

Efficient Large-Scale Power Grid Analysis Based on Preconditioned Krylov-Subspace Iterative Methods

Tsung-Hao Chen and Charlie Chung-Ping Chen
 Electrical and Computer Engineering
 University of Wisconsin-Madison
 Madison, WI 53706-1691

tchen@cae.wisc.edu

chen@engr.wisc.edu

ABSTRACT

In this paper, we propose *preconditioned Krylov-subspace iterative methods* to perform efficient DC and transient simulations for large-scale linear circuits with an emphasis on power delivery circuits. We also prove that a circuit with inductors can be simplified from MNA to NA format, and the matrix becomes an s.p.d matrix. This property makes it suitable for the *conjugate gradient with incomplete Cholesky decomposition as the preconditioner*, which is faster than other direct and iterative methods. Extensive experimental results on large-scale industrial power grid circuits show that our method is over 200 times faster for DC analysis and around 10 times faster for transient simulation compared to SPICE3. Furthermore, our algorithm reduces over 75% of memory usage than SPICE3 while the accuracy is not compromised.

1. INTRODUCTION

Due to the increasing complexity and power consumption of VLSI chips, power grid analysis has become an important issue. A robust power network design has to guarantee the correctness of circuit functionalities without slowing down operations. An improper design of power grid will result in excessive drops and fluctuations in the voltages supplied to the devices. If the voltage drop becomes too large, it will not only increase the gate delays, but even worse, cause logical errors. Numerous researchers studied the impact and proposed solutions to the problem [1, 2, 3, 4].

There are many sources of power fluctuation such as IR -drop, Ldi/dt -drop, and resonance issues. Traditionally power grid analysis is often emphasized on IR -drop. Recently, due to the rapidly increasing operation frequency, the dynamic power fluctuation caused by LdI/dt also becomes significant [5]. To further reduce the power fluctuation, large amount of on-chip decoupling capacitors are added to act as temporary on-chip local power supplies. Thus, to accurately model and verify the quality of power delivery, both capacitance and inductance should be considered in the power grid analysis problem. Because of the tremendous amount of the power delivery elements, general-purpose circuit simulators such as SPICE are not adequate for power grid analysis because of CPU speed and memory limitation. Although there are several methods [6, 7] proposed to alleviate this problem by sparsification, the accuracy is compromised.

In this paper, we propose *preconditioned Krylov-subspace iterative methods* to perform fast DC and transient simulations for large-scale linear circuits with an emphasis on

power delivery circuits. This method has been shown to be significantly faster than traditional iterative methods without preconditioning. In order to take advantage of the fast convergence of these methods, we derive and prove that the Nodal Analysis is feasible for general RLC circuits and the system matrix for transient simulation is indeed symmetric positive definite (s.p.d), which is long believed not feasible. Furthermore, we develop an effective preconditioning scheme based on the *incomplete Cholesky decomposition method* to significantly reduce the run time and memory requirements.

2. REVIEW OF MNA EQUATIONS

First, we briefly review the *Modified Nodal Analysis* (MNA) equations of RLC circuits. Given a linear circuit, the adjacency matrix, \mathbf{A} , can be determined from the directed graph by the following rule.

$$\mathbf{A}_{ij} = \begin{cases} +1 & \text{if node } j \text{ is the source of branch } i \\ -1 & \text{if node } j \text{ is the sink of branch } i \\ 0 & \text{otherwise} \end{cases}$$

Obviously each row of the adjacency matrix only contains two non-zero elements (only one if the branch connects to the ground). This matrix represents the connectivity of a circuit, and the Kirchhoff's law in terms of it is

$$\text{KCL: } \mathbf{A}^T \mathbf{i}_b = \mathbf{0} \quad \text{and} \quad \text{KVL: } \mathbf{A} \mathbf{v}_n = \mathbf{v}_b, \quad (1)$$

where \mathbf{i}_b and \mathbf{v}_b are the vectors of branch currents and voltages respectively, and \mathbf{v}_n is the vector of the node voltages.

For the power grid analysis, we are only interested in passive elements, RLC, and also assume that the circuit only contains independent current sources. (We will explain voltage sources later.) Therefore, the adjacency matrix, the branch voltages and the branch currents can be partitioned into these forms.

