Research on the performance of multifrontal method in power system simulations

ZHAO Wen-Kai¹*, GONG Wei-Zheng², XIE Bang-Peng¹, WAN Jia-Lin¹

¹ Zhangjiang Science City Energy Service Center, Pudong Power Supply Company of Shanghai Municipal Electric Power Company of State Grid, Shanghai 200122, China
² Electric Power Dispatching & Control Center, East Subsection of State Grid Corporation of China, Shanghai 200120, China

* Corresponding author. E-mail address: zwk601@163.com (W. Zhao)

Abstract: Efficiency of power system simulation has close relationship to the sparse LU factorization algorithm. Multifrontal method is a notable LU factorization member. This paper investigates the performance of multifrontal method in power system simulations. Performance of multifrontal method heavily relies on the frontal matrix size. Frontal matrix size can be represented by front size. Front size is subject to the treewidth of power network. A treewidth approximation method (TAM) is presented to approximate the treewidth of power network. Two derivatives of TAM are given to tackle the sparse matrices from power flow calculation and small signal stability eigenvalue analysis respectively. Simulation results verify the accuracy and efficiency of TAM and its derivatives. Further results indicate that most power networks have low treewidth, which is prone to degrading the performance of multifrontal method. Multifrontal method is able to gain its performance in power system simulations when the front size of the involved matrix is large enough.

1. Introduction

The solution of sparse linear system is one of the most computationally intensive parts in large-scale power system simulations. Since applications of online monitoring system require high computing speed, efficiency of sparse computations is vital to large-scale power system simulations.

Direct methods, with natures of generality and robustness, are widely used in scientific computations. Since the 1960s, many sparse techniques considering special structure of power network have been developed to improve the computational efficiency. Optimally ordered triangular factorization [1] and sparse vector method [2] are two remarkable techniques used for power flow calculation and transient stability simulation respectively. Appearance of more delicate variants contributes to the prosperity of sparse vector method in the field of power system simulations in the 1990s [3] [4]. After that, LU factorization methods in linear sparse solvers have gradually evolved into two categories [5]. One is the left-looking LU factorization, and the other is the right-looking LU factorization.

The multifrontal method, a variant of right-looking LU, was first elaborated in [6]. With the utilization of high level BLAS (Basic Linear Algebra Subprograms) [7], multifrontal method is able to
achieve prominent performance in some cases. Meanwhile, it is yet known that for very sparse problems with small frontal matrix sizes, multifrontal method would be rendered inefficient if high level BLAS cannot be fully utilized [8]. Comparisons of different factorization methods have been made with matrices arising from different application backgrounds [9] [10]. Results turn out that the efficiency of LU factorization method is much related to the application environment.

In recent years, multifrontal method has been brought into power system simulations [11]-[13]. Reference [11] employs the unsymmetric multifrontal method to solve the differential algebraic equations in power system time-domain simulation. Results show that efficiency gain of multifrontal method against Gaussian elimination methods and other sparse solvers can be orders of magnitude larger. In [12], the multifrontal preconditioned iterative methods are proposed for power system time-domain simulation, which achieves great computational efficiency compared to other iterative and direct methods. A parallel power flow calculation using multifrontal method with graphic processing unit is presented in [13], where the performance improvement is very limited due to large amounts of small dense matrices encountered during factorization.

Facing the opposite views on the performance of multifrontal method, we present this paper to evaluate the applicability of multifrontal method in power system simulations. In order to realize this target, this paper begins with statistical analysis, which shows that frontal matrix size is decisive to the performance of multifrontal method. A well-designed and reasonable experiment indicates that matrix sparsity pattern is an underlying factor for different frontal matrix sizes.

Generally, there are two types of sparsity structures in power system simulations, one is the network structure associated with the admittance matrix in static simulations, and the other is the sparsity structure associated with the coefficient matrix in dynamic simulations. First of all, the pattern of admittance matrix is analyzed. After front size and treewidth are introduced to describe the matrix sparsity pattern, the performance of multifrontal method shall be predicted by the treewidth of power network. A treewidth approximation method (TAM) is proposed to approximate the treewidth of power network efficiently. The mathematical justification of TAM is also given in the appendix. Then, matrix patterns from power flow calculation and small signal stability analysis are analyzed as representatives of static and dynamic simulations respectively. Two derivatives of TAM are further presented to illustrate how to evaluate the performance of multifrontal method ahead of numeric computations in both static and dynamic power system simulations.

The remaining sections are organized as follows. Section 2 gives an overview and efficiency analysis of the multifrontal method. In Section 3, the relationship between front size and treewidth is established. Section 4 presents TAM and its derivatives. Simulation results with analysis and discussion are provided in Section 5. Section 6 concludes this paper.

2. Overview of multifrontal method

Direct methods aim at solving a sparse linear system \(Ax = b\) by sparse triangular factorization. When \(A\) is square, an LU factorization is performed to generate the \(LU\) product. Ordering and pivoting strategies are employed to maintain sparsity and numerical stability respectively.

Most sparse solvers take the following three successive phases to solve \(Ax = b\), where \(x\) is the solution.

1. A symbolic analysis phase completes the preparation work. This phase parses the structure of \(A\), finds an ordering sequence of elimination to reduce fill-ins, establishes data structures, and determines the upper bound of memory usage.

2. A numeric factorization phase factorizes \(A\) into its triangular factors, using different factorization schemes. This is the most time-consuming part.

3. A solve phase performs forward and backward substitutions by \(L\) and \(U\) on \(b\) to obtain \(x\).

In this paper, left-looking LU factorization is taken as a reference comparison with multifrontal method. CSparse, a concise but efficient sparse solver, is chosen as the implementation of left-looking LU factorization. Explanations of left-looking LU factorization and CSparse can be found in [5].
2.1. Right-looking multifrontal LU factorization

Gaussian elimination is a typical instance of right-looking LU factorization. At each step, the outer product of pivot row and pivot column is subtracted from the remaining submatrix. This process is referred to as rank-1 update.

