An Efficient and Reliable Power Flow Solution Method for Large Scale Ill-Conditioned Cases Based on the Romberg's Integration Scheme

Marcos Tostado-Véliz[1], Salah Kamel[2,3], Francisco Jurado[1,*]

[1]*Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén, Spain*
[2]*Electrical Engineering Department, Faculty of Engineering, Aswan University, 81542 Aswan, Egypt*
[3]*State Key Laboratory of Power Transmission Equipment and System Security and New Technology, Chongqing University, Chongqing 400030, China*

Abstract **–** Recently, the Power Flow solution in large scale ill-conditioned systems has attracted huge attention. Despite the considerable efforts conducted in this line, this issue may be considered an open topic nowadays. Consequently, the paper aims at filling this gap by proposing a novel robust and efficient Power Flow solution technique inspired by the Romberg's Integration Scheme. The resulting algorithm is numerically stable and has a computational burden comparable to the Newton's technique. Several numerical results in different ill-conditioned systems from 3012-, to 13659-bus systems serve to show the performance of the developed method. Other available Power Flow solvers are considered for comparison. In all studied cases, results obtained with the developed method were promising, outperforming the other considered methodologies.

Keywords: Power Flow, Large-scale networks, Ill-conditioned cases, Romberg's method.

---

*Corresponding author, Tel.: +34 953 648518; Fax: +34 953 648586.
E-mail addresses: fjurado@ujaen.es (F. Jurado), mtostado@ujaen.es (M. Tostado), skamel@aswu.edu.eg (S. Kamel)

1.- Introduction

The Power Flow (PF) problem supposes some important numerical challenges. One of the most studied is the PF solution of ill-conditioned systems. Nowadays, ill-conditioned cases appear very frequently [1], which has recently motivated many research efforts focused on the solution of this kind of cases. Most of these works have been conducted on the solution of large scale ill-conditioned systems. These systems entail important difficulties since, along an acceptable numerical robustness, a high degree of computational efficiency is demanded in order to properly handle large vectors and matrix computations. After a careful review of the available literature, the authors consider that the efficient solution of large scale ill-conditioned systems is still an open topic.

PF problem finds numerous applications such as power system planning, operation or security analysis [2]. The first PF formulations were treated in the mid 50's [3]. The most standard PF solution based on Newton's technique was proposed [4]. Posteriorly, the intrinsic decoupling characteristics of the PF equations were exploited [5, 6], in order to develop a very efficient solver. However, this kind of technique presents a poor behaviour in distribution or stressed systems. On the other hand, various second order methodologies have been also explored [7]. The so-called DC Power Flow [8], has been widely used in some power system applications like security analysis. It exploits a linear approximation of the PF equations in order to obtain a solution directly without needing an iterative procedure. However, this kind of technique does not solve the reactive power problem, which is not arguable in some problems like Voltage Stability Analysis. Nevertheless, very recent works has aimed to overcome this drawback [9, 10]. The High order Newton-like methods have been extensively studied for solution of nonlinear systems [11, 12], however, its application for PF analysis has not been fully studied yet.

Finally, the authors refer the interested reader to the recent critic review [13], for a further analysis of the state of art of Newton-based PF solvers.

Alternatively, different PF formulations have been proposed and investigated. In [14], a novel formulation based on current injection approach was developed and posteriorly embellished in [15]. This formulation is interesting since the most of elements of Jacobian matrix are constant and the computation time is fast. However, the PF equations tend to be worse conditioned when they are raised in current injections form, consequently, this approach is not suitable for ill-conditioned cases. Kamel et al proposed a hybrid current-power mismatch formulation in [16]. It aims at taking advantage of the two approaches. The PF equations were posed in dq coordinates in [17, 18]. More recently, the complex form of the PF equations has been studied in [19].

The so-called ill-conditioned systems pose some numerical challenges for most of conventional PF solution methods. Typically, ill-conditioning of the PF equations is propitiated by large R/X ratios or when the loading level of the system approaches its Maximum Loadability point (MLP). Nevertheless, it is difficult to discern if a power network is naturally ill-conditioning by just measure these factors. During this work, we have followed the definition of ill-conditioned system provided by Milano in [20], which is provided below.

Definition 1. *Ill-conditioned system:* a power system case will be called ill-conditioned (or ill-posed) if, despite its solution exists, it is not reachable using the Newton-Raphson method (NR) and a flat start (e.g. all load voltage magnitudes equal to 1 and all voltage angles equal to 0).

The solution of ill-conditioned cases was initially tackled by Sasson et al [21], Iwamoto and Tamura [22] and other authors by developing methodologies based on the "optimal multiplier approach" [23]. This kind of techniques are quite robust and barely

affected by the starting guess. However, they have been reported very slow in some references (see e.g. [12]). The optimal multiplier-based techniques have been profusely used for miscellaneous PF related tools. For example, in [24, 25], this kind of techniques are considered for managing with unsolvable cases, proposing measures for restoring the solvability of the system. On the other hand, the optimal multiplier-based PF solution methods have been used for calculating the low voltage solutions of the PF equations [26] and for facing some important challenges related with the PF calculation such as automatic controls, heavy load, or problematic contingency, among others [27].

Ajjarapu and Christy proposed the well-known Continuation Power Flow [28], in order to avoid the difficulties experienced by Newton-like methods in the vicinity of MLP. This tool has found multiple applications in voltage stability analysis, however, it has not been profusely used for PF solution due to its low degree of efficiency. Alternatively, the singularity of the Jacobian matrix at turning points may be avoided using the Regularization or Levenberg-type methods [29, 30]. Nevertheless, this kind of techniques may occasionally converge to a value which is not actually a valid solution for the PF equations [31].

On the basis of the Continuous Newton's method [32], any numerical technique (e.g. the family of Runge-Kutta formulas) can be considered for PF solution. This idea was firstly exploited by Milano in [20] and posteriorly by the authors in [33-35]. This kind of methods are robust enough to be used in most of ill-conditioned cases, nevertheless, they are inefficient to be widely employed in large scale systems. In addition, although wider convergent than NR, they are still sensitive to the starting guess.

The PF problem may also be raised as an artificial dynamic system, and be solved using integration techniques like the ode routines of Matlab. This idea has been exploited in several references [1, 36]. The resulting dynamic paradigm is barely affected by the

initial point, however, it is computationally expensive and no tractable in large scale systems.

As deduced from the literature review, most of available robust PF solution methods are in fact not efficient enough to be widely used in large scale systems. This is an important drawback since this aspect has limited the application of this kind of solvers in industry applications, where the efficiency turns out to be a critical aspect especially for online applications. Taking into account that ill-conditioned systems are becoming more frequent [1], it is suitable to develop robust PF solvers efficient enough to be considered for industry tools. This paper aims at filling this gap by proposing a robust and efficient PF solver inspired by the Romberg's integration scheme [37]. Although the proposed PF method is based on a numerical integration technique, it presents some important differences with respect those techniques derived from the Continuous Newton's method [20] (See Section 3.4).

