Received: 15 July 2021 | Revised: 12 September 2021 | Accepted: 20 October 2021

DOI: 10.1002/2050-7038.13194

RESEARCH ARTICLE

WILEY

Solving realistic large-scale ill-conditioned power flow cases based on combination of numerical solvers

Marcos Tostado-Véliz[1] | **Hany M. Hasanien**[2] | **Rania A. Turky**[3] | **Salah Kamel**[4] | **Francisco Jurado**[1]

[1]Department of Electrical Engineering, University of Jaén, Jaén, Spain

[2]Electrical Power and Machines Department, Faculty of Engineering, Ain Shams University, Cairo, Egypt

[3]Electrical Engineering Department, Faculty of Engineering and Technology, Future University in Egypt, Cairo, Egypt

[4]Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan, Egypt

Correspondence
Francisco Jurado, Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén, Spain.
Email: fjurado@ujaen.es

Abstract

With the increasing electricity consumption and difficulty in upgrading existing infrastructures, ill-conditioned power flow (PF) cases are becoming more frequent nowadays. In this context, classical robust solvers may be unsuitable for realistic networks, which typically encompass thousands of buses, because of their high computational burden or low convergence rate. This article tackles this issue by proposing a novel PF solver, which presents acceptable robustness and efficiency in solving large-scale ill-conditioned systems. The proposed algorithm collects the advantage of various numerical solvers, which by separate present different weaknesses, but actuating in coordination their strengths can be jointly exploited. More precisely, the robust Forward-Euler and Trapezoidal rules are combined with the efficient Darvishi cubic technique. Thereby, an original predictor-corrector algorithm is developed to effectively coordinate the different numerical algorithms involved, obtaining a robust but efficient yet solution procedure. Various large-scale ill-conditioned benchmark systems are studied under different stressing conditions. The results obtained with the developed technique are promising, outperforming other robust and standard PF solvers.

KEYWORDS

computational efficiency, ill-conditioned cases, large-scale systems, power flow

List of Symbols and Abbreviations: t, time instant; $(.)^{sch}$, scheduled value; i, denotes the ith bus of the system; k, Denotes the kth iteration of an iterative algorithm; n_l, total number of PQ buses; n_g, total number of PV buses; n, size of the power flow state vector ($n = 2n_l + n_g$ in polar coordinates); N, number of iterative loops; ω, state vector of a dynamic system; Δt, time step; P, nodal active power injection; Q, nodal reactive power injection; $V\angle\delta$, nodal voltage; $Y\angle\theta$, element of the admittance matrix; h, step size; ε, convergence tolerance; k^{max}, maximum number of iterations allowed; g, power flow equations; $[\cdot]^T$, transpose operator; $[\cdot]^{-1}$, inverse operator; $\|\cdot\|_\infty$, infinity norm; ∇_x, vector/matrix gradient (first derivative) with respect to x; $\max\{a,b\}$, returns b if $b > a$ and a otherwise; $\min\{a,b\}$, returns b if $b < a$ and a otherwise; x, y, z, power flow state vector; δ_{PV}, voltage angles at PV buses; δ_{PQ}, voltage angles at PQ buses; V_{PQ}, voltage magnitudes at PQ buses; g_x, Jacobian matrix formed by the first partial derivatives of g with respect x; I, identity matrix; BEM, Backward-Euler method; HKW, Heun-King Werner approach; HOLM, high order Levenberg-Marquardt solver; LU, lower-upper; NR, Newton-Raphson technique; OMP, optimal multiplier method in polar coordinates; PF, power flow; TR, trapezoidal rule applied to PF analysis; 3OD, darvishi's cubic method applied to PF analysis.

1 | INTRODUCTION

1.1 | Context & motivation

In power system analysis, power flow (PF) is likely the most important computational tool,[1] finding multiple applications in planning and operation of power systems, along energy markets or security analysis, among others.[2,3] From a pure mathematical point of view, PF consists of solving a set of nonlinear Equations. A nonlinear system may be well- or ill-conditioned, depending on the condition of the Jacobian matrix.[4] Particularizing for the PF problems, Milano[5] gave a formal definition of ill-conditioned PF cases:

> **Definition 1.** Ill-conditioned PF cases: A PF problem is ill-conditioned if, although its solution exists, it is not reachable using traditional solvers and a flat start (ie, all voltage magnitudes equal to 1 p.u. in PQ buses and all voltage angles equal to 0 rad. in PV and PQ buses).

Traditionally, ill-conditioned issues with PF problems appear due to various aspects such as high loading conditions, the high R/X ratio (especially important in distribution networks), weakly meshed structure or connection, or very low impedance branches, among others.[6] Although Ill-conditioned PF problems were well-studied at the early 1980s, a resurgent interest in these kinds of cases have been observed at the beginning of the 21st century. This is due to two reasons:

On the one hand, ill-conditioned problems are becoming more frequent nowadays. This is mainly due to a growing demand along the overall difficulty in upgrading existing infrastructures.[4] These two facts led the power systems to be operated close to their operability limits. In such situation, PF problem frequently presents ill-conditioned issues.[6,7] This context puts on manifest the importance of properly dealing with ill-conditioned systems nowadays. In addition, it is expected this situation keeps in the near and far future. Numerical issues provoked by ill-conditioned cases have been traditionally addressed using numerical PF solvers. Conventionally, a PF solver can be defined as robust if it is able to solve ill-conditioned cases for which the traditional Newton-Raphson technique (NR) fails to converge.

On the other hand, existing robust PF solvers are typically very inefficient because they involve computationally cost operations such as multiple factorizations. This feature makes such kind of solvers not competitive with the traditional NR, which presents a lower computational burden. This characteristic hinders the applicability of robust PF techniques in online tools, which are gaining importance nowadays for security analysis or optimization routines, among others.[8] This way, the applicability of robust PF techniques in industry tools has been quite limited.[9] Therefore, one can consider that a PF solver is efficient if its computational burden is comparable or lower than that showed by NR. In this sense, operators may face important challenges due to the unsolvability of PF problems, especially in stressed systems.[10,11]

In the light of the reasons above, power system operators will likely need updating traditional computational tools to effectively address the challenges posed by ill-conditioned cases. In this regard, the development of novel robust PF solvers with acceptable efficiency has gained notable importance.

1.2 | Literature review

Engineering community became to be concerned about ill-conditioned PF cases at the early 1980s, when Stott reported converging problems in 11- and 43-bus test systems.[12] This issue was tackled by Iwamoto and Tamura,[13] who developed a minimization technique to improve the convergence properties of the traditional NR. This proposal consists of developing the second-order Taylor expansion of the PF equations and, on the basis of this extended formulation, posing a minimization problem by which the magnitude of Newton's descent vector is modified, thus ensuring the reduction of the residuals. This technique has been considered the main benchmark robust PF solver, and other authors have proposed different improvements based on it.[14] During this decade, other works contributed to this pool further exploiting Newton-like solvers[15] or using high-order truncations of the Taylor series of the PF equations.[16] However, these solvers were not as popular as Iwamoto's technique and have not been further developed and studied.

With the advent of the 21st century, the interest in the solution of PF ill-conditioned systems has reemerged. One clear milestone in this line was the application of the continuous Newton's method for PF analysis.[5] This technique establishes a common framework for the solution of nonlinear equations and analysis of dynamic systems. In this

regard, NR can be easily assimilated to the Forward-Euler technique. Assuming a dynamic system denoted by f, the Forward-Euler method proceeds as follows[5]:

$$\omega_{t+\Delta t} = \omega_t + \Delta t f(\omega_t), \tag{1}$$

where ω is the state vector, t stands for the time instant, and Δt is the incremental step size.

