

Technical University of Denmark



## Super-Positioning of Voltage Sources for Fast Assessment of Wide-Area Thévenin Equivalents

Møller, Jakob Glarbo; Jóhannsson, Hjörtur; Østergaard, Jacob

*Published in:*

I E E Transactions on Smart Grid

*Link to article, DOI:*

[10.1109/TSG.2016.2639559](https://doi.org/10.1109/TSG.2016.2639559)

*Publication date:*

2017

*Document Version*

Early version, also known as pre-print

[Link back to DTU Orbit](#)

*Citation (APA):*

Møller, J. G., Jóhannsson, H., & Østergaard, J. (2017). Super-Positioning of Voltage Sources for Fast Assessment of Wide-Area Thévenin Equivalents. I E E Transactions on Smart Grid, 8(3), 1488-1493. DOI: 10.1109/TSG.2016.2639559

## DTU Library

Technical Information Center of Denmark

---

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# Super-Positioning of Voltage Sources for Fast Assessment of Wide-Area Thévenin Equivalents

Jakob Glarbo Møller, *Student Member, IEEE*, Hjörtur Jóhannsson, *Member, IEEE*,  
and Jacob Østergaard, *Senior Member, IEEE*

**Abstract**—A method for superimposing voltage sources is sought optimized by using a sparse triangular solver and multi-processing. A revision to the method is suggested which exploits Schur’s complement of the network admittance matrix and optimal re-use of computations. The algorithm is implemented and parallelized for shared memory multiprocessing. The proposed algorithm is tested on a collection of large test systems and performance is found to be significantly better than the reference method. The algorithm will thereby facilitate a speed-up of methods relying on Thévenin Equivalent representation such as the Thévenin equivalent method for contingency assessment.

**Index Terms**—Algorithms, Power system analysis computing, Thévenin equivalent

## I. INTRODUCTION

The work described in this paper is motivated by the recent development of a power system solver which relies on Thévenin equivalent representation of the network seen from each point of constant voltage. The method can be used to detect bifurcations that would render normal power flow methods non-convergent without giving further hints to the origin of the problem [1]. It is intended for analyzing contingencies in an online security assessment based on a steady-state hypothesis from a state estimator. Profiling the method has revealed that computation of Thévenin equivalents takes a significant share of the time spent by the algorithm [2]. This directed the author’s attention towards efficient algorithms for Thévenin equivalent computations. However, the field of application of Thévenin equivalents is broad and the work presented may inspire others working with different applications of the famous network equivalent.

Thévenin equivalent representation can be applied in determining critical limits to power transmission [3]. Steady-state stability of a generator may for example be expressed as a power versus angle relationship derived from a network equivalent formed on basis of Thévenin’s theorem [4].

Assessment of long-term voltage stability has also been suggested to use Thévenin equivalent representations [5].

In [6] it was demonstrated that Thévenin Equivalents effectively can be applied in assessment of aperiodic small-signal rotor angle stability. In [7] methods relying on the same system representation was applied in determining preventive actions necessary for stabilizing machines prone to aperiodic instability.

Authors are with Center for Electric Power and Energy, Technical University of Denmark. Contact: jglmo@elektro.dtu.dk

The SOSPO project has received funding from the The Danish Council for Strategic Research under grant agreement no. 11-116794, www.sospo.dk

Usage of PMU measurements for directly assessing Thévenin equivalents seen from a point of common connection has been suggested with the aim of supporting local control decisions [8]. If a central system operator is to predict how such local controllers will respond is it of vital importance to be able to assess those Thévenin equivalents at the control center.

A decomposition technique where nodes or branch elements are substituted by a Thévenin equivalent representing the remaining system has been suggested for parallel processing of power system simulations [9]. Another method for domain decomposition rely on network reduction by formulation of the Schur complement. With this method one can reduce variables inside sub-domains of a problem while preserving the relationship between interface variables acting on the global domain [10]. The Schur complement decomposition method has previously been promoted for parallelizing dynamic simulations on power systems [11].

Thévenin’s theorem was formulated independently by Hermann von Helmholtz in 1853 and Léon Charles Thévenin in 1883 [12]. It is apparent from the above examples that the system representation still has wide application in the field of power system analysis.

Efficient computation of Thévenin equivalents has been addressed in a number of references.

Hajj published in 1976 a method for computing Thévenin equivalents for multi-port networks which determines unknown port-variables from inverse solutions to a system of linear equations such as Kirchoff’s nodal equations [13].

Sommer and Jóhannsson demonstrated the efficiency of applying Schur’s complement in an algorithm for computing Thévenin impedances on large networks [14]. They develop a sparsity oriented left-looking algorithm and prove that the method is more efficient than previous inverse solvers for Thévenin equivalent computations.

Wang et al. presented method for super-imposing voltage and current sources to obtain Thévenin equivalents seen from a load bus [15]. They apply Schur’s complement for reducing the computational burden.

The problem of obtaining Thévenin equivalents seen from a generator has been addressed by Dmitrova et al. in [16]. The derivation departs in circuit theory and results in formulation of a set of coefficients describing the coupling between the Thévenin voltage seen from each generator and all voltage sources in the system. The method demonstrated good accuracy. However, no benchmark with respect to computational efficiency was provided.

