A Robust and Efficient Two-Stage Algorithm for Power Flow Calculation of Large-Scale Systems

Kunjie Tang, *Student Member, IEEE*, Shufeng Dong, Jie Shen, Chengzhi Zhu, and Yonghua Song, *Fellow, IEEE*

***Abstract*—To satisfy the robust and real-time requirements of power flow calculation for large-scale power systems, a globally convergent method is proposed with trust-region techniques, which shows satisfying robustness and convergence. Then, this method is combined with Newton's method to achieve a two-stage algorithm, benefiting from their different advantages. In the first stage, the proposed globally convergent method is used for searching power flow solution. When the values of state variables in a certain iteration are close enough to the real operational point, the algorithm enters the second stage to use Newton's method to achieve the solution. This two-stage algorithm can achieve an accurate solution for solvable cases, and can also achieve a least-square solution, which is an approximate solution for unsolvable cases. Numerical experiments show that the proposed globally convergent method and two-stage algorithm have better robustness and efficiency than the existing methods in previous research. They have universality for well- and ill-conditioned systems as well as the cases under ill-conditioned operational modes, heavy loads and inappropriate initial values. They can also handle different limit violations, such as reactive power limit in PV buses and voltage limit with tap changers action, which could significantly benefit real practice.**

***Index Terms*—Ill-conditioned, large-scale, power flow calculation, robustness, trust-region. two-stage algorithm.**

I. Introduction

POWER flow calculation (PFC) is the backbone of power system analysis, which obtains the steady state voltage magnitude and angle of buses and power flow of branches given the generator outputs and loads. In essence, PFC is a nonlinear equation solving process. Usually, power flow problems can be classified into three categories [1]: i) well-conditioned case, of which traditional PFC method, such as Newton's method, can converge to the accurate solution under flat start; ii) ill-conditioned but solvable case, of which traditional PFC method will diverge if the initial values are far from the real operational point; iii) unsolvable case, of which the power flow solution does not exist.

With the expansion and increasing complexity of power systems, ill-conditioned cases and unsolvable cases are more common, and PFC is faced with challenges of convergence and computational efficiency. Traditional methods often diverge for some systems and operational modes, such as heavy loads, branch fault, etc. An ideal PFC method should have a satisfying performance for all the three categories of cases: i) high efficiency for solving well-conditioned cases; ii) powerful robustness and convergence for ill-conditioned but solvable cases; iii) provides a solution close to the feasibility boundary rather than diverge for unsolvable cases. However, for the most time, convergence and efficiency are contradictories. Thus, it is necessary to achieve a balance between convergence and efficiency, introducing a method with global convergence properties and high computational efficiency to satisfy real-time requirements of large-scale power systems.

Newton's method is one of the common methods for well-conditioned cases [2]. It is an iterative algorithm with a quadratic convergence. It has a satisfying performance for most well-conditioned systems. However, its convergence is sensitive to initial values, which leads to divergence under some ill-conditioned systems and inappropriate initial values far from the real operational point. Some research proposes several Newton-like methods to increase the convergence rate and reduce the time consumption [3]–[6], but their effectiveness is still limited to well-conditioned systems, and often diverge when they are used to solve ill-conditioned systems [7].

Many research achievements focus on PFC of the ill-conditioned systems. Iwamoto's method, based on second-order Taylor series, is a benchmark for PFC of the ill-conditioned systems [8], but its convergence is poor compared with later research. Reference [9]–[11] have presented different approaches based on the Levenberg-Marquardt (L-M) method to solve ill-conditioned systems, and [7], [12]–[14] further present several modified approaches based on high-order L-M method. These approaches based on L-M method have powerful robustness because their essence is an optimization problem and has satisfying global convergence. However, they usually require a larger amount of computation in each iteration and need more iterations than Newton's method to reach the same accuracy requirements,

Manuscript received January 4, 2019; revised April 3, 2019; accepted April 28, 2019. Date of publication May 2, 2019; date of current version October 24, 2019. This work was supported by the Science and Technology Program of State Grid Corporation of China under Grant 52110418000N. Paper no. TPWRS-00009-2019. *(Corresponding author: Shufeng Dong.)*

K. Tang and S. Dong are with the College of Electric Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: tangkunjie1994@zju.edu.cn; dongshufeng@zju.edu.cn).

J. Shen and C. Zhu are with the State Grid Zhejiang Electric Power Corporation, Hangzhou 310027, China (e-mail: 13606809141@139.com; chengzhi_zhu@163.com).

Y. Song is with the State Key Laboratory of Internet of Things for Smart City and the Department of Electrical and Computer Engineering, University of Macau, Macau 999078, China, and also with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: yhsongcn@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TPWRS.2019.2914431

especially for well-conditioned cases. Reference [1] presents a continuous Newton's method and homotopy for ill-conditioned and badly-initialized PFC based on fourth-order Runge-Kutta (RK4) formula, and [15] further presents a modification based on that by combining RK4 and Broyden's method. These methods based on continuous Newton's method usually suffer from large computational burden because they need to solve multiple linear equations in each iteration [14] or calculate many dense matrix operations [15], especially in large-scale systems. Reference [16], [17] propose the factorized load flow, which consists of a linear stage and a nonlinear stage, by introducing auxiliary vectors. This method is more robust than Newton's method, but it needs to solve linear equations twice in each iteration (kLkNL scheme). Also, more time is needed for forming auxiliary vectors and matrices.

Most of the previous research discussed above only test their algorithms based on systems with less than 3,000 buses [1], [7]–[17]. However, with the expansion of power systems, the robustness, convergence rate and efficiency of the algorithms need to be reevaluated in large-scale systems.

To deal with the challenges discussed above, this paper proposes a globally convergent PFC method with trust-region techniques and combines it with Newton's method to achieve a robust and efficient two-stage algorithm for PFC of large-scale systems. The main contribution of the paper is that it:

i) proposes a novel globally convergent PFC method with trust-region techniques, which guarantees convergence and robustness, in various situations, including well-conditioned systems, ill-conditioned systems, ill-conditioned operational modes, heavy loads and inappropriate initial values.

ii) proposes a specific combination approach to combine the globally convergent PFC method with traditional Newton's method to obtain a two-stage algorithm, which has higher convergence rate and computational efficiency than the proposed globally convergent PFC method. The two-stage algorithm can achieve an accurate power flow solution for solvable cases, and can also achieve the least square solution, which is an approximate solution, for unsolvable cases.

iii) presents specific approaches when the proposed method is used for handling limit violations, such as reactive power limits in PV buses, voltage limits.

Numerical experiments show that the proposed globally convergent PFC method and two-stage algorithm have better robustness and efficiency than the existing methods in previous research, especially for large-scale and seriously ill-conditioned cases.

The rest of the paper is organized as follows: Section II introduces a novel globally convergent method for PFC. Section III introduces the approach to combine the globally convergent method with Newton's method, and a novel two-stage algorithm for PFC is presented. Section IV introduces how the proposed approach handles limit violations in realistic cases. Numerical experiments demonstrate its convergence, robustness, and efficiency in Section V. Finally, conclusions are drawn in Section VI.