Stability and fixed points of the developed solver are analysed. Moreover, several large scale ill-conditioned systems under different demand scenarios serve to show the performance of the proposed method, which is compared with other available techniques.

Remainder of this paper is organized as follows. Section 2 outlines the necessary background. The developed PF solver is deeply described in Section 3. An illustrative example is considered for describing the developed solution procedure and check its effectiveness in Section 4. Section 5 presents various numerical experiments with results. Finally, the main conclusions are duly drawn in Section 6.

2.- Background

2.1.- PF problem and its solution using NR

In their most general form, the PF equations are established as a set of n nonlinear algebraic equations as follows:

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

where, $\mathbf{g}$: $\mathbb{R}^n \longmapsto \mathbb{R}^n$ are the PF equations, which in polar coordinates are given by:

$$\mathbf{g}(\boldsymbol{x}) = \begin{cases} \mathbf{g_P}, \text{For all buses} \\ \mathbf{g_Q}, \text{For PV buses} \end{cases} \tag{2}$$

$$g_{P_i} = P_i^{sch} - \sum_{j=1}^{n} |V_i| \left|V_j\right| \left|Y_{ij}\right| \cos\left(\theta_{ij} - \delta_i + \delta_j\right) \tag{3}$$

$$g_{Q_i} = Q_i^{sch} - \sum_{j=1}^{n} |V_i| \left|V_j\right| \left|Y_{ij}\right| \sin\left(\theta_{ij} - \delta_i + \delta_j\right) \tag{4}$$

where, $P_i^{sch} \in \mathbb{R}$ and $Q_i^{sch} \in \mathbb{R}$ are the active and reactive power injected at i^{th} bus, respectively. $V_i \angle \delta_i \in \mathbb{C}$ is the complex voltage at i^{th} bus. $Y_{ij} \angle \theta_{ij} \in \mathbb{C}$ is the ij^{th} element of the admittance matrix. In polar coordinates, the voltage angles at PQ and PV buses along the voltage magnitudes at PQ buses constitute the unknowns. Thus, the PF state vector $\boldsymbol{x} \in \mathbb{R}^n$ may be written as follows:

$$\boldsymbol{x} = \left[\boldsymbol{\delta_{PV}}, \boldsymbol{\delta_{PQ}}, \boldsymbol{V_{PQ}}\right]^T \tag{5}$$

where, $\boldsymbol{\delta_{PV}} \in \mathbb{R}^{n_g}$ is the vector of voltage angles at PV buses, $\boldsymbol{\delta_{PQ}} \in \mathbb{R}^{n_g}$ is the vector of voltage angles at PQ buses and $\boldsymbol{V_{PQ}} \in \mathbb{R}^{n_l}$ is the vector of voltage magnitudes at PQ buses. On the other hand, $n_g \in \mathbb{N}$ and $n_l \in \mathbb{N}$ represent the total number of PV and PQ buses, respectively.

Solution of the set of equations (1) using NR leads to the following iterative map:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \left[\mathbf{g}'\left(\boldsymbol{x}^{(k)}\right)\right]^{-1} \mathbf{g}\left(\boldsymbol{x}^{(k)}\right) \tag{6}$$

where, the superindexes indicate the iteration counter and $\mathbf{g}' = \nabla_{\boldsymbol{x}} \mathbf{g} \in \mathbb{R}^{n \times n}$ is the Jacobian matrix of the PF equations.

2.2.- Romberg's Integration Scheme

In this Section, let us briefly outline the main aspects related to the Romberg's Integration Scheme. To do that, a generic integral over the interval $[a, b]$ is considered (namely $\int_a^b f(x)dx$). The Romberg's method solves this problem by making up a particular linear combination of its values obtained by different numerical grids. During this stage, the variables are recursively refined for, posteriorly, making use of the Richardson's Extrapolation formula [38]. Then, the result of this step comes back to the first stage defining a recursive loop, which is repeated until a desired level of accuracy is achieved. The main steps of the Romberg's integration scheme for solving the integral $\int_a^b f(x)dx$ are summarized in Fig. 1.

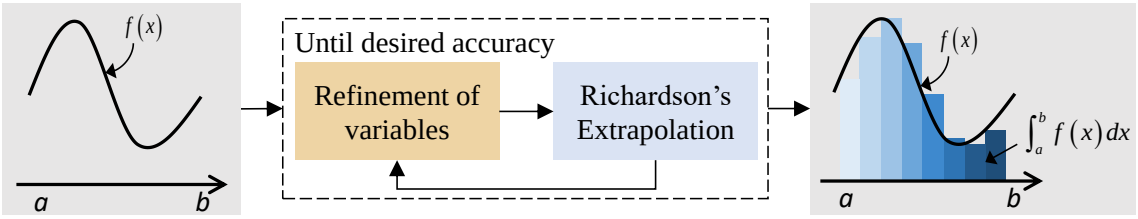

Fig. 1 Flowchart of Romberg's Integration method

3.- Proposed PF solver

3.1.- Foundations and Formulation

A nonlinear equation is occasionally assimilated as an integral problem, actually, the Newton's map (6) can be deduced from this approach (see [39]). Then, one can immediately guess that any numerical integration routine could be fully exploited for solving the PF equations (1). This Section is devoted on explaining a PF solver based on the Romberg's Integration scheme described in Section 2.2. Firstly, let us define for convenience:

$$\boldsymbol{\phi}(\boldsymbol{u}, \boldsymbol{v}) = -[\mathbf{g}'(\boldsymbol{u})]^{-1}\mathbf{g}(\boldsymbol{v}) \tag{7}$$

where, $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ are arbitrary vectors. One should note that, in the case of $\boldsymbol{u} = \boldsymbol{v} = \boldsymbol{x}$, equation (7) takes the form of the Newton's increment vector (6).

Considering $\boldsymbol{x}^{(k)}$ the initial point of the interval under study (i.e. the point a by assimilating with the procedure described in Section 2.2), the other extreme point of this interval is calculated as follows:

$$\boldsymbol{b} = \boldsymbol{x}^{(k)} + H\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right) \tag{8}$$

where, $H \in \mathbb{R}^+$ is a discrete step size. Equation (8) says that our considered interval will be $\left[\boldsymbol{x}^{(k)}, \boldsymbol{b}\right]$, where $\boldsymbol{b}$ corresponds with the increment of the PF state vector using NR with a step size H such as it is described in Fig. 2 for the case of $n = 1$.