Different from Gaussian elimination, multifrontal method assembles the small dense matrices (called frontal matrices) that will be updated by previous outer products, and factorizes the updated frontal matrices by high level BLAS to produce new contribution blocks. Detailed description of multifrontal method on the symmetric positive definite linear system is given by Liu in [14].

Except for symmetric cases, Davis proposed a new strategy to accommodate multifrontal method for unsymmetric-pattern matrices [15]. Davis has incorporated both symmetric and unsymmetric strategies into his solver UMFPACK [15]-[18]. In power system simulations, most matrices have patterns much close to structural symmetry, thus only the symmetric case in UMFPACK is concerned in this paper.

The symmetric strategy in UMFPACK factorizes $PAQ$ into the $LU$ product. In symbolic analysis, a column preordering $Q$ is generated by AMD (Approximating Minimum Degree) algorithm [19] based on the pattern of $A + A^T$. The assembly of frontal matrices during numeric factorization is guided by column elimination tree. Threshold partial pivoting is used, with diagonal entries preferred. Pivoting entries are chosen to make a compromise between the sparsity and the numerical stability. BLAS routines are equipped to improve cache utility. If multiple eliminations occur in one frontal matrix, level-3 BLAS routines TRSM and GEMM are used. Otherwise, GER is used for rank-1 update of the contribution block.

2.2. Efficiency analysis

When the same pre-ordering and pivoting strategies are adopted, left-looking and right-looking LU factorizations have little difference in terms of fill-ins and flops (floating-point operations) [15]. Since left-looking LU factorization can be executed in time proportional to the number of flops [20], fill-in count is an explicit indicator for the efficiency of left-looking LU method. However, acceleration of multifrontal method is generally attributed to the use of high level BLAS. Thus, apart from fill-ins or flops, performance of multifrontal method is also related to frontal matrix sizes.

To demonstrate the importance of frontal matrix sizes in multifrontal method, a comparison between UMFPACK and CSparse is carried out on a number of matrices with different frontal matrix sizes. All the matrices are taken from the Harwell-Boeing set of the UF sparse matrix collection [21]. These selected matrices have nearly symmetric patterns so that UMFPACK adopts the symmetric strategy.

![Fig. 1. Run time ratio of UMFPACK to CSparse for 120 sparse matrices. Run time includes only symbolic and numeric phases. Run time ratio greater than 1.0 indicates that UMFPACK is slower than CSparse.](image)

From Fig. 1, it can be seen that UMFPACK equipped with BLAS becomes more efficient than CSparse only if the largest frontal matrix exceeds a threshold size. As a result, performance of
multifrontal method relies heavily on frontal matrix sizes. If the max frontal matrix size is relatively small, multifrontal method will lose its acceleration effect or perhaps even be decelerated to some extent.

3. Graph theory based analysis of frontal matrix size
An intuitive inspection is that frontal matrix sizes are subject to the sparsity of involved matrix. However, matrices with the same sparsity but with different sparsity patterns shall have very different frontal matrix sizes. Fig. 2 depicts the sparsity patterns of a real power network admittance matrix and a randomly generated matrix with the same size and the same sparsity.

Fig. 2. Sparsity patterns of a power network admittance matrix and a randomly generated matrix with the same size and the same degree of sparsity.

UMFPACK factorization results of these two matrices and some other pairs of matrices, which are constructed in the same way, are listed in Table 1. It can be extrapolated effortlessly from Fig. 2 and Table 1 that matrices with the same size and the same degree of sparsity can have totally different frontal matrix sizes. Therefore, matrix sparsity pattern instead of degree of sparsity is a decisive factor for frontal matrix sizes.

Table 1Comparison of max frontal matrix size between admittance matrices and randomly generated matrices

| Grid         | Size | Nonzero Count | Max Frontal Matrix Size |
|--------------|------|---------------|-------------------------|
| case2746wp   | 2746 | 9292          | 24                      |
| random2746   | 2746 | 9298          | 228                     |
| case3375wp   | 3375 | 11510         | 30                      |
| random3375   | 3375 | 11501         | 280                     |
| ECG09        | 5473 | 18757         | 9                       |
| randomECG09  | 5473 | 18748         | 923                     |
| ECG15        | 8241 | 28797         | 9                       |
| randomECG15  | 8241 | 28787         | 760                     |

Admittance matrices are taken from MATPOWER [22] and east China power grids. UMFPACK adopts the symmetric strategy and records the max frontal matrix size during numeric factorization. Nonzero count refers to the number of nonzero entries before factorization.

3.1. Quantitative indicators for frontal matrix size
In previous sections, max frontal matrix size is regarded as a representative of frontal matrix sizes. For a specific matrix, if different node ordering sequences are used during factorization, the max frontal matrix size shall be different. To eliminate this ambiguity, definitions of front size and min front size
are introduced.

Given a sparse symmetric positive definite matrix $A$ and a specific ordering sequence, front size of $A$ under this ordering sequence is defined to be the dimension of the largest update matrix during multifrontal factorization.

There is a close relationship between the front size and the structure of Cholesky factor $L$. Column count of $L$ is defined to be the number of nonzero entries in each column of $L$. Apparently, front size equals maximum column count minus one. Therefore, given a specific ordering sequence, a symbolic Cholesky factorization will find the structure of $L$ and get the front size under this ordering.

Min front size of matrix $A$ is defined to be a minimum of the front sizes over all possible orderings. The ordering leading to the min front size is defined as the optimal ordering. It is difficult to obtain the optimal ordering of a given matrix, thus min front size is usually approximated instead.