II. A Novel Globally Convergent PFC Method

This section proposes a novel globally convergent method based on a modification of the L-M method with trust-region techniques [18]–[24]. A preliminary introduction of trust-region methods is presented in Section II-A. Based on that, the steps of the method proposed in this paper are presented in Section II-B. The selection of some parameters is discussed in Section II-C. Conditions for convergence are presented in Section II-D.

A. Preliminary Introduction of the Trust-Region Methods

Line-search methods and trust-region methods are two classes of iterative methods for solving an unconstrained optimization problem (1)

$$\min f(\boldsymbol{x}) \tag{1}$$

where *f* is a twice continuously differentiable function.

Line-search methods aim to find a search step in a specific direction, but the trust-region methods try to find a neighborhood around the current step in which a quadratic model agrees with the objective function. In specific, at each iteration, the search step $\boldsymbol{d}_k$ is obtained by solving the following subproblem [25], [26]

$$\min u_k(\boldsymbol{d}) = f_k + \boldsymbol{g}_k^{\mathrm{T}}\boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^{\mathrm{T}}\boldsymbol{h}_k\boldsymbol{d} \tag{2}$$

subject to

$$\|\boldsymbol{d}\| \le \alpha_k \tag{3}$$

where

$$f_k = f(\boldsymbol{x}_k), \boldsymbol{g}_k = \nabla f(\boldsymbol{x}_k), \boldsymbol{h}_k = \nabla^2 f(\boldsymbol{x}_k) \tag{4}$$

and δ_k is a trust-region radius.

A key point for trust-region methods is to update radius α_k by calculating a ratio τ_k. To calculate this ratio, a merit function $\varphi(x)$ should be first taken. Then, τ_k is usually expressed as

$$\tau_k = \frac{\varphi(\boldsymbol{x}_k) - \varphi(\boldsymbol{x}_k + \boldsymbol{d}_k)}{u_k(0) - u_k(\boldsymbol{d}_k)} \tag{5}$$

where the numerator is called the actual reduction, while the denominator is called the predicted reduction. Some research modifies the actual reduction and the predicted reduction to achieve modern trust-region methods with better performance.

The general steps of trust-region methods are as follows [25], [26]:

a) Initialization. Set $k = 1$, initial point x_1, initial trust-region radius α_1, convergence accuracy requirement *r* and several constants $0 < p_{\mathrm{L}} \le p_{\mathrm{H}} < 1$.

b) Compute f_k, $\boldsymbol{g}_k$, and $\boldsymbol{h}_k$ (or its approximation). If $||\boldsymbol{g}_k|| \le r$, stop.

c) Solve the subproblem (2)–(3) to determine $\boldsymbol{d}_k$.

d) Compute u_k and τ_k.

e) If $\tau_k > p_{\mathrm{L}}$, then set $\boldsymbol{x}_k = \boldsymbol{x}_k + \boldsymbol{d}_k$.

f) Update trust-region radius α_k (increase, decrease, or keep it unchanged) according to the relative sizes of τ_k, p_{L}, and p_{H}.

As shown in the steps, the constants p_{L} and p_{H} are used for defining the ranges of values for τ_k, for which a search step is judged successful or not (this is the role of p_{L}) and, if successful, whether very successful or not (this is the role of p_{H}) [26]. Thus, the trust-region radius is updated by the comparison among p_{L}, p_{H}, and τ_k. p_{L} and p_{H} are user-defined empirical parameters based on numerical experiments.

The modern trust-region methods are considered as the following achievement of the L-M method. However, as shown in the steps above, trust-region methods need to update Hessian matrix or its approximation and solve a quadratic optimization in each iteration, which need more storage space and cost more time than the L-M method. However, the technique of trust-region radius controlling is useful for improving the convergence, and thus, this technique can be applied into the L-M method to control the damping factor and judge whether a search step should be accepted. Therefore, a modified L-M method with trust-region techniques is applied in this paper.

B. Steps of the Proposed Method

The power flow equations can be expressed as nonlinear equations:

$$\boldsymbol{F}(\boldsymbol{X}) = \mathbf{0} \tag{6}$$

Where $\boldsymbol{X}$ represents state variables of equations in PFC. In this paper, the state variables are bus voltage magnitudes and angles when there is no limit violation. However, when limit violation exists, the state variables are changed, which will be further discussed in Section IV.

The detailed steps of the algorithm are as follows:

Algorithm I

a) Set: i) initial values of the state variables of $\boldsymbol{X}_1$; ii) several constant parameters $0 < p_0 \le p_{\mathrm{L}} \le p_{\mathrm{H}} < 1$, $\alpha_1 > m > 0$, $s > 1$; iii) convergence accuracy requirement of residual vector r; iv) the current iteration number $k = 1$;

b) Calculate the damping factor in the k-th iteration λ_k:

$$\lambda_k = \alpha_k \Gamma(\boldsymbol{F}(\boldsymbol{X}_k)) \tag{7}$$

where $\Gamma(\boldsymbol{F}(\boldsymbol{X}_k))$ indicates a function of $\boldsymbol{F}(\boldsymbol{X}_k)$.

c) Calculate the search step $\Delta\boldsymbol{X}_k$ in the k-th iteration, where $\boldsymbol{J}(\boldsymbol{X}_k)$ represents the Jacobian matrix of $\boldsymbol{F}$ evaluated at $\boldsymbol{X}_k$.

$$\Delta\boldsymbol{X}_k = -[\boldsymbol{J}(\boldsymbol{X}_k)^{\mathrm{T}}\boldsymbol{J}(\boldsymbol{X}_k) + \lambda_k \boldsymbol{I}]^{-1}\boldsymbol{J}(\boldsymbol{X}_k)^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{X}_k) \tag{8}$$

d) Take (9) as the merit function:

$$\varphi(x) = \|\boldsymbol{F}(\boldsymbol{X}_k)\|^2 \tag{9}$$

Thus, the actual reduction and the predicted reduction at the k-th iteration are defined as follows:

$$Ared_k = \|\boldsymbol{F}(\boldsymbol{X}_k)\|^2 - \|\boldsymbol{F}(\boldsymbol{X}_k + \Delta\boldsymbol{X}_k)\|^2 \tag{10}$$

$$Pred_k = \|\boldsymbol{F}(\boldsymbol{X}_k)\|^2 - \|\boldsymbol{F}(\boldsymbol{X}_k) + \boldsymbol{J}(\boldsymbol{X}_k)\Delta\boldsymbol{X}_k\|^2 \tag{11}$$

e) Calculate the ratio τ_k:

$$\tau_k = \frac{Ared_k}{Pred_k} \tag{12}$$

f) Compare τ_k with p_0 to decide whether to accept the current search step:

$$\boldsymbol{X}_{k+1} = \begin{cases} \boldsymbol{X}_k + \Delta\boldsymbol{X}_k & \tau_k > p_0 \\ \boldsymbol{X}_k & \tau_k \le p_0 \end{cases} \tag{13}$$

Steps d)–f) use the technique in the trust-region methods proposed in Section II-A to calculate a ratio for judging whether the current search step should be accepted. Here, p_0 shares a similar function with p_{L} in trust-region methods.

g) Adjust the self-adaptive factor:

$$\alpha_{k+1} = \begin{cases} s\alpha_k & \tau_k < p_L \\ \alpha_k & p_L \le \tau_k \le p_H \\ \max\{\frac{\alpha_k}{s}, m\} & \tau_k > p_H \end{cases} \tag{14}$$

Here, p_{L} and p_{H} are used for adjusting self-adaptive factor, similar to p_{L} and p_{H} in trust-region methods for updating trust-region radius.

h) If (15) is satisfied, go to Step i). Otherwise, let $k = k + 1$ and return to Step b).

$$\min\{\left\|\boldsymbol{J}(\boldsymbol{X}_k)^{\mathrm{T}}\boldsymbol{F}(\boldsymbol{X}_k)\right\|, \|\boldsymbol{F}(\boldsymbol{X}_k)\|_\infty\} < r \tag{15}$$

i) If $||\boldsymbol{F}(\boldsymbol{X}_k)||_\infty < r$, the solution of PFC under required accuracy is obtained. Otherwise, the algorithm cannot obtain the accurate solution, but the sequence of $\boldsymbol{X}$ converges to a least square solution, which can be considered as an approximate solution of PFC.