This publication reviews the procedure suggested in [16] with respect to computational efficiency. Initial attempts on use of triangular factorization and multiprocessing are found to be inadequate for contingency assessment on large systems. A revision is therefore proposed on basis of Schur's complement. It is further demonstrated how the revised algorithm may be parallelized on a shared memory platform.

The efficiency of applying Schur's complement in obtaining Thévenin voltages seen from a voltage source node has, to the author's knowledge, not been investigated prior to this publication. Nor has it been demonstrated how such an approach can be adapted for multiprocessing. It is expected that the algorithm will be particularly applicable in circuits with a large fraction of nodes with automatic voltage control, such as a transmission network, or a distribution network with high shares of distributed generation.

Section II describes the method for superposition presented in [16]. A revision is then proposed which emphasizes reuse of computations and network reduction. In section III the revised algorithm is implemented for shared memory multiprocessing and tested with respect to scalability. IV gives perspectives on the work presented and V concludes the paper.

## II. ALGORITHM DEVELOPMENT

Thévenin equivalent consists of the Thévenin equivalent impedance  $Z_{th}$  and the Thévenin equivalent voltage  $V_{th}$  such that the Thévenin equivalent seen from a node  $i$  obeys:

$$V_{th,i} = V_i - Z_{th,i}I_i \quad (1)$$

Here  $V_i$  is the node voltage and  $I_i$  is the current injected at node  $i$ . A node may connect voltage sources, current sources and shunt impedances to the network. All loads are at first represented by their impedance and the network is energized by voltage sources only. Voltage sources in a power system could be terminals of generators with automatic voltage regulators or internal voltages of manually excited machines. An admittance matrix may be block-wise partitioned such that nodes with no sources ( $ns$ ) are assigned lower indices while nodes with voltage sources ( $vs$ ) are assigned higher indices.

$$\begin{bmatrix} 0 \\ I_{vs} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{ns} & \mathbf{Y}_{link} \\ \mathbf{Y}_{link}^T & \mathbf{Y}_{vs} \end{bmatrix} \begin{bmatrix} V_{ns} \\ V_{vs} \end{bmatrix} \quad (2)$$

As loads are represented by their admittance value no current is injected in  $ns$ -nodes. By itself  $\mathbf{Y}_{ns}$  represents admittances of a system where all voltage sources have been short-circuited. The Thévenin impedance seen from the  $vs$ -node  $i$  is obtained by injecting a unit test current at node  $i$  and taking the reciprocal of the resulting voltage at node  $i$ . This is achieved by solving the inverse problem in (3).

$$\begin{bmatrix} \mathbf{Y}_{ns} & \mathbf{Y}_{link(:,i)} \\ \mathbf{Y}_{link(i,:)} & \mathbf{Y}_{vc(i,i)} \end{bmatrix} \begin{bmatrix} \vdots \\ V_{i|I_i=1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (3)$$

$$Z_{th,i} = V_{i|I_i=1}^{-1} \quad (4)$$

The Thévenin equivalent voltage at a given node is composed of contributions from the voltage sources in the network. These contributions are scaled and rotated by a complex

coefficient and superimposed to form the Thévenin voltage. The coefficients were named grid transformation coefficients (**GTC**) by Dmitrova et al. in [16]. The coefficients satisfy:

$$V_{th,vs} = \mathbf{GTC}V_{vs} \quad (5)$$

The grid transformation coefficients may be obtained as the ratio between the open-circuit voltages  $V_{i|I_j=1}^{OC}$  and  $V_{j|I_j=1}$  at nodes  $i$  and  $j$  when a unit current is injected in node  $j$  and remaining voltage sources in the system are short-circuited.

$$V_{th,i} = \sum_{j \neq i} \frac{|vs| V_{i|I_j=1}^{OC}}{V_{j|I_j=1}} \cdot V_j \quad i, j \in vs \quad (6)$$

The Thévenin voltage seen from  $vs$ -node  $i$  is then obtained by scaling the contributions from all other  $vs$ -node voltages  $j \in vs$  by such a ratio and adding them together. The open-circuit voltages  $V_{i|I_j=1}^{OC}$  used to compute the coefficients mentioned above are found by solving the inverse problem in (7) for all combinations of  $vs$ -nodes  $i$  and  $j$ .

$$\begin{bmatrix} \mathbf{Y}_{ns} & \mathbf{Y}_{link(:,i)} & \mathbf{Y}_{link(:,j)} \\ \mathbf{Y}_{link(i,:)} & \mathbf{Y}_{vs(i,i)} & \mathbf{Y}_{vs(i,j)} \\ \mathbf{Y}_{link(j,:)} & \mathbf{Y}_{vs(j,i)} & \mathbf{Y}_{vs(j,j)} \end{bmatrix} \begin{bmatrix} \vdots \\ V_{i|I_j=1}^{OC} \\ V_{j|I_j=1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