### 3.2. Effective indicator for matrix sparsity pattern

Since sparse matrix can be viewed as the adjacent matrix of its associated graph, matrix sparsity pattern is mathematically equivalent to topology of the associated graph. Topology of a graph is related to many parameters, such as elimination tree height, treewidth, etc. Elimination tree height dictates the minimum steps for a parallel LU factorization. Treewidth of a graph measures how close this graph is to a tree [23]. Conceptions of tree-decomposition and treewidth are introduced with terms and definitions from Bodlaender [24].

Given a sparse symmetric positive definite matrix $A$, we use $G$ to denote the undirected graph associated with $A$.

A tree-decomposition of graph $G = (V, E)$ is a pair $\{(X_i \mid i \in I), T = (I, F)\}$ where $T$ is a tree and $\{X_i\}$ is a collection of subsets of $V$, such that

- $\bigcup X_i = V, i \in I$
- For all $(v, w) \in E$, there exists $i \in I$ with $v, w \in X_i$.
- For all $i, j, k \in I$, if $j$ is on the path from $i$ to $k$ in $T$, then $X_i \cap X_k \subseteq X_j$.

The number of vertices in $X_i$ is denoted by $|X_i|$. The treewidth of a tree-decomposition is equal to $\max |X_i| - 1$. The treewidth of graph $G$ is defined as the minimum treewidth over all possible tree-decompositions. We use $td(G)$ to denote tree-decomposition of $G$ and $tw(G)$ to denote treewidth of $G$ for short.

Bodlaender has also concluded that [24],

$$
treewidth = \min \text{ front size}$$

$$
treewidth \leq \min \text{ etree height} \leq \text{ treewidth} \times \log n$$

Clearly, treewidth is an effective indicator, from which other topological parameters of graph $G$ can be determined easily. In this sense, if the treewidth of original network could be calculated accurately, the performance of multifrontal method would be evaluated ahead of real factorization.

It should be noted that treewidth is nor the length of neither the width of elimination tree nor the length of factorization path. Therefore, traditional ordering methods, like MDML or MDMNP, can shorten the factorization path and broaden the elimination tree [3] [4], although, they are unable to enlarge the treewidth itself.

### 4. Analysis of sparsity structures in power system

#### 4.1. Approximating treewidth of power network

Calculation of treewidth is NP-hard [24]. Various approximation algorithms have been developed for obtaining the treewidth precisely [25]. On one hand, most of them are time consuming and lead to low efficiency when the system scale is large. On the other hand, an exact solution of treewidth is not necessary since partial pivoting might increase the front size by a small amount. Consequently, it is preferable to calculate the treewidth approximately for large-scale power systems so as to achieve high efficiency.
Theorem 1 figures out a means of formulating a tree-decomposition of a given graph based on the structure of the Cholesky factor $L$ of the adjacent matrix $A$. The proof of Theorem 1 is given in Appendix A.

**Theorem 1** Given a symmetric positive definite matrix $A$, its associated graph $G = (V, E)$ and a specific ordering $\pi$. The structure of Cholesky factor $L^*$ under ordering $\pi$ can be denoted as $\{L_i| i \in V\}$, where $L_i$ is the set of row indices in column $i$ of $L^*$. $T = (V, F)$ is the elimination tree. Then, the pair $((L_i| i \in V), T = (V, F))$ is a tree-decomposition of graph $G$, and max column count of $L^*$ minus 1 is the corresponding treewidth.

In power system, the admittance matrix is structurally symmetric and numerically positive semi-definite. According to Theorem 1, a symbolic Cholesky factorization on the admittance matrix will generate a tree-decomposition of the network as well as the treewidth of this tree-decomposition. Therefore, a treewidth approximation method for power networks is developed below.

**Treewidth Approximation Method (TAM)**

Step 1: An AMD ordering is applied to the admittance matrix to reduce fill-ins.

Step 2: A symbolic Cholesky factorization is performed on the admittance matrix to get the column pattern of Cholesky factor $L^*$.

Step 3: The maximum column count of $L^*$ is found to obtain the treewidth of power network.

It should be noted that the result calculated by the treewidth approximation method is the treewidth of tree-decomposition under AMD ordering. Other ordering algorithms, such as nested dissection, minimum degree, etc., can also be used to replace AMD. The reason why we choose AMD ordering is that AMD is used in both UMFPACK and CSparse. In addition, the result returned by TAM is the upper bound of the treewidth of power network. If the ordering is optimal, TAM would obtain the treewidth of power network.

Another notable characteristic of TAM is that the computed treewidth is exactly equal to front size plus one, which just coincides with the maximum column count of Cholesky factor.

4.2. Illustration of TAM on IEEE 9-node system

A small example is given here to show the relationship between treewidth, front size and elimination tree. The IEEE 9-node system is depicted in Fig. 3(a). Node ordering sequence is generated by AMD. The structure of the $L^*$ matrix during factorization is shown in Fig. 3(b). The elimination tree is shown in Fig. 3(c).

![Illustration of IEEE 9-node system, its $L$ matrix during factorization and elimination tree.](image)

Fig. 3. Illustration of IEEE 9-node system, its $L$ matrix during factorization and elimination tree.

All the frontal matrices encountered during multifrontal factorization are listed in Fig. 4(a). The corresponding tree-decomposition is shown in Fig. 4(b).
It is not hard to see that the $i$th frontal matrix in Fig. 4(a) is actually assembled from the nonzero pattern of column $i$ of $L^*$ in Fig. 3(b). On one hand, nodes with row indices in the $i$th frontal matrix can be gathered to form the $i$th clique. In this way, 8 cliques, namely {1,5}, {2,3}, {3,4,8}, {4,5,8}, {5,8,9}, {6,7}, {7,8,9} and {8,9} have been formed for the IEEE-9 case. On the other hand, nodes in one clique have definite sequences from nodes in another clique. E.g., clique {2,3} is the descendent of clique {3,4,8} since node 3, node 4 and node 8 are the parents of node 2. With the application of Theorem 1, a tree-decomposition as shown in Fig. 4(b) is formulated by concatenating all the cliques according to the elimination tree in Fig. 3(c).

