# Power System Sparse Matrix Statistics

F. Safdarian, Z. Mao, W. Jang, and T. J. Overbye
Department of Electrical and Computer Engineering
Texas A&M University
College Station, TX
{fsafdarian; zeyumao2; wjang777; overbye} @tamu.edu

***Abstract*--This paper provides practice-oriented statistics on the scalability and the growth of power system sparse matrix computational complexity, with the results based on models of real and synthetic electric grids, including very large grids with up to 110,195 buses. The statistics include how the computational effort of factorizing a Jacobian matrix and the factorization path length scale with the system size $n$, which shows the number of buses. The study shows the number of nonzeros in the Jacobian matrix after factorization grows as $n^{1.07}$, the time to factor the matrix grows as $n^{1.38}$, and Forward (F) /Backward (B) substitution time grows as $n^{1.17}$. In addition, applying sparse vector methods, the fast forward/fast backward substitution (FF/FB) grows as $n^{0.45}$, which shows an improvement in the computational effort. Taking advantage of the statistics mentioned in this paper, the trend, scaling, and computation complexity of factorization steps can be easily predicted.**

***Index Terms*—Power flow, computation complexity, sparsity, factorization path, fills, approximate minimum degree algorithm.**

## I. Introduction

As is common in many fields, in electric transmission system analysis a key computational challenge is the solution of **Ax** = **b** where **A** is an n-dimensional square matrix and **b** is known. For the transmission grid analysis, **A** is usually structurally symmetric and quite sparse, with its sparsity dependent upon the transmission system topology. A common solution technique for such sparse systems, first introduced in [1] and [2], is to factor **A** into a lower triangular matrix **L** and an upper triangular matrix **U** with **A** = **LU**. Then **x** is determined by defining **y** = **Ux**, solving for **y** in **Ly** = **b** with a process known as forward substitution (F), and then solving for **x** in **y** = **Ux** with a process known as backward substitution (B). The matrix factorization and the forward/backward substitution (F/B) can take advantage of system sparsity. One purpose of this paper is to show how the factorization and the F/B scale with the grid size.

In some power system applications, the computational complexity can be significantly improved by taking advantage of sparse vector methods, first introduced in [3]. Sparse vector methods can be used when **b** is sparse. With sparse vector methods, there are two common classes of problems, both of which require that **A** first be factored and then selected elements of **y** be calculated using a process known as a fast forward substitution (FF). The first class is that if only a few elements of **x** are desired, they can be determined quite quickly using a fast backward substitution (FB). A common application of the FF/FB is to determine selected diagonal elements of the inverse of **A**. The second class is if all, or most, of the elements of **x** are desired a regular backward substitution can be used with the **y** calculated using the FF. As noted in [3], the computational complexity required to the FF and FB depend essentially linearly on the length of **A**'s factorization paths. Another purpose of this paper is to show how factorization paths scale with the system size.

Several references in the literature propose factorization methods for sparse matrices in general. Work [4] reviews various sparse matrices that arise in optimization. Reference [5] introduces the construction and properties of a factorized sparse approximate inverse preconditioning that is well suited for implementation on modern parallel computers. In [6], the use of an out-of-core sparse matrix package for the numerical solution of partial differential equations involving complex geometries arising from aerospace applications is discussed. In [7], the authors propose an interpolation between two common directions for sparse matrix factorization: a cheap, inefficient number of iterations over sparse search directions (e.g., coordinate descent), and an expensive number of iterations in well-chosen search directions (e.g., conjugate gradients). They show how to perform cheap iterations along nonsparse search directions, provided that these directions can be extracted from a sparse factorization. Authors of [8] design and implement a parallel and fully algebraic preconditioner based on an approximate sparse factorization using low-rank matrix compression for indefinite systems using hierarchical matrices and randomized sampling. In [9], a large-scale network embedding algorithm of sparse matrix factorization is proposed. Reference [10] introduces a domain-specific code generator that optimizes sparse matrix computations by decoupling the symbolic analysis phase from the numerical manipulation stage in sparse codes.