C. Parameters Selection

As seen in Algorithm I, the function of the damping factor, as well as several constant parameters, need to be determined.

Damping factor λ is one of the most important parameters in Algorithm I, which influences the convergence rate. There are some common selections in previous research in mathematics [18]–[24]. For example, [18] uses $\Gamma(\boldsymbol{F}(\boldsymbol{X}_k)) = ||\boldsymbol{F}(\boldsymbol{X}_k)||$, [19] uses $\Gamma(\boldsymbol{F}(\boldsymbol{X}_k)) = ||\boldsymbol{F}(\boldsymbol{X}_k)||^2$ and [20] uses $\Gamma(\boldsymbol{F}(\boldsymbol{X}_k)) = ||\boldsymbol{F}(\boldsymbol{X}_k)||/(1 + ||\boldsymbol{F}(\boldsymbol{X}_k)||)$. However, such expressions do not bring a high convergence rate in PFC.

According to the characteristics of the convergence curves during iterations of PFC, it is reasonable to combine different expressions of damping factor to achieve better convergence. Specifically, the following expression of the damping factor is proposed:

$$\Gamma(\boldsymbol{F}(\boldsymbol{X}_k)) = \begin{cases} \|\boldsymbol{F}(\boldsymbol{X}_k)\|_\infty, \|\boldsymbol{F}(\boldsymbol{X}_k)\| \ge g \\ \|\boldsymbol{F}(\boldsymbol{X}_k)\|^2, \|\boldsymbol{F}(\boldsymbol{X}_k)\| < g \end{cases} \tag{16}$$

where g is a critical value, which is a user-defined parameter. Usually, $g = 1$ can achieve a satisfying performance.

This expression selection is different from that in most previous research achievements in mathematics. In previous achievements, when $||\boldsymbol{F}(\boldsymbol{X}_k)||$ is very large, the expression selection is aimed to make the damping factor not too large, so the search step is not too small and the algorithm can move fast to the solution set in initial iterations. However, in PFC, especially in PFC of ill-conditioned systems, the search step in initial steps should be small enough, otherwise, a large number of 'failed steps'-$\tau_k \leq p_0$ in Step f)-will be produced, which reduces the efficiency. Here, $||\boldsymbol{F}(\boldsymbol{X}_k)||_\infty$ is applied. It is smaller than $||\boldsymbol{F}(\boldsymbol{X}_k)||$ but usually larger than 1 in initial iterations. Thus, the search step will not be too small, nor too large, so that Algorithm I can have a better convergence performance in initial iterations.

When $\boldsymbol{X}_k$ is close to a power flow solution, $||\boldsymbol{F}(\boldsymbol{X}_k)||$ is close to 0 in final iterations. Thus, $||\boldsymbol{F}(\boldsymbol{X}_k)||^2$ is applied, instead of common selections such as $||\boldsymbol{F}(\boldsymbol{X}_k)||$, because $||\boldsymbol{F}(\boldsymbol{X}_k)||^2$ makes the damping factor smaller and the search step will be larger to realize fast convergence in final iterations.

In summary, the proposed damping factor expression mainly has two advantages for reducing iterations: i) shorter and safer search steps in initial iterations to reduce the number of failed steps; ii) larger search steps in final iterations to realize faster convergence.

Further numerical experiments in Section V demonstrate the effectiveness of the expression selection of damping factor.

Besides, the prespecified constant parameters in Algorithm I are also important for the performance of the algorithm. They must be carefully chosen based on numerical results.

D. Conditions for Convergence

1) Conditions for Global Convergence:

Theorem I [20]*:* If $\boldsymbol{F}(\boldsymbol{X})$ is continuously differentiable and Lipschitz continuous, and $\boldsymbol{J}(\boldsymbol{X})$ is Lipschitz continuous, i.e., there exist positive constants L_1 and L_2 such that

$$\|\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{J}(\boldsymbol{y})\| \leq L_1 \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{D} \tag{17}$$

$$\|\boldsymbol{F}(\boldsymbol{x}) - \boldsymbol{F}(\boldsymbol{y})\| \leq L_2 \|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{D} \tag{18}$$

where $\boldsymbol{D}$ represents the domain of definition of $\boldsymbol{F}(\boldsymbol{X})$ and $\boldsymbol{J}(\boldsymbol{X})$. Then, Algorithm I terminates in finite iterations or satisfies:

$$\lim_{k\to\infty} \inf \left\| \boldsymbol{J}(\boldsymbol{X}_k)^{\mathrm{T}} \boldsymbol{F}(\boldsymbol{X}_k) \right\| = 0 \tag{19}$$

The specific proof of Theorem I is presented in [20], [24]. Power flow functions $\boldsymbol{F}(\boldsymbol{X})$ are continuously differentiable. The state variables are all bounded in practical engineering so that it is not hard to prove that $\boldsymbol{F}(\boldsymbol{X})$ is Lipschitz continuous for each variable and eigenvalues of $\boldsymbol{J}(\boldsymbol{X})$ are bounded. Then, it can be proved that there exist L_1 and L_2 to make (17) and (18) hold. According to Theorem I, Algorithm I has global convergence properties.

2) Conditions for Local Convergence: The local convergence of Algorithm I is discussed in [20], [24]: if there exists a solution x^*, and the initial value is on some neighborhood of x^*, where $||\boldsymbol{F}(\boldsymbol{X})||$ provides a local error bound on this neighborhood, the sequence will converge to x^*. Power flow equations admit multiple solutions. However, the initial values are usually close to the real-world operational point, so that $\boldsymbol{X}_k$ will finally converge to this point. Section V-B-3) also demonstrates this conclusion.

III. A Novel Two-Stage PFC Algorithm

This section presents an approach that combines the proposed globally convergent method with traditional Newton's method. The combined algorithm consists of two stages. Section III-A illustrates the meaning of the combination approach by comparing different advantages between Algorithm I and traditional Newton's method. Section III-B gives the specific combination approach and stage division of the proposed algorithm. Advantages of the proposed two-stage PFC algorithm are summarized in Section III-C.

A. The Meaning of Combination

In fact, Algorithm I and traditional Newton's method can both obtain the solution of power flow equations. However, under different systems, operational modes and initial values, these two methods have significantly different performances. The meaning of combining these two algorithms is to make use of their different advantages.

