

Received July 14, 2021, accepted July 28, 2021, date of publication August 3, 2021, date of current version August 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102447

# A Novel Power Flow Solution Paradigm for Well and Ill-Conditioned Cases

MARCOS TOSTADO-VÉLIZ<sup>1</sup>, TALAL ALHARBI<sup>2</sup>, (Member, IEEE),  
OMAR ALRUMAYH<sup>3</sup>, (Member, IEEE), SALAH KAMEL<sup>4</sup>,  
AND FRANCISCO JURADO<sup>1</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electrical Engineering, University of Jaén, 23700 EPS Linares, Jaén, Spain

<sup>2</sup>Department of Electrical Engineering, College of Engineering, Qassim University, Buraydah 52571, Saudi Arabia

<sup>3</sup>Department of Electrical Engineering, College of Engineering, Qassim University, Unaizah 56452, Saudi Arabia

<sup>4</sup>Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt

Corresponding author: Talal Alharbi (atalal@qu.edu.sa)

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

**ABSTRACT** This paper develops a novel four-stage power flow solver for ill-conditioned systems. Although the developed solver could be considered efficient, it is not competitive with the Newton-Raphson method in well-conditioned cases. With the aim of being fully competitive in a wide range of cases and scenarios, the developed algorithm is integrated within a novel efficient solution paradigm. As a result, a robust and efficient solution framework, competitive in both well and ill-conditioned cases, is obtained. The new proposals are tested in various well and ill-conditioned cases from 30-, to 13,659-buses. Results obtained with the developed solvers are promising.


**INDEX TERMS** Power-flow analysis, ill-conditioned cases, computational efficiency.

## I. INTRODUCTION

### A. MOTIVATION

Power flow calculation in ill-conditioned cases has been profusely studied for decades. As a result, a plethora of robust techniques is available in the literature. However, very few of these methods have found widespread usage in industry, and, currently, the conventional Newton-Raphson (NR) is still assumed as the most standard power flow solver. This is due to the fact that very few robust methods are available, which cannot compete with NR, especially in large-scale networks.

Well-conditioned systems are the most common situation in power system operation; thus, NR has offered a satisfying performance in most cases. However, ill-conditioned cases are nowadays more common due to the observed increasing energy demand in modern power systems and the difficulties in upgrading the network infrastructures. These facts currently are forcing the power components to operate close to their functional limits [1], [2]. Consequently, robust power flow solvers may presumably play a vital role in a future power system paradigm.

The associate editor coordinating the review of this manuscript and approving it for publication was Gokhan Apaydin .

### B. LITERATURE REVIEW

Power flow is likely the most important computational tool in power system analysis. It finds a wide variety of applications such as voltage stability analysis [3], optimal power flow [4] or state estimation [5], among others. Power flow calculation has been covered in multiple works since its first formulations in the mid-'50s [6]. Some references classified power flow problems into three different categories [7]:

- Well-conditioned cases*: this is still the most common situation. In these cases, the conventional solvers, such as NR, converge to an accurate solution from a flat start. These cases do not entail convergence difficulties for most solvers. For this kind of system, apart from the conventional NR, a variety of decoupled techniques [8], linear models [9], hybrid AC/DC solvers [10] and high order Newton-like methods [11] have been proposed.
- Ill-conditioned cases*: despite the fact that these systems are still solvable, conventional NR fails to reach the solution from a flat start [7]. Robust techniques have proved to be efficacious for solving ill-conditioned cases.
- Bifurcation points*: on these points, the power flow solution does exist; however, the Jacobian matrix is singular or near singular, which provokes instability of

the Newton-based methods. Continuation techniques [3] or Regularization schemes [12] prevent the singularity of the Jacobian matrix at bifurcation points.

- d. *Unsolvable cases*: in this case, the power flow solution does not exist. This is typically due to the loading level has surpassed the maximum loadability point of the system. Different approaches devoted to providing approximated solutions [13] or proposing countermeasures for restoring solvability [14] have been developed for unsolvable cases.

This paper is focused on ill-conditioned cases. The power flow solution of this kind of case came to be studied at the end of '70s [15], [16] by the so-called optimal multiplier methods. These first works proposed a minimization approach derived from a second-order Taylor expansion of the power flow equations in rectangular coordinates. As a result, a so-called optimal multiplier is calculated, which modifies the Newton's increment vector to be conducted in a direction where the residuals are reduced. The optimal multiplier solution methods are, theoretically, no divergent. However, the calculated optimal multiplier tends to be very short or almost null when the solution is approached [17]. This undesirable behaviour nullifies the effect of the increment vector and provokes that the optimal multiplier solution methods might be trapped on a local minimum. The optimal multiplier techniques are not suitable for the polar form of the power flow equations due to the presence of transcendental functions [18].

The optimal multiplier techniques can be included within the family of line search techniques or trust-region methods [19]–[21]. These methods aim to be globally convergent; however, this important feature is typically achieved by deteriorating their convergence properties. Thus, these solvers frequently present a notably linear convergence, which supposes an important drawback w.r.t to NR or other conventional solvers. In addition, if the starting guess used for initializing the iterative procedure is far away from the solution, the convergence characteristics of the Newton's descent vector are lost, and the algorithm may be trapped on a local minimum [22].

Another category of robust techniques is based on the Continuous Newton's method [23]–[25]. This paradigm poses a formal analogy between the Forward Euler method and most of the available optimal multiplier approaches. On the basis of this analogy, Milano deduced that any integration routine (e.g. the family of the explicit Runge-Kutta formulas), can be considered for developing robust solution methods [7]. A solver based on the 4<sup>th</sup> order Runge-Kutta formula was developed in this same reference, which was demonstrated to be more competitive than the optimal multiplier method in [16]. The Continuous Newton's philosophy has been exploited for developing a variety of robust solvers based on different Adams-Bashforth methods [26], Bulirsch-Stoer algorithm [27] and Runge-Kutta formulas [28]. As justified in [7], only the explicit formulation of the Continuous Newton's method is attractive for power flow analysis, due to the implicit formulation would require the factorization of the

Hessian matrix. However, the same author recently proposed an implicit form in [20]. Nevertheless, in this latter reference, it has been proved that the effect of the Hessian matrix is immaterial and, consequently, the explicit formulation of the Continuous Newton's paradigm is by far more competitive than the implicit one. Those methods based on the Continuous Newton's paradigm are widely rather than globally convergent. In addition, they share some of the disadvantages of the Newton's based techniques, and for example, they fail at turning points or unsolvable cases. Also, these solvers are quite inefficient compared to NR; one should note that a power flow approach based on a Runge-Kutta formula would require as matrix factorizations as the order of the integration technique considered. Thus, the low order Runge-Kutta methods are usually more competitive than the high order ones [28].

The Levenberg-type methods have recently gained popularity for power flow analysis [2], [12], [29]–[31]. These techniques are, in fact, modifications of NR, which aim to avoid the vulnerability of the conventional technique at bifurcation points [32], [33]. It is achieved by adding some elements on the diagonal of the Jacobian matrix. The most common approach is based on adding the identity operator to the Jacobian matrix. The elements of this identity matrix are pondered by a series of damping factors, which ideally must approach zero as the algorithm evolves; thus, usually, an adaptive mechanism is preferred [2], [12], [30], [31]. The series of damping factors are the most critical aspect of the Levenberg-type technique. As commented in [29], the damping factor's value strongly influences the accuracy of the results obtained by the Levenberg method. Also, the series of the damping factors may lead to a very slow mapping if they are not carefully selected; however, a unified criterion for tuning it does not exist. It is worth mentioning that the Levenberg algorithm is more computationally expensive than NR, as an extra matrix-matrix product and an extra matrix-vector product have to be computed each iteration.

In some references, the power flow equations have been raised as an artificial dynamic system [1], [34], [35]. The equilibrium points of the proposed dynamic systems are, in fact, the solution of the power flow equations, so that solving the resulting set of ordinary differential equations is equivalent to solve the set of nonlinear algebraic power flow equations. By making use of the Lyapunov theory, it is demonstrated that the equilibrium points of the equivalent set of differential equations present an exponential asymptotic convergence. Thus, the resulting algorithm is barely affected by the starting guess; hence the obtained mapping is widely convergent. Nonetheless, the resulting algorithm has to be solved using costly calculations like the ode routines in Matlab, making this approach impractical for large realistic systems [1].

Apart from the solvers referenced above, a series of techniques primarily devoted to distribution systems have been studied for decades. Some of them are based on the Backward-Forward and ladder algorithms [36], [37], which iteratively "sweep" the network until the difference between

two consecutive solutions becomes within a specific tolerance. These approaches are based on linear operations; therefore, they are barely affected by some network characteristics like the R/X ratio. Nevertheless, despite that they avoid heavy calculations like matrix factorizations, these techniques are usually time-consuming due to their strong linear convergence, which is especially relevant in heavy loading systems (see [38, Sec V.D]). In addition, this kind of method presents difficulties to manage with PV buses even with weakly meshed networks [39]. Another family of power flow solvers for distribution networks is, in fact, based on the well-known Gauss method. The most famous approaches of this family are the Z-implicit [40], [41], and the loop algorithms [42]. These kinds of methodologies normally are efficient than the Backward-Forward and ladder algorithms. In addition, they can easily accommodate meshed networks. Nonetheless, issues for incorporating PV buses are still found, as pointed out in [43].

### C. CONTRIBUTIONS AND PAPER ORGANIZATION

Motivated by a constant effort to develop efficient and robust power flow solvers, this paper proposes a novel robust four-stage algorithm (4SA). Although it is quite efficient and competitive with the most available robust solvers, it is not competitive with NR in well-conditioned systems. For expanding the suitability of 4SA, it is also integrated within a developed solution paradigm.