Given the importance of understanding how computations scale with system size, there is surprisingly little information in the power system literature about the computational complexity of power systems' sparse matrix calculations with the sole exception of [11]. Using electric grids with up to 320

 Presented at the 2022 IEEE Texas Power and Energy Conference, College Station, TX, February 2022.

buses, work [11] showed that the computation complexity of matrix factorization is $n^{1.4}$ and that of F/B is $n^{1.2}$. No similar statistics exist for sparse vector methods in the literature, which are introduced in [3] and [12]. The efficiency of sparse vector methods and average factorization path lengths are compared between systems with up to a few thousand buses in [3]. The authors of [12] improve parallel computations of sparse vector methods by incorporating bus ordering methods with matrix partitioning schemes to preserve the sparsity in the inverse of **L** and **U** and decrease the length of the factorization paths. When factorizing a sparse matrix, some originally zero values can become nonzero; these values are called "fills" (fill-ins). It is desired to order matrix **A** prior to the factorization in a way that the number of fills is minimized to preserve the sparsity as much as possible; since the computation complexity to factor a sparse matrix depends on the number of nonzeros and the way of ordering has a significant impact on the number of fills [2].

Factorization and sparse vector methods are widely used in power system problems. In steady-state analysis, **A** can be the Jacobian matrix used to solve an AC power flow (ACPF), the susceptance matrix to solve a DC power flow (DCPF), or the matrix used in a time-domain simulation solution. Reference [13] presents statistics of computational time required to build the admittance matrix of test systems ranging from 200 to 70,000 nodes using a sparse matrix approach and parallel computing. Reference [14] studies the impact of partitioning the network on the reduction of computational burden on larger systems such as the Eastern Interconnection (EI) model with 5838 buses. Factorization is also used in sensitivity analysis as an efficient way to quickly assess the potential problematic power flow solutions [15]. Another recent application of the sparsity technique includes transient stability simulations using ordering and a multipath sparse vector method [16].

In this paper, statistics are provided for a number of different actual and electric grid models ranging in size from a small island up to covering much of North America. Also, each of the studied grids is an original full-scale transmission system model, as opposed to being an equivalence portion of a larger grid. Equivalencing a grid is when a part of a larger system with a fewer number of buses is selected for study and represents the connections with the separated parts. The drawback of equivalencing is that as the grid is equivalenced, some original characteristics are lost [17]. For example, interconnection flows between the equivalenced area and external areas of the system may significantly change.

## II. Numerical Results

This section provides statistics to show how the factorization of Jacobian matrices and the factorization paths grow along with the increase of the system size. The size of studied real grids varies from 109 buses up to 110,195 buses. For Jacobian matrix, a single matrix element can be real, complex, or blocks. In the studied Jacobian matrix for ACPF, elements are stored using two by two matrix blocks. The presented statistics in this paper refer to the number of these blocks. Therefore, the actual size of **A** is two times larger than the presented values and the number of actual elements is four times larger. The other main assumptions include **A** is a nonsingular matrix and the diagonals have nonzero values originally or by ordering which is common in similar studies.

The growth of different statistics such as the number of nonzeros after factorization, average and longest lengths of factorization paths, F/B substitution time and factorization time versus network size are calculated. The trend of each pattern is estimated considering a metric to measure the accuracy of fit for regression models. This metric is called the coefficient of determination ($R^2$) and is calculated as follows.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{1}$$

$$SS_{tot} = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{2}$$

$$SS_{res} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

where $SS_{res}$ is the sum of squares of residuals, $SS_{tot}$ is the total sum of squares, $y_i$ is the $i^{\text{th}}$ actual data point, $\bar{y}$ is the mean of the actual data, $\hat{y}_i$ is the $i^{\text{th}}$ predicted data point. $R^2$ is a number between 0 and 1. In general, $R^2$ values close to 1 ($SS_{res} \cong 0$) indicate that the model perfectly fits the data. On the other hand, $R^2$ values close to 0 represent a weak fitting on the data [18].

Simulations are carried out using PowerWorld [19], Python and MATLAB on a computer with an Intel(R) Core(TM) i7-9750H 2.59 GHz CPU and 32GB of RAM. The Approximate Minimum Degree Algorithm (AMD) [20], which is much faster than other ordering methods that compute an exact degree [21], is applied for ordering and KLU [22] is used for symbolic factorization [22-24].

In order to validate the results with the most widely used ordering methods, Minimum Degree (MD) algorithm [1, 2], AMD [20], Nested Dissection (ND) [25], and Multilevel Nested Dissection (MND) [26] are implemented and similar trends are achieved. The comparison of the number of fills with these methods is shown in Fig. 1.

