An Improved Version of the Continuous Newton's Method for Efficiently Solving the Power-Flow in Ill-conditioned Systems

Marcos Tostado-Véliz[1], Manuel A. Matos[2,3], João A. Peças Lopes[2,3], Francisco Jurado[1,*]

[1]*Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén, Spain*

[2] *Institute for Systems and Computer Engineering, Technology and Science, Porto 4200-465 Portugal*

[3]*Faculty of Engineering, University of Porto, Porto 4200-465, Portugal*

Abstract – This paper tackles the efficient Power-Flow solution of ill-conditioned cases. In that sense, those methods based on the Continuous Newton's philosophy look very promising, however, these methodologies still present some issues mainly related with the computational efficiency or the robustness properties. In order to overcome these drawbacks, we suggest several modifications about the standard structure of the Continuous Newton's method. Thus, the standard Continuous Newton's paradigm is firstly modified with a frozen Jacobian scheme for reducing its computational burden; secondly, it is extended for being used with High-order Newton-like method for achieving higher convergence rate and, finally, a regularization scheme is introduced for improving its robustness features. On the basis of the suggested improvements, a Power-Flow solution paradigm is developed. As example, a novel Power-Flow solver based on the introduced solution framework and the 4th order Runge-Kutta formula is developed. The novel technique is validated in several realistic large-scale ill-conditioned systems. Results show that the suggested modifications allow to overcome the drawbacks presented by those methodologies based on the Continuous Newton's method. On the light of the results obtained it can be also claimed, that the developed solution paradigm constitutes a promising framework for developing robust and efficient Power-Flow solution techniques.

Keywords: Power-Flow, Ill-conditioned power systems, Continuous Newton's method, High order Newton-like methods, Lavretiev's Regularization

---

*Corresponding author, Tel.: +34 953 648518; Fax: +34 953 648586.
E-mail addresses: fjurado@ujaen.es (F. Jurado), mtostado@ujaen.es (M. Tostado-Véliz), mam@fe.up.pt (M.A. Matos), jpl@fe.up.pt (J.A.P. Lopes)

1.- Introduction

1.1.- Motivation

The Power-Flow (PF) is a very important computational tool which finds multiple applications in planning and operation of power systems [1]. In PF calculation, the nonlinear model of a power system is solved for obtaining the steady operational state of the system.

While most of power systems are easily solvable using the Newton-Raphson technique (NR) [2], other systems are difficult to solve using NR. The latter are typically called ill-conditioned systems and they are becoming more frequent [3]. Consequently, although solution of PF is a classic and well-studied topic, solution of this problem in ill-conditioned systems is currently attracting important research efforts.

1.2.- Literature Review

For well-conditioned cases, multiple solvers have been developed during the twentieth century. Among the most popular, one can cite the variety of decoupled techniques [4], linear models [5], quasi and dishonest Newton's approaches [6], Forward-Backward algorithms [7] or High Order Newton-like techniques [8], among others [9, 10]. This kind of techniques experience different issues in ill-conditioned cases [11]. Many papers relate the ill-conditioning of a power system with its high R/X ratio or heavy loading conditions [12]. In such cases, NR may still converge without any problem, however, its convergence features are normally deteriorated. Milano provided a formal definition of an ill-conditioned system in [13], which has been followed in this paper.

Definition 1. *Ill-conditioned case: a power flow problem is called ill-conditioned when, despite its solution exists, it is not reachable using NR and a flat start (e.g. all voltage magnitudes equal to* 1 *pu at PQ buses and all voltage angles equal to* 0 *deg).*

Milano directly relates the ill-conditioning with the concept of Region of Attraction (RoA), which is defined as the set of starting guesses which a power flow solution method converges. Thus, a power flow problem can be defined as ill-conditioned if the flat start lies outside of the Roa of NR. Ill-conditioned cases consequently entail important drawbacks for conventional solvers. As alternative, a wide variety of robust methods have been developed at late twentieth century and early twenty-first century for solving ill-conditioned problems.

At 80-90's, various authors exploited the idea of the optimal multiplier-based PF approaches [14-16]. This kind of techniques are based on calculating a multiplier, as result of an optimization problem raised from the second order expansion of the PF equations in rectangular coordinates [14]. The reason of using the rectangular form of the PF equations is the non-existence of transcendental functions, which implies that terms with order higher than two vanish from the Taylor expansion of the PF equations [16]. Nevertheless, rectangular formulations still present some disadvantages, which motivated the study of this kind of techniques in polar coordinates [15]. The optimal multiplier-based solution methods are quite robust and theoretically non-divergent, which has enabled their usage for unsolvable cases [17, 18]. However, these methods are still not competitive with conventional solvers in some cases; for example, they tend to converge slowly due to the optimal multiplier tend to decrease when the algorithm evolves close to the solution [6, 13].

NR and Newton-based techniques such as in [8], fail at bifurcation points where the Jacobian matrix is singular [19]. For such cases, continuation [20] and regularization

methods [12] have showed be efficacious. The former are based on parameterizing the PF equations with the loading level. This kind of techniques are able to calculate all PF solutions for different loading conditions and estimate the maximum loadability point of a network, however, these techniques aim at calculating multiple PF solutions, which, although is interesting for other related tools like voltage stability analysis, supposes a drawback in comparison with conventional techniques, which aims at calculating just a snapshot of the system for a specific load profile. So that, if a continuation technique, for example, needs to calculate 10 solution points, its computational cost will be 10 times higher than that required by, for example, NR (one can see also results in [21]). The Regularization techniques are devoted on improving the condition number of the Jacobian matrix by adding some elements on its diagonal [22]. They are undoubtedly more competitive than continuation approaches, however, their computational burden is still higher than NR as normally an extra matrix-matrix product and a denser LU factorization are required. In addition, these techniques normally present very low convergence rate when the starting guess lies far away to the solution.

The Continuous Newton's method (CN), establishes a formal analogy between the solution of the PF equations using NR and the solution of a set of ordinary differential equations (ODEs) using the Forward-Euler technique (FE). On the basis of this analogy, any numerical method can be considered for solving PF. This idea was firstly exploited by Milano in [13] and the authors in [23-27]. This kind of PF solvers are reasonably efficient and scalable, however, their robustness properties are frequently based on empirical results and they may be weak in some cases [25]. In addition, they may require heavy calculations, for example, in [28] a PF solution method based on the 4^{th} order Runge-Kutta formula was proposed, which requires four LU factorizations each iteration.

So that, it can be concluded that its computational cost is approximately four times higher than NR.

Another category of robust PF solvers lies in the category of the dogleg or trust-region methods [29, 30]. They are basically modifications of NR or even a more sophisticated scheme with respect the optimal multiplier-based methods. Thus, instead of an optimal multiplier resulting from an optimization problem, a step size is calculated using a plethora of coefficients and heuristic-based rules. This step truncates the Newton's increment vector with the object of keeping the evolution inside a secure region of convergence. As regularization methods, this kind of techniques are robust and efficient, however, they still fail at bifurcation points and evolve very slowly when the starting point is far away to the solution.

By using the Lyapunov theory, the PF equations can be raised as an artificial dynamical system. Thus, PF can be solved using numerical integration techniques. The resulting iterative solution procedure is barely affected by the starting point [31]. However, computational cost of this solution paradigm is very expensive, which limits its application in large-scale systems [3].

Recently, the homotopic and holomorphic theorems have gained popularity for developing robust PF solvers [21, 32-34]. However, applicability of the Holomorphic Embedded method (HE) for PF analysis is still considered an open topic. For example, HE represents an infinite number of formulations, each one with different numerical properties. In addition, this methodology has turned out to be less efficient than NR [21] due to some heavy computations as convolutions. An elegant alternative to homotopic or holomorphic-based methods was proposed by the authors in [35], where a novel family of robust and efficient PF solvers was proposed on the basis on a simple Newton-based solution paradigm.

1.3.- Contributions

As reader can check from the preceding subsection, most of robust PF solvers present some important drawbacks like heavy computational burden or weakness in some cases. Among the considered techniques or family of methods, those PF solvers based on CN seem the most promising in the sense of efficiency and simplicity. However, this kind of techniques still present some important drawbacks, which suppose an important barrier for a widely application of these methods:

- Although quite efficient, the CN-based methodologies might still be not competitive in comparison with other techniques like NR. This is due to this kind of methods requires as matrix factorizations as the order of the numerical integration technique used.
- The CN-based techniques, at least those studied in [13, 23-26], present very low convergence rate. This fact along the previous drawback have made these methods totally inefficient in comparison with the conventional solvers.
- The CN-based methodologies fail close at bifurcation points, where the Jacobian matrix is close singular [13] and the mapping may experience numerical instabilities.
- Although wider convergent than NR, those methodologies based on CN may still present a narrow RoA, so that, they may be not efficacious in ill-conditioned cases.