Theorem 1 also tells that the treewidth of the tree-decomposition in Fig. 4(b) is 2 (since $\max |X_i|$ is equal to 3 in this case). However, it is quite clear that the width of the elimination tree in Fig. 3(c) is 3. In fact, the treewidth depicts the elimination tree from a vertical perspective while the width of the elimination tree is defined from a horizontal perspective.

### 4.3. Approximating front size of power system matrices

For topology-based power system simulations, such as power flow calculation and small signal stability eigenvalue analysis, min front size of coefficient matrix $J$ can be derived approximately from the treewidth of power network.

Since $J$ is numerically unsymmetric and almost structurally symmetric, definitions of front size and min front size of $J$ are slightly different from those defined for symmetric positive definite case. In [5], Davis has shown that, if matrix $J$ is diagonally dominant and has a symmetric nonzero pattern, then the nonzero patterns of $J$’s LU factorization products $L$ and $U$ are identical with the nonzero patterns of $L^*$ and $L^{**}$ respectively, where $L^*$ is the Cholesky factor of a symmetric positive definite matrix $J^*$ with the same nonzero pattern as $J + J^T$. Therefore, for power system computations, front size of $J^*$ can be designated as front size of $J$ safely. In the same way, min front size of $J^*$ is designated as min front size of $J$. Graph $H$ associated with $J^*$ is designated as the graph of $J$.

Because $J$ is available, front size of $J$ can be calculated directly by the proposed treewidth approximation method with the admittance matrix replaced by $J^*$. Besides this option, an alternative way is to derive the treewidth of $H$ from the treewidth of $G$ immediately, where $G$ stands for the associated graph of admittance matrix $A$.

For power flow calculation, the Jacobian matrix $J$ can be formulated by splitting any node, $n_i$, for instance, in admittance matrix $G$ into two nodes, $n_{ip}$ and $n_{iq}$. If node $n_i$ and node $n_j$ are connected before splitting, the new nodes $n_{ip}$, $n_{iq}$, $n_{jp}$ and $n_{jq}$ are fully connected in $J$. For any tree-decomposition, $(\{X_i | i \in V\}, T = (V, F))$, of $G$, the tree decomposition of $H$ can be derived by expanding every set $X_i = \{n_1, n_2, \ldots, n_j\}$ into a new set $X_{new} = \{n_{ip}, n_{iq}, n_{jp}, n_{jq}, \ldots, n_{ip}, n_{jq}\}$. Therefore, treewidth of $H$ is not larger than twice the treewidth of $G$, namely,

$$tw(H) \leq 2 \times tw(G) \tag{2}$$

For small signal stability eigenvalue analysis, the key point is to solve a sparse linear system in the
form of [26],
\[
\begin{bmatrix}
J_1 + \sigma I & J_2 \\
J_3 & J_4
\end{bmatrix}
\begin{bmatrix}
x \\
z
\end{bmatrix}
=
\begin{bmatrix}
b \\
0
\end{bmatrix}
\]  
(3)

Fig. 5. Augmented Jacobian matrix in power system modeling. \(J_i\) is derived from the differential-algebraic equations of dynamic components. \(J_s\) is derived from the modeling of electrical network, which has its sparsity pattern resembling the Jacobian matrix in power flow calculation. \(J_2\) and \(J_3\) reflect the interconnection of component in \(J_i\) with the network in \(J_s\).

Fig. 5 depicts the augmented Jacobian matrix \(J\). \(J_i\) has the same structure as the power flow Jacobian matrix where all nodes are PQ type. We use \(V(J_i)\) to denote all nodes in \(J_i\) and \(H(J_i)\) to denote the subgraph of \(H\) induced by \(V(J_i)\). A tree-decomposition \(td(H(J_i)) = (\{X_i | i \in V(J_i)\}, T = (V(J_i), F))\) can be formulated based on the tree-decomposition of \(G\).

\(J_i\) is comprised of a sequence of diagonal blocks \(M_1, M_2, \ldots, M_k\). Each diagonal block represents a connected dynamic component. We use \(V(M_i)\) to denote all nodes in \(M_i\) and \(H(M_i)\) to denote the subgraph of \(H\) induced by \(V(M_i)\). For each subgraph \(H(M_i)\), their tree-decomposition \(td(H(M_i))\) can be formulated according to Theorem 1.

Tree-decomposition of \(H\) can be derived by integrating the tree-decompositions of \(H(J_i)\) and \(H(M_i)\) in the following way. Each block \(M_i\) in \(J_i\) is only related to a two-by-two block with row indices \(\{m_{i1}, m_{i2}\}\) on the diagonal of \(J_i\). Now that \(\{m_{i1}, m_{i2}\}\) is an edge in \(H(J_i)\), there exists a set \(X_p\) in \(td(H(J_i))\) satisfying \(\{m_{i1}, m_{i2}\}\in X_p\). Nodes \(\{m_{i1}, m_{i2}\}\) are added to all sets in \(td(H(M_i))\) and an edge is added arbitrarily between \(X_p\) and a set in \(td(H(M_i))\) to obtain a new tree-decomposition. After all blocks \(M_i\) in \(J_i\) are processed, the tree-decomposition of \(H\) is established. Therefore, treewidth of \(H\) is less than or equal to the maximum of twice the treewidth of \(G\) and the treewidth of \(H(M_i)\) plus 2, namely,

\[
tw(H) \leq \max(2 \times tw(G), tw(H(M_i)) + 2)
\]  
(4)

5. Simulation and discussion
To verify the observations aforementioned and to validate the effectiveness of TAM, several sets of experiments are carried out. Programs are executed serially on a PC with Intel i5-2320 CPU and 8 GB memory.