### A. Re-use of triangular factorization

Taking advantage of the fact that the block matrix  $\mathbf{Y}_{ns}$  is the same in all cases allow a significant problem reduction [14]. This is achieved by conducting only a single  $LU$ -decomposition such that  $\mathbf{Y}_{ns} = \mathbf{L}_{ns}\mathbf{U}_{ns}$  and employing a left-looking algorithm for decomposition of the remaining rows and columns of (3). By this procedure the column and row that should be appended to  $\mathbf{U}_{ns}$  and  $\mathbf{L}_{ns}$  respectively to form the factorization used in solving (3) are found by:

$$\hat{\mathbf{U}}_{(:,i)} = \mathbf{L}_{ns}^{-1}\mathbf{Y}_{link(:,i)} \quad (8)$$

$$\hat{\mathbf{L}}_{(:,i)}^T = \mathbf{U}_{ns}^{-1}\mathbf{Y}_{link(i,:)}^T \quad (9)$$

The Thévenin impedance seen from  $i$  is the inverse of the  $i$ th diagonal element in the Schur complement  $\mathbf{Y}_{eq} = \mathbf{Y}/\mathbf{Y}_{ns}$  of an admittance matrix where  $ns$ -nodes are eliminated [14]. If  $\hat{\mathbf{L}}$  is unit diagonal this equivalent admittance is found as:

$$Z_{th,i}^{-1} = \hat{\mathbf{U}}_{(i,i)} = \mathbf{Y}_{vs(i,i)} - \hat{\mathbf{L}}_{(i,:)}\hat{\mathbf{U}}_{(:,i)} \quad (10)$$

The grid transformation coefficients can be obtained naively by solving the inverse problem of (7) for each of the  $(|vs|-1)^2$  entries in **GTC**. Or one may use the same approach as in (10) for obtaining  $\hat{\mathbf{U}}_{j,j}$  and the sub-sequent  $\hat{\mathbf{U}}_{j,i}$  and  $\hat{\mathbf{U}}_{i,j}$  needed to represent the factorization of the admittance matrix in (7).

$$V_{i|I_j=1}^{OC} = -\frac{\hat{\mathbf{U}}_{(j,i)}}{\hat{\mathbf{U}}_{(j,j)}}\hat{\mathbf{U}}_{(i,i)}^{-1}$$

$$V_{j|I_j=1} = \hat{\mathbf{U}}_{(j,j)}^{-1} \Rightarrow$$

$$\mathbf{GTC}(j,i) = \hat{\mathbf{U}}_{(j,i)} \cdot Z_{th,i}$$

This procedure is followed in algorithm 1. The outer for-loop determines the Thévenin impedance using the same approach as in [14]. The inner for-loop determines the coupling between open-circuit voltages and current injections. Algorithm 1 takes as input the block partitioned admittance matrix given in (2), the number of nodes  $n$  and the number of  $vs$ -nodes  $k$ .

---

**noresize 1** Obtain Thévenin equivalents
 

---

$\mathbf{L}_{ns}, \mathbf{U}_{ns} \leftarrow$  factorization of  $(\mathbf{Y}_{ns})$

**for** each  $vs$ -node  $i$  **do**

$\hat{U}_{i(:,i)} \leftarrow$  solve( $\mathbf{L}_{ns}, Y_{Link(:,i)}$ )

$\hat{L}_{(:,i)}^T \leftarrow$  solve( $\mathbf{U}_{ns}^T, Y_{Link(i,:)}^T$ )

$\hat{U}_{(i,i)} \leftarrow Y_{vs(i,i)} - \hat{L}_{(i,:)} \hat{U}_{(:,i)}$

$Z_{th,(i)} \leftarrow \hat{U}_{(i,i)}^{-1}$

**for** each remaining  $vs$ -node  $j \neq i$  **do**

$\hat{U}_{i,j} \leftarrow$  solve  $\left( \left[ \begin{array}{c|c} \mathbf{L}_{ns} & \mathbf{0} \\ \hline \hat{L}_{(i,:)} & 1 \end{array} \right], \left[ \begin{array}{c} Y_{Link(:,j)} \\ Y_{Link(i,j)} \end{array} \right] \right)$

$\mathbf{GTC}(i,j) \leftarrow -\hat{U}_{i,j(n-k+1)} Z_{th,i}$

**end for**

**end for**

**return**  $Z_{th}$  and  $\mathbf{GTC}$

---

### B. Parallelization of Algorithm 1

A test was conducted in Matlab on the PTI-WECC-1648 test system seen in table I. Algorithm 1 determines Thévenin equivalents for all voltage sources in 11.0s. Such execution time is not satisfactory for contingency assessment where many scenarios must be analysed. Therefore the benefit of multiprocessing is investigated. Profiling the algorithm show that 2% of the runtime is spent on  $LU$ -decomposition. Thus 98% parallelism may be achieved by solving the two nested for-loops for smaller chunks of the  $vs$ -nodes in parallel. Results of the parallelization is shown in figure 1. It is found that the algorithm scales well on up to 12 processors at which point the increase in communication overhead seems to dominate the advantage of adding more processors.  $\mathbf{L}_{ns}$ ,  $\mathbf{U}_{ns}$  and  $\mathbf{Y}_{link}$  must be accessible in their entirety for all processes. The cluster at which the algorithm is tested is composed of nodes of 2 dual core CPUs which means that the matrices must be copied to a new memory location for every 4 processors. This might explain why the speed-up saturates around 12 processors for the given problem. The wall-time passed during execution on 12 processors is 1.2s which is still not quite satisfactory if the equivalents are to be used in contingency assessment.

