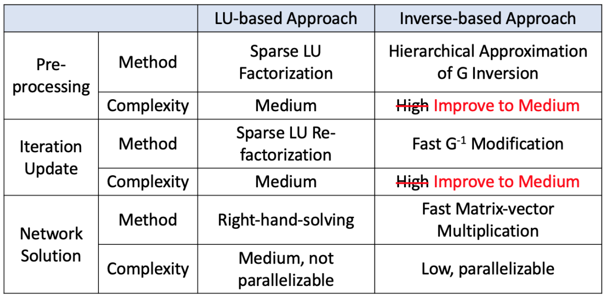
An Efficient Network Solver for Electromagnetic Transient Simulation of Power Systems Based on Hierarchical Inverse Computation and Modification

Lu Zhang, Bin Wang, Vivek Sarin, Weiping Shi, P. R. Kumar, and Le Xie
Texas A&M University
College Station, TX, 77843, USA

2021 North American Power Symposium (NAPS) | 978-1-6654-2081-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/NAPS52732.2021.9654532

***Abstract*—In both power system transient stability and electromagnetic transient (EMT) simulations, up to 90% of the computational time is devoted to solve the network equations, i.e., a set of linear equations. Traditional approaches are based on sparse LU factorization, which is inherently sequential. In this paper, EMT simulation is considered and an inverse-based network solution is proposed by a hierarchical method for computing and store the approximate inverse of the conductance matrix. The proposed method can also efficiently update the inverse by modifying only local sub-matrices to reflect changes in the network, e.g., loss of a line. Experiments on a series of simplified 179-bus Western Interconnection demonstrate the advantages of the proposed methods.**

***Index Terms*—inverse-based network solver, Electromagnetic transient (EMT) simulation, hierarchical approximation.**



		LU-based Approach	Inverse-based Approach
Pre-processing	Method	Sparse LU Factorization	Hierarchical Approximation of G Inversion
	Complexity	Medium	~~High~~ Improve to Medium
Iteration Update	Method	Sparse LU Re-factorization	Fast G^{-1} Modification
	Complexity	Medium	~~High~~ Improve to Medium
Network Solution	Method	Right-hand-solving	Fast Matrix-vector Multiplication
	Complexity	Medium, not parallelizable	Low, parallelizable

Fig. 1: Comparison of LU-based and inverse-based approaches, where red texts are contributions of this paper.

I. Introduction

Transient stability (TS) and electromagnetic transient (EMT) simulations has been utilized in many critical applications in modern power industry, e.g., system dynamic security assessment, device insulation design and validation of control strategy [1][2]. In practice, simulation speed is always a key aspect of simulation tools, especially for real-time applications and large-scale systems [3]. As the penetration level of inverter-based generations, such as wind and solar, continues to increase in the upcoming decades, the complexity and dimension of the underlying systems will drastically increase along with more dynamic risks, e.g., sub-synchronous oscillations. This poses a challenge to the inevitable need for fast simulations of large-scale systems [4].

Two major types of dynamic simulation in power industry are TS simulation, and EMT simulation. In both types, solving the network equations, i.e., a set of linear equations, is the most time-consuming step, taking up to 90% of the overall computational time [5][6][7]. The most popular approach is based on LU factorization [8][9], which is adopted by many commercial simulation tools. However, this approach is sequential in nature, making it difficult to be applied to parallel computing platforms.

In contrast, the inverse-based approach exhibits great potential in speeding up the network solution by parallel matrix-vector multiplication. Fig. 1 compares the LU-based and inverse-based approaches where black texts are our previous effort [10], [11]. In [10][11], we explored a fast direct inverse-based network solver, where the solving step outperforms the LU-based approach by $2.8\times$ for large networks ($>$1000 buses) in runtime, even when sequentially computed. However, it requires significant time and memory to calculate and store the inverse of the large conductance matrix G.

