
8    End
```
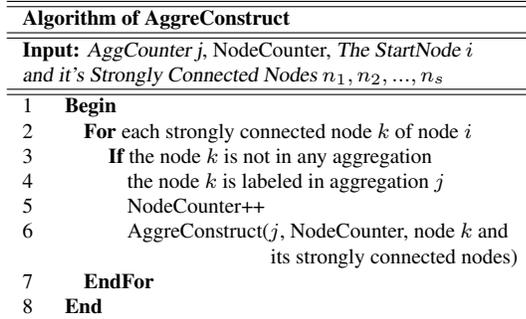
Figure 2. Algorithm of AggreConstruct

troublesome errors and can be employed in the following matrix compensation algorithm.

5.3 Matrix Compensation Algorithm

Given a linear algebraic system equation $Ae^g = 0$ with dimension N and the global error estimation e^g from Section 5.2, a node i is defined as a strong node if $e_i^g \geq \lambda \max(e_1^g, \cdots, e_N^g)$, and is defined as a weak node if $e_i^g < \lambda \max(e_1^g, \cdots, e_N^g)$. Here, $e_1^g, \cdots, e_N^g$ are the elements of vector e^g, and λ is a positive constant which is less than 1 and is chosen to be 0.25 in our algorithm.

The weakly connected coefficients related to the strong nodes are properly distributed to aggregated nodes with suitable weights as shown in Fig. 4, and the effects related to the weak nodes are simply neglected. With this compensation step, each modified aggregated sub-matrix is isolated from other sub-matrices and a better reduced system can be constructed to achieve a smaller error which is demonstrated in Section 6. The matrix compensation algorithm is shown in Fig. 4. The compensation procedure exactly matches with the weight calculation step in the traditional AMG methods. The modified sub-matrices are used for the coarse grid construction introduced in the next
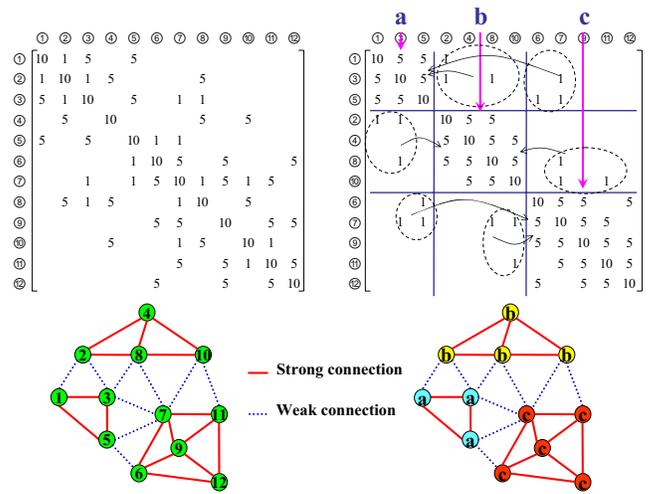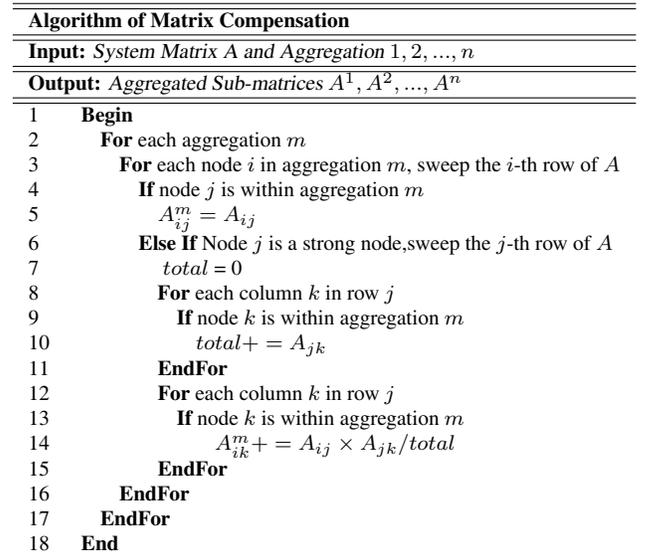

Figure 3. An example of aggregation.

