IET Generation, Transmission & Distribution

Research Article

Development of combined Runge–Kutta Broyden's load flow approach for well- and ill-conditioned power systems

ISSN 1751-8687
Received on 22nd May 2018
Revised 2nd September 2018
Accepted on 21st September 2018
E-First on 22nd October 2018
doi: 10.1049/iet-gtd.2018.5633
www.ietdl.org

Marcos Tostado-Véliz[1], *Salah Kamel*[2,3], *Francisco Jurado*[1] ✉
[1]*Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén, Spain*
[2]*Department of Electrical Engineering, Faculty of Engineering, Aswan University, 81542 Aswan, Egypt*
[3]*State Key Laboratory of Power Transmission Equipment & System Security and New Technology, Chongqing University, Chongqing 400030, People's Republic of China*
✉ *E-mail: fjurado@ujaen.es*

Abstract: Load flow (LF) is an extensively used tool in planning and operation of power systems. Formulation of LF problem can be assimilated as a set of autonomous ordinary differential equations, therefore, many numeric methods can be used to solve this problem. However, LF methods often need to compute one or more Jacobian matrix inversions in each iteration. Owing to this fact, these methods might not be computationally efficient. In this study, the authors propose combined Runge–Kutta Broyden's LF (RK4B) method in order to reduce the required Jacobian matrix inversion to only one in the first iteration. In this proposed method, Broyden's approach is employed in fourth-order Runge–Kutta method. In addition, two modifications of the proposed method are presented to reduce the number of iterations and improve the computational performance. The proposed method and the two modifications are validated using several well- and ill-conditioned cases. Results show that the combined approach has better computational performance than the classical multistage numeric methods, besides it preserves the robustness features of fourth-order Runge–Kutta method.

1 Introduction

Load flow (LF) analysis is the most important and essential tool for operating and planning power systems [1, 2]. Newton–Raphson (NR) LF method can be considered as the standard method that applied in industrial applications [3]. It has good convergence characteristics and an acceptable robustness. However, NR method started to lose its ability to converge fast with dramatically increasing in power systems size. In addition, the Jacobian elements of NR method have to be updated during the iterative process. In order to overcome these drawbacks, fast-decoupled-load-flow method was introduced in [4]. This method based on a simplified Jacobian matrix achieving fast convergence but normally employing more iterations. A linear version of the LF problem was proposed in [5]. Since it is based on a linear formulation of the problem, several issues related to the inversion of the Jacobian matrix have been avoided. Moreover, the reactive power problem has not been considered. Recent techniques have been proposed to consider the reactive power balance in the problem [6, 7]. Apart from the aforementioned methods, other LF formulation and techniques have been proposed over decades [8–13].

The performance of a LF method is strongly linked to the condition of the system which depends on the following three factors [14]:

i. Heavy loading at some buses.
ii. High ratios of lines *R*/*X*.
iii. The initial guess point is outside of the region of attraction.

Ill-conditioned cases may provoke convergence problems in most of standard LF methods [15]. Consequently, several robust LF methods have been developed in order to solve the ill-conditioned systems. Robust LF methods may be broadly classified as second-order methods, methods based on the continuous Newton's method (CN) and other based on the Levenberg–Marquardt method (LM).

The second-order methods employ a second-order Taylor series expansion to formulate the LF problem, instead of the most typical first-order formulation employed by the NR. Iwamoto and Tamura [16] have presented Iwamoto's method (IM) which is based on second-order Taylor series. In this method, an optimal multiplier is calculated in each iteration which modifies the corrector vector in order to avoid the divergence. This method has outstanding robustness features. However, its convergence is normally very slow due to the optimal multiplier frequently tend to reduce the corrector vector and brake the convergence. In fact, it usually fails due to an excessive number of iterations, and it rarely diverges. The IM can be considered as the most standard second-order LF method as it has been widely referenced in the literature [13, 17], nevertheless, other second-order techniques have been developed [18–22].

The CN was introduced in [23] and, later, it was adapted for solving the LF problem after reformulating to a set of autonomous ordinary differential equations (ODE) [17]. The techniques that based on the CN show good performance in large-scale systems.

Since numerical methods often show multistage procedures, these techniques require a high number of Jacobian matrix inversions during the LF solution.

In [24], the LM has been applied for solving the LF problem. However, this method has several convergence difficulties such as the rate of convergence depends hugely of the parameter λ, and the convergence rate tends to grow with λ. However, the convergence becomes unsafe since the final solution is not accurate enough if λ is high. With the aim to overcome these difficulties, high-order-Levenberg–Marquardt method (HOLM) has been proposed in [25]. This method has good performance in ill-conditioned systems. However, its performance depends on a set of prespecified parameters, which must be carefully chosen.

In [17], RK4 has been presented for solving the LF problem. Although NR employs one matrix inversion in each iteration, RK4 needs to calculate up to four matrix inversions in each iteration. In addition, the required number of iterations in case of RK4 is normally higher than NR. Owing to these facts, large number of Jacobian matrix inversions must be calculated in case of RK4 compared with NR. It is well known that the heaviest computational part of a LF iteration is the inversion of Jacobian matrix [17]. As an example, the distribution of computational time for a NR iteration in IEEE 118-bus system is shown in Fig. 1. From

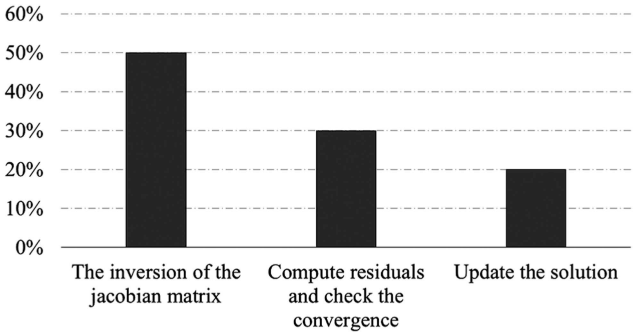

Fig. 1 *Distribution of computational time for one NR iteration in the IEEE 118-bus system*

this figure, it can be observed that Jacobian matrix inversion is the main computational burden of any Newton-like method. Obviously, this drawback is increased in multistage methods like RK4.