To further develop the technique of parallelizable network solvers, this paper proposes a fast and memory-efficient hierarchical approximation method for calculating and store the inverse of the large admittance/conductance matrices established from power grids. Different from traditional Diakoptic [12] and conventional hierarchical approaches [13][14], the proposed method decomposes the entire network into several sub-networks, then implicitly computes matrix inverse by using the inversions of sub-matrices hierarchically. A fast modification approach is further presented based on the inverted matrix by modifying only local entries and sub-matrices to reflect changes in network, e.g., loss of a line. The advantage of the proposed method is demonstrated only on EMT simulation in this paper, while similar results are expected for TS simulation.

The key contributions of this paper are as follows. 1) An efficient hierarchical method for calculating and store G^{-1}, which can be used for the inverse based solution of linear equations involved in large-scale power system TS and EMT simulations; 2) A fast and memory-efficient G^{-1} modification approach to handle scenarios where G changes, which may be used for simulating fault and topology changes.

Preferred email for correspondence: Le Xie, le.xie@tamu.edu.

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.

II. Brief Review of EMT Simulation

In EMT simulation, the network equations are a set of linear equations, as shown in (1), which are derived from the nodal formulation and discretized by the numerical integration [15].

$$G\mathbf{v}(t) = \mathbf{i}_{\text{in}}(t) + \mathbf{i}_{\text{his}}(t - \Delta t), \tag{1}$$

where G is the conductance matrix, $\mathbf{v}(t)$ is the vector of nodal voltages, $\mathbf{i}_{\text{in}}(t)$ is the vector of current injections, and $\mathbf{i}_{\text{his}}(t - \Delta t)$ is the vector of historical currents, and all values are real.

The traditional solution to (1) is LU-based which requires sequential forward backward substitute in the solving stage. This paper focuses on a fully parallelizable inverse-based approach which rewrites (1) into a matrix-vector multiplication:

$$\mathbf{v}(t) = G^{-1}\mathbf{i}(t), \mathbf{i}(t) = \mathbf{i}_{\text{in}}(t) + \mathbf{i}_{\text{his}}(t - \Delta t). \tag{2}$$

Our past efforts [11] shows that, once G^{-1} is calculated, the network solution time can be reduced from O(N^2) to O($N \log N$), where N is the system size, even with sequential computation, and parallelizable schemes may be employed to further speed up the simulation.

III. Hierarchical Approximation of G Inverse

Unlike G, G^{-1} cannot be formed directly from the network. Even though G^{-1} can be found via inversion of G, explicit inversion is very expensive. Thus, this section introduces a hierarchical approximation approach for computing G^{-1} with much lower time and storage requirements.

A. Approximation of G Inverse

Assuming G and G^{-1} are symmetric. Express G and G^{-1} in a block matrix format according to a partition of network G into sub-network G_1 and G_2,

$$G = \begin{bmatrix} G_1 & H \\ H^T & G_2 \end{bmatrix} \text{ and } G^{-1} = \begin{bmatrix} A & M \\ M^T & B \end{bmatrix}, \tag{3}$$

where the sizes of G_1 is $m \times m$, G_2 is $n \times n$, H is $m \times n$, and $m/3$, $n/3$ are the number of buses represented by G_1 and G_2, and each bus has three phases.

To compute the approximate G^{-1}, named as $\widetilde{G}^{-1}$, we want to find a matrix making the off-diagonals of $G\widetilde{G}^{-1}$ equal to 0, and the diagonals close to the identity matrix. Therefore, we wish to restrict sub-matrices A and B as follows,

$$\widetilde{G}^{-1} = \begin{bmatrix} G_1^{-1} & M \\ M^T & G_2^{-1} \end{bmatrix}. \tag{4}$$

We then find M and M^T.