Algorithm of Matrix Compensation
Input: *System Matrix A and Aggregation* $1, 2, ..., n$
Output: *Aggregated Sub-matrices* $A^1, A^2, ..., A^n$

```
1    Begin
2       For each aggregation m
3          For each node i in aggregation m, sweep the i-th row of A
4             If node j is within aggregation m
5                A_ij^m = A_ij
6             Else If Node j is a strong node,sweep the j-th row of A
7                total = 0
8                For each column k in row j
9                   If node k is within aggregation m
10                     total+ = A_jk
11               EndFor
12               For each column k in row j
13                  If node k is within aggregation m
14                     A_ik^m+ = A_ij × A_jk/total
15               EndFor
16         EndFor
17      EndFor
18   End
```

Figure 4. Algorithm of Matrix Compensation

subsection.

5.4 Mapping Operator Construction

In this section, we summarize the coarse grid construction of the aggregation-based AMG method. After executing the matrix compensation algorithm, those modified aggregated sub-matrices are isolated from each other, and the eigenvalue decomposition procedure is performed to each modified aggregated sub-matrix. Finally, the eigenvector corresponding to the smallest eigenvalue of each sub-matrix is used to compose the inter-grid transfer operator P, and the coarse grid operator can be constructed by the Galerkin operator $A^c = RAP$, where $R = P^T$. This mapping operator construction of AbAMG is determined from the global information of system matrix, and only needs to be performed once. The experimental results are shown in Section 6.
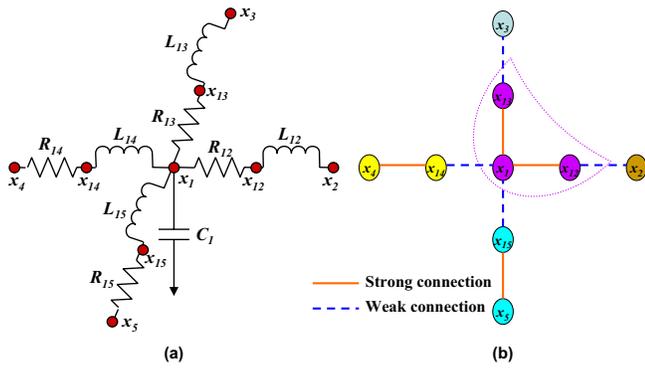
Figure 5. Structure of aggregation with size 3

5.5 Practicability of Aggregation AMG Method for Power Network Analysis

Before showing the experimental results of the proposed AbAMG method, we discuss the practicability of aggregation AMG method for the power network analysis. The aggregation AMG method is applied to the $Ax = b$ problem in Equation (3), where A is equal to $(2C_n/h + G_n + hA_{l_n}^T L^{-1} A_{l_n}/2)$, and the unknown variable vector x consists of all nodal voltages of power delivery circuit at each specific time.

From the predictive technology model (PTM) developed by the Berkeley university (http://www.eas.asu.edu/ ptm/), the unit value of R, L, and C in the $0.13\mu m$ technology are $0.046\Omega/\mu m$, $1.69pH/\mu m$, and $0.13011fF/\mu m$, respectively. For the length of wire segment being around $100\mu m$, the values of $2C_n/h$, G_n, and $hA_{l_n}^T L^{-1} A_{l_n}/2$ vary from 5E-3 to 8E-3, 2E-1 to 4E-1, and 2E-2 to 5E-2, respectively. The value of G_n term is over 8 times larger than other terms of A. From the aggregation algorithm discussed in Section 5.1, we can know that the determination of the aggregation of A is dominated by the effects of the resistances of G_n. This R-dominant phenomenon still exists if we choose different reasonable wire lengths or more advanced technologies such as $0.9\mu m$ or $0.65\mu m$ technologies. For the on-chip power delivery network with mesh structure, most aggregations have the size with 3 as shown in Fig. 5, and the maximum size of aggregation is equal to 4 if the via is connected to the structure of Fig. 5. Hence, the eigenvectors of each sub-matrix can be easily solved by using the QR algorithm [14].

6 Experimental Results

Circuit Size	MaxVD (v)	AMG		AbAMG without compensation		AbAMG with compensation	
		MaxE (%)	AvgE (%)	MaxE (%)	AvgE (%)	MaxE (%)	AvgE (%)
49.6K	0.150	1.173	0.077	1.165	0.077	0.973	0.067
199.2K	0.168	1.107	0.066	1.059	0.065	0.885	0.058
448.8K	0.165	1.170	0.065	1.072	0.064	0.955	0.056
798.4K	0.158	1.130	0.067	1.081	0.067	0.967	0.059

Table 1. Error percentage of RLKC circuits

This section demonstrates the speed and accuracy of the developed AbAMG solver. The power delivery networks are randomly generated as mesh circuits which consist of lumped

RLKC segments with many independent time varying current sources, and the supply voltage is 1 volt. This solver is implemented in C++ language and tested on a Pentium IV 3.4-GHz machine with 3 GB memory.

