

Efficient Analysis of Large-Scale Power Grids based on a Compact Cholesky Factorization

Hong Li, Jitesh Jain, Venkataramanan Balakrishnan and Cheng-Kok Koh
School of Electrical and Computer Engineering
Purdue University, West Lafayette, IN 47907-1285
{li73,jjain,ragu,chengkok}@ecn.purdue.edu

Abstract

We present fast algorithms for the DC and transient analyses of large-scale power grids. We reveal a compact Cholesky factorization for the coefficient matrix of the system of linear equations encountered in power grid analysis problems. By exploiting this compact structure, we obtain techniques for fast matrix inversion and matrix-vector multiplication. This new method takes full advantage of the special structure of power grids. Experimental results show that our method is more than hundred times faster for DC analysis and around ten times faster for transient simulation compared to Hspice, with these gains continuing to increase with problem size.

1 Introduction

With increasing operating frequencies and decreasing noise margins, the design of reliable on-chip power delivery networks has emerged as a critical challenge. Accurate and efficient simulation of power grids has become a necessity, and improper simulation might yield chip designs that fail when implemented. Power grid analysis includes two sub-problems, DC analysis and transient analysis. DC analysis is used to find steady-state node voltages, required for IR-drop analysis [2]. Transient analysis is concerned with finding voltage waveforms, which is essential for capturing voltage fluctuations in P/G networks [11].

Power grids are traditionally described as large-scale linear systems. Both DC and transient analysis problems can be formulated as solving linear equations of the following form (see Section 2 for details):

$$Ax = b. \quad (1)$$

As device sizes continue to shrink, the models that capture the effects of P/G networks tend to involve large though

sparse and structured matrices A . Hence, traditionally employed solvers become impractical due to their extraordinary demands on memory and computation. General algorithms for solving linear equations such as LU decomposition do not take the advantage of the special structure of power meshes, making them impractical. A number of iterative methods have been proposed specifically for the analysis of power grids in literature. A multi-grid approach that exploits the grid structure by mapping the original system to a coarsened grid, and remapping back to the original grid by interpolation was described in [10]. A different approach was proposed in [4], based on preconditioned Krylov-subspace methods. Iterative algorithms based on node-by-node and row-by-row traversal were proposed for IR-drop analysis in [15]. Statistical techniques based on random walks [14] are very fast and can efficiently solve for a small number of node voltages without solving the entire network. However, the typical verification of a P/G circuit requires the analysis of the entire network, or at least a portion of the network. In such cases, random walk techniques are not very efficient as every node has to be simulated individually [8]. Iterative methods work very well on certain problems, however they sacrifice a certain degree of accuracy for a lower run time. In addition, the convergence of these methods depends on the circuit parameters and the simulation time may vary a lot for circuits of the same size. Hence, an efficient and numerically stable direct solver is highly desirable.

In this paper, we exploit the structure of power grid meshes to reveal a compact representation for the Cholesky factorization of A^{-1} (see (1)), and propose a fast direct linear equation solver. The compact structure of the Cholesky factors is used to develop algorithms for fast matrix inversions and fast matrix-vector multiplications. The computational requirements with this new algorithm are independent of the specifics of the underlying power grid circuit. This is in contrast with all the other state-of-art algorithms which solve (1) iteratively, whose convergence rates depend

simplicity of illustration, we will be considering that each block in the block tridiagonal matrix is of size $m \times m$. However, all the theorems and formulae, mentioned henceforth, are directly applicable to the more general case when diagonal blocks are of different dimensions and the off diagonal blocks are rectangular.

3 A Fast Power Grid Analyzer

We start by introducing the elegant compact structure of the inverse of block tridiagonal matrices in Lemma 3.1.