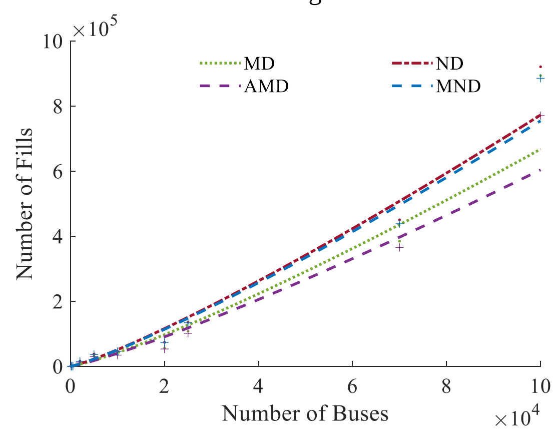

Fig. 1. The number of fills vs. the number of buses.

As it can be observed, using different methods does not change the growth pattern of the number of fills. This is mainly because the size of studied grids are very large, and small variations in the number of fills are negligible compared to the growth in the number of buses. Please note that interconnectedness is also important defining the size of the grids based on radial/mesh in the sub grids and it can be further studied in future work. In this paper, it is assumed that interconnectedness increases as the number of buses increase.

The actual grids are studied for F/B substitution time and the time to perform factorization as well as the average factorization path, its standard deviation (STD) and the longest length of factorization paths. Table I shows the statistics on the results based on the factorization on the actual grids.

TABLE I. THE F/B SUBSTITUTION TIME, THE FACTORIZATION TIME, THE AVERAGE FACTORIZATION PATH, THE STD OF AVERAGE FACTORIZATION PATH, AND THE LONGEST LENGTH OF FACTORIZATION PATHS ON ACTUAL GRIDS

| n | Time (ms) | | Factorization path | | |
|---|---|---|---|---|---|
| | F/B | Factorization | Ave length | STD of ave length | Longest length |
| 109 | 0.02 | 0.02 | 7.96 | 2.89 | 13 |
| 482 | 0.03 | 0.09 | 22.15 | 4.73 | 33 |
| 961 | 0.07 | 0.02 | 28.93 | 5.30 | 37 |
| 2,294 | 0.28 | 0.7 | 33.07 | 9.36 | 55 |
| 2,522 | 0.24 | 0.4 | 25.34 | 5.89 | 43 |
| 7,098 | 1.58 | 6.2 | 94.73 | 29.94 | 146 |
| 20,131 | 1.94 | 5.4 | 78.26 | 16.57 | 118 |
| 22,338 | 1.81 | 7.1 | 74.49 | 12.88 | 106 |
| 23,128 | 2.9 | 7.5 | 75.16 | 14.39 | 112 |
| 62,605 | 18.41 | 52.4 | 172.54 | 28.55 | 253 |
| 86,691 | 19.87 | 88.3 | 205.24 | 41.27 | 302 |
| 87,081 | 47.74 | 101 | 194.82 | 38.92 | 280 |
| 110,195 | 32.82 | 142 | 218.74 | 48.83 | 323 |

Estimating the growth trend from Table I, the trend of factorization time, the expected time for factorization grows as $n^{1.38}$ , where n is the number of buses. In addition, F/B time grows as $n^{1.17}$. However, using sparse vector methods, the trend of the average factorization path is $1.08n^{0.45}$ and the longest length of factorization path increases as $1.77n^{0.44}$. According to these trends, the expected computation complexity for the average factorization path is $n^{0.45}$ and for the longest factorization path is $n^{0.44}$. This shows an improvement in the computation complexity, using sparse vector methods.

For further comparison, synthetic grids [27, 28], ranging in size from 40 buses to 82,000 buses [29] are also studied and the patterns are compared with actual grids, which are considered as the benchmark. Details on creating these synthetic grids are found in [27] and the grids are validated based on actual grids in [28]. Table II shows the statistics on factorization including the approximate F/B substitution time, the approximate factorization time, the average factorization path, the STD of average factorization path, and the longest length of factorization paths, for the synthetic grids and Figure 2 shows the trend of factorization time and F/B substitution time.

Fig. 3 shows the trends of average and the longest factorization path versus the number of buses in both real and synthetic grids. Table III shows a summary of computation complexities on the studied parameters and their accuracy metric $R^2$. It is observed that the statistics of synthetic grids are very close to the statistics from actual grids. The slight difference is mainly because the initial ordering has an impact on the number of fills and factorization paths. Also, as it is expected, the number of nonzeros of **A** in each block before factorization (BF) grows linearly with an increase in the system size. After factorization (AF), because of the added fills, the growth factor of synthetic grids is $n^{1.05}$ , which is close to $n^{1.07}$ for the real grids as it is shown in Fig. 4.