First, an efficient and accurate time domain solver Induct-Wise [1] is used to demonstrate the accuracy of our method. The results are shown in TABLE 1. The "MaxVD" is the maximum voltage drop of circuit, and is around 15% of the supply voltage for each circuit. The "MaxE" means the maximum error percentage, and "AvgE" is the average error percentage. Even without executing the matrix compensation algorithm, our AbAMG solver can still achieve a smaller maximum error than the traditional AMG method [9], and its average error is also less than or equal to the AMG method. With the help of matrix compensation algorithm, the AbAMG can further reduce the maximum error to be less than 0.973% (AMG is 1.173%) for each test circuit, and the average error to be less than 0.067% (AMG is 0.077%). The above demonstrates that the aggregation-based AMG method can accurately capture the error directions which can not be eliminated in the relaxation procedure of fine grid nodes, and our matrix compensation algorithm does improve the accuracy.

To show the efficiency of AbAMG solver, the analysis of DC and 50 transient time steps are executed and the results are compared with InductWise [1] and the conventional AMG method [9]. The results are shown in TABLE 2 for different RLKC circuits. The "RT" is the CPU run time, "Mem" denotes the memory usage, "RT*" is the run time of AbAMG without compensation, and "RT**" is the run time of AbAMG with compensation. The S_{In}, S_{AMG} and S_* are the speedup of AbAMG with compensation respect to InductWise, AMG, and AbAMG without compensation. The significant speed improvement, over 26 times faster than InductWise [1], over 1.25 times faster than AMG, and over 1.12 times faster than AbAMG without compensation, and less memory usage, 40% of the memory usage in [1] are observed. The proposed AbAMG solver only takes 167.6 CPU seconds to solve a circuit with size being 2.4472M, and this indicates that our simulator is very efficient in solving power delivery networks and capable of handling the circuit with size over millions.

The number of the none zero terms of system matrix and the iterative cycles needed for 100 time step transient analysis of AMG or AbAMG are shown in Table 3. Here, "Fine Grid NZ", and "Coarse Grid Nz" are the numbers of the non-zero terms of original system matrix, and the reduced system matrix, respectively. The "Cycle", "Cycle*", and "Cycle**" are the total numbers of multilevel cycles needed in the AMG method, the AbAMG without compensation, and the AbAMG with compensation, respectively. From the results, we can see that the number of non-zero terms in the reduced system matrix of AbAMG is only one third of the reduced system matrix of AMG method. In the other words, the reduced system matrix by using AbAMG is sparser than the matrix by using the AMG method. Hence, its computational load is much lighter than the AMG's. Further-

Circuit Size	InductWise [1]		AMG		AbAMG					
	RT (s)	Mem (MB)	RT (s)	Mem (MB)	RT* (s)	RT** (s)	Mem (MB)	Speed Up S_{In}	S_{AMG}	S_*
49.6K	78.34	111	3.95	46	3.59	2.97	40	26.38	1.33	1.21
199.2K	391.70	424	15.88	182	14.24	12.72	156	30.80	1.25	1.12
448.8K	1576.00	994	38.19	407	32.59	28.31	351	55.67	1.35	1.15
798.4K	2903.00	1547	68.22	721	59.33	51.81	624	56.03	1.32	1.15
1.248M	×	×	105.59	1130	93.75	83.16	974	-	1.27	1.13
1.7976M	×	×	152.36	1627	137.31	119.36	1401	-	1.28	1.15
2.4472M	×	×	×	×	196.53	167.60	1907	-	-	1.17

Table 2. Runtime and Speed Up Comparison of RLKC Circuits. "×" denotes this methodology is out of memory.

