

Article

On Various High-Order Newton-Like Power Flow Methods for Well and Ill-Conditioned Cases

Talal Alharbi ¹, Marcos Tostado-Véliz ², Omar Alrumayh ³ and Francisco Jurado ^{2,*}¹ Department of Electrical Engineering, College of Engineering, Qassim University, Buraydah 52571, Saudi Arabia; atalal@qu.edu.sa² Department of Electrical Engineering, University of Jaén, 23700 Linares, Spain; mtostado@ujaen.es³ Department of Electrical Engineering, College of Engineering, Qassim University, Unaizah 56452, Saudi Arabia; o.alrumayh@qu.edu.sa

* Correspondence: fjurado@ujaen.es

Abstract: Recently, the high-order Newton-like methods have gained popularity for solving power flow problems due to their simplicity, versatility and, in some cases, efficiency. In this context, recent research studied the applicability of the 4th order Jarrat's method as applied to power flow calculation (PFC). Despite the 4th order of convergence of this technique, it is not competitive with the conventional solvers due to its very high computational cost. This paper addresses this issue by proposing two efficient modifications of the 4th order Jarrat's method, which present the fourth and sixth order of convergence. In addition, continuous versions of the new proposals and the 4th order Jarrat's method extend their applicability to ill-conditioned cases. Extensive results in multiple realistic power networks serve to show the performance of the developed solvers. Results obtained in both well and ill-conditioned cases are promising.



Citation: Alharbi, T.; Tostado-Véliz, M.; Alrumayh, O.; Jurado, F. On Various High-Order Newton-Like Power Flow Methods for Well and Ill-Conditioned Cases. *Mathematics* **2021**, *9*, 2019. <https://doi.org/10.3390/math9172019>

Academic Editor: Juan Eloy Ruiz-Castro

Received: 5 July 2021

Accepted: 13 August 2021

Published: 24 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: power flow analysis; high-order Newton-like methods; 4th order Jarrat's method; Continuous Newton's method

1. Introduction

Power flow calculation (PFC) supposes the most critical tool in power system analysis. It provides the initial conditions for contingency analysis or fault calculations to be an essential tool in market and planning studies [1]. In essence, PFC is stated as a non-linear problem that is customarily solved using iterative techniques.

With the growth of network size and complexity, traditional solvers are facing increasing issues on stability and computational efficiency [2]. For the sake of example, the so-called ill-conditioned systems are becoming more common [3]. In such cases, traditional solvers such as the Newton–Raphson (NR) may experience instability problems or yield inaccurate solutions [1]. These issues have motivated a recent trend on developing novel techniques for PFC, essentially devoted to either ill-conditioned cases [4–7], large-scale realistic systems [8,9], or jointly tackling both issues [2,10].

Regarding ill-conditioned cases, a huge variety of robust methods is available. The optimal multiplier-based methods were maybe the earliest robust solvers [11]. This kind of technique is quite robust and, theoretically, not divergent, which has motivated their usage to provide approximate solutions of unsolvable cases [12]. However, they may be quite inefficient or trapped on local minimums, such as those reported in [4,7].

For much of the twentieth century, ill-conditioned cases were quite unusual and their analyses were almost limited to research works and academic studies. However, ill-conditioned systems are more frequently observed nowadays, which has recently motivated people to find novel schemes suitable for more complex and realistic networks. The solvers based on the continuous Newton's method [4,6,7] are attractive for realistic systems; nevertheless, they are still relatively inefficient as various factorizations are required at each iteration. In addition, these solvers fail at turning points where the Jacobian is singular [4].

For circumventing the weaknesses of the continuous Newton-like solvers at turning points, continuation approaches [13] or Levenberg-type methods [14] have been applied. The latter has gained popularity over the continuation approaches due to its simplicity and efficiency. However, this class of techniques still presents essential issues. As reported in [14], the Levenberg-type methods may converge to nonphysical solutions. This inaccuracy is mainly due to the so-called damping factor, for which does not exist a unified tuning criterion as manifested in [2,5,10]. In addition, they are still computationally not competitive with conventional solvers.

As observed, most robust techniques are not computationally attractive. Besides the still prevalence of well-conditioned networks, this issue induces the usage of NR in industry software. However, the high-order Newton-like methods (HONL) have recently arisen as direct competitors of the conventional solvers [8,9]. The main attractiveness of HONL is their higher order of convergence, which leads them to a fast convergence, especially in stiff problems (heavy load or high R/X ratio). Additionally, these techniques are versatile enough to incorporate alternative power flow formulations such as those developed in [15]. The main limitation of HONL is their application in ill-conditioned cases, where they usually fail since they share similar robustness characteristics with the conventional NR.

In view of the above discussions, HONL seems an attractive alternative to conventional solvers; however, very few works have been conducted on studying the applicability of these methods in PFC. Motivated by the good characteristics of HONL for PFC, we have further studied the applicability of the 4th Jarrat's method (4OJ) (the applicability of this technique for PFC was previously studied in [8]). The authors honestly believe that the main limitation of 4OJ is its very high computational cost due to the required solution of a matrix system. This issue indeed supposes a barrier for the broad application of this method in industry tools. This paper deals with this drawback by developing two novel efficient solvers with 4th and 6th order of convergence based on 4OJ. The main attraction of the introduced techniques is that they avoid the solution of the matrix system, which is replaced by much cheaper solutions of linear systems. The continuous versions of the new proposals are also presented for extending their applicability to ill-conditioned cases. The developed methods are deduced from the original structure of 4OJ; then, using the Taylor expansion technique (e.g., see [16]), the developed mappings are properly developed from achieving high convergence rates with a low computational cost. A variety of realistic cases are studied to show the performance of our proposals.

The remainder of the paper is organized as follows. Section 2 outlines the necessary background. Section 3 develops two computationally attractive modifications of 4OJ with 4th and 6th order of convergence. Section 4 presents the continuous versions of the developed methods. A simple switching strategy for incorporating the reactive limits at PV buses is presented in Section 5. Various numerical experiments with results are provided in Section 6. Section 7 discusses possible real-life applications of the developed nonlinear solvers. This paper is concluded with Section 8.

2. Background

The PF equations are stated in their most general form as a set of n algebraic non-linear equations as follows:

$$\mathbf{g}(\mathbf{x}) = \mathbf{0} \quad (1)$$

where $\mathbf{g} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are the power flow equations and $\mathbf{x} \in \mathbb{R}^n$ is the vector of unknowns (state vector). The set of Equation (1) may be formulated in polar or rectangular coordinates, current injections-based mismatches [17], d-q reference frame [15], or even in complex form [18]. We have not been limited to a specified formulation of (1), since the introduced solvers are versatile enough for incorporating most of the available forms of the power flow equations.

Since (1) is non-linear, its solution cannot be directly obtained. The most conventional NR is traditionally used for solving (1). This technique is defined for the following iterative map:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (2)$$

where $\mathbf{g}' = \nabla_{\mathbf{x}} \mathbf{g} \in \mathbb{R}^{n \times n}$ is the Jacobian matrix of (1) w.r.t. \mathbf{x} , $[\ast]^{-1} : \mathbb{R}^{n \times n} \mapsto \mathbb{R}^{n \times n}$ is the inverse operator, and the subindexes denote the iteration counter. Alternatively, one may consider 4OJ for PFC as follows [19]:

$$\mathbf{y}_k = \mathbf{x}_k - \frac{2}{3} [\mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (3)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\mathbf{I} - \frac{3}{2} [3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)]^{-1} (\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)) \right] [\mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (4)$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix. The mapping (2) has a quadratic convergence rate while 4OJ has a fourth-order convergence [19]. Thereby, the mapping (3) and (4) should converge, consuming fewer iterations than NR. However, the mapping (3) and (4) involves the solution of a matrix system, for which up to n linear systems have to be solved, each one with a computational cost of $\mathcal{O}(n^3)$. Therefore, this calculation supposes a computational cost equal to $\mathcal{O}\left(\sum_n n^3\right) = \mathcal{O}(n^4)$, while the heaviest part of NR is the factorization of the Jacobian matrix with a cost of $\mathcal{O}(n^3)$. Both mappings (2) and (3) and (4) usually fail in ill-conditioned systems. A formal definition of an ill-conditioned case was provided by Milano in [4]:

Definition 1. *Ill-conditioned case: A power flow case can be categorized as ill-conditioned if, despite its solution does exist, it is not reachable using NR and a flat start (i.e., all initial voltage magnitudes equal to 1 p.u. and all voltage angles equal to 0 rad).*

3. Efficient Modifications of 4OJ

This section is devoted to developing efficient modifications of 4OJ, which are suitable for PFC in industry tools. The new proposals focus on avoiding the solution of the matrix in (4), which supposes the heaviest part of this mapping. This way, the new solvers are developed from the original 4OJ structure (3) and (4) and high convergence properties are deduced by using the Taylor expansion technique [16].