TABLE II. THE F/B SUBSTITUTION TIME, THE FACTORIZATION TIME, THE AVERAGE FACTORIZATION PATH, THE STD OF AVERAGE FACTORIZATION PATH, AND THE LONGEST LENGTH OF FACTORIZATION PATHS ON SYNTHETIC GRIDS

| n | Time (ms) | | Factorization path | | |
|---|---|---|---|---|---|
| | F/B | Factorization | Ave length | STD of ave length | Longest length |
| 40 | 0.004 | 0.008 | 7.83 | 2.52 | 12 |
| 42 | 0.01 | 0.014 | 10.88 | 3.68 | 16 |
| 150 | 0.01 | 0.027 | 15.45 | 7.74 | 34 |
| 200 | 0.01 | 0.036 | 14.42 | 3.50 | 21 |
| 500 | 0.03 | 0.06 | 17 | 4.25 | 28 |
| 500 | 0.04 | 0.09 | 22.78 | 6.77 | 40 |
| 2000 | 0.34 | 1.589 | 49.26 | 14.44 | 90 |
| 10000 | 1.31 | 6.528 | 101.91 | 27.76 | 154 |
| 10000 | 1.29 | 6.371 | 105.94 | 25.91 | 163 |
| 25000 | 5.34 | 37.972 | 182.4 | 62.21 | 296 |
| 30000 | 8.27 | 49.753 | 158.34 | 26.61 | 208 |
| 70000 | 24.26 | 150.655 | 312.23 | 117.27 | 624 |
| 82000 | 30.09 | 156.245 | 281.01 | 132.43 | 624 |

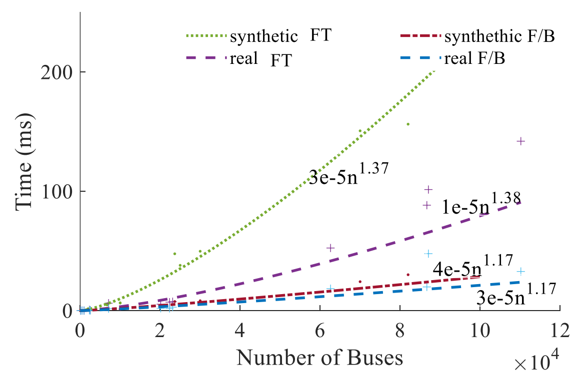

Fig. 2. The Factorization time and F/B time vs. the number of buses.

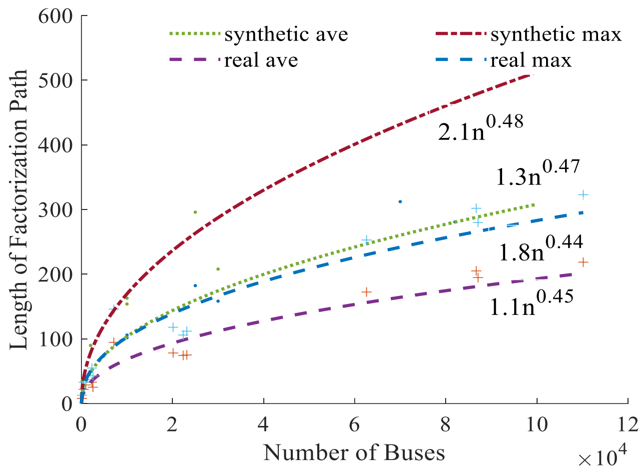


Fig. 3. The average and the longest factorization path vs. the number of buses.

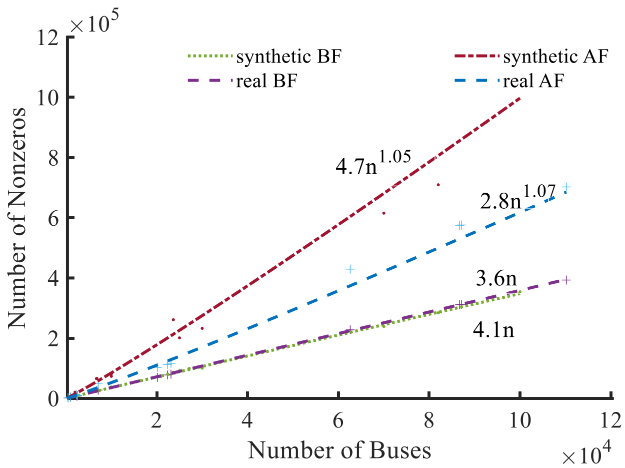


Fig. 4. The number of nonzeros B/A factorization vs. the number of buses.