The developed techniques find some similitudes with the optimal multiplier or line search techniques. For example, our proposals are also based on the truncation of the Newton's increment vector (i.e. modifying the increment vector by multiplying it by a step size value). However, the introduced paradigms aim at overcoming some of the difficulties posed by this kind of technique. Specifically, they are developed with the aim to avoid the slow convergence and locally trapping phenomena that some robust algorithms (e.g. [16], [19] or [28]), may experience when the starting guess is not suitable or lies far away from the solution. Unlike other line search techniques based on finding the largest step-size that satisfies the Armijo-Goldstein conditions, our proposals aim to find the most suitable truncation (step-size) of the Newton's increment vector using a novel search strategy over the state vector space. Thus, the a priori truncation of the Newton's increment vector is determined by finding the so-called saddle point, which is defined as that point on the Newton's direction beyond which the residuals grow; in other words, the Newton's increment vector is not a descent direction beyond the saddle-point. This saddle-point is calculated through various stages, devoted to progressively refining the value of the calculated truncation in order to ensure a good trade-off between robustness and fast convergence. The truncation which produces the saddle-point is assumed to be the most suitable approach for updating the vector of variables as the quadratic convergence features of the NR technique does not deteriorate excessively. Thus, by this approach, the global

convergence of the iterative procedure is not ensured to promote a fast convergence.

As commented, the developed 4SA is not competitive with NR in a well-conditioned system. Therefore, the developed algorithm is integrated within a suitable solution framework which, by making proper use of some results obtained at early stages of 4SA, is able to heuristically determine if the studied system is well or ill-conditioned; in the former case, our method proceeds as NR, while 4SA solves ill-conditioned cases. Thus, the proposed paradigm can efficiently solve both well and ill-conditioned systems, which is not typically encountered on the available robust solvers. This outstanding characteristic makes the developed solution paradigm interesting and suitable for industrial applications, where the system's conditioning is frequently unknown, and a high degree of efficiency in the calculations is typically required.

The computational burden of the developed paradigm is comparable to NR in both well and ill-conditioned cases, making our method scalable to large and even complex-scale networks. This paper is limited to present a simple illustrative version of our paradigm in polar coordinates; however, it is sufficiently versatile to be adapted to different formulations, functional limits and scenarios. Various numerical results are provided in order to validate and show the performance of the developed paradigm.

The remainder of this paper is organized as follows. Due to our proposal is described in this paper using the power flow formulation in polar coordinates, the power flow equations in this form and its solution using NR are described in Section II. The developed 4SA for ill-conditioned cases is introduced in Section III. In Section IV, 4SA is fully integrated within a developed paradigm for effectively solving well and ill-conditioned cases. Various numerical results on various well and ill-conditioned systems are presented in Section V. The paper is concluded with Section VI.

## II. PRELIMINARIES

The power flow equations in polar coordinates are defined as a set of  $n$  nonlinear algebraic equations as follows:

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

where:

$$\mathbf{g}(\mathbf{x}) = \begin{cases} \mathbf{g}_P, & \text{For all buses} \\ \mathbf{g}_Q, & \text{For PQ buses} \end{cases} \quad (2a)$$

$$g_{P_i} = P_i^{sp} - \sum |V_i| |V_j| |Y_{ij}| \cos(\theta_{ij} - \delta_j + \delta_i) \quad (2b)$$

$$g_{Q_i} = Q_i^{sp} - \sum |V_i| |V_j| |Y_{ij}| \sin(\theta_{ij} - \delta_j + \delta_i) \quad (2c)$$

$$\mathbf{x} = [\delta_g, \delta_l, \mathbf{V}_l]^T \quad (3)$$

where,  $P_i^{sp} \in \mathbb{R}$  and  $Q_i^{sp} \in \mathbb{R}$  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.  $n_l \in \mathbb{N}$  is the total number of PQ buses,  $n_g \in \mathbb{N}$  is the total number of PV buses,  $\delta_g \in \mathbb{R}^{n_g}$  is the

vector of voltage angles at PV buses,  $\delta_I \in \mathbb{R}^m$  is the vector of voltage angles at PQ buses and  $V_I \in \mathbb{R}^m$  is the vector of voltage angles at PQ buses.

As (1) are nonlinear equations, their solution cannot be obtained directly. The power flow problem has been customarily handled using a plethora of iterative techniques. The most popular is NR method, which is defined by the following map:

$$\phi_k = [\mathbf{g}'(\mathbf{x}_k)]^{-1} \mathbf{g}(\mathbf{x}_k) \tag{4a}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \phi_k \tag{4b}$$

where,  $\mathbf{g}' = \nabla_{\mathbf{x}} \mathbf{g} \in \mathbb{R}^{n \times n}$  is the Jacobian matrix and the subindexes denote the iteration counter. Indeed, equation (4a) corresponds with the Newton's increment vector, which is desirable to be a descent direction in order to reduce the residuals as the mapping (4) evolves. Nonetheless, it is well-known that the mapping (4) may be unstable in ill-conditioned systems [7].

### III. THE DEVELOPED FOUR STAGE ALGORITHM

The developed 4SA aims to improve the convergence abilities of NR in ill-conditioned systems. Fig. 1 shows a schematic summary of the developed 4SA. Each stage is further explained in the following sections.

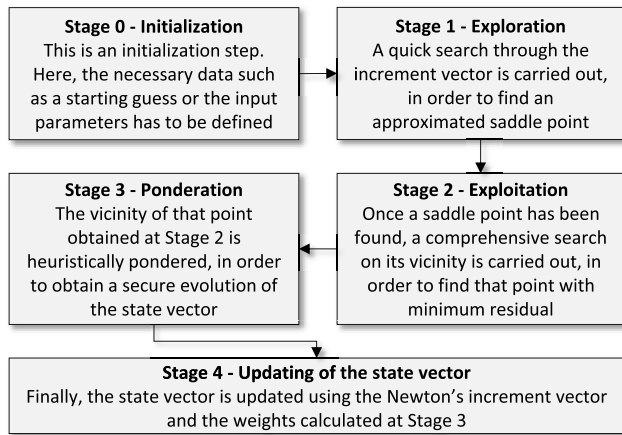


FIGURE 1. Schematic summary of the developed 4SA.

#### A. STAGE 1-EXPLORATION

Let us consider a variety of “intermediate” state vectors calculated as:

$$\mathbf{y}_k^j = \mathbf{x}_k - z_k^j \phi_k \tag{5}$$

where,  $z^j \in \mathbb{R}^+$  are different truncations, which are collected in the so-called exploration vector  $\mathbf{z}$ . The exploration vector can be built using Algorithm 1. In this case, the infinity norm of the residuals is assumed to be a good indicator of the proximity of the solution [29].

As observed, the largest truncation is equal to 1, i.e. the largest truncation in fact does not modify  $\phi$ . For those problems where NR successfully converge, the lowest residual is

#### Algorithm 1 Construction of the Exploration Vector

```

1: Let  $\xi_k$  be given
2: Initialize  $j \leftarrow 1$  and  $flag \leftarrow 0$ 
3: while  $flag == 0$  do
4:    $z_k^j \leftarrow j \xi_k$ 
5:   if  $z_k^j \geq 1$  then
6:      $z_k^j \leftarrow 1$ 
7:      $flag \leftarrow 1$ 
8:   end
9:    $j \leftarrow j + 1$ 
10: end do
11: return exploration vector  $z_k \leftarrow [z_k^1, z_k^2, \dots, 1]^T$ 
  
```

typically observed for  $z^j = 1$  (i.e. at  $\mathbf{x}_{k+1}$ ), since (4a) is a descent direction on the  $y$ -space. However, in ill-conditioned problems, this behaviour is not generally observed and, beyond some determined truncation, the descending trending is lost and the residual grows (see Fig. 2), it means that the Newton's increment vector is not a descent direction over the whole  $y$ -space. The furthest point (4a) is still a descent direction for  $\mathbf{g}$ , called the saddle point. The developed 4SA assumes that the saddle point offers the best trade-off between robustness and convergence rate; consequently, its corresponding truncation should be considered for modifying the Newton's increment vector. The main objective of the Exploration stage is to locate an approximation of the saddle point and its corresponding truncation  $\hat{h}^L$ , which can be achieved by using Algorithm 2. The elements of the exploration vector are calculated in increasing order using the so-called jump parameter  $\xi$ , which is given by:

$$\xi_k = \min \left( \|\phi_k\|_{\infty}^{-1}, 1 \right) \tag{6}$$

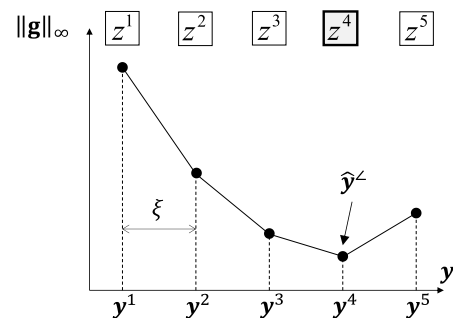


FIGURE 2. Graphical interpretation of the approximated saddle point  $\hat{y}^L$ . As observed, beyond this point the residual grows. In this example, the approximated saddle point is located for  $z^4$ . One should note that the accuracy with  $\hat{y}^L$  is estimated, strongly depends on the value of the jump parameter.