3.1. Elimination of the Matrix System in (3)

For reducing the computational cost of 4OJ, the matrix system involved in the mapping (3) and (4) should be eliminated. In this sense, it could be replaced by the solution of an equivalent linear system, which supposes a much lighter calculation [4]. Applying this modification to the mapping (3) and (4), one obtains:

$$\mathbf{y}_k = \mathbf{x}_k - \frac{2}{3} [\mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (5)$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (6)$$

The mapping (5) and (6) requires two LU decompositions, the solution of two linear systems, two Jacobian evaluations and a function evaluation. Therefore, it is notably cheaper than (3) and (4) as the main computation in (5) and (6) is the LU decomposition with a computational cost of $\mathcal{O}(n^3)$. In addition, it can be easily checked that the two solutions of the power flow problem (1) (say \mathbf{x}^* such that $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$) are fixed points for (5) and (6) since $\mathbf{x}_{k+1}|_{\mathbf{x}^*} = \mathbf{x}^*$. The convergence rate of the mapping (5) and (6) remains to be studied, which is addressed with the Theorem 1.

Theorem 1. Convergence of the mapping (5) and (6): Let $\mathbf{g} : D \subseteq \mathbb{R}^n \mapsto \mathbb{R}^n$ be sufficiently differentiable at each point of an open neighborhood D of $\mathbf{x}^* \in \mathbb{R}^n$, this is a solution of the system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Let us suppose that $\mathbf{g}(\mathbf{x})$ is continuous and nonsingular in \mathbf{x} . Then, the sequence $\{\mathbf{x}_k\}_{k \geq 0}$ obtained using the mapping (5) and (6) converges linearly to \mathbf{x}^* .

Proof of Theorem 1. To demonstrate the Theorem 1, we use the Taylor expansion technique (e.g., see [16]). This way, the Taylor expansion of $\mathbf{g}(\mathbf{x}_k)$ and $\mathbf{g}'(\mathbf{x}_k)$ is given by:

$$\mathbf{g}(\mathbf{x}_k) = \mathbf{g}'(\mathbf{x}^*) \left[\mathbf{e} + C_2 \mathbf{e}^2 + C_3 \mathbf{e}^3 + C_4 \mathbf{e}^4 + C_5 \mathbf{e}^5 + C_6 \mathbf{e}^6 + \mathcal{O}(\mathbf{e}^7) \right] \quad (7)$$

$$\mathbf{g}'(\mathbf{x}_k) = \mathbf{g}'(\mathbf{x}^*) \left[I + 2C_2 \mathbf{e} + 3C_3 \mathbf{e}^2 + 4C_4 \mathbf{e}^3 + 5C_5 \mathbf{e}^4 + 6C_6 \mathbf{e}^5 + \mathcal{O}(\mathbf{e}^6) \right] \quad (8)$$

where $\mathbf{e} = \mathbf{x}_k - \mathbf{x}^* \in \mathbb{R}^n$, $\mathbf{e}^i = \underbrace{(\mathbf{e}, \mathbf{e}, \dots, \mathbf{e})}_{i\text{-times}}$, $C_j = (1/j!) \left[\mathbf{g}'(\mathbf{x}^*) \right]^{-1} \mathbf{g}^{(j)}(\mathbf{x}^*) \in L_i(\mathbb{R}^n, \mathbb{R}^n)$, $\mathbf{g}^{(j)} \in L(\mathbb{R}^n \times \dots \times \mathbb{R}^n, \mathbb{R}^n)$ and $\left[\mathbf{g}'(\mathbf{x}^*) \right]^{-1} \in L(\mathbb{R}^n)$. Now, let us assume that:

$$\left[\mathbf{g}'(\mathbf{x}^{(k)}) \right]^{-1} = \left[c_1 I + c_2 \mathbf{e}^{(k)} + c_3 \mathbf{e}^{(k)^2} + c_4 \mathbf{e}^{(k)^3} + c_5 \mathbf{e}^{(k)^4} \right] \left[\mathbf{g}'(\mathbf{r}) \right]^{-1} + \mathcal{O}(\mathbf{e}^{(k)^6}) \quad (9)$$

the different c 's $\in \mathbb{R}$ can be calculated by using the following inverse definition:

$$\left[\mathbf{g}'(\mathbf{x}^{(k)}) \right]^{-1} \mathbf{g}'(\mathbf{x}^{(k)}) = \mathbf{g}'(\mathbf{x}^{(k)}) \left[\mathbf{g}'(\mathbf{x}^{(k)}) \right]^{-1} = I \quad (10)$$

and solving the resulting linear system one obtains:

$$\left[\mathbf{g}'(\mathbf{x}_k) \right]^{-1} = \left[I + X_2 \mathbf{e} + X_3 \mathbf{e}^2 + X_4 \mathbf{e}^3 + X_5 \mathbf{e}^4 + X_6 \mathbf{e}^5 \right] \left[\mathbf{g}'(\mathbf{x}^*) \right]^{-1} + \mathcal{O}(\mathbf{e}^6) \quad (11)$$

where:

$$X_2 = -2C_2 \quad (12)$$

$$X_3 = 4C_2^2 - 3C_3 \quad (13)$$

$$X_4 = -8C_2^3 + 12C_2 C_3 - 4C_4 \quad (14)$$

$$X_5 = 16C_2^4 - 36C_2^2 C_3 + 16C_2 C_4 + 9C_3^2 - 5C_5 \quad (15)$$

$$X_6 = -32C_2^5 + 96C_2^3 C_3 - 48C_2^4 C_4 - 54C_2 C_3^2 + 20C_2 C_5 + 24C_3 C_4 - 6C_6 \quad (16)$$

making use of (5), one can write:

$$\mathbf{d} = \mathbf{e} - \frac{2}{3} \left[\mathbf{g}'(\mathbf{x}_k) \right]^{-1} \mathbf{g}(\mathbf{x}_k) = \frac{4}{3} \left(-C_2^2 + C_3 \right) \mathbf{e}^3 + \frac{2}{3} C_2 \mathbf{e}^2 + \frac{1}{3} \mathbf{e} + \mathcal{O}(\mathbf{e}^4) \quad (17)$$

where $\mathbf{d} = \mathbf{y}_k - \mathbf{x}^* \in \mathbb{R}^n$. The Taylor expansion of $\mathbf{g}(\mathbf{y}_k)$ and $\mathbf{g}'(\mathbf{y}_k)$ around \mathbf{x}^* is given by:

$$\mathbf{g}(\mathbf{y}_k) = \mathbf{g}'(\mathbf{x}^*) \left[\mathbf{d} + C_2 \mathbf{d}^2 + C_3 \mathbf{d}^3 + \mathcal{O}(\mathbf{d}^4) \right] \quad (18)$$

$$\mathbf{g}'(\mathbf{y}_k) = \mathbf{g}'(\mathbf{x}^*) \left[I + 2C_2 \mathbf{d} + 3C_3 \mathbf{d}^2 + \mathcal{O}(\mathbf{d}^3) \right] \quad (19)$$