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_i \\ \mathbf{A}_g \\ \mathbf{A}_c \\ \mathbf{A}_l \end{bmatrix}, \quad \mathbf{v}_b = \begin{bmatrix} \mathbf{v}_i \\ \mathbf{v}_g \\ \mathbf{v}_c \\ \mathbf{v}_l \end{bmatrix}, \quad \mathbf{i}_b = \begin{bmatrix} \mathbf{i}_i \\ \mathbf{i}_g \\ \mathbf{i}_c \\ \mathbf{i}_l \end{bmatrix}$$

The subscripts i , g , c , and l stand for branches which contain independent current sources, resistors, capacitors, and inductors, respectively.

The relationships between branch currents and voltages are as follows

$$\mathbf{i}_i = -\mathbf{I}_s(t), \quad \mathbf{i}_g = \mathcal{G} \mathbf{v}_g, \quad \mathbf{i}_c = C \frac{d}{dt} \mathbf{v}_c, \quad \mathbf{v}_l = \mathcal{L} \frac{d}{dt} \mathbf{i}_l \quad (2)$$

In Equation(2), $\mathbf{I}_s(t)$ is the vector of current source values. \mathcal{G} and C are diagonal matrices whose diagonal elements are positive for any physical circuit. The diagonal elements of matrix \mathcal{L} are the values of self inductances, and the off-diagonal terms are mutual inductances. Although \mathcal{L} is not a diagonal matrix, it still satisfies the s.p.d property.

By the method of MNA, we combine Equation (1) and (2), and eliminate unnecessary branch currents. In RLC circuits, only the branch currents running through inductors have to be kept in the equations, and then we obtain the following:

$$\tilde{\mathbf{G}} \mathbf{x} + \tilde{\mathbf{C}} \dot{\mathbf{x}} = \tilde{\mathbf{B}} \mathbf{i}_i, \quad (3)$$

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA
 Copyright 2001 ACM 0-89791-88-6/97/05 ...\$5.00.

$$\text{in which } \tilde{\mathbf{G}} = \begin{bmatrix} \mathbf{G} & \mathbf{A}_l^T \\ -\mathbf{A}_l & \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{C}} = \begin{bmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathcal{L} \end{bmatrix}, \quad (4)$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}_n \\ \mathbf{i}_l \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} -\mathbf{A}_l^T \\ \mathbf{0} \end{bmatrix}$$

and $\mathbf{G} = \mathbf{A}_g^T \mathcal{G} \mathbf{A}_g$, $\mathbf{C} = \mathbf{A}_c^T \mathcal{C} \mathbf{A}_c$. Notice that the matrices $\tilde{\mathbf{G}}$ and $\tilde{\mathbf{C}}$ in (4) still remain *symmetric*.

3. NA FOR LINEAR CIRCUITS

Although MNA provides a good solution for general circuits, the introduction of extra current variables makes the system matrix non-positive definite, which is crucially necessary for the efficiency and fast convergence of both iterative and direct methods since the Cholesky decomposition takes only half of multiplications and memory references than the LU decomposition. However, it was believed that it is infeasible to use the Nodal Analysis (NA) for general RLC circuits due to the need of additional current variables. In the following theorem, we show that by eliminating the current variables, we can obtain an NA formulation for RLC circuits, which is s.p.d. Hence the Cholesky decomposition and the conjugate gradient method can be used in this problem.

THEOREM 1. *The system matrices of the transient simulation for Forward Euler, Backward Euler, or Trapezoidal integration approximations can be formulated as the following NA scheme, $\tilde{\mathbf{A}} \mathbf{v}_n = \mathbf{b}$, where \mathbf{v}_n is the nodal voltages, and $\tilde{\mathbf{A}}$ equals to $[\mathbf{G} + \frac{2}{\Delta t} \mathbf{C} + \frac{\Delta t}{2} \mathbf{L}]$.*

PROOF. To make the discussion brief, we only show the proof of the Trapezoidal case and some details are omitted. By the Trapezoidal differential approximation, we can replace $\tilde{\mathbf{C}} \dot{\mathbf{x}}$ with its finite difference formula and get

$$\tilde{\mathbf{C}} \mathbf{x}(t + \Delta t) = \tilde{\mathbf{C}} \mathbf{x}(t) + \frac{\Delta t}{2} \tilde{\mathbf{C}} \dot{\mathbf{x}}(t + \Delta t) + \frac{\Delta t}{2} \tilde{\mathbf{C}} \dot{\mathbf{x}}(t)$$

