A new power flow method for radial networks

Manuel A. Matos, Member, IEEE

Abstract—The need of fast algorithms for radial power distribution networks that take advantage of their particular structure has been increasing, namely due to the use of genetic algorithms and meta-heuristics for optimization in planning and operation.

In this paper, a new method for power flow calculation in radial networks is presented. It uses an iterative process along the branches, in a way similar to other methods, but the main idea is very different from previous approaches, since it is based on the exact power flow solution for a single branch and also because it provides a complete solution (not only voltage magnitudes). The method is fast and robust for different types of networks and loads, including heavy loads.

The paper includes the theoretical derivation of the method an illustration example and tests with benchmarking networks.

Index Terms—Power Distribution, Load flow analysis, Iterative methods, Planning, Operation.

I. INTRODUCTION

The special structure of radial networks has lead, in the past, to a number of specialized algorithms that tried to take advantage of the absence of meshes to simplify the calculations and save memory [1]-[5]. In some cases, the methods are extended to weekly meshed networks with some success.

Calculation of the power flow in radial networks was not a priority in the past, since approximate methods were sufficient to have a general picture of the power flow and, if necessary, a general-purpose method (like Newton-Raphson) could always be used. Use of genetic algorithms and meta-heuristics in the optimization of distribution networks, however, lead to the need of fast calculation methods with some degree of accuracy [6]. On the other hand, more and more DMS (Distribution Management Systems) [6] are being developed and installed, and fast methods for radial networks are again welcomed. Finally, dispersed generation connected to distribution networks is growing, and adequate algorithms are needed to deal with it.

II. FOUNDATION OF THE METHOD

The main idea of the method is to use the exact power flow solution for one branch, when we know the voltage in the sending end (\(V_0\)) and the injected power in the receiving end (\(S_1\)), \(z_1\) being the impedance of branch 0-1 (see Figure 1).

![Network with a single branch](Image)

Of course, we have:

\[
V_0 - V_1 = z_1 \left( -\frac{S_1}{V_1} \right)^*
\]

or

\[
V_0^* V_1 - |V_1|^2 + z_1^* S_1 = 0 \tag{1}
\]

The analytical solution of this complex quadratic equation may be found in a number of ways. For instance, changing to rectangular coordinates and assuming that \(\theta_0=0\) (so \(V_0=e^{\theta_0}\)), \(V_1=e^{\theta_1} f_1\), \(z_1=r_1+jx_1\) and \(S_1=P_1+jQ_1\), we get:

\[
e_0 \left( e_1 + jf_1 \right) - \left( e_1^2 + f_1^2 \right) + \left( r_1 - jx_1 \right) \left( P_1 + jQ_1 \right) = 0
\]

and, after separating real and imaginary parts:

\[
e_0 e_1 - \left( e_1^2 + f_1^2 \right) + \left( r_1 P_1 + x_1 Q_1 \right) = 0 \tag{2a}
\]

\[
e_0 f_1 + \left( r_1 Q_1 - x_1 P_1 \right) = 0 \tag{2b}
\]

Now, from (2b) we take immediately the value of \(f_1\), since all the other quantities are known, and then we find \(e_1\) by substituting \(f_1\) in (2a) and solving the (real) quadratic equation, where \(e_1\) will be the biggest solution. This analytical solution of the power flow problem is well known, but generally not used, since it is only applicable to the trivial case of two buses.

![Simple radial network model](Image)

If we have now a radial network with successive branches 1, 2..., where node 0 is the root, with a specified voltage \(V_0=e^{\theta_0}\) (see Figure 2).
In the first branch, (1) transforms to:

\[ V_0 - V_1 = z_1 \left( - \sum_{k=1}^{n} S_k \frac{V_i}{V_k} \right) \]

or, multiplying by \( V_1 \) and conjugating:

\[ e_0 V_1^2 - |V_i|^2 + z_1^* \sum_{k=\text{succ}(i)} S_k \left( V_i \frac{V_i}{V_k} \right) = 0 \]  
(3)

Similar expressions may be established for the remaining nodes, in each case using the predecessor node voltage as a constant. The general expression \((i=1..n)\) is therefore:

\[ V_{\text{pred}(i)}^* V_i - |V_i|^2 + z_i^* \sum_{k=\text{succ}(i)} S_k \left( V_i \frac{V_i}{V_k} \right) = 0 \]  
(4)

The idea is then to apply (4), beginning in the first node after the root - which corresponds to (3) – in order to successively calculate the voltages, until getting the leaves of the tree. This corresponds to an iteration that can be repeated with the updated values of the voltages, until some convergence criterion is met.