Now, $\left(3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k) \right)$ can be calculated as follows:

$$3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k) = \mathbf{g}'(\mathbf{x}^*) \left[2I + W_2 \mathbf{e}^2 + W_3 \mathbf{e}^3 + W_4 \mathbf{e}^4 + W_5 \mathbf{e}^5 + W_6 \mathbf{e}^6 + \mathcal{O}(\mathbf{e}^7) \right] \quad (20)$$

where:

$$W_2 = 4C_2^2 - 2C_3 \quad (21)$$

$$W_3 = 4(-2C_2^3 + 6C_2C_3 - C_4) \quad (22)$$

$$W_4 = -4(C_2^2C_3 + 2C_3^2) - 5C_5 \quad (23)$$

$$W_5 = 4(C_3^2C_2 - C_2^3C_3) - 6C_6 \quad (24)$$

$$W_6 = -16(C_2^4C_3 + C_3^3) \quad (25)$$

Inversion of (17) yields:

$$\left[3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)\right]^{-1} = \left[\frac{1}{2}\mathbf{I} + \mathbf{Z}_2\mathbf{e}^2 + \mathbf{Z}_3\mathbf{e}^3 + \mathbf{Z}_4\mathbf{e}^4 + \mathbf{Z}_5\mathbf{e}^5\right] \left[\mathbf{g}'(\mathbf{x}^*)\right]^{-1} + \mathcal{O}(\mathbf{e}^6) \quad (26)$$

where:

$$\mathbf{Z}_2 = \frac{1}{2}\mathbf{C}_3 - \mathbf{C}_2^2 \quad (27)$$

$$\mathbf{Z}_3 = 2\mathbf{C}_2^3 - 3\mathbf{C}_2\mathbf{C}_3 + \frac{8}{9}\mathbf{C}_4 \quad (28)$$

$$\mathbf{Z}_4 = 2\mathbf{C}_2^4 - \mathbf{C}_2^2\mathbf{C}_3 - \frac{2}{3}\mathbf{C}_2\mathbf{C}_4 - \frac{3}{2}\mathbf{C}_3^2 + \frac{65}{54}\mathbf{C}_5 \quad (29)$$

$$\mathbf{Z}_5 = -8\mathbf{C}_2^5 + 20\mathbf{C}_2^3\mathbf{C}_3 - \frac{32}{9}\mathbf{C}_2^2\mathbf{C}_4 - 10\mathbf{C}_2\mathbf{C}_3^2 - \frac{10}{27}\mathbf{C}_2\mathbf{C}_5 + \frac{4}{9}\mathbf{C}_3\mathbf{C}_4 + \frac{40}{27}\mathbf{C}_6 \quad (30)$$

Finally, the convergence order of the mapping (5) and (6) is obtained by developing (6), which yields:

$$\mathbf{x}_{k+1} - \mathbf{x}^* = \mathbf{e} - \left[3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)\right]^{-1} \mathbf{g}(\mathbf{x}_k) = \frac{1}{2}\mathbf{e} + \mathcal{O}(\mathbf{e}^2) \quad (31)$$

As observed in (31), the term \mathbf{e} has not vanished and, consequently, the convergence of (5) and (6) is linear. \square

3.2. A Modification of (4) with 4th Order of Convergence

Clearly, the mapping (5) and (6) is computationally cheaper than 4OJ since the solution of a matrix system is replaced by the solution of a linear one. However, as seen in the previous section, the method (5) and (6) presents linear convergence, which supposes an important drawback with respect to 4OJ. With the aim of preserving the convergence features of 4OJ, the step (6) can be modified using the information of both \mathbf{x} and \mathbf{y} points, which leads to the following iterative map:

$$\mathbf{y}_k = \mathbf{x}_k - \frac{2}{3} \left[\mathbf{g}'(\mathbf{x}_k)\right]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (32)$$

$$\mathbf{x}_{k+1} = \alpha\mathbf{x}_k + \beta\mathbf{y}_k - \left[3\mathbf{g}'(\mathbf{y}_k) - \mathbf{g}'(\mathbf{x}_k)\right]^{-1} \mathbf{g}(\mathbf{x}_k) \quad (33)$$

where $\alpha, \beta \in \mathbb{R}$. These parameters are introduced for achieving fourth-order of convergence. Their values are deduced in Theorem 2.

Theorem 2. Convergence of the mapping (32) and (33): Let $\mathbf{g} : D \subseteq \mathbb{R}^n \mapsto \mathbb{R}^n$ be sufficiently differentiable at each point of an open neighborhood D of $\mathbf{x}^* \in \mathbb{R}^n$, this is a solution of the system $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Let us suppose that $\mathbf{g}(\mathbf{x})$ is continuous and nonsingular in \mathbf{x} . Then, the sequence $\{\mathbf{x}_k\}_{k \geq 0}$ obtained using the mapping (32) and (33) converges to \mathbf{x}^* with order 4 if and only if $\alpha = \frac{1}{4}$ and $\beta = \frac{3}{4}$.

Proof of the Theorem 2. Deduction (7)–(31) is also valid in this case since (5) is equal to (32). In this case, one needs to develop (33) as follows:

$$x_{k+1} - x^* = \alpha e + \beta d - \left[3g'(y_k) - g'(x_k) \right]^{-1} g(x_k) = \Omega_4 e^4 + \Omega_3 e^3 + \Omega_2 e^2 + \Omega_1 e + \mathcal{O}(e^5) \quad (34)$$

where:

$$\Omega_1 = \left(\frac{\beta}{3} + \alpha - \frac{1}{2} \right) \quad (35)$$

$$\Omega_2 = \left(\frac{2\beta}{3} - \frac{1}{2} \right) C_2 \quad (36)$$

$$\Omega_3 = \left(\frac{4\beta}{3} (-C_2^2 + C_3) - C_3 + C_2^2 \right) \quad (37)$$

$$\Omega_4 = \left(\frac{5}{2} C_2 C_3 - C_2^3 - \frac{25}{18} C_4 \right) \quad (38)$$

for achieving 4th order of convergence, the terms Ω_1 , Ω_2 , and Ω_3 must vanish to (34). This is achieved for $\alpha = \frac{1}{4}$, $\beta = \frac{3}{4}$. \square

It is worth noting that x^* is still a fixed point for the mapping (32) and (33) since $y_k|_{x^*} = x^*$ and $x_{k+1}|_{x^*} = \frac{1}{4}x^* + \frac{3}{4}x^* = x^*$.

3.3. A Modification of (29) with 6th Order of Convergence

The mapping (29) can be further developed for increasing its convergence order by adding a third step with frozen Jacobian [20], as follows:

$$y_k = x_k - \frac{2}{3} \left[g'(x_k) \right]^{-1} g(x_k) \quad (39)$$

$$z_k = \frac{1}{4}x_k + \frac{3}{4}y_k - \left[3g'(y_k) - g'(x_k) \right]^{-1} g(x_k) \quad (40)$$

$$x_{k+1} = z_k - \gamma \left[3g'(y_k) - g'(x_k) \right]^{-1} g(z_k) \quad (41)$$

where $\gamma \in \mathbb{R}$ is introduced for increasing the convergence rate. As seen, the step (41) does not substantially contribute to increase the computational cost of (32) and (33), since no additional factorizations are necessary. Instead, the step (41) contributes with cheap computations like a function evaluation and the solution of a linear system. It can also be easily checked that x^* is still a fixed point for (39)–(41). The mapping (39)–(41) is able to achieve up to 6th order of convergence as demonstrated in Theorem 3.