As observed in Fig. 2, the larger  $\xi$ , the less accurate estimation of the saddle point. However, this latter aspect is not especially relevant at this stage since only a first approximation to the saddle point aims to be calculated. The Exploration stage usually ends when an approximation of the saddle point

**Algorithm 2** Exploration Stage

```

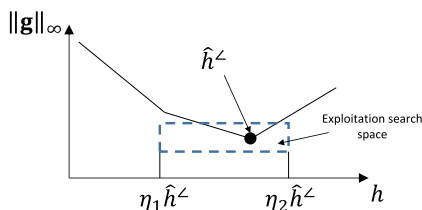
1: Let  $x_k, z_k$  and  $\phi_k$  be given
2: Initialize  $j \leftarrow 1$  and  $saddle \leftarrow 0$ 
3: Take  $y_k^0 \leftarrow x_k$ 
4: while  $saddle == 0$  do
5:   if  $z_k^j == 1$  then
6:      $\hat{h}_k^L \leftarrow 1$ 
7:     break # No saddle point
8:   end if
9:   Calculate  $y_k^j = x_k - z_k^j \phi_k$ 
10:  if  $\|g(y_k^j)\|_\infty > \|g(y_k^{j-1})\|_\infty$  then
11:    if  $j == 1$  then
12:      break # Exploration fails
13:    else
14:       $saddle \leftarrow 1$ 
15:       $\hat{h}_k^L \leftarrow z_k^{j-1}$ 
16:    end if
17:  end if
18:   $j \leftarrow j + 1$ 
19: end do
20: return approximated saddle truncation  $\hat{h}_k^L$ 

```

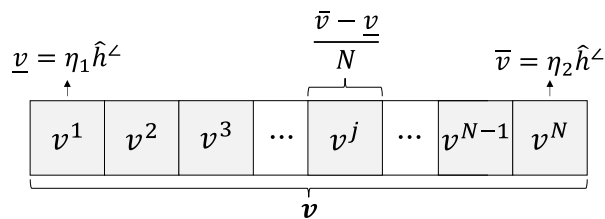
namely  $\hat{y}^L$  is located. One should observe in Algorithm 2 that the Exploration stage may fail if the saddle point is located at  $y^0$ , i.e. (4a) is not a descent direction. It may also occur that  $\hat{y}^L$  does not exist, this is typical of well-conditioned cases. In the former case, it is determined that the system is not solvable for the given loading conditions and the algorithm stops prematurely. In the latter case, the largest truncation (i.e.  $\hat{h}^L = 1$ ) is taken.

**B. STAGE 2-EXPLOITATION**

The objective of the Exploitation stage is to calculate a more accurate approximation of the saddle point. To do that, the vicinity of  $\hat{h}^L$  is further explored. The exploitation search space is defined by two parameters  $\eta_1 < 1$  and  $\eta_2 > 1$  as observed in Fig. 3. The parameters  $\eta_1$  and  $\eta_2$  determine the amplitude of the search space. The search space should not be too wide in order to facilitate its efficient exploration. Indeed, the search space is in practice split into  $N$  intervals (see Fig. 4). So that, the wider search space, the higher  $N$  in order to ensure an accurate estimation of the saddle point



**FIGURE 3.** Representation of the Exploitation search space in the vicinity of the approximated saddle truncation  $\hat{h}_k^L$ .



**FIGURE 4.** Schematic representation of the exploitation vector. Unlike to the exploration vector, the exploitation vector has  $N$  elements which are defined by the bounds  $\underline{v}$  and  $\bar{v}$ .

( $\tilde{h}^L$ ). This fact is relevant since this stage has a computational burden proportional to  $N$ . Hence, if  $N$  has to be high, the computational cost of this stage would also be high, affecting the performance of 4SA. Nevertheless, the parameter  $N$  might be selected high at first iterations, when the solution is presumably still far and the algorithm may potentially fail, and progressively decrease it as the algorithm satisfactory evolves (see Section IV). It is found that  $\eta_1 \sim 0.5$  and  $\eta_2 \sim 1.2$  work satisfactorily in most cases.

The entire process for defining the Exploitation search space is shown in Algorithm 3 (where the operator  $(a : \Delta : b)$  builds a vector whose first element is  $a$ , the last element is  $b$ , and the step is  $\Delta$ ). As a result, the exploitation vector is built and collects  $N$  different truncations  $v^j$  (see Fig. 4).

**Algorithm 3** Construction of the Exploitation Search Space

```

1: Let  $N_k, \eta_1$  and  $\eta_2$  be given
2: Calculate  $\underline{v} = \eta_1 \hat{h}_k^L$ 
3: Calculate  $\bar{v} = \eta_2 \hat{h}_k^L$ 
4: if  $\bar{v} > 1$  then
5:    $\bar{v} \leftarrow 1$ 
6: end if
7: Create  $v_k \leftarrow \underline{v} : (\underline{v} - \bar{v}) / N_k : \bar{v}$ 
8: return exploitation vector  $v_k \leftarrow [v_k^1, v_k^2, \dots, v_k^{N_k}]^T$ 

```

At the Exploitation stage, different intermediate state vectors are calculated using the truncations of the exploitation vector as:

$$y_k^j = x_k - v_k^j \phi_k \tag{7}$$

The Exploitation stage is carried out as follows. Up to  $N$  elements as (7) are calculated using the different truncations collected in the exploitation vector. Then, the accurate saddle truncation  $\tilde{h}^L$  is determined as that element of  $v$  which produces the vector (7) with the lowest residual. This process is summarized in Algorithm 4 using pseudocode.

**C. STAGE 3-PONDERATION**

As a result of the Exploitation stage, a second approximation to the saddle truncation  $\tilde{h}^L$  is obtained, which is presumably more accurate than  $\hat{h}^L$ . However, we cannot assert that this truncation exactly yields the saddle point since it is only

**Algorithm 4** Exploitation Stage

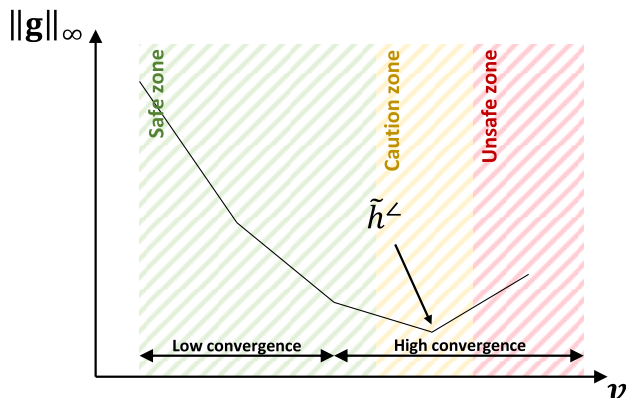
```

1: Let  $x_k, v_k$  and  $\phi_k$  be given
2: Initialize  $j \leftarrow 1$  and  $q \leftarrow 0$ 
3: for  $j = [1, 2, \dots, N_k]$  do
4:   Calculate  $y_k^j = x_k - v_k^j \phi_k$ 
5:   if  $j == 1$  then
6:      $q \leftarrow 1$ 
7:   else
8:     if  $\|g(y_k^j)\|_\infty < \|g(y_k^q)\|_\infty$  then
9:        $q \leftarrow j$ 
10:    end if
11:  end if
12: end do
13: return accurate saddle truncation  $\tilde{h}_k^\angle \leftarrow v_k^q$ 

```

obtained with very accurate Exploration and Exploitation stages ( $\xi \downarrow\downarrow$  or  $N \uparrow\uparrow$ ), which is unaffordable in practice. Therefore, the truncation obtained at Stage 2 should not be considered for modifying the state vector (2d) and directly updating it. This stage aims to determine which truncation is more secure for updating the state vector. If one considers all truncations up to the exact saddle point as safe and all evolution beyond it as unsafe, we can divide the exploitation vector into two zones. However, as commented, the exact saddle point is unknown. Therefore, it is not suitable to reduce all the deductions to only two zones; instead, consider three ones. Thus, a caution space is added between the safe and unsafe zones. This new zone lies around  $\tilde{h}_k^\angle$ , and it can be defined as these points that one cannot assert if the evolution is safe or not.

As commented, each truncation of  $v$  can be categorized as safe, caution or unsafe. Nevertheless, one should also put into value how each truncation affects the convergence rate of the developed 4SA. Thus, the lower truncations are associated with a low convergence rate, while the higher ones normally produce a faster evolution of the state vector (close to quadratic convergence). A priori, one shall consider the fastest evolution of the state vector for converging employing less iterations; however, as observed in Fig. 5, those



**FIGURE 5.** The exploitation vector as a function of the situation of  $\tilde{h}_k^\angle$ .

elements of the exploitation vector with higher convergence rate usually lie on the unsafe zone. As shown in this figure, the exploitation vector can also be divided as a function of the speed of convergence; thus, the lower truncations are associated with a low convergence rate while the higher ones normally produce a faster evolution of the state vector. To sum up, the different  $v^j$  can be categorized attending to different criteria. The main issue is that some ponderations offer good and bad properties at the same time. To solve this problem, the Ponderation stage aims at assigning a weight to each element of  $v$  following the rules collected in Table 1. Although these rules are based on heuristic foundations and, consequently, they are not universal, the adopted approaches are typically well-behaved and work quite well, as demonstrated in Section V. As observed, the assigned weights aim to balance between a safe and fast evolution of the state vector.

**TABLE 1.** Weights assigned to each  $v^j$  as a function of the zone and convergence speed.

Zone	Weight	
	High convergence	Low convergence
Safe	High	Medium
Caution	High	Medium
Unsafe	Low	Low

Following the guidelines of Table 1, the different weights are assigned using the following function:

$$w_k^j = e^{-(h_k^j - \mu) / 2\sigma} \tag{8}$$

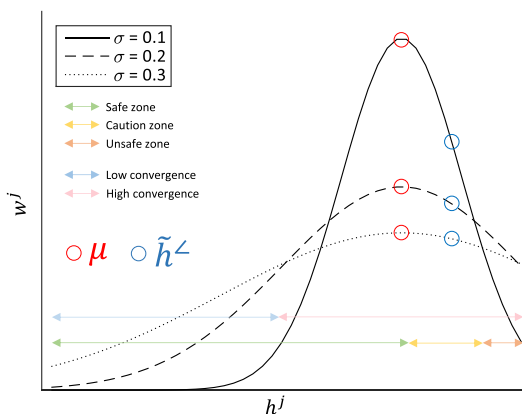
where,  $h^j$  is the  $j^{th}$  element of the so-called ponderation vector  $h$ ,  $\sigma \in \mathbb{R}^+$  is the standard deviation,  $e$  is the natural exponential and  $\mu = \eta_\mu \tilde{h}_k^\angle$  with  $\eta_\mu$  slightly lower than 1. The function (8) is used in this work since it naturally weights the different elements of the exploitation vector according to the rules presented in Table 1. Also, the function (8) compensates the heuristic foundations adopted in Table 1; hence those elements of the exploitation vector with presumably bad characteristics (for instance, points within the unsafe zone), are not nullified. Instead, the function (8) automatically assigns them a low weight to reduce their influence on updating the state vector. This approach also compensates for the intrinsic not-totally accurate estimation of the saddle point after carrying out the Exploration and Exploitation stages. Thus, even the impact on the selection of some arbitrary parameters ( $\eta_1, \eta_2, N$  and  $\eta_\mu$ ) is reduced in order to facilitate the implementation of the developed 4SA.