### C. Schur's complement and Thévenin equivalents

Performance of algorithm 1 is dissatisfying, but it turns out that there is potential for improvement. It is found that the problem  $\mathbf{L}_{ns}^{-1} Y_{link(:,i)}$  is solved  $|vs|$  times when solving for  $\hat{U}_i$  and  $\hat{U}_{j,i}$  for all  $j$  where only the last element is different from  $\hat{U}_i$ . The benefit of re-using  $\hat{U}_i$  is therefore a performance enhancement of an order of magnitude.

Re-using computations and vectorisation of for-loops leads to a revised expression for the  $\mathbf{GTC}$ -matrix:

$$\mathbf{GTC} = \mathcal{I} - \mathcal{D}(Z_{th}) \mathbf{Y}_{eq} \quad (11)$$

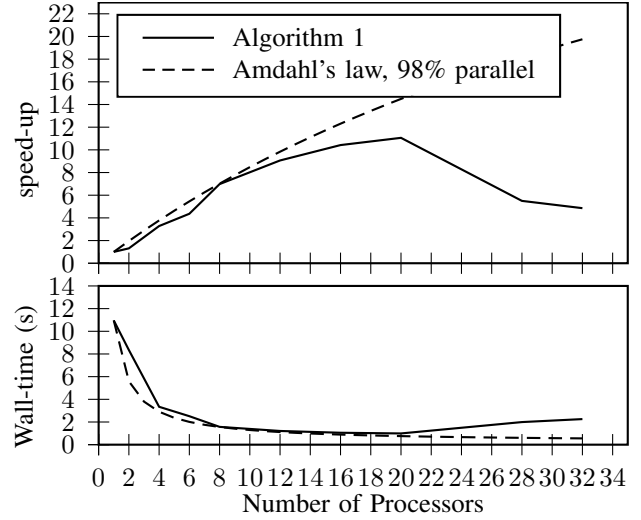


Fig. 1. Strong scaling of algorithm 1 compared to Amdahl's law for a 98% parallel algorithm. For the given problem size the proposed algorithm scales well on up to 12 processors. CPU: Intel Xeon X5550 Dual Core @ 2.66 GHz

Here  $\mathcal{I}$  is the identity matrix,  $\mathcal{D}(Z_{th})$  is a diagonal matrix with Thévenin impedances and  $\mathbf{Y}_{eq}$  is the Schur complement of an admittance matrix where  $ns$ -nodes are eliminated. The validity the above equation is proven in the following.

It will here be demonstrated how the Schur complement is related to computations of Thévenin equivalents. The starting point is the partitioned network from (2). For the sake of generality loads are represented by a current source  $cs$  injecting a current of arbitrary direction in the network.

$$\begin{bmatrix} I_{cs} \\ I_{vs} \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_{cs} & \mathbf{Y}_{Link} \\ \mathbf{Y}_{link}^T & \mathbf{Y}_{vs} \end{bmatrix} \begin{bmatrix} V_{cs} \\ V_{vs} \end{bmatrix} \quad (12)$$

Eliminating  $V_{cs}$  from (12) yields;

$$I_{vs} + \mathbf{Q}_{ac} \cdot I_{cs} = \mathbf{Y}_{eq} V_{vs} \quad (13)$$

$$\mathbf{Y}_{eq} = \mathbf{Y}_{vs} - \mathbf{Y}_{link}^T \mathbf{Y}_{cs}^{-1} \mathbf{Y}_{link} \quad (14)$$

$$\mathbf{Q}_{ac} = -\mathbf{Y}_{link}^T \mathbf{Y}_{cs}^{-1} \quad (15)$$

$\mathbf{Y}_{eq}$  is called the Schur complement matrix and  $\mathbf{Q}_{ac}$  is the accompanying matrix [17].

The Thévenin impedance seen from a node  $i$  is the impedance to be measured when all voltage sources are short circuited and all current sources are open-circuited. For the impedance matrix  $\mathbf{Z}_{cs} = \mathbf{Y}_{cs}^{-1}$  this translates to the following expressions:

$$Z_{th,i} = \begin{cases} \mathbf{Z}_{cs,(i,i)} & \text{for } i \in cs \\ \mathbf{Y}_{eq}^{-1}(i,i) & \text{for } i \in vs \end{cases} \quad (16)$$

With the definition of Thévenin voltage given in (6) the Thévenin voltage at  $vs$ -nodes may be expressed as;

$$\begin{aligned} V_{th,vs} &= V_{vs} - Z_{th,vs} I_{vs} \\ &= V_{vs} - \mathcal{D}(Z_{th,vs})(\mathbf{Y}_{eq} V_{vs} - \mathbf{Q}_{ac} I_{cs}) \Rightarrow \end{aligned}$$

$$V_{th,vs} = (\mathcal{I} - \mathcal{D}(Z_{th,vs}) \mathbf{Y}_{eq}) \cdot V_{vs} + \mathcal{D}(Z_{th,vs}) \mathbf{Q}_{ac} \cdot I_{cs} \quad (17)$$