III. CALCULATION PROCESS DETAILS

A. Equations

In order to conduct the iterative process, (4) is conveniently transformed to:

\[ V_{\text{pred}(i)}^* V_i - |V_i|^2 + z_i^* \left( S_i + \sum_{k=\text{succ}(i)} S_k \left( \frac{V_i}{V_k} \right) \right) = 0 \]  
(5)

where \((p)\) denotes the iteration. Once it’s clear that the voltage of the predecessor node is always known when we calculate the updated value of \( V_i \), we may write, simply (the meaning of \( S_i' \) is obvious):

\[ V_{\text{pred}(i)}^* V_i - |V_i|^2 + z_i^* S_i' = 0 \]  
(6)

corresponding to the model of Figure 3. As mentioned before, a formulation in rectangular coordinates is the best way to solve the equation in order to get \( V_i \), avoiding the need for trigonometric calculations with small angles.

Note that other modified forms of (4) could be used, but our results shown that they are less efficient than (5).

With the proposed formulation, it is not necessary to estimate initial values for the voltages, but only to consider that:

\[ V_k^{(0)} = V_i^{(0)} \quad \text{all } k \in \text{succ}(i) \]  
(7)

This corresponds to using, in the first iteration,

\[ V_{\text{pred}(i)}^* V_i - |V_i|^2 + z_i^* \left( S_i + \sum_{k=\text{succ}(i)} S_k \right) = 0 \]

instead of (6). Of course, if good initial values are known, they may always be used with the normal version of the equation.

B. Iterative process

As mentioned before, the method progresses, in each iteration, from the root to the leaves, with successive use of (6). The updated values of \( V_i \) are immediately used in their successors’ equations.

The sequence of calculations is very straightforward and similar to other forward sweep methods, so we’ll only sketch it with the help of Figure 4.

Since node 0 is the root (\( V_0 \) is known), calculation of \( V_1 \) is the first step of an iteration. In the second step, \( V_2 \) and \( V_3 \) may be calculated independently. \( V_3 \) and \( V_6 \) will follow, and \( V_4 \) will be updated in the final step of the iteration. This shows the possibility of partial parallel calculation in typical distribution networks, with important time savings.

The updated values of the voltages are then used in a new top-down iteration, and the process is repeated until a specified tolerance on the voltages’ successive values is met. A final convergence test on the specified injected power is recommended.

C. Branch model

Up to this point, a simple model for the branches has been used, considering only the branch impedance, which is common in distribution networks. However, a more detailed model can be used if necessary, with minor changes in the
equations. This feature may be important in underground networks, where the capacitance of the cables is not negligible.

\[
V_{\text{pred}(i)} = z_i V_i - y_{si} + \sum_{k \in \text{scon}(i)} y_{sk} V_k
\]

Figure 5 - Detailed branch model

Figure 5 shows the typical π model for a branch, where \( y_{si} \) is the semi-admittance of the branch. The other variables are the same of Figure 3. Because there is more than one branch connected to each node, it is now convenient to define a branch admittance for each node \( i \):

\[
Y_{si} = y_{si} + \sum_{k \in \text{scon}(i)} y_{sk}
\]

and (6) transforms into

\[
\left( V_{\text{pred}(i)} - z_i^* \sum_{k \in \text{scon}(i)} y_{sk} V_k^* \right) V_i - \left( 1 + z_i^* Y_{si}^* \right) |V_i|^2 + z_i^* S_i = 0
\]

In order to maintain the structure of the equation, and since \( 1 + z_i^* Y_{si}^* \) is a constant, we may now write:

\[
V_{\text{pred}(i)}^* - z_i^* \sum_{k \in \text{scon}(i)} y_{sk} V_k^* \left( \frac{V_i}{V_i} \right) - \left( 1 + z_i^* Y_{si}^* \right) |V_i|^2 + z_i^* S_i = 0
\]