Thereby, it is concluded that any other numerical integration routine could be applied to PF analysis. This idea was successfully proved in Reference 5, where the fourth-order Runge-Kutta formula serves as background for developing a robust and efficient PF solver. Motivated by these results, the authors further extended the applicability of the continuous Newton's method to PF analysis, developing robust solvers based on Runge-Kutta techniques,[17] Adams-Bashforth's methods,[18] and Bulirsch-Stoer algorithm.[19] This family of solvers presents acceptable robustness and easy codification in standard software. Nonetheless, they are still few efficient. As a sake of example, the fourth-order Runge-Kutta[5] PF technique requires up to 4 lower-upper (LU) factorizations, whereas the conventional NR only requires one. More recently, Milano further has developed the formulation of the continuous Newton's method to implicit solvers.[20]

Other authors have addressed the solution of ill-conditioned systems by developing first-[4,21] and second-order[22] dynamic paradigms. These solvers are sustained in the Lyapunov theory, by which the solution of a system of nonlinear equations can be conceived as a dynamic system whose equilibrium points correspond with the physical meaningful solutions of the PF problem. Keeping this in mind, the PF problem can be solved by using numerical integration routines like *ode* solvers in MATLAB. This way, the algorithm is widely convergent compared with traditional iterative techniques like NR. Nevertheless, the computational cost of the solution routine exponentially grows with the system size, which makes the dynamic paradigm unsuitable to large-scale systems at least parallel routines, or very powerful machines were employed.[4]

Another category of solvers corresponds to the Levenberg-based techniques.[9,23,24] By these methods, the condition of the Jacobian matrix is enhanced by introducing an artificial parameter called damping factor and a so-called regularization matrix. The idea behind Levenberg methods is initializing the iterative procedure with a high value of the damping factor. Thereby, the condition of the Jacobian matrix is low since the elements of its diagonal have high values. As the iterative procedure evolves, the damping factor has to decrease in order to obtain the actual solution of the PF equations. This way, the treatment of the damping factor is critical in the Levenberg-based solvers. Despite that, a unified criterion to update this parameter has not been studied yet.

Other approaches take advantage of continuation techniques.[25] These kinds of methods track the whole P-V curve in each node of the system by sequentially solving a series of PF problems for different loading conditions. In this sense, these kinds of methods are clearly not computationally competitive with other traditional solvers like NR.[26] Recent advances in this field have been focused on exploiting homotopy[27,28] and holomorphic[29] techniques. In this regard, Tylavsky et al have profusely studied holomorphic algorithms for estimating the limit loading levels of power systems[30] and calculating all the type-1 (low voltage) PF solutions.[31] On the other hand, Chiang, et al have developed novel methodologies[32] and derived theoretical studies.[33] One of the main disadvantages of holomorphic-based methods is the possibility of calculating non-physical solutions, which lacks importance in industry tools.[7]

Finally, other robust methods could not be included in the categories above. Mokhlis et al[34] have recovered the idea of optimal multiplier methods, similar to the conventional Iwamoto's technique.[13] In this case, the authors exploit the quadratic properties of the PF equations to improve the calculation of the multiplier and accelerate the convergence. Some authors have considered the implicit Z-bus method for PF calculation in radial or weakly meshed networks[35]; however, this technique may encounter numerical convergence issues in networks with a large number of PV buses.[36] On the other hand, the authors have developed various robust and efficient solvers,[37,38] which arise from the combination of various numerical techniques. These kinds of methods have offered promising results in various large-scale benchmarks ill-conditioned cases.

1.3 | Contributions and paper organization

As exposed in Section 1.1, it is expected power system analysis will face various challenges in the near future related to PF calculation in ill-conditioned systems. The main related research gaps which motivate this work are listed below:

- Most of existing robust PF solvers are quite inefficient, at least compared with NR. This feature limits their applicability in real cases and industry tools.

- Some robust approaches are quite rigid in the sense of facilitating their codification in standard software. In addition, their code structure is not versatile enough to incorporate control strategies such as reactive limits or on-load tap changers.
- Most of the existing robust PF solvers present a very low convergence order; that is, they require many iterations to achieve the PF solution, as in the case of the optimal multipliers methods.[19]

Motivated by the good results obtained in References 37 and 38, this article aims at further studying the idea of combining different numerical approaches to develop robust and efficient PF solvers. In essence, this approach aims at jointly exploiting the advantages of the different methods involved, while their weaknesses are compensated by the strengths of the other techniques. This way, a holistic framework is obtained, which jointly exploits the salient features of the different solvers. In this case, the cubic Darvishi method[39] is combined with the Euler and Trapezoidal rules[17] in a predictor-corrector paradigm, by which the Euler method produces a robust predicted value of the PF state vector and the Trapezoidal rule acts in coordination with the cubic technique to accelerate the convergence. For the sake of clarity, the main contributions of this article are numerated below:

- Developing a novel robust and efficient PF solver, derived from the combination of various numerical approaches.
- Proposing a holistic framework for the combination of the different numerical techniques involved, with the aim of exploiting them in a coordinated way.
- Deriving a numerical stability analysis of the novel proposal.
- Developing a simple but reliable computation implementation of the developed PF solver, proposing adaptive mechanisms for the different parameters involved.
- Validating the new proposal by performing extensive numerical simulations in various benchmarks large-scale ill-conditioned systems, and comparing the results obtained with the developed technique with other state-of-art robust solvers.

In the rest of this article, Section 2 sets out the foundations of the developed solver from a qualitative point of view. Section 3 describes the necessary mathematical background. Section 4 presents the novel PF solver and explains its computational implementation. Various numerical experiments are presented in Section 5. The article is concluded with Section 6.

2 | FOUNDATIONS AND REMARKS

In this section, the solution adopted for PF calculation is justified based on qualitative foundations. As commented above, the developed PF solver is based on the combination of various numerical methods. In this sense, the new proposal aims at jointly exploiting the strengths of each technique, while the effect of their weaknesses is minimized. Thereby, the selection of the methods involved should not be performed randomly. Therefore, combining two robust but inefficient techniques lacks of sense, since the resulting solver will be robust but very inefficient. In this regard, the drawbacks of the original methods would not be overcome. Contrarily, if two efficient, but weak methods are combined, the result will be an efficient technique with bad performance in ill-conditioned cases. In this regard, the involved methods should be complementary. As a sake of example, the Heun-King-Werner method[38] clearly exploits this idea. This technique combines the robust properties of the Heun rule applied to PF analysis, with the high convergence rate of King-Werner's methods.

One of the most important weaknesses of some robust methods is their low convergence rate, which usually requires many iterations to converge. For instance, the solvers based on the continuous Newton's method typically present linear convergence,[17] while NR presents quadratic convergence rate. In addition, these kinds of methods have a high computational cost, because the large number of LU factorizations required each iteration. However, their acceptable robustness makes these approaches an interesting alternative to solve ill-conditioned systems. To accelerate its order of convergence, cubic Darvishi's technique[39] is considered. This method is quite efficient as only one Jacobian factorization is required each iteration, while it achieves a third order of convergence. This feature allows Darvishi's method to usually converge faster than NR employing less number of iterations and consuming less computation time.[39] However, this technique is quite weak.[40] To compensate for this drawback, Darvishi's method is combined with the Trapezoidal rule,[17] which provides robustness to the algorithm.[40]

The different approaches have not been therefore chosen arbitrarily. The idea is combining robust and efficient techniques to jointly exploit their advantages. In this sense, the characteristics of the selected techniques have been already proved in the literature. Thus, the numerical robustness of the Forward-Euler method and the Trapezoidal technique has been already proved in References 5 and 17, while the cubic convergence features along its computational efficiency were firstly theoretically derived in Reference 41 and posteriorly empirically proved in Reference 39. It is worth noting that other combinations may eventually lead to promising numerical solvers; however, the use of different numerical approaches in PF problems is continuously updating and more studies should be performed prior to consideration of another kind of methodologies like super-cubic order Newton-like methods.[42,43]

The three considered techniques are merged in a predictor-corrector scheme, by which the Forward-Euler method firstly calculates a robust approach of the state vector. Taking this value as initialization, the Darvishi-Trapezoidal loop is iteratively performed, progressively refining the value of the state vector, which should notably accelerate the convergence of the solution procedure. As a result, it is expected to obtain a robust but efficient yet methodology.

