IET Generation, Transmission & Distribution

Research Article

# Modified tensor method to power flow analysis

ISSN 1751-8687
Received on 18th July 2018
Revised 5th April 2019
Accepted on 28th June 2019
E-First on 15th August 2019
doi: 10.1049/iet-gtd.2018.6187
www.ietdl.org

*Erika Pequeno dos Santos[1,2], Bruna Kobay[1,2], Mauro Augusto da Rosa[1,2], Roberto de Souza Salgado[1], Diego Issicaba[1,2]* ✉

*[1]Department of Electrical and Electronic Engineering, Federal University of Santa Catarina (UFSC), Campus Reitor João David Ferreira Lima, Trindade 88040-900, Florianópolis – SC, Brazil*

*[2]INESC P&D Brasil, R. José Caballero 15, Gonzaga 11055–300, Santos – SP, Brazil*

✉ *E-mail: diego.issicaba@ufsc.br*

**Abstract:** This study introduces a modified tensor-based power flow method designed using complex voltages represented in their rectangular form. Tensor-based methods employ the second-order Taylor series expansion of the power flow equations to estimate a correction vector to be added to the iterative increments of the Newton–Raphson method. In the proposed approach, the estimation of the correction vector is improved by taking advantage of a relation between the Jacobian and tensor arrays and symmetries between matrix–vector products involving Jacobian matrices. Result analysis highlights the efficiency and robustness of the proposed method in comparison with the original tensor-based approach using test systems and an actual system.

## 1 Introduction

The power flow problem is one of the most classical problems in power system analysis. It consists of determining the bus complex voltages in a power system in steady-state operation. The problem is usually formulated as a set of non-linear algebraic equations which are solved through iterative methods like the Newton–Raphson method [1] and variations such as the fast decoupled method [2]. The solution of the power flow equations is still a subject of research interest to the power engineering community [3–6] and related contributions promote variations to optimal power flow analysis, voltage stability analysis and so forth.

The Newton–Raphson method has become the standard technique to address the problem [7]. It is based on a first-order approximation of a vectorial function defined using the power flow equations. As instances of applications, Kulworawanichpong [8] uses non-linear current mismatches instead of power mismatches to model the power flow problem, which is solved using the conventional Newton–Raphson method. Although the authors show that advantages can be expected from this approach, the numerical tests shown in the paper concern only small size networks. In [9], the coupling $P\delta$–$QV$ is exploited in the conventional Newton–Raphson power flow together with the use of constant matrices in the iterative scheme. The numerical results obtained through this approach exhibit better performance (in terms of number of iterations) than the conventional decoupled methods (XB and BX). However, only small IEEE test systems (14, 30, 57, 118 buses) have been used in the numerical simulations. An alternative formulation to the power flow problem has been recently proposed in [10] by extending the factorised state estimation solution methodology to the power flow problem. This can be seen as a particular state estimation problem with equality constraints only and non-singular Jacobian matrices. The authors show that this scheme reduces the number of iterations for convergence becoming an interesting alternative to solving the problem.

Some extensions to the Newton–Raphson method have been designed using second-order approximations assuming complex voltages are represented in the polar form [11, 12]. Similarly, higher-order variations have been recently designed and well described in [4]. Other approaches focus on the representation of complex voltages in the rectangular form. The advantage of this representation is that voltage variables are written in terms of quadratic equations. This allows to represent the power flow equations through second-order Taylor series expansions without truncation processes. In [13], this feature has been exploited by demonstrating that the second-order term is equal to the power flow equations evaluated at an operation point given by the difference between the solution vector and an initial estimate. Using this result, variations have been developed [14–16] with reports of improvements in efficiency.

Tensor methods have been proposed as a general mathematical tool to solve sets of non-linear equations [17, 18]. They have been used in the past to formulate steady-state power system problems, and to show that solving a set of non-linear equations through the Newton–Raphson method is equivalent to the solution of a linear incremental tensor method [19, 20]. Among the applications of second-order terms to power flow analysis, the tensor-based method designed in [21] can be highlighted. The main concept behind this method, named tensor method version II in the reference, is the estimation of the impact of the second-order term using the Newton's step direction. The procedure introduces a correction vector which is shown to improve the iterative process in terms of robustness, number of iterations and computing time. Extensions of this formulation have been already used in voltage stability evaluations in [22, 23].

This paper proposes a modification in the tensor-based power flow method provided in [21], to better use the second-order Taylor series expansion of the vector function corresponding to the power flow equations. Its main contribution is to show how to exploit the linear relationship between the Jacobian matrix and the power flow variables to improve the tensor-based method. Following is shown: (i) a demonstration of an important property involving the symmetry of matrix–vector products computed by using Jacobian matrices; (ii) how to adjust the Jacobian matrix of the tensor-based method to take into account the demonstrated property. As a consequence, a better direction of search is obtained if an adjustment in the Jacobian matrix is used to compute the correction vector. This speeds up and increases the robustness of the iterative process. Numerical results emphasise the efficiency and robustness of the proposed method in comparison with its original counterpart using test systems and an actual system.

The paper is organised as follows. In Section 2, problem formulation is briefly presented. In Section 3, aspects of the tensor method proposed in [21] are discussed. The proposed modified tensor method is presented in Section 4. Numerical results and final remarks are outlined in Sections 5 and 6, respectively.

## 2 Problem formulation in vectorial form

Power flow equations can be written by assuming the complex bus voltages are represented either in their polar or rectangular forms. Using the latter, the set of bus complex power injections in a network can be computed in a vector form as

$$S(\boldsymbol{e},\boldsymbol{f}) = (\boldsymbol{D_e} + j\boldsymbol{D_f})(\boldsymbol{I_e} - j\boldsymbol{I_f}) \tag{1}$$

where $\boldsymbol{e}$ and $\boldsymbol{f}$ are vectors with the real and imaginary parts of the bus voltages, respectively, while $\boldsymbol{I_e}$ and $\boldsymbol{I_f}$ are vectors with the real and imaginary parts of bus injected currents. The notation $\boldsymbol{D_u}$ represents a diagonal matrix with diagonal values given by the entries of vector $\boldsymbol{u}$.

Assuming the network admittance matrix is expressed as $\boldsymbol{Y} = \boldsymbol{G} + j\boldsymbol{B}$, then

$$\boldsymbol{I_e} + j\boldsymbol{I_f} = (\boldsymbol{G} + j\boldsymbol{B})(\boldsymbol{e} + j\boldsymbol{f}) = \boldsymbol{Ge} - \boldsymbol{Bf} + j(\boldsymbol{Be} + \boldsymbol{Gf}) \tag{2}$$

resulting that $\boldsymbol{I_e} = \boldsymbol{Ge} - \boldsymbol{Bf}$ and $\boldsymbol{I_f} = \boldsymbol{Be} + \boldsymbol{Gf}$.

By combining (1) and (2), the active and reactive powers injected in each bus can be written as

$$\boldsymbol{P}(\boldsymbol{e},\boldsymbol{f}) = \boldsymbol{D_e}\boldsymbol{Ge} - \boldsymbol{D_e}\boldsymbol{Bf} + \boldsymbol{D_f}\boldsymbol{Be} + \boldsymbol{D_f}\boldsymbol{Gf} \tag{3}$$

$$\boldsymbol{Q}(\boldsymbol{e},\boldsymbol{f}) = \boldsymbol{D_f}\boldsymbol{Ge} - \boldsymbol{D_f}\boldsymbol{Bf} + \boldsymbol{D_e}\boldsymbol{Be} + \boldsymbol{D_e}\boldsymbol{Gf} \tag{4}$$

Mismatch equations can be defined for bus active power injections, bus reactive power injections and bus voltage magnitudes as

$$\boldsymbol{\Delta P}(\boldsymbol{e},\boldsymbol{f}) = \boldsymbol{P_s} - \boldsymbol{P}(\boldsymbol{e},\boldsymbol{f}) \tag{5}$$

$$\boldsymbol{\Delta Q}(\boldsymbol{e},\boldsymbol{f}) = \boldsymbol{Q_s} - \boldsymbol{Q}(\boldsymbol{e},\boldsymbol{f}) \tag{6}$$

$$\boldsymbol{\Delta V}(\boldsymbol{e},\boldsymbol{f}) = \boldsymbol{D_{V_s}}\boldsymbol{V_s} - \boldsymbol{D_e}\boldsymbol{e} - \boldsymbol{D_f}\boldsymbol{f} \tag{7}$$

where $\boldsymbol{P_s}$, $\boldsymbol{Q_s}$ and $\boldsymbol{V_s}$ are vectors with pre-specified values for the active power injections, reactive power injections and voltage magnitudes. Each bus-related mismatch is a quadratic function of the variables in $\boldsymbol{e}$ and $\boldsymbol{f}$.