Theorem 3. Convergence of the mapping (39)–(41): Let $g: D \subseteq \mathbb{R}^n \mapsto \mathbb{R}^n$ be sufficiently differentiable at each point of an open neighborhood D of $x^* \in \mathbb{R}^n$, this is a solution of the system $g(x) = 0$. Let us suppose that $g(x)$ is continuous and nonsingular in x . Then, the sequence $\{x_k\}_{k \geq 0}$ obtained using the mapping (39)–(41) converges to x^* with order 6 if and only if $\gamma = 2$.

Proof. For the mapping (39)–(41), equation (34) is still valid, but it needs to be rewritten:

$$r = \frac{1}{4}e + \frac{3}{4}d - \left[3g'(y_k) - g'(x_k) \right]^{-1} g(x_k) = \left(\frac{5}{2} C_2 C_3 - C_2^3 - \frac{25}{18} C_4 \right) e^4 + \mathcal{O}(e^5) \quad (42)$$

where $r = z_k - x^* \in \mathbb{R}^n$. Taylor expansion of $g(z_k)$ around x^* is given by:

$$g(z_k) = g'(x^*) \left[r + C_2 r^2 + \mathcal{O}(r^3) \right] = g'(x^*) \left[V_4 e^4 + V_5 e^5 + V_6 e^6 + \mathcal{O}(e^7) \right] \quad (43)$$

where:

$$V_4 = \frac{5}{2} C_2 C_3 - C_2^3 - \frac{25}{18} C_4 \quad (44)$$

$$V_5 = 5C_2^2C_3 - \frac{2}{9}C_2C_4 - 4C_2^4 + C_3^2 - \frac{46}{27}C_5 \quad (45)$$

$$V_6 = -21C_2^3C_3 - \frac{5}{6}C_2C_5 - \frac{11}{6}C_3C_4 + \frac{47}{9}C_2^2C_4 + \frac{29}{2}C_2C_3^2 + 6C_2^5 - \frac{107}{54}C_6 \quad (46)$$

By developing (41) one obtains:

$$x_{k+1} - x^* = r - \gamma \left[3\mathbf{g}'(y_k) - \mathbf{g}'(x_k) \right]^{-1} \mathbf{g}(z_k) = \Lambda_6 e^6 + \Lambda_5 e^5 + \Lambda_4 e^4 + \mathcal{O}(e^7) \quad (47)$$

where:

$$\Lambda_4 = \gamma \left(\frac{1}{2}C_2^3 + \frac{25}{36}C_4 - \frac{5}{4}C_2C_3 \right) + \frac{5}{2}C_2C_3 - C_2^3 - \frac{25}{18}C_4 \quad (48)$$

$$\Lambda_5 = \gamma \left(2C_2^4 - \frac{1}{2}C_3^2 + \frac{23}{27}C_5 - \frac{5}{2}C_2^3C_3 + \frac{1}{9}C_2C_4 \right) + 5C_2^3C_3 - \frac{2}{9}C_2C_4 - 4C_2^4 + C_3^2 - \frac{46}{27}C_5 \quad (49)$$

$$\Lambda_6 = \gamma \left(-4C_2^5 + \frac{107}{108}C_6 + \frac{27}{2}C_2^3C_3 + \frac{5}{12}C_2C_5 + \frac{29}{18}C_3C_4 - 4C_2^2C_4 - \frac{17}{2}C_2C_3^2 \right) - 21C_2^3C_3 - \frac{5}{6}C_2C_5 - \frac{11}{6}C_3C_4 + \frac{47}{9}C_2^2C_4 + \frac{29}{2}C_2C_3^2 + 6C_2^5 - \frac{107}{54}C_6 \quad (50)$$

One may check that the terms Λ_4 and Λ_5 vanish in equation (47) for $\gamma = 2$, which completes the proof. \square

3.4. Comparison of the Developed Methods with Other Conventional Solvers

Comparing NR (mapping (2)) with the developed solvers (32) and (33) and (39)–(41), one can clearly check that the NR is more efficient since only one LU factorization is required whereas the developed techniques require two. In addition, it is clear that the developed methods involve more computations like vectors sums. However, the developed solvers achieve higher convergence rate, which presumably leads to a faster convergence. In this sense, it is difficult to analyze which approach is more efficient. To solve this issue, it is desirable to compare the developed techniques with NR and 4OJ using some measure index. Various indexes are available in the literature for comparing the efficiency of different non-linear solvers. In this case, we have considered the following one [21]:

$$FEI = p^{\frac{1}{CO}} \quad (51)$$

where, $p \in \mathbb{R}^+$ is the order of convergence, and CO stands for the total computational cost of an iteration in terms of flops (see [21] for further details). The Index (51) measures the degree of efficiency of a non-linear solver. If two methods are compared, the one with the highest FEI would be the most efficient (i.e., a priori). Figure 1 plots the value of (51) for NR, 4OJ, and the developed mappings (taking the values of α , β , and γ deduced from Theorems 2 and 3) for different system sizes (n). As seen from this figure, 4OJ presents the lowest efficiency index while the developed mapping (39)–(41) shows the highest one. It is also worth noting that the two developed mappings showed a better efficiency index than NR.

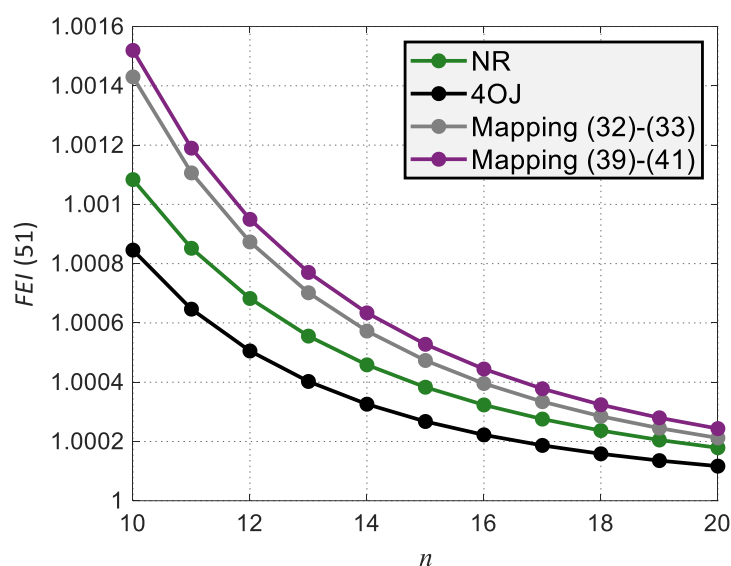


Figure 1. Value of the efficiency index (51) for NR, 4OJ, and the developed methods (32) and (33) and (39)–(41).

4. Continuous Versions of the Developed Solvers

The continuous Newton's method [4] is based on the analogy of NR and the forward Euler method. Following this same idea, one may observe the similitude of the mappings (3) and (4), (32) and (33), and (39)–(41) with the family of Runge–Kutta formulas [22]. Thus, whereas (3) and (4) and (32) and (33) share structure with the 2-stage Runge–Kutta methods (e.g., the Trapezoidal rule), the mapping (39)–(41) keeps similar to the 3-stage Runge–Kutta formulas. This analogy allows for directly applying the continuous Newton's framework to those techniques, thus extending their applicability to ill-conditioned problems. This way, by introducing the discrete step size $h \in \mathbb{R}^+$, the mappings (3) and (4), (32) and (33), and (39)–(41) may be respectively converted to their following continuous counterparts. In this case, we have considered the well-known Ralston's method [22], which considers a modified version of the forward Euler formula taking 2/3 parts of the step size. Keeping this on mind, the expressions (52)–(58) represent the continuous versions of the mappings (3) and (4), (32) and (33), and (39)–(41), respectively:

$$y_k = x_k - \frac{2h}{3} [g'(x_k)]^{-1} g(x_k) \quad (52)$$

$$x_{k+1} = x_k - h \left[I - \frac{3}{2} [3g'(y_k) - g'(x_k)]^{-1} (g'(y_k) - g'(x_k)) \right] [g'(x_k)]^{-1} g(x_k) \quad (53)$$