$$\begin{aligned} G \times \widetilde{G}^{-1} &= \begin{bmatrix} G_1 & H \\ H^T & G_2 \end{bmatrix} \begin{bmatrix} G_1^{-1} & M \\ M^T & G_2^{-1} \end{bmatrix} \\ &= \begin{bmatrix} I + MH^T & G_1^{-1}H + MG_2 \\ G_2^{-1}H^T + M^T G_1 & I + M^T H \end{bmatrix}. \end{aligned} \tag{5}$$

Forcing the off-diagonals to be 0, we get

$$\begin{cases} G_1^{-1}H + MG_2 = 0, \\ G_2^{-1}H^T + M^T G_1 = 0. \end{cases} \tag{6}$$

Thus,

$$M = -G_1^{-1}HG_2^{-1}. \tag{7}$$

Plug (7) into (4),

$$\widetilde{G}^{-1} = \begin{bmatrix} G_1^{-1} & -G_1^{-1}HG_2^{-1} \\ -G_2^{-1}H^T G_1^{-1} & G_2^{-1} \end{bmatrix}. \tag{8}$$

Note that the approximation inverse approach works for both real and complex matrices, while in this paper, real matrix is considered since the conductance matrix involved in nodal formulation based EMT simulation is real-valued.

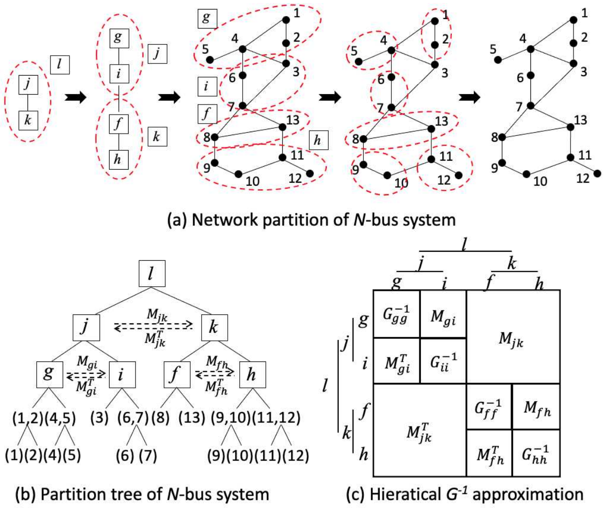
In power systems, matrix $H_{m\times n}$ in G is determined by the network, i.e., the transmission lines connecting two groups of buses s_1 and s_2, where s_1 contains m buses, and s_2 contains n buses. Suppose there are k transmission lines between s_1 and s_2, then $H = h_1 h_2^T$, where h_1 is an $m \times k$ matrix representing the interaction from s_1 to s_2, and h_2 is an $n \times k$ matrix representing the interaction from s_2 to s_1. In general, k is much less than m and n. We can rewrite M as follows.

$$M = -(G_1^{-1}h_1)(G_2^{-1}h_2)^T. \tag{9}$$

If we multiply the matrices in the order given by (9) instead of (7), the computation cost will be reduced to $km^2 + kn^2 + kmn$. In addition, minimizing k will further reduce the time.

B. Hierarchical Approximation of G Inverse

To further reduce the time to compute G^{-1}, we propose the hierarchical approximation, which is illustrated in Fig. 2.



(a) Network partition of *N*-bus system

(b) Partition tree of *N*-bus system

(c) Hieratical G^{-1} approximation

Fig. 2: Hierarchical approximation for calculating G^{-1}.

The process is as described. First, we recursively partition the network into multiple sub-networks. In Fig. 2 (a), network l is first bi-partitioned into sub-networks j and k, then j and k are further bi-partitioned into sub-network g, i, and f, h, respectively, so on so forth until each sub-network contains only one bus. The partition thereby builds a binary tree where the root represents the whole network, the internal node represents a group of buses, and the leaf represents an individual bus, shown as Fig. 2 (b). Correspondingly, the network partition divides G^{-1} into a hierarchy, where each

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.