Expressions (5)–(7) can be written in terms of a vector function $\boldsymbol{h}(\hat{\boldsymbol{x}})$ containing $3n_b$ scalar functions and $2n_b$ variables, where $n_b$ is the number of system buses and $\hat{\boldsymbol{x}} = [\boldsymbol{e},\boldsymbol{f}]^{\mathbf{T}}$. Usually, the terms $V\delta$, PQ and PV are employed to differentiate types of buses where values are pre-specified for voltage magnitude and angle, active and reactive power injections, and active power injection and voltage magnitude, respectively. The power flow equations are then defined by using a subset of $\boldsymbol{h}(\hat{\boldsymbol{x}})$ composed of $n_{\mathrm{PV}} + n_{\mathrm{PQ}}$ functions related to active power mismatches, $n_{\mathrm{PQ}}$ functions associated to reactive power mismatches, and $n_{\mathrm{PV}}$ functions related to voltage mismatches, where $n_{\mathrm{PV}}$ and $n_{\mathrm{PQ}}$ correspond to the number of PQ and PV buses, respectively. Traditionally, only one bus is chosen to be $V\delta$, resulting in a set of $2n_b - 2$ functions which can be written as $\boldsymbol{g}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a vector with real and imaginary parts of bus voltages in the PQ and PV buses. The solution of the power flow problem consists of determining values for the $2n_b - 2$ bus voltage variables in $\boldsymbol{x}$ in order to reduce the mismatches to zero, or at least to values close to zero for a given tolerance.

## 3 Discussions on the tensor-based method proposed in [21]

The power flow problem can be represented in a compact form as

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g_s} - \boldsymbol{g_c}(\boldsymbol{x}) \tag{8}$$

where $\boldsymbol{g}(\boldsymbol{x})$ is a vector function of mismatches computed at $\boldsymbol{x}$, $\boldsymbol{g_s}$ is a vector of specified values and $\boldsymbol{g_c}(\boldsymbol{x})$ is a vector function of power injections and squares of voltage magnitudes computed at $\boldsymbol{x}$.

The vector function $\boldsymbol{g}(\boldsymbol{x})$ can be approximated by its first-order Taylor series expansion computed at $\boldsymbol{x}$ plus a direction $\boldsymbol{d_n}$ as follows:

$$\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d_n}) \approx \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_n \tag{9}$$

where $\boldsymbol{J}(\boldsymbol{x})$ is the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ evaluated in $\boldsymbol{x}$.

Using (9), starting from a solution estimate $\boldsymbol{x}^{(0)}$, the direction for which $\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d_n})$ is a null vector can be computed by using the expression $\boldsymbol{d}_n = -[\boldsymbol{J}(\boldsymbol{x})]^{-1}\boldsymbol{g}(\boldsymbol{x})$. The Newton–Raphson method employs this concept in an iterative procedure based on the update rule

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}_n^{(k)} \tag{10}$$

where

$$\boldsymbol{d}_n^{(k)} = -\left[\boldsymbol{J}(\boldsymbol{x}^{(k)})\right]^{-1}\boldsymbol{g}(\boldsymbol{x}^{(k)}) \tag{11}$$

However, since $\boldsymbol{g}(\boldsymbol{x})$ is a quadratic function of $\boldsymbol{x}$, $\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d})$ can be calculated without approximations using the second-order Taylor series expansion

$$\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\mathbf{T}}\boldsymbol{T}\boldsymbol{d} \tag{12}$$

where $\boldsymbol{T}$ is a constant tridimensional array named *tensor*.

Although (12) allows expanding $\boldsymbol{g}(\boldsymbol{x})$ in the direction $\boldsymbol{d}$ without truncation, the analytic determination of $\boldsymbol{d}$ is a problem yet to be solved. Nevertheless, the second-order expansion can be utilised to design improved iterative methods to solve (8), as documented in [21, 23]. In fact, using (8) we have

$$\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{g_s} - \boldsymbol{g_c}(\boldsymbol{x} + \boldsymbol{d})\,. \tag{13}$$

In the null point $\boldsymbol{x} = \boldsymbol{0}$, the function $\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d})$ can be expressed using (12) as

$$\boldsymbol{g}(\boldsymbol{d}) = \boldsymbol{g}(\boldsymbol{0}) + \boldsymbol{J}(\boldsymbol{0})\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\mathrm{T}}\mathrm{T}\boldsymbol{d} = \boldsymbol{g_s} + \frac{1}{2}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d} \tag{14}$$

or, using (13), as

$$\boldsymbol{g}(\boldsymbol{d}) = \boldsymbol{g_s} - \boldsymbol{g_c}(\boldsymbol{d}) \tag{15}$$

By combining (14) and (15), we have

$$\boldsymbol{g_c}(\boldsymbol{d}) = -\frac{1}{2}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}\,. \tag{16}$$

which is an expression introduced in [13] using a different deduction.

Therefore, using (16), the second-order Taylor expansion in (12) can be written as

$$\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d} - \boldsymbol{g_c}(\boldsymbol{d}) \tag{17}$$

Let now $\boldsymbol{d}_t$ be a term to be added to $\boldsymbol{d}_n$ to achieve the direction $\boldsymbol{d}$. Then, the expansion in (17) can be expressed as

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) &= \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}_n + \boldsymbol{d}_t) \\ &= \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})(\boldsymbol{d}_n + \boldsymbol{d}_t) - \boldsymbol{g_c}(\boldsymbol{d}_n + \boldsymbol{d}_t) \\ &= \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_n + \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_t - \boldsymbol{g_c}(\boldsymbol{d}_n + \boldsymbol{d}_t) \end{aligned} \tag{18}$$

By noticing that $\boldsymbol{d}_n = -[\boldsymbol{J}(\boldsymbol{x})]^{-1}\boldsymbol{g}(\boldsymbol{x})$, expression (18) can be further simplified as follows:

$$\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{d}) = \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_t - \boldsymbol{g_c}(\boldsymbol{d}_n+\boldsymbol{d}_t) \tag{19}$$

Since the perfect direction $\boldsymbol{d}$ is unknown, the approximation $\boldsymbol{g_c}(\boldsymbol{d}_n+\boldsymbol{d}_t) \approx \boldsymbol{g_c}(\boldsymbol{d}_n)$ can be employed assuming that the incremental variation $\boldsymbol{d}$ is predominantly composed of the direction provided by the linear model of the Newton–Raphson method. As a consequence, the additional direction for which $\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{d})$ is a null vector can be estimated using the expression $\boldsymbol{d}_t = [\boldsymbol{J}(\boldsymbol{x})]^{-1}\boldsymbol{g_c}(\boldsymbol{d}_n)$. The tensor method applies this expression in an iterative procedure according to the update rule

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}_n^{(k)} + \boldsymbol{d}_t^{(k)} \tag{20}$$

where $\boldsymbol{d}_n^{(k)}$ is computed using (11) and

$$\boldsymbol{d}_t^{(k)} = \left[\boldsymbol{J}(\boldsymbol{x}^{(k)})\right]^{-1}\boldsymbol{g_c}(\boldsymbol{d}_n^{(k)}) \tag{21}$$

The tensor-based strategy has shown to provide increased efficiency and robustness in comparison to the Newton–Raphson method and a tensor approach based on interpolations. The algorithm introduced in [21] has the particularity of performing calculations to verify if the application of the term $\boldsymbol{d}_t$ is indeed able to further reduce the mismatches, that is, if

$$\left\|\boldsymbol{g}(\boldsymbol{x}^{(k)}+\boldsymbol{d}_n^{(k)}+\boldsymbol{d}_t^{(k)})\right\|_\infty \le \left\|\boldsymbol{g}(\boldsymbol{x}^{(k)}+\boldsymbol{d}_n^{(k)})\right\|_\infty \tag{22}$$

If not, only the standard Newton direction $\boldsymbol{d}_n$ is utilised and (10) is chosen as update rule. Since theoretically the tensor-based complementary direction should improve the direction $\boldsymbol{d}$, the need for this procedure suggests that there is the possibility of improving the estimation of $\boldsymbol{d}_t$.