The main steps of the Ponderation stage, along the process for building the ponderation vector, are summarized in Algorithm 5 using pseudocode.

Fig. 6 plots the weights assigned by (8) as a function of  $\tilde{h}_k^\angle$  for different values of the parameter  $\sigma$ . As observed, the different weights are assigned according to the criteria collected in Table 1. It is worth noting that the highest weight is assigned to  $\mu$ , which is a few lower than  $\tilde{h}_k^\angle$ . Thus, the

**Algorithm 5** Ponderation Stage

- 1: Let  $N_k$ ,  $\tilde{h}_k^\angle$ ,  $\eta_\mu$  and  $\sigma$  be given
- 2: Create  $\mathbf{h}_k = [h_k^1, h_k^2, \dots, h_k^{N_k}]^T \leftarrow \tilde{h}_k^\angle/N_k : \tilde{h}_k^\angle/N_k : \tilde{h}_k^\angle$
- 3: Calculate  $\mu = \eta_\mu \tilde{h}_k^\angle$
- 4: Initialize  $j \leftarrow 1$
- 5: **for**  $j = [1, 2, \dots, N_k]$  **do**
- 6:     Calculate  $w_k^j = e^{-(h_k^j - \mu)/2\sigma}$
- 7: **end do**
- 8: Calculate  $\mathbf{w}_k \leftarrow \text{sum}(\mathbf{w}_k)^{-1} \mathbf{w}_k$
- 9: **return** weights vector  $\mathbf{w}_k \leftarrow [w_k^1, w_k^2, \dots, w_k^{N_k}]^T$


**FIGURE 6.** Weights assigned by function (8) for different  $\sigma$ .

highest weight is always assigned to  $\mu$ . As commented,  $\eta_\mu$  should be slightly lower than 1. Indeed, since the saddle point is never exactly determined, it is preferable to assign the highest weights to those points lower than  $\tilde{h}_k^\angle$ , since these values are presumably close to the safe zone and, consequently, algorithm's robustness is more guaranteed. The different zones are determined based on heuristic criteria; thus, the lower truncations lie in the safe, low convergence zone, while most of the higher truncations are in the unsafe high convergence area. It is worth noting that it is unsuitable to assign high weights to  $\tilde{h}_k^\angle$  since, as commented, one cannot know if this truncation has lied in the safe or unsafe zone.

It is also remarkable how the parameter  $\sigma$  affects to (8). Low values of  $\sigma$  provoke that (8) gives a very high weight to  $\mu$ , while the other truncations of the exploitation vector are nullify. In other words, if  $\sigma$  is fixed very low, it is assumed that  $\mu$  is the most suitable truncation for the Newton's increment vector, so this approach is not generally recommended since, as commented, the saddle point is never determined with total accuracy. Instead, it is preferable to assign a high value of  $\sigma$ . This approach assumes that the exploitation search space is not too wide, allowing a correct exploration of the Exploitation vector. In practice, it is observed that  $\sigma \sim 0.5 - 1$  works satisfactorily in most cases.

**D. STAGE 4-UPDATING**

Finally, the state vector (2d) is updated, taking into account the weights calculated at the previous stage. Thus, the considered most suitable truncation is given by:

$$\omega_k = \mathbf{w}_k^T \mathbf{h}_k \quad (9)$$

And the state vector is updated as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \omega_k \boldsymbol{\phi}_k \quad (10)$$

The main steps of Stage 4 are summarized in Algorithm 6 using pseudocode.

**Algorithm 6** Updating Stage

- 1: Let  $\mathbf{x}_k$ ,  $\boldsymbol{\phi}_k$ ,  $\mathbf{w}_k$  and  $\mathbf{h}_k$  be given
- 2: Calculate  $\omega_k = \mathbf{w}_k^T \mathbf{h}_k$
- 3: Calculate  $\mathbf{x}_{k+1} = \mathbf{x}_k - \omega_k \boldsymbol{\phi}_k$
- 4: **return** solution  $\mathbf{x}_{k+1}$

**E. COMPUTATIONAL BURDEN**

The main calculations of the developed 4SA are summarized in Table 2. The new proposal only requires a Jacobian evaluation and a LU factorization. In that sense, its computational burden is similar to NR. However, 4SA needs multiple function evaluations in the stages 1 and 2. Although this computation is generally cheap, it undoubtedly supposes an extra computational burden with respect to NR. In the following section, we propose a paradigm for avoiding these calculations.

**IV. IMPLEMENTATION OF THE DEVELOPED ALGORITHM**

As commented in the previous section, the developed 4SA is inevitably less efficient than NR. This disadvantage makes our proposal less attractive for solving well-conditioned systems, where NR undoubtedly offers better performance. To tackle this issue, the developed 4SA is integrated within a novel solution paradigm. This novel solution framework avoids the usage of 4SA in those cases easily solvable with NR. Thus, the developed 4SA is only used in ill-conditioned systems or when the convergence does not seem ensured. Now, the main question arises: how is it determined if 4SA is suitable or not? To answer this question, one can take advantage of the information brought by 4SA for employing the following rules:

- The developed 4SA can heuristically determine if the case is well or ill-conditioned using the jump parameter. It is worth noting that this parameter is, in fact, inversely proportional to the infinity norm of the Newton's increment vector (see eq. (6)). When a nonlinear system is ill-conditioned, some elements of the Newton's vector tend to be abnormally large due to the ill-conditioning of the Jacobian matrix. Therefore, it may be reasonable to assume that the degree of ill-conditioning of a system is inversely proportional to the value of  $\xi$ . Of course, the well-known condition number of the

TABLE 2. Main calculations of 4SA and their computational costs.

Case	NR	BT	HLM	RBS	BE	4SA	Developed Paradigm
case_ieee30 (default)	2.66 (2)	8.71 (6)	4.86 (3)	26.60 (10)	12.42 (9)	8.36 (4)	2.66 (2)
case118 (default)	4.92 (3)	13.89 (9)	8.48 (4)	41.47 (11)	19.36 (11)	13.40 (5)	4.92 (3)
case1354pegase (default)	34.36 (4)	128.14 (13)	79.75 (5)	306.60 (12)	131.28 (12)	64.70 (5)	34.36 (4)
case3012wp (default)	44.22 (2)	131.20 (6)	178.65 (5)	586.40 (10)	247.20 (10)	139.45 (5)	44.22 (2)
case3375wp (default)	74.22 (3)	308.02 (13)	243.18 (6)	789.36 (12)	327.48 (12)	154.50 (5)	74.22 (3)
case13659pegase (default)	497.25 (5)	1701.30 (15)	2654.08 (13)	3590.47 (13)	1281.84 (12)	688.80 (6)	497.25 (5)
case_ieee30 (flat)	3.98 (3)	9.60 (9)	8.10 (5)	29.27 (11)	15.23 (11)	10.47 (5)	3.98 (3)
case118 (flat)	5.36 (4)	19.12 (13)	9.60 (5)	35.68 (12)	15.74 (11)	11.53 (5)	6.56 (4)
case1354pegase (flat)	42.94 (5)	175.23 (18)	159.51 (10)	357.71 (14)	142.17 (13)	77.63 (6)	42.94 (5)
case3012wp (flat)	Diverge	398.26 (19)	500.17 (14)	820.90 (14)	321.41 (13)	195.24 (7)	132.64 (6)
case3375wp (flat)	Diverge	469.01 (20)	526.87 (13)	920.88 (14)	354.77 (13)	216.33 (7)	148.43 (6)
case13659pegase (flat)	Diverge	1913.54 (17)	6941.45 (34)	4142.83 (15)	1602.34 (15)*	803.60 (7)	696.13 (7)

\* Low Voltage solution

Jacobian matrix may also be used for this purpose [28]; however, the threshold considered for determining if a nonlinear system is ill-conditioned based on the condition number is strongly case-dependent. Indeed, it is observed that the condition number not only depends on the conditioning of the Jacobian matrix, and it is also affected by other characteristics; for instance, large-scale cases frequently present large condition numbers even if they are not ill-conditioned [28]. In addition, calculating the condition number may be computationally expensive. For these reasons, the jump parameter is preferred to estimate the degree of ill-conditioning of the problem and act accordingly. Thus, at the beginning of the algorithm, it can be determined if NR is suitable for solving the studied case or not. Based on our experience, most well-conditioned systems yield  $\xi_0 > 0.5$ ; hence this threshold can be considered for determining the usage of NR or 4SA.

- As 4SA evolves, the size of the exploitation and ponderation vectors can be reduced using the following rule:

if  $\|g(x_{k+1})\|_\infty < \|g(x_k)\|_\infty$  then  $N_{k+1} = \min(\text{round}(N_k/3), 1)$  end if (11)

where,  $\text{round}(\cdot)$  rounds to the nearest integer. As observed, the rule (11) is only activated if the residual is reduced at  $k + 1$ , so that the computational burden is reduced as less function evaluations are required during the process. When  $N = 1$ , the Exploration and Ponderation stages are immaterial and, consequently, NR can be used instead.

- If the result of both the Exploration and the Exploitation stages is equal to 1, one can guess that the Newton's evolution vector is safe and, consequently, NR can be reliably employed.

In conclusion, the developed paradigm consists of carrying out 4SA and switching to NR when some of the above conditions are matched. Furthermore, the flowchart of the developed power flow solution paradigm is shown in Fig. 7.