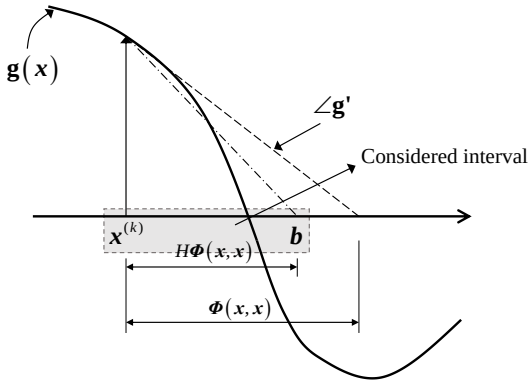

Fig. 2 This figure describes how the extreme $\boldsymbol{b}$ is calculated using a step size-based Newton's method (one should note that this figure describes the one variable case)

Now, following the Romberg's procedure, the vector of variables $\boldsymbol{x}$ can be refined by evaluating it at different points. This is achieved as follows:

$$\boldsymbol{c}_j = \boldsymbol{c}_{j-1} + h_j \sum_{l=0}^{j-1} \boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_l\right), \text{with } \boldsymbol{c}_0 = \boldsymbol{x}^{(k)} \tag{9}$$

where:

$$h_j = {}^{h_{j-1}}/_{2^{j-1}}, \text{with } h_1 = H \tag{10}$$

Basically, equation (9) defines a recursive loop which the variables are evaluated using different fine discrete steps namely h_j. Clearly, some criterion should be considered for finalizing the loop described by (9). For the Romberg's method, the accuracy of the results obtained is used for breaking this recursive algorithm, we have attended in this

case to computational aspects. One should note that, each step of the recursive formula (9), a function evaluation, a vector sum and a linear system solution have to be computed. Therefore, more steps hopelessly entail higher computational burden. In this case, we propose to interrupt this loop when $j = N$, where $N \in \mathbb{N}$ is a predefined parameter.

Finally, we apply the Richardson's Extrapolation between $\boldsymbol{b}$ and $\boldsymbol{c}_N$ such as it is defined in [40]:

$$\boldsymbol{x}^{(k+1)} = \frac{1}{r^{\psi}-1}\left(r^{\psi}\boldsymbol{c}_N - \boldsymbol{b}\right) \tag{11}$$

where, $r, \psi \in \mathbb{R}$. It is worth noting that, unlike to the Romberg's standard procedure, we simply apply the Richardson's Extrapolation once, without defining a recursive formula with the loop (9). Simulation results show that this approach is normally enough to obtain a good approximation of the PF solution (namely $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$). Consequently, this approach is recommended in order to reduce as much as possible the computational cost of the proposed algorithm. Fig. 3 summarized the main steps involved in the proposed PF solver inspired by the Romberg's Integration method.

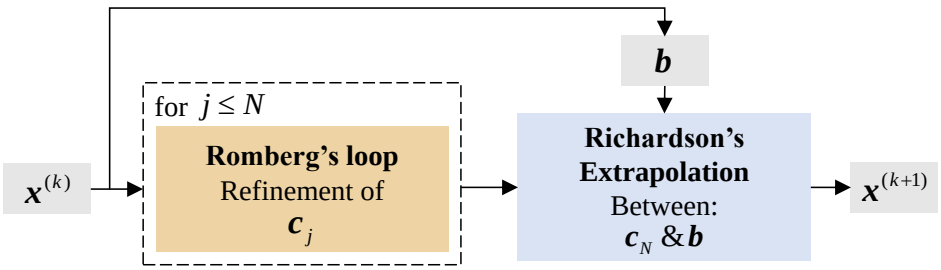

Fig. 3 Main steps of the developed PF solver

3.2.- Computational Implementation

Resulting algorithm described in Section 3.1 presents various degrees of freedom. While r and ψ can be tuned following the guidelines provided in [40], and do not need to

be updated during the iterative procedure, we recommend to establish some updating mechanism for H and N.

It is worth noting that both H and N have a direct impact on the overall performance of the proposed algorithm. Thus, the larger H should imply the faster convergence, however, it may compromise the numerical stability of the method. On the other hand, high values of N should entail more accurate results and, consequently, a high degree of robustness, nevertheless, a higher computational burden is obtained as counterpart as we discussed before.

Intuitively, one can guess that a large value of $\boldsymbol{\phi}$ may affect the numerical stability of the method. In fact, the step size H is introduced in (8) for counteracting this effect. This idea suggests the following rule for updating the step size:

$$H = \max\left(0.1, \min\left(\left\|\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right)\right\|_{\infty}^{-1}, 1\right)\right) \tag{12}$$

By rule (12), we aim to tune H inversely proportional to $\boldsymbol{\phi}$. In this case, we have empirically bounded H to the range $[0.1, 1]$, in order to avoid too short or too large values of this parameter.

In the case of N, let us consider that this value is initialized in a sufficiently high value. Then, it can be updated by the following simple rule:

$$\text{if } \left\|\mathbf{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_{\infty} < \left\|\mathbf{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_{\infty} \tag{13a}$$

$$\text{then } N \leftarrow \max(N-1, 1) \tag{13b}$$

With the aim of reducing the overall computational cost as much as possible, N is progressively reduced as the algorithm properly progresses. In this case, $\|\mathbf{g}\|_{\infty}$ has been taken as index for determining if the algorithm is successfully converged.

Finally, taking into account the rules (12) and (13), the proposed algorithm for solving the PF equations using the developed PF solver is summarized in Algorithm 1 using

pseudocode, and Fig. 4 in form of flowchart. In this case, we consider that the procedure has failed if the iteration counter surpasses a predetermined threshold namely $k^{\max}$.

Algorithm 1: Developed PF solver based on the Romberg's Integration Scheme

Let $\boldsymbol{x}^{(0)}$, ε, $k^{\max}$, N, r and ψ be given
Initialize iteration counter $k = 0$
while $\left\|\mathbf{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_{\infty} \geq \varepsilon$ **do**
    $H \leftarrow$Solve (12) *# Update the step size*
    Solve (8) *# Calculate b*
    $j = 1$
    for $j = 1{:}\,N$ **do** *# Romberg's loop*
        Solve (9)
    end do
    Solve (11) *# Richardson's Extrapolation*
    if $\left\|\mathbf{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_{\infty} \geq \varepsilon$ **then**
      if $k = k^{\max}$ **then**
        break *# Fail*
      end if
      if $\left\|\mathbf{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_{\infty} < \left\|\mathbf{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_{\infty}$ **then**
          $N \leftarrow$ Solve (13)
      end if
    end if
    $k \leftarrow k + 1$
end do
return solution $\boldsymbol{x}^{(k)}$