$$y_k = x_k - \frac{2h}{3} [g'(x_k)]^{-1} g(x_k) \quad (54)$$

$$x_{k+1} = \frac{1}{4}x_k + \frac{3}{4}y_k - h [3g'(y_k) - g'(x_k)]^{-1} g(x_k) \quad (55)$$

$$y_k = x_k - \frac{2h}{3} [g'(x_k)]^{-1} g(x_k) \quad (56)$$

$$z_k = \frac{1}{4}x_k + \frac{3}{4}y_k - h [3g'(y_k) - g'(x_k)]^{-1} g(x_k) \quad (57)$$

$$x_{k+1} = z_k - 2h [3g'(y_k) - g'(x_k)]^{-1} g(z_k) \quad (58)$$

The step size can be adapted for each iteration based on the local truncation error. Exploiting the similitude of (52)–(58) with the family of Runge–Kutta techniques, the local truncation error can be estimated using the half-step method as follows:

$$\zeta = \|\mathbf{y}_k - \mathbf{x}_{k+1}\|_\infty \quad (59)$$

where $\|\cdot\|_\infty : \mathbb{R}^n \rightarrow \mathbb{R}^+$ is the infinity norm. Then, (60) can be used for computing the step size using the following heuristic rule [4]:

$$\begin{aligned} \text{if } \zeta > 100 \text{ then } h &\leftarrow \max\{0.75h, 0.5\} \\ \text{if } \zeta \leq 100 \text{ then } h &\leftarrow \min\{1.2h, 1\} \end{aligned} \quad (60)$$

Based on these rules, the step size increases if the truncation error is lower than a given threshold and decreases if the truncation error is greater than a given threshold. In this case, the step size is lower bounded to 0.5 for avoiding being trapped in a local minimum, while the step size is upper bounded to 1 since the considered mappings achieved their maximum convergence rate for this value. It is worth noting that if h is not lower limited, the mappings (52)–(58) would provide a solution close to the feasibility of the power flow equations, as discussed in [4,12]. The section is completed by summarizing the main steps of PFC using the mapping (52) and (53) (also applicable for (54) and (55) and (56)–(58)) in the Algorithm 1 using pseudocode. Here, $\|\mathbf{g}\|_\infty$ has been considered as convergence criterion and $\varepsilon \in \mathbb{R}^+$ is the convergence tolerance. In addition, the algorithm fails if the iteration counter surpasses a given threshold k^{\max} , which usually indicates divergence.

Algorithm 1 main steps of PFC using (46)

```

1:  Let  $\mathbf{x}_0, \varepsilon, k^{\max}$  and  $h$  be given
2:  Initialize  $k \leftarrow 0$ 
3:  while  $\|\mathbf{g}(\mathbf{x}_k)\|_\infty \geq \varepsilon$  do
4:    Solve (52) and (53) # or (54) and (55), (56)–(58)
5:    if  $\|\mathbf{g}(\mathbf{x}_k)\|_\infty < \varepsilon$  then
6:      break # Convergence
7:    else
8:      Update  $k \leftarrow k + 1$ 
9:      if  $k > k^{\max}$  then
10:        break # Fail
11:      end if
12:    end if
13:    Update  $h$  using (60)
14:  end do
15:  return solution  $\mathbf{x}_k$ 

```

Stability of the Mappings (52)–(58)

The mappings (52)–(58) can be interpreted as dynamic systems, and their stabilities may be studied by calculating the eigenvalues of the Jacobian of the function $\mathbf{G} = \mathbf{x}_{k+1} - \mathbf{x}_k$ at \mathbf{x}^* [4].

For the mapping (52) and (53), one can write:

$$\mathbf{G} = -h \left[\mathbf{I} - \frac{3}{2} \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right]^{-1} \left(\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right) \right] \left[\mathbf{g}'(\mathbf{x}) \right]^{-1} \mathbf{g}(\mathbf{x}) \quad (61)$$

by differentiating (61) with respect to \mathbf{x} , one obtains:

$$\begin{aligned} \nabla_{\mathbf{x}} \mathbf{G} = -h \left[-\frac{3}{2} \left(\nabla_{\mathbf{x}} \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right]^{-1} \left(\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right) + \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right]^{-1} \left(\mathbf{g}''(\mathbf{y}) - \mathbf{g}''(\mathbf{x}) \right) \right) \right] \\ \left[\mathbf{g}'(\mathbf{x}) \right]^{-1} \mathbf{g}(\mathbf{x}) - h \left[\mathbf{I} - \frac{3}{2} \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right]^{-1} \left(\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(\mathbf{x}) \right) \right] \left(\nabla_{\mathbf{x}} \left[\mathbf{g}'(\mathbf{x}) \right]^{-1} \mathbf{g}(\mathbf{x}) + \mathbf{I} \right) \end{aligned} \quad (62)$$

where $\mathbf{g}'' = \nabla_{xx}\mathbf{g} \in \mathbb{R}^{n \times n^2}$ is the Hessian matrix of the power flow equations (see [23]). Easily, one can check that $\mathbf{y}|_{x^*} = x^*$ for (52) and (53), so, evaluating (62) at x^* yields:

$$\nabla_x \mathbf{G}|_{x^*} = -h\mathbf{I} \quad (63)$$

Equation (63) says that all the eigenvalues of $\nabla_x \mathbf{G}|_{x^*}$ are negative real, which implies that x^* , if it exists, is asymptotically stable. For the mapping (54) and (55), the function \mathbf{G} cannot be directly obtained. However, since in this case $\mathbf{y}|_{x^*} = x^*$, we can write:

$$\mathbf{G}|_{x^*} = -h \left[2\mathbf{g}'(x^*) \right]^{-1} \mathbf{g}(x^*) \quad (64)$$

and by differentiating (64) w.r.t. x , one obtains:

$$\nabla_x \mathbf{G}|_{x^*} = -h \left(\nabla_x \left[2\mathbf{g}'(x^*) \right]^{-1} \mathbf{g}(x^*) + \left[2\mathbf{g}'(x^*) \right]^{-1} \mathbf{g}'(x^*) \right) = -\frac{h}{2}\mathbf{I} \quad (65)$$

by result (65), one can conclude that x^* is also asymptotically stable for (54) and (55). Finally, for the mapping (56)–(58):

$$\mathbf{G} = -h \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}(x) - 2h \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}(z) \quad (66)$$

by differentiating (66) w.r.t. x , one obtains:

$$\nabla_x \mathbf{G} = -h \left(\nabla_x \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}(x) + \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}'(x) + 2 \left(\nabla_x \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}(x) + \left[3\mathbf{g}'(\mathbf{y}) - \mathbf{g}'(x) \right]^{-1} \mathbf{g}'(z) \right) \right) \quad (67)$$

Since that $\mathbf{y}|_{x^*} = x^*$ for the mapping (56)–(58), one can assert that that $z|_{x^*} = x^*$; so, this result can be substituted into (67), which yields:

$$\nabla_x \mathbf{G}|_{x^*} = -\frac{3}{2}h\mathbf{I} \quad (68)$$

Since the eigenvalues of (68) have not a real positive part, one can conclude that x^* is asymptotically stable for the mapping (56)–(58).

5. Handling with Reactive Limits of Generators

One main concern in PFC is the consideration of different equipment limits [9]. In this sense, the formulations presented on this work do not directly take into account the reactive limits at PV buses. Nonetheless, consideration of this aspect is straightforward by incorporating a common switching PV-PQ strategy, similar to other works like [6,9]. Basically, it consists on converting those PV buses with any limit violation to PQ, taking the limit hit as injected reactive power. The proposed strategy is summed in the following steps.