It is worth mentioning that a more straightforward approach may consist of simply running NR and 4SA simultaneously and, after convergence, take the more reliable solution. However, this procedure is not functional at all and

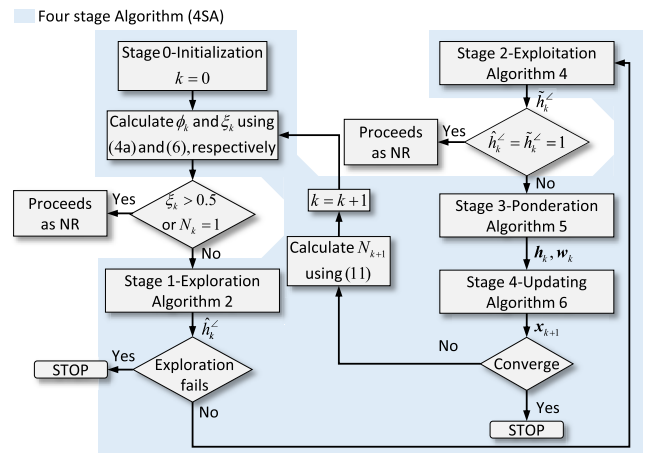


FIGURE 7. Flowchart of the developed power flow solution paradigm.

the developed algorithm would outperform this procedure. For example, in the case of ill-conditioned systems, NR just diverges so it is unnecessary to run this algorithm. In addition, simultaneously running both methods suppose an important computational burden that can be avoided by implementing the developed solution framework. On the other hand, it is difficult to know a priori if a power system is naturally ill or well-conditioned. In this sense, it is necessary an indicator that determines the degree of ill-conditioning of a system without needing human assistance. Thus, the algorithm automatically determines which method is more suitable for solving the network. In the developed framework, this issue is addressed by the parameter  $\xi$ .

V. NUMERICAL EXPERIMENTS

The developed 4SA and the solution paradigm, described in Section IV and summarized in the flowchart in Fig. 7, have been tested along NR, the conventional Backtracking line search technique (BT) (e.g. see [44]), the High order Levenberg-Marquardt method (HLM) [30], the Backward-Euler technique (BE) [22], and the Reverse Bulirsch-Stoer power flow solution method (RBS) [27]. For HLM, the so-called damping factor is updated each iteration by  $\|g\|_\infty^{1.3}$ .



**TABLE 3. Solution times [ms] and (total iterations) for the studied systems with base case conditions.**

Case	NR	BT	HLM	RBS	BE	4SA	Developed Paradigm
case_ieee30 (default)	2.66 (2)	8.71 (6)	4.86 (3)	26.60 (10)	12.42 (9)	8.36 (4)	2.66 (2)
case118 (default)	4.92 (3)	13.89 (9)	8.48 (4)	41.47 (11)	19.36 (11)	13.40 (5)	4.92 (3)
case1354pegase (default)	34.36 (4)	128.14 (13)	79.75 (5)	306.60 (12)	131.28 (12)	64.70 (5)	34.36 (4)
case3012wp (default)	44.22 (2)	131.20 (6)	178.65 (5)	586.40 (10)	247.20 (10)	139.45 (5)	44.22 (2)
case3375wp (default)	74.22 (3)	308.02 (13)	243.18 (6)	789.36 (12)	327.48 (12)	154.50 (5)	74.22 (3)
case13659pegase (default)	497.25 (5)	1701.30 (15)	2654.08 (13)	3590.47 (13)	1281.84 (12)	688.80 (6)	497.25 (5)
case_ieee30 (flat)	3.98 (3)	9.60 (9)	8.10 (5)	29.27 (11)	15.23 (11)	10.47 (5)	3.98 (3)
case118 (flat)	5.36 (4)	19.12 (13)	9.60 (5)	35.68 (12)	15.74 (11)	11.53 (5)	6.56 (4)
case1354pegase (flat)	42.94 (5)	175.23 (18)	159.51 (10)	357.71 (14)	142.17 (13)	77.63 (6)	42.94 (5)
case3012wp (flat)	Diverge	398.26 (19)	500.17 (14)	820.90 (14)	321.41 (13)	195.24 (7)	132.64 (6)
case3375wp (flat)	Diverge	469.01 (20)	526.87 (13)	920.88 (14)	354.77 (13)	216.33 (7)	148.43 (6)
case13659pegase (flat)	Diverge	1913.54 (17)	6941.45 (34)	4142.83 (15)	1602.34 (15)*	803.60 (7)	696.13 (7)

\* Low Voltage solution

Regarding BE, the implementation denoted by ICNM-J1 [22] has been tested. In the case of RBS, the guidelines provided in [27] have been followed. For our methods,  $N_0 = 20$ ,  $\eta_1 = 0.50$ ,  $\eta_2 = 1.20$ ,  $\eta_\mu = 0.75$  and  $\sigma = 1$  have been taken.

All tested techniques have been implemented within the Matpower environment [45]. Matpower provides an extensive Matlab-based library for power system analysis. This library fully exploits the Matlab capabilities for handling large sparse systems as those typically encountered in power system analysis. Hence, factorizations and linear systems solutions are carried out using sparse facilities such as the minimum degree ordering algorithm or partial pivoting. In addition, all the tested cases are available in Matpower database [46]–[48]. They have been denoted as they are labelled in this software. Two cases are contemplated; thus, the studied cases are solved from a flat start or the default starting point provided in Matpower.

All reported times have been obtained on a 64-bit i5-9400F Intel Core personal computer (2.90 GHz, 8 GB of RAM), as the average value of 100 simulations; on the other hand,  $\|g\|_\infty < 10^{-6}$  has been taken as a convergence criterion.

#### A. GENERAL COMPARISON FOR BASE CASES

Table 3 provides the results obtained for base cases. As seen, both 4SA and the developed paradigm successfully solved all studied systems. Both techniques were well-behaved generally for different cases. However, some methods outperformed 4SA in a few cases. This issue is solved by the developed solution paradigm, which offered the best trade-off between robustness and efficiency. The good convergence properties observed when using the developed techniques, are undoubtedly due to the accurate estimation of the most suitable truncation of the Newton's increment vector after carrying out the Exploration, Exploitation and Ponderation stages. Thus, the developed algorithms do not excessively deteriorate the quadratic convergence features of NR, as other algorithms do. For instance, RBS and BE truncate the Newton's increment vector using a so-called step size, which follows an algorithm-based update rule rather than observing the residuals' behaviour as 4SA and the developed solution

paradigm do. Therefore, the quadratic convergence behaviour of NR is usually preserved and the solution is typically reached in a reasonable number of iterations.

It is worth observing that the developed paradigm offered the same results compared with NR in well-conditioned cases. This result is logic since as commented in Section IV and described in Fig. 7, the developed solution algorithm proceeds as NR in well-conditioned systems. On the other hand, the developed solution framework is able to detect ill-conditioned systems and perform as 4SA in such cases until the convergence is ensured, then, the algorithm switches to NR to speed up the convergence.

The conventional NR showed good performance with the default initialization; however, it failed in some cases when a flat start was used. When converged, NR was along the developed solution paradigm, the most competitive technique. This is due to the proposed solution framework was able to identify ill-conditioned cases accurately. According to the definition given in the Introduction and [7], those systems where NR failed from a flat start can be categorized as ill-conditioned. Such is the case of 'case3012wp', 'case3375wp' and 'case13659pegase', as seen in Table 3. As shown in Table 4, the value of  $\xi_0$  allows to discriminate between well and ill-conditioned cases; thereby, the value of this variable is quite short in these three cases. Thus, the developed solution paradigm performed as NR in those well-conditioned cases, saving iterations and solution time. It is also worth noting that the value of  $\xi_0$  abruptly varies from well to ill-conditioned cases; hence, the threshold adopted in this paper ( $\xi_0 > 0.5$ ), which was taken based on personal experience, has not to be defined finely (note that one could

**TABLE 4. The classification of each case as a function of the Jump parameter at the first iteration.**

Case	Default		Flat	
case_ieee30	$\xi_0 = 1$	Well	$\xi_0 = 1$	Well
case118	$\xi_0 = 1$	Well	$\xi_0 = 1$	Well
case1354pegase	$\xi_0 = 1$	Well	$\xi_0 = 1$	Well
case3012wp	$\xi_0 = 1$	Well	$\xi_0 = 0.16$	Ill
case3375wp	$\xi_0 = 1$	Well	$\xi_0 = 0.16$	Ill
case13659pegase	$\xi_0 = 0.97$	Well	$\xi_0 = 0.06$	Ill

take  $\xi_0 > 0.8$  instead and the result would be the same). In other words, this threshold has not to be selected carefully; actually, it could take different values, and the degree of ill-conditioning would still be correctly estimated. In the light of these results, it is demonstrated that the value of  $\xi_0$  supposes a good indicator to determine the degree of ill-conditioning of a system; in addition, unlike the condition number, the jump parameter seems case-independent, which enables its broad utilization.

The considered line search method (BT), successfully solved all the studied cases. However, one can see a clear deterioration of the convergence properties in some cases with respect to NR. Typically, this technique required a large number of iterations to find the solution; this is especially remarkable when a flat start is used, which makes this technique few practical in ill-conditioned cases.

HLM was quite robust. However, its efficiency was deteriorated as the size of the system grows. This is due to it requires a matrix-matrix product for each iteration. This operation is much costly in large-scale systems. In addition, it shows a low convergence rate when the starting point lies far away from the actual solution. Thus, it generally required many iterations for converging from a flat start.

