

## SIMULATION OF A BALANCED LOW-VOLTAGE ELECTRICAL GRID USING A SIMPLIFIED NETWORK MODEL

MÁRTON GREBER <sup>\*1</sup> AND ATTILA FODOR<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering and Information Systems, Faculty of Information Technology, University of Pannonia, Egyetem u. 10., Veszprém, H-8200, HUNGARY

A simulation method for low-voltage balanced distribution networks is proposed in this article. The novel method of node powers is based on the general calculation technique of node voltages. By researching only balanced networks, single-phase equivalents of the three-phase system are applicable. For the description of power lines, various parameters and matrices are available. In this work a simplified model is applied by using a purely resistive one. The active power results are solved through an iterative process. A main accomplishment is that the number of iterations needed is independent of the size of the network, and the process rapidly converges. Validation of the method is performed on the IEEE European Low-Voltage Test Feeder network. The simulation results confirm the achievements described in this paper.

**Keywords:** low-voltage distribution system, grid simulation, smart grid, method of node voltages, IEEE European Low-Voltage Test Feeder

### 1. Introduction

In recent years a lot of research has been conducted and progress made in the field of smart grid applications. Therefore, the demand for cloud-based systems with integrated simulation capabilities has increased. The calculation of the voltages, currents and powers of the components of the electrical (smart) grid is not easily achieved. This has resulted in the development of custom calculation methods which have the benefit of being fine-tuned for a particular application.

One of the fundamental network calculation methods - in a general sense - is the node voltages method. From a mathematical perspective, this method is based on solving a system of linear equations. As a result, this method can be implemented through various frameworks. One proposed solution revolves around using an open source discrete event simulator called OMNeT++ [1]. The models for electrical components need to be constructed and the simulation is conducted via message handling. Another approach used Coloured Petri nets to model electrical networks [2]. A network model needs to be constructed for the simulation, which consists of a propagation process. A solution is presented for basic network types, while complex ones are calculated through decomposition. Both of these methods offer solutions but the method of node voltages regards currents as an input.

The more common approach to calculations of distribution systems calculation is via the method of power

flow. This uses complex numbers to distinguish between active and reactive power. For the given nodes, both active and reactive power, voltage and phase angle are required to formulate the solution [3]. This poses a non-linear problem, furthermore, the system of equations consists of real and imaginary subsets. Over the years, several pieces of research have dealt with this subject and the method known as DC power flow developed. By restricting the parameters the calculation was simplified, namely the voltage angles as well as generated and consumed active powers. This method assumes small differences in voltage angle and lossless lines [4]. The biggest downside of the method is that it cannot be used to calculate line losses because of the assumptions, although efforts are being made to overcome this obstacle [5].

By taking into account the extent of a given power system, different parameters of the power line model become dominant [6]. In low-voltage systems the distance between adjacent nodes is smaller than in high-voltage systems. By using this information a new set of restrictions is proposed, inspired by the DC power flow. It can be regarded as a complementary method since the line reactances are neglected. Utilising a resistive transmission line model offers the possibility to calculate distribution losses. This method has been developed from the method of node voltages.

In this article the issues of harmonic currents [7] and unbalanced networks [8] are not examined, it is assumed that the currents as well as voltages are sinusoidal and the grid is balanced with balanced three-phase loads. The

\*Correspondence: [greber.marton@virt.uni-pannon.hu](mailto:greber.marton@virt.uni-pannon.hu)

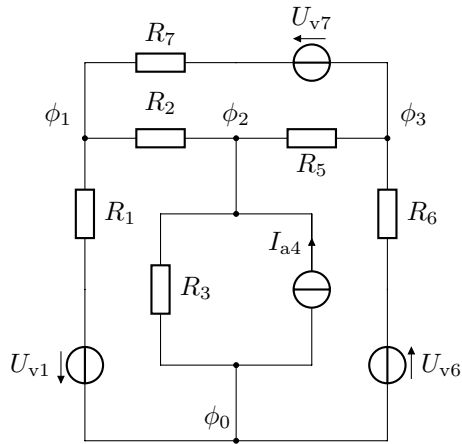


Figure 1: Example network

neutral wire can be omitted by restricting the set of networks in balanced low-voltage systems. This enables the application of a one-line equivalent circuit.

## 2. Method of node voltages

In distribution network calculations the main emphasis is on power flow. That means the voltages and angles are calculable when the generated and consumed powers are given [9]. The known formula for calculating the active power in a DC network is the following:

$$P = UI, \quad (1)$$

but the observed systems operate on alternating currents, hence the notion of AC power needs to be introduced [10]. Since voltages and currents are time-varying quantities, these are expressed as complex numbers, namely  $\bar{U}$  and  $\bar{I}$ . The complex power is calculated by multiplying the conjugate of  $\bar{I}$  by the voltage

$$\bar{S} = \bar{U} \bar{I}^*. \quad (2)$$

The phase angle of the complex power is defined by the difference between the angles of voltage and current:

$$\varphi = \arg(\bar{U}) - \arg(\bar{I}). \quad (3)$$

By observing the real and imaginary parts of the complex number, the active ( $P$ ) and reactive powers ( $Q$ ) are obtained:

$$P = \text{Re}(\bar{S}), \quad Q = \text{Im}(\bar{S}). \quad (4)$$

A known method for the analysis of electrical networks is the method of node voltages. The component values, e.g. the sources of resistance, voltage and current are given, the unknown variables are the node voltages. The calculation steps are best explained through an example network as shown in Fig. 1.