3 | MATHEMATICAL BACKGROUND

3.1 | PF solution using NR

In polar coordinates and power residuals, the PF problem is described by a set of n nonlinear equations, as follows:

$$\mathbf{g}(\boldsymbol{x}) = \begin{cases} g_{\mathrm{P}_i} = P_i^{\mathrm{sch}} - \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\cos\left(\theta_{ij} - \delta_i + \delta_j\right), \forall i \in \{\mathrm{PQ}, \mathrm{PV}\} \\ g_{Q_i} = Q_i^{\mathrm{sch}} - \sum_{j=1}^{n} |V_i||V_j||Y_{ij}|\sin\left(\theta_{ij} - \delta_i + \delta_j\right), \quad \forall i \in \{\mathrm{PQ}\} \end{cases} \tag{2}$$

$$\boldsymbol{x} = [\boldsymbol{\delta_{PV}}, \boldsymbol{\delta_{PQ}}, \boldsymbol{V_{PQ}}]^T, \tag{3}$$

where $P_i^{\mathrm{sch}} \in \mathbb{R}$ and $Q_i^{\mathrm{sch}} \in \mathbb{R}$ are the active and reactive power injected at ith bus, respectively. $V_i \angle \delta_i \in \mathbb{C}$ is the complex voltage at ith bus. $Y_{ij} \angle \theta_{ij} \in \mathbb{C}$ is the ijth element of the admittance matrix. In (2), PQ and PV denote the set of PQ and PV buses, respectively. In the PF state vector (3), $\boldsymbol{\delta}_{\mathbf{PV}} \in \mathbb{R}^{n_g}$ is the vector of voltage angles at PV buses, $\boldsymbol{\delta}_{\mathbf{PQ}} \in \mathbb{R}^{n_l}$ is the vector of voltage angles at PQ buses, and $\boldsymbol{V}_{\mathbf{PQ}} \in \mathbb{R}^{n_l}$ 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, whereas n refers to the size of the PF state vector ($n = 2n_l + n_g$).

The set (2) is clearly nonlinear because trigonometric functions and product of variables. In this sense, its solution cannot be directly obtained. Among all available nonlinear solvers, NR is by far the most widely employed in PF analysis. This solver is defined by the following iterative map:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\mathbf{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{x}_k), \tag{4}$$

where the subscript denotes the kth iteration of the procedure, $\mathbf{g}' \in \mathbb{R}^{n\times n}$ is the Jacobian matrix of (2), and $[\cdot]^{-1}: \mathbb{R}^{n\times n} \longmapsto \mathbb{R}^{n\times n}$ is the inverse operator. The map (4) has quadratic convergence and a low computational burden (only requires one LU decomposition each iteration). However, its convergence properties are local. It means that the reachability of the solution with quadratic convergence is only ensured when the algorithm is evolved close to the solution. This issue limits the applicability of NR in ill-conditioned cases.[5] It is worth noting the similitude of the mappings (1) and (4), which motivated the use of the continuous Newton's method for PF analysis.[5]

3.2 | PF solution using the Forward-Euler method

The Forward-Euler method as applied to PF analysis is a simple modification of the mapping (4) to be assimilated to the procedure (1). In this sense, the increment vector is modified by introducing an artificial step size $h \in \mathbb{R}^+$ and the function f is replaced by the PF equations, as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - h[\mathbf{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{x}_k) \tag{5}$$

As Milano pointed out,[5] the mapping (5) is a generalization of the optimal multiplier methods.[13] Following the framework (5), the different techniques differ in the way that h is calculated/updated. While the optimal multiplier techniques normally pose a minimization problem, the Forward-Euler solver updates the step size under heuristic criterion, like the simple updating rule proposed in Reference 5 or other mechanisms developed in Reference 17. The heuristic rules are normally devoted to properly update the step size as the algorithm successfully evolves. For example in Reference 17, the infinity norm of the residual is taken as indicator to decide if the step size can be enlarged or not. Frequently, these updating mechanisms include bounds to preserve the numerical stability (ie, not divergence) of the methodology. In this sense, the numerical stability of a PF solver is defined as the feature of the numerical mapping that allows its convergence. Thus, if a PF solver is stable, its convergence is not guaranteed, but, if contrary, the PF solver would be unable to successfully converge. Unstable solvers are normally characterized by an erratic pattern in which the residuals tend to exponentially grow each iteration.[40]

3.3 | PF solution using the Trapezoidal rule

Under the continuous Newton's paradigm,[5] any numerical integration technique can be applied to PF analysis. Keeping this in mind, the map (6) is the simple extension of (5) applying the Trapezoidal scheme.[17]

$$\begin{cases} \boldsymbol{y}_k = \boldsymbol{x}_k - h[\mathrm{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{x}_k) \\ \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \dfrac{h}{2}\left([\mathbf{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{x}_k) + [\mathbf{g}'(\boldsymbol{y}_k)]^{-1}\mathbf{g}(\boldsymbol{y}_k)\right) \end{cases}, \tag{6}$$

where $\boldsymbol{y} \in \mathbb{R}^n$ is the PF state vector, which simply differs from $\boldsymbol{x}$ in its value. The main disadvantage of (6) compared with (5) is the necessity of evaluating (and factorizing) two Jacobian matrixes at each iteration.

3.4 | PF solution using the Darvishi's cubic method

The Darvishi's method is an efficient high-order Newton-like technique, which achieves a third order of convergence[39,41,44] with just one Jacobian evaluation per iteration, as follows.

$$\begin{cases} \boldsymbol{y}_k = \boldsymbol{x}_k - [\mathbf{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{x}_k) \\ \boldsymbol{x}_{k+1} = \boldsymbol{y}_k - [\mathbf{g}'(\boldsymbol{x}_k)]^{-1}\mathbf{g}(\boldsymbol{y}_k) \end{cases} \tag{7}$$

Although the mapping (7) is less efficient than NR since it involves more computations, the whole iterative procedure is typically more competitive due to its higher convergence rate. Moreover, additional calculations are principally vectors sums, which can be efficiently addressed by conventional software.[5]

4 | THE DEVELOPED PF SOLVER

The developed solver combines the different techniques explained in the previous section. To this end, a predictor-corrector scheme is used. In this paradigm, the predicted value should be sufficient robust[45]; thereby, it should be calculated by a robust solver. The new proposal uses the Forward-Euler method to this end, as follows:

$$\boldsymbol{z}^1 = \boldsymbol{z}^0 - h\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\mathbf{g}\left(\boldsymbol{z}^0\right), \tag{8}$$

where $\boldsymbol{z}^0 = \boldsymbol{x}_k$. The second step (corrector) corresponds to a refinement of the value calculated at the predictor stage. In this case, assuming a robust value calculated by the predictive method, one can use an efficient high-order technique to accelerate

the convergence. The mapping (7) is efficient and presents cubic convergence; however, it is numerically weak.[40] To keep the algorithm sufficiently robust, but augmenting the convergence rate as much as possible, it is proposed to combine the Trapezoidal rule (6) with the high-order Scheme (7). In this sense, it is interesting to note that the Darvishi's method achieves high-order convergence by evaluating the function in various points. The convergence rate of the mapping (7) can be further augmented by adding additional steps in which the Jacobian matrix is fixed and only the function has to be evaluated in different points.[49,Eq. (2)] These ideas lead to define the second step of the developed solution algorithm as an iterative loop as follows:

$$\boldsymbol{z}^{j+1} = \boldsymbol{z}^0 - \frac{h\left[\mathbf{g}'(\boldsymbol{z}^0)\right]^{-1}}{2}\left(\mathbf{g}\left(\boldsymbol{z}^{j-1}\right) + \mathbf{g}\left(\boldsymbol{z}^j\right)\right), \text{for } j = 2,3,...,N \tag{9}$$

One can easily check that (6) and (9) have in essence the same structure. However, the Scheme (9) replaces the two Jacobian evaluations by the combination of two functions at different points. This arrangement brings a high convergence rate as in the case of (7) preserving the efficiency of the iterative procedure.