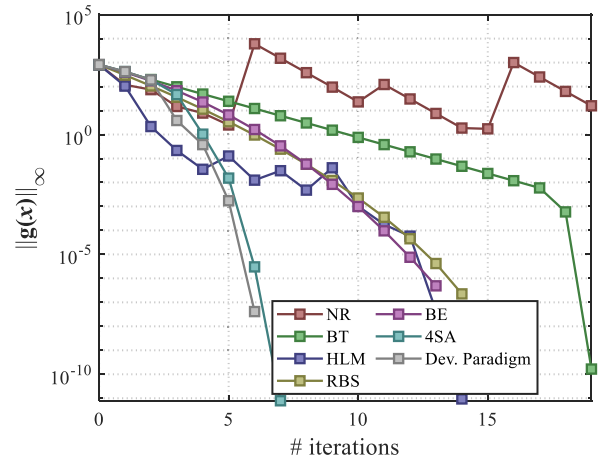
RBS is a reliable method but inefficient due to the enormous amounts of LU factorizations required. Following the guidelines presented in [27], this technique requires six factorizations at the first iteration, four at the second iteration and 2 for the remaining iterations. In addition, linear convergence is observed during the whole iterative procedure due to the influence of the step, which barely reaches its optimal value for the last iterations.

BE under its implementation, namely ICNM-J1 in [22], is quite efficient as only a LU factorization is required, and the calculation of the Hessian matrix is avoided. As RBS, this technique frequently shows a very linear convergence pattern. In addition, it occasionally converged to the low voltage solution while the developed techniques always converged to the stable operating point.

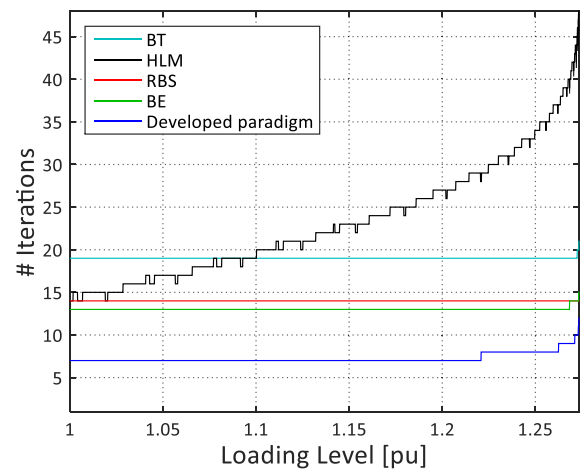
Figure 8 provides a better overview of the convergence properties of the different studied solvers. More precisely, this figure plots the convergence profiles for the ‘case3012wp’ with flat initialization. As seen in this figure, while NR shows an oscillatory pattern which indicates divergence, remainder solvers successfully converged. The developed solution paradigm and 4SA performs similarly during the first iterations. This is due to the developed paradigm is able to identify this case as ill-conditioned and, consequently, 4SA is run at the beginning of the iterative procedure. After a few iterations, the developed solution framework determined that the convergence is ensured and switched to the conventional NR, thus speeding up the convergence.

**B. INFLUENCE OF THE LOADING LEVEL**

Fig. 9 plots the total iterations for different loading levels for the ‘case3012wp’ from a flat start (since 4SA and the developed paradigm showed the same number of iterations



**FIGURE 8.** Convergence profiles for the ‘case3012wp’ from a flat start.



**FIGURE 9.** Total iterations for different loading levels for the ‘case3012wp’ from a flat start.

in all cases, only the latter has been reported in this figure). Here, the loading level has been modified according to:

$$P_i^{sp} = \lambda P_i^{sp}, \quad \text{for all buses} \quad (12a)$$

$$Q_i^{sp} = \lambda Q_i^{sp}, \quad \text{for PQ buses} \quad (12b)$$

where  $\lambda \in \mathbb{R}^+$  is the loading level. As seen from Fig. 8, the performance of HLM gets worse as the loading level grows, employing too many iterations close to the maximum loadability point. BT, RBS and BE are barely affected by the loading level and their convergence rates stay almost constant. However, the developed methods were the most competitive, employing fewer iterations than the remainder techniques.

**C. VALIDATION WITH REACTIVE LIMITS ENABLED AT PV BUSES**

Although the developed formulation of 4SA does not take into account the equipment limits, they can be easily incorporated using any mechanism available for NR. Let us here consider that switch logic strategy described in [49], for

**TABLE 5. Solution times [ms] and (total iterations) for the studied systems with reactive limits.**

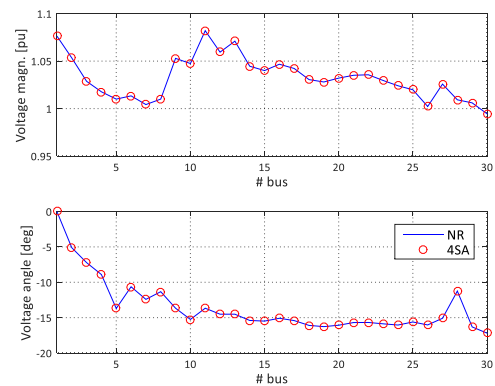
Case	NR	BT	HLM	RBS	BE	4SA	Developed Paradigm
case_jeec30 (default)	5.83 (5)	15.95 (13)	11.82 (6)	47.58 (20)	18.62 (19)	15.23 (9)	5.83 (5)
case118 (default)	7.66 (5)	25.98 (17)	17.22 (7)	75.63 (21)	28.61 (21)	24.38 (10)	7.66 (5)
case1354pegase (default)	75.73 (9)	298.10 (30)	211.25 (13)	880.78 (34)	266.65 (33)	154.30 (15)	75.73 (9)
case3012wp (default)	117.43 (6)	351.81 (16)	399.18 (11)	1807.52 (29)	526.70 (29)	296.72 (13)	117.43 (6)
case3375wp (default)	172.30 (8)	535.95 (22)	590.56 (14)	2522.60 (36)	728.70 (36)	429.30 (17)	172.30 (8)
case13659pegase (default)	590.84 (7)	2715.20 (24)	3544.51 (17)	6777.86 (24)	1885.11 (23)	1031.22 (11)	590.84 (7)
case_jeec30 (flat)	6.24 (6)	18.33 (16)	13.65 (8)	50.42 (21)	20.16 (21)	16.24 (10)	6.24 (6)
case118 (flat)	8.59 (6)	31.30 (21)	21.75 (10)	80.28 (22)	30.23 (22)	24.38 (10)	8.59 (6)
case1354pegase (flat)	83.22 (10)	311.44 (35)	285.72 (18)	923.47 (36)	272.64 (34)	162.40 (16)	83.22 (10)
case3012wp (flat)	Diverge	619.28 (29)	740.80 (20)	2001.50 (33)	624.36 (32)	349.11 (15)	224.00 (10)
case3375wp (flat)	Diverge	702.27 (29)	878.60 (21)	2633.15 (38)	766.44 (37)	484.61 (19)	272.49 (11)
case13659pegase (flat)	Diverge	2941.20 (26)	7817.85 (38)	7250.63 (26)	2218.24 (27)*	1145.13 (12)	877.50 (9)

\* Low Voltage solution

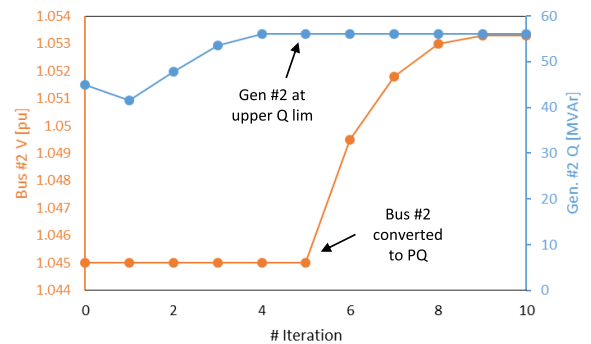
incorporating the generators’ reactive power limits within the power flow calculation using NR. This strategy basically consists of converting those PV buses with any limit violation to PQ buses, taking the hit limit as injected reactive power. In these converted buses, the injected reactive power is now taken as an independent variable. The nodal voltage freely varies; then, starting from the calculated solution without reactive violations, the process is repeatedly carried out until a feasible solution is reached. Otherwise, the case is declared unfeasible if a feasible solution has not been found with all the PV buses converted to PQ. It is worth noting that this strategy is quite universal, and other references and software have adopted similar approaches (e.g. see [7], [45] or [50]). Nevertheless, the developed solution algorithms are versatile enough to incorporate other more sophisticated approaches such as those described in [1] and [51].

The aim of this section is to simply validate the developed power flow solution techniques when some functional limits are taken into account; hence the strategy described in [49] and above described has been implemented within 4SA and the developed paradigm. Fig. 10 shows the nodal voltage angles and magnitudes for the ‘case30’ with reactive limits obtained by 4SA and NR from a flat start (since the developed paradigm performs in this case as NR, their results are not shown). As observed, 4SA obtained very accurate results. Fig. 11 illustrates the implemented bus switching mechanism. As shown, the injected reactive power and voltage magnitude at bus #2 during execution of 4SA. As observed, the reactive power grows to keep the nodal voltage magnitude constant until its limit is reached. After that, this bus is converted to PQ type. Then, the reactive power is kept constant, and the nodal voltage magnitude varies until a feasible solution is reached. It is worth noting that this same strategy can be considered for taking into account other controls or functional limits like transformers tap changers.

It is well known that incorporating an outer switching mechanism to consider the equipment limits may deteriorate the convergence properties of a PF solver; this is because more PQ buses and, consequently, variables and equations are incorporated into the system each iteration of the outer loop. This is illustrated in Fig. 12, where

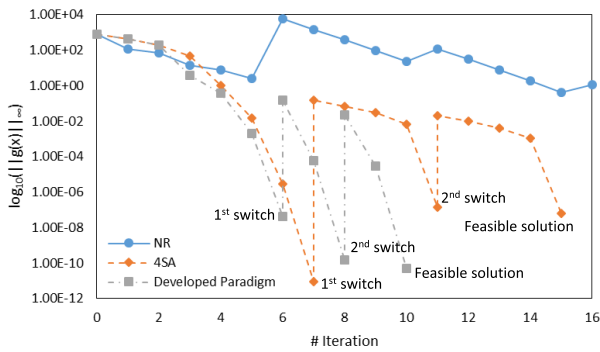


**FIGURE 10. Comparison of the nodal voltage magnitudes and angles obtained by 4SA and NR for the ‘case30’ from a flat start with reactive limits enabled.**



**FIGURE 11. Evolution of the reactive power injected at bus #2 for the ‘case30’ and its nodal voltage magnitude with reactive limits enabled using 4SA and the developed solution paradigm from a flat start.**

the convergence profiles of NR, 4SA and the developed paradigm for the ‘case3012wp’ from a flat start are plotted. As observed, NR failed to solve the base case because it oscillates during a large number of iterations. In this case, the developed paradigm and 4SA performed well in case of considering the generators reactive limits. As seen, the proposed methods successfully follow the switching mechanism to a feasible solution after the solution of 3 power flow problems.