One key component of the process is the definition of a directed graph for the example network shown in Fig. 2. The reference directions in the graph are arbitrary.

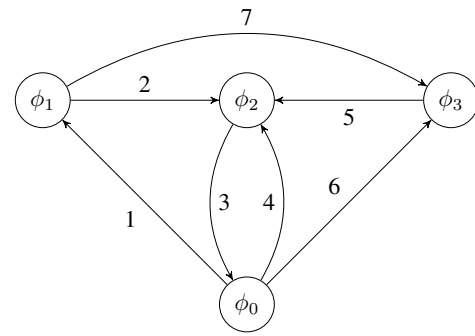


Figure 2: The reference directed graph

By using the graph in addition to Kirchoff's first law, the nodal equations can be written in the following form:

$$\begin{cases} I_1 - I_2 - I_7 = 0 \\ I_2 - I_3 + I_4 + I_5 = 0 \\ I_7 - I_5 + I_6 = 0 \end{cases} \quad (5)$$

After that, Ohm's law is used to express the edge currents with regard to the nodal voltages:

$$\begin{cases} I_1 = -G_1(\phi_1 - U_{v1}) \\ I_2 = G_2(\phi_1 - \phi_2) \\ I_3 = G_3\phi_2 \\ I_4 = I_{a4} \\ I_5 = G_5(\phi_3 - \phi_2) \\ I_6 = -G_6(\phi_3 + U_{v6}) \\ I_7 = G_7(\phi_1 + U_{v7} - \phi_3) \end{cases} \quad (6)$$

where the conductance of the arm of the network is denoted by  $G$ . If these steps are followed, the system of equations must be rearranged. The final form can be obtained by substituting the currents into Eq. (5) and rearranging it to determine the nodal voltages [11]:

$$\phi = G_e^{-1} I_e \quad (7)$$

The notation represents the dimensions of the elements it contains: the node voltage vector is obviously denoted by  $\phi$ , the nodal admittance matrix by  $G_e$  and the excitation vector by  $I_e$ . At first sight, this equation does not account for the voltage sources in the branches. Since these elements can be described in terms of current dimensions, the excitation vector takes the following form:

$$I_e = \begin{bmatrix} -G_1 U_{v1} + G_7 U_{v7} \\ -I_{a4} \\ -G_7 U_{v7} + G_6 U_{v6} \end{bmatrix}. \quad (8)$$

### 2.1 The generalized nodal equations

The above-mentioned method describes the working principle of the technique, but it is not suitable for algorithmic applications. A generalized approach is needed. One has to define column vectors for the voltage sources:

$$U_v^T = [u_{v1} \quad u_{v2} \quad \dots \quad u_{vn}], \quad (9)$$

current sources:

$$\mathbf{I}_a^\top = [i_{a1} \quad i_{a2} \quad \dots \quad i_{an}], \quad (10)$$

and admittances:

$$\mathbf{Y}^\top = [y_1 \quad y_2 \quad \dots \quad y_n], \quad (11)$$

where  $\mathbf{Y}$  refers to the complex conductance known as the admittance. Each scalar of the vectors contains the aforementioned properties of a particular edge. For the representation of the graph, an incidence matrix ( $\mathbf{A}$ ) is used of  $(m-1) \times n$  dimensions, and describes connections between nodes and edges:

$$a_{i,j} = \begin{cases} 0, & \text{if } j \text{ is not connected to } i \\ 1, & \text{if } j \text{ is pointing away from } i \\ -1, & \text{if } j \text{ is pointing towards } i \end{cases}, \quad (12)$$

where  $i = 1, 2, \dots, m-1$  represents the nodes of the graph and  $j = 1, 2, \dots, n$  denotes the edge index set. As with the direction of the graph, the reference of  $\mathbf{A}$  is also arbitrary, it is only manifested when multiplied by  $(-1)$ . These matrices can be constructed algorithmically and facilitate the use of the following equation [12]:

$$\phi = \mathbf{Y}_A^{-1} \mathbf{A} (\text{diag}(\mathbf{Y}) \mathbf{U}_v - \mathbf{I}_a), \quad (13)$$

where:

$$\mathbf{Y}_A = \mathbf{A} \text{diag}(\mathbf{Y}) \mathbf{A}^\top. \quad (14)$$

A method which can be used to compute node voltages by defining a graph represented by the incidence matrix that includes the electrical properties of the edges is proposed as a result.