Due to these reasons, we honestly believe that the standard CN framework is not widely applicable for the resolution of large-scale realistic systems, such as industry tools have to solve. This paper deals with those limitations by introducing several modifications on the original structure of CN. Thus, the contributions of this paper may be summed in the following points:

- A novel formulation of CN which the Jacobian matrix is only factorized once each iteration is proposed. Thus, the introduced solution paradigm equating its computational burden with NR.
- In this paper, we propose to extend the formulation of CN to the high-order Newton-like techniques with the aim of speeding up its convergence properties.
- Finally, in order to improve the robustness properties of the CN-based methodologies, we propose to introduce regularization schemes [36] within its iterative solution procedure. This modification is expected to enhance the stability at turning points as the condition number of the Jacobian is improved.

As result of introducing the modifications above, a novel PF solution paradigm is developed. The main advantages brought by the introduced solution framework with respect other related works [13, 23-26], can be summed in the following points:

- The introduced solution paradigm is more efficient and converges faster than those techniques based on CN [13, 23-26].
- Unlike to other related papers [13, 23-26], the developed paradigm is stable even near to bifurcation points due to the usage of regularization schemes.
- The developed solution framework is more scalable as only one LU factorization is required each iteration.

In addition, the developed solution paradigm preserves the main advantages of the conventional CN framework. In addition, it is kept conceptually simple to be to be easily coded in most of commercial software or even for research and educational tools, and it could be easily extensible to other PF formulations. The developed paradigm, as CN, allows to consider any numerical method for developing robust and efficient solvers. For the sake of exemplify, as in [13, 28], we have developed a novel PF solver based on the

proposed solution framework and the 4th order Runge-Kutta formula. The novel technique is validated in several numerical results realistic large-scale systems.

1.4.- Paper Organization

Remainder of this paper is organized as follows. PF problem and its solution based on NR and CN are described in Section 2. Section 3 describes the suggested improvements about the traditional structure of CN. On the basis of the introduced modifications, a common PF solution paradigm and a novel PF solver are developed in Section 4. Section 5 validates the developed PF solver using various well-known test systems. Section 6 describes and comments several simulations with the aim of validating the developed PF solver. This paper is concluded with Section 7.

2.- Background

2.1.- PF Solution by NR

Let us consider the PF problem in polar coordinates as a set of n nonlinear algebraic equations given by:

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

where $\boldsymbol{g}: \mathbb{R}^n \longmapsto \mathbb{R}^n$ are the PF equations and $\boldsymbol{x} \in \mathbb{R}^n$ is the PF state vector (interested readers can find detailed information about the PF problem in [37]). Solution of (1) using NR leads to the following iterative map:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \tag{2}$$

where $\boldsymbol{g}' = \nabla_{\boldsymbol{x}}^{T} \boldsymbol{g} \in \mathbb{R}^{n \times n}$ is the PF Jacobian matrix formed by the first partial derivatives of (1) w.r.t. $\boldsymbol{x}$, and superscript denotes the k^{th} iteration of the solution procedure.

2.2.- CN and its Application to PF Analysis

Now, let us consider the following set of ODEs:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) \tag{3}$$

Solution of (3) using FE leads to the following map:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + h\boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) \tag{4}$$

where $h \in \mathbb{R}^+$ is a discrete step size. A formal analogy between (2) and (4) could be observed if one defines [13]:

$$\boldsymbol{f}(\boldsymbol{x}) = -[\boldsymbol{g}'(\boldsymbol{x})]^{-1}\boldsymbol{g}(\boldsymbol{x}) \tag{5}$$

On the basis of this analogy, any other numerical integration routine can be applied to solve the PF equations, by just applying the following generic formulation:

$$\begin{cases} \boldsymbol{k_1} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) \\ \boldsymbol{k_2} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)} + h(a_{21}\boldsymbol{k_1})\right) \\ \vdots \\ \boldsymbol{k_p} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)} + h\sum_{i=1}^{p-1} a_{p,i}\,\boldsymbol{k_i}\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p} b_i\,\boldsymbol{k_i} \end{cases} \tag{6}$$

where a's, b's$\in \mathbb{R}$ are the Runge-Kutta weight and coefficients, respectively [38]. Basically, a CN-based PF solver is defined by these coefficients and its order p. In [13, 28], a PF solver based on the 4th order Runge-Kutta method was proposed while other CN-based techniques were studied in [26].

Those methods defined by (6) are asymptotically stable if the starting guess $\boldsymbol{x}^{(0)}$ lies inside of the Region of Attraction [13]. Consequently, these techniques are still sensitive to the initialization of the iterative procedure. In addition, they fail at turning points where the Jacobian matrix is singular.

It is worth comparing the computational cost of the mappings (6) and (2). In the former, the Jacobian matrix has to be factorized p times. If we assume that the factorization of the Jacobian matrix is the heaviest part of a PF calculation [13], one can deduce that (2) would be p times more efficient than (6). In fact, (6) is even less efficient if one takes into account the total number of function and Jacobian evaluations. These computations, although may be efficiently addressed, should be also considered [39].

3.- Proposed Improvements for CN

3.1.- Reducing the Total Number of Matrix Factorizations

As commented, one of the main drawbacks of (6) and other CN-based methods [26], is its computational cost derived by the total number of matrix factorizations required each iteration. In order to alleviate this computational burden, let us bring the following theorem.

Theorem 1 [40]. *Let r be a simple zero of a function $g(x)$ and let ϕ define an iterative method of order p. Then, a composite method $\psi(x)$ introduced by Newton's method*

$$\psi(x) = \phi(x) - \frac{g(\phi(x))}{g'(x)} \tag{7}$$

defines an iterative method of order $p + 1$

The theorem 1 is the basis of the multipoint methods [41], which achieves high convergence order by evaluating the function in several points rather than the Jacobian, which is just evaluated (and factorized) at $\boldsymbol{x}^{(k)}$. One can find some similitudes between (6) and a multipoint iterative scheme. Keeping this in mind, we can define the following function:

$$\boldsymbol{\phi}(\boldsymbol{w},\boldsymbol{z}) = -[\boldsymbol{g}'(\boldsymbol{w})]^{-1}\boldsymbol{g}(\boldsymbol{z}) \tag{8}$$

where $\boldsymbol{w}, \boldsymbol{z} \in \mathbb{R}^n$ are arbitrary vectors. As (6) is the application of a Runge-Kutta method for integrating (5), one can apply any Runge-Kutta formula for integrating (8) as follows [13, 28]:

$$\begin{cases} \boldsymbol{k_1} = \boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right) \\ \boldsymbol{k_2} = \boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + h(a_{21}\boldsymbol{k_1})\right) \\ \vdots \\ \boldsymbol{k_p} = \boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p-1} a_{p,i}\,\boldsymbol{k_i}\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p} b_i\,\boldsymbol{k_i} \end{cases} \tag{9}$$

The main merit of (9) about (6) is that Jacobian matrix is only evaluated and factorized once at $\boldsymbol{x}^{(k)}$. This supposes an important computational saving since (9) is computationally comparable with NR. Nevertheless, it is worth mentioning that any method defined by (9) is still less efficient than NR due to (8) has to be evaluated p times.

Unlike multipoint methods, it is not expected that any method defined by (9) achieves high convergence order due to the influence of h, which keeps the convergence linear. This is explained in the fact that h plays a similar role to the optimal multiplier for the techniques in [14-16]. The convergence rate of these techniques may be studied by the Taylor expansion technique (see [39]). Taylor Expansions of $\boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right)$ and $\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)$ around $\boldsymbol{r}$ (where $\boldsymbol{r} \in \mathbb{R}^n$ is a PF solution such as $\boldsymbol{g}(\boldsymbol{r}) = \boldsymbol{0}$) are given by:

$$\boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) = \boldsymbol{g}'(\boldsymbol{r})\left[\boldsymbol{e}^{(k)} + \boldsymbol{C_2}\boldsymbol{e}^{(k)^2} + \boldsymbol{C_3}\boldsymbol{e}^{(k)^3} + \mathcal{O}\left(\boldsymbol{e}^{(k)^4}\right)\right] \tag{10}$$

$$\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right) = \boldsymbol{g}'(\boldsymbol{r})\left[\boldsymbol{I} + 2\boldsymbol{C_2}\boldsymbol{e}^{(k)} + 3\boldsymbol{C_3}\boldsymbol{e}^{(k)^2} + \mathcal{O}\left(\boldsymbol{e}^{(k)^3}\right)\right] \tag{11}$$

where $\boldsymbol{e}^{(k)} = \boldsymbol{x}^{(k)} - \boldsymbol{r} \in \mathbb{R}^n$, $\boldsymbol{e}^i = \underbrace{(\boldsymbol{e}, \boldsymbol{e}, \dots, \boldsymbol{e})}_{i-\text{times}}$, $\boldsymbol{C}_j = (1/j!)[\boldsymbol{g}'(\boldsymbol{r})]^{-1}\boldsymbol{g}^{(j)}(\boldsymbol{r}) \in L_i(\mathbb{R}^n, \mathbb{R}^n)$, $\boldsymbol{g}^{(j)} \in L(\mathbb{R}^n \times \dots \times \mathbb{R}^n, \mathbb{R}^n)$, $[\boldsymbol{g}'(\boldsymbol{r})]^{-1} \in L(\mathbb{R}^n)$ and $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. Inversion of (11) yields [39]:

$$\left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} = \left[\boldsymbol{I} - 2\boldsymbol{C}_2\boldsymbol{e}^{(k)} + (4\boldsymbol{C}_2^2 - 3\boldsymbol{C}_3)\boldsymbol{e}^{(k)^2} + \mathcal{O}\left(\boldsymbol{e}^{(k)^3}\right)\right][\boldsymbol{g}'(\boldsymbol{r})]^{-1} \tag{12}$$