sub-matrix, including G^{-1} itself, is further divided into 4 sub-matrices, until each sub-matrix is a single element. In Fig. 2 (c), group j consists of 7 buses and group k consists of 6 buses, thus the interaction within j is expressed by a 7×7 sub-matrix G_{jj}^{-1}, where the row and column indices correspond to buses in j; the interaction within k is expressed by a 6×6 sub-matrix G_{kk}^{-1}, where the row and column indices correspond to buses in k, and the interaction between j and k is expressed by a 7×6 sub-matrix M_{jk}, where the row indices correspond to buses in j, and the column indices correspond to buses in k. Based on the hierarchically partitioned G^{-1}, we apply the approximate inverse approach recursively, eventually construct the entire G^{-1} without any direct matrix inversion, i.e., G_{jj}^{-1} is constructed by G_{gg}^{-1}, G_{ii}^{-1} and M_{gi}, and M_{gi} is calculated by (9), where G_{gg}^{-1} is a 6×6 matrix, G_{ii}^{-1} is a 3×3 matrix, h_{gi} is a 6×3 matrix, and h_{ig} is a 3×3 matrix.

To improve the accuracy of the approximation, we pre-define a node threshold d_{th} to limit the size of the minimum group of buses. If a sub-network contains fewer than d_{th} buses, the partition stops. For the example in Fig. 3, we set d_{th} as 4, thus node g, i, f, and h are the minimal sub-network where the partition stops. To achieve high approximation accuracy and low computation cost, the network partition algorithm needs to minimize the edges between two sub-networks. In this paper, we adopted the partition algorithm used in [11].

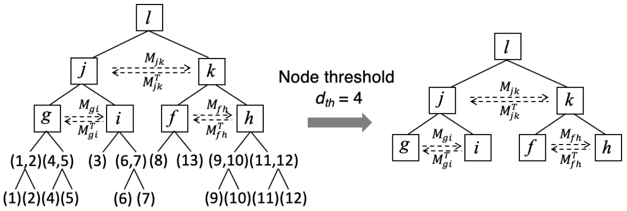


Fig. 3: Partition tree of G^{-1} by node threshold d_{th} = 4.

The algorithm is presented below. The input is the root of the partition tree i, with its left child being l and right child being r. Each tree node stores the bus indices of the sub-network. The outputs are M_{lr} and the approximation of G_{ii}^{-1}.

```
Algorithm 1: Hierarchical_Ginv($i$)
if $i$ has less than $d_{th}$ buses then
    direct compute the $G_{ii}^{-1}$;
else
    $G_l^{-1}$ = Hierarchical_Ginv($l$);
    $G_r^{-1}$ = Hierarchical_Ginv($r$);
    $H = G_{lr}$; Decompose $H = h_l * h_r^T$, where $h_l$ is an
      $l \times k$ matrix, $h_r$ is an $r \times k$ matrix, $k$ is the
      number of edges between $l$ and $r$;
    $M_{lr} = -(G_l^{-1} h_l)(G_r^{-1} h_r)^T$,
      $G_{ii}^{-1} = \begin{bmatrix} G_l^{-1} & M_{lr} \\ M_{lr}^T & G_r^{-1} \end{bmatrix}$;
end
return $M_{lr}, G_{ii}^{-1}$
```

The benefits of this approach are 1) Save computation of matrix inversion without direct invert G. In the algorithm, we only need to direct invert the diagonal matrices of each size bounded by node threshold d_{th}; 2) Reduce storage requirement. Instead of storing the entire dense matrix G^{-1}, this approach only requires to store diagonal matrices.

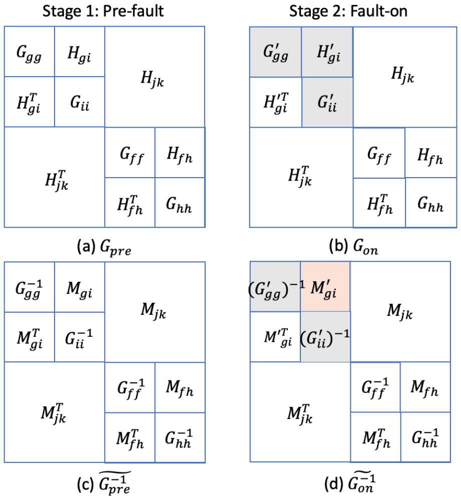
IV. Fast G Inverse Modification

This section introduces a fast G^{-1} modification approach based on Algorithm 1 to handle network changes.