The algorithm may be finalized with (9); however, an additional final step can be added, with the aim of further refining the state vector calculated in (9). Nonetheless, this step should be still sufficiently robust to avoid degrading the reliability of the algorithm. In this sense, the state vector is finally updated using the Forward-Euler method, as follows:

$$\boldsymbol{x}_{k+1} = \boldsymbol{z}^{N+1} - h\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\mathbf{g}\left(\boldsymbol{z}^{N+1}\right) \tag{10}$$

For the sake of summarizing, Figure 1 shows the flowchart of the developed algorithm. It is interesting to note that only one LU decomposition is necessary in the whole developed algorithm. This way, although more function evaluations and vector algebraic operations are required, the new proposal is computationally competitive with NR since the factorization of the Jacobian matrix is by far the heaviest calculation.[5,40]

4.1 | Numerical stability analysis

In this section, the numerical stability of the developed algorithm is proved. The numerical stability of an iterative solver does not guarantee its convergence, but it is a necessary condition for avoiding its divergence. In this regard, nonlinear solvers can be conceived as dynamic systems.[5] For this purpose, it is convenient introducing the following definition.[46]

Definition 2. Hyperbolic point asymptotically stable: Let $\mathbf{G}$ be defined as $\mathbf{G} = \boldsymbol{x}^{(k+1)} - \boldsymbol{x}^{(k)}$. Then, $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*) = \mathbf{0}$ is said to be an equilibrium point of this function. On the other hand, $\boldsymbol{x}^*$ is called hyperbolic

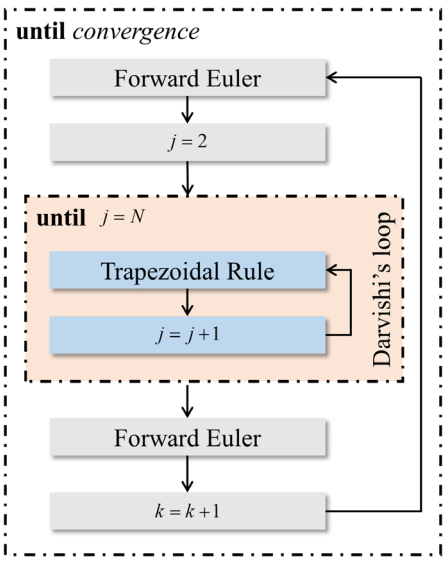

FIGURE 1 Conceptual flowchart of the developed PF solver

if and only if all the eigenvalues associated with the Jacobian of **G** at the equilibrium point have a nonzero real part. In addition, $\boldsymbol{x}^*$ is said to be asymptotically stable (or sink), if all the eigenvalues associated with the Jacobian of **G** at $\boldsymbol{x}^*$ have a negative real part. Finally, $\boldsymbol{x}^*$ is said to be type-m if the Jacobian associated with **G** at $\boldsymbol{x}^*$ has exactly m eigenvalues with a positive real part.

From the Definition 2, one expects from a PF solution (namely $\boldsymbol{x}^*$ such that $\mathbf{g}(\boldsymbol{x}^*)=\mathbf{0}$) to be a type-0 equilibrium point to assert that the developed algorithm is numerically stable. In order to check if this extreme is true for the developed solver, it is necessary to define the function **G** for the new algorithm, as follows:

$$\boldsymbol{G}=-h\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}\left(\boldsymbol{z}^0\right)+\mathbf{g}\left(\boldsymbol{z}^{N+1}\right)\right)-\frac{h}{2}\sum_{j=2}^{N}\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}\left(\boldsymbol{z}^j\right)+\mathbf{g}\left(\boldsymbol{z}^{j-1}\right)\right) \tag{11}$$

By differentiating (11) with respect $\boldsymbol{x}$, one obtains:

$$\begin{aligned}\nabla_{\boldsymbol{x}}\mathbf{G}=&-h\left[\nabla_{\boldsymbol{x}}\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}\left(\boldsymbol{z}^0\right)+\mathbf{g}\left(\boldsymbol{z}^{N+1}\right)\right)+\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}'\left(\boldsymbol{z}^0\right)+\mathbf{g}'\left(\boldsymbol{z}^{N+1}\right)\right)\right]-\frac{h}{2}\sum_{j=2}^{N}\nabla_{\boldsymbol{x}}\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}\left(\boldsymbol{z}^j\right)+\mathbf{g}\left(\boldsymbol{z}^{j-1}\right)\right)\\&-\frac{h}{2}\sum_{j=2}^{N}\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\left(\mathbf{g}'\left(\boldsymbol{z}^j\right)+\mathbf{g}'\left(\boldsymbol{z}^{j-1}\right)\right)\end{aligned} \tag{12}$$

At this point, it is interesting to note that:

$$\boldsymbol{z}^j\big|_{\boldsymbol{x}^*}=\boldsymbol{x}^*,\text{for}\,j=0,1,2,...,N+1 \tag{13}$$

Therefore, evaluation of (12) at $\boldsymbol{x}^*$ yields:

$$\nabla_{\boldsymbol{x}}\mathbf{G}\big|_{\boldsymbol{x}^*}=-h[\mathbf{0}+2\boldsymbol{I}]-\mathbf{0}-h(N-1)\boldsymbol{I}, \tag{14}$$

where $\boldsymbol{I}\in\mathbb{R}^{n\times n}$ is the identity matrix. Simplification of (14) yields:

$$\nabla_{\boldsymbol{x}}\mathbf{G}\big|_{\boldsymbol{x}^*}=-h(N+1)\boldsymbol{I} \tag{15}$$

From (15) one can deduce that the solution of the PF is asymptotically stable for the developed solver. In addition, this point is a sink if $h>0$.

4.2 | Computational implementation

The developed algorithm presents a degree of freedom namely the step size h. To initialize and update it, it is proposed to use heuristic principles,[5] similar to other related techniques.[38] Intuitively, the shorter step size, the more numerically stable, but slower the algorithm is. Under these ideas, the following updating rule is proposed.

$$h=\max\left(0.1,\min\left(\left\|\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\mathbf{g}\left(\boldsymbol{z}^0\right)\right\|_{\infty}^{-1},1\right)\right) \tag{16}$$

By the rule (16), the step size is inversely proportional to the update vector. Thereby, if the Newton's increment vector is too large, the step size is fixed shorter to compensate it and avoid numerical instability. The updating algorithm (16) also limits the step size by upper and lower bounds, which have been set in this case equal to 0.1 and 1, respectively. These values respond to empirical evidences observed by the authors.