Lemma 3.1 ([9]) *Let $A \in \mathbb{R}^{nm \times nm}$ be a block-tridiagonal matrix with each block of size $m \times m$. Then there exist four sequences of matrices $\{P_i\}, \{Q_i\}, \{R_i\}, \{S_i\} \in \mathbb{R}^{m \times m}$ such that*

$$A^{-1} = \begin{pmatrix} P_1 Q_1 & P_1 Q_2 & \cdots & P_1 Q_n \\ R_2 S_1 & P_2 Q_2 & \cdots & P_2 Q_n \\ \vdots & \vdots & \ddots & \vdots \\ R_n S_1 & R_n S_2 & \cdots & P_n Q_n \end{pmatrix}. \quad (4)$$

Thus, for $j \geq i$, the (i, j) block entry of A^{-1} equals $P_i Q_j$, and for $i > j$, the (i, j) block entry of A^{-1} equals $R_i S_j$.

The Cholesky factorization of a symmetric positive definite matrix A computes an upper triangular matrix U such that $A = U^T U$. The matrix U is called a Cholesky factor of A , and is denoted $U = \text{chol}(A)$. We now demonstrate that Cholesky factorization of the inverse of a block tridiagonal matrix also has a compact representation.

We begin with a simple extension to Lemma 3.1.

Lemma 3.2 *Let $U_A \in \mathbb{R}^{nm \times nm}$ be an upper block bidiagonal matrix with each block of size $m \times m$. Then there exist two sequences of matrices $\{X_i\}, \{Y_i\} \in \mathbb{R}^{m \times m}$ such that*

$$U_A^{-1} = \begin{bmatrix} X_1 Y_1 & X_1 Y_2 & X_1 Y_3 & \cdots & X_1 Y_n \\ & X_2 Y_2 & X_2 Y_3 & \cdots & X_2 Y_n \\ & & X_3 Y_3 & \cdots & X_3 Y_n \\ & & & \ddots & \vdots \\ & & & & X_n Y_n \end{bmatrix}. \quad (5)$$

This follows from Lemma 3.1, and the fact that the inverse of an upper triangular matrix is also upper triangular.

Without loss of generality, X_1 can be chosen as any positive definite matrix, leaving only $2n - 1$ matrices of size $m \times m$ as independent parameters in (5). Therefore, while U_A^{-1} is dense, it can be compactly described by $(2n - 1)m^2$ independent parameters, consistent with the number of nonzero entries in U_A .

We then propose the following theorem.

Theorem 3.1 *Let $A \in \mathbb{R}^{nm \times nm}$ be a symmetric positive-definite block-tridiagonal matrix with each block of size $m \times m$. Then there exist two sequences of matrices $\{X_i\}, \{Y_i\} \in \mathbb{R}^{m \times m}$ such that $\text{chol}(A^{-1}) = U_A^{-1}$ is given by (5).*

Proof: Using straightforward manipulations, it can be shown that the block tridiagonal matrix A defined in (2) can be written as

$$A = U_A U_A^T,$$

where

$$U_A = \begin{bmatrix} U_1 & B_1 U_2^{-T} & & & \\ & U_2 & B_2 U_3^{-T} & & \\ & & \ddots & \ddots & \\ & & & U_{n-1} & B_{n-1} U_n^{-T} \\ & & & & U_n \end{bmatrix},$$

and where

$$\begin{aligned} U_i &= (\text{chol}(\Sigma_i^{-1}))^{-1}, \\ \Sigma_n &= A_n, \\ \Sigma_i &= A_i - B_i \Sigma_{i+1}^{-1} B_i^T, \quad i = n-1, \dots, 2, 1. \end{aligned} \quad (6)$$

As $A = U_A U_A^T$, $A^{-1} = U_A^{-T} U_A^{-1}$. Hence, $\text{chol}(A^{-1}) = U_A^{-1}$. Now, U_A is an upper block bidiagonal matrix. Therefore, an appeal to Lemma 3.2 completes the proof. \square

It should be noted that Theorem 3.1 is applicable to the case when diagonal blocks (A_i) are of different sizes and off-diagonal blocks (B_i) are rectangular. Σ_i and U_i can be calculated from (6) in the same fashion, with each Σ_i and U_i having the same size as corresponding A_i .