### 3. Network model

In order to reduce the amount of computational power that is needed for simulations, a simplified network model is used. The feeder points of low-voltage grids are transformer substations which convert forward the desired power from the medium voltage side into the base voltage, therefore, can be represented as voltage sources. Power flows from the feeder points to the customers via transmission lines. Compared to medium- or high-voltage lines, the length between consecutive nodes is smaller. As a result, these can be modelled as series resistances. Customers are represented as current sources with consumer references of course.

The computationally demanding part of the method of node voltages is to invert  $\mathbf{Y}_A$ . In the case of complex analyses, the system of equations is separated into real and imaginary parts. Through one iteration cycle, two inverse calculations are needed. If voltages and currents are calculated as root mean square (RMS) values, the simulation method only requires real numbers. Therefore, the effort and time to calculate one cycle is halved.

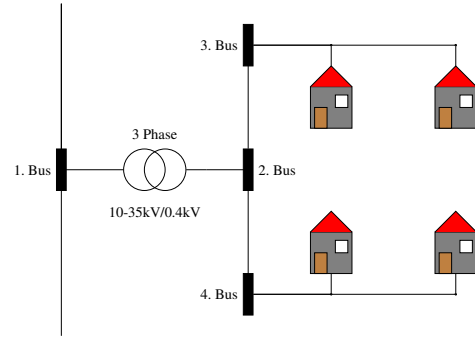


Figure 3: Low-voltage distribution system

#### 3.1 Topology verification

In the structure of a low-voltage distribution network, some rules are noticeable. The transmission line forms a power rail to which the consumers can connect, as is shown in Fig. 3. These mainly consist of households with single-phase connections, i.e. one phase and the neutral wire are used [13].

The method of node voltages can be applied to general circuits, on the other hand, the proposed method of node powers can be applied to distribution networks. These form a tighter set, therefore, the topology needs to be checked to ensure it works properly.

If a network contains  $m$  nodes,  $m$  node equations can be obtained. On the other hand, only  $m-1$  equations are linearly independent. Therefore, one node can be omitted, namely the 0 V node is omitted in the proposed methods. If an edge is connected to this point, it will have a non-zero column sum. If it is not connected to this point, it will have a column sum of zero. Using this, the criteria for the validation of topology can be formulated. The column sum for an arbitrary edge containing a current source in  $\mathbf{A}$  cannot be equal to zero:

$$\forall i_{a,j} \neq 0, j \in \{1, \dots, n\} \rightarrow \sum_{i=1}^{m-1} a_{i,j} \neq 0. \quad (15)$$

Similarly, the column sum for an arbitrary edge containing a voltage source in  $\mathbf{A}$  cannot be equal to zero:

$$\forall u_{v,j} \neq 0, j \in \{1, \dots, n\} \rightarrow \sum_{i=1}^{m-1} a_{i,j} \neq 0. \quad (16)$$

In the case of an edge that possesses admittance, the column sum in  $\mathbf{A}$  must be zero:

$$\forall y_j \neq 0, j \in \{1, \dots, n\} \rightarrow \sum_{i=1}^{m-1} a_{i,j} = 0. \quad (17)$$

In Algorithm 1, a pseudo code is shown that implements the above-mentioned criteria. It returns a Boolean value and is only true if the network topology is appropriate.

**Algorithm 1** Topology verification**INPUT:**  $\mathbf{A}$ ,  $\mathbf{Y}$ ,  $\mathbf{I}_a$ ,  $\mathbf{U}_v$ **OUTPUT:**  $IsValidTopology$ 

```

1:  $IsValidTopology = 1$ 
2:  $n = \text{NumberOfColumns}(\mathbf{A})$ 
3: for  $i = 1$  to  $n$  do
4:    $s = 0$ 
5:   if  $\mathbf{I}_a(i) \neq 0$  then
6:      $s = \text{sum}(\mathbf{A}(:, i))$ 
7:     if  $s == 0$  then
8:        $IsValidTopology = 0$ 
9:     end if
10:  end if
11:  if  $\mathbf{U}_v(i) \neq 0$  then
12:     $s = \text{sum}(\mathbf{A}(:, i))$ 
13:    if  $s == 0$  then
14:       $IsValidTopology = 0$ 
15:    end if
16:  end if
17:  if  $\mathbf{Y}(i) \neq 0$  then
18:     $s = \text{sum}(\mathbf{A}(:, i))$ 
19:    if  $s \neq 0$  then
20:       $IsValidTopology = 0$ 
21:    end if
22:  end if
23: end for

```

#### 4. Method of node powers

Since the simulated households are single-phase consumers, the corresponding active power can be calculated as follows:

$$P = U I \cos(\varphi). \quad (18)$$

Let us define the power factor ( $PF$ ) and current-source active power ( $\mathbf{P}_a$ ) vectors for further matrix calculations. Single elements in both of these describe properties with regard to a single edge.