In fact, notice that the assumption $\boldsymbol{g_c}(\boldsymbol{d}_n+\boldsymbol{d}_t) \approx \boldsymbol{g_c}(\boldsymbol{d}_n)$ implies by (16) that

$$\begin{aligned} &\boldsymbol{g_c}(\boldsymbol{d}_n) - \boldsymbol{g_c}(\boldsymbol{d}_n+\boldsymbol{d}_t) \approx \boldsymbol{0} \\ &-\frac{1}{2}\boldsymbol{d}_n^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_n + \frac{1}{2}(\boldsymbol{d}_n+\boldsymbol{d}_t)^{\mathrm{T}}\boldsymbol{T}(\boldsymbol{d}_n+\boldsymbol{d}_t) \approx \boldsymbol{0} \\ &\frac{1}{2}\boldsymbol{d}_n^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{d}_t^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_n + \frac{1}{2}\boldsymbol{d}_t^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_t \approx \boldsymbol{0} \end{aligned} \tag{23}$$

This analysis indicates that two terms of the second-order expansion of $\boldsymbol{g}(\boldsymbol{x})$ which are linearly dependent of $\boldsymbol{d}_t$ are actually unused in the estimation of $\boldsymbol{d}_t$. The proposed modified tensor method exploits this linearity to improve the correction vector $\boldsymbol{d}_t$ taking advantage of mathematical relations between the tensor and Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ as well as an equality involving matrix–vector multiplications computed with the Jacobian matrix.

## 4 Modified tensor method

This section presents a proof for equality $\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_t = \boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_n$, the proposed modification in the updated rule of the tensor-based method and the resulting algorithm.

### 4.1 Symmetry between matrix–vector products involving the Jacobian matrix

In order to prove the equality $\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_t = \boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_n$, let us assume two direction vectors $\boldsymbol{d}_n^h$ and $\boldsymbol{d}_t^h$, in $\mathfrak{R}^{2n_b}$, written in the form

$$\boldsymbol{d}_n^h = [\boldsymbol{a}, \boldsymbol{b}]^{\mathrm{T}} \tag{24}$$

$$\boldsymbol{d}_t^h = [\boldsymbol{r}, \boldsymbol{s}]^{\mathrm{T}} \tag{25}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ correspond to increments on the real and imaginary parts of bus voltages in vector $\boldsymbol{d}_n^h$, while $\boldsymbol{r}$ and $\boldsymbol{s}$ are increments on the real and imaginary parts of the bus voltages in vector $\boldsymbol{d}_t^h$, respectively.

The Jacobian matrix of $\boldsymbol{h}(\hat{\boldsymbol{x}})$ evaluated in $\hat{\boldsymbol{x}}$ can be expressed as

$$\boldsymbol{J}^h(\boldsymbol{y}) = \begin{bmatrix} \boldsymbol{J}_1^h(\hat{\boldsymbol{x}}) & \boldsymbol{J}_2^h(\hat{\boldsymbol{x}}) \\ \boldsymbol{J}_3^h(\hat{\boldsymbol{x}}) & \boldsymbol{J}_4^h(\hat{\boldsymbol{x}}) \\ \boldsymbol{J}_5^h(\hat{\boldsymbol{x}}) & \boldsymbol{J}_6^h(\hat{\boldsymbol{x}}) \end{bmatrix} \tag{26}$$

where

$$\boldsymbol{J}_1^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta P}}{\partial \boldsymbol{e}} = -\boldsymbol{D_e G} - \boldsymbol{D_f B} - \boldsymbol{D_{Ge}} + \boldsymbol{D_{Bf}} \tag{27}$$

$$\boldsymbol{J}_2^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta P}}{\partial \boldsymbol{f}} = -\boldsymbol{D_f G} + \boldsymbol{D_e B} - \boldsymbol{D_{Be}} - \boldsymbol{D_{Gf}} \tag{28}$$

$$\boldsymbol{J}_3^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta Q}}{\partial \boldsymbol{e}} = -\boldsymbol{D_f G} + \boldsymbol{D_e B} + \boldsymbol{D_{Be}} + \boldsymbol{D_{Gf}} \tag{29}$$

$$\boldsymbol{J}_4^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta Q}}{\partial \boldsymbol{f}} = +\boldsymbol{D_e G} + \boldsymbol{D_f B} - \boldsymbol{D_{Ge}} + \boldsymbol{D_{Bf}} \tag{30}$$

$$\boldsymbol{J}_5^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta V}}{\partial \boldsymbol{e}} = -2\boldsymbol{D_e} \tag{31}$$

$$\boldsymbol{J}_6^h(\boldsymbol{y}) = \frac{\partial \boldsymbol{\Delta V}}{\partial \boldsymbol{f}} = -2\boldsymbol{D_f} \tag{32}$$

As a consequence, by using (24)–(32), the following multiplications can be performed:

$$\boldsymbol{J}_1^h(\boldsymbol{d}_n^h)\boldsymbol{r} = -\boldsymbol{D_a G r} - \boldsymbol{D_b B r} - \boldsymbol{D_{Ga} r} + \boldsymbol{D_{Bb} r} \tag{33}$$

$$\boldsymbol{J}_1^h(\boldsymbol{d}_t^h)\boldsymbol{a} = -\boldsymbol{D_r G a} - \boldsymbol{D_s B a} - \boldsymbol{D_{Gr} a} + \boldsymbol{D_{Bs} a} \tag{34}$$

$$\boldsymbol{J}_2^h(\boldsymbol{d}_n^h)\boldsymbol{s} = -\boldsymbol{D_b G s} + \boldsymbol{D_a B s} - \boldsymbol{D_{Ba} s} - \boldsymbol{D_{Gb} s} \tag{35}$$

$$\boldsymbol{J}_2^h(\boldsymbol{d}_t^h)\boldsymbol{b} = -\boldsymbol{D_s G b} + \boldsymbol{D_r B b} - \boldsymbol{D_{Br} b} - \boldsymbol{D_{Gs} b} \tag{36}$$

$$\boldsymbol{J}_3^h(\boldsymbol{d}_n^h)\boldsymbol{r} = -\boldsymbol{D_b G r} + \boldsymbol{D_a B r} + \boldsymbol{D_{Ba} r} + \boldsymbol{D_{Gb} r} \tag{37}$$

$$\boldsymbol{J}_3^h(\boldsymbol{d}_t^h)\boldsymbol{a} = -\boldsymbol{D_s G a} + \boldsymbol{D_r B a} + \boldsymbol{D_{Br} a} + \boldsymbol{D_{Gs} a} \tag{38}$$

$$\boldsymbol{J}_4^h(\boldsymbol{d}_n^h)\boldsymbol{s} = +\boldsymbol{D_a G s} + \boldsymbol{D_b B s} - \boldsymbol{D_{Ga} s} + \boldsymbol{D_{Bb} s} \tag{39}$$

$$\boldsymbol{J}_4^h(\boldsymbol{d}_t^h)\boldsymbol{b} = +\boldsymbol{D_r G b} + \boldsymbol{D_s B b} - \boldsymbol{D_{Gr} b} + \boldsymbol{D_{Bs} b} \tag{40}$$

$$\boldsymbol{J}_5^h(\boldsymbol{d}_n^h)\boldsymbol{r} = -2\boldsymbol{D_a r} \tag{41}$$

$$\boldsymbol{J}_5^h(\boldsymbol{d}_t^h)\boldsymbol{a} = -2\boldsymbol{D_r a} \tag{42}$$

$$\boldsymbol{J}_6^h(\boldsymbol{d}_n^h)\boldsymbol{s} = -2\boldsymbol{D_b s} \tag{43}$$

$$\boldsymbol{J}_6^h(\boldsymbol{d}_t^h)\boldsymbol{b} = -2\boldsymbol{D_s b} \tag{44}$$