Thus, one can calculate:

$$\underbrace{\boldsymbol{x}^{(k+1)} - \boldsymbol{r}}_{\boldsymbol{e}^{(k+1)}} = \underbrace{\boldsymbol{x}^{(k)} - \boldsymbol{r}}_{\boldsymbol{e}^{(k)}} - \mu\left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1}\boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) = (1-\mu)\boldsymbol{e}^{(k)} + \mu\boldsymbol{C}_2\boldsymbol{e}^{(k)^2} + \mathcal{O}\left(\boldsymbol{e}^{(k)^3}\right) \tag{13}$$

where $\mu \in \mathbb{R}^+$ may be understood as the optimal multiplier for the techniques in [14-16] or the step size for the methods in [13, 23-28]. For achieving quadratic convergence the term $\boldsymbol{e}^{(k)}$ should vanish in (13), which is achieved for $\mu = 1$. It means that the techniques [13-16, 23-28] achieve their highest convergence order when the effect of the optimal multiplier or the step size vanish. However, this effect allows these techniques be wider convergent, so that, one strategy for effectively solving ill-conditioned systems consists on intentionally allowing linear convergence. By (13) is also deduced that $\mu \leq 1$, so that the step size should be upper bounded.

3.2.- Increasing the Convergence Rate of CN

Although robust and efficient, CN-based methods typically require many iterations for converging in comparison with NR [13, 26] due to their linear convergence as commented in Section 3.1. In this case, it is worth noting that CN is the continuous counterpart of NR, which has quadratic convergence order. However, one should know that a huge variety of third [42], fourth [43] or even higher order Newton-like methods

[44] are available in the literature. Thus, we can follow the same ideas applied to NR in order to develop continuous counterparts of other high order Newton-like techniques.

For the sake of exemplify, let us consider the cubic method defined in [42]:

$$\begin{cases} \boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} - \left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} - \left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \boldsymbol{g}\left(\boldsymbol{y}^{(k)}\right) \end{cases} \tag{14}$$

The mapping (14) is still tractable as the Jacobian matrix is factorized once each iteration. One should note (14) can be easily written as follows:

$$\begin{cases} \boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + \boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \boldsymbol{\varphi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}\right) \end{cases} \tag{15}$$

where:

$$\boldsymbol{\varphi}(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{z}) = -[\boldsymbol{g}'(\boldsymbol{w})]^{-1}\left(\boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{z})\right) \tag{16}$$

where $\boldsymbol{u} \in \mathbb{R}^n$ is an arbitrary vector. Although trivial, map (16) allows to easily extend the continuous philosophy to method (14) by applying different numerical integration methods for $\boldsymbol{f}$ and $\boldsymbol{\varphi}$. This idea can be extended to the generic formulation (6) and the following map is obtained:

$$\begin{cases} \boldsymbol{k_1} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)}\right) \\ \boldsymbol{k_2} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)} + h(a_{21}\boldsymbol{k_1})\right) \\ \vdots \\ \boldsymbol{k_p} = \boldsymbol{f}\left(\boldsymbol{x}^{(k)} + h\sum_{i=1}^{p-1} a_{p,i}\, \boldsymbol{k_i}\right) \\ \boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p} b_i\, \boldsymbol{k_i} \\ \boldsymbol{v_1} = \boldsymbol{\varphi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}\right) \\ \boldsymbol{v_2} = \boldsymbol{\varphi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)} + h(c_{21}\boldsymbol{v_1})\right) \\ \vdots \\ \boldsymbol{v_q} = \boldsymbol{\varphi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)} + h\sum_{i=1}^{q-1} c_{p,i}\, \boldsymbol{v_i}\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + h\sum_{i=1}^{q} d_i\, \boldsymbol{v_i} \end{cases} \tag{17}$$

Since (14) is a 2-step technique, two numerical integration methods are required. On the one hand, that method defined by p, a's and b's is devoted to integrate the first step. On the other hand, that method with order q and Runge-Kutta weight and coefficients c's and d's, respectively, is devoted to integrate the second step.

As (14), Most of available high order Newton-like methods are, in fact, multipoint iterative schemes (see Theorem 1). The reader can check that the ideas introduced here can be applied to other high order Newton-like techniques in the same manner. Thus, for a three-step method, three different numerical integration techniques are needed, and so on.

3.3.- Introducing Regularization

Frequently, instability experienced by the map produced by (2) and other Newton-like methods, is due to the discontinuity of the inverse operator, in other words, the Jacobian matrix is not continuously invertible [45]. On the face of this situation, regularization methods arise as a powerful solution, improving the stability produced by the map (2).

The most typical regularization schemes are the Tikhonov [46] and Lavrentiev [47]. While the former is typically more stable and produces better condition numbers [48], the latter is less expensive and produces sparser matrices. For those reasons, we have only considered the Lavrentiev's regularization, however, the Tikhonov's scheme can be used in the same way. The Lavrentiev's regularization can be considered in PF by simply defining the following function:

$$\tilde{\boldsymbol{f}}(\boldsymbol{x},\alpha) = -[\boldsymbol{g}'(\boldsymbol{x}) + \alpha \boldsymbol{I}]^{-1}\boldsymbol{g}(\boldsymbol{x}) \qquad (18)$$

where $\alpha \in \mathbb{R}^+$ is the so-called regularization parameter. Basically, regularization consists in adding some elements to the Jacobian matrix in order to improve its conditioning. In that sense, regularization is similar to the parametrization used in the Continuation Power-Flow, however, since the Jacobian matrix is modified, it is possible that solution found does not correspond with the actual solution of the problem [21]. Here, the regularization parameter plays a key role since, when $\alpha = 0$, function (14) corresponds with (5) and the accuracy of the solution is ensured. Thus, α is typically updated each iteration and it is reduced until zero when the solution is approached. Other regularization matrices instead of $\boldsymbol{I}$ have been proposed in the literature (see e.g. [49]). However, the identity operator is by far the most widely used and studied.

Finally, as (6) is defined as the map resulting of applying a Runge-Kutta formula for integrating (5), the following map is the application of this family of numerical techniques for integrating (18):

$$\begin{cases} \widetilde{\boldsymbol{k}}_{\boldsymbol{1}} = \tilde{\boldsymbol{f}}\left(\boldsymbol{x}^{(k)}, \alpha\right) \\ \widetilde{\boldsymbol{k}}_{\boldsymbol{2}} = \tilde{\boldsymbol{f}}\left(\boldsymbol{x}^{(k)} + h\left(a_{21}\widetilde{\boldsymbol{k}}_{\boldsymbol{1}}\right), \alpha\right) \\ \vdots \\ \widetilde{\boldsymbol{k}}_{\boldsymbol{p}} = \tilde{\boldsymbol{f}}\left(\boldsymbol{x}^{(k)} + h\sum_{i=1}^{p-1} a_{p,i}\,\widetilde{\boldsymbol{k}}_{\boldsymbol{i}}, \alpha\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p} b_i\,\widetilde{\boldsymbol{k}}_{\boldsymbol{i}} \end{cases} \tag{19}$$

4.- Developed PF Solver

The modifications of CN described the preceding sections, are focused on improving either the computational performance or the robustness properties of CN. In this case, these modifications have been described separately, however, it is suitable using them simultaneously. For example, it is known that high order Newton-like methods are more sensitive to the starting guess than NR [50]. Thus, it is expected from the map (17) to be

weaker than (6). In that sense, it may be suitable to use regularization in consonance with the improvement described in Section 3.2.