In this paper, a novel methodology based on the combination of RK4 and Broyden's method (BM) is proposed. However, the main contributions of this paper can be summarised as follows:

- Development of efficient LF methods for solving well- and ill-conditioned power systems;
- New LF methodology (RK4B) based on a combination between RK4 and BM has been proposed to reduce the required number of Jacobian matrix inverse to be only one in the first iteration;
- Using the proposed methodology, the computational burden of RK4 method has been addressed;
- Proposing two modifications of this methodology to improve the convergence characteristics and reduce the computation time;
- In both modifications, the required number of Jacobian matrix inversions has been reduced to ≤25% of this required in case of RK4;
- The effectiveness of the proposed methodology has been validated using different well- and ill-conditioned test systems;
- The obtained results proved that the proposed Scheme 1 and Scheme 2 give low number of iterations and computation time compared with the other LF methods. Consequently, both proposed schemes are reliable in solving the well- and ill-conditioned systems.

The rest of the paper is organised as follows. Overview of LF problem formulation and CN method are presented in Section 2. The proposed methodology and the two modifications are presented in Section 3. Simulations and results are presented in Section 4. Finally, the main conclusions are presented in Section 5.

2 LF problem

LF raises the non-linear relationship between the nodal voltages and power injections. Let us outline the main features of the LF problem and its solution using NR method. First, the active and reactive nodal power mismatches are given as follows [26]:

$$\Delta P_i = P^{\text{sch}} - \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\cos(\delta_i - \delta_j - \theta_{ij}) \tag{1}$$

$$\Delta Q_i = Q^{\text{sch}} - \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\sin(\delta_i - \delta_j - \theta_{ij}) \tag{2}$$

where ΔP_i and ΔQ_i denote the active and reactive power mismatches at bus i, respectively, P^{sch} and Q^{sch} denote the injected active and reactive power at bus i, respectively, $V_i\angle\delta_i$ is the complex voltage at bus i, $Y_{ij}\angle\theta_{ij}$ is the ijth element of admittance matrix and n is the total number of buses.

The state vector of the LF problem is formed by the voltage angles at PV and PQ buses and voltage magnitudes at PQ buses as

$$\boldsymbol{x} = \{\boldsymbol{\delta}_{\text{PV}} \cup \boldsymbol{\delta}_{\text{PQ}} \cup \boldsymbol{V}_{\text{PQ}}\} \tag{3}$$

where $\boldsymbol{\delta}_{\text{PV}}$ and $\boldsymbol{\delta}_{\text{PQ}}$ are the voltage angle vectors of PV and PQ buses, respectively, and $\boldsymbol{V}_{\text{PQ}}$ is the voltage magnitude vector of PQ buses.

In order to simplify the notation, compact versions of (1) and (2) can be used onwards

$$\boldsymbol{g}(\boldsymbol{x}) = 0 \tag{4}$$

Since (4) is non-linear, a numerical technique (e.g. iterative methods, homotopy approaches etc) must be used for solving them. NR is considered the most standard LF method. A generic kth NR iteration for solving the LF can be defined as

$$\begin{aligned} \Delta\boldsymbol{x}^{(k)} &= -\boldsymbol{J}_x\left(\boldsymbol{x}^{(k)}\right)^{-1}\boldsymbol{g}(\boldsymbol{x}^{(k)}) \\ \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}^{(k)} \end{aligned} \tag{5}$$

where $\boldsymbol{J}_x$ is the Jacobian matrix of the system, which is formed from the first-order partial derivatives of (4) with respect to the state vector and the superscript k is the current iteration. LF algorithm stops if the following condition is satisfied:

$$\max(\text{abs}[\boldsymbol{g}(\boldsymbol{x})]) \le \epsilon \tag{6}$$

where ϵ is a preset convergence parameter. Expression (6) means if the maximum value of (4) is smaller than ϵ, it can be concluded that the algorithm is converged. Typically, the number of iterations is limited ($k_{\max}$). If the algorithm is not converged before reaching to the maximum number of iterations, the LF procedure stops. If it occurs, it is said that the algorithm has failed to find the solution.

2.1 Continuous Newton's method

In this subsection, CN methodology is briefly explained. Equation (7) presents a set of ODE

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{7}$$

The explicit Euler method is considered the simplest method for integrating (7) as

$$\begin{aligned} \Delta\boldsymbol{x}^{(k)} &= \Delta t\,\boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) \\ \boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \Delta\boldsymbol{x}^{(k)} \end{aligned} \tag{8}$$

where Δt is the time step. An analogy between (6) and (8) is easily established if one defines

$$\boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) = -\boldsymbol{J}_x\left(\boldsymbol{x}^{(k)}\right)^{-1}\boldsymbol{g}(\boldsymbol{x}^{(k)}) \tag{9}$$

On the basis of this analogy, the NR iteration can be viewed as the explicit Euler method with $\Delta t = 1$. Given this statement, any well-assessed numerical method can be successfully adapted for solving the LF. In [17], it was demonstrated that the CN converges asymptotically to the solution if the initial guess exists inside of the so-called region of attraction.

3 Proposed methodology

New LF methodology (RK4B) based on a combination between RK4 and BM is proposed to improve the performance of the RK4 and reduce the required number of Jacobian matrix inverse to be only one in the first iteration. This methodology is able to efficiently solve the well- and ill-conditioned power systems with fast convergence and low computation time, where the required number of Jacobian inversions is reduced.