Substituting the above equation into Equation(3), and collecting the variables, we can get the following equation.

$$\left(\tilde{\mathbf{G}} + \frac{2}{\Delta t} \tilde{\mathbf{C}} \right) \mathbf{x}(t + \Delta t) = - \left(\tilde{\mathbf{G}} - \frac{2}{\Delta t} \tilde{\mathbf{C}} \right) \mathbf{x}(t) + \mathbf{B} \mathbf{i}_l(t + \Delta t) + \mathbf{B} \mathbf{i}_l(t) \quad (5)$$

Substituting (2) and (4) to (5) and performing block matrix operations, we can obtain two equations:

$$\left(\mathbf{G} + \frac{2}{\Delta t} \mathbf{C} \right) \mathbf{v}_n(t + \Delta t) + \mathbf{A}_l^T \mathbf{i}_l(t + \Delta t) = \left(-\mathbf{G} + \frac{2}{\Delta t} \mathbf{C} \right) \mathbf{v}_n(t) + \mathbf{A}_l^T \mathbf{i}_l(t) + \mathbf{A}_l^T (\mathbf{I}_s(t + \Delta t) + \mathbf{I}_s(t)) \quad (6)$$

$$-\mathbf{A}_l \mathbf{v}_n(t + \Delta t) + \frac{2}{\Delta t} \mathcal{L} \mathbf{i}_l(t + \Delta t) = \mathbf{A}_l \mathbf{v}_n(t) + \frac{2}{\Delta t} \mathcal{L} \mathbf{i}_l(t) \quad (7)$$

Since \mathcal{L} is positive definite, it is also nonsingular. This ensures the existence of \mathcal{L}^{-1} which is also positive definite. Thus, we can multiply $\frac{\Delta t}{2} \mathcal{L}^{-1}$ to both sides of Equation (7) and obtain

$$\mathbf{i}_l(t + \Delta t) = \frac{\Delta t}{2} \mathcal{L}^{-1} \mathbf{A}_l (\mathbf{v}_n(t + \Delta t) + \mathbf{v}_n(t)) + \mathbf{i}_l(t)$$

Substituting into (6), we derive

$$\left(\mathbf{G} + \frac{2}{\Delta t} \mathbf{C} + \frac{\Delta t}{2} \mathbf{A}_l^T \mathcal{L}^{-1} \mathbf{A}_l \right) \mathbf{v}_n(t + \Delta t) = \left(-\mathbf{G} + \frac{2}{\Delta t} \mathbf{C} - \frac{\Delta t}{2} \mathbf{A}_l^T \mathcal{L}^{-1} \mathbf{A}_l \right) \mathbf{v}_n(t) + \mathbf{A}_l^T (\mathbf{I}_s(t + \Delta t) + \mathbf{I}_s(t)) \quad (8)$$

which can be written as

$$\tilde{\mathbf{A}} \mathbf{v}_n = \mathbf{b} \quad (9)$$

$$\text{where } \tilde{\mathbf{A}} = \mathbf{G} + \frac{2}{\Delta t} \mathbf{C} + \frac{\Delta t}{2} \mathbf{L} \quad (10)$$

Similar to \mathbf{G} and \mathbf{C} , we define $\mathbf{L} = \mathbf{A}_l^T \mathcal{L}^{-1} \mathbf{A}_l$. \square

THEOREM 2. *A matrix $\tilde{\mathbf{A}}$ is stamped from a RLC circuit in NA format, then $\tilde{\mathbf{A}}$ is symmetric positive-definite.*

PROOF. The inverse of an s.p.d matrix is also an s.p.d matrix, and $\mathbf{x}^T (\mathbf{A}_l^T \mathcal{L}^{-1} \mathbf{A}_l) \mathbf{x} = (\mathbf{A}_l \mathbf{x})^T \mathcal{L}^{-1} (\mathbf{A}_l \mathbf{x}) > 0$. Since \mathbf{G} , \mathbf{C} and \mathbf{L} are all s.p.d matrices, the s.p.d of $\tilde{\mathbf{A}}$ can be proved by linearity readily. \square