Keeping this in mind, we propose to use the following methodology for solving PF, which uses simultaneously the three modifications described in Section 3:

$$\begin{cases} \widetilde{\boldsymbol{k}}_{\boldsymbol{1}} = \widetilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}, \alpha\right) \\ \widetilde{\boldsymbol{k}}_{\boldsymbol{2}} = \widetilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + h\left(a_{21}\widetilde{\boldsymbol{k}}_{\boldsymbol{1}}\right), \alpha\right) \\ \vdots \\ \widetilde{\boldsymbol{k}}_{\boldsymbol{p}} = \widetilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p-1} a_{p,i}\,\widetilde{\boldsymbol{k}}_{\boldsymbol{i}}, \alpha\right) \\ \boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + h\sum_{i=1}^{p} b_i\,\widetilde{\boldsymbol{k}}_{\boldsymbol{i}} \\ \widetilde{\boldsymbol{v}}_{\boldsymbol{1}} = \widetilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \alpha\right) \\ \widetilde{\boldsymbol{v}}_{\boldsymbol{2}} = \widetilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)} + h(c_{21}\widetilde{\boldsymbol{v}}_{\boldsymbol{1}}), \alpha\right) \\ \vdots \\ \widetilde{\boldsymbol{v}}_{\boldsymbol{q}} = \widetilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)} + h\sum_{i=1}^{q-1} c_{q,i}\,\widetilde{\boldsymbol{v}}_{\boldsymbol{i}}, \alpha\right) \\ \boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + h\sum_{i=1}^{q} d_i\,\widetilde{\boldsymbol{v}}_{\boldsymbol{i}} \end{cases} \tag{20}$$

where:

$$\widetilde{\boldsymbol{\phi}}(\boldsymbol{w}, \boldsymbol{z}, \alpha) = -[\boldsymbol{g}'(\boldsymbol{w}) + \alpha\boldsymbol{I}]^{-1}\boldsymbol{g}(\boldsymbol{z}) \tag{21}$$

$$\widetilde{\boldsymbol{\varphi}}(\boldsymbol{w}, \boldsymbol{u}, \boldsymbol{z}, \alpha) = -[\boldsymbol{g}'(\boldsymbol{w}) + \alpha\boldsymbol{I}]^{-1}\left(\boldsymbol{g}(\boldsymbol{u}) + \boldsymbol{g}(\boldsymbol{z})\right) \tag{22}$$

Equation (20) defines a generic formulation which allows, as (17), using two numerical integration methods. It is worth mentioning that solution paradigm (20) presents low computational burden, as Jacobian matrix is only factorized once each iteration. In addition, (20) is expected to be more reliable than (6) at turning points due to the regularization scheme employed.

In [13, 28], the author used the 4th order Runge-Kutta for developing a robust PF solver, in this paper, without loss of generality, we have followed the same idea. Thus, applying the 4th order Runge-Kutta formula for the two steps involved in (16), the following PF solver is defined:

$$
\begin{cases}
\tilde{\boldsymbol{k}}_1 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}, \alpha\right) \\
\tilde{\boldsymbol{k}}_2 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + \frac{h}{2}\tilde{\boldsymbol{k}}_1, \alpha\right) \\
\tilde{\boldsymbol{k}}_3 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + \frac{h}{2}\tilde{\boldsymbol{k}}_2, \alpha\right) \\
\tilde{\boldsymbol{k}}_4 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)} + h\tilde{\boldsymbol{k}}_3, \alpha\right) \\
\boldsymbol{y}^{(k)} = \boldsymbol{x}^{(k)} + \frac{h}{6}\left(\tilde{\boldsymbol{k}}_1 + 2\tilde{\boldsymbol{k}}_2 + 2\tilde{\boldsymbol{k}}_3 + \tilde{\boldsymbol{k}}_4\right) \\
\tilde{\boldsymbol{v}}_1 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \alpha\right) \\
\tilde{\boldsymbol{v}}_2 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)} + \frac{h}{2}\tilde{\boldsymbol{v}}_1, \alpha\right) \\
\tilde{\boldsymbol{v}}_3 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)} + \frac{h}{2}\tilde{\boldsymbol{v}}_2, \alpha\right) \\
\tilde{\boldsymbol{v}}_4 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)} + h\tilde{\boldsymbol{v}}_3, \alpha\right) \\
\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)} + \frac{h}{6}\left(\tilde{\boldsymbol{v}}_1 + 2\tilde{\boldsymbol{v}}_2 + 2\tilde{\boldsymbol{v}}_3 + \tilde{\boldsymbol{v}}_4\right)
\end{cases}
\tag{23}
$$

Method defines by (23) has been called Improved 4th order Runge-Kutta PF method (IRK4-PF). As observed, this technique presents various differences with respect those methods studied in [13, 28]:

- While the techniques in [13, 28] are devoted on integrating (5), our proposal integrates (21) and (22).
- The mapping (23) is considerably cheaper as just one matrix factorization is required each iteration.
- The mapping (23) uses two numerical techniques while the methods [13, 28] only considers one numerical arrangement.
- Our method incorporates regularization by integrating (21) and (22), while the methods in [13, 28] not.

As commented, this technique is expected to be less efficient than NR due to it involves other computations like vectors and matrix sums and products, however, this kind of calculations can typically be efficiently handled [13]. Nevertheless, IRK4-PF is focused on solving ill-conditioned systems, i.e. those cases which NR fails, therefore, IRK4-PF should be compared with other robust solvers rather than NR.

It is worth mentioning that the solution adopted in this paper (4th order Runge-Kutta in both steps), may not be the best choice. Other lighter schemes as Midpoint might offer better performance (see results obtained in [26]). However, with the aim of properly comparing with that methodology proposed in [13, 28], let us consider (23) in this paper, while the usage of other numerical integration methods will be studied in future works.

4.1.- Parameters Tuning

Developed IRK4-PF involves two parameters namely the step size (h) and regularization parameter (α). In this Section, we propose different updating mechanisms for these parameters.

As commented in Section 3.1, the developed solver achieves its highest convergence rate for $h = 1$, however, if h is fixed too large, the map defines by (23) may become unstable. A logic strategy is starting with a small step size and increasing it if the solution is properly approached. For determining if the solution is properly approached, we can use a Line Search strategy [29]. Thus, let us define the following indexes:

$$A_{red} = \left\| \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \right\|_{\infty}^{2} - \left\| \boldsymbol{g}\left(\boldsymbol{x}^{(k+1)}\right) \right\|_{\infty}^{2} \tag{24}$$

$$P_{red} = \left\| \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \right\|_{\infty}^{2} - \left\| \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) + \left[\boldsymbol{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{*} \boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right) \right\|_{\infty}^{2} \tag{25}$$

where $[\cdot]^{*}$ stands for the adjoint operator. Using (24) and (25) we can define the following ratio:

$$r = {}^{A_{red}}/_{P_{red}} \tag{26}$$

Then, the following rule is used for updating the step size:

$$h = \begin{cases} \min(\sigma h, h_{\max}), & \text{if } r < \rho \\ h, & \text{if } r \geq \rho \end{cases} \tag{27}$$

where $\sigma > 1$, $h_{\max} \in \mathbb{R}^+$ and $\rho \in \mathbb{R}^+$. Rule (27) determines that the solution is correctly approached if $r < \rho$, then, the step size could be enlarged. Thus, the parameter σ should be slightly higher than 1 in order to properly accelerate the convergence but avoiding abrupt changes on the step size parameter for avoiding numerical instability. As commented in Section 3.1, the step size should be upper bounded $h \leq 1$, this is the role of $h_{\max}$, which is recommended to be fixed $h_{\max} = 1$. Finally, A typical choice that often yields good results is $\rho = 1 \times 10^{-4}$ [29].

Regarding the regularization parameter, it should be progressively reduced until zero for ensuring the accuracy of the results [22]. In this paper, the following simple updating mechanism is used [51]:

$$\alpha = q^k \alpha \tag{28}$$

where $q \in (0,1)$. This section is concluded by summarizing in Algorithm 1 the proposed PF solution procedure based on IRK4-PF using pseudocode. In this case, $\|\boldsymbol{g}(\boldsymbol{x})\|_\infty$ is considered as convergence criterion. Algorithm also stops if the iteration counter k surpasses a threshold $k_{\max}$.

Algorithm 1: Main steps of IRK4-PF

Let $\boldsymbol{x}^{(0)}$, ε, $k_{\max}$, h, $h_{\max}$, σ, ρ, α and q be given
Initialize iteration counter $k \leftarrow 0$
while $\left\|\boldsymbol{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_\infty \geq \varepsilon$ **do**
    Solve (23)
    if $\left\|\boldsymbol{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_\infty < \varepsilon$
        break *#Convergence*
    else
        $k \leftarrow k + 1$
        if $k > k_{\max}$
            break *#Fail*
        end
        $h \leftarrow$ solve (27)
        $\alpha \leftarrow$ solve (28)
    end
return solution $\boldsymbol{x}^{(k)}$

For the different simulations presented, let us assume that ρ and q can be fixed according to the recommendations provided in [29] and [51], respectively. The other parameters could be tuned following logical strategies, thus we have taken $h_{\max} = 1$ and $\sigma = 1.1$. Regarding the initialization of h and α, there are not analytic rules for this purpose, so that these values have to be fixed heuristically or empirically. Regarding the step size, as commented, a logic strategy consists on taking it short at the beginning of the iterative procedure, when the solution is far away. With respect the regularization parameter, it should be initialized in a proper value for ensuring a good condition number of the Jacobian. A strategy that often brings good results is initializing h equal to 0.5 and α equal to 0.1.