$$\mathbf{PF}^\top = [\cos(\varphi_1) \quad \cos(\varphi_2) \quad \dots \quad \cos(\varphi_n)], \quad (19)$$

$$\mathbf{P}_a^\top = [p_{a1} \quad p_{a2} \quad \dots \quad p_{an}]. \quad (20)$$

Voltage can be expressed as the electric potential difference between two points. If the incidence matrix represents node-edge relations, transposing it will describe edge-node relations. Multiplying it with the node voltages column vector will result in edge voltages:

$$\mathbf{U} = \mathbf{A}^\top \phi. \quad (21)$$

By determining these notations and definitions, the basic equation of node voltage can be extended:

$$\begin{cases} \phi = \mathbf{Y}_A^{-1} \mathbf{A}(\text{diag}(\mathbf{Y}) \mathbf{U}_v - \mathbf{I}_a) \\ \mathbf{P}_a = \text{diag}(\mathbf{PF}) \text{diag}(\mathbf{A}^\top \phi) \mathbf{I}_a \end{cases}, \quad (22)$$

In the case of the method of node voltages, the node voltages are calculated using the given current consumption. However, in the case of the method of node powers,

the node voltages must be calculated with regard to the consumption, given in terms of the active power. Eq. 22 opens up the possibility of finding a solution following a trial and error procedure.

In other words, if only one current source in this system of equations is changed, all the node voltages will also be changed. Let us consider the case where the current of one particular customer is changed until its active power becomes equal to its actual value. Nonetheless, if a second consumer is to be set in the same fashion, the first one will be ruined. Through positive changes in current to the second source, a net drop in voltage will occur. Since power is the product of current and voltage, the current was untouched so the active power will be less. Therefore, it is also clear that if the second current source is decreased, the active power of the first source will exceed its actual value.

#### 4.1 Constant iteration current

In the aforementioned problem, the solution is acquired through an iterative process to build up the unknown currents of the system gradually, instead of trying to determine them individually. It is necessary to choose a value of the current for the iterations:  $I_{iter}$ . In the first step, every element of the current vector is zero, however, if an edge contains a consumer, its value will be set as the iteration current. The problem, namely that by changing one current, all the node voltages will also change, still exists. However, if the iteration current is sufficiently small, the ability to build up the parameters from the ground is viable. To select a source for the actual iteration, a new variable was defined:

$$dPP = \frac{\mathbf{P}_a^{sim}}{\mathbf{P}_a}. \quad (23)$$

This can be calculated for all consumer edges per iteration and provides information about how similar the simulated active power is to its desired value. It is obvious that the edge with the smallest  $dPP$  requires the highest degree of correction, therefore, it is incremented by  $I_{iter}$ . Consequently, in every iteration the branch with the minimum  $dPP$  needs to be identified, using a simple minimum search. After appropriate incrementation, the node voltages must be calculated in order to determine the new values of power. This procedure can be observed in Algorithm 2.

In an ideal case, the algorithm converges into the desired power vector and the following expression will be true:

$$\forall i \in \{1, \dots, k\} \rightarrow dPP_i = 1, \quad (24)$$

where  $k$  denotes the number of consumers in the grid. It is clear that the rate of convergence and the accuracy of the algorithm are heavily influenced by  $I_{iter}$ . If this rate is particularly small, the results will be precise ( $dPP = 1$ ). However, in this case the number of iteration cycles will be enormous because the function described is analogous

**Algorithm 2** Constant iteration current**INPUT:**  $A, Y, P_a, U_v, PF, I_{iter}, m, n$ **OUTPUT:**  $\phi, I_a$ 

```

1:  $I_a = \text{zeros}(n, 1)$ 
2: for  $i = 1$  to  $n$  do
3:   if  $P_a(i) \neq 0$  then
4:      $I_a(i) = I_{iter}$ 
5:   end if
6: end for
7:  $\phi = \text{CalculateNodeVoltages}(A, Y, U_v, I_a)$ 
8:  $P_a^{\text{sim}} = \text{diag}(\text{diag}(A^T \cdot \phi) \cdot I_a) \cdot PF$ 
9:  $[dPP, index] = \min(P_a^{\text{sim}}/P_a)$ 
10: while  $dPP < \epsilon$  do
11:    $I_a(index) = I_a(index) + I_{iter}$ 
12:    $\phi = \text{CalculateNodeVoltages}(A, Y, U_v, I_a)$ 
13:    $P_a^{\text{sim}} = \text{diag}(\text{diag}(A^T \cdot \phi) \cdot I_a) \cdot PF$ 
14:    $[dPP, index] = \min(P_a^{\text{sim}}/P_a)$ 
15: end while

```

to  $\frac{1}{x}$ . This can be observed by determining the actual number of iterations needed for the process:

$$\sum_{i=1}^k \frac{I_{ai}}{I_{iter}} \quad (25)$$

as the desired current will consist of portions of  $I_{iter}$  on every edge. According to how the desired current is divided by the iteration current, overshoots are possible. Therefore, a value of  $\epsilon$  is needed in order to secure a suitable exit condition for the loop.