For the sake of simplicity, let us define the vectors

$$\boldsymbol{\alpha}_{nt} = \boldsymbol{J}_1^h(\boldsymbol{d}_n^h)\boldsymbol{r} + \boldsymbol{J}_2^h(\boldsymbol{d}_n^h)\boldsymbol{s} \tag{45}$$

$$\boldsymbol{\beta}_{nt} = \boldsymbol{J}_3^h(\boldsymbol{d}_n^h)\boldsymbol{r} + \boldsymbol{J}_4^h(\boldsymbol{d}_n^h)\boldsymbol{s} \tag{46}$$

$$\boldsymbol{\gamma}_{nt} = \boldsymbol{J}_5^h(\boldsymbol{d}_n^h)\boldsymbol{r} + \boldsymbol{J}_6^h(\boldsymbol{d}_n^h)\boldsymbol{s} \tag{47}$$

$$\boldsymbol{\alpha}_{tn} = \boldsymbol{J}_1^h(\boldsymbol{d}_t^h)\boldsymbol{a} + \boldsymbol{J}_2^h(\boldsymbol{d}_t^h)\boldsymbol{b} \tag{48}$$

$$\boldsymbol{\beta}_{tn} = \boldsymbol{J}_3^h(\boldsymbol{d}_t^h)\boldsymbol{a} + \boldsymbol{J}_4^h(\boldsymbol{d}_t^h)\boldsymbol{b} \tag{49}$$

$$\boldsymbol{\gamma}_{tn} = \boldsymbol{J}_5^h(\boldsymbol{d}_t^h)\boldsymbol{a} + \boldsymbol{J}_6^h(\boldsymbol{d}_t^h)\boldsymbol{b} \tag{50}$$

Let $\boldsymbol{z}$ and $\boldsymbol{q}$ be vectors in $\mathfrak{R}^{n_b}$. Let also $\boldsymbol{A}$ be a matrix in $\mathfrak{R}^{n_b \times n_b}$. Then, the following equality holds:

$$\boldsymbol{D}_z\boldsymbol{A}\boldsymbol{q} = \boldsymbol{D}_{Aq}\boldsymbol{z} \tag{51}$$

Then, using (33)–(50), we have

$$\begin{aligned} \boldsymbol{\alpha}_{nt} = & -\boldsymbol{D}_a\boldsymbol{G}\boldsymbol{r} - \boldsymbol{D}_b\boldsymbol{B}\boldsymbol{r} - \boldsymbol{D}_{Ga}\boldsymbol{r} + \boldsymbol{D}_{Bb}\boldsymbol{r} \\ & -\boldsymbol{D}_b\boldsymbol{G}\boldsymbol{s} + \boldsymbol{D}_a\boldsymbol{B}\boldsymbol{s} - \boldsymbol{D}_{Ba}\boldsymbol{s} - \boldsymbol{D}_{Gb}\boldsymbol{s} \\ = & -\boldsymbol{D}_{Gr}\boldsymbol{a} - \boldsymbol{D}_{Br}\boldsymbol{b} - \boldsymbol{D}_r\boldsymbol{G}\boldsymbol{a} + \boldsymbol{D}_r\boldsymbol{B}\boldsymbol{b} \\ & -\boldsymbol{D}_{Gs}\boldsymbol{b} + \boldsymbol{D}_{Bs}\boldsymbol{a} - \boldsymbol{D}_s\boldsymbol{B}\boldsymbol{a} - \boldsymbol{D}_s\boldsymbol{G}\boldsymbol{b} \\ = & \ \boldsymbol{\alpha}_{tn} \end{aligned} \tag{52}$$

$$\begin{aligned} \boldsymbol{\beta}_{nt} = & -\boldsymbol{D}_b\boldsymbol{G}\boldsymbol{r} + \boldsymbol{D}_a\boldsymbol{B}\boldsymbol{r} + \boldsymbol{D}_{Ba}\boldsymbol{r} + \boldsymbol{D}_{Gb}\boldsymbol{r} \\ & +\boldsymbol{D}_a\boldsymbol{G}\boldsymbol{s} + \boldsymbol{D}_b\boldsymbol{B}\boldsymbol{s} - \boldsymbol{D}_{Ga}\boldsymbol{s} + \boldsymbol{D}_{Bb}\boldsymbol{s} \\ = & -\boldsymbol{D}_{Gr}\boldsymbol{b} + \boldsymbol{D}_{Br}\boldsymbol{a} + \boldsymbol{D}_r\boldsymbol{B}\boldsymbol{a} + \boldsymbol{D}_r\boldsymbol{G}\boldsymbol{b} \\ & +\boldsymbol{D}_{Gs}\boldsymbol{a} + \boldsymbol{D}_{Bs}\boldsymbol{b} - \boldsymbol{D}_s\boldsymbol{G}\boldsymbol{a} + \boldsymbol{D}_s\boldsymbol{B}\boldsymbol{b} \\ = & \ \boldsymbol{\beta}_{tn} \end{aligned} \tag{53}$$

$$\begin{aligned} \boldsymbol{\gamma}_{nt} = & -2\boldsymbol{D}_a\boldsymbol{r} - 2\boldsymbol{D}_b\boldsymbol{s} \\ = & -2\boldsymbol{D}_r\boldsymbol{a} - 2\boldsymbol{D}_s\boldsymbol{b} \\ = & \ \boldsymbol{\gamma}_{tn} \end{aligned} \tag{54}$$

From (52)–(54), we have

$$\boldsymbol{J}^h(\boldsymbol{d}_n^h)\boldsymbol{d}_t^h = \begin{bmatrix} \boldsymbol{\alpha}_{nt} \\ \boldsymbol{\beta}_{nt} \\ \boldsymbol{\gamma}_{nt} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\alpha}_{tn} \\ \boldsymbol{\beta}_{tn} \\ \boldsymbol{\gamma}_{tn} \end{bmatrix} = \boldsymbol{J}^h(\boldsymbol{d}_t^h)\boldsymbol{d}_n^h \tag{55}$$

Let $\boldsymbol{d}_n$ and $\boldsymbol{d}_t$ be vectors obtained by eliminating the entries corresponding to the $V\delta$ bus in $\boldsymbol{d}_n^h$ and $\boldsymbol{d}_t^h$, respectively. Also, let $\boldsymbol{J}_{\text{red}}^h(\hat{\boldsymbol{x}})$ be a matrix resultant from the elimination of the lines in $\boldsymbol{J}^h(\hat{\boldsymbol{x}})$ corresponding to functions not considered in $\boldsymbol{g}(\boldsymbol{x})$. Since the bus complex voltage at the swing bus is pre-specified in the power flow problem, the derivative of $\boldsymbol{g}(\boldsymbol{x})$ with respect to the real and imaginary parts of the voltage at the swing bus is zero. As a consequence, using (55) we have

$$\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_t = \boldsymbol{J}_{\text{red}}^h(\boldsymbol{d}_n^h)\boldsymbol{d}_t^h = \boldsymbol{J}_{\text{red}}^h(\boldsymbol{d}_t^h)\boldsymbol{d}_n^h = \boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_n \tag{56}$$

Equality (56) has been used in the proposed update rule of the modified tensor method.