**FIGURE 12.** Convergence profiles for the ‘case3012wp’ with reactive limits from a flat start.

In order to fully show how the developed techniques perform in the presence of reactive limits, we have compared them with the other considered solvers. In all cases, the switching strategy previously described has been incorporated within the different studied PF techniques. Table 5 compares the different studied solvers to consider the generators’ reactive limits. As seen for the base cases, the developed techniques were well-behaved and showed very competitive results.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, a novel power solution method has been proposed to solve well and ill-conditioned cases. It is carried out through four stages, devoted to determining the optimal evolution of the Newton’s increment vector. Although the developed method is more efficient than most available robust solvers, the developed 4SA is still not competitive with NR in well-conditioned cases. This disadvantage would limit the potential application of the developed technique in commercial packages. To overcome this drawback, 4SA has been integrated within an efficient paradigm, which can guess if the studied case is well or ill-conditioned. In the former case, NR is used while 4SA is employed for solving the ill-conditioned cases. The developed paradigm also switches to NR when the convergence seems ensured.

Various numerical results have been provided in a variety of cases and scenarios. The results showed that 4SA is more competitive than most robust, tested techniques; however, it is still less efficient than NR in most cases. The integration of 4SA within the developed framework brought superior characteristics. Thus, the developed solution paradigm was generally the most competitive method, offering the best trade-off between robustness and efficiency.

Despite the promising results obtained, the developed solution framework still has some limitations as it fails at turning points and unsolvable cases. Future works should be devoted to expanding the applicability of our proposal to these cases.

## REFERENCES

- [1] N. Xie, F. Torelli, E. Bompard, and A. Vaccaro, “Dynamic computing paradigm for comprehensive power flow analysis,” *IET Gener., Transmiss. Distrib.*, vol. 7, no. 8, pp. 832–842, Aug. 2013, doi: [10.1049/iet-gtd.2012.0350](https://doi.org/10.1049/iet-gtd.2012.0350).
- [2] K. Tang, S. Dong, J. Shen, C. Zhu, and Y. Song, “A robust and efficient two-stage algorithm for power flow calculation of large-scale systems,” *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5012–5022, Nov. 2019, doi: [10.1109/TPWRS.2019.2914431](https://doi.org/10.1109/TPWRS.2019.2914431).
- [3] H. Sheng and H.-D. Chiang, “CDFLOW: A practical tool for tracing stationary behaviors of general distribution networks,” *IEEE Trans. Power Syst.*, vol. 29, no. 3, pp. 1365–1371, May 2014, doi: [10.1109/TPWRS.2013.2289917](https://doi.org/10.1109/TPWRS.2013.2289917).
- [4] Z. Yan and Y. Xu, “Real-time optimal power flow: A Lagrangian based deep reinforcement learning approach,” *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3270–3273, Jul. 2020, doi: [10.1109/TPWRS.2020.2987292](https://doi.org/10.1109/TPWRS.2020.2987292).
- [5] Y. Chen, Z. Zhang, Y. Lang, J. Ma, and S. Zheng, “Generalised-fast decoupled state estimator,” *IET Gener., Transmiss. Distrib.*, vol. 12, no. 22, pp. 5928–5938, May 2018, doi: [10.1049/iet-gtd.2018.5076](https://doi.org/10.1049/iet-gtd.2018.5076).
- [6] J. B. Ward and H. W. Hale, “Digital computer solution of power-flow problems [includes discussion],” *Trans. Amer. Inst. Electr. Eng. III, Power App. Syst.*, vol. 75, no. 3, pp. 398–404, Jan. 1956, doi: [10.1109/AIEEPPAS.1956.4499318](https://doi.org/10.1109/AIEEPPAS.1956.4499318).
- [7] F. Milano, “Continuous Newton’s method for power flow analysis,” *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 50–57, Feb. 2009, doi: [10.1109/TPWRS.2008.2004820](https://doi.org/10.1109/TPWRS.2008.2004820).
- [8] L. Wang and X. R. Lin, “Robust fast decoupled power flow,” *IEEE Trans. Power Syst.*, vol. 15, no. 1, pp. 208–215, Feb. 2000, doi: [10.1109/59.852123](https://doi.org/10.1109/59.852123).
- [9] J. Yang, N. Zhang, C. Kang, and Q. Xia, “A state-independent linear power flow model with accurate estimation of voltage magnitude,” *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3607–3617, Sep. 2017, doi: [10.1109/TPWRS.2016.2638923](https://doi.org/10.1109/TPWRS.2016.2638923).
- [10] K. Murari and N. P. Padhy, “A network-topology-based approach for the load-flow solution of AC–DC distribution system with distributed generations,” *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1508–1520, Mar. 2019, doi: [10.1109/TII.2018.2852714](https://doi.org/10.1109/TII.2018.2852714).
- [11] M. Tostado-Veliz, S. Kamel, and F. Jurado, “A novel family of efficient power-flow methods with high convergence rate suitable for large realistic power systems,” *IEEE Syst. J.*, vol. 15, no. 1, pp. 738–746, Mar. 2021, doi: [10.1109/JSYST.2020.2980156](https://doi.org/10.1109/JSYST.2020.2980156).
- [12] M. Tostado-Veliz, S. Kamel, T. Alquthami, and F. Jurado, “A three-stage algorithm based on a semi-implicit approach for solving the power-flow in realistic large-scale ill-conditioned systems,” *IEEE Access*, vol. 8, pp. 35299–35307, 2020, doi: [10.1109/ACCESS.2020.2975058](https://doi.org/10.1109/ACCESS.2020.2975058).
- [13] T. J. Overbye, “A power flow measure for unsolvable cases,” *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1359–1365, Aug. 1994, doi: [10.1109/59.336130](https://doi.org/10.1109/59.336130).
- [14] T. J. Overbye, “Computation of a practical method to restore power flow solvability,” *IEEE Trans. Power Syst.*, vol. 10, no. 1, pp. 280–287, Feb. 1995, doi: [10.1109/59.373951](https://doi.org/10.1109/59.373951).
- [15] S. Iwamoto and Y. Tamura, “A fast load flow method retaining nonlinearity,” *IEEE Trans. Power App. Syst.*, vol. PAS-97, no. 5, pp. 1586–1599, Sep. 1978, doi: [10.1109/TPAS.1978.354650](https://doi.org/10.1109/TPAS.1978.354650).
- [16] S. Iwamoto and Y. Tamura, “A load flow calculation method for ill-conditioned power systems,” *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 4, pp. 1736–1743, Apr. 1981, doi: [10.1109/TPAS.1981.316511](https://doi.org/10.1109/TPAS.1981.316511).
- [17] F. Milano, *Power System Modelling and Scripting*. New York, NY, USA: Springer, 2010.
- [18] J. E. Tate and T. J. Overbye, “A comparison of the optimal multiplier in polar and rectangular coordinates,” *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1667–1674, Nov. 2005, doi: [10.1109/TPWRS.2005.857388](https://doi.org/10.1109/TPWRS.2005.857388).
- [19] R. Idema and D. J. P. Lahaye, “Computational methods in power system analysis,” in *Atlantis Studies in Scientific Computing in Electromagnetics*, W. Schilders, Ed. Paris, France: Atlantis Press, 2014.
- [20] T. D. Niri, M. Heydari, and M. M. Hosseini, “Two nonmonotone trust region algorithms based on an improved Newton method,” *J. Appl. Math. Comput.*, vol. 64, nos. 1–2, pp. 179–194, Apr. 2020, doi: [10.1007/s12190-020-01350-7](https://doi.org/10.1007/s12190-020-01350-7).
- [21] A. Hosseini Dehmiry, “The global convergence of the BFGS method under a modified Yuan-Wei-Lu line search technique,” *Numer. Algorithms*, vol. 84, no. 2, pp. 781–793, Jun. 2020, doi: [10.1007/s11075-019-00779-7](https://doi.org/10.1007/s11075-019-00779-7).