Although the step size could be kept fixed through the developed algorithm, it is interesting to update it within the Darvishi's loop exploiting the information provided by the points calculated at this stage. This way, the rule (16) can be applied as

$$h = \max\left(0.1, \min\left(\left\|\left[\mathbf{g}'\left(\boldsymbol{z}^0\right)\right]^{-1}\mathbf{g}\left(\boldsymbol{z}^j\right)\right\|_\infty^{-1}, 1\right)\right) \tag{17}$$

The other degree of freedom (N) determines how many times the Darivshi's loop is executed. This parameter can be simply set sufficiently large to increase the order of convergence. Previous experiments carried out by the authors showed that $N = 5$ works quite well in a wide range of systems and scenarios. However, as the algorithm properly evolves, it is convenient to reduce the value of N in order to avoid unnecessary calculations. In this sense, the following "if" condition is used:

$$\textbf{if } \left\|\mathbf{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_\infty < \left\|\mathbf{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_\infty \textbf{ then } N \leftarrow \max(N-1, 1) \tag{18}$$

By (18), the value of N is progressively reduced if the residuals decrease to a minimum of 1. The overall developed algorithm is summarized in Algorithm 1 using pseudocode. Here, $\|\mathbf{g}\|_\infty \geq \varepsilon$ has been considered a convergence criteria, where $\varepsilon \in \mathbb{R}^+$ is a preset convergence threshold. Also, the developed solution procedure is considered failed if $k > k_{\max}$, which normally indicates divergence.

Algorithm 1. *Developed PF solver based on the proposed algorithm*

Let $\boldsymbol{x}^{(0)}$, ε, $k^{\max}$ and N be given
Initialize iteration counter $k = 0$
while $\left\|\mathbf{g}\left(\boldsymbol{x}^{(k)}\right)\right\|_\infty \geq \varepsilon$ **do**
  $h \leftarrow$ Solve (16) *# Update the step size*
  Solve (8)
  $j = 27$: **for** $j = 2 : N$ **do** *# Darvishi's loop*
    $h \leftarrow$ Solve (17) *# Update the step size*
    Solve (9)
  end do
  Solve (10)
  if $\left\|\mathbf{g}\left(\boldsymbol{x}^{(k+1)}\right)\right\|_\infty \geq \varepsilon$ **then**
    if $k = k^{\max}$ **then**
      break *# Fail*
    end if
  end if
  if $\left\|\mathbf{g}\left(\boldsymbol{x}^{(\boldsymbol{k+1})}\right)\right\|_\infty < \left\|\mathbf{g}\left(\boldsymbol{x}^{(\boldsymbol{k})}\right)\right\|_\infty$ **then**
    $N \leftarrow$ Solve (18)
  end if
  $k \leftarrow k + 1$
end do
return solution $\boldsymbol{x}^{(k)}$

5 | NUMERICAL EXPERIMENTS

This section is devoted to presenting various numerical experiments to check the effectiveness and efficiency of the developed PF solver. Since the new proposal is focused on large-scale ill-conditioned cases, the following systems have been considered.

- The 3012- (System 1) and 3374-bus (System 2) snapshots of the Polish transmission system at 2007-08 winter evening peak.[47]

- The double 2869-bus[48] (System 3), 13 659-bus[49,50] (System 4) and its quadruple version[48] (System 5) portions of the European transmission system from the EU Pegase project.

The main characteristics of the studied systems are summarized in Table 1. As seen in this table, the condition numbers of these systems are sufficiently large to denote their ill-conditioning character, especially if they are compared with other references.[6,24] In addition, these systems can be categorized as ill-conditioned according to Definition 1 (see results in Table 2). The performance of the developed solution framework is compared with the following benchmark PF solvers.

- Conventional NR in polar coordinates.
- Optimal multiplier method in polar coordinates[14] (OMP).
- Darvishi's cubic method applied to PF analysis[39] (3OD).
- High-order Levenberg-Marquardt solver[24] (HOLM).
- Backward-Euler method[20] (BEM).
- Trapezoidal rule applied to PF analysis[17] (TR).
- Heun-King-Werner approach[38] (HKW).

All the tested methods have been coded in Matpower v7.0[51] under the MATLAB R2019a environment. The convergence tolerance ε has been set equal to 10^{-5}, while the different techniques are considered failed if convergence is not achieved after 500 iterations. In order to determine the accuracy of the tested methods, the solutions calculated have been compared with the actual correct solution (ie, the point in which the residuals are zero) of the considered systems. Due to the studied systems have been widely analyzed and are freely available for research purposes, their correct solutions have been reported and are available in Matpower; thus, they can be used for validating PF solvers.

TABLE 1 Main characteristics of the studied systems

				Load			
System #	**# Buses**	**# Branches**	**# Gener.**	**MW**	**Mvar**	**No. of variables (n)**	**Condition number**
1	3012	3572	502	27 169	10 200	5725	1.56×10^7
2	3374	4161	596	48 363	19 527	6355	2.00×10^7
3	5738	9165	1020	264 874	58 015	10 455	5.80×10^7
4	13 659	20 467	4092	381 431	98 523	23 225	7.60×10^8
5	54 636	81 871	16 368	1 525 727	394 093	92 903	5.50×10^9

TABLE 2 Total number of iterations for base cases

System #	**NR**	**OMP**	**3OD**	**HOLM**	**BEM**	**TR**	**HKW**	**Devel.**
1 (Def.)	2	9	1	4	8	18	3	2
2 (Def.)	3	22	2	5	11	25	5	3
3 (Def.)	6	25	Fail	11	11	27	7	6
4 (Def.)	5	25	Fail	11	11	27	8	5
5 (Def.)	6	25	Fail	11	11	27	7	6
1 (FLAT)	Fail	33	Fail	14	12	31	11	6
2 (Flat)	Fail	34	Fail	13	13	32	11	6
3 (Flat)	Fail	32	Fail	21	12	31	11	7
4 (Flat)	Fail	27[a]	Fail	32[b]	14[a]	30	9	8
5 (Flat)	Fail	27[a]	Fail	34[b]	14[a]	30	12[b]	8

[a]Inaccurate solution.
[b]Low voltage solution.

5.1 | Convergence rates

First tests are focused on analyzing the convergence rates of the different solvers. Table 2 provides the total iterations required for each technique for solving the studied systems. To get a good overview of the different methods, the simulations are performed from default starting point and the conventional flat start. The considered default starter is enabled in Matpower by default for PF calculations, and it is considered a better initialization than the flat start. This is due to this starting point is closer to the solution than the flat point, as evidenced in Table 2. Indeed, the different PF solvers normally converged faster from the default point. Moreover, NR is more robust starting from this point since it converged in all systems without any problem.

As seen in Table 2, NR and the developed method performed similarly starting from the default initial guess. On the other hand, 3OD converged faster in the Systems 1 and 2, whereas it failed in the Systems 3 to 5. The remainder solvers converged much slowly, which strengthens the idea that most of robust PF techniques may be unsuitable in well-conditioned cases.[40] When a flat start is considered, NR and 3OD failed in solving all studied cases. This circumstance, according to Definition 1, evidences the ill-conditioning nature of the studied networks. Indeed, in ill-conditioned cases, NR fails if a flat start is used,[5] which supposes the main challenges of these PF cases. In such cases, the developed solver was clearly the most competitive method, successfully converging with less iterations than the remainder solvers.

In addition, it is worth noting that some solvers achieved convergence to inaccurate solutions or, instead of calculating to the high voltage solution, (ie, the type-0 solutions on the top of the P-V curve[31]), some techniques achieved the low voltage one (ie, the type-1 solutions on the bottom of the P-V curve[31]). To illustrate it, Figure 2 compares the voltage angles calculated by BEM from a flat start with the correct solution of the system 5 (only 100 buses are plotted for simplicity). As seen, angles obtained with BEM usually take extreme values, which is symptomatic of unstable voltage regions.[45] On the other hand, HOLM did not actually calculate the low voltage solution of the System 5, but results obtained differed by ∼5% in some cases with respect to the correct solution, which has been considered unacceptable in this paper. These evidences are in concordance with the conclusions drawn in other research,[23,38] where it was evidenced the problems of Levenberg-based methods for achieving physical solutions.

Figure 3 compares the convergence profiles of the different solvers from a flat start. As observed, HOLM was the fastest technique at first iterations. However, the new proposal achieved a high convergence rate after fifth iteration approximately, outperforming HOLM and converged faster than the remainder methods.