While the statement of Theorem 3.1 provides a theoretically elegant parameterization of a Cholesky factor of the inverse of a block-tridiagonal matrix, the computation of the underlying parameters $\{X_i\}$ and $\{Y_i\}$ is beset by numerical problems for even modest sized problems [5]. Indeed, for matrices with sizes as small as 1000, the computation becomes ill-conditioned. We next propose a different, stable, parameterization of the Cholesky factor via the use of new sequences $\{R_i = Y_{i-1}^{-1} Y_i\}_{i=1}^{n-1}$ and $\{D_i = X_i Y_i\}_{i=1}^n$. It will be demonstrated in Section 4 that the new parameterization offers dramatically improved numerical stability. A paper with details regarding the new parameterization and its numerical stability has been submitted elsewhere for consideration [12].

Theorem 3.2 *Let $A \in \mathbb{R}^{nm \times nm}$ be a symmetric positive-definite block-tridiagonal matrix with each block of size $m \times m$. Then there exist two sequences of matrices $\{R_i\}, \{D_i\} \in \mathbb{R}^{m \times m}$ such that*

$$\begin{aligned} \text{chol}(A^{-1}) &= U_A^{-1} \\ &= \begin{bmatrix} D_1 & D_1 R_1 & D_1 R_1 R_2 & \cdots & (D_1 R_1 \cdots R_{n-1}) \\ & D_2 & D_2 R_2 & \cdots & (D_2 R_2 \cdots R_{n-1}) \\ & & & \ddots & \vdots \\ & & & & D_{n-1} R_{n-1} \\ & & & & D_n \end{bmatrix}. \end{aligned} \quad (7)$$

Proof: Using Theorem 3.1, we know that $\text{chol}(A^{-1})$ is of the form shown in (5). Defining two new sequences $\{R_i = Y_{i-1}^{-1} Y_i\}_{i=1}^{n-1}$ and $\{D_i = X_i Y_i\}_{i=1}^n$ completes the proof. \square

3.1 Computing a compact Cholesky factorization

Theorems 3.1 and 3.2 asserted merely the existence of compact parameterizations. We now present a procedure for calculating the sequences $\{D_i, R_i\}$ referred to in Theorem 3.2, beginning with the matrix A given in (2). Let $U_A^{-1}(i)$ denote the i th block row of U_A^{-1} and $U_A(j)$ the j th block column of U_A , and consider the identity $U_A^{-1}U_A = I$.

From $U_A^{-1}(i)U_A(i) = D_iU_i = I$, we obtain $D_i = U_i^{-1} = \text{chol}(\Sigma_i^{-1})$.

From $U_A^{-1}(i)U_A(i+1) = 0$, we obtain $B_iU_{i+1}^{-T} + R_iU_{i+1} = 0$, or $R_i = -B_iU_{i+1}^{-T}U_{i+1}^{-1}$. Using (6), we obtain $R_i = -B_i\Sigma_{i+1}^{-1}$.

Therefore, the procedure for calculating $\{D_i, R_i\}$ is as follows:

Calculating $\text{chol}(A^{-1})$ from A

```

 $\Sigma_n = A_n;$ 
 $D_n = \text{chol}(\Sigma_n^{-1});$ 
for  $i = n - 1$  down to 1
     $R_i = -B_i\Sigma_{i+1}^{-1};$ 
     $\Sigma_i = A_i + R_iB_i^T;$ 
     $D_i = \text{chol}(\Sigma_i^{-1});$ 
end

```

3.2 Fast matrix-vector multiplication

We next consider the matrix-vector multiplication $A^{-1}b$, where $A^{-1} = U_A^{-T}U_A^{-1}$, with U_A given in (7). This can be accomplished in two steps: Calculate $y = U_A^{-1}b$ and $x = U_A^{-T}y$.