### 4.2 Proposed algorithm

Observe that, since (15) and (16) hold for any $\boldsymbol{d}$, the first derivative of $\boldsymbol{g}(\boldsymbol{x})$ can be expressed as

$$\boldsymbol{J}(\boldsymbol{x}) = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = -\frac{\partial \boldsymbol{g}_c(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{T}_0 \tag{57}$$

As a consequence, by using (16) and (57), the correction vector introduced in [21], that is $\boldsymbol{d}_t = [\boldsymbol{J}(\boldsymbol{x})]^{-1}\boldsymbol{g}_c(\boldsymbol{d}_n)$, can be written as

$$\boldsymbol{d}_t = -[\boldsymbol{J}(\boldsymbol{x})]^{-1}\frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_n \tag{58}$$

However, by using (16) and (57) in (19), one can compute the second-order Taylor expansion as function of Jacobian matrices, estimates and directions as follows:

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) &= \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{d}_n^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_n + \frac{1}{2}\boldsymbol{d}_n^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{d}_t^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_n \\ &\quad + \frac{1}{2}\boldsymbol{d}_t^{\mathrm{T}}\boldsymbol{T}\boldsymbol{d}_t \\ &= \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_n + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_n \\ &\quad + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_t \end{aligned} \tag{59}$$

and, using (56) we have

$$\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d}) = \boldsymbol{J}(\boldsymbol{x})\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_n + \boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_t + \frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_t \tag{60}$$

By neglecting only the term $(1/2)\boldsymbol{J}(\boldsymbol{d}_t)\boldsymbol{d}_t$, the complementary direction for which $\boldsymbol{g}(\boldsymbol{x} + \boldsymbol{d})$ is a null vector can be computed using the expression

$$\boldsymbol{d}_t = -[\boldsymbol{J}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{d}_n)]^{-1}\frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n)\boldsymbol{d}_n \tag{61}$$

This allows estimating the solution $\boldsymbol{x}$ iteratively according to the update rule

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}_n^{(k)} + \boldsymbol{d}_t^{(k)} \tag{62}$$

where $\boldsymbol{d}_n^{(k)}$ is computed using (11) and

$$\boldsymbol{d}_t^{(k)} = -\left[\boldsymbol{J}(\boldsymbol{x}^{(k)}) + \boldsymbol{J}(\boldsymbol{d}_n^{(k)})\right]^{-1}\frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n^{(k)})\boldsymbol{d}_n^{(k)} \tag{63}$$

The resulting modified tensor-based approach is summarised in Algorithm 1. All steps of the modified tensor-based method and its original counterpart are exhibited in Fig. 1.

*Algorithm 1:* Algorithm procedure

1: Choose an initial estimate $\boldsymbol{x}^{(0)}$;
2: Compute $\boldsymbol{g}(\boldsymbol{x}^{(0)})$;
3: $k = 0$;
4: **while** $\left\|\boldsymbol{g}(\boldsymbol{x}^{(k)})\right\|_\infty \geq$ tolerance **do**
5: Compute the Newton direction:

$$\boldsymbol{d}_n^{(k)} = -\left[\boldsymbol{J}(\boldsymbol{x}^{(k)})\right]^{-1}\boldsymbol{g}(\boldsymbol{x}^{(k)});$$

6: Compute the tensor-based additional direction:

$$\boldsymbol{d}_t^{(k)} = -\left[\boldsymbol{J}(\boldsymbol{x}^{(k)}) + \boldsymbol{J}(\boldsymbol{d}_n^{(k)})\right]^{-1}\frac{1}{2}\boldsymbol{J}(\boldsymbol{d}_n^{(k)})\boldsymbol{d}_n^{(k)};$$

7: Update the estimate:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{d}_n^{(k)} + \boldsymbol{d}_t^{(k)};$$

8: Compute $\boldsymbol{g}(\boldsymbol{x}^{(k+1)})$;
9: $k = k + 1$;
10: **end while**

In the algorithm, as usual, PV buses can be converted in PQ buses if reactive power generation limits are reached during the iterations.

It must be pointed out that the Jacobian matrix is a linear function of $x$, that is $\boldsymbol{J}(\boldsymbol{x}) = \boldsymbol{x}^{\mathrm{T}}\boldsymbol{T}_0$, as stated in (57). Then, the coefficient matrix of the linear system in (61) can be written as

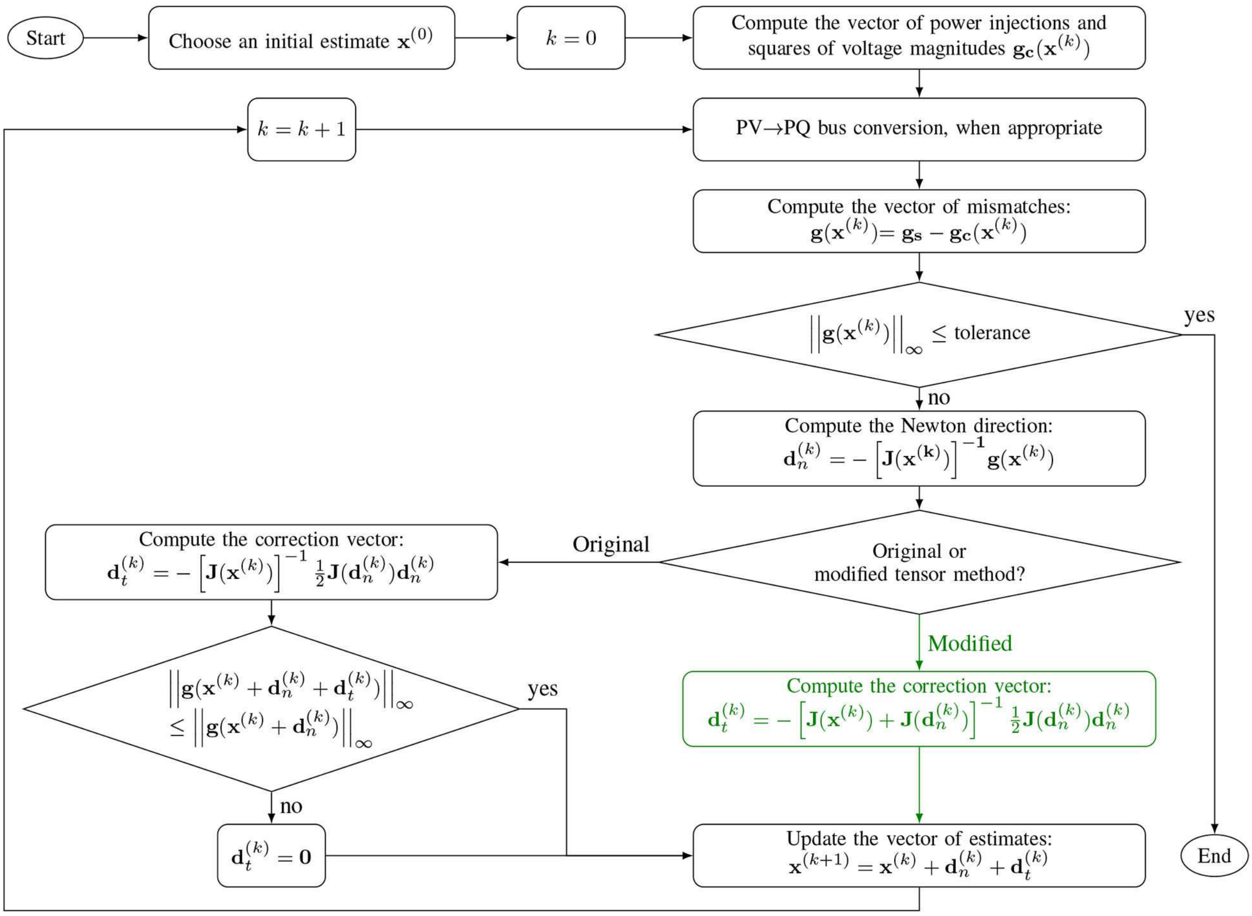



**Fig. 1** *Steps of the modified and original tensor-based methods*

$$\begin{aligned} \boldsymbol{J}(\boldsymbol{x}) + \boldsymbol{J}(\boldsymbol{d}_n) &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{T}_0 + \boldsymbol{d}_n^{\mathrm{T}}\boldsymbol{T}_0 \\ &= (\boldsymbol{x} + \boldsymbol{d}_n)^{\mathrm{T}}\boldsymbol{T}_0 \\ &= \boldsymbol{J}(\boldsymbol{x} + \boldsymbol{d}_n) \end{aligned} \tag{64}$$

where $\boldsymbol{J}(\boldsymbol{d}_n)$ can be seen as an adjustment applied to the matrix $\boldsymbol{J}(\boldsymbol{x})$ and $\boldsymbol{J}(\boldsymbol{x} + \boldsymbol{d}_n)$ represents the Jacobian matrix that would be computed at the next iteration if the Newton direction were used. Since all terms of the Taylor expansion which are linearly dependent of $\boldsymbol{d}_t$ are utilised in the formulation, the application of the adjustment $\boldsymbol{J}(\boldsymbol{d}_n)$ provides a better estimate for $\boldsymbol{d}_t$, improving the search direction towards the power flow solution. This is the main contribution of the proposed method with respect to the one introduced in [21]. That is, from the theoretical point of view, we show the symmetry of the products stated by (56), which is subsequently used to update the Jacobian matrix during each iteration. Additionally, one might notice also that the test in (22) is not utilised in the proposed approach.