IV. SOME ENHANCEMENTS

A. PV nodes

Although the method is primary intended to deal with PQ (or impedance) nodes, it possible to consider PV nodes as well. The existence of a PV node affects the calculations of itself and of all its predecessors. Regarding the first issue, we start by rearranging (4) in order to isolate the constant terms (note that, to deal with the more general equation (9), a similar process is possible):

\[
V_{\text{pred}(i)}^* V_i + z_i^* S_i = |V_i|^2 + z_i^* \sum_{k \in \text{scon}(i)} S_k \left( \frac{V_i}{V_k} \right) = 0
\]

\[
V_{\text{pred}(i)}^* V_i + z_i^* jQ_i = |V_i|^2 + z_i^* \sum_{k \in \text{scon}(i)} S_k \left( \frac{V_i}{V_k} \right) = 0
\]

\[
V_{\text{pred}(i)}^* V_i + z_i^* jQ_i + C = 0
\]

where \( C \) is a complex constant with obvious meaning. Now, we must solve (10) to calculate \( V_i \) and \( Q_i \). The best way is to eliminate \( Q_i \) from the two real equations that result from (10) and then use the fact that we know \( |V_i| \) to obtain the real and imaginary parts of \( V_i \). The, using (10) again, we’ll obtain \( Q_i \).

For control purposes, it is convenient to calculate also the generated reactive power, using:

\[
Q_i^G = Q_i + Q_i^{\text{load}}
\]

If \( Q_i^G \) is outside its limits, the adequate limit must be used instead, while the bus is temporarily classified as a PQ bus. The values of \( Q_i \) obtained in the process will be used in the next iteration, in the calculation of the predecessors of node \( i \). However, this is not sufficient for the algorithm to work, since in the first calculation of a node, we need the values of \( Q_i \) for all its successors. We may get ahead of this problem by using \( Q_k^{(0)} = 0 \) as initial values for all PV nodes or, if non-trivial initial values are available for the \( V_i \), by using (10) from the leaves to the root (that is, in the opposite direction of the normal algorithm) before beginning the iteration process.

We summarize now the inclusion of PV nodes in the general algorithm:

a) Estimate initial values for \( Q_k \) of all the PV buses;

b) When reaching a PV node in the iterative process, use (10) to obtain \( V_i \) and \( Q_i \);

c) Save the value of \( Q_i \) to be used in the next iteration.

B. Dispersed generation

Dispersed generation is now frequent in distribution networks, namely by means of asynchronous machines. Modeling these nodes as traditional PQ or PV nodes has been tried, but that approach doesn’t capture correctly the behavior.

D. Node admittances

It is also easy to include constant node admittances \( Y_i \), i.e., capacitors or reactors connected to node \( i \), or loads represented by a constant impedance. In fact, it is sufficient to change (8) in order to include \( Y_i \), and then use (9) as the general expression of the algorithm:

\[
Y_{si} = y_{si} + \sum_{k \in \text{scon}(i)} y_{sk} + Y_i
\]
of asynchronous generators [7].

We’ll implement here the idea of “PX node”, developed in [7] and [8]. Briefly, the generated reactive power of one such unit can be approximated by \( Q_i^G = -V_i^2/X_i \) (negative, since the machine actually gets reactive power from the network), with \( X_i = f(V_i) \).

In order to integrate this kind of node in our method, we may use tabulated values for \( X_i = f(V_i) \), or a simple model like \( X_i = (2 - V_i^2)/X_i^0 \), where \( X_i^0 \) is the value of the magnetizing reactance at nominal voltage [8]. In any case, we’ll use \( |V_i| \) to get \( X_i \) and then include it in the expression of \( Y_{si} \):

\[
Y_{sl} = y_{sl} + \sum_{k \in \text{con}(i)} y_{sk} + V_i - j \frac{1}{X_i}
\]

In this case, \((8^*)\) must be updated for all the PX nodes in each iteration (for the remaining nodes, it is a constant). On the other hand, if a battery is installed in the node, as it usually happens, it may be included as \( Y_i \) in \((8^*)\), as explained before. No other changes in the algorithm are necessary.

V. ILLUSTRATION EXAMPLE

The performance of the method was tested with case studies found in the literature. The results for two of them are reported here: a 12-bus network [1] and a 85-bus rural network [2]. In both cases, detailed data can be found in the references and will not be repeated here.

VI. ACKNOWLEDGMENT

The author would like to thank students Agostinho Sousa, and Nuno Ribeiro, both from the Faculty of Engineering of the University of Porto. Agostinho developed a test program for the new method, and Nuno helped in the comparison studies.

VII. REFERENCES