4.2.- Illustrative Examples

Next, the procedure described by the Algorithm 1 is illustrated by using the following 2-dimensional example [14]:

$$\boldsymbol{g}(\boldsymbol{x})\begin{cases} g_1 = 2x_1^2 - 2x_1x_2 + 2x_2^2 - 2.24 \\ g_2 = -2x_1^2 - x_1x_2 + 2x_2^2 - 0.64 \end{cases} \tag{29}$$

Let us consider the following parameters $h_{\max} = 1$, $\sigma = 1.1$, $\rho = 1 \times 10^{-4}$ and $q = 0.2$. As starting point we take $\boldsymbol{x}^{(0)} = [1,1]^T$ as in [14] which the residuals are $\boldsymbol{g}(\boldsymbol{x}^{(0)}) = [-0.24, -1.64]^T$. With this estimation, the Jacobian of (29) is given by:

$$\boldsymbol{g}'(\boldsymbol{x}^{(0)}) = \begin{bmatrix} 2.1 & 2 \\ -5 & 1.1 \end{bmatrix} \tag{30}$$

Let us initialize the step size equal to 0.5 and $\alpha = 0.1$. So, from (23) one has:

$$\begin{cases} \tilde{\boldsymbol{k}}_1 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)}, 0.1\right) = [-0.245, 0.377]^T \\ \tilde{\boldsymbol{k}}_2 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)} + \frac{0.5}{2}\tilde{\boldsymbol{k}}_1, 0.1\right) = [-0.156, 0.232]^T \\ \tilde{\boldsymbol{k}}_3 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)} + \frac{0.5}{2}\tilde{\boldsymbol{k}}_2, \alpha\right) = [-0.189, 0.292]^T \\ \tilde{\boldsymbol{k}}_4 = \tilde{\boldsymbol{\phi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)} + 0.5\tilde{\boldsymbol{k}}_3, 0.1\right) = [-0.068, 0.052]^T \\ \boldsymbol{y}^{(0)} = \boldsymbol{x}^{(0)} + \frac{0.5}{6}\left(\tilde{\boldsymbol{k}}_1 + 2\tilde{\boldsymbol{k}}_2 + 2\tilde{\boldsymbol{k}}_3 + \tilde{\boldsymbol{k}}_4\right) = [0.917, 1.123]^T \\ \tilde{\boldsymbol{v}}_1 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, 0.1\right) = [-0.371, 0.557]^T \\ \tilde{\boldsymbol{v}}_2 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{y}^{(0)} + \frac{0.5}{2}\tilde{\boldsymbol{v}}_1, 0.1\right) = [-0.121, 0.063]^T \\ \tilde{\boldsymbol{v}}_3 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{y}^{(0)} + \frac{0.5}{2}\tilde{\boldsymbol{v}}_2, 0.1\right) = [-0.219, 0.325]^T \\ \tilde{\boldsymbol{v}}_4 = \tilde{\boldsymbol{\varphi}}\left(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)}, \boldsymbol{y}^{(0)} + 0.5\tilde{\boldsymbol{v}}_3, 0.1\right) = [-0.099, 0.006]^T \\ \boldsymbol{x}^{(1)} = \boldsymbol{y}^{(0)} + \frac{0.5}{6}\left(\tilde{\boldsymbol{v}}_1 + 2\tilde{\boldsymbol{v}}_2 + 2\tilde{\boldsymbol{v}}_3 + \tilde{\boldsymbol{v}}_4\right) = [0.821, 1.235]^T \end{cases} \tag{31}$$

The residuals for $\boldsymbol{x}^{(1)}$ are:

$$\boldsymbol{g}\left(\boldsymbol{x}^{(1)}\right) = [0.129, 0.048]^T, \left\|\boldsymbol{g}\left(\boldsymbol{x}^{(1)}\right)\right\|_\infty = 0.129 \tag{32}$$

Let us assume that (32) does not meet the required convergence tolerance, so that the developed algorithm has to continue with the following step and h and α are updated using (27) and (28), respectively.

$$A_{red} = 1.64^2 - 0.129^2 = 2.673 \tag{33}$$

$$P_{red} = 1.64^2 - \left\| \begin{bmatrix} 0.129 \\ 0.048 \end{bmatrix} + \begin{bmatrix} 2.1 & 2 \\ -5 & 1.1 \end{bmatrix}^* \begin{bmatrix} 0.129 \\ 0.048 \end{bmatrix} \right\|_\infty^2 = 2.563 \tag{34}$$

$$r = \frac{2.673}{2.563} = 1.043 > 1 \times 10^{-4} \tag{35}$$

$$\alpha = 0.2^1 0.1 = 0.02 \tag{36}$$

By the result (35), the step size is not updated, so that, it is equal to 0.5 for the following iteration while α is decreased by (36). Fig. 1 shows how h and α evolves for the problem (29). As observed, behaviour of these parameters was as expected. The step size tends to grow during the iterative procedure, whit the aim of speeding up the convergence while α decreases for ensuring the accuracy of the results obtained.

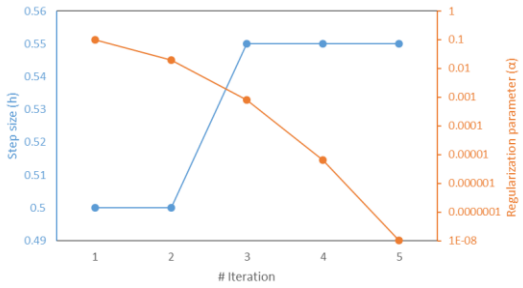

Fig. 1 Evolution of the step size and the regularization parameter for the example (29).

Next, let us consider the following scalar function:

$$g = \tan^{-1} x \tag{37}$$

NR fails for solving (37) taking $x^{(0)} = 1.5$ as showed Fig. 2. Oppositely, this function is easily solved using IRK4-PF in 3 iterations as showed Fig. 3.

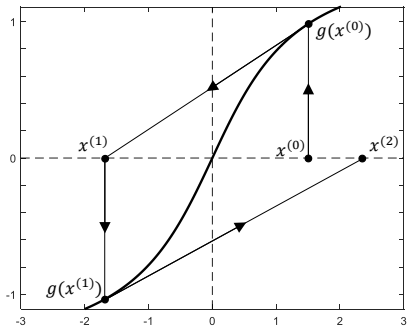

Fig. 2 Failure of NR for solving (37)

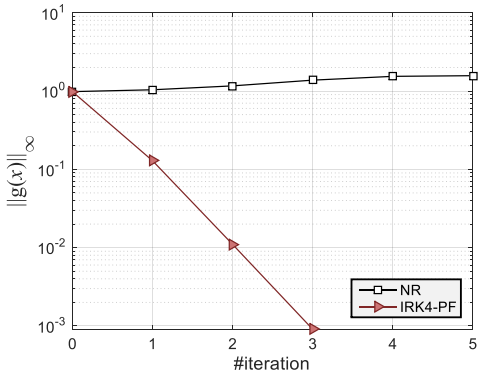

Fig. 3 Convergence profiles for the example (37)

4.3.- Handling Reactive Limits

Equation (23) does not take into account the generators' reactive limits, however, they could be easily incorporated using the strategy described in the flowchart of Fig. 4. This mechanism is simple and well-known. Nevertheless, the formulation developed in this paper is versatile enough to incorporate other strategies available in the literature. It is worth mentioning that similar strategies can be adopted for incorporating other equipment limits or controls like transformer taps or shift changers.

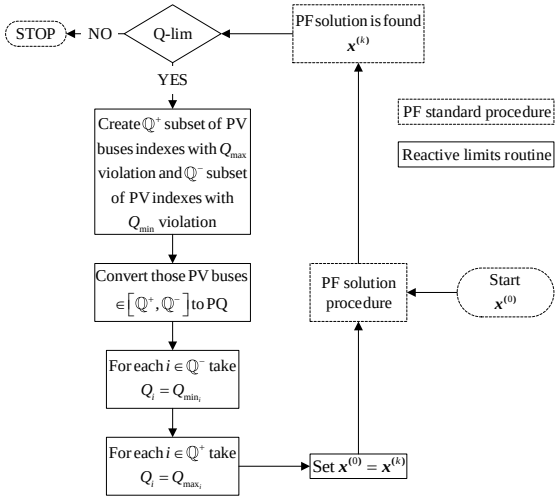

Fig. 4. Proposed flowchart for considering the generators' reactive limits.

5.- Validation

Before providing numerical results, the developed IRK4-PF should be validated using a well-known test case. For this purpose, we have used the well-known IEEE 30-, 57-, 118-, and 300-bus systems [52]. Using the values recommended in Section 4, the performance of IRK4-PF has been compared with NR. Table 1 shows the total iterations required by these methods for solving the different IEEE cases from a flat start ($\varepsilon = 10^{-5}$).

Table 1 Total Iterations for IEEE Cases

IEEE case	NR	IRK4-PF
30-bus	3	4
57-bus	3	4
118-bus	4	5
300-bus	4	5

As seen in Table 1, IRK4-PF required one more iteration in all cases, nevertheless, its convergence properties are reasonable if one takes into account that this technique is not focused on well-conditioned cases. Fig. 5 shows the voltage angles and magnitudes obtained in the IEEE cases by NR and IRK4-PF. From this figure one can observe that our method obtained very accurate results. Fig. 6 plots the evolution of h and α in the IEEE 30-bus case. As observed, the behaviour of these parameters was as expected. As commented, the developed IRK4-PF should usually present better condition numbers, so that the factorization of the Jacobian matrix is more stable due to the regularization scheme introduced. Fig. 7 plots the condition number of the factorization for the IEEE 30-bus case with NR and IRK4-PF. As seen in this figure, IRK4-PF produced better condition numbers, which are progressively increased as the regularization parameter decreases.