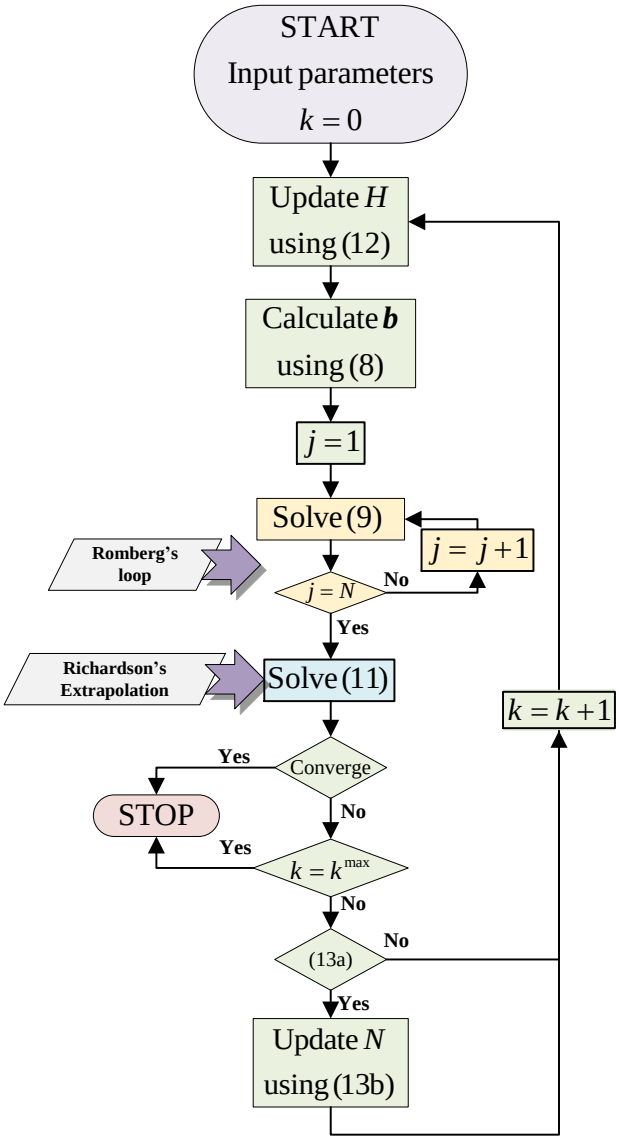

Fig. 4 Flowchart of the developed PF solver

3.3.- Computational burden

The factorization of the Jacobian matrix typically supposes the heaviest computational part of any PF calculation [20]. Iterative procedure described in Algorithm 1 only requires a factorization each iteration, while remainder calculations are mainly vector sums, function evaluations and linear systems solutions, which can be, generally, efficiently addressed.

3.4.- Main differences with the solvers based on the Continuous Newton's method

Since the developed PF solver is inspired by the Romberg's scheme, one may obtain the misleading conclusion that the proposed algorithm is in fact based on the Continuous Newton's method. However, several differences with respect to this kind of solvers may be observed.

- Those methodologies based on the Continuous Newton's method, are mainly devoted on integrating the following function (see [20, 33-35]):

  $$\mathbf{f}(\boldsymbol{u}) = -[\mathbf{g}'(\boldsymbol{u})]^{-1}\mathbf{g}(\boldsymbol{u}) \quad (14)$$

  While the developed technique is mainly devoted on integrating (7).

- A direct consequence of the preceding point is the higher degree of efficiency showed by the developed solver with respect the techniques considered in [20, 33-35]. Thus, while the proposed method only requires a factorization each iteration, those techniques based on the Continuous Newton's method require as many factorizations as function (14) evaluations.

- The techniques based on the Continuous Newton's method are conceived by mimicking the original structure of the numerical integration method under study. Oppositely, our method has been developed by modifying the standard Romberg's procedure. Therefore, the proposed PF solver should be considered inspired by the Romberg's integration scheme rather than established by directly adapting a numerical integration routine.

On the basis of the exposed differences, the developed PF solution method should be considered original rather than based on the Continuous Newton's method.

3.5.- Fixed Points of the proposed solver

Since the developed method based on the Romberg's scheme is devoted on solving the PF equations (1), it is expected from the PF solution (namely $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$) to

be a fixed point of the developed algorithm. The following Theorem demonstrates that $\boldsymbol{x}^*$ is a fixed point of the developed iterative procedure and, consequently, the proposed solver constitutes a PF solution method.

Theorem 1. *Fixed points of the developed algorithm:* let $\boldsymbol{x}^*$ be a solution of (1) such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$. Then, $\boldsymbol{x}^*$ is a fixed point of the iterative solution procedure described by the Algorithm 1.

Proof. Evaluating (8) at $\boldsymbol{x}^*$ one obtains:

$$\boldsymbol{b}|_{\boldsymbol{x}^*} = \boldsymbol{x}^* + H\boldsymbol{\phi}(\boldsymbol{x}^*, \boldsymbol{x}^*) = \boldsymbol{x}^* \tag{15}$$

While for (9) we have:

$$\begin{aligned} &\boldsymbol{c}_1|_{\boldsymbol{x}^*} = \boldsymbol{x}^* + H\boldsymbol{\phi}(\boldsymbol{x}^*, \boldsymbol{x}^*) = \boldsymbol{x}^* \\ &\boldsymbol{c}_2|_{\boldsymbol{x}^*} = \boldsymbol{x}^* + {}^{H}/_{2}\,[\boldsymbol{\phi}(\boldsymbol{x}^*, \boldsymbol{x}^*) + \boldsymbol{\phi}(\boldsymbol{x}^*, \boldsymbol{x}^*)] = \boldsymbol{x}^* \\ &\vdots \\ &\boldsymbol{c}_N|_{\boldsymbol{x}^*} = \boldsymbol{x}^* + \textstyle\sum_{j=0}^{N-1} \boldsymbol{\phi}(\boldsymbol{x}^*, \boldsymbol{x}^*) = \boldsymbol{x}^* \end{aligned} \tag{16}$$

Finally, evaluating (11) at $\boldsymbol{x}^*$:

$$\boldsymbol{x}^{(k+1)}\big|_{\boldsymbol{x}^*} = \frac{1}{r^{\psi}-1}\left(r^{\psi}\boldsymbol{x}^* - \boldsymbol{x}^*\right) = \boldsymbol{x}^* \tag{17}$$

Which completes the proof. □

3.6.- Stability Analysis

Firstly, let us introduce the following definitions.

Definition 2. *Hyperbolic point:* let $\mathbf{G} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ be the dynamic system associated with an iterative procedure. Then, $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$ is an equilibrium point of this system. In addition, $\boldsymbol{x}^*$ is called hyperbolic point if and only if all the eigenvalues associated with the Jacobian of $\mathbf{G}$ at the equilibrium point have a nonzero real part.

Definition 3. *Hyperbolic point asymptotically stable:* let $\mathbf{G} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ be the dynamic system associated with an iterative procedure. If $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$ is a

hyperbolic point of this system, it is said to be asymptotically stable (or sink), if all the eigenvalues associated with the Jacobian of $\mathbf{G}$ at $\boldsymbol{x}^*$ have a negative real part.