TABLE III. THE GROWTH AND ACCURACY OF NONZEROS AFTER FACTORIZATION, F/B SUBSTITUTION TIME, THE FACTORIZATION TIME, THE AVERAGE FACTORIZATION PATH, AND THE LONGEST LENGTH

| n | Nonzeros AF | Time (ms) | | Factorization path | |
|---|---|---|---|---|---|
| | | F/B | Factorization | Ave length | Max length |
| Real grids | $n^{1.07}$ | $n^{1.17}$ | $n^{1.38}$ | $n^{0.45}$ | $n^{0.44}$ |
| Real grids $R^2$ | 1 | 0.83 | 0.98 | 0.95 | 0.95 |
| Synthetic grids | $n^{1.05}$ | $n^{1.17}$ | $n^{1.37}$ | $n^{0.47}$ | $n^{0.48}$ |
| Synthetic grids $R^2$ | 1 | 1 | 0.82 | 0.98 | 0.93 |

## III. CONCLUSION AND FUTURE WORK

The sparse matrix statistics of large power systems with a wide variety of sizes are presented. The computational effort of factorizing the Jacobian matrix, the average/longest length of factorization paths, and the time to perform factorization are studied for the power system models with up to a hundred thousand buses. The paper shows how factorization time and the F/B substitution time scale with the grid size. The estimated growth of factorization time is $n^{1.38}$ and the expected growth of F/B substitution time is $n^{1.17}$. In addition, the paper shows how factorization paths scale with system size. The average factorization path is expected to grow as $n^{0.45}$. This shows how applying sparse vector methods improves the computation complexity since FF/FB substitution time is proportional to the length of factorization path.

In the future, we are interested in focusing on graph partitioning, and the analysis of sub-graph complexity as introduced in [30, 31] and the application of graph partitioning on the complexity of power systems ‘sparse matrices.



## ACKNOWLEDGMENT

This work was partially supported through funding provided by the U.S. National Science Foundation in Award 1916142, the U.S. Department of Energy (DOE) under award DE-OE0000895, the US ARPA-E, and PSERC.

## REFERENCES


[1] N. Sato and W. Tinney, "Techniques for exploiting the sparsity or the network admittance matrix," *IEEE Transactions on Power Apparatus and Systems*, vol. 82, no. 69, pp. 944-950, 1963.

[2] W. F. Tinney and J. W. Walker, "Direct solutions of sparse network equations by optimally ordered triangular factorization," *Proceedings of the IEEE*, vol. 55, no. 11, pp. 1801-1809, 1967.

[3] W. Tinney, V. Brandwajn, and S. Chan, "Sparse vector methods," *IEEE transactions on power apparatus and systems*, no. 2, pp. 295-301, 1985.

[4] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, "Sparse matrix methods in optimization," *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 562-589, 1984.

[5] L. Y. Kolotilina and A. Y. Yeremin, "Factorized sparse approximate inverse preconditionings I. Theory," *SIAM Journal on Matrix Analysis and Applications*, vol. 14, no. 1, pp. 45-58, 1993.

[6] D. P. Young, R. G. Melvin, F. T. Johnson, J. E. Bussoletti, L. B. Wigton, and S. S. Samant, "Application of sparse matrix solvers as effective preconditioners," *SIAM Journal on Scientific and Statistical Computing*, vol. 10, no. 6, pp. 1186-1199, 1989.

[7] M. T. Schaub, M. Trefois, P. Van Dooren, and J.-C. Delvenne, "Sparse matrix factorizations for fast linear solvers with application to Laplacian systems," *SIAM Journal on Matrix Analysis and Applications*, vol. 38, no. 2, pp. 505-529, 2017.

[8] P. Ghysels, S. L. Xiaoye, C. Gorman, and F.-H. Rouet, "A robust parallel preconditioner for indefinite systems using hierarchical matrices and randomized sampling," in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2017: IEEE, pp. 897-906.

[9] J. Qiu *et al.*, "Netsmf: Large-scale network embedding as sparse matrix factorization," in *The World Wide Web Conference*, 2019, pp. 1509-1520.