To compute $y = U_A^{-1}b$, we partition $y = \{y_i\}_{i=1}^n$ and $b = \{b_i\}_{i=1}^n$ such that $\{y_i\}$ and $\{b_i\}$ have m rows. Then

$$y_j = D_j b_j + D_j R_j b_{j+1} + \cdots + D_j R_j R_{j+1} \cdots R_{n-1} b_n$$

$$= D_j \underbrace{(b_j + R_j b_{j+1} + \cdots + R_j R_{j+1} \cdots R_{n-1} b_n)}_{t_j},$$

$$y_{j+1} = D_{j+1} b_{j+1} + D_{j+1} R_{j+1} b_{j+2} + \cdots + D_{j+1} R_{j+1} \cdots R_{n-1} b_n$$

$$= D_{j+1} \underbrace{(b_{j+1} + R_{j+1} b_{j+2} + \cdots + R_{j+1} R_{j+2} \cdots R_{n-1} b_n)}_{t_{j+1}}.$$

A closer look at the above formula reveals that we can recursively calculate sequence $\{t_j\}$:

$$t_n = b_n, \quad t_j = b_j + R_j t_{j+1}, \quad j = n-1, n-2, \dots, 2, 1.$$

Similar comments apply to the computation of $x = U_A^{-T}y$.

Using these observations, we propose the following procedure to compute $x = A^{-1}b$.

Calculating $y = U_A^{-1}b$

```

 $t_n = b_n; y_n = D_n t_n;$ 
for  $i = n - 1$  down to 1
     $t_i = b_i + R_i t_{i+1};$ 
     $y_i = D_i t_i;$ 
end

```

Calculating $x = U_A^{-T}y$

```

 $x_1 = D_n^T y_1;$ 
for  $i = 2$  to  $n$ 
     $x_i = R_{i-1}^T x_{i-1} + D_i^T y_i;$ 
end

```

This algorithm takes $O(nm^2)$ operations, as compared to $O(n^2m^2)$ operations required with standard multiplication.

3.3 CCF algorithm

Combining the fast compact Cholesky factorization and fast matrix-vector multiplication, we obtain the fast linear equation solver (Compact Cholesky Factorization algorithm).

Solving $Ax = b$

```

 $\Sigma = A_n; D_n = \text{chol}(\Sigma^{-1}); t_n = b_n; y_n = D_n t_n;$ 
for  $i = n - 1$  down to 1
     $R_i = -B_i \Sigma^{-1};$ 
     $t_i = b_i + R_i t_{i+1};$ 
     $\Sigma = A_i + R_i B_i^T;$ 
     $D_i = \text{chol}(\Sigma^{-1});$ 
     $y_i = D_i t_i;$ 
 $x_1 = D_n^T y_1;$ 
for  $i = 2$  to  $n$ 
     $x_i = R_{i-1}^T x_{i-1} + D_i^T y_i;$ 
end

```

3.4 Computational complexity

Let m be the block size for D_i and N be the total number of nodes in the grid. Calculation of each $\{D_i\}_{i=1}^N$ or $\{R_i\}_{i=1}^{N-1}$ takes $O(m^3)$ operations and multiplication of A^{-1} with a vector requires $O(Nm)$ operations. Therefore, the overall computation required by the CCF algorithm is $O(\frac{N}{m}m^3 + Nm) = O(m^2N)$.

4 Experimental Results

The proposed Compact Cholesky Factorization (CCF) algorithm has been implemented in C on an i686 Linux PC.

Hspice, Sparse LU, and Random Walk algorithm (RW) [14] were chosen as existing algorithms for comparison. Sparse LU (SLU) is the best known direct solution for solving $Ax = b$, when A is a sparse matrix. In SLU, matrices are stored in Sparse Row Format which results in dramatic computational savings. It has been implemented using UMFPACK [6]. RW is one of the best known iterative methods for solving power grids. To report the results with RW, we used the author's implementation, taken from [1]. In the experiments, we used the tolerance of 1×10^{-3} as the convergence criteria for RW.