3.1 Broyden's method

BM is a quasi-Newton method that can be used to find roots of non-linear equations [27]. In this method, Jacobian matrix and its inverse are calculated only one time at the first iteration. The iterative procedure of BM can be summarised asfollows.

At $k = 0$

$$\boldsymbol{A}^{(0)} = \boldsymbol{J}_x\left(\boldsymbol{x}^{(0)}\right)^{-1} \tag{10}$$

For $k \geq 1$

$$\boldsymbol{J}_x\left(\boldsymbol{x}^{(k)}\right)^{-1} \simeq \boldsymbol{A}^{(k)} = \boldsymbol{A}^{(k-1)} + \frac{\left(\boldsymbol{s}^{(k)} - \boldsymbol{A}^{(k-1)}\boldsymbol{y}^{(k)}\right)\boldsymbol{s}^{(k)^{\mathrm{T}}}\boldsymbol{A}^{(k-1)}}{\boldsymbol{s}^{(k)^{\mathrm{T}}}\boldsymbol{A}^{(k-1)}\boldsymbol{y}^{(k)}} \tag{11}$$

where $\boldsymbol{s}$ and $\boldsymbol{y}$ at kth iteration can be computed as

$$\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}^{(k-1)} \tag{12}$$

$$\boldsymbol{y}^{(k)} = \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) - \boldsymbol{g}\left(\boldsymbol{x}^{(k-1)}\right) \tag{13}$$

It can be easily noticed that Jacobian matrix inverse is only explicitly computed one time in the first iteration. In the following iterations, an approximate version of Jacobian matrix inverse is calculated through vectors and matrix sums and products which can be efficiently executed. It may be assumed that these computations have not a relevant impact in the overall performance of the algorithm [17].

3.2 Development of a combined LF methodology based on RK4 and BM

It is worth to mention that the proposed methodology can be easily applied to other numerical methods such as Ralston method and Runge–Kutta–Fehlberg's method. In this paper, the proposed methodology is just adapted to the RK4.

A generic kth iteration of RK4 when it is applied to LF problem can be defined as [17]

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right)\\
\boldsymbol{x}_2 &= \boldsymbol{x}^{(k)} + \frac{h}{2}\boldsymbol{k}_1\\
\boldsymbol{k}_2 &= \boldsymbol{f}(\boldsymbol{x}_2)\\
\boldsymbol{x}_3 &= \boldsymbol{x}^{(k)} + \frac{h}{2}\boldsymbol{k}_2\\
\boldsymbol{k}_3 &= \boldsymbol{f}(\boldsymbol{x}_3)\\
\boldsymbol{x}_4 &= \boldsymbol{x}^{(k)} + \boldsymbol{k}_3\\
\boldsymbol{k}_4 &= \boldsymbol{f}(\boldsymbol{x}_4)\\
\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2 + 2\boldsymbol{k}_3 + \boldsymbol{k}_4)
\end{aligned} \tag{14}$$

where h is the step size. The step size in RK4 can be updated during the iterative process based on half-step method as [17]

$$\begin{aligned}
&\zeta = \max\left\{\mathrm{abs}\left(\boldsymbol{k}_2 - \boldsymbol{x}^{(k+1)}\right)\right\}\\
&\text{if } \zeta > 0.01 \text{ then } h \leftarrow \max\{0.985h, 0.75\}\\
&\text{if } \zeta \leq 0.01 \text{ then } h \leftarrow \max\{1.015h, 0.75\}
\end{aligned} \tag{15}$$

From the above equations, it can be observed that the RK4 needs to invert the Jacobian matrix four times in each iteration.

In the proposed fourth-order Runge–Kutta–Broyden's methodology (RK4B) methodology, the main benefits of BM are exploited to improve the performance of RK4 in solving LF problem.

In the RK4B, (9) must be modified as

$$\boldsymbol{f}^*(\boldsymbol{x}_s) = -\boldsymbol{A}_s^{(k)}\boldsymbol{g}(\boldsymbol{x}_s^{(k)}) \tag{16}$$

where s is the stage of RK4B method, i.e. $s \in [1, 2, 3, 4]$. In each stage, $\boldsymbol{A}_s$ is calculated using the information of the current and previous stages, except at stage 1, when the information of the last stage at the previous iteration is employed instead. This procedure can be summarised as follows.

For $s = 1$

$$\boldsymbol{A}_1^{(k)} = \boldsymbol{A}_4^{(k-1)} + \frac{\left(\boldsymbol{s}_1^{(k)} - \boldsymbol{A}_4^{(k-1)}\boldsymbol{y}_1^{(k)}\right)\boldsymbol{s}_1^{(k)^{\mathrm{T}}}\boldsymbol{A}_4^{(k-1)}}{\boldsymbol{s}_1^{(k)^{\mathrm{T}}}\boldsymbol{A}_4^{(k-1)}\boldsymbol{y}_1^{(k)}} \tag{17}$$

$$\boldsymbol{s}_1^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{x}_4^{(k-1)} \tag{18}$$

$$\boldsymbol{y}_1^{(k)} = \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) - \boldsymbol{g}(\boldsymbol{x}_4^{(k-1)}) \tag{19}$$

For $s \in [2, 3, 4]$

$$\boldsymbol{A}_s^{(k)} = \boldsymbol{A}_{s-1}^{(k)} + \frac{\left(\boldsymbol{s}_s^{(k)} - \boldsymbol{A}_{s-1}^{(k)}\boldsymbol{y}_s^{(k)}\right)\boldsymbol{s}_s^{(k)^{\mathrm{T}}}\boldsymbol{A}_{s-1}^{(k)}}{\boldsymbol{s}_s^{(k)^{\mathrm{T}}}\boldsymbol{A}_{s-1}^{(k)}\boldsymbol{y}_s^{(k)}} \tag{20}$$

$$\boldsymbol{s}_s^{(k)} = \boldsymbol{x}_s^{(k)} - \boldsymbol{x}_{s-1}^{(k)} \tag{21}$$

$$\boldsymbol{y}_s^{(k)} = \boldsymbol{g}\left(\boldsymbol{x}_s^{(k)}\right) - \boldsymbol{g}(\boldsymbol{x}_{s-1}^{(k)}) \tag{22}$$

BM only uses (9) at the first iteration, while (16) is used for the remainder iterations. Thus, the required matrix inversions are reduced to only one. Therefore, the general procedure of the proposed RK4B can be summarised as follows.