TABLE I  
TESTSYSTEM DATA

Case	no. of buses	no. of voltage sources	non-zeros in $Y$	non-zeros in $GTC$
Nordic32	46	20	160	126
Pegase1354	1354	260	4774	29260
PTI-WECC-1648	1648	313	6680	56791
Polish-Summer	2737	193	9263	31314
Polish-Winter	2746	382	9292	64174
PTI-EECC-7917	7917	1325	32211	1132107
Pegase9241	9241	1445	37655	533487

Similar relation can be obtained for the  $cs$ -nodes:

$$\begin{aligned} V_{th,cs} &= V_{cs} - Z_{th,cs} I_{cs} \\ &= -\mathbf{Y}_{cs}^{-1} \mathbf{Y}_{link} V_{vs} + \mathbf{Y}_{cs}^{-1} I_{cs} - \mathcal{D}(Z_{th,cs}) I_{cs} \Rightarrow \end{aligned}$$

$$V_{th,cs} = -\mathbf{Z}_{cs} \mathbf{Y}_{link} V_{vs} + (\mathbf{Z}_{cs} - \mathcal{D}(Z_{th,cs})) I_{cs} \quad (18)$$

The expression for  $cs$ -nodes in (18) is equivalent to an expression used for assessing voltage stability in [15].

In [16] loads were represented by their admittance value. Under such system representation the current injection at a load bus is zero and (17) turns into the expression stated in (11).

### III. IMPLEMENTATION AND TEST OF REVISED ALGORITHM

Equation (11) is implemented in algorithm 2. The test systems used are shown in table I. The Pegase and Polish systems can be found in Matpower 5.1 [18]. The PTI systems are included among the PSS/E 33.0 examples.

The algorithm is implemented in C++ with SuiteSparse [19].  $\mathbf{Y}_{ns}$  is ordered to an approximate minimum degree (AMD) prior to  $LU$ -decomposition. The triangular solves are implemented with a sparse solver to exploit the small number of non-zero entries in  $\mathbf{Y}_{link}$  and  $\mathbf{Y}_{link}^T$ . Wall-time spent by executing algorithm 2 on test systems of various sizes are shown in Fig. 3. The times stated include time spent on AMD ordering and  $LU$ -decomposition of  $\mathbf{Y}_{ns}$ . The performance enhancement introduced by revising the algorithm is substantial. The 1648 bus WECC test system which took algorithm 1 11.0s to solve is solved in just 25ms by algorithm 2.

---

#### noresize 2 Schur complement Thévenin equivalent

---

```

 $\mathbf{L}_{ns}, \mathbf{U}_{ns} \leftarrow$  factorization of  $(\mathbf{Y}_{ns})$ 
 $\hat{\mathbf{U}} \leftarrow$  solve $(\mathbf{L}_{ns}, \mathbf{Y}_{link})$ 
 $\hat{\mathbf{L}}^T \leftarrow$  solve $(\mathbf{U}_{ns}^T, \mathbf{Y}_{link}^T)$ 
 $\mathbf{Y}_{eq} \leftarrow \mathbf{Y}_{vs} - \hat{\mathbf{L}}\hat{\mathbf{U}}$ 
 $Z_{th} \leftarrow \mathcal{D}(\mathbf{Y}_{eq})^{-1}$ 
 $GTC \leftarrow \mathcal{I} - \mathcal{D}(Z_{th})\mathbf{Y}_{eq}$ 
return  $Z_{th}$  and  $GTC$ 

```

---

The precision of algorithm 2 is evaluated by comparing the resulting Thévenin voltages from algorithms 1 and 2 respectively. The error is stated in table II in terms of sum of squared

TABLE II  
RESIDUALS AND LARGEST VECTOR ERROR FOR THÉVENIN VOLTAGES COMPUTED WITH ALGORITHM 2 COMPARED TO ALGORITHM 1

Case	$\sum r^2$	$\ TVE\ _\infty$
Nordic32	$1.88 \cdot 10^{-30}$	$1.06 \cdot 10^{-13}$
Pegase-1354	$3.56 \cdot 10^{-26}$	$1.25 \cdot 10^{-11}$
PTI-WECC-1648	$9.80 \cdot 10^{-27}$	$6.34 \cdot 10^{-12}$
Polish-Summer	$2.11 \cdot 10^{-26}$	$6.06 \cdot 10^{-12}$
Polish-Winter	$9.56 \cdot 10^{-27}$	$2.93 \cdot 10^{-12}$
PTI-EECC-7917	$1.07 \cdot 10^{-25}$	$2.04 \cdot 10^{-11}$
Pegase-9241	$5.94 \cdot 10^{-25}$	$2.72 \cdot 10^{-11}$

TABLE III  
PARTIAL EXECUTION TIMES FOR POLISH SUMMER AND WINTER CASES

Case	Time on LU-factorization	Time on solve and multiply
Polish-Summer	2.5ms	17.2ms
Polish-Winter	1.7ms	27.9ms

residuals and as the single largest total vector error (TVE%) as defined in IEEE C37.118. On basis of these results algorithm 2 is found to have good precision.