THEOREM 3. *In the case when there is no mutual inductance in the circuit, the matrix $\tilde{\mathbf{A}}$ is diagonally dominant, that is $\tilde{a}_{jj} \geq \sum_{i=1, i \neq j}^n |\tilde{a}_{ij}|$, $j = 1, \dots, n$.*

PROOF. Since the mutual inductance is absent, the inductance matrix \mathcal{L} and its inverse \mathcal{L}^{-1} become diagonal matrices, which is also the case for \mathbf{G} and \mathbf{C} . \mathbf{G} , \mathbf{C} and \mathbf{L} are all diagonally dominant matrices, then we have $\tilde{\mathbf{A}}$ is also a diagonally dominant matrix. \square

4. POWER GRID MODEL AND STAMPING

Our power grid model is shown as Figure 1. There are two layers in the model. The top and the bottom layers represent the high and low voltage supplies respectively. Between these two layers are decoupling and other parasitic capacitances, and independent current sources. The current sources are pulse currents generated when the gates are switching. We use SPICE to simulate the behavior when gates are switching, and transfer the waveforms of currents draining from the power supply into piecewise linear current sources. These currents can represent the currents drained from either gates or functional unit blocks.

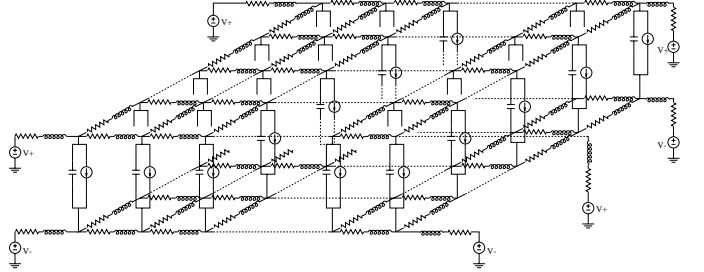


Figure 1: Power Grid Model

Note that we include inductors in our model, which has not been concerned in previous studies. The effect of inductance is not obvious in low frequency operation. When the frequency is high, we cannot ignore the impact. As mentioned before, because of inductors, we have to insert additional variables of branch currents in MNA. That makes the problem more difficult to solve.

From Equation(10), if the circuit to be simulated does not include mutual inductance, \mathcal{L} is only a diagonal matrix. Then \mathcal{L}^{-1} is nothing but a diagonal matrix whose diagonal values equal to the reciprocal of each corresponding diagonal element, l_i , of the matrix \mathcal{L} . Hence, all the elements can be directly stamped into the matrix by Norton equivalent circuits, which is depicted in Figure 2.

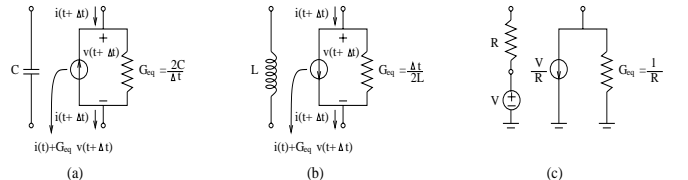


Figure 2: Norton Equivalent for Stamping (a) Capacitor (b) Inductor (c) Independent Voltage Source

Since every element converts into a Norton equivalent circuit, the resistor part (G_{eq}) can be directly stamped into the matrix, $\tilde{\mathbf{A}}$, and the independent current part can be stamped into the right hand side, \mathbf{b} , of the Equation(9). This process is exactly the same as Equation(8). Figure 2(c) stands for the independent voltage sources. By using Norton equivalent circuits, we eliminate one node that connects to the voltage source. This allows no extra current variables introduced into the problem, and keeps the matrix s.p.d.

5. PRECONDITIONED METHODS

5.1 PCG Algorithm

The conjugate gradient (CG) algorithm is one of the best known iterative techniques for solving sparse s.p.d linear systems [9, 10]. This method is a realization of an orthogonal projection technique on to the Krylov subspace $\mathcal{K}_m(r_0, \tilde{\mathbf{A}})$. In other words, the search direction of each iteration is A-orthogonal. This guarantees that CG can finish in n iterations. The finite termination property makes CG more excellent than other iterative methods in converging rate. Since the s.p.d is proven in Theorem 2, we can take advantage of this property and minimize the residual by the CG.