At $k = 0$

$$\begin{aligned}
\boldsymbol{k}_1 &= \boldsymbol{f}\left(\boldsymbol{x}^{(0)}\right)\\
\boldsymbol{x}_2^{(0)} &= \boldsymbol{x}^{(0)} + \frac{h}{2}\boldsymbol{k}_1\\
\boldsymbol{k}_2^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_2^{(0)}\right)\\
\boldsymbol{x}_3^{(0)} &= \boldsymbol{x}^{(0)} + \frac{h}{2}\boldsymbol{k}_2^*\\
\boldsymbol{k}_3^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_3^{(0)}\right)\\
\boldsymbol{x}_4^{(0)} &= \boldsymbol{x}^{(0)} + h\boldsymbol{k}_3^*\\
\boldsymbol{k}_4^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_4^{(0)}\right)\\
\boldsymbol{x}^{(1)} &= \boldsymbol{x}^{(0)} + \frac{h}{6}(\boldsymbol{k}_1 + 2\boldsymbol{k}_2^* + 2\boldsymbol{k}_3^* + \boldsymbol{k}_4^*)
\end{aligned} \tag{23}$$

For $k \geq 1$

$$\begin{aligned}
\boldsymbol{k}_1^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_1^{(k)}\right)\\
\boldsymbol{x}_2^{(k)} &= \boldsymbol{x}^{(k)} + \frac{h}{2}\boldsymbol{k}_1^*\\
\boldsymbol{k}_2^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_2^{(k)}\right)\\
\boldsymbol{x}_3^{(k)} &= \boldsymbol{x}^{(k)} + \frac{h}{2}\boldsymbol{k}_2^*\\
\boldsymbol{k}_3^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_3^{(k)}\right)\\
\boldsymbol{x}_4^{(k)} &= \boldsymbol{x}^{(k)} + h\boldsymbol{k}_3^*\\
\boldsymbol{k}_4^* &= \boldsymbol{f}^*\left(\boldsymbol{x}_4^{(k)}\right)\\
\boldsymbol{x}^{(k+1)} &= \boldsymbol{x}^{(k)} + \frac{h}{6}(\boldsymbol{k}_1^* + 2\boldsymbol{k}_2^* + 2\boldsymbol{k}_3^* + \boldsymbol{k}_4^*)
\end{aligned} \tag{24}$$

For the sake of clarify, a flowchart of the proposed RK4B methodology is shown in Fig. 2.

3.3 Two modifications for RK4B

Despite that RK4B reduces the computational burden of the RK4, the number of iterations is still high. In order to address this issue, two modifications for RK4B are proposed. These modifications improve the performance of RK4B and reduce the required number of iterations compared with RK4. The first proposed modification is based on starting the iterative procedure with RK4 and switching

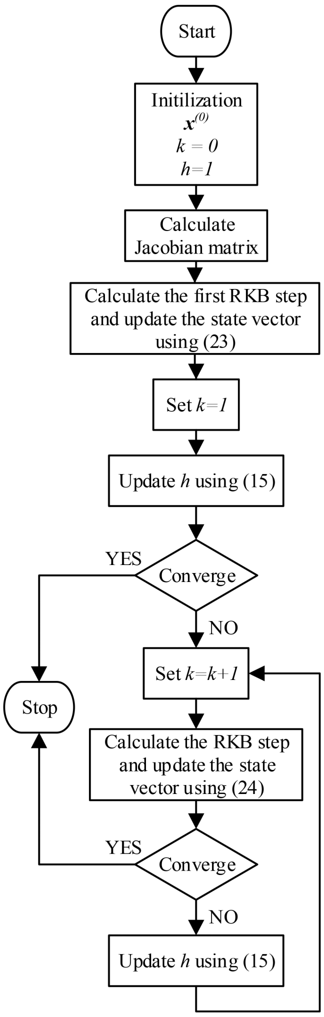

Fig. 2 *LF solution process based on the proposed methodology*

Table 1 Total required number of Jacobian matrix inverse for different methods

Method	Total number of $J_x(x)^{-1}$ computed
NR	K
RK4	$4K$
RK4B	1
Scheme 1	$1 + K^*$
Scheme 2	K

K = total number of iterations.
K^* = total number of NR iterations.

to NR when the size of error vector $\max\{abs(\Delta x^{(k)})\}$ become smaller than 10^{-1}. Generally, if the size of the corrector vector is small enough, it can be concluded that the solution of the LF is being achieved. Hence, we can safely accelerate the algorithm using faster technique (like NR).

This modification is based on the fact that the LF algorithm is able to converge easily after reaching to a specific convergence. This modification will be called scheme 1 onwards.