1) The Advantage of Algorithm I: As analyzed in Section II, Algorithm I has extensive applicability and it works even under some extremely ill-conditioned systems, operational modes, and inappropriate initial values, due to its global convergence properties. Thus, the robustness of Algorithm I is much better than that of Newton's method. By combining these two methods and taking Algorithm I as the first stage, the two-stage algorithm will have better convergence and robustness.

2) The Advantage of Newton's Method: Compared with Newton's method, computational efficiency of each iteration in Algorithm I is lower, because there are several matrix-matrix, matrix-vector and vector operations in Algorithm I, such as in Step c) and d), which are not needed in Newton's method. This difference in efficiency will be very significant in large-scale systems.

On the other hand, numerical experiments further show that, under the same accuracy requirement, Algorithm I usually needs more iterations than Newton's method, which further reduces the efficiency.

By combining these two methods, the two-stage algorithm will have higher computational efficiency.

B. Combination Approach

The main logic and steps of the proposed two-stage algorithm after combination are shown in Fig. 1.

As shown in Fig. 1, in Stage I, iterations in Algorithm I are used for searching the solution from the initial values. Also, in this stage, it is necessary to determine the condition to enter Stage II. If the condition is not satisfied during the iterations, Algorithm I will be continued to achieve the least square solution-approximate solution for PFC. If the condition is satisfied, the algorithm enters Stage II, and Newton's method will use the current values of state variables as initial values. However, this condition cannot guarantee the solvability of the power

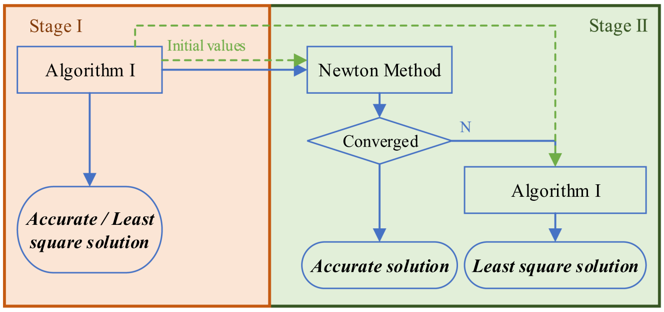

Fig. 1. Stage division of the proposed two-stage PFC algorithm.

system equations. Thus, if Newton's method converges, the accurate solution is achieved, otherwise, the algorithm goes back to Algorithm I, and Algorithm I will use the values of state variables before entering Stage II as initial values to achieve least square solution finally.

Thus, there is one key point in the combination approach. That is to judge whether the values of state variables in the current iteration are close enough to the real operational point.

On the one hand, the value of λ_k can be used as an index in the condition for entering Stage II. On the other hand, considering that the convergence of Newton's method is very sensitive to initial values, the current power mismatch vector $\boldsymbol{F}(\boldsymbol{X}_k)$ is also an important index in the condition. Based on mass numerical experiments, an empirical condition is that, if (13) is satisfied, the algorithm enters Stage II:

$$\begin{aligned}&\exists k \geq 3, \lambda_{k-2} > s\lambda_{k-1} > s^2\lambda_k, \|\boldsymbol{F}(\boldsymbol{X}_k)\|_\infty < 1,\\&\|\boldsymbol{F}(\boldsymbol{X}_{k-2})\|_\infty > \|\boldsymbol{F}(\boldsymbol{X}_{k-1})\|_\infty > \|\boldsymbol{F}(\boldsymbol{X}_k)\|_\infty\end{aligned} \tag{20}$$

where k is the current iteration number. If (20) is not satisfied, the algorithm stays in Stage I and iterations in Algorithm I are continued to convergence and achieve the least square solution, which is an approximate solution for power flow equations.

Finally, the specific implementation of the two-stage algorithm is shown in Fig. 2.

C. Advantages of the Two-Stage PFC Algorithm

The proposed two-stage PFC algorithm makes full use of different features of Algorithm I and Newton's method. Two advantages of this algorithm are as follows:

1) Robustness in Initial Iterations: From Algorithm I, it can be seen that λ_k is large in the initial iterations of the two-stage algorithm. Thus, a shorter and safer search step will be taken. This will slow down the progress, but also increase the robustness.

2) High Efficiency and Convergence in Later Iterations: In later iterations in Stage II, Newton's method has higher efficiency and quadratic convergence. If (20) is not satisfied during iterations in Stage I, or Newton's method diverges in Stage II, however, Algorithm I will be continued. In later iterations, λ_k is close to 0, and the algorithm will also have quadratic convergence of Gauss Newton's method, avoiding the sawtooth oscillation.

IV. Handling Limit Violations

The PFC algorithm should consider limit violations in realistic cases. Providing a detailed account of how any realistic case can be handled by the proposed algorithm is clearly beyond a single paper. However, two typical realistic cases will be discussed as representatives to validate the proposed algorithm. Finally, generalized approaches for removing limit violations will be presented.

A. Handling Reactive Power Limits in PV Buses

Assumed that all the PV buses are under local generator control, the PV-PQ switching logic in each iteration of the proposed algorithm is as shown in Fig. 3 [27]. Q_i is the reactive power injection at Bus i. $Q_{i,max}$ and $Q_{i,min}$ are the upper and lower limit of it. $Q_i\prime$ is the calculation result of reactive power injection at Bus i. U_i is the updated voltage magnitude of Bus i. $U_{i,g}$ is the voltage setpoint of Bus i.

In addition, considering that the above switching logic may lead to frequent switch of some PV buses, which reduces the efficiency or even leads to divergence, a supplement for the above logic is that when type-switching times of a bus is larger than N (e.g., $N = 3$), this bus is fixed as a PQ-bus which will not be switched in the following iterations.

This switching logic is usually used in Newton's method. The switching logic influences the update of $\boldsymbol{J}$, $\boldsymbol{F}$, and $\boldsymbol{X}$, while the other matrices and parameters are all calculated from $\boldsymbol{J}$, $\boldsymbol{F}$, and $\boldsymbol{X}$, which will not be influenced directly by the switching logic so that the logic is also applicable to Algorithm I without any extra step. Furthermore, the proposed two-stage algorithm applies this switching logic in each iteration to handle reactive power limits in PV buses.

B. Handling Voltage Limits With Tap Changers Action

Without losing generality, it is assumed that transformer tap changers are used to control voltage by single-criterion, i.e., one tap changer constrains the voltage magnitude of one bus when the voltage of this bus violates its limit. The switching logic for handling voltage limits with tap changers action in each iteration is as shown in Fig. 4, where U_i is the updated voltage magnitude of Bus i; $U_{i,max}$ and $U_{i,min}$ are the upper and lower voltage limit of Bus i; t_i is the ratio of the tap changer used for controlling Bus i (t_i is the ratio of primary voltage to secondary voltage, and Bus i is considered as the secondary side); $t_{i,0}$ is the initial ratio of the tap changer used for controlling Bus i when U_i is not constrained. The detailed information of the modified Jacobian matrix is shown in [28].

Both local control and remote control can be handled with the proposed switching logic. The Jacobian matrix modification for a remote transformer is in a similar way with that for a local one. However, remote control usually requires changes in the tap changer ratio larger than would be required for local control. This might increase convergence difficulties.