Definition 4. *Type-m equilibrium point:* let $\boldsymbol{x}^*$ be hyperbolic equilibrium point of the system $\mathbf{G} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$ associated with an iterative procedure. Then, $\boldsymbol{x}^*$ is said to be type-m if the Jacobian associated with $\mathbf{G}$ at $\boldsymbol{x}^*$ has exactly m eigenvalues with a positive real part.

From the Definitions 2-4 one expects from the PF solution $\boldsymbol{x}^*$ to be a type-0 equilibrium point for the iterative procedure described by the Algorithm 1. The following Theorem demonstrates that $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$ is a type-0 solution of the proposed solver.

Theorem 2. *Stability of the developed method:* let $\boldsymbol{x}^*$ be a PF solution such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$, then, it is a type-0 equilibrium point for the iterative procedure described in the Algorithm 1 if the following:

$$H\left[\frac{r^{\psi}\left(2^{N-1}-1\right)}{2^{N-1}\left(r^{\psi}-1\right)}+1\right] > 0 \tag{18}$$

holds.

Proof. Let us rewrite (11) as follows:

$$\boldsymbol{x}^{(k+1)} = \frac{1}{r^{\psi}-1}\left[r^{\psi}\left(\boldsymbol{x}^{(k)} + H\sum_{j=0}^{N-1}\frac{1}{2^{j-1}}\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_j\right)\right) - \boldsymbol{x}^{(k)} - H\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right)\right] \tag{19}$$

Reordering (18):

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{H}{r^{\psi}-1}\left[-\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right) + r^{\psi}\sum_{j=0}^{N-1}\frac{1}{2^{j-1}}\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_j\right)\right] \tag{20}$$

Taking into account that $\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_0\right) = \boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right)$:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{H}{r^{\psi}-1}\left[\left(r^{\psi}-1\right)\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right) + r^{\psi}\sum_{j=1}^{N-1}\frac{1}{2^{j-1}}\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_j\right)\right] \tag{21}$$

At this point, the dynamic system associated with the developed solver may be written as follows:

$$\mathbf{G} = H\left[\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{x}^{(k)}\right) + \frac{r^{\psi}}{r^{\psi}-1}\sum_{j=1}^{N-1}\frac{1}{2^{j-1}}\boldsymbol{\phi}\left(\boldsymbol{x}^{(k)}, \boldsymbol{c}_j\right)\right] \tag{22}$$

Evaluating the Jacobian associated with $\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{x})$ and $\boldsymbol{\phi}\left(\boldsymbol{x}, \boldsymbol{c}_j\right)$ at equilibrium point, one obtains:

$$\nabla_{\boldsymbol{x}}\boldsymbol{\phi}(\boldsymbol{x}, \boldsymbol{x})|_{\boldsymbol{x}^*} = -\nabla_{\boldsymbol{x}}[\mathbf{g}'(\boldsymbol{x}^*)]^{-1}\mathbf{g}(\boldsymbol{x}^*) - [\mathbf{g}'(\boldsymbol{x}^*)]^{-1}\mathbf{g}'(\boldsymbol{x}^*) = -\boldsymbol{I} \tag{23}$$

$$\nabla_{\boldsymbol{x}}\boldsymbol{\phi}\left(\boldsymbol{x}, \boldsymbol{c}_j\right)\Big|_{\boldsymbol{x}^*} = -\nabla_{\boldsymbol{x}}[\mathbf{g}'(\boldsymbol{x}^*)]^{-1}\mathbf{g}\left(\boldsymbol{c}_j\big|_{\boldsymbol{x}^*}\right) - [\mathbf{g}'(\boldsymbol{x}^*)]^{-1}\mathbf{g}'\left(\boldsymbol{c}_j\big|_{\boldsymbol{x}^*}\right) = -\boldsymbol{I} \tag{24}$$

where $\boldsymbol{I} \in \mathbb{R}^{n\times n}$ is the identity matrix. Therefore, we finally obtain:

$$\nabla_{\boldsymbol{x}}\mathbf{G}|_{\boldsymbol{x}^*} = -H\left[\boldsymbol{I} + \frac{r^{\psi}\left(2^{N-1}-1\right)}{2^{N-1}\left(r^{\psi}-1\right)}\boldsymbol{I}\right] \tag{25}$$

Which completes the proof. □

4.- An illustrative example

In order to fully describe the developed PF solver, let us consider an illustrative 2-bus example which is showed in Fig. 5. This case will be solved using the procedure described in Algorithm 1, taking $\varepsilon = 10^{-3}$, $r = 2$ and $\psi = 8$. On the other hand, the parameter N is initialized equal to 2.

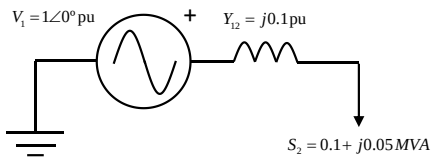

Fig. 5 An illustrative 2-bus example

The iterative procedure is initialized from a flat start, therefore, one has:

$$\boldsymbol{x}^{(0)} = [0, 1]^T \tag{26}$$

$$\mathbf{g}\left(\boldsymbol{x}^{(0)}\right) = [0.1, 0.05]^T \tag{27}$$

The very first step consists of calculating the increment vector $\boldsymbol{\phi}$ and initializing H using (7) and (8), respectively. From this step, one obtains:

$$\boldsymbol{\phi}\left(\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(0)}\right) = [-0.01, -0.005]^T \tag{28}$$

$$H = 1 \tag{29}$$

Now, the vector $\boldsymbol{b}$ can be calculated using (8), thus, the following value is obtained:

$$\boldsymbol{b} = [-0.01, 0.995]^T \tag{30}$$

Next, the algorithm evolves within the so-called Romberg's loop, for recursively refining the vectors $\boldsymbol{c}_j$ using (9). In this case, the following values are obtained.

$$\boldsymbol{c}_0 = \boldsymbol{x}^{(0)} = [0, 1]^T \tag{31}$$

$$\boldsymbol{c}_1 = \boldsymbol{x}^{(0)} = [-0.015, 0.9925]^T \tag{32}$$

$$\boldsymbol{c}_2 = \boldsymbol{x}^{(0)} = [-0.0163, 0.9918]^T \tag{33}$$

Finally, the state vector is updated using the Richardson's Extrapolation (11). From this step, one obtains:

$$\boldsymbol{x}^{(1)} = [-0.015, 0.9925]^T \tag{34}$$

Now, the convergence criterion is checked as follows:

$$\mathbf{g}\left(\boldsymbol{x}^{(1)}\right) = [-0.0493, -0.0238]^T \tag{35}$$

$$\left\|\mathbf{g}\left(\boldsymbol{x}^{(1)}\right)\right\|_{\infty} = 0.0493 \tag{36}$$

As (36) is greater than ε, more iterations have to be computed. Finally, the developed method achieves a solution within the required convergence tolerance at second iteration:

$$\boldsymbol{x}^{(2)} = [-0.010, 0.9949]^T \tag{37}$$

5.- Numerical Experiments

Several numerical experiments serve to show the performance of the developed method. The following PF solvers are used hereinafter for comparison purposes:

- Standard NR in polar coordinates.
- The Optimal Multiplier-based PF solver in polar coordinates such it was described in [23] (OMP).