The second modification consists of improving the value of $x^{(k)}$ in each iteration, employing a NR iteration before the RK4B step (24). It is expected that the robustness of the RK4 will be preserved while the NR intermediate step accelerates the convergence. This approach will be called scheme 2 onwards. While the first iteration of the proposed scheme 2 is executed in the same manner as RK4B (23), following iterations are computed as follows:

$$\begin{aligned}
\tilde{x}^{(k)} &= x^{(k)} + f\left(x^{(k)}\right)\\
k_1^* &= f^*\left(\tilde{x}^{(k)}\right)\\
x_2^{(k)} &= \tilde{x}^{(k)} + \frac{h}{2}k_1^*\\
k_2^* &= f^*\left(x_2^{(k)}\right)\\
x_3^{(k)} &= \tilde{x}^{(k)} + \frac{h}{2}k_2^*\\
k_3^* &= f^*\left(x_3^{(k)}\right)\\
x_4^{(k)} &= \tilde{x}^{(k)} + hk_3^*\\
k_4^* &= f^*\left(x_4^{(k)}\right)\\
x^{(k+1)} &= \tilde{x}^{(k)} + \frac{h}{6}\left(k_1^* + 2k_2^* + 2k_3^* + k_4^*\right)
\end{aligned} \tag{25}$$

With the aim of comparing the proposed modifications, the total required number of Jacobian matrix inversions for different methods are given in Table 1.

From this table, it can be observed that the proposed methods reduce notably the required number of matrix inversions computed during the iterative procedure compared with RK4.

The flowchart of Fig. 2 can be easily modified in order to incorporate both Scheme 1 and Scheme 2. In the case of Scheme 1, the whole flowchart is replaced by the standard NR procedure when the convergence is ensured. On the other hand, Scheme 2 can be achieved based on the same flowchart by replacing the expression of (24) by (25).

4 Tests and results

In this section, the proposed methods are tested using several well- and ill-conditioned cases, in order to check their robustness and efficiency. All simulations have been carried out using Personal Computer Intel Core i5-7500 3.4 GHz and MATPOWER 6.0® software package [28]. The convergence tolerance is $\epsilon = 10^{-7}$ while the initial step size is taken $h = 1$ and it is updated in each iteration according to (15). In all cases, the flat initial guess point is considered.

The proposed methods are compared with the following well-known LF methods:

i. NR method [3].
ii. RK4 method [17].
iii. IM [16].
iv. LM [24].
v. HOLM [25].

4.1 Well-conditioned test systems

In this section, standard IEEE 30-bus, IEEE 57-bus, IEEE 118-bus and IEEE 300-bus systems are used to validate the proposed methods. Details of these systems can be found in [29, 30].

The computation time of the proposed methods, NR, IM, LM and HOLM are given in Table 2. Each case study has been run 100 times and average time has been calculated. Results from Table 2 have been obtained without considering generator and other equipment limits.

All studied IEEE systems are naturally well-conditioned, hence, most LF methods converged without problem. From this table, it can be observed that the proposed RK4B has low computation time compared with the other methods. In addition, the two proposed modifications have the lowest computation times compared with RK4B itself and the other methods. In Table 3, the required number of iterations for all LF methods is given.

From Table 3, it can be observed that the proposed RK4B and RK4 employ the same number of iterations for the studied well-conditioned systems. However, the two proposed modifications have improved notably the number of iterations of the RK4B to be smaller than the other robust methods. It is worth to mention that

Table 2 Computation time (second) for different LF methods (well-conditioned systems)

Method	IEEE 30-bus	IEEE 57-bus	IEEE 118-bus	IEEE 300-bus
NR	0.004	0.006	0.007	0.051
Iwamoto	0.062	0.072	0.127	0.16
LM	0.018	0.022	0.032	0.43
HOLM	0.014	0.016	0.028	0.57
RK4	0.054	0.067	0.099	0.23
RK4B	0.019	0.027	0.047	0.13
Scheme 1	0.008	0.010	0.016	0.088
Scheme 2	0.012	0.015	0.021	0.092

Table 3 Required number of iterations for different LF methods (well-conditioned systems)

Method	IEEE 30-bus	IEEE 57-bus	IEEE 118-bus	IEEE 300-bus
NR	3	4	4	5
Iwamoto	62	60	83	128
LM	9	11	10	26
HOLM	6	6	8	23
RK4	19	20	23	23
RK4B	19	20	23	25
Scheme 1	5	5	6	7
Scheme 2	4	4	4	4

Table 4 Computation time (second) and number of iterations in case of considering the generator's reactive power limits

Method	IEEE 30-bus	IEEE 57-bus	IEEE 118-bus	IEEE 300-bus
NR	0.009 (7)	0.006 (4)	0.013 (8)	0.022 (11)
Iwamoto	0.110 (104)	0.072 (60)	0.30 (171)	1.04 (347)
LM	0.024 (18)	0.018 (11)	0.055 (20)	0.44 (35)
HOLM	0.018 (11)	0.016 (6)	0.045 (16)	0.79 (33)
RK4	0.115 (39)	0.067 (20)	0.198 (46)	0.524 (62)
RK4B	0.031 (39)	0.027 (20)	0.088 (46)	0.49 (64)
Scheme 1	0.013 (10)	0.010 (5)	0.031 (12)	0.12 (16)
Scheme 2	0.017 (8)	0.015 (4)	0.042 (8)	0.21 (12)

Table 5 Computation time (second) and number of iterations for different LF methods (ill-conditioned systems)

Method	11-bus	13-bus	20-bus	43-bus
NR	fail	fail	fail	fail
Iwamoto	0.012 (10)	0.033 (36)	0.078 (85)	0.011 (5)
LM	0.300 (426)	0.017 (6)	0.022 (7)	0.033 (10)
HOLM	fail	0.009 (6)	0.010 (5)	0.011 (6)
RK4	0.042 (19)	0.044 (19)	0.052 (21)	0.064 (21)
RK4B	0.022 (24)	0.015 (19)	0.021 (23)	0.024 (22)
Scheme 1	0.009 (10)	0.006 (5)	0.007 (5)	0.00 9 (6)
Scheme 2	0.009 (6)	0.008 (4)	0.009 (4)	0.009 (4)

Table 6 Computation time in seconds and (number of iterations) for different LF methods in case of IEEE 30-bus with large *R*/*X* ratios