- (1) Firstly, the power flow is solved by ignoring the reactive limits. After that, it is checked if any limit violation at PV buses occurs. If not, the process is ended and the solution x_k is declared feasible; otherwise go to step 2.
- (2) Those PV buses are converted to PQ. For those, the violated reactive limit is taken as the new injected reactive power, and the voltage magnitude is incorporated into the unknown vector as a variable.
- (3) Taking x_k as starting point, the power flow problem is solved, then go to step 1.

If there are not PV buses in Step 1, the problem is declared unfeasible since there does not exist a solution in which the reactive limits are not violated. It is worth noting that similar strategies can be considered for incorporating other functional limits or controls.

6. Numerical Experiments

Along the developed methods (32) and (33) and (39)–(41) (with $\alpha = \frac{1}{4}$, $\beta = \frac{3}{4}$ and $\gamma = 2$), NR, 4OJ, and the Reverse-Bulirsch–Stoer algorithm (RBS) [6] have been imple-

mented in polar coordinates on Matpower [24]. The Algorithm 1 has been followed as benchmark for implementing the tested solvers, omitting the line 13 when not needed and replacing (52) and (53) in line 4 by the mapping considered. Various realistic systems such as the 69-bus radial network, the 2000-bus Texas synthetic grid along various portions of the Polish and the European transmission systems have been considered [25–27]. The tested networks have been denoted by the name used in Matpower (see [28] for details). All simulations have been carried out from a flat start, taking $\varepsilon = 10^{-5}$ and $k^{\max} = 1000$.

6.1. Convergence Rates for Well-Conditioned Cases

Table 1 presents the total iterations for different realistic well-conditioned cases ignoring the reactive limits. As seen in this table, the developed mappings alongside 4OJ were able to converge faster than the other tested solvers. This was expected since NR has quadratic convergence while RBS usually converges linearly due to the influence of the step size, while 4OJ and (32) and (33) have 4th order of convergence and (39)–(41) has 6th order of convergence. Figure 2 shows the convergence rate of 4OJ and the mappings (32) and (33) and (39)–(41) in the ‘case9241pegase’. As observed, 4OJ and the mapping (32) and (33) converged linearly for the first iteration, while the mapping (39)–(41) converged almost quadratically. In the second iteration, 4OJ and the mapping (32) and (33) achieved 3rd order of convergence while the mapping (39)–(41) achieved 6th order of convergence and reached the required convergence tolerance. For the third iteration, when the algorithms have evolved very close to the solution, 4OJ and the mapping (32) and (33) achieved 9th order of convergence. These results demonstrate the local convergence characteristics of the studied techniques. Nevertheless, the developed mapping (39)–(41) has shown superior convergence features than 4OJ and (32) and (33).

Table 1. Total iterations for well-conditioned cases.

Case	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	3	8	2	2	1
case_ACTIVSg2000	5	12	3	3	2
case3120sp	5	13	3	3	2
case9241pegase	5	13	3	3	2

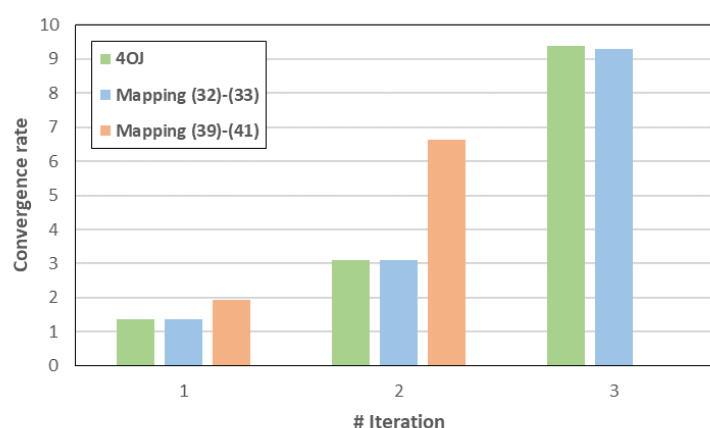


Figure 2. Convergence rates of 4OJ and the developed methods for the ‘case9241pegase’.

6.2. Convergence Rates for Ill-Conditioned Cases

Next, let us study how the continuous methods (52)–(58) perform in some ill-conditioned cases. Table 2 shows the total iterations of these methods along the continuous implementation of NR (‘Simple Robust Method’ (SRM) in [4]) and RBS, in two snapshots of the Polish transmission system (these systems are ill-conditioned as reported in [6] according to the Definition 1, so that NR, 4OJ, (32) and (33), and (39)–(41) fail for solving them). As in

preceding simulations, the reactive limits are ignored in this case. For NR and the mappings (52)–(58), the step size is initialized equal to 0.5, while the guidelines in [6] have been followed for RBS. In the studied cases, SRM performed poorly, employing a huge amount of iterations for converging, while the developed mappings were the most competitive. Figure 3 shows the convergence rate of the mappings (52)–(58) for the ‘case3012wp’ along the value of the step size each iteration. As observed, these techniques performed linearly when h was short. As the algorithms evolved, the step size was enlarged by making use of the rule (60) up to be equal to 1, then, the mappings (52)–(58) quickly achieved high convergence rates due to the algorithms are already evolving close to the solution.

Table 2. Total iterations of various continuous methods for ill-conditioned cases.

Case	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	27	13	5	5	5
Case3375wp	27	13	5	5	5

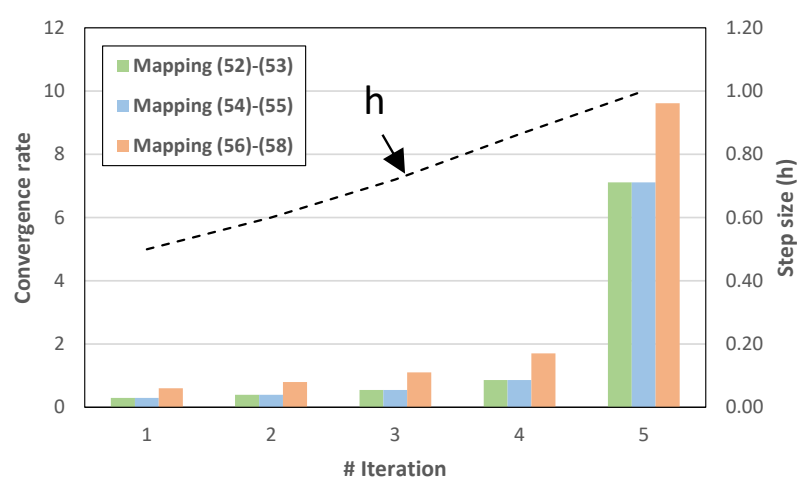


Figure 3. Step size and convergence rates of 4OJ and the developed methods for the ‘case3012wp’.

6.3. Convergence Rates for Limit Cases

The loading level of a system may affect the performance and reliability of a power flow solution method. In this regard, that loading level closest to the point of collapse supposes the most demanding conditions from a power flow solver. We have studied the performance of the considered techniques for such a point (limit loading level namely λ^{lim}). The limit loading level for each system has been calculated using the continuation power flow [13]. Tables 3 and 4 provide the total iterations for well and ill-conditioned cases with limit loading conditions, respectively, ignoring the reactive limits at PV buses. Regarding well-conditioned cases, the mapping (39)–(41) required fewer iterations than NR, RBS, 4OJ, and the mapping (32) and (33). As expected, 4OJ and the mapping (32) and (33) required the same number of iterations due to its 4th order of convergence. In ill-conditioned cases, the mapping (56)–(58) was the most competitive.

Table 3. Total iterations for well-conditioned cases with limit load conditions.

Case	λ^{lim}	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	3.2116	8	9	7	7	4
case_ACTIVSg2000	1.0236	9	12	5	5	4
case3120sp	1.5653	11	13	6	6	5
case9241pegase	1.0767	10	13	5	5	4

Table 4. Total iterations for ill-conditioned cases with limit load conditions.