Similar with the approach for handling voltage limit with transformer tap changers action, phase-shifters can be used for controlling branch power to avoid branch power limit violation;

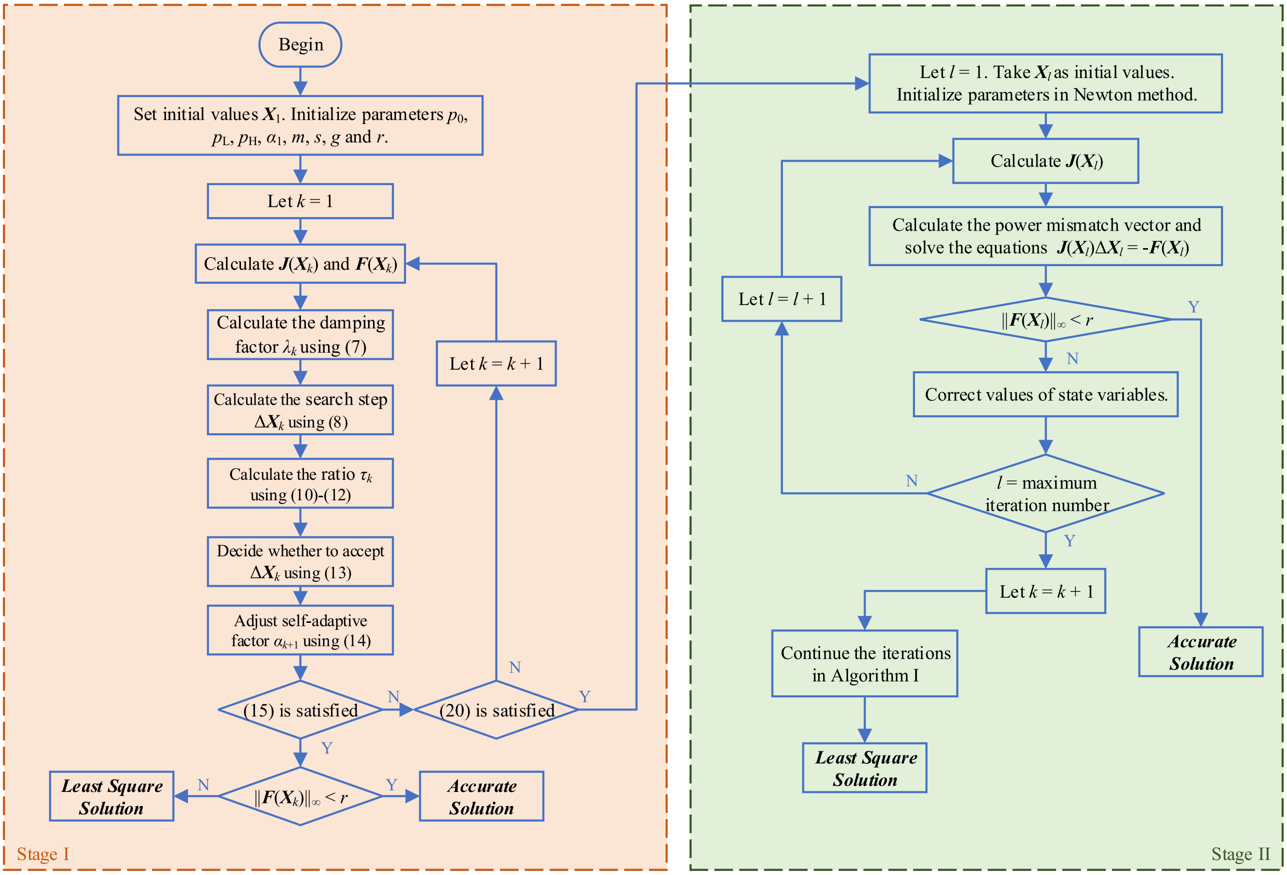
Begin
Set initial values $\boldsymbol{X}_1$. Initialize parameters p_0, p_L, p_H, α_1, m, s, g and r.
Let $k = 1$
Calculate $\boldsymbol{J}(\boldsymbol{X}_k)$ and $\boldsymbol{F}(\boldsymbol{X}_k)$
Calculate the damping factor λ_k using (7)
Calculate the search step $\Delta\boldsymbol{X}_k$ using (8)
Calculate the ratio τ_k using (10)-(12)
Decide whether to accept $\Delta\boldsymbol{X}_k$ using (13)
Adjust self-adaptive factor α_{k+1} using (14)
(15) is satisfied
N
(20) is satisfied
N
Let $k = k + 1$
Y
Y
$\|\boldsymbol{F}(\boldsymbol{X}_k)\|_\infty < r$
N
Least Square Solution
Y
Accurate Solution
Stage I
Let $l = 1$. Take $\boldsymbol{X}_l$ as initial values. Initialize parameters in Newton method.
Calculate $\boldsymbol{J}(\boldsymbol{X}_l)$
Calculate the power mismatch vector and solve the equations $\boldsymbol{J}(\boldsymbol{X}_l)\Delta\boldsymbol{X}_l = -\boldsymbol{F}(\boldsymbol{X}_l)$
Let $l = l + 1$
$\|\boldsymbol{F}(\boldsymbol{X}_l)\|_\infty < r$
Y
N
Correct values of state variables.
N
l = maximum iteration number
Y
Let $k = k + 1$
Continue the iterations in Algorithm I
Least Square Solution
Accurate Solution
Stage II

Fig. 2. Implementation of the two-stage algorithm.

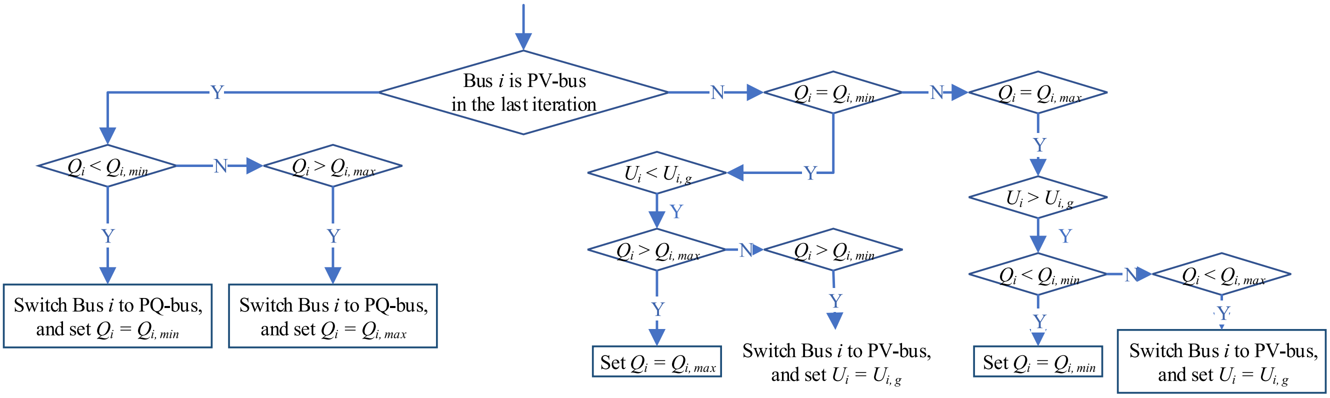

Fig. 3. PV-PQ switching logic.

reactive power compensators can be used for controlling voltage to avoid bus voltage limit violation.