Method	$R_{\text{New}} = 1 \times R_{\text{Old}}$	$R_{\text{New}} = 2 \times R_{\text{Old}}$	$R_{\text{New}} = 3 \times R_{\text{Old}}$
	$X_{\text{New}} = 0.5 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$
NR	0.004 (3)	0.005 (4)	0.005 (4)
Iwamoto	0.205 (122)	0.133 (82)	0.159 (89)
LM	0.015 (7)	0.023 (11)	0.037 (15)
HOLM	0.013 (5)	0.018 (7)	0.020 (8)
RK4	0.060 (20)	0.054 (19)	0.054 (19)
RK4B	0.021 (20)	0.019 (19)	0.019 (19)
Scheme 1	0.006 (4)	0.008 (5)	0.010 (6)
Scheme 2	0.012 (4)	0.012 (4)	0.012 (4)

Table 7 Computation time in seconds and (number of iterations) for different LF methods in case of IEEE 57-bus with large *R*/*X* ratios

Method	$R_{\text{New}} = 1 \times R_{\text{Old}}$	$R_{\text{New}} = 2 \times R_{\text{Old}}$	$R_{\text{New}} = 3 \times R_{\text{Old}}$
	$X_{\text{New}} = 0.5 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$
NR	0.006 (4)	0.006 (4)	0.006 (4)
Iwamoto	0.225 (122)	0.148 (81)	0.188 (89)
LM	0.018 (8)	0.035 (14)	0.056 (18)
HOLM	0.016 (6)	0.023 (8)	0.030 (10)
RK4	0.075 (21)	0.067 (20)	0.067 (20)
RK4B	0.030 (21)	0.027 (20)	0.030 (21)
Scheme 1	0.007 (4)	0.010 (5)	0.013 (6)
Scheme 2	0.015 (4)	0.015 (4)	0.015 (4)

the proposed Scheme 2 has employed very few iterations to converge (almost the same number that standard NR).

The performance of the proposed methods is also validated when generator's reactive power limits are considered. In the course of LF solution, the calculated reactive power of each generator is checked if it is within the limits or not. If some limit is reached, then it is set to equal the limit violated and the bus voltage magnitude is released, where the connected bus is converted to PQ bus [17, 28]. In this case, new row and column have to be added to the Jacobian matrix for each converted bus. Table 4 presents the required computation time and total number of iterations (in parenthesis) for the IEEE studied systems considering the generator's reactive power limits.

From Table 4, it can be observed that the proposed approach has improved the computation time of the RK4. Moreover, the proposed two modifications reduce the number of iterations with respect to RK4 and remainder robust LF methods.

4.2 Ill-conditioned test systems

In order to check the robustness of the proposed methods, the naturally ill-conditioned 11-bus, 13-bus and 43-bus systems [14] are used. Furthermore, the 20-bus system [31] is considered. The computation time and total number of iterations (in parenthesis) for different LF methods after solving these systems are given in Table 5.

From Table 5, it can be observed that the proposed methods preserve the robustness features of the RK4 and improve its efficiency. It should be noticed that the proposed RK4B occasionally employs more iterations than the RK4. However, this drawback has been overcome by the proposed two modifications, which are capable of notably improving the overall performance of the RK4 in all studied cases.

The proposed methods have been tested for large *R*/*X* ratios cases. Results for different *R*/*X* ratios in the IEEE 30-bus, IEEE 57-bus, IEEE 118-bus and IEEE 300-bus cases have been reported in Tables 6–9, respectively.

From Tables 6–9, it can be concluded that the proposed methods exhibit good performance in case of high *R*/*X* ratios. In all studied cases, the proposed methods have successfully converged. Moreover, they have ensured a high degree of efficiency. For instance, the proposed RK4B and RK4 employed the same number of iterations. On the other hand, the proposed Scheme 1 and Scheme 2 have employed less iterations than the LM and HOLM methods.

4.3 Summary of results

The main contribution of this paper is to reduce the computational burden of the RK4, by reducing the number of Jacobian matrix

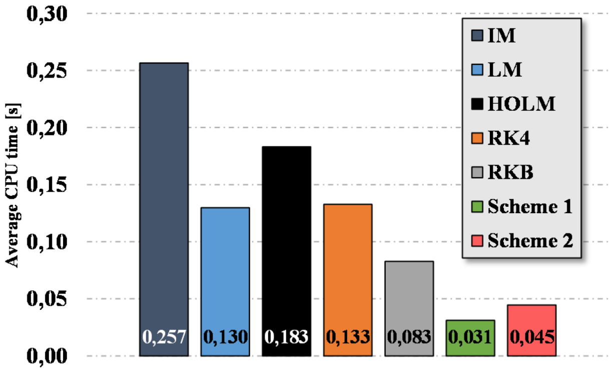

Fig. 3 *Average computation time of different robust LF methods for the studied cases*

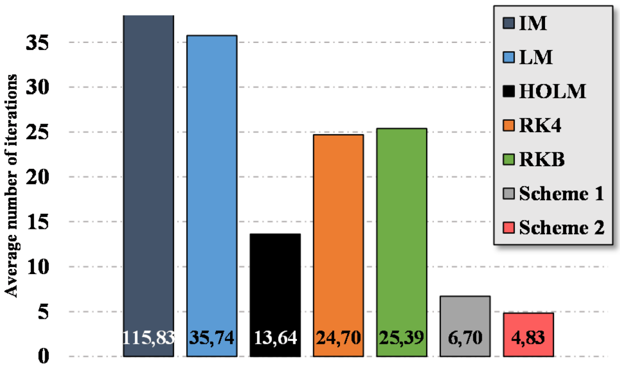

Fig. 4 *Average number of iterations of different robust LF methods for the studied cases*

inversions computed during LF solution. Therefore, the proposed methods can ensure higher efficiency than RK4 by consuming less time for solving the LF problem. Figs. 3 and 4 show a comparison of the average computational time and number of iterations employed for different robust LF solvers for the studied cases (omitting those cases which the method has failed). These values have been obtained by calculating the arithmetic mean of all values reported in Tables 2–9.