A. *Modification Algorithm*

Take the network in Section III.B as an example. Suppose a fault occurs between bus 4 and bus 6. Denote G_{pre}, G_{on} as conductance matrices in the pre-fault stage and fault-on stage, respectively. The construction of G_{pre}, G_{on}, G_{pre}^{-1} and G_{on}^{-1} is illustrated in Fig. 4.

In the pre-fault stage, G_{pre} is constructed according to the network before the fault occurs, shown in Fig. 4 (a). We hierarchically approximate $\underset{\sim}{G_{pre}^{-1}}$ by Algorithm 1. Name the approximation of G_{pre}^{-1} as $\widetilde{G_{pre}^{-1}}$, shown in Fig. 4 (c).



(a) G_{pre}

(b) G_{on}

(c) $\widetilde{G_{pre}^{-1}}$

(d) $\widetilde{G_{on}^{-1}}$

Fig. 4: Modification of G^{-1} under a fault between g and i.

In the fault-on stage, G_{pre} is updated to G_{on} by modifying corresponding elements in G_{gg}, G_{ii}, and H_{gi} due to the fault occurred between bus 4 and bus 6. Rename the modified block matrices as $G'_{gg}, G'_{ii}, H'_{gi}$, shown in the Fig. 4 (b). Since only three blocks are modified, while others stay the same. According to Algorithm 1, $\widetilde{G_{on}^{-1}}$ is formed by modifying $(G'_{gg})^{-1}, (G'_{ii})^{-1}$, and M'_{gi} only, shown in Fig. 4 (d).

As described, the modification of G^{-1} updates a relative small portion of sub-blocks and avoids re-construct the entire G^{-1}. The blocks which need to be updated are easily located by the partition tree. In this example, bus 4 belongs to node g, and bus 6 belongs to node i. Starting from node g and i, we trace up to their ancestors until reaching their lowest common ancestor (LCA) which is node j. Thus node j is the smallest group that contains both bus 4 and 6, and the smallest group that is directly influenced by the fault. Therefore, the interaction within node j needs to be modified, thus all matrices that describe interactions between nodes along the

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.

paths from g to j and from i to j need to be modified. Since node g and i are leaves, the interaction within node g and i which represented as $(G_{gg}^{'})^{-1}$ and $(G_{ii}^{'})^{-1}$ are computed directly by inversion. The interaction between node g and i represented as $M_{gi}^{'}$ is computed by (7). The interaction within node j is approximated by interaction within node g, node i and between node g and i by (4). The tree with modification nodes is shown in Fig. 5, where the grey nodes are updated by directly inverse, the orange node is the LCA of grey nodes estimated by combining grey nodes.

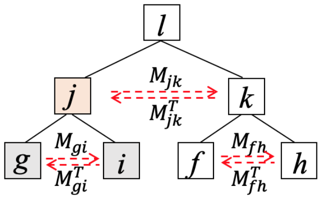


Fig. 5: The updated tree of G^{-1} under a fault.

The detailed algorithm is presented below. Suppose the network is partitioned and G^{-1} is approximated by Algorithm 1 at beforehand. The inputs are node i and j, where node i contains the fault bus i, node j contains the fault bus j.