## 5 Numerical results

This section introduces a complete example of the application of the proposed algorithm as well as a comparative analysis involving the original tensor method and the proposed approach.

### 5.1 Worked example

Aiming at illustrating the application of the proposed approach, the main steps of the algorithm are described using a 3-bus system whose data and single line diagram are shown in Fig. 2.

- Step 1: A flat start is chosen as initial estimate as follows:

$$\boldsymbol{x}^{(0)} = [1.0000, 1.0000, 0.0000, 0.0000]^{\mathrm{T}}$$

where the first two elements correspond to the real parts of complex voltages at buses 2 and 3, while the remaining elements denote the imaginary parts of complex voltages at the same buses.

- Step 2: Using (8), mismatches can be determined as

$$\boldsymbol{g}(\boldsymbol{x}^{(0)}) = \begin{bmatrix} -0.8000 \\ -0.7500 \\ -0.5000 \\ -0.2500 \end{bmatrix} - \begin{bmatrix} 0.0000 \\ 0.0000 \\ -0.1250 \\ -0.1000 \end{bmatrix} = \begin{bmatrix} -0.8000 \\ -0.7500 \\ -0.3750 \\ -0.1500 \end{bmatrix}$$

corresponding to mismatches in active power at bus 2, active power at bus 3, reactive power at bus 2 and reactive power at bus 3.

- Step 5: The Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ evaluated in $\boldsymbol{x}^{(0)}$ is given by

$$\boldsymbol{J}(\boldsymbol{x}^{(0)}) = \begin{bmatrix} 0 & 0 & -5.1250 & +2.5000 \\ 0 & 0 & +2.5000 & -4.6000 \\ -4.8750 & +2.5000 & 0 & 0 \\ +2.5000 & -4.4000 & 0 & 0 \end{bmatrix}$$

The Newton direction can be computed using (10) as

$$\boldsymbol{d}_n^{(0)} = [-0.1332, -0.1098, -0.3206, -0.3373]^{\mathrm{T}}$$

- Step 6: The Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ evaluated in $\boldsymbol{d}_n^{(0)}$ can be computed as

$$\boldsymbol{J}(\boldsymbol{d}_n^{(0)}) = \begin{bmatrix} -0.8433 & +0.8016 & +0.2745 & -0.3331 \\ +0.8433 & -0.8016 & -0.2745 & +0.3331 \\ +1.0578 & -0.3331 & +2.3631 & -0.8016 \\ -0.2745 & +0.6550 & -0.8433 & +2.2341 \end{bmatrix}$$

The complementary tensor-based direction can be computed using (63) as

$$\boldsymbol{d}_t^{(0)} = [-0.1743,\ -0.1766,\ +0.0039,\ -0.0049]^{\mathrm{T}}$$

- Step 7: Using (62), the solution estimate is updated as follows:

$$\boldsymbol{x}^{(1)} = [+0.6925,\ +0.7136,\ -0.3167,\ -0.3422]^{\mathrm{T}}$$

- Step 8: The updated mismatches are computed using (8) as

$$\boldsymbol{g}(\boldsymbol{x}^{(1)}) = \begin{bmatrix} -0.8000 \\ -0.7500 \\ -0.5000 \\ -0.2500 \end{bmatrix} - \begin{bmatrix} -0.8039 \\ -0.7461 \\ -0.4249 \\ -0.1864 \end{bmatrix} = \begin{bmatrix} +0.0039 \\ -0.0039 \\ -0.0751 \\ -0.0636 \end{bmatrix}$$

providing a maximum absolute mismatch value of 0.0751. The application of the tensor direction as stated in [21] would result in a maximum absolute mismatch value of 0.1219.

It is noteworthy to mention that the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{x})$ evaluated in $\boldsymbol{d}_t^{(0)}$ is given by

$$\boldsymbol{J}(\boldsymbol{d}_t^{(0)}) = \begin{bmatrix} -0.0123 & -0.0098 & +0.4416 & -0.4358 \\ +0.0123 & +0.0098 & -0.4416 & +0.4358 \\ +1.3014 & -0.4358 & -0.0516 & +0.0098 \\ -0.4416 & +1.1539 & -0.0123 & +0.0541 \end{bmatrix}$$

As a consequence, the matrix–vector products

$$\boldsymbol{J}(\boldsymbol{d}_n^{(0)})\boldsymbol{d}_t^{(0)} = [+0.0081,\ -0.0081,\ -0.1123,\ -0.0822]^{\mathrm{T}}$$

$$\boldsymbol{J}(\boldsymbol{d}_t^{(0)})\boldsymbol{d}_n^{(0)} = [+0.0081,\ -0.0081,\ -0.1123,\ -0.0822]^{\mathrm{T}}$$

provide the same result, as proven in Section 4.1.

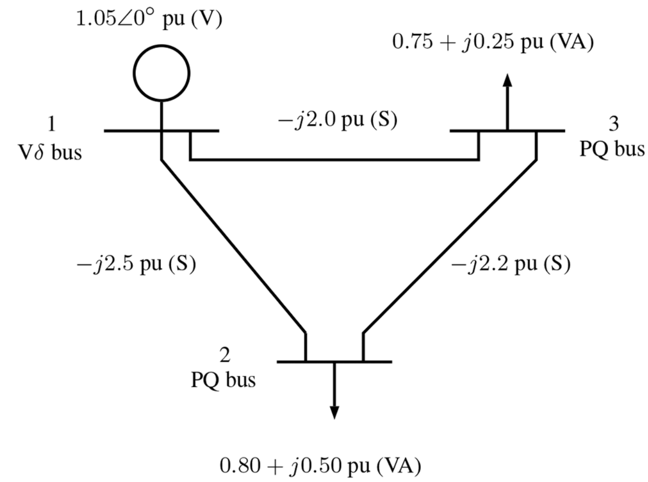



**Fig. 2** *Worked example: 3-bus test system*

### *5.2 Comparative analysis*

Several analyses have been performed using test systems [24] and an actual Brazilian system. All results have been validated with the MATPOWER tool [25]. Although comparisons with the Newton–Raphson method have been fully stressed in [21], an initial comparative analysis is provided. Tables 1 and 2 show results for a MATPOWER test system with 1354 buses, corresponding to part of the European high voltage transmission network [26], as well as for the IEEE 14-bus and IEEE 300-bus test systems. Load factors are used as multiplying factors to power generation and demand, meaning that both bus power generation and demand are multiplied by the indicated load factor as follows:

$$P_{g_i} = \rho P_{g_i}^0 \tag{65}$$

$$P_{d_i} = \rho P_{d_i}^0 \tag{66}$$

$$Q_{g_i} = \rho Q_{g_i}^0 \tag{67}$$

$$Q_{d_i} = \rho Q_{d_i}^0 \tag{68}$$

where $\rho$ is the load factor; $P_{g_i}, P_{d_i}, Q_{g_i}, Q_{d_i}$ denote the active power generation, active power demand, reactive power generation and reactive power demand at bus $i$, respectively; $P_{g_i}^0, P_{d_i}^0, Q_{g_i}^0, Q_{d_i}^0$ are the active power generation, active power demand, reactive power generation and reactive power demand at bus $i$, respectively, given in [24, 25].

The acronyms MTM, TM and NR are used to indicate the modified tensor method, tensor method and Newton–Raphson method (implemented in MATPOWER), respectively. Aiming at comparing the effect of the tensor-based correction vectors, reactive power limits have been neglected. All analyses have been performed assuming a tolerance of $10^{-8}$, a flat start and a maximum number of iterations of 50.