Basically the blue plot represents a function similar to  $f(x) = c \bmod x$ , where  $c$  denotes a given number. In the case of the method described,  $c$  stands for a desired load current and  $x$  represents possible iteration currents used in the algorithm. According to how the desired current is divided by the actual iteration current, false values can be calculated. The periodic increase in the error is a property of the modulo operation, since the remainder increases until an integer multiple of  $x$  is identified. Then the error is equal to zero but begins to increase again. The red line represents the number of iterations needed to converge into a final solution. If  $x \ll c$ , the maximum "overshoot" by the modulo operator is relatively small, on the other hand, if  $x < c$ , more significant errors can occur. It is clear that in the first case many more iterations are necessary than in the second. Since  $c$  cannot be determined beforehand, unnecessarily large errors can occur which represents the weakness of the algorithm.

It is clear that a compromise must be made between the run-time and precision, as illustrated in Fig. 4.

#### 4.2 Dynamic iteration current

To fix the weaknesses of the algorithm elaborated on in the previous chapter, an advanced version was developed. The first aspect, in which there is room for improvement, is to obtain reasonable initial values of  $I_a$ . The process

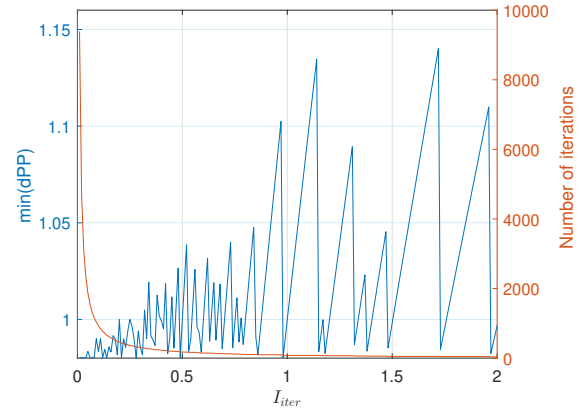


Figure 4: Minimum error - number of iterations

does not have to originate from  $I_{ai} = 0$ , consequently, a considerable amount of cycles can be skipped. In distribution networks the deviation from the nominal voltage ( $U_n$ ) is always regulated by standards, for example, in Hungary it is approximately  $\pm 7.5\%$ . Using this restriction, a general estimation can be made for the nodes. The following  $\min()$  and  $\max()$  operators relate to the values of the given function. The possible interval between current values can be formulated as follows:

$$\min(I_a) = \frac{p_{ai}}{U_n \cos(\varphi_i)} \quad , \quad (26)$$

$$\max(I_a) = \frac{p_{ai}}{(1 - D)U_n \cos(\varphi_i)}$$

where  $D$  is the aforementioned deviation value. This means overestimating the voltage results in the minimum value of the current. Setting the starting values of the currents according to the minimum approximation is adequate. To calculate the maximal remaining error, the minimum value of  $dPP_i$  needs to be calculated:

$$dPP_i = \frac{U_n(1 - D) \frac{p_{ai}}{U_n \cos(\varphi_i)} \cos(\varphi_i)}{p_{ai}} = 1 - D. \quad (27)$$

The logic behind this equation is as follows: the fraction in the numerator is the minimum current estimation the  $U_n(1 - D)$ , on the other hand, is the worst case scenario in terms of the voltage. The deviation is defined by  $D$ , so under no circumstances can the voltage drop below  $U_n(1 - D)$ , once the initial value has been set. Because  $D$  is small, this approach alone solves a huge part of the problem, since by taking the Hungarian voltage levels as an example, a  $dPP$  value greater or equal to 0.925 is achieved!

Since the initial value problem has been solved, the remaining iterations can also be improved by taking the aforementioned method one step further. In order to take the absolute error in the active power into account, the following variable is introduced:

$$dP = P_a - P_a^{\text{sim}}. \quad (28)$$

which can also be formulated using the relative error:

$$dP = P_a(1 - dPP). \quad (29)$$

Since the whole algorithm is founded on an iterative method, it would be suitable to use the minimum estimation process of the worst-case scenario in the following iterations. The idea in theory, however, is similar to the power value that is being approximated changes. Only the remaining error component needs to be recalculated. Therefore, the approximation will take the given  $dP_i$  into consideration instead of the whole  $p_{ai} \cdot U_n$  can still be used for this, but is not ideal. Once the initial values are set, a node voltage calculation is performed, thus the new node voltages can be used. As a result, the new values must be used as the upper limit of the voltage. Finally, the current generated for a particular consumer can be represented by the following series:

$$I_{ai} = \frac{1}{\cos(\varphi_i)} \left( \underbrace{\frac{p_{ai}}{U_n}}_{j=1} + \underbrace{\frac{dP'_i}{\phi'_i}}_{j=2} + \underbrace{\frac{dP''_i}{\phi''_i}}_{j=3} + \dots \right), \quad (30)$$

where  $j = 1, 2, \dots$  denotes the number of iterations. If  $j \rightarrow \infty$ , the simulated values approach the desired active powers. In theory this would mean that an infinite number of iterations would be necessary. The maximum error or minimum  $dPP$  can be calculated as shown before not only for the initial values but also for the upcoming iteration currents. Since in every consequent cycle the absolute error from the previous cycle is corrected, the remaining  $dP$  will be corrected by a minimum  $dPP$  equal to  $(1 - D)$ . Since the fitted values are multiplied, the minimum value of the  $j$ -th iteration can be calculated as follows:

$$dPP_{\min} = 1 - D^j. \quad (31)$$

The process of the method of dynamic node powers is shown in Algorithm 3. In the literature review, the