[10] K. Cheshmi, S. Kamil, M. M. Strout, and M. M. Dehnavi, "Sympiler: transforming sparse matrix codes by decoupling symbolic analysis," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017, pp. 1-13.

[11] F. L. Alvarado, "Computational complexity in power systems," *IEEE Transactions on Power Apparatus and Systems*, vol. 95, no. 4, pp. 1028-1037, 1976.

[12] F. L. Alvarado, D. C. Yu, and R. Betancourt, "Partitioned sparse A/sup-1/methods (power systems)," *IEEE Transactions on Power Systems*, vol. 5, no. 2, pp. 452-459, 1990.

[13] F. Gonzalez-Longatt, M. N. Acosta, M. Andrade, E. Vazquez, H. R. Chamorro, and V. K. Sood, "Multi-Core Platform of Admittance Matrix Formation of Power Systems: Computational Time Assessment," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, 2020: IEEE, pp. 1-6.

[14] M. K. Enns, W. F. Tinney, and F. L. Alvarado, "Sparse matrix inverse factors (power systems)," *IEEE Transactions on Power Systems,* vol. 5, no. 2, pp. 466-473, 1990.

[15] K. S. Shetye, T. J. Overbye, A. B. Birchfield, J. D. Weber, and T. L. Rolstad, "Computationally Efficient Identification of Power Flow Alternative Solutions with Application to Geomagnetic Disturbance Analysis," in *IEEE Texas Power and Energy Conference (TPEC)*, 2020.

[16] T. Xiao, J. Wang, Y. Gao, and D. Gan, "Improved sparsity techniques for solving network equations in transient stability simulations," *IEEE Transactions on Power Systems,* vol. 33, no. 5, pp. 4878-4888, 2018.

[17] J. B. Ward, "Equivalent circuits for power-flow studies," *Electrical Engineering,* vol. 68, no. 9, pp. 794-794, 1949.

[18] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to linear regression analysis*. Wiley, 2001.

[19] "PowerWorld Simulator." [Online]. Available: www.powerworld.com.

[20] P. R. Amestoy, T. A. Davis, and I. S. Duff, "Algorithm 837: AMD, an approximate minimum degree ordering algorithm," *ACM Transactions on Mathematical Software (TOMS),* vol. 30, no. 3, pp. 381-388, 2004.

[21] A. George and J. W. Liu, "The evolution of the minimum degree ordering algorithm," *Siam review,* vol. 31, no. 1, pp. 1-19, 1989.

[22] T. A. Davis and E. Palamadai Natarajan, "Algorithm 907: KLU, a direct sparse solver for circuit simulation problems," *ACM Transactions on Mathematical Software (TOMS),* vol. 37, no. 3, pp. 1-17, 2010.

[23] J. Guo, H. Liang, S. Ai, C. Lu, H. Hua, and J. Cao, "Improved approximate minimum degree ordering method and its application for electrical power network analysis and computation," *Tsinghua Science and Technology,* vol. 26, no. 4, pp. 464-474, 2021.

[24] X. Chen, Y. Wang, and H. Yang, "NICSLU: An adaptive sparse matrix solver for parallel circuit simulation," *IEEE transactions on computer-aided design of integrated circuits and systems,* vol. 32, no. 2, pp. 261-274, 2013.

[25] A. George, "Nested dissection of a regular finite element mesh," *SIAM Journal on Numerical Analysis,* vol. 10, no. 2, pp. 345-363, 1973.

[26] T. N. Bui and C. Jones, "A heuristic for reducing fill-in in sparse matrix factorization," Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA …, 1993.

[27] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on power systems,* vol. 32, no. 4, pp. 3258-3265, 2016.

[28] A. B. Birchfield *et al.*, "A metric-based validation process to assess the realism of synthetic power grids," *Energies,* vol. 10, no. 8, p. 1233, 2017.

[29] [Online]. Available: https://electricgrids.engr.tamu.edu/.

[30] D. Delling, A. V. Goldberg, I. Razenshteyn, and R. F. Werneck, "Graph partitioning with natural cuts," in *2011 IEEE International Parallel & Distributed Processing Symposium*, 2011: IEEE, pp. 1135-1146.

[31] D. LaSalle and G. Karypis, "Multi-threaded graph partitioning," in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, 2013: IEEE, pp. 225-236.