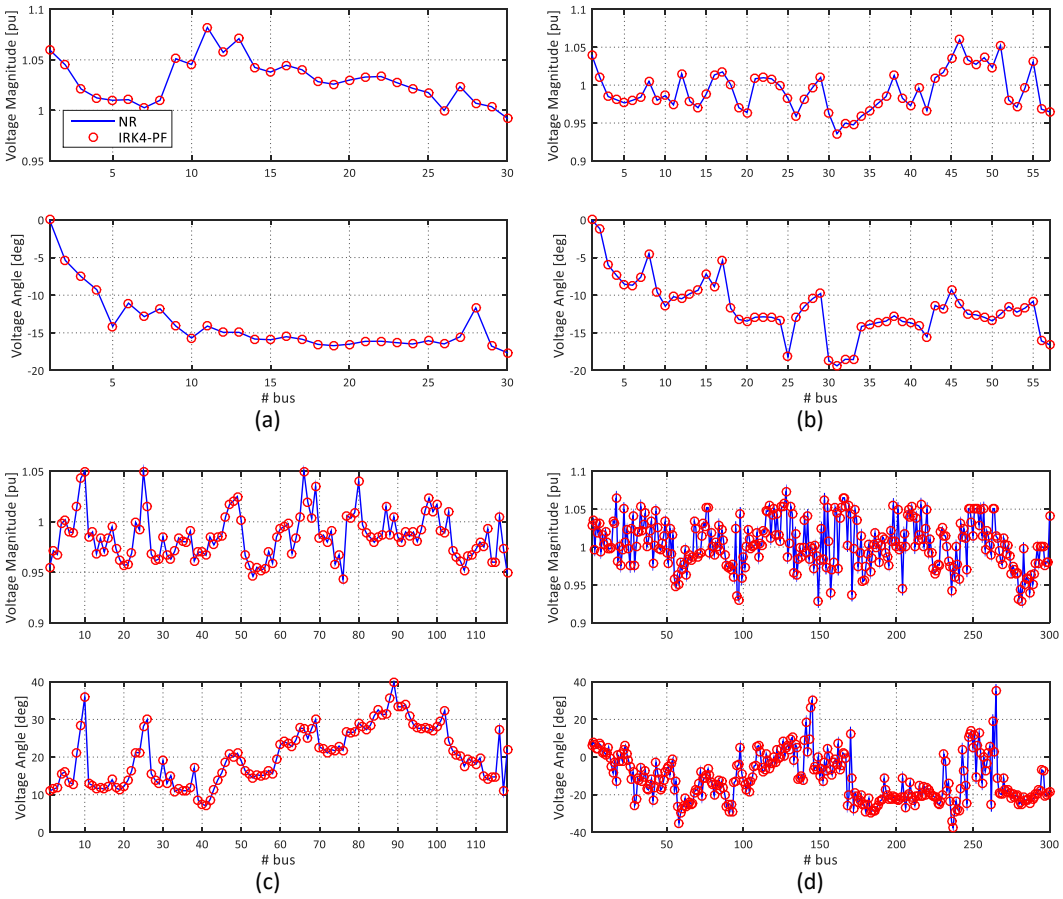

Fig. 5 Voltage angles and magnitudes obtained by NR and IRK4-PF in the IEEE 30-bus (a), 57-bus (b), 118-bus (c) and 300-bus (d).

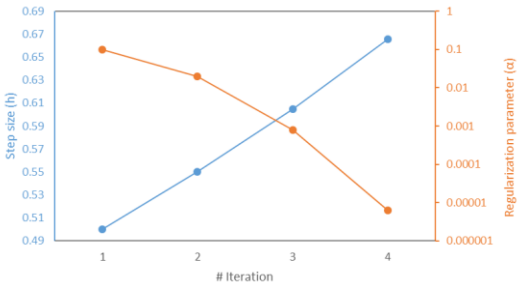

Fig. 6 Evolution of the step size and the regularization parameter for the IEEE 30-bus case.

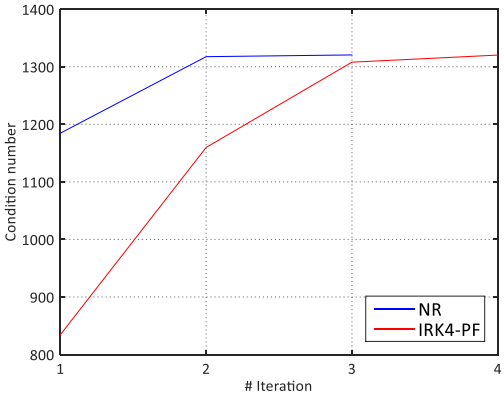

Fig. 7 Condition numbers of the factorization for the IEEE 30-bus case.

6.- Simulations and results

Several simulations have been made in order to show the performance of the developed IRK4-PF. The following PF solvers have been considered for comparison:

- Standard NR.
- Iwamoto's method [14]. Since this paper considers the PF problem in polar form, results reported correspond with the Iwamoto's method in polar coordinates developed in [15].
- Various PF solvers based on CN method like the Midpoint rule (MP-PF) [26], the 3rd order Heun's method (H3-PF) [26], the 4th order Runge-Kutta (RK4-PF) [13, 28] and the Composite Newton-Cotes formula (CNC-PF) [27].
- Hybrid 4th order Runge-Kutta-Broyden's method (RK4B-PF) developed in [23].
- Reverse-Bulirsch-Stoer (RBS-PF) in [25].
- Developed IRK4-PF.

For simulations, the following parameters have been taken for PF solution based on IRK4-PF: $h_{\max} = 1$, $\sigma = 1.1$, $\rho = 1 \times 10^{-4}$ [29] and $q = 0.2$ [51], while the step size has been initialized equal to 0.5 and the regularization parameter equal to 0.1. For RBS,

we have followed the guidelines facilitated in reference [25] for properly tuning the involved parameters. In order to be properly compared with IRK4-PF; those methods based on CN (i.e. MP-PF, H3-PF, RK4-PF, CNC-PF and RK4B-PF) use (27) for updating the step size and the same parameters and initializations that IRK4-PF.

All simulations have been run using a flat start (i.e. all initial voltage magnitudes equal to 1 pu and all initial voltage angles equal to 0 deg at PQ buses). In addition, $\varepsilon = 10^{-5}$ has been imposed as convergence criterion. Finally, $k_{\max} = 2000$ has been taken.

All simulations have been run under Windows 10 on a 64-bit i5-9400F Intel Core personal computer (2.90 GHz, 8 GB of RAM) using Matpower v7.0 [53]. It is worth mentioning that all reported solution times have been calculated as the average value of 100 simulations, with the aim of avoiding potential influence of parallel computer activities.

6.1.- Studied Systems

Several realistic large-scale ill-conditioned systems based on the Polish transmission system [54], European Transmission System [55, 56] and Synthetic grids [57] have been considered. The main characteristics of the studied systems, along the total number of variables n and the condition number of the Jacobian matrix at $k = 0$ [12] are summarized in Table 2. All considered systems are available within Matpower's database [54].

Table 2. Main Characteristics of the Studied Systems

		case3012wp	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
Buses		3012	3374	13659	70000
Branches		3572	4161	20467	88207
Generators		502	596	4092	10390
Load	MW	27169	48363	381431	594659
	MVar	10200	19527	98523	158529
n		5725	6355	23225	134104
Cond. number		2.9×10^6	4.3×10^6	1.5×10^8	4.8×10^8

The studied systems ranging from 3012- to 70000-bus, in addition, they are based on realistic transmission systems and are naturally ill-conditioned. Therefore, they are considered suitable for validating the developed PF solution method described by the Algorithm 1.

6.2.- Base Cases

Table 3 reports the total number of iterations required by the considered PF solution techniques for solving the studied systems for base conditions.

Table 3. Total Iterations for Base Cases

Method	*case3012wp*	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
NR	Fail	Fail	Fail	Fail
Iwam.	367	350	48†	Fail
MP-PF	29	30	27	Fail
H3-PF	16	17	15†	Fail
RK4-PF	21	22	Fail	Fail
RK4B-PF	21	22	Fail	Fail
RBS-PF	13	13	14	Fail
CNC-PF	Fail	Fail	7†	Fail
IRK4-PF	6	6	6	9

†: *Low voltage solution*

As expected, NR failed in all studied systems, which is symptomatic of ill-conditioning. RK4-PF and RK4B-PF failed in the *case13659pegase* while MP-PF, CNC-PF and the Iwamoto's method converged to the low voltage solution in this system. The Iwamoto's method along with MP-PF, H3-PF, RK4-PF, RK4B-PF, RBS and CNC-PF failed in the *case_ACTIVSg70k*, consequently, only the developed IRK4-PF successfully solved all studied systems. This outstanding robustness features are undoubtedly due to the improvement introduced in Section 3.3. For all cases, our method required less iterations than the other tested techniques.