Algorithm 2: Modification(i, j)

```
Find the LCA of node $i$ and $j$, name it as node $k$;
UpdateM($k$, $i$, $j$);
return
Function UpdateM($k, i, j$):
    if $k$ is a leaf in the partition tree then
        if $k$ is the same as $i$ or $j$ then
            direct compute the $G_{kk}^{-1}$;
        end
        return
    end
    UpdateM(left child of $k$, $i$, $j$);
    UpdateM(right child of $k$, $i$, $j$);
    Update $M_{lr}^{-1}$ by (9), where $l$ and $r$ are the left and
      right child of $k$, respectively;
    return;
```

The computation cost of the fast G^{-1} modification approach highly depends on the fault location. In the best case, the LCA of node i and j is itself, so that we don't need to update any M matrices, just update the diagonal matrix locally. In the worst case, the LCA of node i and j is the root, so that besides updating the diagonal matrix locally, we also need to update multiple M matrices from leaf to root.

B. Time Complexity

Theorem 1. *Given an N-bus power system which is hierarchically partitioned, the time complexity of fast G^{-1} modification algorithm based on hierarchical approximation for G inversion is $O(n^2)$, where n is the smallest group of buses which contain buses directly connected to the fault.*

Proof. Suppose the N-bus power system network is partitioned into multiple groups, thereby produces a binary tree where the root represents the whole network, the leaves represents small groups of buses where each group contains no more than d_{th} buses. Therefore, the partition tree has $O(N)$ nodes and the height of the tree is $O(\log N)$.

Consider when a fault occurs between bus i and j, which belongs to leaves i and j in the partition tree, respectively. Let node k be the lowest common ancestor of leaves i and j and let the number of buses under node k be n. According to Algorithm 2, the time complexity of fast G^{-1} modification algorithm only depends on n. Let $T(n)$ be the time complexity of fast G^{-1} modification algorithm under a fault, then we have

$$T(n) \leq \begin{cases} C & \text{if } n \leq d_{th} \\ 2T(\frac{n}{2}) + rn^2 & \text{otherwise,} \end{cases}$$

where C is a constant, $T\left(\frac{n}{2}\right)$ is the time complexity of updating diagonal matrices due to the partition, and rn^2 is the upper bound on time complexity of updating the off-diagonal matrices M, r is the number of edges between two groups.

To solve the recurrence relation, we have

$$\begin{aligned} T(n) &\leq 2T\left(\frac{n}{2}\right) + rn^2 \\ &= 2\left(2T\left(\frac{n}{4}\right) + r\left(\frac{n}{2}\right)^2\right) + rn^2 = \cdots \\ &= \frac{n}{d_{th}}T(d_{th}) + rn^2 \sum_{i=0}^{\log(n/d_{th})} \frac{1}{2^i} \\ &< \frac{n}{d_{th}} + 2rn^2 = O(n^2). \end{aligned}$$

The sum to $\log n/d_{th}$ is due to the fact that the length of the path from node i to k, and j to k is at most $\log n/d_{th}$. □

V. Case Study

This section presents comprehensive case studies for EMT simulation on 179-bus based systems to evaluate the performance of the proposed approaches. The data and one-line diagrams of the 179-bus system can be found in [16]. In the simulation, Algorithm 1 is used to approximate G^{-1}. If a fault occurs, Algorithm 2 is used to update G^{-1}. For the network solution, the low-rank approximation based approach is adopted to further compress G^{-1}, then fast matrix-vector multiplication [11] is applied to solve the network equations.

A. Node threshold v.s. G^{-1} Accuracy

In this test, G matrix for a 179-bus system is used for evaluating the impact of the node threshold d_{th} on the accuracy of G^{-1}. Fig. 6 shows that as d_{th} increases from 2 to 74, the error exponentially drops from 2.5×10^{-3} to 1.8×10^{-9}. However, as d_{th} increases, the computation cost increases. To balance the trade-off between error and computation cost, d_{th} is set to 74 in our test cases.

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.