Despite that the proposed RK4B employs few more iterations (especially in ill-conditioned cases) than RK4, it consumes much less computation time. However, the LM and HOLM normally ensure a higher efficiency than RK4B; this issue is overcome by the proposed Scheme 1 and Scheme 2 which have shown the smallest computation time and number of iterations.

It is worth to notice that the LM has employed many iterations for solving the ill-conditioned 11-bus case while the HOLM has failed. However, the proposed methods have converged correctly in all studied cases employing a reasonable number of iterations.

5 Conclusions

This paper has proposed a methodology based on BM and fourth-order Runge–Kutta for LF solution of well- and ill-conditioned systems. Using this methodology, the required number of Jacobian matrix inversions are reduced to only one at the first iteration of LF solution. Consequently, the computation time is reduced compared with the original RK4 method.

Moreover, two simple but effective modifications of the proposed RK4B have been proposed. Both are based on the use of the NR in a safe manner, thus, the convergence can be accelerated while the robustness is preserved. These two modifications have been called Scheme 1 and Scheme 2.

Several tests have been carried out for well- and ill-conditioned systems and some relevant conclusions can be extracted as:

i. The proposed methodology constitutes a robust LF method. It is able to converge in ill-conditioned systems, while the other methods diverge in some cases.

Table 8 Computation time in seconds and (number of iterations) for different LF methods in case of IEEE 118-bus with large R/X ratios

Method	$R_{\text{New}} = 1 \times R_{\text{Old}}$	$R_{\text{New}} = 2 \times R_{\text{Old}}$	$R_{\text{New}} = 3 \times R_{\text{Old}}$
	$X_{\text{New}} = 0.5 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$
NR	0.007 (4)	0.007 (4)	0.009 (5)
Iwamoto	0.515 (195)	0.375 (120)	0.401 (139)
LM	0.074 (22)	0.074 (22)	0.105 (25)
HOLM	0.045 (11)	0.045 (11)	0.066 (12)
RK4	0.109 (24)	0.099 (23)	0.099 (23)
RK4B	0.051 (24)	0.047 (23)	0.047 (23)
Scheme 1	0.016 (6)	0.016 (6)	0.016 (6)
Scheme 2	0.021 (4)	0.021 (4)	0.021 (4)

Table 9 Computation time in seconds and (number of iterations) for different LF methods in case of IEEE 300-bus with large R/X ratios

Method	$R_{\text{New}} = 1 \times R_{\text{Old}}$	$R_{\text{New}} = 2 \times R_{\text{Old}}$	$R_{\text{New}} = 3 \times R_{\text{Old}}$
	$X_{\text{New}} = 0.5 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$	$X_{\text{New}} = 1 \times X_{\text{Old}}$
NR	0.016 (5)	0.018 (6)	diverge
Iwamoto	0.89 (296)	0.54 (178)	diverge
LM	0.28 (15)	0.89 (83)	diverge
HOLM	0.32 (12)	1.89 (84)	diverge
RK4	0.42 (24)	0.387 (23)	diverge
RK4B	0.322 (24)	0.35 (27)	diverge
Scheme 1	0.125 (6)	0.162 (8)	diverge
Scheme 2	0.172 (4)	0.25 (5)	diverge

ii. The proposed methodology is computationally efficient and stable. The computation time employed in both well- and ill-conditioned systems is smaller than RK4.

iii. The two proposed modifications present a high degree of robustness and computational performance compared with the other robust LF methods. The required number of Jacobian matrix inversions has been reduced to ≤25% of this required in case of RK4.

iv. Both proposed modifications of the RK4B have different features; Scheme 1 is faster than Scheme 2, while Scheme 1 normally employs more iterations.

v. Despite that the LM and HOLM have shown better computational performance than the proposed RK4B, they have shown convergence difficulties in the ill-conditioned 11-bus case. Moreover, the proposed modifications have improved the performance of the RK4B in both number of iterations and computational time in all studied cases.

The methodology proposed in this paper can be easily applied to other LF solvers based on Runge–Kutta formulas. The future work will focus on exploring the performance of the BM in other numerical methods such as Ralston and Runge–Kutta–Fehlberg's methods.

6 References

[1] Kundur, P.: '*Power system stability and control*' (McGraw-Hill, New York, 1994)

[2] Grainger, J.J., Stevenson, W.D.: '*Power system analysis*' (McGraw-Hill, New York, 1994)

[3] Tinney, W.F., Hart, C.E.: 'Power flow solution by Newton's method', *Trans. Power Appar. Syst.*, 1967, **PAS-86**, pp. 1449–1460

[4] Stott, B., Alsac, O.: 'Fast decoupled load flow', *IEEE Trans. Power Appar. Syst.*, 1974, **PAS-93**, pp. 859–869

[5] Powell, L.: '*DC load flow*' (Mc Graw Hill, New York, 2004)

[6] 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

[7] Yang, J., Zhang, N., Kang, C., *et al.*: 'A state-independent linear power flow model with accurate estimation of voltage magnitude', *IEEE Trans. Power Syst.*, 2017, **32**, (5), pp. 3607–3617

[8] Stott, B.: 'Review of load-flow calculation methods', *Proc. IEEE*, 1974, **62**, (7), pp. 916–929

[9] Yan, P., Sekar, A.: 'Steady-state analysis of power system having multiple FACTS devices using line-flow-based equations', *IEEE Proc. Gener. Transm. Distrib.*, 2005, **152**, (1), pp. 31–39

[10] Mallick, S., Rajan, D.V., Thakur, S.S.*, et al.*: 'Development of a new algorithm for power flow analysis', *Int. J. Electr. Power Energy Syst.*, 2011, **33**, (8), pp. 1479–1488

