Electrical Power and Energy Systems 63 (2014) 901–907

Contents lists available at ScienceDirect

Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

A fast power flow method for radial networks with linear storage and no matrix inversions

A.C. Lisboa [a,b,1], L.S.M. Guedes [a,b,*], D.A.G. Vieira [a,b], R.R. Saldanha [b]

[a] ENACOM Handcrafted Technologies, Rua Prof. José Vieira de Mendonça 770, 31310-260 Belo Horizonte, Brazil
[b] Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Av. Antônio Carlos 6627, 31270-901 Belo Horizonte, Brazil

ARTICLE INFO

Article history:
Received 4 June 2014
Received in revised form 28 June 2014
Accepted 6 July 2014
Available online 22 July 2014

Keywords:
Radial distribution system
Forward/backward sweep algorithms
Power flow
Linear storage

ABSTRACT

This paper presents a modified direct approach for the forward/backward sweep power flow method. Taking advantage of the special topological characteristics of the radial network, an algorithm with linear storage complexity is defined. These features are summarized in the incidence matrix, which becomes a lower triangular matrix after the vertex ordering. This new formulation allows to solve linear systems of equations instead of explicitly inverting matrices during the iterative process, leading to a lower computational burden. Therefore, the proposed method is time and memory-efficient. The results show that the proposed method improves the storage and time complexity without any loss of accuracy, making it a robust and efficient method.

Introduction

Power flow is a fundamental tool for analysis, optimization and planning of electric energy distribution systems. Many activities, such as reconfiguration [1,2], restoration [3], long-term planning [4], use power flow algorithms. These applications demand numerous power flow solutions. For example, Guedes et al. [1] performed up to twenty thousand power flow calculations in the large-scale distribution system reconfiguration. Moreover, on-line decisions become more common, increasing the demand for efficient methods. In this context, computational gains are essential for the development of power systems.

In recent decades, several methods were developed for different network features (e.g. topology, line parameters, load balance), including distribution networks. Usually, distribution networks have lines with high resistance–reactance ratio and radial configuration, which may render the Jacobian matrix ill-conditioned. Particularly in the planning level, distribution networks are modeled with balanced loads.

Specialized power flow algorithms for distribution systems have been proposed, considering proper modification of existing methods. In [5] an ameliorative method of Newton–Raphson is presented, which is based on down-hill method. Penido et al. 2008 [6] proposed the use of current injections equations in the Newton method. Novel methods were also developed as the direct Z_{BR} method [7]. As an alternative to the aforementioned methods, approaches based on forward/backward sweep processes have been proposed. These methods take advantage of the unique path that connects any load bus to the source bus in a radial network. The general algorithm consists of two basic steps, backward sweep and forward sweep, which are repeated until convergence is achieved. The backward sweep is fundamentally a current or power flow summation from far end buses to the source bus, which may include voltage updates. The forward sweep is a voltage drop calculation from the source to the far end buses [8]. These algorithms have different convergence criterion (e.g. maximum active and reactive power mismatch in each node, maximum node voltage mismatch and/or total power losses mismatch), they are simple to implement and very fast for radial or weakly meshed distribution systems. The speed is high due to low computational burden to perform each iteration.

In 2003, Teng introduced a direct approach by defining the bus injection to branch current (BIBC) matrix and the branch current to bus voltage (BCBV) matrix [9]. AlHajri and El-Hawary developed a method based in a single matrix, called as radial configuration matrix (RCM), and its direct descendant matrices, the inverse and transposed inverse [10]. Methods to incorporate three-phase transformers into the forward/backward sweep-based distribution power flow was presented by Xiao et al. in 2006 [11], Teng in 2008 [12] and Elsaiah et al. in 2011 [13]. Recent studies in sweep

* Corresponding author at: ENACOM Handcrafted Technologies, Rua Prof. José Vieira de Mendonça 770, 31310-260 Belo Horizonte, Brazil.

E-mail addresses: adriano.lisboa@enacom.com.br (A.C. Lisboa), lucas.guedes@enacom.com.br (L.S.M. Guedes), douglas.vieira@enacom.com.br (D.A.G. Vieira), rodney@cpdee.ufmg.br (R.R. Saldanha).

[1] Principal corresponding author.

http://dx.doi.org/10.1016/j.ijepes.2014.07.002
0142-0615/

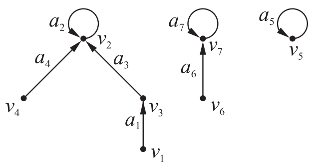

Fig. 1. A forest with root vertices v_2, v_7 and v_5 and parent vector $p = [3\ 2\ 2\ 2\ 5\ 7\ 7]^{\mathrm{T}}$.

methods focus on matrix formulations in order to improve computational burden [14–17].

This work presents a new forward/backward sweep method to solve the radial power flow problem. The topological characteristics were exploited in the construction of the incidence matrix, which allowed to solve systems of linear equations instead of inverting matrices explicitly. The proposed method has a linear storage complexity and a lower computational burden. A series of tests were applied to validate and evaluate the new approach for large-scale distribution systems. Computational test show the feasibility and the effectiveness of the method.