Case	λ^{lim}	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	1.2734	27	13	7	7	6
Case3375wp	1.1586	27	13	7	7	6

6.4. Convergence Rates with Reactive Limits

Next, the total iterations with reactive limits at PV buses are presented in Tables 5 and 6 for well and ill-conditioned cases, respectively. These limits have been incorporated in the tested methods using the strategy described in Section 5. As observed, the mapping (39)–(41) required the least number of iterations in well-conditioned cases, while 4OJ and the mapping (32) and (33) outperformed NR and RBS in all cases. For ill-conditioned cases, the mapping (56)–(58) required fewer iterations than SRM, RBS, and the mappings (52)–(55). It is worth noting that these two latter techniques outperformed SRM and RBS in all cases.

Table 5. Total iterations for well-conditioned cases with reactive limits.

Case	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	3	8	2	2	1
case_ACTIVSg2000	13	39	8	8	5
case3120sp	13	48	8	8	6
case9241pegase	12	41	7	7	5

Table 6. Total iterations for ill-conditioned cases with reactive limits.

Case	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	53	29	15	15	13
Case3375wp	57	32	19	19	16

6.5. Solution Times

Tables 7–12 provide the solution times for the simulations reported in Tables 1–6, respectively. These results have been obtained on a 64-bit i5-9400F Intel Core personal computer (2.90 GHz, 8 GB of RAM), as the average value of 100 simulations.

Table 7. Solution times (ms) for well-conditioned cases.

Case	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	3.66	23.21	6.96	4.22	2.34
case_ACTIVSg2000	70.73	542.04	7764.41	84.46	60.16
case3120sp	96.82	800.57	18,625.39	113.15	79.51
case9241pegase	320.10	2677.31	629,536.74	379.32	265.16

Table 8. Solution times (ms) for ill-conditioned cases.

Case	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	473.65	767.98	21,374.45	178.03	183.12
Case3375wp	524.10	848.08	26,557.90	196.71	201.40

Table 9. Solution times (ms) for well-conditioned cases with limit load conditions.

Case	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	9.76	26.10	24.36	14.77	9.36
case_ACTIVSg2000	127.35	542.04	9705.50	140.75	120.32
case3120sp	212.96	800.54	27,938.10	226.32	198.80
case9241pegase	640.20	2677.35	629,536.75	632.20	530.32

Table 10. Solution times (ms) for ill-conditioned cases with limit load conditions.

Case	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	473.58	768.04	29,924.23	249.27	219.72
Case3375wp	524.07	848.12	37,181.06	275.38	241.68

Table 11. Solution times (ms) for well-conditioned cases with reactive limits.

Case	NR	RBS	4OJ	(32) and (33)	(39)–(41)
case69	3.66	23.21	6.96	4.22	2.34
case_ACTIVSg2000	183.95	1761.63	15,528.80	225.20	150.40
case3120sp	251.68	2955.84	37,250.80	301.76	238.56
case9241pegase	768.24	8443.95	881,351.45	885.08	662.90

Table 12. Solution times (ms) for ill-conditioned cases with reactive limits.

Case	SRM [4]	RBS	(52) and (53)	(54) and (55)	(56)–(58)
Case3012wp	929.62	1713.32	64,123.35	534.15	476.06
Case3375wp	1106.37	2087.68	100,920.02	747.46	644.48

For well-conditioned cases, the mapping (32) and (33) was occasionally slower than NR, which may be contradictory with the conclusions of Section 3.3. For explaining that, one should note the total amount of LU factorizations required by each method. Since it is the heaviest part of NR and the mapping (32) and (33), the required solution time will be quite proportional to the total amount of these calculations. For example, the mapping (32) and (33) needed six factorizations for the ‘case_ACTIVSg2000’, while NR needed five. It also explains why the mapping (39)–(41) was the most competitive technique while RBS needed six factorizations for the first iteration, four iterations for the second, and two for each remainder iteration; thus, it was required for the ‘case_ACTIVSg2000’ 30 factorizations in total. The reason behind the low efficiency of 4OJ was, as commented, the solution of a matrix system.

Similar conclusions can be extracted for RBS in ill-conditioned cases. Here, SRM turned out to be inefficient due to the large number of iterations needed. The mapping (52) and (53), as 4OJ, was very inefficient due to the calculation of a matrix system.

For limit load conditions and reactive limits enforcement, the mapping (39)–(41) was the most efficient in well-conditioned cases, while the mapping (56)–(58) was the winner in ill-conditioned cases.

7. Potential Applications of the Developed Solvers

The developed nonlinear solvers introduced in this paper are focused on PFC. To validate them, various realistic systems have been tested and the results were promising. However, the developed techniques can find other applications that may be valuable for real-life tools. Further analysis of such applications is out of scope of this paper, but this section is dedicated to overview them.

In the area of electrical engineering, the developed solvers could offer good results in other related tools. For example, the so-called continuation power flow (CPF) [13] is performed in two stages called predictor and corrector. During the corrector stage, the power flow problem is solved. This process is repeated various times until the whole continuation curve is drawn, which may imply many PFCs and therefore the whole process can be very time-consuming. Recent studies [29] have demonstrated that the usage of HONL techniques may alleviate the computational cost of CPF, which may suppose an unvaluable contribution for real-time security tools, in which CPF is widely employed.

Water (or liquid) flow models are usually nonlinear [30,31]. To analyze dynamics behavior of water flow, some models based on differential equations, such as the Richards

equations, can be linearized, and then solved using iterative methods. In [31], NR, Picard iteration, and L-scheme methods were compared for this problem showing that not only computational efficiency, but also numerical stability are important in water flow problems. In this area, the developed solvers and their continuous counterparts may offer very good performance. Indeed, while the mappings (32) and (33) and (39)–(41) have been proved to be more efficient than NR, the continuous mappings (52)–(58) can offer good numerical stability in ill-posed problems.

Finally, computational efficiency is of great interest to solve nonlinear coefficient inverse problems for partial differential equations. Especially when the coefficients are time-dependent and therefore various systems have to be solved, usually in real-time applications. Clear examples of these kinds of problems are the Moore–Gibson–Thompson equation [32] or the the nonlinear Oskolkov’s system [33].

8. Conclusions and Future Works

This paper deals with the applicability of 4OJ for PFC. This technique is not computationally competitive, at least in realistic systems, due to its high computational cost; more specifically, the solution of a matrix system involved in the solution procedure of 4OJ makes this technique unattractive for industry tools. Motivated by this issue, the authors of this paper have proposed two modifications of 4OJ, which aim at circumventing the computational issues manifested by 4OJ. The new methodologies focus on eliminating the matrix system, replacing it with cheaper calculations like solution linear systems. The new proposals have 4th and 6th order of convergence and present a better efficiency index.

Additionally, this paper has discussed the continuous counterparts of 4OJ and the developed methods. These formulations aim at extending the applicability of the developed solvers to ill-conditioned cases, which are more frequently observed nowadays. Thereby, the new proposals suppose a valuable contribution for industry tools.

Extensive numerical experiments have been performed. The developed techniques have been compared with NR, 4OJ, SRM, and RBS. Both well and ill-conditioned systems have been contemplated for analyzing the performance of the developed solvers and their continuous counterparts. Results revealed the superiority of the new proposals compared with other traditional PF solvers. The new methodologies outperformed NR and RBS in all studied cases in both convergence rates and solution times. An in-depth comparison with 4OJ revealed the clear superiority of our methods and serves to highlight the difficulties of 4OJ to be applied to PF analysis.

Future works should be devoted to validating the novel techniques in other related problems such as those reviewed in Section 7, thus, as other PF formulations like [15,17] or [18].