It is interesting to explore how the loading level affects the convergence performance of the different solvers. It is well known that PF solvers typically degrades their convergence features under heavy loading conditions.[19,24] To analyze this point, loading conditions of the studied systems are intentionally increased close to the maximum loadability point. To do that, the active and reactive power at PQ buses along the active power at PV buses are progressively increased by a real parameter $\lambda \in \mathbb{R}^+$ until all solvers failed.[52] Table 3 reports the total iterations in such circumstances from a flat start. As expected, the studied solvers normally required more iterations to converge compared with the base case. In some cases, stressing conditions have provoked other undesirable effects like in the case of HOLM, which suffered accuracy issues in all the studied cases. Even under these demanding conditions, the developed solver was well-behaved, successfully solving all the studied cases requiring less iterations than the other methods.

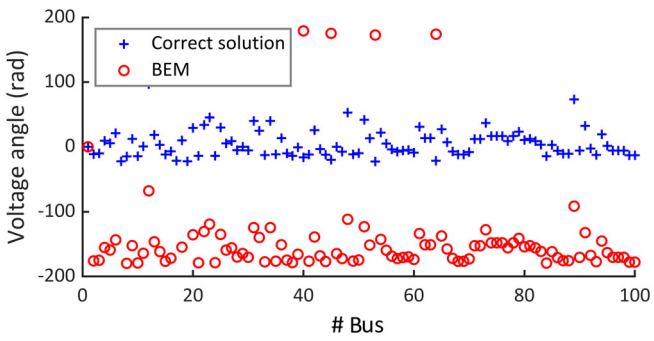

FIGURE 2 Comparison of the voltage angles calculated by BEM in the system 5 from a flat start and the correct solution of the case

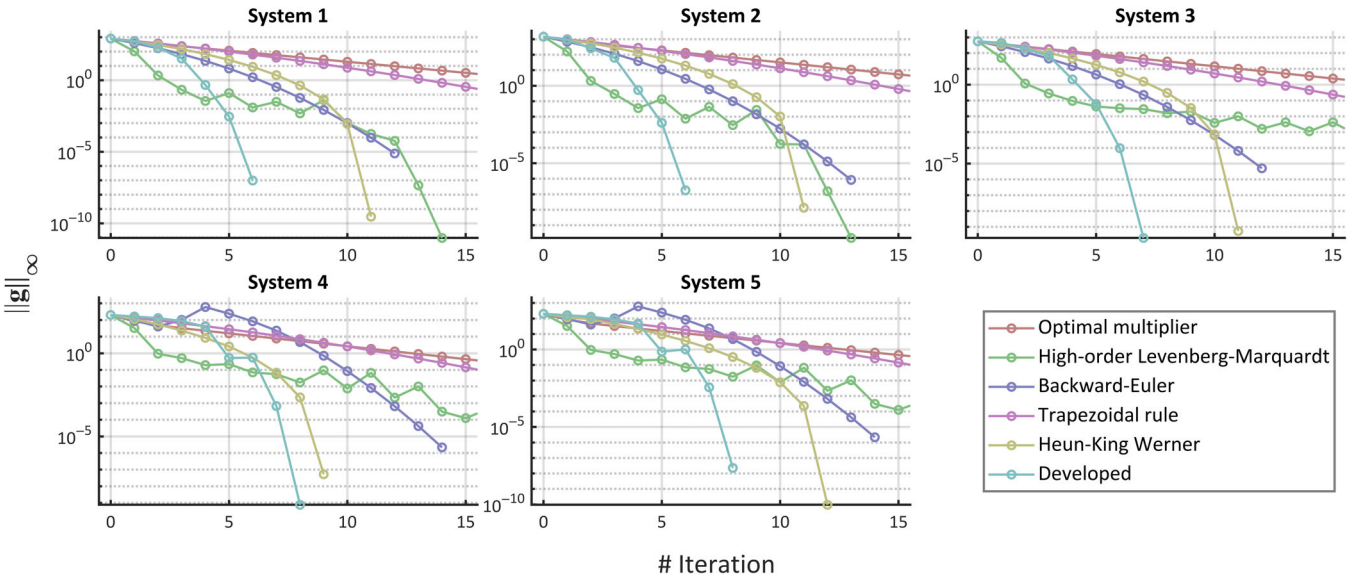

FIGURE 3 Convergence profiles for base cases from a flat start

TABLE 3 Total iterations for limit load cases from a flat start

System #	NR	OMP	3OD	HOLM	BEM	TR	HKW	Devel.
1 (Flat) $\lambda = 1.2734$	Fail	34	Fail	46[a]	14	31	14	10
2 (Flat) $\lambda = 1.1586$	Fail	35	Fail	47[a]	14	32	14	10
3 (Flat) $\lambda = 1.0230$	Fail	33	Fail	151[a]	14	31	13	10
4 (Flat) $\lambda = 1.0017$	Fail	28	Fail	164[a]	12	30	10	10
5 (Flat) $\lambda = 1.0017$	Fail	28	Fail	182[a]	12	30	11[b]	10

[a]Inaccurate solution.
[b]Low voltage solution.

TABLE 4 Solution times (ms) for base cases from a flat start

System #	OMP	HOLM	BEM	TR	HKW	Devel.
1	703.23	354.62	262.56	1310.06	476.08	135.73
2	791.18	342.42	307.32	1517.44	523.49	151.81
3	1226.56	888.93	461.40	2335.23	831.05	279.11
4	2518.02	3112.32	1317.54	5658.90	1703.25	764.16
5	10 346.67	13 673.44	5396.02	23 335.50	9388.32	3152.50

5.2 | Solution times

Next, the computational burden of the different solvers is analyzed. In this sense, the solution times consumed in PF calculation are compared. In this regard, simulations are performed on a 3.4 GHz Intel Core i7-8750H CPU 2.2 GHz personal laptop (16.00 GB RAM). Tables 4 and 5 report the solution times for the results reported in Tables 2 and 3, respectively. Reported results are calculated as the average value of 100 simulations, to minimize the impact of background computational activities. As seen, in all cases, the novel solution approach was the most efficient technique. Promising results obtained with the developed solver are due to its low computational burden (comparable with NR), since only one LU factorization is performed each iteration. In this sense, computational time is quite proportional to the total LU decompositions performed in the whole iterative procedure.[5] Figure 4 compares the total factorizations

TABLE 5 Solution times (ms) for limit load cases from a flat start

System #	OMP	HOLM	BEM	TR	HKW	Devel.
1	724.54	1165.18	306.32	1310.06	605.92	226.20
2	814.45	1237.98	330.96	1517.44	666.26	253.00
3	1264.89	6391.83	538.30	2335.23	982.15	398.73
4	2611.28	15 950.64	1129.32	5658.90	1892.50	955.22
5	10 729.88	73 193.12	4625.16	23 335.50	8605.96	3940.60

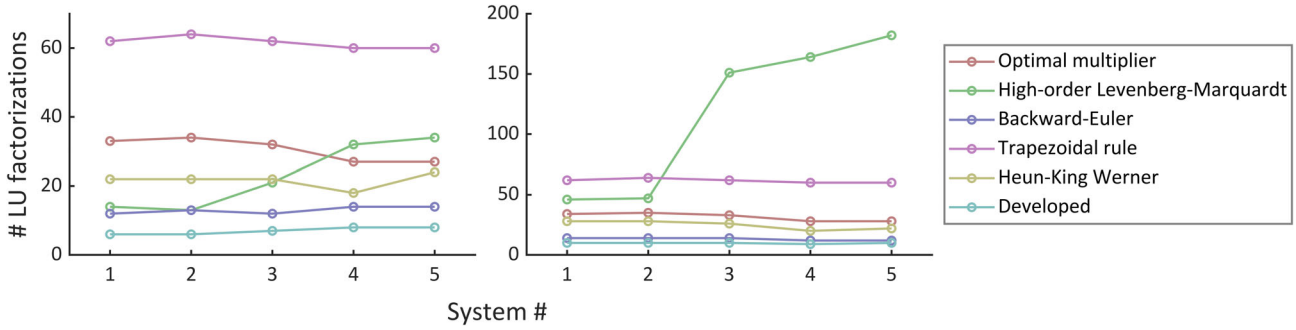

FIGURE 4 Comparison of the total LU decompositions for base (upper) and limit load (bottom) cases from a flat start