C. Generalized Approaches for Removing Limit Violations

Generally, two classes of limits exist in PFC, adjustable variable limits, such as limits of transformer taps, phase-shifter taps, reactive power compensators, generator power, etc., and constrained variable limits, such as limits of bus voltage magnitude, branch flow, etc.

The adjustable variable and the associated constrained variable are usually treated as a pair, i.e., one of them will always be considered as a state variable for Jacobian matrix formation, and the other will be considered as a non-state variable [29]. When a state variable violates its limits, fix the variable at the violated limit, allow the associated adjustable /constrained variable to replace the original state variable with this associated variable. This approach usually only involves the modification of the Jacobian matrix and corresponding linear equations. Other modifications are needless.

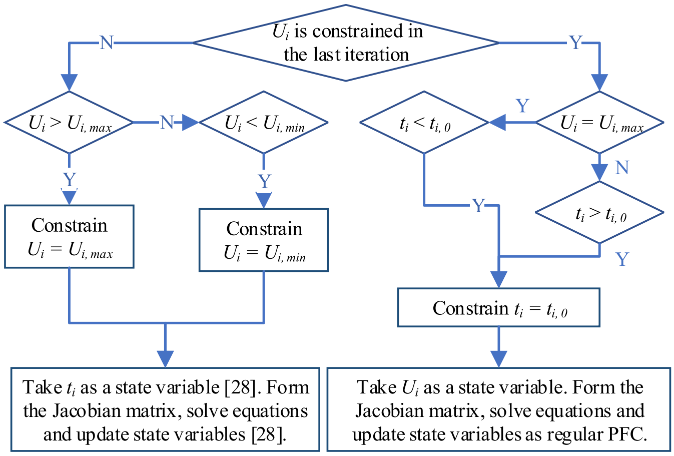
U_i is constrained in the last iteration
N
Y
$U_i > U_{i,max}$
N
$U_i < U_{i,min}$
$t_i < t_{i,0}$
Y
$U_i = U_{i,max}$
N
Y
Y
Y
$t_i > t_{i,0}$
Y
Constrain $U_i = U_{i,max}$
Constrain $U_i = U_{i,min}$
Constrain $t_i = t_{i,0}$
Take t_i as a state variable [28]. Form the Jacobian matrix, solve equations and update state variables [28].
Take U_i as a state variable. Form the Jacobian matrix, solve equations and update state variables as regular PFC.

Fig. 4. Voltage-tap changer ratio switching logic.

In the proposed Algorithm I and two-stage algorithm, all the steps are based on the Jacobian matrix. Thus, to handle the limit violations, the only necessary modification is to modify the Jacobian matrix depending on the adjustable/constrained variable and other steps are the same with those in the algorithms regardless limit violations.

V. Numerical Experiments

This section uses several cases from MATPOWER for numerical experiments. The programs for the test run on the Windows 10 of 64 bits. The CPU is Intel Core i7-7700 K, with 4.20 GHz master frequency and 32GB memory. The programming language used is MATLAB R2015b. The convergence tolerance of PFC is set to 1e-6 p.u. All the time cost data in the following figures and tables is the average consumed time of 10 times calculations. According to numerical experiment results and previous research [18]–[24], parameters in Algorithm I are set as: $p_0 = $ 1e-4, $p_\mathrm{L} = 0.25$, $p_\mathrm{H} = 0.75$, $\alpha_1 = 1$, $m = $ 1e-8, $s = 4$ and $g = 1$.

The proposed Algorithm I and two-stage algorithm have been compared with seven other methods as follows:

i) Traditional Newton's method in MATPOWER 6.0.
ii) Iwamoto's method [8].
iii) The high-order LM method (HOLM) [12].
iv) Newton Raphson-Jacobian Marquardt method (NRJM) [9].
v) The 4th-order Runge-Kutta method (RK4) [1].
vi) The RK4-Broyden method (RK4B) [15].
vii) The factorized load flow method (FLF) [16].

A. Effectiveness of Expression Selection of Damping Factor

The expression of damping factor is closely related to the performance of Algorithm I. To test the effectiveness of the proposed expression in (16), Table I compares the iteration numbers under different expressions of damping factor. Here, flat start is applied, and variable limits are ignored.

TABLE I
Iteration Numbers under Different Expressions of Damping Factor

Case	(16)	$\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|$ [16]	$\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|^2$ [17]	$\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|$ $/(1+\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|)$ [18]	$\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|^{1/\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|}$ (if $\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|\geq 1$) $\|\|\boldsymbol{F}(\boldsymbol{X}_k)\|\|^{1+1/k}$ (otherwise) [20]
case118	5	6	6	6	6
case300	7	8	8	8	8
case1354pegase	7	8	9	7	8
case2383wp	8	9	10	9	9
case3375wp	9	9	10	9	9
case6515rte	13	19	13	>200	>200
case9241pegase	10	12	11	11	11
case13659pegase	11	18	13	22	26

It can be seen that, for all these cases from small to large, the proposed expression of damping factor (16) can make Algorithm I have the least iteration number and highest convergence rate, compared with other common expressions of damping factor in previous research. This demonstrates the effectiveness of the expression selection in this paper.

B. Robustness and Efficiency Analysis

This section uses some cases to demonstrate robustness and efficiency of Algorithm I and the two-stage algorithm under well- and ill-conditioned systems as well as cases under ill-conditioned operational modes, heavy loads and inappropriate initial values. Here, variable limits are ignored.

1) Well-conditioned Systems, Ill-conditioned Systems, and Ill-conditioned Operational Modes: Table II compares the time consumptions and iteration numbers between the proposed two-stage algorithm and other algorithms. Here, flat start is applied.

As seen in Table II, the proposed Algorithm I and the two-stage algorithm can effectively converge with a high convergence rate for all these cases. It can achieve accurate solutions for both well- and ill-conditioned systems and converge to the least square solutions for three ill-conditioned operational modes. Besides, the numbers of successful/failed search steps in Algorithm I and Stage I in the two-stage algorithm are shown in Table II. The number of failed steps is 0 for all cases except 'case6515rte'. Although Algorithm I has 2 failed steps for 'case6515rte', the proposed method is well-behaved generally for different cases.

Newton's method has a higher convergence rate and less time cost for well-conditioned systems than the proposed Algorithm I and two-stage algorithm, but it will diverge for ill-conditioned systems. Besides, Newton's method cannot diverge to an approximate solution for unsolvable cases like the proposed algorithm. Thus, the proposed algorithms have better universality than Newton's method.

Iwamoto's method has better robustness than Newton's method, by introducing the 'optimal multiplier'. It converges for some ill-conditioned systems such as 'case3375wp' and 'case9241pegase', but it needs more iterations and costs more time than Algorithm I and two-stage algorithm. For some other ill-conditioned systems, it has a very low convergence rate and cannot converge to the required accuracy within 200 iterations. For the unsolvable cases with ill-conditioned operational modes, Iwamoto's method fails because the optimal multiplier equals approximately to 0 in each iteration.