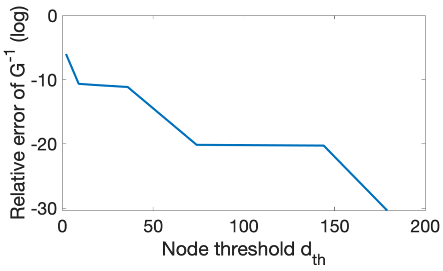


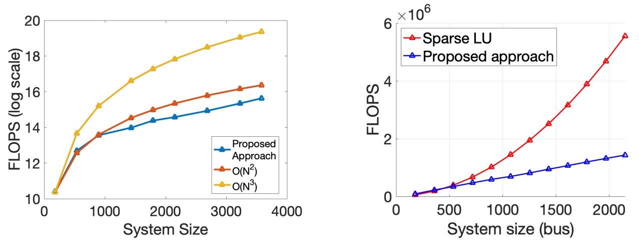
Fig. 6: Relative error of G^{-1} with node threshold d_{th}.

B. CPU time in terms of FLOPS

In this test, a series of large systems consist of multiple copies of a 179-bus system interconnected as an array. A similar system is used in [17]. The FLOPS involved in approximate G^{-1} by Algorithm 1 and the network solution by fast matrix-vector multiplication [11] at one time step are counted. The FLOPS count includes total numbers of floating point addition and multiplication.

1) FLOPS for hierarchical approximation of G inverse: Traditionally, the time complexity of matrix inverse is O(N^3), where N is the system size. Fig. 7(a) shows that in the series of 179-bus based systems, the FLOPS of Algorithm 1 is O(N^2).

2) FLOPS for network solution: Fig. 7(b) shows that in the series of 179-bus based systems with up to 2148 buses, the maximum FLOPS of LU-based approach is 5.56×10^6, while the fast matrix-vector multiplication [11] is 1.43×10^6, leading to an 74% reduction compared with LU-based approach.



(a) Flops for G^{-1} computing

(b) Flops for network solution

Fig. 7: FLOPS comparison.

C. EMT fault simulation

In this test, a three-phase ground fault on a 179-bus system is simulated with a 20μs time step and a total simulation time length of 60ms. The fault between bus 1 and bus 81 is added at 10ms and lasts for 20ms. The fault resistance is 10 Ω. The partial network with the fault is shown in Fig. 8.

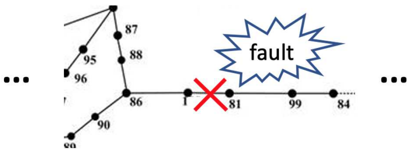
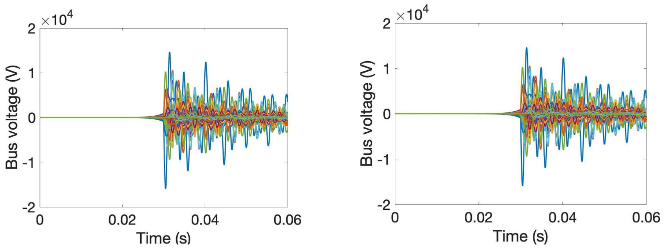


Fig. 8: The 179-bus system with a fault.

1) Accuracy: An LU-based network solver is used as a reference. The relative error of the bus voltage is measured at each simulation step. As shown in Fig. 9 and 10, the result by the proposed approach matches well with the reference. The maximal relative error of bus voltage is 7.4×10^{-5}.



(a) LU-based Approach

(b) Proposed approach

Fig. 9: Accuracy comparison.

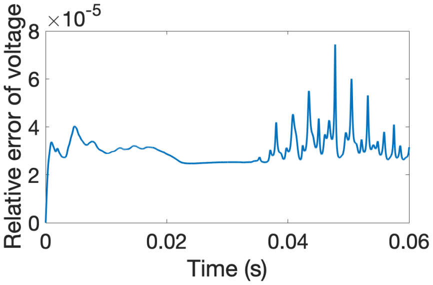


Fig. 10: Relative error of bus voltage on 179-bus system.