Author Contributions: Conceptualization, M.T.-V. and F.J.; methodology, M.T.-V. and T.A.; software, T.A. and O.A.; validation, T.A., O.A. and F.J.; formal analysis, M.T.-V., T.A. and O.A.; investigation, M.T.-V.; resources, T.A., O.A. and F.J.; data curation, T.A. and O.A.; writing—original draft preparation, M.T.-V.; writing—review and editing, T.A., O.A. and F.J.; visualization, M.T.-V. and T.A.; supervision, T.A., O.A. and F.J.; project administration, F.J.; funding acquisition, T.A. and O.A. All authors have read and agreed to the published version of the manuscript.

Funding: Deanship of Scientific Research, Qassim University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The researchers would like to thank the Deanship of Scientific Research, Qassim University for funding the publication of this project.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Karimi, M.; Shahriari, A.; Aghamohammadi, M.R.; Terzija, V. Application of Newton-based load flow methods for determining steady-state condition of well and ill-conditioned power systems: A review. *Int. J. Elect. Power Energy Syst.* **2019**, *113*, 298–309. [CrossRef]
- Tang, K.; Dong, S.; Shen, J.; Zhu, C.; Song, Y. A Robust and Efficient Two-Stage Algorithm for Power Flow Calculation of Large-Scale Systems. *IEEE Trans. Power Syst.* **2019**, *34*, 5012–5022. [CrossRef]
- Xie, N.; Torelli, F.; Bompard, E.; Vaccaro, A. Dynamic computing paradigm for comprehensive power flow analysis. *IET Gener. Transmiss. Distrib.* **2013**, *7*, 832–842. [CrossRef]
- Milano, F. Continuous Newton's Method for Power Flow Analysis. *IEEE Trans. Power Syst.* **2009**, *24*, 50–57. [CrossRef]
- 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. Transmiss. Distrib.* **2016**, *10*, 3017–3022. [CrossRef]
- Tostado-Véliz, M.; Kamel, S.; Jurado, F. A Robust Power Flow Algorithm Based on Bulirsch-Stoer Method. *IEEE Trans. Power Syst.* **2019**, *34*, 3081–3089. [CrossRef]
- Milano, F. Implicit Continuous Newton Method for Power Flow Analysis. *IEEE Trans. Power Syst.* **2019**, *34*, 3309–3311. [CrossRef]
- Derakhshandeh, S.Y.; Pourbagher, R. Application of high-order Newton-like methods to solve power flow equations. *IET Gener. Transmiss. Distrib.* **2016**, *10*, 1853–1859. [CrossRef]
- Tostado-Véliz, M.; Kamel, S.; Jurado, F. A Novel Family of Efficient Power-Flow Methods With High Convergence Rate Suitable for Large Realistic Power Systems. *IEEE Syst. J.* **2021**, *15*, 738–746. [CrossRef]
- Tostado-Véliz, M.; Kamel, S.; Alquthami, T.; Jurado, F. A Three-Stage Algorithm Based on a Semi-Implicit Approach for Solving the Power-Flow in Realistic Large-Scale ill-Conditioned Systems. *IEEE Access* **2020**, *8*, 35299–35307. [CrossRef]
- Iwamoto, S.; Tamura, Y. A Load Flow Calculation Method for Ill-Conditioned Power Systems. *IEEE Trans. Power Appar. Syst.* **1981**, *PAS-100*, 1736–1743. [CrossRef]
- Overbye, T.J. A power flow measure for unsolvable cases. *IEEE Trans. Power Syst.* **1994**, *9*, 1359–1365. [CrossRef]
- Zhang, X.-P.; Ju, P.; Handschin, E. Continuation three-phase power flow: A tool for voltage stability analysis of unbalanced three-phase power systems. *IEEE Trans. Power Syst.* **2005**, *20*, 1320–1329. [CrossRef]
- Milano, F. Analogy and Convergence of Levenberg's and Lyapunov-Based Methods for Power Flow Analysis. *IEEE Trans. Power Syst.* **2016**, *31*, 1663–1664. [CrossRef]
- Saleh, S.A. The Formulation of a Power Flow Using d-q Reference Frame Components-Part I: Balanced 3 Systems. *IEEE Trans. Ind. Appl.* **2016**, *52*, 3682–3693. [CrossRef]
- Cordero, A.; Villalba, E.G.; Torregrosa, J.R.; Triguero-Navarro, P. Convergence and Stability of a Parametric Class of Iterative Schemes for Solving Nonlinear Systems. *Mathematics* **2021**, *9*, 86. [CrossRef]
- Penido, D.R.R.; de Araujo, L.R.; Carneiro, S.; Pereira, J.L.R.; Garcia, P.A.N. Three-Phase Power Flow Based on Four-Conductor Current Injection Method for Unbalanced Distribution Networks. *IEEE Trans. Power Syst.* **2008**, *23*, 494–503. [CrossRef]
- Pires, R.; Mili, L.; Chagas, G. Robust complex-valued Levenberg-Marquardt algorithm as applied to power flow analysis. *Int. J. Elect. Power Energy Syst.* **2019**, *113*, 383–392. [CrossRef]
- Khirallah, M.Q.; Hafiz, M.A. Solving system of non-linear equations using family of Jarratt methods. *Int. J. Differ. Equ. App.* **2013**, *12*, 2–69. [CrossRef]
- Traub, J.F. *Iterative Methods for the Solution of Equations*; Chelsea: New York, NY, USA, 1982.
- Cordero, A.; Hueso, J.L.; Martínez, E.; Torregrosa, J.R. A modified Newton-Jarratt's composition. *Numer. Algor.* **2010**, *5*, 87–99. [CrossRef]
- Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; Wiley: Hoboken, NJ, USA, 2003.
- Iwamoto, S.; Tamura, Y. A Fast Load Flow Method Retaining Nonlinearity. *IEEE Trans. Power Appar. Syst.* **1978**, *PAS-97*, 1586–1599. [CrossRef]
- 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*, 12–19. [CrossRef]
- Birchfield, A.B.; Xu, T.; Gegner, K.M.; Shetye, K.S.; Overbye, T.J. Grid Structural Characteristics as Validation Criteria for Synthetic Networks. *IEEE Trans. Power Syst.* **2017**, *32*, 3258–3265. [CrossRef]
- Josz, C.; Fliscounakis, S.; Maeght, J.; Panciatici, P. AC Power Flow Data in Matpower and QCQP Format: iTesla, RTE Snapshots, and PEGASE. *arXiv* **2016**. Available online: <https://arxiv.org/abs/1603.01533> (accessed on 9 June 2021).
- Fliscounakis, S.; Panciatici, P.; Capitanescu, F.; Wehenkel, L. Contingency Ranking with Respect to Overloads in Very Large Power Systems Taking into Account Uncertainty, Preventive, and Corrective Actions. *IEEE Trans. Power Syst.* **2013**, *28*, 4909–4917. [CrossRef]
- Matpower User's Manual. Available online: <https://matpower.org/docs/manual.pdf> (accessed on 9 June 2020).
- Tostado-Véliz, M.; Kamel, S.; Jurado, F. Development and Comparison of Efficient Newton-Like Methods for Voltage Stability Assessment. *Elect. Power Comp. Syst.* **2020**, *48*, 1798–1813. [CrossRef]
- Difonzo, F.V.; Masciopinto, C.; Vurro, M.; Berardi, M. Shooting the Numerical Solution of Moisture Flow Equation with Root Water Uptake Models: A Python Tool. *Water Res. Manag.* **2021**, *35*, 2553–2567. [CrossRef]
- Illiano, D.; Pop, I.S.; Radu, F.A. Iterative schemes for surfactant transport in porous media. *Comput. Geosci.* **2021**, *25*, 805–822. [CrossRef]

-
32. Tekin, I. Inverse problem for a nonlinear third order in time partial differential equation. *Math. Methods Appl. Sci.* **2021**, *44*, 9571–9581. [[CrossRef](#)]
 33. Fedorov, V.E.; Ivanova, N.D. Inverse problem for Oskolkov's system of equations. *Math. Methods Appl. Sci.* **2017**, *40*, 6123–6126. [[CrossRef](#)]