Superior convergence features of IRK4-PF can be clearly appreciated in Fig. 8. In this figure, it can be observed as the residual showed by IRK4-PF was always lower than that

manifested by the remainder techniques. This fact along with the results reported in Table 2 demonstrates that improvement introduced in Section 3.2 brings higher convergence features.

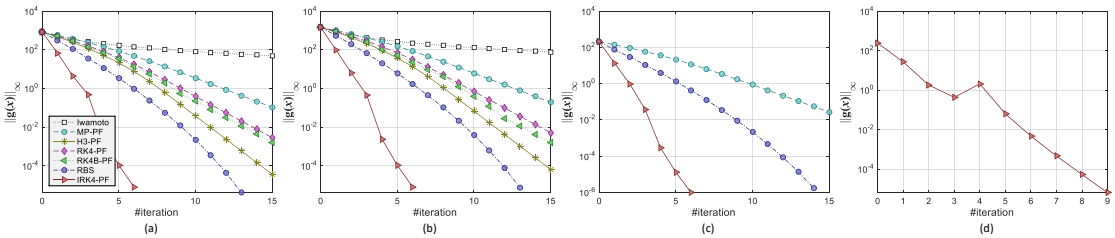

Fig. 8 Convergence profiles for the *case3012wp* (a), *case3375wp* (b), *case13659pegase* (c) and *case_ACTIVSg70k* (d).

Tables 4 and 5 report the solution times and the total factorizations required for base cases. Since the factorization of the Jacobian matrix is the heaviest part of a PF calculation, that method which required the minimum number of these computations should exhibit the most competitive execution time. Thus, the developed IRK4-PF was the most efficient method. Regarding RK4B-PF, it does not require to factorize the Jacobian matrix, however, it has to be explicitly inverted at first iteration, which is heavier than a LU decomposition. In addition, this technique requires various expensive vectors and matrix computations.

Table 4. Solution Times [s] for Base Cases

Method	*case3012wp*	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
Iwam.	7.11	7.55	4.15†	--
MP-PF	0.99	1.15	4.31	--
H3-PF	0.81	0.97	3.59†	--
RK4-PF	1.41	1.64	--	--
RK4B-PF	31.02	38.34	--	--
RBS-PF	0.77	0.87	3.92	--
CNC-PF	--	--	0.64†	--
IRK4-PF	0.11	0.15	0.43	6.26

†: *Low voltage solution*

Table 5. Comparison of the Total Factorizations Computed for Base Cases

Method	*case3012wp*	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
Iwam.	367	350	48	--
MP-PF	58	60	54	--
H3-PF	48	51	45	--
RK4-PF	84	88	--	--
RK4B-PF	0	0	--	--
RBS-PF	30	30	32	--
CNC-PF	--	--	7	--
IRK4-PF	6	6	6	9

6.3.- Reactive limits enforcement

Now, let us analyse the performance of considered PF solvers with reactive limits enabled. For taking into account these limits, the strategy described in Section 4.3 has been considered. The total number of iterations and solution times (in parenthesis) for this scenario are reported in Table 6. As expected, more iterations are normally required in comparison with the base case scenario as various PF problems have to be solved for reaching a feasible solution. Nevertheless, similar conclusions as for the base case scenario can be extracted in this case.

Table 6 Comparison of the total number of iterations and computation time [s] (in parenthesis) with reactive limits

Method	*case3012wp*	*case3375wp*	*case13659pegase*	*case_ACTIVSg70k*
NR	Fail	Fail	Fail	Fail
Iwam.	777 (15.36)	765 (16.81)	85 (7.39)[†]	Fail
MP-PF	59 (2.06)	66 (2.57)	46 (7.39)	Fail
H3-PF	35 (1.81)	41 (2.37)	29 (6.99)[†]	Fail
RK4-PF	45 (3.06)	51 (3.88)	Fail	Fail
RK4B-PF	45 (71.90)	50 (102.27)	Fail	Fail
RBS-PF	29 (1.84)	32 (2.39)	24 (6.88)	Fail
CNC-PF	Fail	Fail	14 (1.28) [†]	Fail
IRK4-PF	13 (0.34)	15 (0.72)	10 (1.04)	27 (34.43)

[†]: *Low voltage solution*

6.4.- Influence of the Loading Level

It is well-known that the loading level affects the performance of PF solvers, typically increasing the total number of iterations required for achieving the PF solution [58]. This Section is devoted on studying how this topic affects the performance of the developed

technique. To do that, various simulations are carried out. Fig. 9 plots the total iterations required by IRK4-PF for solving the *case3012wp* for different loading and generation levels. As observed, the developed technique may require up to three more iterations with respect the base case when the loading level is increased. If the generation also grows, the stress of the system is alleviated and our method performs better. IRK4-PF may fail if the loading level surpasses the maximum loadability point of the system. This occurs when the loading is heavily increased but the generation not.

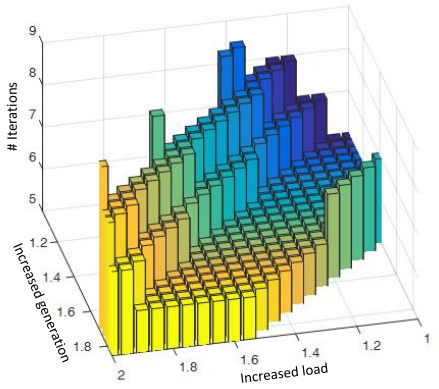

Fig. 9 Total Iterations employed by IRK4-PF for solving the *case3012wp* with different load and generation profiles (1 = base case).

Next, let assume that not all loads are increased. In the following simulation, a percentage of loads are increased by 30%, while the generation is kept as base case. Which loads are increased are determined randomly and 10 different simulations are considered for each percentage. Fig. 10 shows how the average total iterations grows as more loads are increased. Fig. 11 reflects the opposite scenario, i.e. all the loads are increased by 30% while a percentage of units increases their generation by 30%. As observed, our method tends to employ less iteration as more generators are able to increase their delivered power.

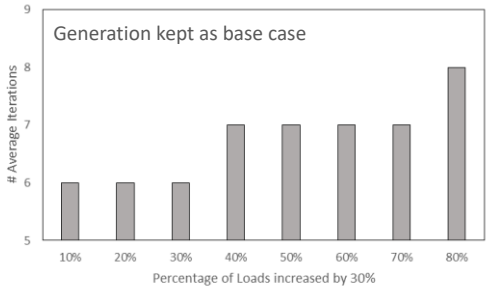

Fig. 10 Average total iterations for 10 simulations which a percentage of loads is increased by 30% while the generation is kept as base case. Results obtained by IRK4-PF for the *case3012wp*.

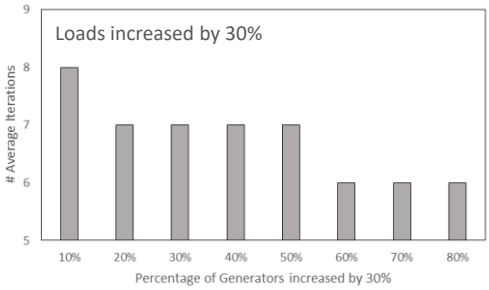

Fig. 11 Average total iterations for 10 simulations which a percentage of generators increase their power delivered by 30% while loads are increased by 30%. Results obtained by IRK4-PF for the *case3012wp*.

7.- Conclusions and Future Works

In this paper, several modifications on the original structure of CN paradigm have been introduced, for improving either its computational efficiency or robustness properties. As result, an alternative solution framework has been developed. This proposal allows to develop robust and PF solvers considering different numerical techniques (as CN). In this paper, a novel PF solver using the 4th order Runge-Kutta method has been developed for illustrating the features of the developed paradigm. Various simulations have been carried out in several realistic large-scale ill-conditioned systems have served for showing the performance of our proposal. The developed solver has been compared with various conventional and modern robust techniques, showing a competitive performance.

The developed solution framework is opened to several future works. For example, the applicability of other numerical integration techniques should be studied. In addition,

other PF solvers based on the introduced solution paradigm might offer good results in other Power System tools like the Continuation Power-Flow [20]. These topics should be covered in future works.

References

[1] Grainger JJ, Stevenson WD. Power System Analysis. New York: McGraw-Hill; 1994.

[2] Tinney WF, Hart CE. Power flow solution by Newton's method. IEEE Trans Power App Syst 1967; PAS-86: 1449-60.

[3] Xie N, Torelli F, Bompard E, Vaccaro A. Dynamic computing paradigm for comprehensive power flow analysis. IET Gener Transm Distrib 2013; 7(8): 832-42.

[4] Wang L, Lin XR. Robust fast decoupled power flow. IEEE Trans Power Syst 2000; 15(1): 208-15.

[5] Yang J, Zhang N, Kang C, Xia Q. A State-Independent Linear Power Flow Model With Accurate Estimation of Voltage Magnitude. IEEE Trans Power Syst 2017; 32(5): 3607-17.