It is relevant to investigate how the algorithm performs under different loading scenarios. It has therefore been attempted conduct a study on two scenarios based on the Polish system under winter peak load of 25GW and summer off-peak load of 11GW. The difference in execution times between the two cases could indicate a correlation between system load and execution time. From table III it is apparent that it is more costly to perform the triangular solves and matrix multiplications for the highly loaded case. However, referring to table I it is apparent that this case has twice as many nodes with active voltage control and twice as many non-zero entries in the  $GTC$ -matrix. Thus, the algorithm computes twice as many vector products for the Polish-Winter case than the Polish-Summer case. On this basis it is not possible to identify a correlation between the system loading and the execution time. In fact the precision stated in table II is better for the winter case than the summer case.

Though algorithm 2 performs much better than algorithm 1 the execution time from figure 3 shows a polynomial relation to the system size. Thus, slightly larger systems may exhaust the algorithm substantially. It is investigated how multiprocessing may assist in solving this problem.

#### A. Parallelization of Algorithm 2

The algorithm is parallelized for shared memory processing using OpenMP. Each thread solves a chunk of the triangular problems  $\hat{\mathbf{U}} = \mathbf{L}_{ns}^{-1} \mathbf{Y}_{link}$  and  $\hat{\mathbf{L}}^T = \mathbf{U}_{ns}^T \mathbf{Y}_{link}^T$ . The matrix products needed to determine  $\mathbf{Y}_{eq} = \mathbf{Y}_{vs} - \hat{\mathbf{L}}\hat{\mathbf{U}}$  are implemented as a sparse implementation of Fox's algorithm where each chunk of  $\hat{\mathbf{L}}$  is passed from thread to thread until all sub-matrices of  $\mathbf{Y}_{eq}$  are computed [20]. The layout of the parallel algorithm 2 is presented in figure 2 where 2.a shows the triangular solves and 2.b shows Fox's algorithm applied from the viewpoint of a thread  $j$ .

The parallel implementation of algorithm 2 is accessible online<sup>1</sup>.

The scaling study shows that multiprocessing can be applied to achieve a speed-up of Algorithm 2. However, the speed-up is determined by the size of the problem. Fig. 3 shows the correlation between wall-time spent on Algorithm 2 and the number of buses in the test systems. Speed-up, computed as the execution time on one thread per execution time on  $n$  threads, is given in figure 4. In most cases the speed-up is about 2 for a cluster of 4 processors. Thus the multiprocessing efficiency is about 50%. The PTI-EECC-7941 shows a greater speed-up. This is likely due to the denser **GTC**-matrix. The Nordic32 system is simply too small to benefit from multiprocessing. In fact the execution time increases when more processors are added to the job due to the increase in communication overhead.

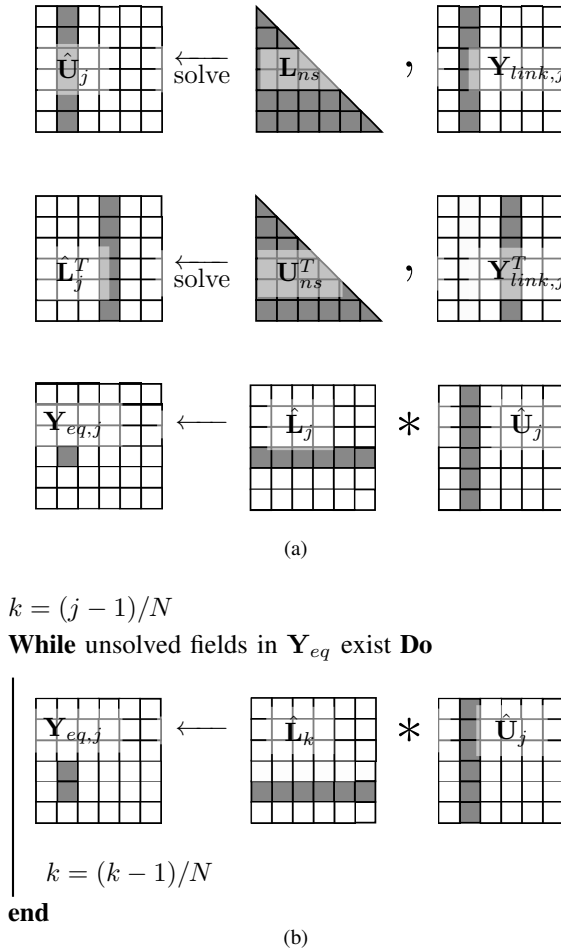


Fig. 2. Parallelization of algorithm 2 seen from processor  $j \in N$ . (a) Triangular solves are done in chunks. (b) Chunks of results are multiplied by Fox's algorithm

#### IV. PERSPECTIVES OF THE PROPOSED ALGORITHM

Together (17) and (18) form a coherent theorem of the relationship between Thévenin equivalents and Schur complements.