---

**Algorithm 3** Dynamic iteration current
 

---

**INPUT:**  $U_n, A, Y, P_a, U_v, PF$ 
**OUTPUT:**  $\phi, I_a$ 

- 1:  $\phi = \text{ones}(m, 1) \cdot U_n$
  - 2:  $I_a = P_a / ((A^T \cdot \phi) \cdot PF)$ , if  $\frac{0}{0} : I_{ai} \rightarrow 0$
  - 3:  $\phi = \text{CalculateNodeVoltages}(A, Y, U_v, I_a)$
  - 4:  $P_a^{\text{sim}} = \text{diag}(\text{diag}(A^T \cdot \phi) \cdot I_a) \cdot PF$
  - 5:  $dPP = \min(P_a^{\text{sim}} / P_a)$
  - 6: **while**  $dPP < \epsilon$  **do**
  - 7:  $I_{\text{iter}} = (P_a - P_a^{\text{sim}}) / (U \cdot PF)$ , if  $\frac{0}{0} : I_{ai} \rightarrow 0$
  - 8:  $I_a = I_a + I_{\text{iter}}$
  - 9:  $\phi = \text{CalculateNodeVoltages}(A, Y, U_v, I_a)$
  - 10:  $P_a^{\text{sim}} = \text{diag}(\text{diag}(A^T \cdot \phi) \cdot I_a) \cdot PF$
  - 11:  $dPP = \min(P_a^{\text{sim}} / P_a)$
  - 12: **end while**
- 

Newton-Raphson method was examined in more detail which uses the mismatch in power by incorporating the Jacobian matrix in order to achieve convergence. The proposed error recalculation method was developed whilst taking that process into consideration. Another possible solution to the calculation of powers could be an algorithm, in which the total power is recalculated rather than the power mismatch. The explicit expression for convergence was determined first, and the total power recalculation method did not suggest better results.

Although through simulation and observation, it was noted that both require the same amount of iterations. Using the Monte Carlo method, a large number of random networks were created in order to monitor the run-time. Results showed the presence of slight deviations with regard to each other. Nevertheless, the two methods require almost the same amount of computational time, as is shown in Fig. 5.

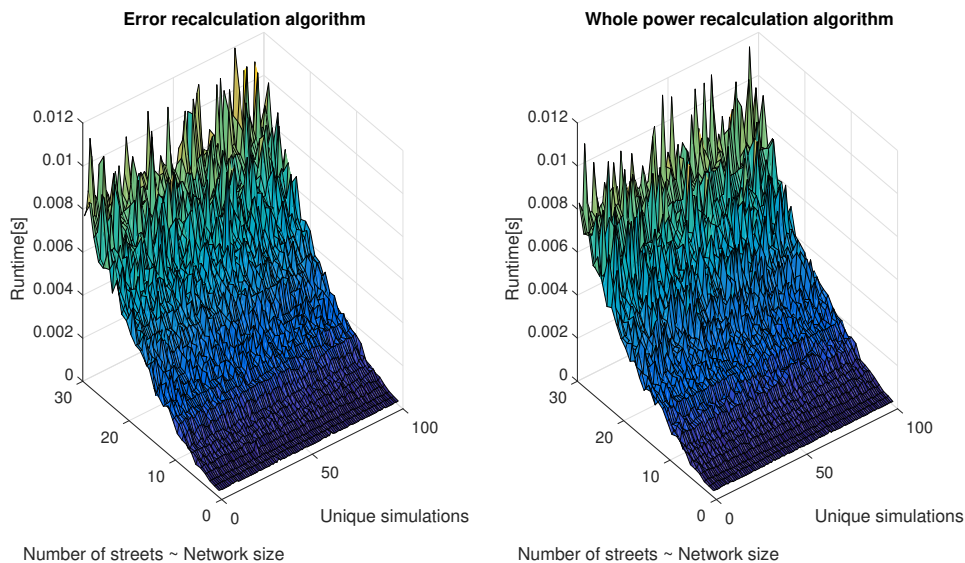


Figure 5: Monte Carlo simulation

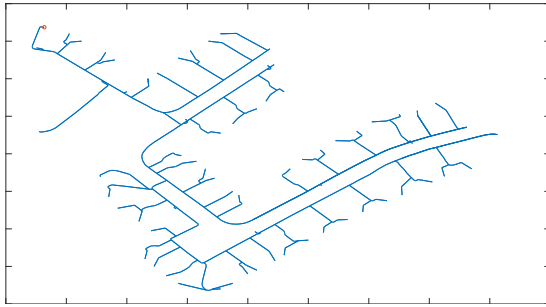


Figure 6: The topology of the test network

## 5. Simulation results

In order to verify the proposed method properly, a reference network with available simulation data was required. For verification purposes the IEEE European Low-Voltage Test Feeder network [14] was used. The simulation process consisted of the implementation of the stated algorithm and importation of network data using MATLAB. The topology of the test grid is shown in Fig. 6.