Another attractive feature of CG is that it does not require all previous residuals and search directions to generate an orthogonal vector. The new search direction can be generated only from the previous residual. This makes memory utilization more efficient. In our work, we also implemented the preconditioned generalized minimum residual (GMRES) method. Unlike CG, GMRES has to save all the previous vectors. The experimental result shows that GMRES indeed costs more memory space and run time than CG. However, GMRES can handle non-symmetric matrix, which makes it possible to solve problems containing elements other than RLC.

Compared to direct solvers, lack of robustness is a widely recognized weakness of iterative solvers. Although CG guarantees finite termination, n is still too huge to count on. It can be improved by *precondition* technology. The preconditioned conjugate gradient (PCG) method pre-multiplies (or multiplies, which is not introduced in this paper) an s.p.d matrix \mathbf{M}^{-1} . \mathbf{M} is the preconditioner. The solution space becomes $\mathbf{M}^{-1}\tilde{\mathbf{A}}\mathbf{v}_n = \mathbf{M}^{-1}\mathbf{b}$, and a good preconditioner can make the condition number of $(\mathbf{M}^{-1}\tilde{\mathbf{A}})$ close to 1. This means $(\mathbf{M}^{-1}\tilde{\mathbf{A}}) \approx \mathbf{I}$.

5.2 Incomplete Cholesky Factorization

To find a "good" preconditioner is important for preconditioned iterative methods. The best preconditioner of a matrix is the inverse of the matrix, which makes the preconditioned matrix an identity matrix and the condition number equal to 1. However, the cost of full matrix inversion is very expensive. The idea is to find a relatively inexpensive way to obtain an approximate inversion while minimizing the overall computation cost. One way is to factorize the matrix incompletely and use the inverse of this new matrix as its preconditioner. For example, given an s.p.d matrix A , instead of performing a full Cholesky decomposition of A ($= GG^T$), we perform an *incomplete* Cholesky decomposition of A and get $A = \tilde{G}\tilde{G}^T + E$, where E is the error of this approximation. If the norm of E is sufficiently small, we can anticipate $\tilde{A} = (\tilde{G}^T\tilde{G})^{-1}A \approx \mathbf{I}$ or the condition number of \tilde{A} is significantly smaller than that of A .

There are several variants of incomplete decomposition methods [10]. After implementing most of them, we found the ILUT (Incomplete LU Decomposition with Threshold)

is one of the most effective methods for power grid simulation. It is similar the full LU decomposition except that ILUT throws out the small fill-ins (less than the threshold value) during factorization. This becomes a tradeoff between thresholds and the iteration count. Allowing more fillins will reduce the iteration count with the cost of more memory and longer run time for each iteration.

One thing to notice is that although the complete Cholesky decomposition exists for an s.p.d matrix, the incomplete one may not exist because of the fact that it throws out some elements. However, if a matrix is an *M-Matrix* then its ILUT-type incomplete Cholesky decomposition is guaranteed to exist [10]. An *M-Matrix* is a matrix such that the elements of its inversion are all positive. It turns out that a diagonally dominant matrix is an *M-Matrix* and hence we can apply simple ILUT to the RLC (without mutual inductance) circuit without resort to further modification. In the case when mutual inductances are present, a simple modification to the ILUT algorithm is able to guarantee the existence of the incomplete factorization [10].

ICD with Threshold Control

Since $\tilde{\mathbf{A}}$ is an s.p.d matrix (Theorem 2), the Cholesky decomposition would be the most suitable method due to several reasons. First, the Cholesky decomposition saves half of the multiplications and storage space than LU decomposition. Second, it is stable regardless of the ordering and hence no pivoting is necessary for numerical stability. In order to reduce the number of fillins, we use the RCM (Reverse-Cathill-Mckee) ordering method before decomposing the matrix. After reordering, the new matrix and its Cholesky factor G are both banded, which means we don't need to calculate those elements outside the band. Assume the width of the band is w , the run time is reduced to $\frac{nw^2}{2}$. In our simulations, w can be less than $\frac{n}{100}$.

About the threshold value, there are several different drop-tolerance policies presented in previous works [12]. $ILU(0)$ is a no-fill strategy, which is equivalent to setting threshold to 0. Choosing a fixed threshold value or drop-tolerance level is also used. A simple and well known policy is to set tolerance for element a_{ij} to be $c\sqrt{a_{ii}a_{jj}}$, where c is a constant. The drawback of this policy is that it costs too much to calculate $\sqrt{a_{ii}a_{jj}}$ for each non-zero a_{ij} . Thus, in order to reduce the run time of generating preconditioner, the mean of each diagonal value times a constant is used as the threshold value in our work. That is $T = \frac{c}{n} \sum_{i=1}^n a_{ii}$, and it implies that the number of fillins can be limited under a certain percentage of that with the full factorization.