- The High-Order Levenberg-Marquadt method (HOLM) [29]. In this case, in order to obtain a good performance from this technique, the so-called damping factor has been updated according to $\|\mathbf{g}\|_{\infty}^{1.3}$
- The Reverse-Bulirsch-Stoer solver (RBS) [34]. The parameters involved within this iterative procedure has been adjusted according to the guidelines provided in [34].
- The Richardson-based PF solution technique [40]. For this technique, $r = 2$ and $\psi = 8$ in order to get an optimal performance. In addition, the step size has been updated according to rule (12), with the aim of being straightforward compared with the proposed method.
- The developed Romberg-based methodology using the iterative procedure described in the Algorithm 1. In order to be properly compared with the other solvers, $r = 2$ and $\psi = 8$ have been considered. In addition, the parameter N has been initialized equal to 4. One can check that the condition (18) holds with these values.

All considered techniques have been coded in Matpower [41]. Moreover, $\varepsilon = 10^{-5}$ and $k^{\max} = 2000$ have been taken. We consider this value of $k^{\max}$ suitable for only discriminating the divergence cases. Finally, all simulations have been initialized using a flat start.

In order to validate the solutions reported, it has been compared with the correct one obtained with NR using the default starting point provided within Matpower. Thus, a solution is considered inaccurate if it differs from the correct one greater than 0.01. In this regard, instead of considering the low voltage solution an inaccurate value, these cases have been properly differentiated.

5.1.- Studied Systems

The following large scale ill-conditioned systems have been considered:

- The 3012-, and 3374-bus snapshots of the Polish Transmission system at winter 2007-08 evening peak [42] (*case3012wp* and *case3375wp*, respectively).
- The 13659-bus portion of the European Transmission system from the EU Pegase project [42-44] (*case13659pegase*).

Table 1 summarizes the main characteristics of the tested cases.

Table 1 Main characteristics of the studied systems

		case3012wp	*case3375wp*	*case13659pegase*
Buses		3012	3374	13659
Branches		3572	4161	20467
Generators		502	596	4092
Load	MW	27169	48363	381431
	MVar	10200	19527	98523
n		5725	6355	23225

5.2.- Convergence rates for base cases

The total number of iterations required by a PF solver is typically used for comparing different techniques. Table 2 reports the iterations required by the tested methods for base cases.

Table 2 Total Iterations for Base Cases

Method	*case3012wp*	*case3375wp*	*case13659pegase*
NR	Fail	Fail	Fail
OMP	367	350	48†
HOLM	14	13	32
RBS	13	13	14
Richardson	30	31	38
Romberg	6	6	9

†: Low Voltage Solution

As expected, NR failed in all studied systems. OMP converged to the high voltage solution in the *case3012wp* and *case3375wp*, while it found the low voltage value in the *case13659pegase*. Remainder techniques converged to the high voltage solution in all considered cases.

OMP and Richardson typically required many iterations for converging. HOLM employed a reasonable number of iterations in the *case3012wp* and *case3375wp*, however it needed 32 iterations in the largest case. RBS required 13-14 iterations while the developed PF solver was the most competitive technique employing just 6-9 iterations for obtaining the high voltage solution in all studied cases.

Fig. 6 shows the voltage magnitudes calculated by the developed PF solution method. In this figure, the maximum error with respect to the correct solution of each system is also shown. As observed, the results obtained by the developed method are very accurate.

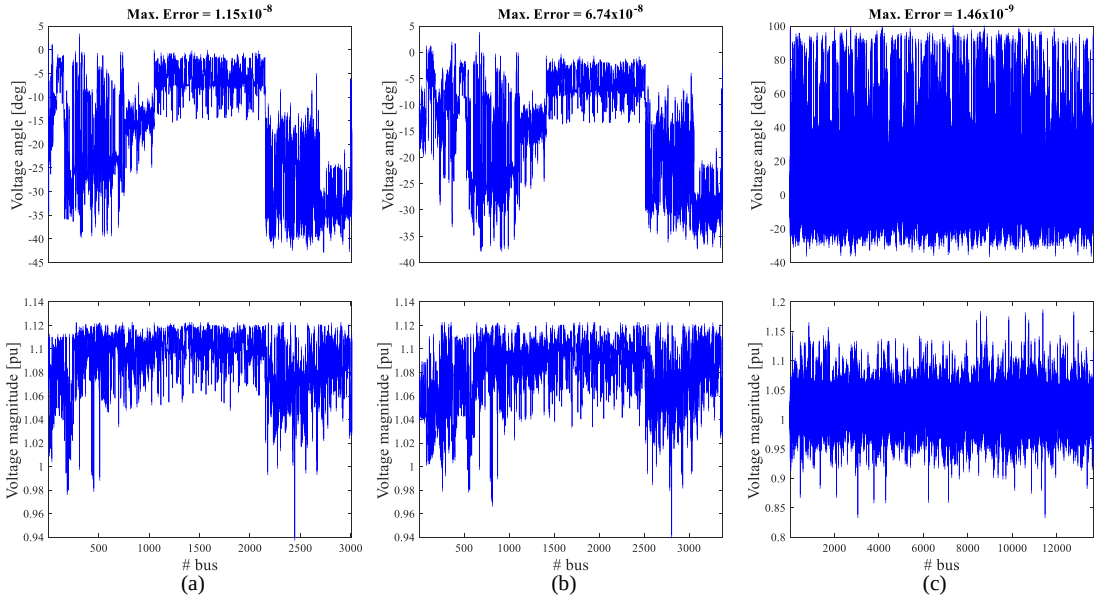

Fig. 6 Voltages calculated by the developed method and the maximum error with respect to the correct solution for the *case3012wp* (a), *case3375wp* (b) and *case13659pegase* (c)

Fig. 7 plots the convergence profiles for base cases. As can be seen, the proposed solver does not perform acceptably at first iterations, when other solvers showed lower residual. This corresponds with short values of H. However, when the step size turns large, the PF solver based on the Romberg's scheme quickly converges. It is worth noting that in the studied cases NR normally failed due to it exhibits an oscillatory pattern.