5.1. Test systems
Matrices from the IEEE standard systems, MATPOWER Polish systems and East China Grids are used for experiments.

Information about admittance matrices and power flow Jacobian matrices are given in Table 2 and Table 3 respectively.

Information about the augmented Jacobian matrices can be found in Table 4. Each generator is formulated as a 10-order dynamic model (6 states for the generator itself and 4 states for the excitation system).
Table 2 Detailed information about admittance matrices in test set

| Grid      | Dimension | Nonzero Count |
|-----------|-----------|---------------|
| IEEE57    | 57        | 213           |
| IEEE118   | 118       | 476           |
| IEEE300   | 300       | 1118          |
| Case2383wp| 2383      | 8155          |
| Case2746wp| 2746      | 9292          |
| Case3375wp| 3375      | 11510         |
| ECG05     | 4171      | 13891         |
| ECG09     | 5473      | 18757         |
| ECG15     | 8241      | 28797         |

Table 3 Detailed information about power flow Jacobian matrices in test set

| Grid      | Dimension | Nonzero Count |
|-----------|-----------|---------------|
| IEEE57    | 106       | 718           |
| IEEE118   | 181       | 1051          |
| IEEE300   | 530       | 3736          |
| Case2383wp| 4438      | 27783         |
| Case2746wp| 5127      | 32117         |
| Case3375wp| 6357      | 40706         |
| ECG05     | 7999      | 53163         |
| ECG09     | 10491     | 71831         |
| ECG15     | 15964     | 111576        |

Table 4 Detailed information about augmented Jacobian matrices in test set

| Grid      | Generator number | Jv Dimension | Dimension | Nonzero Count |
|-----------|------------------|--------------|-----------|---------------|
| IEEE118   | 34               | 236          | 576       | 3346          |
| IEEE300   | 69               | 600          | 1290      | 7191          |
| ECG05     | 375              | 8342         | 12092     | 67230         |
| ECG09     | 472              | 10946        | 15666     | 86631         |
| ECG15     | 300              | 16482        | 19482     | 119608        |

5.2. Experimental results for TAM

TAM and QuickBB [25] are used to calculate the treewidth of power network based on the pattern of admittance matrix. Their comparison in terms of treewidth and run time can be found in Table 5.

| Case      | TAM Treewidth | TAM Run Time (s) | QuickBB Treewidth | QuickBB Run Time (s) |
|-----------|---------------|------------------|-------------------|----------------------|
| IEEE57    | 5             | 0.000048         | 5                 | 0.00316              |
| IEEE118   | 4             | 0.000085         | 4                 | 0.00759              |
| IEEE300   | 6             | 0.00018          | ≤6                | 0.0207               |
Table 5 shows that the difference in results computed by TAM and QuickBB is very small. Meanwhile, run time of TAM is lower than QuickBB. In the essence of mathematics, TAM is a variant of the min-degree heuristic for computing an upper bound of treewidth. In QuickBB, the treewidth is obtained by repeated trial-and-error to narrow the gap between upper and lower bounds. For power system networks, since AMD ordering is close to the optimal ordering, other attempts of ordering may cost a lot of time, while give little improvement to the result of AMD. Therefore, TAM with AMD ordering is more efficient than QuickBB.

Table 5 also demonstrates that generally power network has low treewidth, which coincides with the conclusion drawn in [27]. The reason why power network has low treewidth is given in Appendix B. In addition, matrices from the same set of networks have similar treewidth irrespective of their dimensions, while matrices from different power network sets have different treewidth.

Treewidth is an essential indicator for the topology of graph. As a result, approximating the treewidth of power network is a prerequisite to evaluate the performance of multifrontal method.

5.3. Performance comparison between CSparse and UMFPACK
Comparison results of CSparse and UMFPACK concerning run time and fill-ins of the LU factorization are shown in Table 6 and Table 7. Matrices used in Table 6 are power flow Jacobian matrices, which are shown in Table 3. Matrices used in Table 7 are augmented Jacobian matrices of small signal stability analysis, which are shown in Table 4. Both CSparse and UMFPACK use symmetric strategies. Front size is recorded by UMFPACK.

Table 6 Comparison between UMFPACK and CSparse on power flow Jacobian matrices

| Case     | UMFPACK | CSparse |
|----------|---------|---------|
|          | Run Time | Nonzero Count | Front Size | Run Time | Nonzero Count | Time Ratio |
| IEEE57   | 0.000229 | 1091     | 9          | 0.000102 | 1092       | 2.24       |
| IEEE118  | 0.000433 | 1353     | 8          | 0.000154 | 1355       | 2.81       |
| IEEE300  | 0.001267 | 5518     | 16         | 0.000527 | 5554       | 2.41       |
| 2383wp   | 0.010877 | 44276    | 35         | 0.004168 | 44666      | 2.61       |
| 2746wp   | 0.012280 | 52084    | 36         | 0.004809 | 52492      | 2.55       |
| 3375wp   | 0.015513 | 68795    | 47         | 0.006191 | 68795      | 2.51       |
| ECG05    | 0.014852 | 63945    | 18         | 0.005332 | 68635      | 2.78       |
| ECG09    | 0.019530 | 84085    | 18         | 0.007511 | 92831      | 2.60       |
| ECG15    | 0.029567 | 125092   | 16         | 0.011288 | 137244     | 2.62       |

*Nonzero count refers to the number of nonzero entries after factorization.