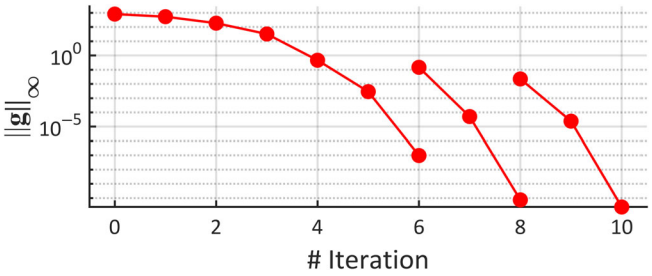

FIGURE 5 Convergence profiles for the system 1 using the developed method from a flat start with reactive limits enabled

required by each solver. As seen, the developed approach needed less LU decompositions to converge, which explains its competitive results.

5.3 | Handling reactive limits

In the mathematical formulation developed in this article, reactive limits of generating units and other equipment restrictions have not been considered. Nevertheless, they can be easily incorporated by a standard PV-PQ switching logic mechanism, as customary in Newton-like solvers.[5,38,52] This procedure is illustrated in Figure 5, where the convergence profiles in the system 1 with reactive limits enabled using the developed solver from a flat start are plotted. As seen, in contrast to the standard solution algorithm, the procedure for incorporating reactive limits concatenates various PF calculations. Indeed, the base case is firstly solved, and then, the algorithm checks if any reactive limit has been violated in PV buses. In affirmative case, those buses are converted to PQ buses, taking the hit limit as reactive injection. With these new circumstances, the system is solved taking the solution of the previous calculation as starting guess. The process is repeated until all generators are within limits or there are not more PV buses, declaring the problem infeasible in such case. As observed in Figure 5, the switching mechanism is repeated three times until achieving a feasible solution. Tables 6 and 7 compare the total iterations and solution times required for the different solvers in the studied cases from a flat start and reactive limits. As seen, the developed solver was still the most competitive one, notably outperforming the remainder approaches. It is worth noting that each loop of the switching mechanism new PQ

TABLE 6 Total iterations with reactive limits from a flat start

System #	NR	OMP	3OD	HOLM	BEM	TR	HKW	Devel.
1 (Flat)	Fail	46	Fail	20	28	64	17	10
2 (Flat)	Fail	44	Fail	20	33	72	19	10
3 (Flat)	Fail	68	Fail	31	39	91	21	12
4 (Flat)	Fail	49[b]	Fail	35[a]	25[b]	51	12	10
5 (Flat)	Fail	Fail	Fail	37[a]	Fail	51	Fail	10

[a]Inaccurate solution.
[b]Low voltage solution.

TABLE 7 Solution times (ms) with reactive limits from a flat start

System #	OMP	HOLM	BEM	TR	HKW	Devel.
1	1175.76	587.20	740.04	3149.44	842.18	271.30
2	1263.24	664.20	957.99	3763.44	999.97	297.12
3	3077.68	1656.64	1799.07	8204.56	1903.44	561.84
4	5521.81	4839.10	2858.00	11 663.19	2754.48	1156.31
5	—	17 953.51	—	45 268.62	—	4451.23

buses is added to the system of equations and, consequently, the system grows with more variables and expressions. This circumstance makes the problem harder to solve until the point that some solvers failed in the system 5.

6 | CONCLUSIONS AND FUTURE WORKS

A novel PF solver suitable to realistic large-scale ill-conditioned cases has been developed. The new proposal is based on the combination of different numerical techniques with the aim of jointly exploiting their strengths and minimizing their debilities. More precisely, the developed technique encompasses the Forward-Euler method, the Darvishi's cubic technique, and the Trapezoidal rule applied to PF analysis. The Forward-Euler and Trapezoidal rules are robust methods, whereas the Darvishi's technique provides higher convergence rate. The involved techniques have been combined under a predictor-corrector scheme, which has results very suitable to their characteristics. Moreover, different heuristic updating rules have been developed for the different parameters involved in the developed solution procedure, which facilitates its computational implementation. Finally, the numerical stability of the developed methodology has been proved, by assimilating it with a dynamical system.

Extensive results have been provided on various large-scale ill-conditioned techniques under different scenarios. The superiority of the developed solver has been checked by comparison with other conventional and modern robust PF solvers. The results obtained by the new proposal are promising. It turned out to be more robust than other methods, successfully solving all the studied systems even under stressed scenarios. The developed solution algorithm is also very efficient, as shown by comparing the solution times of different techniques. In all cases, the proposal is the most efficient methodology, minimizing the number of expensive calculations like LU factorizations. Reliability of the new proposal has been also checked by verifying the accuracy of the solutions calculated. In all cases, the proposed method achieved high accurate solutions, with errors lower than 5% compared with the correct solutions of the studied systems even under demanding conditions and in the presence of equipment limits.

Promising results obtained should be further analyzed in other related tools. In this regard, the developed solver may result an invaluable contribution for voltage stability analysis or expansion planning purposes, in which the high efficiency of the developed solver may be even more notable in comparison with other techniques. In addition, the developed methodology is versatile enough to incorporate other alternative PF formulations. Future works should study the performance of the new proposal with such formulations.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

PEER REVIEW

The peer review history for this article is available at https://publons.com/publon/10.1002/2050-7038.13194.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Marcos Tostado-Véliz https://orcid.org/0000-0001-7076-1065
Hany M. Hasanien https://orcid.org/0000-0001-6595-6423
Salah Kamel https://orcid.org/0000-0001-9505-5386
Francisco Jurado https://orcid.org/0000-0001-8122-7415