Circuit Size	Fine Grid NZ	AMG		AbAMG		
		Coarse Grid NZ	Cycle	Coarse Grid NZ	Cycle*	Cycle**
49.6K	119.5K	92K	118	33.8K	127	101
199.2K	480.0K	370K	113	135.6K	131	100
448.8K	1082.0K	834K	111	305.4K	131	101
798.4K	1925.0K	1484K	114	543.0K	135	103
1.248M	3009.0K	2320K	109	849.0K	131	102
1.7976M	4335.0K	3342K	108	1223.0K	132	100

Table 3. Comparison between AbAMG and AMG

more, because the matrix compensation algorithm can compensate those big error terms from the information of system homogenous problem, $Ae^g = 0$, the number of multilevel cycles used by AbAMG with compensation approaches the minimum number needed (Ideally, the minimum number of cycle for 100 time steps analysis is 100.) and is much smaller than the AMG method. Therefore, the speed of analysis can be improved further.

7 Conclusion

An AbAMG solver for power distribution networks has been developed. The proposed method can handle the circuit size with more than two million in 167.6 CPU seconds, and the maximum error of each test circuit is less than 1%. The significant speed improvement and the less memory usage show that this AbAMG method is very suitable for analyzing the power delivery network. The global mapping operator construction with a matrix compensation algorithm does improve the performance of AbAMG, and a sparser reduced system matrix is performed with the better convergence rate than the traditional AMG method.

When considering the pure RC PG network, the small aggregation size property can be still held by applying the algorithm mentioned in [11]. This will be our future work to preserve the small aggregation size property when dealing with the pure RC PG network.

Acknowledgements

References

[1] T.-H. Chen, C. Luk, H. Kim, and C. C.-P. Chen, "INDUCTWISE: Inductance-wise interconnect simulator and extractor", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , vol. 22, no. 7, pp. 884-894, July 2003.

[2] M. K. Gowan, L. L. Biro, and D. B. Jackson , "Power consideration in the design of the Alpha 21264 microprocessor", *IEEE/ACM Design Automation Conference*, pp. 726-31, Jun. 1998.

[3] T.-H. Chen, and C. C.-P. Chen, "Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods", *IEEE/ACM Design Automation Conference*, pp. 259-62, Jun. 2001.

[4] Y.-M Lee, Y. Cao, T.-H Chen, J. M. Wang, and C. C.-P Chen, "HiPRIME: Hierarchical and Passivity Preserved Interconnect Macromodeling Engine for RLKC Power Delivery" *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 6, pp. 797-806, Jun. 2005.

[5] M. Zhao, R. V. Panda, S. S. Sapantnekar, and D. Blaauw, "Hierarchical analysis of power distribution networks", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 2, pp. 159-68, Feb. 2002.

[6] S. Nassif, and J. Kozhaya, "Fast power grid simulation", *IEEE/ACM Design Automation Conference*, pp. 156-61, Jun. 2000.

[7] J. N. Kozhaya, S. R. Nassif, and F. N. Najm, "Multigrid-like technique for power grid analysis", *IEEE/ACM International Conference on Computer Aided Design*, 2001.

[8] Z. Zhu, B. Yao, and C.-K. Cheng, "Power network analysis using an adaptive algebraic multigrid approach", *IEEE/ACM Design Automation Conference*, pp. 105-8, Jun. 2003.

[9] W. L. Briggs, "A Multigrid Tutorial", *SIAM 2000.*

[10] J. Fish, and V. Belsky, "Generalized aggregation multilevel solver", *International Journal for Numerical Methods in Engineering.*, vol. 40, pp. 4341-61, 1997.

[11] T. E. Giddings and J. Fish, "An algebraic two-level preconditioner for asymmetric positive-definite systems", *International Journal for Numerical Methods in Engineering*, vol. 52, pp. 1443-63, 2001.

[12] W. Leontief, "The Structure of the American Economy", *Oxford U.P., NY, 1951.*

[13] A. Brandt, "Algebraic multigrid theory: the symmetric case", *Applied Mathematics and Computation.*, vol.19 no.1-4, p.23-56, Jul. 1986.

[14] D. S. Watkins, "Fundamentals of Matrix Computations", *John Wiley & Sons, Inc.*, 1991.

[15] L. T. Pillage, R. A. Rohrer, and C. Visweswariah, "Electronic Circuit and System Simulation Methods", *McGRAW-HILL Book Co..*

[16] A. Devgan, "How to efficiently capture on-chip inductance effects: introducing a new circuit element K", *IEEE/ACM International Conference on Computer Aided Design*, pp. 150-5, Nov. 2000.

IEEE COMPUTER SOCIETY