Some of the important network parameters are the following: it consists of 906 nodes which are connected by 927 edges that supply 55 single-phase consumers. Previous runs have shown that approximately 2 iterations are sufficient for engineering purposes. For the sake of accuracy, a limit was set for  $dPP$ , therefore  $\epsilon$  is still shown in Algorithm 3.

The minimum estimations of  $dPP$  predicted that the error should rapidly converge to zero. The simulation verified this statement, the rapid decrease in the absolute error can be seen in Tables 1 - 3. It can be clearly observed that in the case of a network of such a size, the maximum voltage difference can be maintained under 10 mV (Table 4).

The results of the simulation can be seen in Fig. 7 which represents the bus voltages of a single phase. Note that the  $y$  axis is divided into increments of 50 mV.

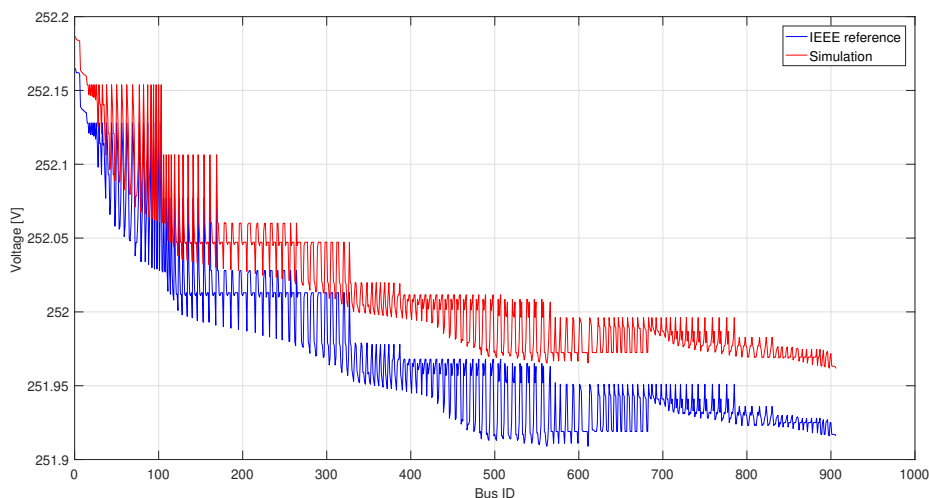


Figure 7: Simulation results for phase 'A'

Table 1: Convergence of  $dP$  in Phase 'A'

Iteration	$\max(dP)$
0.	0.052137599675 W
1.	0.000041085202 W
2.	0.000000032240 W

Table 2: Convergence of  $dP$  in Phase 'B'

Iteration	$\max(dP)$
0.	0.042119472811 W
1.	0.000029088116 W
2.	0.000000019550 W

Table 3: Convergence of  $dP$  in Phase 'C'

Iteration	$\max(dP)$
0.	0.043952897250 W
1.	0.000025174297 W
2.	0.000000014771 W

Table 4: Maximum voltage differences

Phase 'A'	Phase 'B'	Phase 'C'
0.056859 V	0.036058 V	0.023649 V

Since the active power values are reached with an excellent degree of precision, one would expect that the voltage differences would be smaller. However, this effect does not originate from the algorithm, rather from the simplified network model. This facilitates the possibility of achieving small simulation times. Since the introduction of the method of dynamic iteration currents, the number of iterations is independent of the size of the network. The computationally heavy component is the calculation of the inverse of  $Y_A$ . The topology of the network remains unchanged during the iterations. Therefore, the steady state simulation of a three-phase network requires, independent of the network size, only three matrix inversions. The duration of the simulation of this network was 2.29 s at a particular instant.

## 6. Conclusion

A novel simulation method for low-voltage distribution networks is proposed in this paper. The existing method of node voltages is further developed in order to handle active power calculations in low-voltage grids. These systems consist of a unique topology to which the process was fitted. The verification criteria for the network structure was formulated. A solution was proposed in which the number of iterations is independent of the size of the network and the simulation error decreases exponentially. The method was tested and verified on the IEEE European Low-Voltage Test Feeder network. This confirmed the statements about the algorithm. Insignificant errors appeared in the test results but these were the effect of the simplified network model.

In the future, the algorithm could be improved to handle unbalanced distribution networks. With the aid of appropriate modifications, distributed generation could be taken into consideration that accounts for not only power consumption but also generation. The algorithm can serve as a foundation of network diagnostics by using it to detect faults as well as technical or non-technical losses.