REFERENCES

1. Wadhwa CL. *Electrical Power Systems*. 7th ed. Kent, UK: New Academic Science; 2018.
2. Milano F. *Power System Modelling and Scripting*. New York, NY: Springer-Verlag; 2010.
3. Idema R, Lahaya DJP. *Computational Methods in Power System Analysis*. Paris: Atlantic Press; 2014.
4. Xie N, Torelli F, Bompard E, Vaccaro A. Dynamic computing paradigm for comprehensive power flow analysis. *IET Gener Transms Distrib*. 2013;7(8):832-842.
5. Milano F. Continuous Newton's method for power flow analysis. *IEEE Trans Power Syst*. 2009;24(1):50-57.
6. Pourbagher R, Derakhshandeh SY. A powerful method for solving the power flow problem in the ill-conditioned systems. *Int J Electr Power Energy Syst*. 2018;94:88-96.
7. 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 Electr Power Energy Syst*. 2019;113:298-309.
8. Shen Z, Chiang H-D, 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 Electr Power Energy Syst*. 2020;115:105500.
9. Tang K, Dong S, Shen J, Zhu C, Song Y. A robust and efficient two-stage algorithm for power flow calculation of large-scale systems. *IEEE Trans Power Syst*. 2019;34(6):5012-5022.
10. Pourbagher R, Derakhshandeh SY, Golshan MEH. An adaptive multi-step Levenberg-Marquardt continuation power flow method for voltage stability assessment in the ill-conditioned power systems. *Int J Electr Power Energy Syst*. 2022;134:107425.
11. Wang L, Chiang H-D. Toward online line switching for increasing load margins to static stability limit. *IEEE Trans Power Syst*. 2015;31(3):1744-1751.
12. Stott B, Alsac O. Fast decoupled load flow. *IEEE Trans Power Appar Syst*. 1974;PAS-93(3):859-869.
13. Iwamoto S, Tamura Y. A load flow calculation for ill-conditioned power systems. *IEEE Trans Power Appar Syst*. 1981;PAS-100(4):1736-1743.
14. Schaffer MD, Tylavsky DJ. A nondiverging polar-form Newton-based power flow. *IEEE Trans Ind Appl*. 1988;24(5):870-877.
15. Tripathy SC, Prasad GD, Malik OP, Hope GS. Load-flow solutions for ill-conditioned power systems by a Newton-like method. *IEEE Trans Power Appar Syst*. 1982;PAS-101(10):3648-3657.
16. Sachdev MS, Medicherla TKP. A second order load flow technique. *IEEE Trans Power Appar Syst*. 1977;PAS-96(1):189-197.
17. Tostado-Véliz M, Kamel S, Jurado F. Comparison of various robust and efficient load-flow techniques based on Runge-Kutta formulas. *Electr Power Syst Res*. 2019;174:105881.
18. Tostado-Véliz M, Kamel S, Jurado F. Development of different load flow methods for solving large-scale ill-conditioned systems. *Int Trans Electr Energy Syst*. 2019;29(4):e2784.
19. Tostado-Véliz M, Kamel S, Jurado F. A robust power flow algorithm based on Bulirsch-Stoer method. *IEEE Trans Power Syst*. 2019;34(4):3081-3089.
20. Milano F. Implicit continuous Newton method for power flow analysis. *IEEE Trans Power Syst*. 2019;34(4):3309-3311.
21. Xie N, Bompard E, Napoli R, Torelli F. Widely convergent method for finding solutions of simultaneous nonlinear equations. *Electr Pow Syst Res*. 2012;83(1):9-18.
22. Torelli F, Vaccaro A. A second order dynamic power flow model. *Electr Pow Syst Res*. 2015;126:12-20.
23. Milano F. Analogy and convergence of Levenberg's and Lyapunov-based methods for power flow analysis. *IEEE Trans Power Syst*. 2016;31(2):1663-1664.
24. Pourbagher R, Derakhshandeh SY. Application of high-order Levenberg-Marquardt method for solving the power flow problem in the ill-conditioned systems. *IET Gener Transm Distrib*. 2016;10(12):3017-3022.
25. Sheng H, Chiang H-D. CDFLOW: a practical tool for tracing stationary behaviors of general distribution networks. *IEEE Trans Power Syst*. 2014;29(3):1365-1371.

26. Trias A, Marín JL. The holomorphic embedding Loadflow method for DC power systems and nonlinear DC circuits. *IEEE Trans Circuits Syst I Reg Pap*. 2016;63(2):322-333.
27. Wang T, Chiang H-D. On the global convergence of a class of Homotopy methods for nonlinear circuits and systems. *IEEE Trans Circuits Syst II Express Briefs*. 2014;61(11):900-904.
28. Chiang H-D, Wang T. Novel Homotopy theory for nonlinear networks and systems and its applications to electrical grids. *IEEE Trans Control Netw Syst*. 2017;5(3):1051-1060.
29. Rao S, Feng Y, Tylavsky DJ, Subramanian MK. The holomorphic embedding method applied to the power-flow problem. *IEEE Trans Power Syst*. 2016;31(5):3816-3828.
30. Rao S, Tylavsky DJ, Feng Y. Estimating the saddle-node bifurcation point of static power systems using the holomorphic embedding method. *Int J Electr Power Energy Syst*. 2017;84:1-12.
31. Feng Y, Tylavsky DJ. A holomorphic embedding approach for finding the Type-1 power-flow solutions. *Int J Electr Power Energy Syst*. 2018;102:179-188.
32. Chiang H-D, Wang T, Sheng H. A novel fast and flexible holomorphic embedding power flow method. *IEEE Trans Power Syst*. 2018; 33(3):2551-2562.
33. Wang T, Chiang H-D. On the holomorphic and conjugate properties for holomorphic embedding methods for solving power flow equations. *IEEE Trans Power Syst*. 2020;35(4):2506-2515.
34. Mokhlis H, Shahriari A, Bakar AHA, Illias HA, Karimi M. Improved step size Newton Raphson method using quadratic equations properties in ill-conditioned power system. *Int Trans Electr Energy Syst*. 2014;24(9):1323-1342.
35. Zhang G, Wang C, Wang H, Xie N. On the convergence of the implicit Z-bus power flow method for distribution systems. *Electr Power Syst Res*. 2019;171:74-84.
36. Zhao T-Q, Chiang H-D, Koyanagi K. Convergence analysis of implicit Z-bus power flow method for general distribution networks with distributed generators. *IET Gener Transm Distrib*. 2016;10(2):412-420.
37. Tostado M, Kamel S, Jurado F. Several robust and efficient load flow techniques based on combined approach for ill-conditioned power systems. *Int J Electr Power Energy Syst*. 2019;110:349-356.
38. Tostado-Véliz M, Kamel S, Jurado F. An efficient power-flow approach based on Heun and king-Werner's methods for solving both well and ill-conditioned cases. *Int J Elect Power Energy Syst*. 2020;119:105869.
39. Pourbagher R, Derakhshandeh SY. Application of high-order Newton-like methods to solve power flow equations. *IET Gener Transm Distrib*. 2016;10(8):1853-1859.
40. Tostado-Véliz M, Kamel S, Jurado F, Ruiz-Rodriguez FJ. On the applicability of two families of cubic techniques for power flow analysis. *Energies*. 2021;14(14):4108.
41. Darvishi MT, Barati A. A third-order Newton-type method to solve systems of nonlinear equations. *Appl Math Compt*. 2007;187(2): 630-635.
42. Xiao X, Yin H. Achieving higher order of convergence for solving systems of nonlinear equations. *Appl Math Comput*. 2017;311:251-261.
43. Tostado-Véliz M, Kamel S, Jurado F. Two efficient and reliable power-flow methods with seventh order of convergence. *IEEE Syst J*. 2021;15(1):1026-1035.
44. Chun C, Neta B. Developing high order methods for the solution of systems of nonlinear equations. *Appl Math Comput*. 2019;342: 178-190.
45. Sarnari AJ, Živanović R, Al-Sarawi S. Augmenting load flow software for reliable steady-state voltage stability studies. *Int Trans Electr Energy Syst*. 2019;29(9):e12047.
46. Lee J, Chiang H-D. Convergent regions of the Newton homotopy method for nonlinear systems: theory and computational applications. *IEEE Trans Circuits Syst I Fundam Theory Appl*. 2001;48(1):51-66.
47. Matpower Benchmark Systems. https://matpower.org/download/. Accessed July 14, 2021.
48. Tostado-Véliz M, Kamel S, Jurado F. Matpower Ill Conditioned Systems https://zenodo.org/record/3514739. Accessed July 14, 2021.
49. Josz C, Fliscounakis S, Maeght J, Panciatici P. AC Power Flow Data in Matpower and QCQP Format: iTesla, RTE Snapshots, and PEGASE. Available from http://arxiv.org/abs/1603.01533. v3 Accessed July 14, 2021.
50. 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):4909-4917.
51. 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):12-19.
52. Gómez-Expósito A, Gómez-Quiles C. Factorized load flow. *IEEE Trans Power Syst*. 2013;8(4):4607-4614.

How to cite this article: Tostado-Véliz M, Hasanien HM, Turky RA, Kamel S, Jurado F. Solving realistic large-scale ill-conditioned power flow cases based on combination of numerical solvers. *Int Trans Electr Energ Syst*. 2021;31(12):e13194. doi:10.1002/2050-7038.13194