HOLM can converge for all these cases. It has a similar convergence rate for all cases with Algorithm I and the two-stage

TABLE II
COMPARISON OF TIME CONSUMPTIONS AND ITERATION NUMBERS AMONG DIFFERENT ALGORITHMS

Type	Case	Two-Stage Algorithm (Stage I, Stage II, Total) (successful/failed steps)	Algorithm I (successful /failed steps)	Newton	Iwamoto	HOLM	NRJM	RK4	RK4B	FLF
Well-conditioned systems	case118	6.9 (3, 2, 5) (3/0)	6.5 (5) (5/0)	5.0 (4)	7.4 (4)	8.9 (4)	7.2 (4)	50.6 (19)	32.6 (17)	8.5 (3)
	case300	16.5 (3, 3, 6) (3/0)	17.5 (7) (7/0)	10.1 (5)	13.4 (5)	30.1 (6)	42.9 (12)	120.1 (21)	414.7 (19)	15.9 (4)
	case1354pegase	56.3 (3, 3, 6) (3/0)	64.4 (7) (7/0)	37.0 (5)	59.3 (6)	141.5 (7)	(>200)	667.5 (26)	>120000	59.6 (4)
	case2383wp	105.2 (3, 4, 7) (3/0)	139.3 (8) (8/0)	49.4 (4)	93.3 (5)	280.9 (7)	229.6 (8)	845.8 (20)		97.1 (4)
Ill-conditioned systems	case3375wp	142.5 (3, 3, 6) (3/0)	230.7 (9) (9/0)	Diverge	420.1 (16)	403.7 (7)	(>200)	Diverge		183.3 (4)
	case6515rte	376.2 (4, 4, 8) (4/0)	690.9 (13) (11/2)	Diverge	(>200)	1349.2 (9)		Diverge		Diverge
	case9241pegase	504.5 (4, 4, 8) (4/0)	933.4 (10) (10/0)	Diverge	1774.2 (12)	2176.3 (9)		5648.3 (26)		1234.4 (4)
	case13659pegase	891.3 (4, 5, 9) (4/0)	1413.1 (11) (11/0)	Diverge	(>200)	3398.2 (10)		7478.8 (26)		2105.5 (5)
Ill-conditioned operational modes	case118 (Fault: Branch 8-9)	10.4 (9, 0, 9) (9/0)	10.4 (9) (9/0)	Diverge	Fail	15.4 (8)	121.0 (95)	Diverge	Diverge	Diverge
	case118 (Fault: Branch 9-10)	11.4 (9, 0, 9) (9/0)	10.9 (9) (9/0)	Diverge	Fail	15.6 (8)	(>200)	Diverge	Diverge	Diverge
	case118 (Fault: Branch 71-73)	7.8 (6, 0, 6) (6/0)	8.2 (6) (9/0)	Diverge	Fail	9.0 (4)	7.6 (4)	Diverge	Diverge	Diverge

algorithm. However, the efficiency of HOLM is significantly lower, because it needs to solve four linear equations, instead of one, in each iteration. Specifically, it needs around 3∼4 times of time consumption than the two-stage algorithm, and around 2∼2.5 times of time consumption than Algorithm I in large cases with over 3,000 buses.

NRJM has unstable convergence rate for different cases. It will converge very slowly for large-scale ill-conditioned systems and some ill-conditioned operational modes.

RK4 can solve some ill-conditioned systems effectively but not all of them. It will diverge when solving 'case3375wp' and 'case6515rte' for instance. RK4 needs much more iterations and time cost than the proposed algorithms because it needs to solve four linear equations instead of one in each iteration. Moreover, it will diverge under the three ill-conditioned operational modes of 'case118'.

RK4B reduces the computational burden of RK4 by reducing the Jacobian matrix inverse calculation and its formation. However, calculations for replacing the inverse calculation are based on dense matrix operation. The dimension of dense matrices is the same as that of the Jacobian matrix. Therefore, in 'case118' or smaller cases proposed in [15], RK4B will have an acceleration effect. However, for the larger cases, the improvement is ineffective or even leads to worse efficiency than RK4.

FLF is a competitive method. On the one hand, it even has a higher convergence rate than Newton's method. On the other hand, it has better robustness than Newton's method, which can solve numerous ill-conditioned cases, although not all of them. It will diverge when solving 'case6515rte' and the cases under ill-conditioned operational modes (unsolvable cases). Although it needs fewer iterations than the proposed method, it costs more time, especially for larger cases, because it needs to solve linear equations twice instead of once in each iteration.

The two-stage algorithm is an improvement of Algorithm I. It effectively enhances the convergence rate and reduces the time cost by combining Algorithm I with Newton's method, for well- and ill- solvable systems. In addition, it has the same performance with Algorithm I for unsolvable systems, such as the ill-conditioned operational modes in Table I, because the two-stage algorithm will stay in Stage I to calculate least square solutions and Stage I is equivalent to Algorithm I.

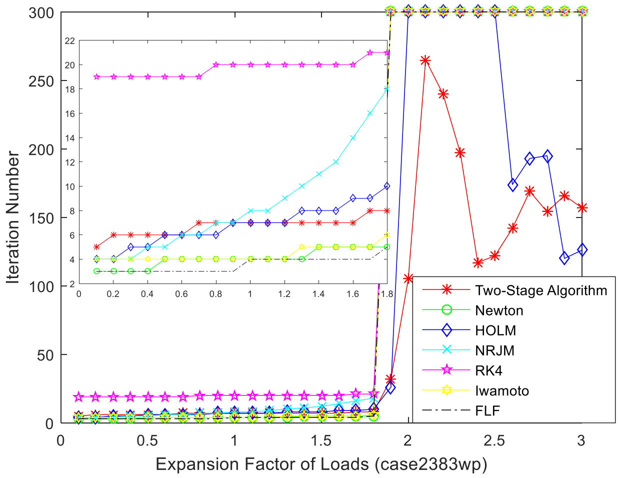

Fig. 5. Iterations under different expansion factors of loads-'case2383wp'.

2) Heavy Loads: Heavy loads may make PFC algorithms unsolvable. Fig. 5 compares the iteration numbers using 'case2383wp'. When the factor is equal to or larger than 1.9, the system is unsolvable. Newton's method, NRJM, RK4, Iwamoto's method, and FLF will diverge. HOLM cannot converge within 300 iterations when the factor is 2.0∼2.5. However, the two-stage algorithm still converges to a least square solution within 300 iterations. For instance, when the factor is 1.9, the two-stage algorithm converges with $||\boldsymbol{F}(\boldsymbol{X})||_{\infty} = 7.41\text{e}-4$. Although it does not satisfy the accuracy requirement 1e-6, it is still a valuable approximation in practical application.