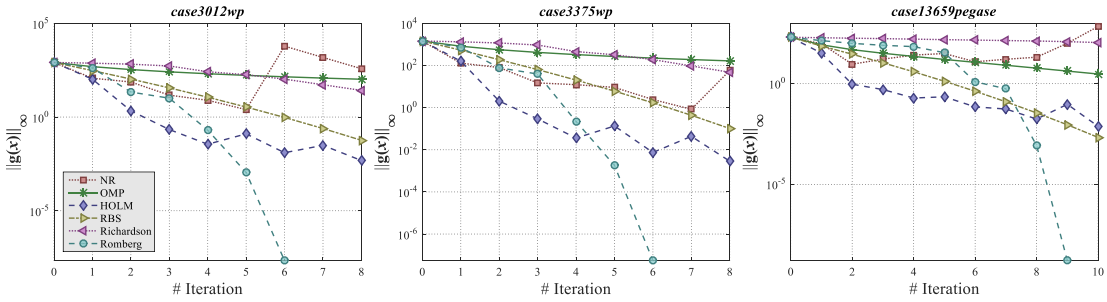

Fig. 7 Convergence profiles for base cases

5.3.- Convergence rates with reactive limits

Now, let us consider the scenario which the generators' reactive limits are enabled. To consider this point, the strategy described in [40] has been incorporated within the considered PF methods. Table 3 provides the total iterations required in this scenario. More iterations compared with the base cases are typically required due to several PF problems have to be solved for obtaining a feasible solution. Nevertheless, similar conclusions as for the base cases can be also drawn in this scenario.

Table 3 Total Iterations with Reactive Limits

Method	*case3012wp*	*case3375wp*	*case13659pegase*
NR	Fail	Fail	Fail
OMP	777	765	85†
HOLM	20	20	35
RBS	29	32	24
Richardson	56	61	55
Romberg	14	18	14

†: Low Voltage Solution

5.4.- Convergence rates for limit cases

Now, we shall to study how the loading level affects the overall performance of the tested methods. To do that, the active and reactive powers injected at PQ buses along the reactive powers injected at PV buses have been increased by multiplying them by a real factor namely $\lambda \in \mathbb{R}^{+}$. Fig. 8 plots the total iterations for different loading levels. As observed, the developed method was more competitive than the other techniques.

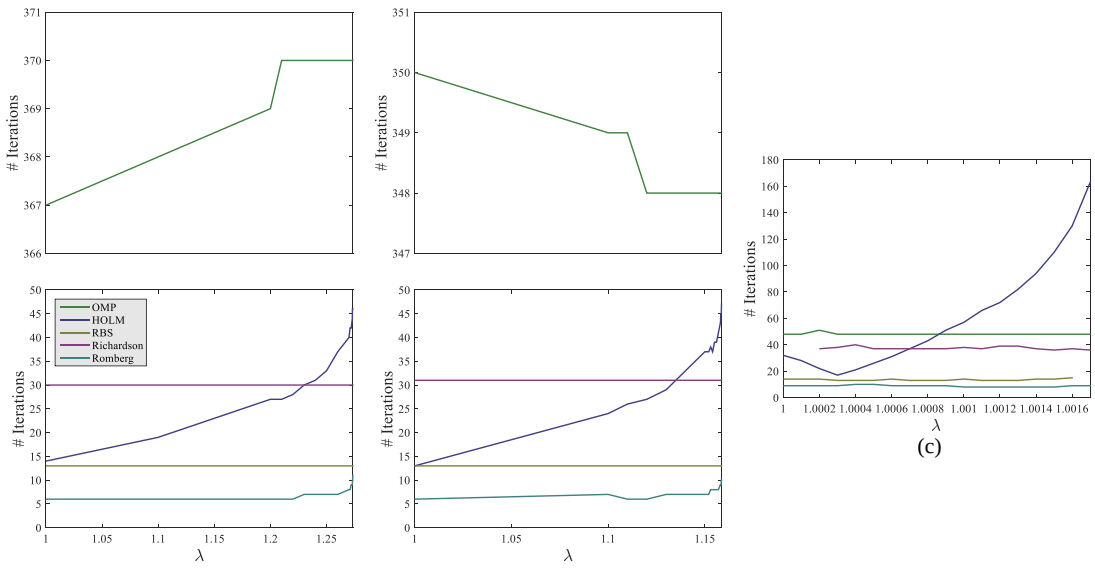

Fig. 8 Total iterations for different loading levels for the *case3012wp* (a), *case3375wp* (b) and *13659pegase* (c)

Next, the considered PF methods have been tested in a more stressful scenario which the loading level of the studied systems has been increased until close to the MLP. Thus, loading levels of the studied systems have been progressively increased in steps up to the fourth decimal until all considered methods failed for converging. Them, the immediately lower value of λ is taken. As example, for the *case3012wp* we have taken $\lambda = 1.2734$ since $\lambda = 1.2735$ gives rise divergence. Table 4 reports the total iterations for this scenario.

Table 4 Total Iterations for Limit Cases

Method	*case3012wp* $\lambda = 1.2734$	*case3375wp* $\lambda = 1.1586$	*case13659pegase* $\lambda = 1.0017$
NR	Fail	Fail	Fail
OMP	370	348	48
HOLM	46	47	163‡
RBS	13	13	Fail
Richardson	30	31	36
Romberg	11	10	9

‡: Inaccurate Solution

In this case, RBS and HOLM successfully solved the *case3012wp* and *case3375wp*, however, the former failed in the *case13659pegase* while the latter obtained an inaccurate solution in this system. Remainder methods converged to the high voltage solution in all

studied cases. Despite that the developed method normally required more iterations compared with the base scenario, it is still the most competitive approach.

5.5.- Solution Times

Now, let us put our attention on the efficiency. Tables 5-7 provide the solution times in seconds consumed by the considered PF methods in the analyzed scenarios (i.e. Tables 2-4 respectively). These results have been obtained on a 64-bit i5-9400F Intel Core personal computer (2.90 GHz, 8 GB of RAM). In order to avoid the influence of other computational activities, the reported results have been calculated as the average values of 100 simulations. As observed, the developed method was the most efficient one, notably outperforming the remainder techniques in all studied cases.