Linear storage sweep power flow method

Radial network representation

Let $\mathcal{F}(\mathbb{V}, \mathbb{A})$ be a directed forest, i.e. a set of independent trees, with n_R root vertices and with arcs towards respective root vertices, where $\mathbb{V} = \{v_1, \ldots, v_n\}$ is the vertex set and $\mathbb{A} = \{a_1, \ldots, a_n\}$ is the arc set. Define the root vertex set as $\mathbb{V}_R \subseteq \mathbb{V}$. The arc set $\mathbb{A}$ is augmented with laces at root vertices so that each arc may be uniquely associated with its origin vertex, as shown in Fig. 1. In this representation the vertex and arc set have the same cardinality, $|\mathbb{V}| = |\mathbb{A}| = n$. Any radial network can have this representation by simply adding loops at the roots.

Since each vertex v_i has a unique parent vertex v_{p_i}, p_i can represent the connectivity of each vertex v_i and, hence, $p \in \{1, 2, \ldots, n\}^n$ can represent the connectivity of an entire forest. For instance, the forest in Fig. 1, where $n = 7$, can be represented by the vector $p = [3\ 2\ 2\ 2\ 5\ 7\ 7]^{\mathrm{T}}$. This vector can be constructed using a breadth-first search starting from root vertices as the vertex ordering, as detailed in Section 'Vertex ordering'.

The incidence matrix

A notable matrix for networks is the incidence matrix, which becomes a square $(n \times n)$-matrix for radial networks, defined by

$$D_{i,j} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } (v_i, v_j) \in \mathbb{A}, \quad i \neq j. \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

This matrix has exactly $2n - n_R$ non-null elements and, hence, it is a sparse matrix with linear storage (i.e. $O(n)$ in the big O notation). Furthermore, if the vertices are sorted according to their tree-level (see Fig. 2), the matrix D becomes lower triangular. In this case, a linear system of equations $Dx = b$ or $D^{\mathrm{T}}x = b$ can be solved for x with exact n additions and, hence, linear time complexity, $O(n)$.

For instance, the incidence matrix for the forest shown in Fig. 1 would be

$$D = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

while for the forest shown in Fig. 2 it would be

$$D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{bmatrix}. \tag{3}$$

Notice that a linear system of equations $Dx = b$ with matrix (3) can be trivially solved by evaluating lower order vertices first (this is called forward substitution), which may not be true for arbitrary vertex orderings (e.g. matrix in (2)). Considering a linear system of equations $Dx = b$, the ith row of D can be interpreted as setting a potential in the vertex v_i as the potential of its parent v_{p_i} plus a delta potential b_i. Similarly, for a system $D^{\mathrm{T}}x = b$, the ith row of D^{T} can be interpreted as setting a flow in the edge (v_i, v_{p_i}) as the flow of immediate lower level incident edges plus a flow injection b_i.

Let e_i be the unit vector with all null-elements, but the ith which is unitary. The solution of $Dx = e_i$ can be interpreted as all sub-tree vertices of vertex v_i taking the same potential, i.e. a unitary potential. Considering that the identity matrix can be written as $[e_1 e_2, \ldots, e_n]$, the inverse of D is also a lower triangular matrix with only zeros and ones (i.e. $DD^{-1} = [e_1 e_2, \ldots, e_n]$ so that $D^{-1} = [D^{-1}e_1 D^{-1}e_2, \ldots, D^{-1}e_n]$). For instance, the inverse of matrix (3) is

$$D^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}. \tag{4}$$

The ith column of this matrix indicates member vertices of the respective vertex v_i. Following this analysis, for each element x_i the linear system of equations solution $x = D^{-\mathrm{T}}b$ can be interpreted as the accumulation of nodal values b into the respective sub-tree root vertex v_i, including b_i itself. Analogously, $x = D^{-1}b$ can be interpreted as the accumulation of edge values b into the respective end vertex v_i of a path to the respective tree root vertex v_j, including b_j itself. For instance, consider the matrix in (3) and its inverse in (4) (see Fig. 2), and let $b = [1\ 2\ 3\ 4\ 5\ 6\ 7]^{\mathrm{T}}$. Then, $D^{-\mathrm{T}}b = [1\ 78\ 3\ 4\ 12\ 6\ 7]^{\mathrm{T}}$ and $D^{-1}b = [1\ 2\ 3\ 5\ 6\ 8\ 13]^{\mathrm{T}}$.

The worst-case storage for D^{-1} occurs for a path graph, where the lower triangular part of D^{-1} is entirely filled with ones, i.e. exactly $n(n+1)/2$ non-null elements. Fig. 3 shows the matrices D and D^{-1} to a path graph with 100 vertices. This implies not only a storage $O(n^2)$, but also a time $O(n^2)$ for $(D^{-1})b$, $b \in \mathbb{R}^n$, considering that D^{-1} is actually computed instead of solving a linear system of equations. This has a strong impact onto numerical methods: it

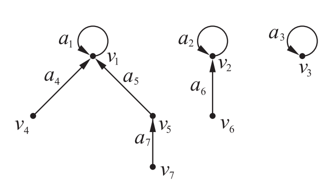

Fig. 2. A forest with root vertices v_1, v_2 and v_3 and parent vector $p = [1\ 2\ 3\ 1\ 1\ 2\ 5]^{\mathrm{T}}$, whose vertices are ordered according to their respective levels.