<sup>1</sup>[https://github.com/jakobglarbomoller/thevenin\\_equivalents](https://github.com/jakobglarbomoller/thevenin_equivalents)

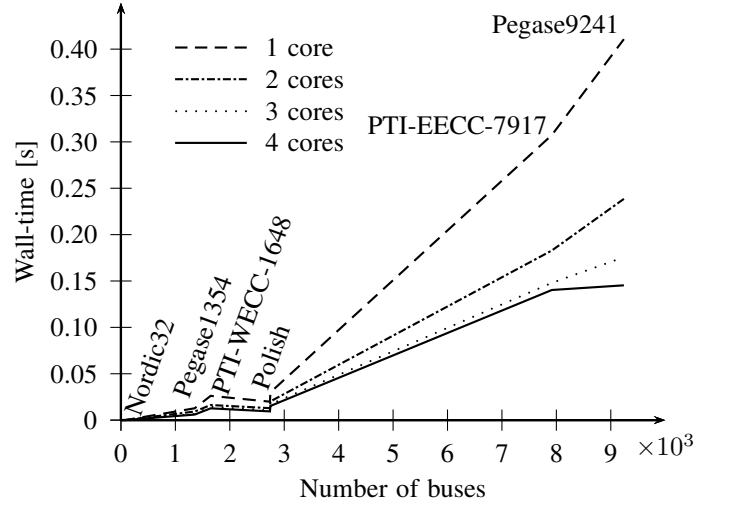


Fig. 3. Performance of modified **GTC**-algorithm. Wall-time in seconds versus system size. CPU: Intel(R) Core(TM) i7-3630QM CPU @ 2.40GHz

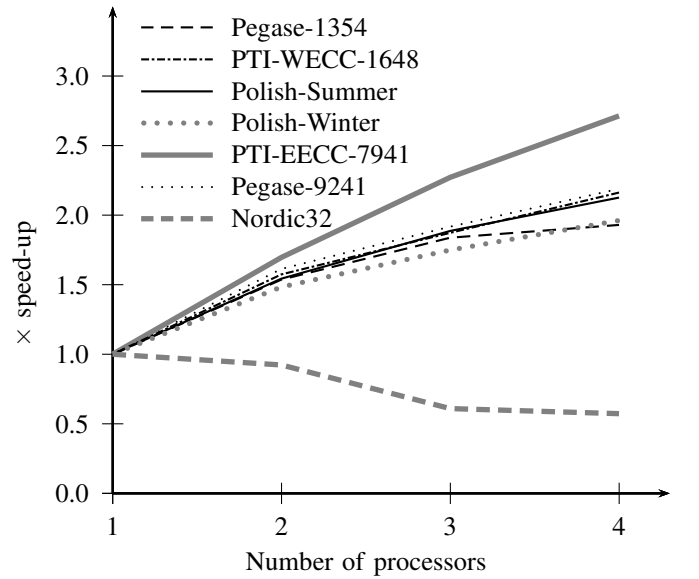


Fig. 4. Speed-up of algorithm 2 per number of processors. The speed-up is execution time on 1 core per execution time on  $N$  cores

The relationship between the Schur complement and Thévenin equivalents was used to obtain a simple and efficient expression of the factors, with which each voltage source in a network contributes to the Thévenin voltages. The proposed algorithm proved to be quite fast. But there may yet be potential for optimizing the computation of Thévenin voltages. It is the matrix multiplication  $\hat{\mathbf{L}} \cdot \hat{\mathbf{U}}$  which accounts for the greater share of execution time. The admittance matrix is sparse but the same is not given for  $\mathbf{Y}_{eq}$  and  $\mathbf{Q}_{ac}$ . One may note the difference between non-zero elements in **GTC** and **Y** provided in table I. It is further noticed in figure 5 that

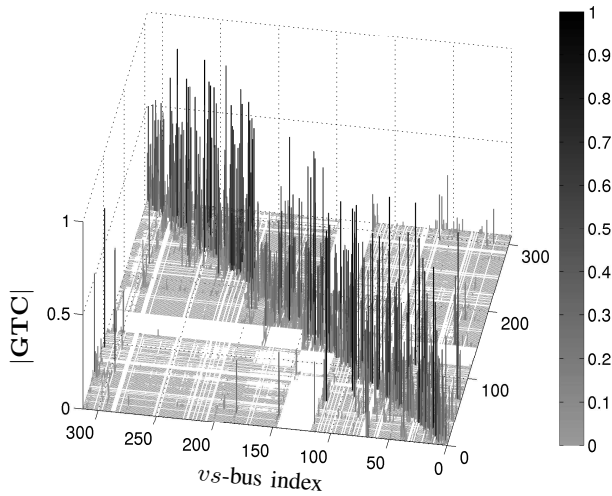


Fig. 5. Magnitudes of the 56,791 nonzero entries in  $\mathbf{GTC}$ -matrix for the PTI-WECC-1648 test system. Many of the nonzero entries appear to have insignificant influence on the Thévenin voltage.

many fill-ins seem quite insignificant and might be omitted by setting a tolerance on the triangular solves. This would however influence the precision of the resulting Thévenin equivalent voltages.