6. SIMULATION RESULT

The PCG power grid simulator was implemented in C language, and simulated on an AlphaPC DP264 666Mhz system. For the test sets, we use the power grid model discussed in Section 4. Every iterative result reaches the precision that the L2 norm of residuals ($\|r\|_2 = \sqrt{\sum_{i=1}^n r_i^2}$) is less than 10^{-10} .

Figure 3 shows the efficiency and robustness of PCG. (a) shows the reduction of nonzeros between complete and incomplete factorizations, and (b) shows the improvement of iteration counts. The result shows that ICDT eliminates more than 90% of nonzeros in full Cholesky decomposition, and reduces more than 95% iterations. As the circuit size goes up, this algorithm has even better improvement. Although the precondition process makes each iteration slower, the total run time of PCG is still much better than CG.

The DC run time and memory usage are shown in Table 1 and Figure 4. Direct methods (SPICE3 and complete

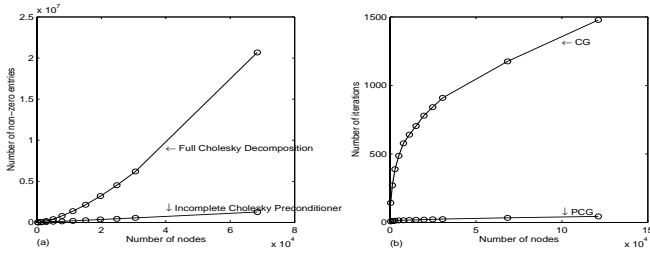


Figure 3: Compare numbers of (a) nonzeros of full and incomplete Cholesky decompositions (b) iterations of preconditioned and non-preconditioned CG

Cholesky) are much slower than iterative methods, and the precondition technology makes iterative methods even better. As mentioned before, the PGMRES is a little bit slower and requires more memory space than PCG. Compared to the performance of SPICE3 and our large-scale linear circuit simulator, our simulator is over 200 times faster than SPICE3 and only requires less than $\frac{1}{5}$ memory space for DC analysis. The circuit with more than 1-million nodes and 5-million elements takes less than 20 minutes to finish the DC analysis.

elements	nodes	SPICE3	CD	CG	PGMRES	PCG
3405	1326	1.59	0.08	0.10	0.05	0.05
13205	5046	36.31	1.17	0.80	0.31	0.29
39905	15126	390.50	10.29	4.66	1.78	1.63
52005	19686	779.83	18.96	14.88	2.78	2.75
81005	30606	2102.10	48.55	15.83	6.26	4.71
181505	68406	-	168.08	54.76	18.52	13.73
322105	121206	-	-	140.24	59.53	30.57
723005	271806	-	-	-	105.30	87.19
2886002	734446	-	-	-	-	674.19
5128005	1014846	-	-	-	-	1181.60

Table 1: DC analysis run time (sec)

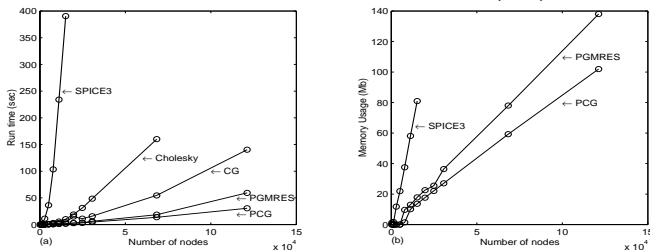


Figure 4: Compare the (a) run time (sec.) (b) memory usage (Mb) of several different methods in DC analysis.

Table 2 and Figure 5 show the results of transient analysis. The model presented in Figure 1 is used in this simulation. Figure 5(a) and (b) compare the waveforms at one of the nodes in the power grid. Since we execute the PCG until the norm of its residuals is below 1×10^{-10} , the accuracy of iterative methods is compatible with the direct solvers. Figure 5(c) shows PCG simulators is about 10 times faster than SPICE3 and much better for larger circuits. We also test some circuits obtained from Hewlett-Packard Company, and the result is equally good.