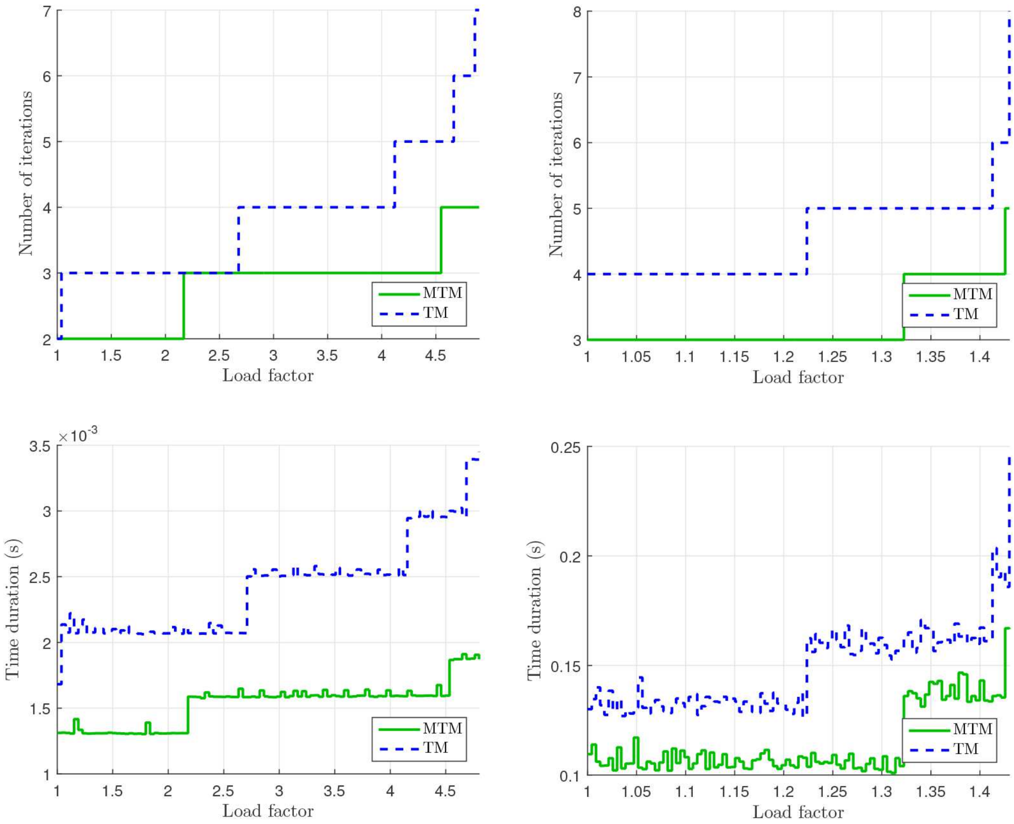
The load factors in Table 1 vary from 1.000 to 1.528, corresponding, respectively, to the base case provided in the data set and the maximum loadability factor. Columns 2–4 show the number of iterations required for convergence, whilst columns 5–7 indicate the active power transmission losses. The results point out that the correction vector improves the iterative processes, reducing the number of iterations for convergence. As an example of CPU computations, using a NR method implemented by excluding the correction vector from the analysis, the NR, TM and MTM endured 8.60, 9.38 and 6.05 s, respectively, for the base case. Although the time per iteration required by the NR, TM and MTM is 1.72, 1.88, 2.02, respectively, the total time durations indicate a positive effect of using the correction vector. It is important to mention that all time computations depend considerably on the level of effort directed towards implementation. As a matter of fact, the NR implemented in MATPOWER is significantly faster than all methods implemented herein. Finally, the active power transmission losses shown in columns 5–7 point out that all methods provided adequate solutions to the power flow problem. A similar conclusion can be withdrawn from the analysis of other systems, as exemplified in Table 2, where the results for the minimum bus voltage (in column 3), maximum bus voltage (in column 4) and active power transmission losses (in column 5) computed for the IEEE 14-bus and IEEE 300-bus test systems are shown.

**Table 1** Comparative analysis with a 1354-bus network

| Load factor | No. of iterations | | | $P_{\mathrm{losses}}$, MW | | |
|---|---|---|---|---|---|---|
| | NR | TM | MTM | NR | TM | MTM |
| 1.000 | 5 | 5 | 3 | 16.635 | 16.636 | 16.636 |
| 1.330 | 5 | 5 | 4 | 31.280 | 31.282 | 31.282 |
| 1.525 | 8 | 7 | 5 | 44.096 | 44.098 | 44.098 |
| 1.528 | 10 | 9 | 6 | 44.419 | 44.421 | 44.421 |

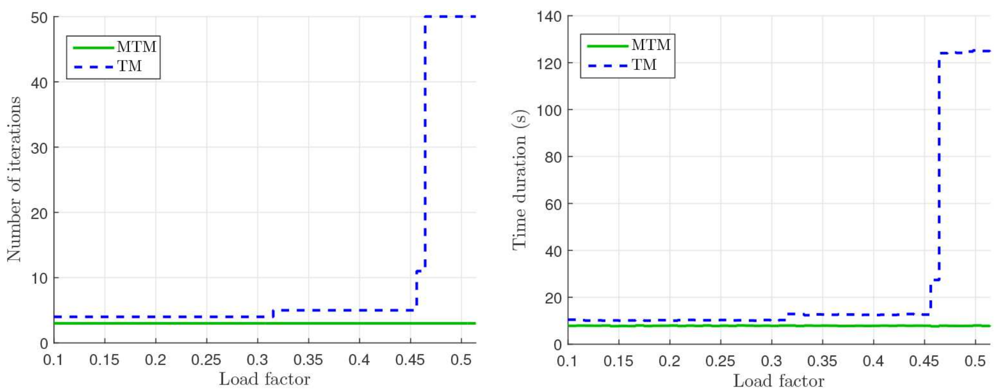
**Table 2** Comparative analysis with the IEEE 14-bus and 300-bus network

| Test system | Method | $V_{\min}$, p.u. | $V_{\max}$, p.u. | $P_{\text{losses}}$, MW |
|---|---|---|---|---|
| IEEE 14-bus | NR | 1.010 | 1.090 | 13.393 |
| | TM | 1.010 | 1.090 | 13.393 |
| | MTM | 1.010 | 1.090 | 13.393 |
| IEEE 300-bus | NR | 0.929 | 1.073 | 408.316 |
| | TM | 0.929 | 1.073 | 408.316 |
| | MTM | 0.929 | 1.073 | 408.316 |




a b

**Fig. 3** *Comparison analysis between tensor-based approaches for the IEEE 14-bus and the IEEE 300-bus test system*
***(a)*** IEEE 14-bus – no. of iterations and elapsed time, ***(b)*** IEEE 300-bus – no. of iterations and elapsed time




a b

**Fig. 4** *Comparison analysis between tensor-based approaches for an actual Brazilian system*
***(a)*** No. of iterations, ***(b)*** Elapsed time

In order to compare the proposed approach with its original counterpart, Fig. 3 shows the number of iterations and computation time of power flow analyses using different load factors for the IEEE 14-bus and the IEEE 300-bus test systems, respectively, for the TM and MTM. Fig. 4 depicts the same information for a Brazilian system with 1916 buses, while Fig. 5 illustrates the effect of the selection of initial estimates on the analysis of the IEEE 300-bus test system. The latter figure exhibits the mismatches per iteration of the TM and MTM for different initial estimates. It is shown that the performance of the MTM depends on the selection of the initial estimate, similarly to the TM. Initial estimates near the solution are expected to require a small number of iterations from both methods.