Table 7 Comparison between UMFPACK and CSparse on augmented Jacobian matrices of small signal stability analysis

| Case     | UMFPACK | CSparse |
|----------|---------|---------|
|          | Run Time | Nonzero Count | Front Size | Run Time | Nonzero Count | Time Ratio |
| IEEE118  | 0.001258 | 4279     | 10         | 0.000425 | 4279       | 2.96       |
| IEEE300  | 0.002793 | 9881     | 16         | 0.001004 | 9881       | 2.78       |
It can be seen from Table 5 and Table 6 that, except IEEE300 system, the front size of power flow Jacobian matrix is less than or equal to twice the network treewidth for all test systems, which accords with (2). For IEEE300 system, since the Jacobian matrix is not diagonally dominant, partial pivoting has to be used for the sake of numeric stability. Then, the node ordering generated by AMD is modified slightly, which leads to a little violation of (2).

Table 5 and Table 7 show that, for small signal stability analysis, the relationship that front size of augmented Jacobian matrix is not larger than twice the network treewidth still holds. This is because in our tests, the order of dynamic model is smaller than twice the power network treewidth. Then the front size of augmented Jacobian matrices is subject to the first factor on the right-hand-side of (4). As for higher order of dynamic model, the front size of augmented Jacobian matrix will be bounded by the second factor on the right-hand-side of (4).

In respect of fill-ins, CSparse and UMFPACK behave almost the same whether the involved matrix is from power flow or small signal stability analysis. This is a natural result since both CSparse and UMFPACK adopt AMD ordering. However, high run time ratios of UMFPACK to CSparse in Table 6 and Table 7 indicate that the BLAS acceleration of UMFPACK is greatly limited due to the small front size of coefficient matrix. Therefore, low treewidth of power network limits the performance of multifrontal method.

5.4. Performance of UMFPACK on matrices with large front size

To evaluate the performance of UMFPACK when the treewidth is large enough, the random network approach from [28] is utilized to generate a sequence of matrices based on the elimination tree of power flow Jacobian matrix. Detailed procedure of random network approach is stated as follows.

Given a power flow Jacobian matrix \( A \), and a series of pre-specified numbers of nonzero entries,

Step 1: The elimination tree of \( A + A^T \) is generated.

Step 2: Edges are added randomly to the tree until nonzero count meets the preset requirement.

Step 3: A new symmetric matrix \( B \) is formed according to the graph generated in Step 2. The diagonal elements are added so that \( B \) is positive definite.

Step 4: Repeating Steps 1-3, a series of matrices are generated.

Power flow Jacobian matrices of 3375wp case and ECG09 case are taken for example. A series of coefficient matrices are generated using the random network approach. Run time of CSparse and UMFPACK is recorded and compared in Fig. 6 and Fig. 7. Front size is recorded by UMFPACK.

![Fig. 6. Run time comparison between CSparse and UMFPACK. Matrices are generated from](image-url)
elimination tree of power flow Jacobian matrix of 3375wp case using random network approach. Run
time includes only symbolic and numeric phases. UMFPACK is equipped with reference BLAS.

Fig. 7. Run time comparison between CSparse and UMFPACK. Matrices are generated from
elimination tree of power flow Jacobian matrix of ECG09 case using random network approach. Run
time includes only symbolic and numeric phases. UMFPACK is equipped with reference BLAS.

A general trend can be found from Fig. 6 and Fig. 7 that as more edges are added to the elimination
tree, topology of the network is changed a lot and front size of the coefficient matrix becomes large.
When the front size reaches to a relatively large value, e.g. 199 in Fig. 6 and 281 in Fig. 7, UMFPACK
outperforms CSparse in terms of computing speed. Therefore, when node interconnections change the
topology of power network and increase the network treewidth to a large extent, multifrontal method
will be suitable for power system simulations.

6. Conclusions
This paper focuses on studying the performance of multifrontal method in power system simulations.
The primary conclusions are drawn below.

Multifrontal method, as a variant of right-looking sparse LU factorization, mainly gets its
acceleration by utilizing high-level BLAS to tackle dense frontal matrices. Hence, small frontal
matrices cannot fully exploit the BLAS acceleration so that the performance of multifrontal method is
diminished.

Frontal matrix size depends on the sparsity pattern rather than the sparsity degree of sparse matrix.
Since the matrix sparsity pattern amounts to the topology of associated graph, a quantitative
relationship between frontal matrix size and graph topology is established through treewidth from the
perspective of graph theory.

In order to assess the performance of multifrontal method before actual computations, TAM is
proposed to approximate the treewidth of power network. Moreover, two derivatives of TAM are also
given for approximating the front size of sparse matrices from power flow calculation and small signal
stability eigenvalue analysis. Simulation results validate the correctness and effectiveness of TAM and
its derivatives.

Generally, power networks have the sparse nature of low treewidth, where frontal matrices are
small and performance of multifrontal method is limited. When the front size is increased by changing
the topology of power network, multifrontal method regains its performance against left-looking
sparse LU factorization.

7. Appendix A
To demonstrate Theorem 1, several definitions and theorems are introduced in advance:

**Definition A.1** Let $A$ be an $n$-by-$n$ sparse symmetric positive definite matrix with Cholesky factor
$L$. The elimination tree of matrix $A$ is defined to be the structure with $n$ nodes such that node $p$ is the
parent of \( j \) if and only if
\[
p = \min\{i > j \mid l_{ij} \neq 0\}
\]

**Theorem A.1** [5] For a Cholesky factorization \( LL^T = A \), and neglecting numerical cancellation, \( a_{ij} \neq 0 \Rightarrow l_{ij} \neq 0 \). That is, if \( a_{ij} \) is nonzero, then \( l_{ij} \) will be nonzero as well.