Representing loads by their admittance value yield a simpler expression for the Thévenin voltage. However, power system loads can vary a lot which would require the Thévenin impedance and  $\mathbf{GTC}$ -matrix to be recalculated. For the expression given in (17)  $\mathbf{Y}_{eq}$  and  $\mathbf{Q}_{ac}$  are dependent only on changes in network topology. Using current source representation of loads should therefore reduce the computations needed to account for load variations.

This might be used in contingency assessments as described in [1] where Thévenin equivalents are used for assessing steady-state stability of generators in a power system. The algorithm can then be applied for all contingencies which changes the bus admittance matrix or the set of nodes with constant voltage.

## V. CONCLUSION

An algorithm for superimposing voltage sources and computing Thévenin equivalents seen from all voltage controlled nodes in a network was evaluated and a revision has been proposed which shows better performance.

Attempts were made to enhance performance of an existing algorithm by using a sparse triangular solver and multiprocessing. Yet, the resulting implementation was found to be too time consuming for the intended application in on-line contingency assessment.

A revision of the algorithm was suggested on basis of prior art in computing Thévenin equivalents seen from a load bus. The revised algorithm exploits a close relation between super-positioning of voltage sources and Schur's complement

of the network admittance matrix. The revision provided an efficient and precise algorithm which enables wide area Thévenin equivalents to be obtained for systems of thousands of buses in fractions of a second. Benchmarking the revised algorithm show a polynomial relation between execution time and problem size. It is therefore demonstrated how additional speed-up may be achieved by means of multiprocessing.

## REFERENCES

- [1] J. Møller, H. Jóhannsson, and J. Østergaard, "Thevenin equivalent method for dynamic contingency assessment," *Proceedings of Power & Energy Society's General Meeting*, 2015.
- [2] J. Møller, H. Jóhannsson, and J. Østergaard, "Computation of steady state nodal voltages for fast security assessment in power systems," *Proceedings of Power Quality and Reliability Conference, (PQ)*, 2014.
- [3] H. Jóhannsson, J. Østergaard, and A. H. Nielsen, "Identification of critical transmission limits in injection impedance plane," *International Journal of Electrical Power & Energy Systems*, vol. 43, no. 1, pp. 433–443, 2012.
- [4] P. Kundur, *Power System Stability and Control*. Mc. Graw-Hill, 1991.
- [5] A. Perez, H. Jóhannsson, and J. Østergaard, "Evaluation of enhancements to thevenin equivalent based methods for real-time voltage stability assessments," *Proceedings of the 5th IEEE PES Innovative Smart Grid Technologies (ISGT) European Conference*, 2014.
- [6] H. Jóhannsson, A. H. Nielsen, and J. Østergaard, "Wide-area assessment of aperiodic small signal rotor angle stability in real-time," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4545–4557, 2013.
- [7] E. Dmitrova, M. L. Wittrock, H. Jóhannsson, and A. H. Nielsen, "Early prevention method for power system instability," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1784–1792, 2015.
- [8] S. M. Abdelkader and D. J. Morrow, "Online tracking of thevenin equivalent parameters using pmu measurements," *IEEE Transactions on Power Systems*, vol. 27, 2012.
- [9] M. A. Tomim, J. R. Marti, T. D. Rybel, and et al., "Mate network tearing techniques for multiprocessor solution of large power networks," *Proceedings of IEEE PES General Meeting*, 2010.
- [10] F. Zhang, *The Schur Complement and Its Applications*. Springer, 2004.
- [11] P. Aristidou, D. Fabozzi, and T. V. Cutsem, "Dynamic simulation of large-scale power systems using a parallel schur-complement-based decomposition method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2561–2570, 2014.
- [12] J. Brittain, "Back to basics; thevenin's theorem," *IEEE Spectrum*, vol. 27, no. 3, 1990.
- [13] I. N. Hajj, "Computation of thevenin and norton equivalents," *IEEE Transactions on Power Systems*, vol. 12, no. 11, pp. 273–274, 1976.
- [14] S. Sommer and H. Jóhannsson, "Real-time thevenin impedance computation," *Proceedings of IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, 2013.
- [15] Y. Wang, I. R. Pordanjani, W. Li, W. Xu, T. Chen, E. Vaahedi, and J. Gurney, "Voltage stability monitoring based on the concept of coupled single-port circuit," *IEEE Transactions on Power Systems*, vol. 26, pp. 2154–2163, 2011.
- [16] E. Dmitrova, H. Jóhannsson, and A. H. Nielsen, "Assessment of the impact that individual voltage source has on a generator's stability," *Proceedings of 10th International Power and Energy Conference, (IPEC)*, 2012.
- [17] F. Dörfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Transactions on Circuits and Systems - I: Regular Papers*, vol. 60, no. 1, pp. 150–163, 2013.
- [18] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive and corrective actions," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4909–4917, 2013.
- [19] T. A. Davis, *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006.
- [20] G. Fox and S. W. Otto, "Matrix algorithms on a hypercube i: Matrix multiplication," *Parallel Computing*, no. 4, 1987.