[11] Kamel, S., Abdel-Akher, M., Jurado, F.: 'Improved NR current injection load flow using power mismatch representation of PV bus', *Int. J. Electr. Power Energy Syst.*, 2013, **53**, pp. 64–68

[12] 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

[13] Saleh, S.A.: 'The formulation of a power flow using d-q reference frame components - part I: balanced 3 ϕ systems', *IEEE Trans. Ind. Appl.*, 2016, **52**, (5), pp. 3682–3693

[14] Tripathy, S.C., Durga Prasad, G., Malik, O.P.*, et al.*: 'Load flow solutions for ill-conditioned power systems by a Newton-like method', *IEEE Trans. Power Appar. Syst.*, 1982, **PAS-101**, (10), pp. 3648–3657

[15] Shahriari, A., Mokhlis, H., Bakar, A.: 'Critical reviews of load flow methods for well, ill and unsolvable condition', *J. Elect. Eng.*, 2012, **63**, (3), pp. 144–152

[16] Iwamoto, S., Tamura, Y.: 'A load fow calculation method for ill-conditioned power systems', *IEEE Trans. Power Appar. Syst.*, 1981, **PAS-100**, pp. 1736–1743

[17] Milano, F.: 'Continuous Newton's method for power flow analysis', *IEEE Trans. Power Syst.*, 2009, **24**, (1), pp. 50–57

[18] Schaffer, M.D., Tylavsky, D.J.: 'A nondiverging polar-form Newton-based power flow', *IEEE Trans. Ind. Appl.*, 1988, **24**, pp. 870–877

[19] Tylavsky, D.J., Crouch, P., Jarriel, L.F.*, et al.*: 'Improved power flow robustness for personal computers', *IEEE Trans. Ind. Appl.*, 1992, **28**, pp. 1102–1108

[20] Braz, L.M.C., Castro, C.A., Murari, C.A.F.: 'A critical evaluation of step size optimization based load flow methods', *IEEE Trans. Power Syst.*, 2000, **15**, (1), pp. 202–207

[21] Bijwe, P.R., Kelapure, S.M.: 'Nondivergent fast power flow methods', *IEEE Trans. Power Syst.*, 2003, **18**, (2), pp. 633–638

[22] Tate, J.E., Overbye, T.J.: 'A comparison of the optimal multiplier in polar and rectangular coordinates', *IEEE Trans. Power Syst.*, 2005, **20**, (4), pp. 1667–1674

[23] Hetzler, S.M.: 'A continuous version of Newton's method', *Coll. Math. J.*, 1997, **28**, (5), pp. 348–351

[24] Milano, F.: 'Analogy and convergence of levenberg's and lyapunov-based methods for power flow analysis', *IEEE Trans. Power Syst.*, 2016, **31**, (2), pp. 1663–1664

[25] 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

[26] Saadat, H.: '*Power system analysis*' (McGraw-Hill, New York, 2002)

[27] Broyden, C.G.: 'A class of methods for solving nonlinear simultaneous equations', *Math. Comput.*, 1965, **19**, (92), pp. 577–593

[28] Zimmerman, R.D., Murillo-Sánchez, 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

[29] Power systems test case archive. University of Washington, Seattle. Available at http://www.ee.washington.edu/research/pstca/

[30] Birchfield, A.B., Xu, T., Gegner, K.M.*, et al.*: 'Grid structural characteristics as validation criteria for synthetic networks', *IEEE Trans. Power Syst.*, 2017, **32**, (4), pp. 3258–3265

[31] Behnam-Guilani, K.: 'Fast decoupled load flow: The hybrid model', *IEEE Trans. Power Syst.*, 1987, **3**, (2), pp. 734–742

7 Appendix: Stability of the proposed methodology

By differentiating (16) with respect to $\boldsymbol{x}$ yields

$$\nabla_x^{\mathrm{T}}\boldsymbol{f}^*(\boldsymbol{x}) = -\boldsymbol{A}\boldsymbol{J}_x(\boldsymbol{x}) - \nabla_x^{\mathrm{T}}\boldsymbol{A}\boldsymbol{g}(\boldsymbol{x}) \tag{26}$$

If we assume that $\boldsymbol{x}^*$ is a solution of (4), we have that

$$\nabla_x^{\mathrm{T}}\boldsymbol{f}^*(\boldsymbol{x}^*) = -\boldsymbol{A}\boldsymbol{J}_x(\boldsymbol{x}^*) \tag{27}$$

By definition, the matrix $\boldsymbol{A}$ should be a reasonable approximation of $\boldsymbol{J}_x(\boldsymbol{x})^{-1}$, if we assume that $\boldsymbol{A} = \boldsymbol{J}_x(\boldsymbol{x})^{-1}$ we have

$$\nabla_x^{\mathrm{T}}\boldsymbol{f}^*(\boldsymbol{x}^*) = -\boldsymbol{I} \tag{28}$$

where $\boldsymbol{I}$ is the identity matrix. Equation (28) indicates that all eigenvalues of $\nabla_x^{\mathrm{T}}\boldsymbol{f}^*(\boldsymbol{x}^*)$ are equal to -1, it indicates that the proposed methodology is asymptotically stable if the initial guess is inside of region of attraction [17]. If we consider that $\boldsymbol{A}$ is not exactly equal to $\boldsymbol{J}_x(\boldsymbol{x})^{-1}$, then (28) is not true. Nevertheless, for normal conditions we can assume that $\boldsymbol{A}$ is very close to $\boldsymbol{J}_x(\boldsymbol{x})^{-1}$. In this case, $\nabla_x^{\mathrm{T}}\boldsymbol{f}^*(\boldsymbol{x}^*)$ would be equal to a diagonal dominant matrix with all real negative eigenvalues. It indicates that (16) would be stable even in bad conditions.