**Theorem A.2** [29] For a Cholesky factorization \( LL^T = A \), and neglecting numerical cancellation, \( l_{ki} \neq 0 \) and \( k > i \) imply that \( i \) is a descendant of \( k \) in the elimination tree \( T \).

**Theorem A.3** [14] If node \( k \) is a descendant of \( j \) in the elimination tree, then the structure of the vector \((l_{jk}, \ldots, l_{nk})\) is contained in the structure of \((l_{ij}, \ldots, l_{in})^T\).

A is given a symmetric positive definite matrix. Let \( G = (V, E) \) be the undirected graph associated with \( A \). A Cholesky factorization factorizes \( A \) into its Cholesky factors \( L \) and the column pattern of \( L \) is computed. A pair \((\{X_i \mid i \in V\}, T = (V, F))\) is constructed as follows:

1. Let \( i_0, i_1, \ldots, i_r \) be the nonzero row index of the \( k \)th column of \( L \), with \( i_0 = k \). The set \( X_k \) is formed as \( \{i_0, i_1, \ldots, i_r\} \).

2. \( T \) is the elimination tree of \( A \).

Since \( k \in X_k \cup X_r = V \), \( i \in I \) follows.

For all \( (v, w) \in E \), there exists an \( a_{vw} \neq 0 \), and vice versa. From Theorem A.1, \( a_{vw} \neq 0 \) means \( l_{vw} \neq 0 \) and \( v, w \in X_w \). Therefore, for all \( (v, w) \in E \), \( v, w \in X_w \).

Given any \( i, j, k \in V \) with \( j \) on the path from \( i \) to \( k \) in \( T \), assuming that there exists \( v \in X_i \cap X_k \), in other word, \( l_{vi} \neq 0 \), \( l_{vk} \neq 0 \). Apparently, \( v > i \) and \( v \neq k \) stand. From Theorem A.2, \( v \) is the ancestor of \( i \) and \( k \). If \( j = v \), \( v \in X_j \) stands. Otherwise, \( j \) is either on the path from \( i \) to \( v \) or on the path from \( k \) to \( v \).

For convenience, suppose \( j \) is on the path from \( i \) to \( v \). From Theorem A.3, since \( i \) is the decedent of \( j \) in elimination tree, the structure of the vector \((l_{jk}, \ldots, l_{nk})^T\) is contained in the structure of \((l_{ij}, \ldots, l_{in})^T\), which means \( l_{ij} \neq 0 \Rightarrow l_{ij} \neq 0 \). Therefore, \( v \in X_j \). Consequently, for all \( i, j, k \in V \), if \( j \) is on the path from \( i \) to \( k \) in \( T \), then \( X_i \cap X_k \subseteq X_j \).

In conclusion, the pair \((\{X_i \mid i \in V\}, T = (V, F))\) constructed is a tree-decomposition of \( G \). The max column count of Cholesky factor \( L \) represents the treewidth of the current decomposition.

8. Appendix B
Three sections are prepared to explain the reason why power network has low treewidth. The first section introduces Lemma B.1 from [30]. The second section depicts the organization of power network. The last section presents the detailed analysis for low treewidth property in power network.

Let \( G = (V, E) \) be the original graph and \( S \subseteq V \) be the subset of nodes to be deleted. We use \( G_{VS} \) to denote the graph after node deletion and \( H_1, H_2, \ldots, H_t \) to denote the connected subgraphs obtained. \( S_i \subseteq S \) is used to denote nodes in \( S \) which have neighbors in \( H_i \). A super graph of \( G, H = (V, E^+) \), is defined by adding edges between nodes in \( S_i \), so that each set \( S_i \) becomes a clique. \( H_i \) is used to denote the subgraph of \( H \) induced by nodes in \( S_i \). Lemma B.1 gives an upper bound of treewidth of \( G \).

**Lemma B.1**
\[
tw(G) \leq tw(H_S) + \max\{tw(H_i)\} + 1
\]

Unlike networks in other engineering fields, organization of power network is based on the geographical wiring diagram. Subsystems are connected from city to city, region to region and state to state to form a whole system. Connections inside each subsystem are close while connections between these subsystems are loose. Removing tie-line nodes can break the system into several independent subsystems. These subsystems shall be further partitioned in the same way recursively. A two-level nested partitioning of IEEE118 case given in Fig. 8 is taken as an example to illustrate this nature.
We use $S$ to denote the set of tie-line nodes in power network. Deleting $S$ from the original graph $G$ will break the system into several subsystems, denoted by $H_i$. A super graph of $G$ is formulated according to section 1. For treewidth of $H_i$, since connections between different subsystems are very loose, tie-lines nodes are quite few, and the size of each clique $S_i$ is supposed to be even small. Hence, treewidth of $H_i$ is limited. Now that each subsystem $H_i$ can be repartitioned in the same way, the treewidth of these subgraphs shall be low as well. To sum up, the treewidth of the original graph is low according to Lemma B.1.

Acknowledgements
The authors are grateful to Prof. Timothy A. Davis of Texas A&M University, USA for his helpful discussions and suggestions in preparing this article. The authors would like to thank the State Energy Smart Grid R&D Center, Shanghai, China, for its support.