Table 5 Solution Times [s] for Base Cases

Method	*case3012wp*	*case3375wp*	*case13659pegase*
OMP	6.82	7.14	4.08†
HOLM	0.48	0.48	6.55
RBS	0.75	0.79	3.85
Richardson	0.52	0.59	3.02
Romberg	0.12	0.14	0.78

†: Low Voltage Solution

Table 6 Solution Times [s] with Reactive Limits

Method	*case3012wp*	*case3375wp*	*case13659pegase*
OMP	14.61	16.08	7.19†
HOLM	0.70	0.81	7.18
RBS	1.78	2.27	6.75
Richardson	0.99	1.18	4.37
Romberg	0.28	0.41	1.23

†: Low Voltage Solution

Table 7 Solution Times [s] for Limit Cases

Method	*case3012wp* $\lambda = 1.2734$	*case3375wp* $\lambda = 1.1586$	*case13659pegase* $\lambda = 1.0017$
OMP	6.92	6.90	4.08
HOLM	1.57	1.82	33.43‡
RBS	0.75	0.79	--
Richardson	0.52	0.59	2.87
Romberg	0.19	0.21	0.78

‡: Inaccurate Solution

6.- Conclusions and Future Works

This paper has developed a robust and efficient PF solver suitable for large scale ill-conditioned systems. It is essentially based on the Romberg's Integration Scheme. The developed method is asymptotically stable and presents a computational cost comparable with NR.

Several numerical results have been analysed in order to discern the suitability of the developed PF solution method. Other conventional and robust techniques have been considered for comparison. The proposed methodology has turned out to be robust and efficient enough to properly manage all the studied large scale ill-conditioned cases, notably outperforming the remainder tested techniques.

Due to the promising results obtained, the developed PF solution technique should be tested in other related tools like Security Analysis [45] or the Continuation Power Flow [28].

References

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

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

[3] Ward JB, Hale HW. Digital computer solution of power flow-problems. Trans Amer Inst Elect Eng Part III, Power Appar Syst 1956; 75(3): 398-404.

[4] Tinney WF, Hart CE. Power Flow Solution by Newton's Method. IEEE Trans Power Appar Syst 1967: PAS-86(11): 1449-60.

[5] Stott B, Alsac O. Fast decoupled load flow. IEEE Trans Power Appar Syst 1974; PAS-93: 859-69.

[6] van Amerongen RAM. A general-purpose version of the fast decoupled load flow. IEEE Trans Power Syst 1989; 4(2): 760-70.

[7] Nagendra Rao PS, Prakasa Rao KS, Nanda J. An Exact Fast Load Flow Method Including Second Order Terms in Rectangular Coordinates. IEEE Trans Power Appar Syst 1982; PAS-101(9): 3261-8.

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

[9] Fatemi SM, Abedi S, Gharehpetian GB, Hosseinian SH, Abedi M. Introducing a Novel DC Power Flow Method With Reactive Power Considerations. IEEE Trans Power Syst 2015; 30(6): 3012-23.

[10] 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.

[11] Derakhshandeh SY, Pourbagher R. Application of high-order Newton-like methods to solve power flow equations. IET Gener Transmiss Distrib 2016; 10(8): 1853-59.

[12] 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.

[13] 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.

[14] da Costa VM, Pereira JLR, Martins N. An augmented Newton-Raphson power flow formulation based on current injections. Int J Elect Power Energy Syst 2001; 23(4): 305-12.

[15] Garcia PAN, Pereira JLR, Carneiro S, Vinagre MP, Gomes FV. Improvements in the representation of PV buses on three-phase distribution power flow. IEEE Trans Power Delivery 2004; 19(2): 894-6.

[16] Kamel S, Abdel-Akher M, Jurado F. Improved NR current injection load flow using power mismatch representation of PV bus. Int J Elect Power Energy Syst 2013; 53: 64-8.

[17] Saleh SA. The Formulation of a Power Flow Using *d-q* Reference Frame Components-Part I: Balanced 3φ Systems. IEEE Trans Ind Appl 2016; 52(5): 3682-93.

[18] Saleh SA. The Formulation of a Power Flow Using *d-q* Reference Frame Components-Part II: Unbalanced 3φ Systems. IEEE Trans Ind Appl 2018; 54(2): 1092-107.

[19] 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: 383-92.

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

[21] Sasson AM, Treviño C, Aboytes F. Improved Newton's Load Flow Through a Minimization Technique. IEEE Trans Power Appar Syst 1971; PAS-90(5): 1974-81.

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

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

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

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

[26] Overbye TJ, Klump RP. Effective calculation of power system low-voltage solutions. IEEE Trans Power Syst 1996; 11(1): 75-82.

[27] Klump RP, Overbye TJ. Techniques for Improving Power Flow Convergence. 2000 Power Engineering Society Summer Meeting (Cat. No.00CH37134) Seattle WA; 2000. pp. 598-603 vol. 1.

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

[29] Pourbagher R, Derakhshandeh SY. Application of high-order Levenberg–Marquardt method for solving the power flow problem in the ill-conditioned systems. IET Gener Transmiss Distrib 2016; 10(12): 3017-22.

[30] Tostado M, Kamel S, Jurado F. An Effective Load-Flow Approach Based on Gauss-Newton Formulation. Int J Elect Power Energy Syst 2019; 113: 573-81.

[31] 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.

[32] Neuberger JW. Continuous Newton's Method. In: Neuberger JW, editor. Sobolev Gradient and Differential Equations, 2nd ed, vol. 1670. Berlin: Springer-Verlag; 2010, p. 79-83.

[33] 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 2019; 29(4): e2784. https://doi.org/10.1002/etep.2784

[34] Tostado-Véliz M, Kamel S, Jurado F. A Robust Power Flow Algorithm Based on Bulirsch-Stoer Method. IEEE Trans Power Syst 2019; 34(4): 3081-89.

[35] 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: 105881. https://doi.org/10.1016/j.epsr.2019.105881

[36] Xie N, Bompard E, Napoli R, Torelli F. Widely convergent method for finding solutions of simultaneous nonlinear equations. Elect Power Syst Res 2012; 83(1): 9-18.

[37] Krylov VI. Approximate Calculation of Integrals. New York: MacMillan; 1962.

[38] Lynch P. Richardson extrapolation: the power of the 2-gon. Math Today 2003; 39(5): 159-60.

[39] Liu Z, Zheng Q, Huang C-E. Third- and fifth-order Newton-Gauss methods for solving nonlinear equations with n variables. Appl Math Comput 2016; 290: 250-7.

[40] Tostado-Véliz M, Kamel S, Jurado F. Robust and efficient approach based on Richardson extrapolation for solving badly initialised/ill-conditioned power-flow problems. IET Gener Transmiss Distrib 2019; 13(16): 3524-33.

[41] 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.

[42] Matpower benchmark systems. https://matpower.org/download/ [accessed 5 March 2020]

[43] Josz C, Fliscounakis S, Maeght J, Panciatici P. AC Power Flow Data in Matpower and QCQP Format: iTesla, RTE Snapshots, and PEGASE. Available: http://arxiv.org/abs/1603.01533 [v3 accessed 5 March 2020]

[44] 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.

[45] Shen Z, Chiang HD, Tang Y, Zhou N. An online line switching methodology with look-ahead capability to alleviate power system overloads based on a three-stage strategy. Int J Elect Power Energy Syst 2020; 115: 105500. https://doi.org/10.1016/j.ijepes.2019.105500