[6] Milano F. Power System Modelling and Scripting. New York: Springer; 2010.

[7] Kersting WH. Distribution System Modeling and Analysis. 2nd ed. Boca Raton, FL: CRC Press; 2006.

[8] Derakhshandeh SY, Pourbagher R. Application of high-order Newton-like methods to solve power flow equations. IET Gener Transmiss Distrib 2016; 10(8); 1853-9.

[9] Sttot B. Review of load flow calculation methods. Proc IEEE 1974; 62(7): 916-29.

[10] Gómez-Expósito A, Conejo A, Cañizares C. Electric Energy Systems: Analysis and Operation. Boca Raton, FL, USA: CRC; 2008.

[11] Karimi M, Shahriari A, Aghamohammadi MR, Marzooghi H, 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.

[12] Pourbagher R, Derakhshandeh SY. A powerful method for solving the power flow problem in the ill-conditioned systems. Int J Elect Power Energy Syst 2018; 94: 88-96.

[13] Milano F. Continuous Newton's Method for Power Flow Analysis. IEEE Trans Power Syst 2009; 24(1); 50-7.

[14] Iwamoto S, Tamura Y. A load flow calculation method for ill-conditioned power systems. IEEE Trans Power App Syst 1981; PAS-100(4): 1736-43.

[15] Braz LMC, Castro CA, Murati CAF. A critical evaluation of step size optimization based load flow methods. IEEE Trans Power Syst 2000; 15(1): 202-7.

[16] Tate JE, Overbye TJ. A comparison of the optimal multiplier in polar and rectangular coordinates. IEEE Trans Power Syst 2005; 20(4): 1667-74.

[17] Overbye TJ. A power flow measure for unsolvable cases. IEEE Trans Power Syst 1994; 9(3): 1359-65.

[18] Overbye TJ. Computation of a practical method to restore power flow solvability. IEEE Trans Power Syst 1995; 10(1): 280-7.

[19] Seydel R. Practical Bifurcation and Stability Analysis: From Equilibrium to Chaos. 2nd ed. New York: Springer-Verlag; 1994.

[20] Ajjarapu V, Christy C. The continuation power flow: a tool for steady state voltage stability analysis. IEEE Trans Power Syst 1992; 7(1): 416-23.

[21] Trias A, Marín JL. The Holomorphic Embedding Loadflow Method for DC Power Systems and Nonlinear DC Circuits. IEEE Trans Circuits Syst I: Regular Papers 2016: 63(2); 322-33.

[22] Milano F. Analogy and Convergence of Levenberg's and Lyapunov-Based Methods for Power Flow Analysis. IEEE Trans Power Syst 2016; 31(2): 1663-4.

[23] Tostado-Véliz M, Kamel S, Jurado F. Development of combined Runge-Kutta Broyden's load flow approach for well- and ill-conditioned power systems. IET Gener Transm Distrib 2018; 12(21): 5723-29.

[24] Tostado-Véliz M, Kamel S, Jurado F. Development of Different Load Flow Methods for solving Large-scale Ill-conditioned Systems. Int Trans Elect Energy Syst 2018; 29(4). https://doi.org/10.1002/etep.2784

[25] Tostado-Véliz M, Kamel S, Jurado F. A robust Power Flow Algorithm Based on Bulirsch-Stoer Method. IEEE Trans Power Syst 2019; 34(4): 3081-9.

[26] Tostado-Véliz M, Kamel S, Jurado F. Comparison of various robust and efficient load-flow techniques based on Runge–Kutta formulas. Elect Power Syst Res 2019; 174. https://doi.org/10.1016/j.epsr.2019.105881

[27] Tostado-Véliz M, Kamel S, Jurado F. A powerful power-flow method based on Composite Newton-Cotes formula for ill-conditioned power systems. Int J Elect Power Energy Syst 2020; 116, 105558.

[28] Suresh Reddy S, Sarat Kumar S, Kumar SVJ. Load Flow Solution for Ill-conditioned Power Systems Using Runge-Kutta and Iwamoto Methods with FACTS Devices. J Theor Appl Inf Technol 2009; 5(6): 693-703.

[29] Idema R, Lahaye DJP. Computational Methods in Power System Analysis. In: Schilders W, editor. Atlantis Studies in Scientific Computing in Electromagnetics, Paris, France: Atlantis Press; 2014.

[30] Zhu J. Optimization of Power System Operation. New York: Wiley-IEEE Press; 2015.

[31] Xie N, Bompard E, Napoli R, Torelli F. Widely convergent method for finding solutions of simultaneous nonlinear equations. Elect Power Syst Res 2012; 83: 9-18.

[32] Rao SD, Tylavsky DJ. Theoretical convergence guarantees versus numerical convergence behavior of the holomorphically embedded power flow method. Int J Elect Power Energy Syst 2018; 95: 166-76.

[33] Feng Y, Tylavsky DJ. A Holomorphic embedding approach for finding the Type-1 power-flow solutions. Int J Elect Power Energy Syst 2018; 102: 179-88.

[34] Yang X, Zhou X. Application of asymptotic numerical method with homotopy techniques to power flow problems. Int J Elect Power Energy Syst 2014; 57: 375-83.

[35] Tostado M, Kamel S, Jurado F. Several robust and efficient load flow techniques based on combined approach for ill-conditioned power systems. Int J Elect Power Energy Syst 2019; 110: 349-56.

[36] Brill M, Schock E. Iterative solutions of ill-posed problems: a survey. In: Vogel A, editor. Model Optimization in Exploration Geophysics, Braunschweig: Vieweg; 1987.

[37] Saadat H. Power System Analysis. 3rd ed. North York: PSA Publishing; 2010.

[38] Butcher, JC. Numerical methods for ordinary differential equations. 2nd ed. Chichester: Wiley: 2008.

[39] Cordero A, Jordán C, Sanabria-Codesal E, Torregrosa JR. Highly efficient iterative algorithms for solving nonlinear systems with arbitrary order of convergence $p+3$, $p \geq 5$. J Comput Applied Math 2017; 330: 748-58.

[40] Traub JF. Iterative Methods for the Solution of Equations. New York: Chelsea; 1982.

[41] Petkovic MS, Neta B, Petkovic LD, Dzunic J. Multipoint Methods for Solving Nonlinear Equations. Cambridge, MA, USA: Academic Press; 2013.

[42] Darvishi MT, Barati A. A third-order Newton-type method to solve systems of nonlinear equations. Appl Math Comput 2007; 187(2): 630-5.

[43] Xiao X, H Yin. Achieving higher order of convergence for solving systems of nonlinear equations. Appl Math Comput 2017; 311: 251-61.

[44] Sharma JR, Gupta, P. An efficient fifth order method for solving systems of nonlinear equations. Comput Math Appl 2014; 67(3): 591-601.

[45] Pereverzev S, Schock B. On the adaptive selection of the parameter in regularization of ill-posed problems. SIAM J Numer Anal 2006; 43(5): 2060-76.

[46] Kang CG, He GQ. A mixed Newton-Tikhonov method for nonlinear ill-posed problems. Appl Math Mech-Engl Ed 2009; 30(6): 741-52.

[47] Argyros IK, George S. Regularization Methods for Ill-Posed Problems with Monotone Nonlinear Part. Punjab Univ J Math 2014; 46(1): 25-38.

[48] George S, Nair MT. Error bounds and parameter choice strategies for simplified regularization in Hilbert scales. Int Eq Oper Th 1997; 29(2): 231-42.

[49] Huang G, Nochese S, Reichel L. Regularization matrices determined by matrix nearness problems. Lin Alg Appl 2016; 502: 41-57.

[50] Ezquerro JA, Hernández MA. An optimization of Chebyshev's method. J Complexity 2009; 25: 343-61.

[51] Liu W, Wu C. A predictor-corrector iterated Tikhonov regularization for linear ill-posed inverse problems. Appl Math Comput 2013; 221: 802-18.

[52] Peyghami S, Davari P, Fotuhi-Firuzabad M, Blaabjerg F. Standard Test Systems for Modern Power System Analysis: An Overview. IEEE Ind Electron Mag 2019; 13(4): 86-105.

[53] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. Matpower: Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education. IEEE Trans Power Syst 2011; 26(1): 12-9.

[54] Matpower's database, http://www.pserc.cornell.edu/matpower/ [accessed 14 January 2020]

[55] Josz C, Fliscounakis S, Maeght J, Panciatici P. AC Power Flow Data in Matpower and QCQP Format: iTesla, RTE Snapshots, and PEGASE, http://arxiv.org/abs/1603.01533 [v3 30 March 2016]

[56] 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(4): 4909-17.

[57] Birchfield AB, Xu T, Gegner KM, Shetye KS, Overbye TJ. Grid Structural Characteristics as Validation Criteria for Synthetic Networks. IEEE Trans Power Syst 2017; 32(4): 3258-65.

[58] Tostado M, Kamel S, Jurado F. Developed Newton-Raphson based Predictor-Corrector load flow approach with high convergence rate. Int J Elect Power Energy Syst 2019; 105: 785-92.