References
[1] W. F. Tinney and J. W. Walker, “Direct solutions of sparse network equations by optimally ordered triangular factorizations,” Proc. IEEE, no. 55, pp. 1801–1809, Nov. 1967.
[2] W. F. Tinney, V. Brandwajn, and S. M. Chan, “Sparse vector methods,” IEEE Trans. Power Syst., vol. PAS-104, no. 3, pp. 295–301, 1985.
[3] G. D. Prasad, A. K. Jana, and S. C. Tripathy, “An efficient node-reordering technique to improve sparse vector methods,” International Journal of Electrical Power & Energy Systems, vol. 12, no. 2, pp. 134–137, 1990.
[4] H. B. Gooi, Y. Q. Wang. “Efficient ordering algorithms for sparse matrix/vector methods,” International Journal of Electrical Power & Energy Systems, vol. 20, no. 1, pp. 53-59, 1998.
[5] T. Davis, “Direct methods for sparse linear systems,” vol. 2, SIAM, 2006.
[6] I. Duff and J. Reid, “The multifrontal solution of indefinite sparse symmetric linear systems,” ACM Trans. Math. Softw., vol. 9, pp. 302–325, 1983.
[7] J. Dongarra, J. D. Croz, and S. Hammarling, “A set of level 3 basic linear algebra subprograms,” ACM Trans. Math. Softw., vol. 16, pp. 1–17, 1990.
[8] I. Duff, “A review of frontal methods for solving linear systems,” Comput. Phys. Commun., vol. 97, pp. 45–52, 1996.
[9] N.I.M. Gould, J.A. Scott, “A Numerical Evaluation of HSL Packages for the Direct Solution of Large Sparse, Symmetric Linear Systems of Equations,” ACM Trans. Math. Softw., vol. 30, no. 3, pp. 300–325, 2004.
[10] N.I.M. Gould, J.A. Scott, Y. Hu, “A Numerical Evaluation of Sparse Direct Solvers for the Solution of Large Sparse Symmetric Linear Systems of Equations,” ACM Trans. Math. Softw., vol. 33, no. 2, pp. 10, 2007.
[11] S. Khaitan, J. McCalley, and Q. Chen, “Multifrontal solver for online power system time-domain simulation,” IEEE Trans. Power Syst., vol.23, no. 4, pp. 1727–1737, Nov. 2008.
[12] S. K. Khaitan and J. D. McCalley, “A class of new preconditioners for linear solvers used in power system time-domain simulation,” IEEE Trans. Power Syst., vol. 25, no. 4, pp.1835–44, Nov. 2010.
[13] X. Li, F. Li, J.M.Clarke. “Exploration of multifrontal method with GPU in power flow computation,” in Proc. of the IEEE Power and Energy Society General Meeting (PES), pp. 1–5, 2013.
[14] J. Liu, “The multifrontal method for sparse matrix solution: Theory and practice,” SIAM Rev., vol. 34, no. 1, pp. 82–109, 1992.
[15] T. Davis, “A column pre-ordering strategy for the unsymmetric-pattern multi-frontal method,” ACM Trans. Math. Softw., vol. 30, no. 2, pp. 165–195, 2004.
[16] T. Davis, “Algorithm 832: UMFPACK—An unsymmetric-pattern multifrontal method,” ACM Trans. Math. Softw., vol. 30, no. 2, pp. 196–199, 2004.
[17] T. Davis and I. Duff, “A combined unifrontal/multifrontal method for unsymmetric sparse matrices,” ACM Trans. Math. Softw., vol. 25, no. 1, pp. 1–19, 1997.
[18] T. Davis and I. Duff, “An unsymmetric-pattern multifrontal method for sparse LU factorization,” SIAM J. Matrix Anal. Appl., vol. 18, no. 1, pp. 120–145, 1997.
[19] T. Davis, P. Amestoy, and I. Duff, “Algorithm 837: AMD, an approximate minimum degree ordering algorithm,” ACM Trans. Math. Softw., vol. 30, no. 3, pp. 381–388, 2004.
[20] J.R. Gilbert and T. Peierls, “Sparse partial pivoting in time proportional to arithmetic operations,” SIAM Journal on Scientific and Statistical Computing, vol 9, no. 5, pp. 862–874, 1988.
[21] The University of Florida Sparse Matrix Collection T. A. Davis and Y. Hu, ACM Transactions on Mathematical Software, Vol 38, Issue 1, 2011, pp 1:1 - 1:25. http://www.cise.ufl.edu/research/sparse/matrices, [2016-01-21 as last accessed].
[22] R. D. Zimmerman, S. Murillo, x. C. E. nchez, and R. J. Thomas, “MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education,” Power Systems, IEEE Transactions on, vol. 26, pp. 12-19, 2011.
[23] A. Ferrara, G. Pan, M.Y. Vardi. “Treewidth in verification: Local vs. global,” Logic for Programming, Artificial Intelligence, and Reasoning. Springer Berlin Heidelberg, pp. 489-503, 2005.
[24] H.L. Bodlaender, J.R. Gilbert, H. Hafsteinsson, et al. “Approximating treewidth, pathwidth, frontsize, and shortest elimination tree,” Journal of Algorithms, vol. 18, no. 2, pp. 238-255, 1995.
[25] V. Gogate, R. Dechter. “A complete anytime algorithm for treewidth,” Proceedings of the 20th conference on Uncertainty in artificial intelligence, pp. 201-208, 2004.
[26] J. Rommes, N. Martins. “Exploiting structure in large scale electrical circuit and power system problems,” Linear Algebra and its Applications, vol. 431, no.3, pp. 318-333, 2009.
[27] Atkins K, Chen J, Anil Kumar V S, et al. “The structure of electrical networks: a graph theory based analysis,” International journal of critical infrastructures, vol. 5, no.3, pp. 265-284, 2009.
[28] F.L. Alvarado. “Computational complexity in power systems,” IEEE Trans. Power App. Syst., vol. 95, no. 4, pp. 1028–1037, 1976.
[29] R. Schreiber. “A new implementation of sparse Gaussian elimination,” ACM Trans. Math. Softw., vol. 8, no. 3, pp. 256-276, 1982.
[30] M. C. Golumic and B. Faltings, “A little lemma on treewidth,” www.lirmm.fr/~paul/ANR/treewidth-little-lemma-paper.ps