We first present the results showing computational requirements for DC analysis of a power grid. Table 1 summarizes the findings. It can be seen that the proposed Compact Cholesky Factorization (CCF) method is more than hundred times faster than Hspice, and up to a few times faster than SLU and random walk. We were unable to run large circuit sizes with Hspice, owing to prohibitive memory and computational requirements; the corresponding entries in the table are marked “—”. “Single node” denotes the time for calculating value of a single node by using Random Walk. It is given in the parentheses in the RW column. As pointed out in [14], and is also seen in our experiments, RW is very fast for calculating voltage at one node (or a set of nodes).

We now present runtime results for RW with varying number of sources and different convergence criteria. A plot of the simulation time (for power grid with 32K nodes) has been shown in Fig. 3. Two separate cases with number of sources being 80 and 160 have been considered. Since CCF is a direct solver, its run time is independent of the number of sources and was 1.83 seconds for both cases. For each case, CCF converged within a relative error of 10^{-15} . Since RW is an iterative procedure, the run time increased with a better tolerance. Also, the run times for RW are different for different number of sources in the power grid. Hence, as pointed out in Section 1, and is also confirmed by our experiments, the simulation time and convergence for iterative methods may vary with circuit parameters. On the other hand, the performance of CCF is independent of circuit parameters.

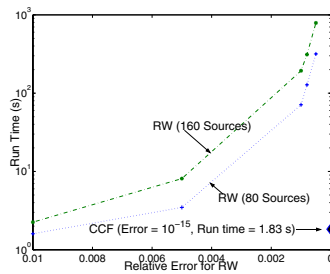


Figure 3. Run time for CCF and RW.

We next consider transient analysis, and present a comparison of accuracy and computational requirements of CCF with Hspice. A power grid model as shown in Fig. 1 was used for transient analysis. A plot of the voltage waveform at a randomly chosen node as obtained from Hspice is plotted along with that obtained by the CCF algorithm in Fig. 4. As the proposed CCF method is a direct algorithm, there is no detectable error in the simulation results. Table 2 shows the run time comparison for transient analysis. The Compact Cholesky Factorization (CCF) method is about 10 times faster than Hspice, with these gains continuing to increase with problem size. The accuracy of the results obtained also substantiates our claim regarding the improved numerical stability of $\{D_i, R_i\}$ sequences.

Fig. 5 and Fig. 6 summarize the run-time comparison for DC and transient analyses between Hspice and CCF.

Table 1. DC analysis Run Time (Sec).

nodes	Hspice	SLU	RW (1 node)	CCF
1K	0.34	0.01	12.07 (1.31)	0.00
5K	5.82	0.32	24.47 (1.29)	0.05
12K	94.34	0.79	51.37 (1.36)	0.14
32K	868.57	2.81	79.24 (1.41)	1.83
100K	—	12.03	102.26 (1.42)	7.84
1M	—	310.37	300.42 (1.50)	91.96

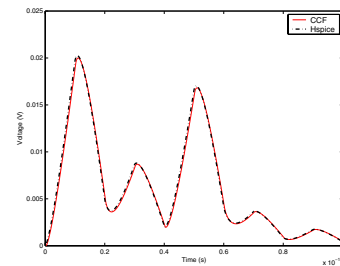


Figure 4. Transient simulation waveform.

Table 2. Transient Simulation Run Time (Sec).

nodes	Hspice	CCF	Speedup
1000	4.2	0.53	7.9
2000	8.75	1.17	7.5
8000	97.85	12.87	7.6
10000	165.03	20.43	8.1
32000	1645.28	93.51	17.59

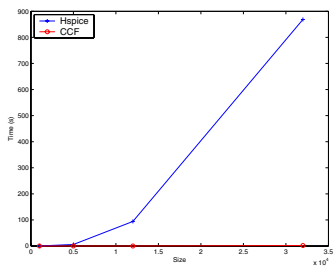


Figure 5. DC analysis run time comparison.