## Acknowledgement

We acknowledge the financial support of Széchenyi 2020 under the EFOP-3.6.1-16-2016-00015. We acknowledge the financial support of Széchenyi 2020 under the GINOP-2.2.1-15-2017-00038.

## Notations

$\phi$	Node-voltage vector
$I_e$	Excitation vector
$U_v$	Voltage-source vector
$I_a$	Current-source vector
$A$	Incidence matrix
$Y$	Admittance matrix
$Y_A$	Nodal admittance matrix
$m$	Number of nodes
$n$	Number of edges
$\cos(\varphi)$	Power factor
$PF$	Power factor vector
$P_a$	Current-source active-power vector
$U$	Branch voltage vector
$dPP$	Delta power percentage
$I_{iter}$	Iteration current
$dP$	Delta power
$D$	Voltage level deviation
$P_a^{sim}$	Simulated power vector
$S$	Complex power
$P$	Active power
$Q$	Reactive power
$\varphi$	Phase angle
$G$	Conductance
$\epsilon$	Threshold value for iteration

## REFERENCES

- [1] Sőrés, M.; Fodor, A.: Simulation of electrical grid with OMNeT++ open source discrete event system simulator, *Hung. J. Ind. Chem.*, 2016 **44**(2), 85–91, DOI: [10.1515/hjic-2016-0010](https://doi.org/10.1515/hjic-2016-0010)
- [2] Pózna, A.I.; Fodor, A.; Gerzson, M.; Hangos, K.M.: Colored Petri net model of electrical networks for diagnostic purposes, *IFAC-PapersOnLine*, 2018 **51**(2), 260–265, DOI: [10.1016/j.ifacol.2018.03.045](https://doi.org/10.1016/j.ifacol.2018.03.045)
- [3] Stagg, G.; El-Abiad, A.H.: Computer methods in power system analysis (McGraw-Hill), international student edn., 1968
- [4] Hertem, D.V.; Verboomen, J.; Purchala, K.; Belmans, R.; Kling, W.L.: Usefulness of DC power flow for active power flow analysis, *The 8th IEE International Conference on AC and DC Power Transmission*, 2006 **1**, DOI: [10.1049/cp:20060013](https://doi.org/10.1049/cp:20060013)
- [5] Stott, B.; Jardim, J.; Alsac, O.: DC power flow revisited, *IEEE T. Power Syst.*, 2009 **24**(3), 1290–1300, DOI: [10.1109/TPWRS.2009.2021235](https://doi.org/10.1109/TPWRS.2009.2021235)
- [6] Das, J.C.: Load flow optimization and optimal power flow (CRC Press), 2017
- [7] Görbe, P.; Magyar, A.; Hangos, K.M.: THD reduction with grid synchronized inverter's power injection of renewable sources, in 2010 International Symposium on Power Electronics Electrical Drives Automation and Motion (SPEEDAM) (IEEE), 1381–1386, DOI: [10.1109/SPEEDAM.2010.5545079](https://doi.org/10.1109/SPEEDAM.2010.5545079)
- [8] Neukirchner, L.; Görbe, P.; Magyar, A.: Voltage unbalance reduction in the domestic distribution area using asymmetric inverters, *J. Clean. Prod.*, 2017 **142**, 1710–1720, DOI: [10.1109/59.575728](https://doi.org/10.1109/59.575728)
- [9] Kersting, W.H.: Distribution system modeling and analysis (CRC Press), 2016, ISBN: 9781439856475
- [10] Bird, J.: Electrical circuit theory and technology (Taylor & Francis), 4th edn., 2010, ISBN: 185617770X
- [11] Wang, X.F.; Song, Y.; Irving, M.: Modern power system analysis (Springer), 2008, DOI: [10.1007/978-0-387-72853-7](https://doi.org/10.1007/978-0-387-72853-7)
- [12] Zhang, F.; Cheng, C.S.: A modified Newton method for radial distribution system power flow analysis, *IEEE T. Power Syst.*, 1997 **12**(1), 389 – 397, DOI: [10.1109/59.575728](https://doi.org/10.1109/59.575728)
- [13] Westinghouse Electric Corporation, Electrical transmission and distribution reference book (East Pittsburgh, PA), 1964
- [14] Schneider, K.P.; Mather, B.A.; Pal, B.C.; Ten, C.W.; Shirek, G.J.; Zhu, H.; Fuller, J.C.; Pereira, J.L.R.; Ochoa, L.F.; de Araujo, L.R.; Dugan, R.C.; Matthias, S.; Paudyal, S.; McDermott, T.E.; Kersting, W.: Analytic considerations and design basis for the IEEE distribution test feeders, *IEEE T. Power Syst.*, 2017 **33**(3), 3181–3188, DOI: [10.1109/TPWRS.2017.2760011](https://doi.org/10.1109/TPWRS.2017.2760011)