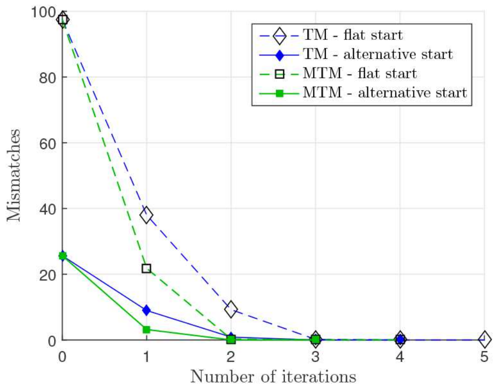



**Fig. 5** *Effect of the improved correction vector for different initial estimates in the analysis of the IEEE 300-bus test system*

All result analysis indicates that the proposed approach is able to converge in less or equal number of iterations than the original tensor method, providing reduced elapsed times. This outcome is expected since the correction vector of the proposed approach takes into account additional terms of the second-order expansion of $\boldsymbol{g}(\boldsymbol{x})$, providing an improved direction estimate in comparison with the original tensor method. There are cases with high load factors in which the proposed approach reaches a solution while the original tensor method diverges, as exemplified in Fig. 4. This outcome is again justified by the use of the improved search direction. At last, we verified that the mismatch test in (22) is relevant to promote the convergence of the original tensor method. For instance, in the analyses of the IEEE 300-bus, there are several factors in [1.08, 1.30] for which the original tensor method is only able to achieve convergence by performing the test and disregarding the correction vector at least once. Conversely, the proposed approach provides an elegant and reliable direction, avoiding the need for the aforementioned testing.

## 6 Conclusions and final remarks

Tensor-based approaches utilises the second-order Taylor series expansion of the power flow equations to compute a correction vector to be added to the incremental variation of the Newton-Rahpson method. The proposed approach employs an improved estimation of the correction vector by adjusting the Jacobian matrix using the incremental variation of the Newton-Rahpson method. This allows to exploit the information brought by the second-order term in a more complete way.

Result analysis indicates that the proposed approach provides a solution to the power flow equations in less or equal iterations and reduced computation time than its original version. In particular cases with high loading, the approach is shown to be more robust, achieving convergence without requiring a test whether the application of the correction vector will indeed reduce mismatches.

Ill-conditioned cases can be further addressed by estimating the correction vector, adding this vector to the incremental variation of the Newton–Raphson method and determining the optimum step size to be applied [14, 16]. Future works will address the application of the modified tensor direction in the determination of the bifurcation points of the power flow equations through continuation power flow and optimisation-based methods. Also, an extension to address the analysis of three-phase power distribution networks is envisioned.

## 7 Acknowledgments

This work was supported by the Coordination for the Improvement of Higher Education Personnel (CAPES) of Brazil and the Institute for Systems and Computer Engineering, Research and Development of Brazil (INESC P&D Brasil). The work was developed in Project TECCON II with Transmissoras Brasileiras de Energia (TBE), under the framework of the Research and Development (R&D) Program regulated by Agência Nacional de Energia Elétrica (ANEEL), code PD-2615-0011-2015.

## 8 References

[1] Tinney, W.F., Hart, C.E.: 'Power flow solution by Newton's method', *IEEE Trans. Power Appar. Syst.*, 1967, **PAS-86**, (11), pp. 1449–1460
[2] Stott, B., Alsac, O.: 'Fast decoupled load flow', *IEEE Trans. Power Appar. Syst.*, 1974, **PAS-93**, (3), pp. 859–869
[3] Fatemi, S.M., Abedi, S., Gharehpetian, G.B., *et al.*: 'Introducing a novel dc power flow method with reactive power considerations', *IEEE Trans. Power Syst.*, 2015, **30**, (6), pp. 3012–3023
[4] Derakhshandeh, S.Y., Pourbagher, R.: 'Application of high-order Newton-like methods to solve power flow equations', *IET Gener. Transm. Distrib.*, 2016, **10**, (8), pp. 1853–1859
[5] Pourbagher, R., Derakhshandeh, S.Y.: 'Application of high-order Levenberg–Marquardt method for solving the power flow problem in the ill-conditioned systems', *IET Gener. Transm. Distrib.*, 2016, **10**, (12), pp. 3017–3022
[6] Liao, X., Liu, K., Zhang, Y., *et al.*: 'Interval method for uncertain power flow analysis based on taylor inclusion function', *IET Gener. Transm. Distrib.*, 2017, **11**, (5), pp. 1270–1278
[7] Exposito, A.G., Ramos, E.R.: 'Augmented rectangular load flow model', *IEEE Trans. Power Syst.*, 2002, **17**, (2), pp. 271–276
[8] Kulworawanichpong, T.: 'Simplified Newton–Raphson power-flow solution method', *Int. J. Electr. Power Energy Syst.*, 2010, **32**, (6), pp. 551–558
[9] Moura, A.P., Moura, A.A.F.: 'Newton–Raphson power flow with constant matrices: a comparison with decoupled power flow methods', *Int. J. Electr. Power Energy Syst.*, 2013, **46**, pp. 108–114
[10] Gómez-Expósito, A., Gómez-Quiles, C.: 'Factorized load flow', *IEEE Trans. Power Syst.*, 2013, **28**, (4), pp. 4607–4614
[11] Sachdev, M.S., Medicherla, T.K.P.: 'A second order load flow technique', *IEEE Trans. Power Appar. Syst.*, 1977, **96**, (1), pp. 189–197
[12] Mokhlis, H., Shahriari, A., Laghari, J.: 'Fast and accurate second order load flow method based on fixed jacobian matrix', *Appl. Math. Comput.*, 2015, **269**, pp. 584–593
[13] Iwamoto, S., Tamura, Y.: 'A fast load flow method retaining nonlinearity', *IEEE Trans. Power Appar. Syst.*, 1978, **PAS-97**, (5), pp. 1586–1599
[14] Iwamoto, S., Tamura, Y.: 'A load flow calculation method for ill-conditioned power systems', *IEEE Trans. Power Appar. Syst.*, 1981, **PAS-100**, (4), pp. 1736–1743
[15] Rao, P.S.N., Rao, K.S.P., Nanda, J.: 'An exact fast load flow method including second order terms in rectangular coordinates', *IEEE Trans. Power Appar. Syst.*, 1982, **PAS-101**, (9), pp. 3261–3268
[16] Ferreira, C.A., Costa, V.M.: 'A second order power flow based on current injection equations', *Int. J. Electr. Power Energy Syst.*, 2005, **27**, (4), pp. 254–263
[17] Bouaricha, A., Schnabel, R.B.: 'Tensor methods for large sparse systems of nonlinear equations', *Math. Program.*, 1998, **82**, (3), pp. 377–400
[18] Schnabel, R.B., Frank, P.D.: 'Tensor methods for nonlinear equations', *SIAM J. Numer. Anal.*, 1984, **21**, (5), pp. 815–843
[19] Portela, C.: '*Análise de redes elétricas. Algumas aplicações*' (Instituto de Alta Cultura Técnica, Lisbon, Portugal, 1970)
[20] Portela, C.: 'Tensorial formulation of some network problems - applications to network analysis and optimization'. Proc. of the Power System Computer Conf., Grenoble, France, September 1972, p. 1
[21] Salgado, R.S., Zeitune, A.F.: 'Power flow solutions through tensor methods', *IET Gener. Transm. Distrib.*, 2009, **3**, (5), pp. 413–424
[22] Salgado, R.S., Moraes, G., Issicaba, D.: 'Determination of the bifurcation points of the power flow equations through optimisation-based methods', *Electr. Power Syst. Res.*, 2018, **158**, pp. 147–157
[23] Salgado, R.S., Zeitune, A.F.: 'A direct method based on tensor calculation to determine maximum loadability power flow solutions', *Electr. Power Syst. Res.*, 2013, **103**, (C), pp. 114–121
[24] Washington, U.: 'Power systems test case archive', Available at https://www2.ee.washington.edu/research/pstca/pf14/pg_tca14bus.htm, accessed on May 2018
[25] Zimmerman, R.D., Murillo-Sanchez, C.E., Thomas, R.J.: 'Matpower: steady-state operations, planning, and analysis tools for power systems research and education', *IEEE Trans. Power Syst.*, 2011, **26**, (1), pp. 12–19
[26] Josz, C., Fliscounakis, S., Maeght, J., *et al.*: 'AC power flow data in MATPOWER and QCQP format: iTesla, RTE snapshots, and PEGASE', 2016, Available at http://arxiv.org/abs/1603.01533, accessed on Jan 2006