2) Computation cost for G inverse modification: The cost of fast G^{-1} modification approach depends on the fault location. In this case, the fault occurs in the same leaf node of the partition tree. Since the node threshold is 74, the proposed approach only needs to invert the matrix with 74×74 elements instead of 179×179 elements, leading to 82.9% reduction.

VI. Concluding Remarks

This paper presents an inverse-based scheme to solve the power network algebraic equations. To break down the bottleneck of computing matrix inversion, this paper proposes a hierarchical approximation of G^{-1}. Further, a fast modification approach is presented for facilitating the contingency analysis. The performance is demonstrated by the EMT simulation on large-scale systems created from a 179-bus base case. The experiments show that the proposed approaches exhibit great advantage in speeding up the network solution with high accuracy. Future work will investigate the scalability of this approach in larger realistic systems.

Acknowledgment

This material is based upon work partially supported by US National Science Foundation under IIS-1636772, U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number DE-EE0009031.

The views expressed herein and conclusions contained in this document are those of the authors and should not be

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.

interpreted as representing the views or official policies, either expressed or implied, of the U.S. NSF, Department of Energy or the United States Government.

The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] F. Milano, *Power System Modelling and Scripting*. Springer, 2010.

[2] A. Ametani, *Numerical Analysis of Power System Transients and Dynamics*. IET, 2014.

[3] K. Abdul-Rahman *et al.*, "Application of high performance computing in California ISO's online DSA system," in *IEEE PES General Meeting*, 2017, pp. 1–7.

[4] B. Kroposki, "Summarizing the technical challenges of high levels of inverter-based resources in power grids," 2019.

[5] L. Gérin-Lajoie and J. Mahseredjian, "Simulation of an extra large network in EMTP: From electromagnetic to electromechanical transients," *Int. Conf. Power Syst. Transients*, Jun. 2019.

[6] Y. Chen *et al.*, "GPU-based techniques of parallel electromagnetic transient simulation for large-scale distribution network," *Automation of Electric Power Systems*, vol. 41, no. 19, pp. 82–88, Oct. 2017.

[7] B. Wang, N. Duan, and K. Sun, "A time–power series-based semi-analytical approach for power system simulation," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 841–851, 2019.

[8] T. Davis, "Summary of available software for sparse direct methods," 01 2009.

[9] T. Davis and E. Natarajan, "(2010) algorithm 907: KLU, a direct sparse solver for circuit simulations problems," *ACM Transactions on Mathematical Software*, vol. 37, no. 3, p. 36:1–36:17.

[10] L. Zhang *et al.*, "Fast electromagnetic transient simulation based on hierarchical low-rank approximation," in *IEEE ISGT*, 2019, pp. 1–5.

[11] L. Zhang *et al.*, "A hierarchical low-rank approximation based network solver for emt simulation," *IEEE Transactions on Power Delivery*, pp. 1–1, 2020.

[12] G. Kron, *Diakoptics: The Piecewise Solution of Large-Scale Systems*. Schenectady, N.Y., General Electric, 1957-58.

[13] S. Börm *et al.*, "Introduction to hierarchical matrices with applications," *Engineering Analysis with Boundary Elements*, vol. 27, no. 5, pp. 405 – 422, 2003.

[14] X. Xiong and J. Wang, "A hierarchical matrix inversion algorithm for vectorless power grid verification," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 543–550.

[15] H. W. Dommel, *EMTP Theory Book*. BPA, 1981.

[16] S. Maslennikov *et al.*, "A test cases library for methods locating the sources of sustained oscillations," in *IEEE PES General Meeting*, Boston, 2016, pp. 1–5.

[17] Y. Song *et al.*, "Fully GPU-based electromagnetic transient simulation considering large-scale control systems for system-level studies," *IET Generation, Transmission Distribution*, vol. 11, no. 11, pp. 2840–2851, 2017.

Authorized licensed use limited to: Texas A M University. Downloaded on November 20,2022 at 00:40:33 UTC from IEEE Xplore. Restrictions apply.