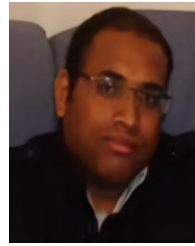
- [22] F. Milano, "Implicit continuous Newton method for power flow analysis," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3309–3311, Jul. 2019, doi: [10.1109/TPWRS.2019.2912485](https://doi.org/10.1109/TPWRS.2019.2912485).
- [23] R. Rianza and P. J. Zufiria, "Weak singularities and the continuous Newton method," *J. Math. Anal. Appl.*, vol. 236, no. 2, pp. 438–462, Aug. 1999, doi: [10.1006/jmaa.1999.6453](https://doi.org/10.1006/jmaa.1999.6453).
- [24] J. W. Neuberger, "Continuous Newton's method," in *Sobolev Gradients and Differential Equations* (Lecture Notes in Mathematics), vol. 1670, J. W. Neuberger Ed. Berlin, Germany: Springer, 2010.
- [25] J. M. Gutiérrez and M. Á. Hernández-Verón, "An acceleration of the continuous Newton's method," *J. Comput. Appl. Math.*, vol. 354, pp. 213–220, Jul. 2019, doi: [10.1016/j.cam.2018.03.013](https://doi.org/10.1016/j.cam.2018.03.013).
- [26] M. Tostado-Véliz, S. Kamel, and F. Jurado, "Development of different load flow methods for solving large-scale ill-conditioned systems," *Int. Trans. Electr. Energy Syst.*, vol. 29, no. 4, Apr. 2019, Art. no. e2784, doi: [10.1002/etep.2784](https://doi.org/10.1002/etep.2784).
- [27] M. Tostado-Véliz, S. Kamel, and F. Jurado, "A robust power flow algorithm based on Bulirsch–Stoer method," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 3081–3089, Jul. 2019, doi: [10.1109/TPWRS.2019.2900513](https://doi.org/10.1109/TPWRS.2019.2900513).
- [28] M. Tostado-Véliz, S. Kamel, and F. Jurado, "Comparison of various robust and efficient load-flow techniques based on Runge–Kutta formulas," *Electr. Power Syst. Res.*, vol. 174, Sep. 2019, Art. no. 105881, doi: [10.1016/j.epsr.2019.105881](https://doi.org/10.1016/j.epsr.2019.105881).
- [29] F. Milano, "Analogy and convergence of Levenberg's and Lyapunov-based methods for power flow analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1663–1664, Mar. 2016, doi: [10.1109/TPWRS.2015.2415455](https://doi.org/10.1109/TPWRS.2015.2415455).
- [30] R. Pourbagher and S. Y. Derakhshandeh, "Application of high-order Levenberg–Marquardt method for solving the power flow problem in the ill-conditioned systems," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 12, pp. 3017–3022, Sep. 2016, doi: [10.1049/iet-gtd.2016.0064](https://doi.org/10.1049/iet-gtd.2016.0064).
- [31] R. Pires, L. Mili, and G. Chagas, "Robust complex-valued Levenberg–Marquardt algorithm as applied to power flow analysis," *Int. J. Electr. Power Energy Syst.*, vol. 113, pp. 383–392, Dec. 2019, doi: [10.1016/j.ijepes.2019.05.032](https://doi.org/10.1016/j.ijepes.2019.05.032).
- [32] K. Amini, F. Rostami, and G. Caristi, "An efficient Levenberg–Marquardt method with a new LM parameter for systems of nonlinear equations," *Optimization*, vol. 67, no. 5, pp. 637–650, Feb. 2018, doi: [10.1080/02331934.2018.1435655](https://doi.org/10.1080/02331934.2018.1435655).
- [33] B. Huang and C. Ma, "A shamanskii-like self-adaptive Levenberg–Marquardt method for nonlinear equations," *Comput. Math. Appl.*, vol. 77, no. 2, pp. 357–373, Jan. 2019, doi: [10.1016/j.camwa.2018.09.039](https://doi.org/10.1016/j.camwa.2018.09.039).
- [34] N. Xie, E. Bompard, R. Napoli, and F. Torelli, "Widely convergent method for finding solutions of simultaneous nonlinear equations," *Electr. Power Syst. Res.*, vol. 83, no. 1, pp. 9–18, Feb. 2012, doi: [10.1016/j.epsr.2011.09.002](https://doi.org/10.1016/j.epsr.2011.09.002).
- [35] F. Torelli and A. Vaccaro, "A second order dynamic power flow model," *Electr. Power Syst. Res.*, vol. 126, pp. 12–20, Sep. 2015, doi: [10.1016/j.epsr.2015.04.014](https://doi.org/10.1016/j.epsr.2015.04.014).
- [36] W. H. Kersting, *Distribution System Modeling and Analysis*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2006.
- [37] F. Hameed, M. Al Hosani, and H. H. Zeineldin, "A modified backward/forward sweep load flow method for islanded radial microgrids," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 910–918, Jan. 2019, doi: [10.1109/TSG.2017.2754551](https://doi.org/10.1109/TSG.2017.2754551).
- [38] M. Tostado-Véliz, S. Kamel, and F. Jurado, "Two efficient and reliable power-flow methods with seventh order of convergence," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1026–1035, Mar. 2021, doi: [10.1109/JSYST.2020.3004667](https://doi.org/10.1109/JSYST.2020.3004667).
- [39] R. J. R. Kumar and A. Jain, "A load flow algorithm for radial systems having renewable energy generation sources," in *Proc. 7th Int. Conf. Inf. Autom. Sustainability*, Colombo, Lanka, Dec. 2014, pp. 1–5, doi: [10.1109/ICIAFS.2014.7069610](https://doi.org/10.1109/ICIAFS.2014.7069610).
- [40] M. Bazrafshan and N. Gatsis, "Convergence of the Z-bus method for three-phase distribution load-flow with ZIP loads," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 153–165, Jan. 2018, doi: [10.1109/TPWRS.2017.2703835](https://doi.org/10.1109/TPWRS.2017.2703835).
- [41] R. Verma and V. Sarkar, "Application of modified gauss-Z iterations for solving the load flow problem in active distribution networks," *Electr. Power Syst. Res.*, vol. 168, pp. 8–19, Mar. 2019, doi: [10.1016/j.epsr.2018.11.005](https://doi.org/10.1016/j.epsr.2018.11.005).
- [42] N. Yang, "Three-phase power flow calculations by direct Z-LOOP method for microgrids with electric vehicle charging demands," *IET Gener., Transmiss. Distrib.*, vol. 7, no. 9, pp. 1002–1010, Sep. 2013, doi: [10.1049/iet-gtd.2012.0535](https://doi.org/10.1049/iet-gtd.2012.0535).
- [43] T. Q. Zhao, H. D. Chiang, and K. Koyanagi, "Convergence analysis of implicit Z-bus power flow method for general distribution networks with distributed generators," *IET Gener., Transmiss. Distrib.*, vol. 10, no. 2, pp. 412–420, 2016, doi: [10.1049/iet-gtd.2015.0679](https://doi.org/10.1049/iet-gtd.2015.0679).
- [44] D. Zhu, "Nonmonotone backtracking inexact quasi-Newton algorithms for solving smooth nonlinear equations," *Appl. Math. Comput.*, vol. 161, no. 3, pp. 875–895, Feb. 2005, doi: [10.1016/j.amc.2003.12.074](https://doi.org/10.1016/j.amc.2003.12.074).
- [45] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011, doi: [10.1109/TPWRS.2010.2051168](https://doi.org/10.1109/TPWRS.2010.2051168).
- [46] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, Jul. 2017, doi: [10.1109/TPWRS.2016.2616385](https://doi.org/10.1109/TPWRS.2016.2616385).
- [47] C. Jozs, S. Fliscounakis, J. Maeght, and P. Panciatici, "AC power flow data in MATPOWER and QCQP format: ITesla, RTE snapshots, and PEGASE," 2016, *arXiv:1603.01533*. [Online]. Available: <http://arxiv.org/abs/1603.01533>
- [48] S. Fliscounakis, P. Panciatici, F. Capitanescu, and L. Wehenkel, "Contingency ranking with respect to overloads in very large power systems taking into account uncertainty, preventive, and corrective actions," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4909–4917, Nov. 2013, doi: [10.1109/TPWRS.2013.2251015](https://doi.org/10.1109/TPWRS.2013.2251015).
- [49] M. Tostado-Véliz, S. Kamel, and F. Jurado, "Robust and efficient approach based on Richardson extrapolation for solving badly initialised/ill-conditioned power-flow problems," *IET Gener., Transmiss. Distribution*, vol. 13, no. 16, pp. 3524–3533, Aug. 2019, doi: [10.1049/iet-gtd.2018.6786](https://doi.org/10.1049/iet-gtd.2018.6786).
- [50] P. Zhu, G. Taylor, and M. Irving, "Performance analysis of a novel Q-limit guided continuation power flow method," *IET Gener., Transmiss. Distrib.*, vol. 3, no. 12, pp. 1042–1051, Dec. 2009, doi: [10.1049/iet-gtd.2008.0504](https://doi.org/10.1049/iet-gtd.2008.0504).
- [51] G. Diaz and C. Gonzalez-Moran, "Fischer-Burmeister-based method for calculating equilibrium points of droop-regulated microgrids," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 959–967, May 2012, doi: [10.1109/TPWRS.2011.2175754](https://doi.org/10.1109/TPWRS.2011.2175754).



**MARCOS TOSTADO-VÉLIZ** was born in Spain, in 1987. He received the Electrical Engineering (Hons.) and master's degrees from the University of Seville, Seville, Spain, in 2016 and 2017, respectively, and the Ph.D. degree from the University of Jaén, Jaén, Spain, in 2020. His primary research interests include the application of numerical methods for power system analysis, multienergy systems modeling, home energy management systems, and microgrids control and optimization.



**TALAL ALHARBI** (Member, IEEE) received the B.Sc. degree in electrical engineering from Qassim University, Saudi Arabia, in 2010, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2015 and 2020, respectively. In 2020, he rejoined the Department of Electrical Engineering, Qassim University, where he is currently an Assistant Professor. His research interests include electric vehicle, power system operation and optimization, energy management, energy storage systems, power electronic and artificial intelligence applications in power systems, and smart grids.



**SALAH KAMEL** received the dual Ph.D. degree (international) from the University of Jaen, Spain (Main), and Aalborg University, Denmark (Host), in January 2014. He is currently an Associate Professor with the Electrical Engineering Department, Aswan University. He is also a Leader of the Power Systems Research Group, Advanced Power Systems Research Laboratory (APSR), Aswan, Egypt. His research interests include power system analysis and optimization, smart grid, and renewable energy systems.



**OMAR ALRUMAYH** (Member, IEEE) received the B.Sc. degree in electrical engineering from Qassim University, Saudi Arabia, in 2011, and the M.A.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, ON, Canada, in 2016 and 2021, respectively. In 2021, he joined the Department of Electrical Engineering, Qassim University, where he is currently an Assistant Professor. His research interests include power system operations, energy management, energy storage systems, and smart grids.



**FRANCISCO JURADO** (Senior Member, IEEE) was born in Linares, Jaén, Spain. He received the M.Sc. and Dr.Ing. degrees from the National University of Distance Education, Madrid, Spain, in 1995 and 1999, respectively. Since 1985, he has been a Professor with the Department of Electrical Engineering, University of Jaén, Jaén. His current research interests include power systems, modeling, and renewable energy.

...