7. CONCLUSION AND FUTURE WORK

We introduced the preconditioned iterative methods to solve sparse matrices. We also prove that the matrix stamped from RLC circuit can be s.p.d. Therefore the ICDT and the PCG are applicable for solving the matrix. We also compare several different simulation methods. The performance of PCG is impressive, and shows that it's much more suitable for simulating large-scale linear circuits, such as power grids, than general-purpose circuit simulators.

# of elements	# of nodes	SPICE3	PCG	speedup
1146	690	3.51	0.34	10.32
4251	2570	26.26	5.69	4.62
16506	9930	313.57	55.82	5.62
64986	39050	5702.08	359.43	15.86
101226	60810	13953.48	783.10	17.82
260826	136210	-	2150.36	-

Table 2: Transient simulation run time (sec)

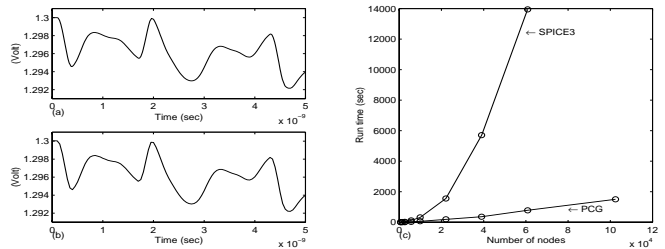


Figure 5: Compare (a) result waveform of SCPIE3 (b) of PCG (c) run time in transient simulation

We didn't discuss the circuits with mutual inductances in this paper, since it involves an inversion of matrix \mathcal{L} and can't be stamped directly from the circuit. Due to the rapidly increasing operation frequency, inductances act a more important role in the power grid analysis. In the future, we plan to include mutual inductances in our simulation engine.

8. ACKNOWLEDGMENTS

This work is partially supported by NSF Grant CCR-0093309, Intel Corp., Avant! Corp., and Hewlett Packard Company. The authors would like to thank Shekhar Borkar, Tanay Karnik, Atila Alvandpour, Ram Krishnamurth, Florin Dartu and Noel Menezes from Intel, Li-pen Yuan and Y.Z. Liao from Avant!, Norman Chang and Shen Lin from HP, and David Blaauw and Rajendran Panda from Motorola for their invaluable discussions and supports. We also thank Wei-Fen Lin for her comments.

9. REFERENCES

- [1] G. Steele, D. Overhauser, S. Rochel, and Z.Hussain, "Full-chip verification methods for DSM power distribution systems," DAC, 1998
- [2] A. Dharchoudhury, R. Panda, D. Blaauw, R. Vaidyanathan, B. Tutuianu, and D. Bearden, "Design and analysis of power distribution networks in PowerPC microprocessors," DAC, 1998.
- [3] H. Chen and D. Ling, "Power supply noise analysis methodology for deep-submicron VLSI chip design," DAC, 1997
- [4] Y. Jjiang and K. Cheng, "Analysis of performance impact caused by power supply noise in deep submicron devices," DAC, 1999
- [5] R. Panda, D. Blaauw, R. Chaudhry, V. Zolotov, B. Young, and R. Ramaraju, "Model and analysis for combined package and on-chip power grid simulation," ISLPED, 2000
- [6] S. Nassif and J. Kozhaya, "Fast power grid simulation," DAC, 2000
- [7] M. Zhao, R. Panda, S. Sapatnekar, T. Edwards, R. Chaudhry, and D. Blaauw, "Hierarchical analysis of power distribution networks," DAC, 2000
- [8] R. Freund, "Passive reduced-order models for interconnect simulation and their computation via Krylov-subspace algorithms," Numerical Analysis Manuscript No.98-3-06, Bell Lab., 1998
- [9] Hasan Dağ, "Iterative methods and parallel computation for power systems," a Dissertation for the Degree of Ph.D., UW-Madison, 1996
- [10] Yousef Saad, "Iterative methods for linear systems," 2nd Ed., 2000
- [11] M. Jones and P. Plassmann, "An improved incomplete Cholesky factorization," Argonne National Laboratory, 1992.
- [12] Michael T. Heath, "Parallel direct methods for sparse linear systems," Parallel Numerical Algorithms, p55-90, 1997