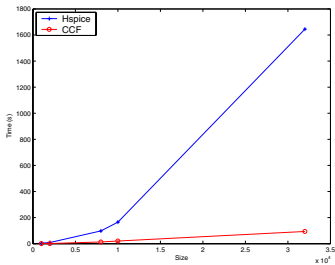


Figure 6. AC analysis run time comparison.

5 Conclusion and future work

We have introduced an efficient technique to perform DC and transient analyses of large-scale power grids based on a compact and fast Cholesky factorization. We have not considered inductance effects in this paper; with increasing operating frequencies, inductance effects may be significant enough to warrant more general RLC network models for power grids. It has been observed that with most circuits, while the inductance matrix L is dense, its inverse typically has a banded structure (see [7] for example). The analysis of power grids with a consideration of inductive couplings is a subject of ongoing investigation.

References

- [1] <http://www.ece.unm.edu/users/qianhf/enversion/enindex.htm>.
- [2] A. H. Ajami, K. Banerjee, A. Mehrotra, and M. Pedram. Analysis of IR-drop scaling with implications for deep submicron P/G network designs. In *International Symposium on Quality Electronic Design*, pages 35–40, 2003.
- [3] U. M. Ascher and L. R. Petzold. Computer methods for ordinary differential equations and differential-algebraic equations. In *SIAM*, 1998.
- [4] T. Chen and C. C. Chen. Efficient large-scale power grid analysis based on preconditioned Krylov-subspace iterative methods. In *Proc. Design Automation Conf.*, pages 559–562, 2001.
- [5] P. Concus and G. Meurant. On computing inv block preconditionings for the conjugate gradient method. *BIT*, (26):493–504, 1986.
- [6] T. A. Davis. UMFPACK version 4.4 user guide. <http://www.cise.ufl.edu/research/sparse/umfpack/>.
- [7] A. Devgan, H. Ji, and W. Dai. How to efficiently capture on-chip inductance effects: Introducing a new circuit element K. In *Proc. Int. Conf. on Computer Aided Design*, pages 150–155, 2000.
- [8] Weikun Guo, Sheldon X. -D. Tan, Zuying Luo, and Xianglong Hang. Partial random walks for transient analysis of large power distribution networks. *IEICE Trans. Electron.*, (12).
- [9] Y. Ikebe. on inverses of Hessenberg matrices. *Linear algebra application*, (24):93–97, 1979.
- [10] J. N. Kozhaya, S. R. Nassif, and F. N. Najm. A multigrid-like technique for power grid analysis. *IEEE Trans. Computer-aided Design*, 21:1148–1160, 2002.
- [11] Y. M and C. C. Chen. the power grid transient simulation in linear time based on 3D alternating-direction-implicit method. In *Proc. Design, Automation and Test in Europe*, pages 1020–1025, 2003.
- [12] Author's name have been omitted for blind review. Numerically stable algorithms for inversion of block tridiagonal and banded matrices, submitted for consideration. *Numerical Linear Algebra Appl.*, 2006.
- [13] H. Qian, S. R. Nassif, and S. S. Sapatnekar. Random walk in a supply network. In *Proc. Design Automation Conf.*, pages 93–98, 2003.
- [14] Haifeng Qian, Sani R. Nassif, and Sachin S. Sapatnekar. Power grid analysis using random walks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(8):1204–1224, 2005.
- [15] Y. Zhong and M. D. F. Wong. Fast algorithms for ir drop analysis in large power grid. In *Proc. Int. Conf. on Computer Aided Design*, pages 351–357, 2005.
- [16] N. Zhu and H. Y. Koh. Power grid modeling technique for hierarchical network analysis. In *International Symposium on Quality Electronic Design*, pages 312–318, 2001.