3) Inappropriate Initial Values: Inappropriate initial values, which are far from the real operational point, usually make some PFC algorithms diverge. The sensitivity to initial values is an important evaluation factor for a PFC algorithm. Fig. 6 compares the iteration numbers among different algorithms under different initial values of voltage magnitude. Here, the initial value of the voltage phase is 0. It can be seen that Newton's method, NRJM, RK4, Iwamoto's method, and FLF only converge when

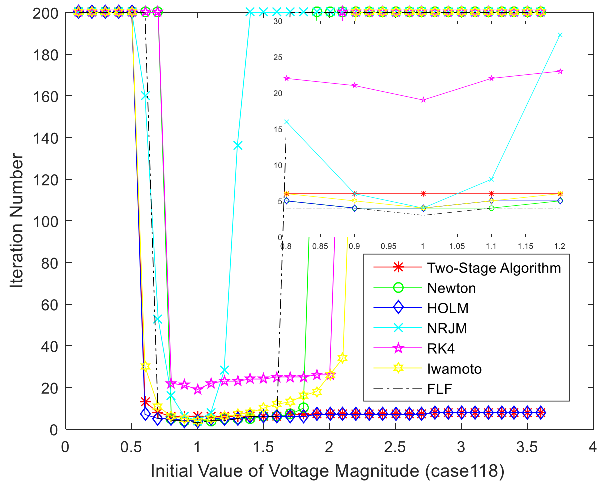

Fig. 6. Iterations under different initial values of voltage magnitude-‘case118’.

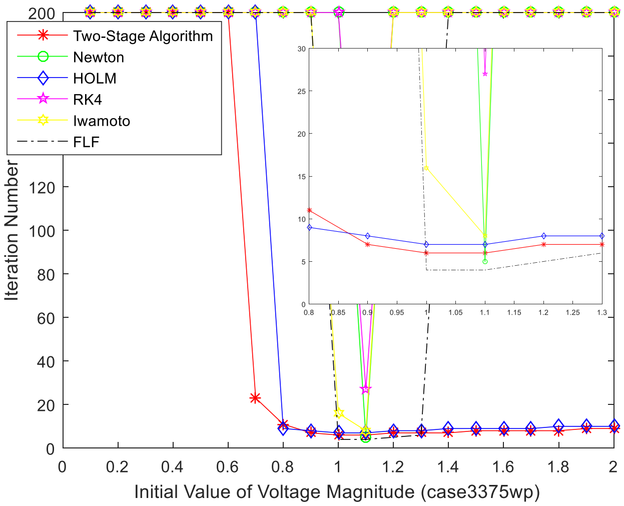

Fig. 7. Iterations under different initial values of voltage magnitude-‘case3375wp’.

the initial value is near to 1.0 p.u. –0.8~1.8 p.u., 0.7~1.3 p.u., 0.8~2.0 p.u., 0.6~2.1 p.u. and 0.7~1.6 p.u., respectively. However, the two-stage algorithm and HOLM have much better robustness. When the initial value is from 0.6 to 1.0 p.u., they can converge in less than 20 iterations. When the initial value is over 1.0 p.u., even over 3.5 p.u., they can converge with less than 10 iterations.

Similarly, Fig. 7 compares the iteration numbers using ‘case3375wp’. It is an ill-conditioned system, and Newton’s method cannot converge under flat start. It can be seen that Newton’s method, RK4, Iwamoto’s method, and FLF only converge when the initial value is 1.1 p.u., 1.1 p.u., 1.0~1.1 p.u. and 1.0~1.3 p.u., respectively. However, the two-stage algorithm and HOLM have much better robustness. They can converge when the initial value is from 0.7 to 2.0 p.u. As mentioned before, although HOLM has almost the same number of iterations with the proposed two-stage algorithm, it will cost much more time, because it needs to solve four equations in each iteration, instead of one.

TABLE III
COMPARISON OF TIME CONSUMPTIONS AND ITERATION NUMBERS CONSIDERING REACTIVE POWER LIMITS IN PV BUSES

Case	Two-Stage Algorithm (Stage I, Stage II, Total) (successful/failed steps)	Algorithm I (successful /failed steps)	Newton
case118	9.9 (3, 2, 5) (3/0)	10.3 (5) (5/0)	8.2 (4)
case2383wp	208.2 (3, 11, 14) (3/0)	297.7 (14) (14/0)	185.9 (13)
case3375wp	356.9 (8, 6, 14) (8/0)	416.1 (14) (14/0)	Diverge
case6515rte	626.2 (9, 4, 13) (9/0)	776.9 (14) (14/0)	Diverge
case13659pegase	864.3 (3, 6, 9) (3/0)	3367.0 (26) (18/8)	Diverge

TABLE IV
COMPARISON OF TIME CONSUMPTIONS AND ITERATION NUMBERS CONSIDERING TAP CHANGERS ACTION FOR HANDLING VOLTAGE LIMITS

Case	Control Mode	Adjustable Tap Changers	Constrained Bus Number	Upper Limit /p.u.	Lower Limit /p.u.
case300	Local	15-17	17	No Modifications	
	Local	159-117	117		
	Local	141-174	174		
	Local	9003-9031	9031		
	Local	9003-9033	9033		
	Local	9003-9038	9038		
	Newton			23.2 (9)	
	Algorithm I (successful /failed steps)			39.9 (11) (11/0)	
	Two-Stage Algorithm (Stage I, Stage II, Total) (successful/failed steps)			30.7 (3, 7, 10) (3/0)	
case300	Local	15-17	17	No Modifications	
	Remote	159-117	118		
	Remote	141-174	178		
	Local	9003-9031	9031		
	Local	9003-9033	9033		
	Local	9003-9038	9038		
	Newton			30.8 (11)	
	Algorithm I			63.0 (19) (11/8)	
	Two-Stage Algorithm			34.4 (3, 9, 12) (3/0)	
case 3375wp	Local	10368-10215	10215	1.04	0.95
	Local	10291-10070	10070	1.04	0.95
	Local	10295-10076	10076	1.04	0.95
	Local	10113-10114	10114	1.09	0.95
	Newton			Diverge	
	Algorithm I			455.1 (15) (15/0)	
	Two-Stage Algorithm			332.2 (3, 12, 15) (3/0)	
case 3375wp	Remote	10368-10215	10217	1.04	0.95
	Remote	10291-10070	10011	1.04	0.95
	Remote	10295-10076	10024	1.04	0.95
	Remote	10113-10114	10127	1.09	0.95
	Newton			Diverge	
	Algorithm I			417.8 (14) (14/0)	
	Two-Stage Algorithm			307.4 (3, 11, 14) (3/0)	

Therefore, from the above analysis, it can be seen that the proposed two-stage algorithm has the best performance comprehensively considering their robustness and efficiency under well- and ill-conditioned systems as well as cases under ill-conditioned operational modes, heavy loads and inappropriate initial values, compared with the existing methods in previous research.

C. Handling Limit Violations

This section uses some cases to validate the proposed algorithm in two realistic cases presented in Section IV.

1) Handling Reactive Power Limits in PV Buses: Table III compares the time consumptions and iteration numbers when different algorithms are used for solving the cases considering reactive power limits in PV buses. Here, flat start is applied and other limits are ignored.

Table III shows that PV-PQ switching logic leads more iterations for all these algorithms. However, both Algorithm I and the two-stage algorithm can still effectively converge with a relatively high convergence rate for both well- and ill-conditioned cases. Besides, the number of failed steps is 0 for most cases, which indicates that the proposed method is still well-behaved generally for handling reactive power limits.

2) Handling Voltage Limits With Tap Changers Action: Table IV compares the time consumptions and iteration numbers when different algorithms are used for solving the cases considering handling voltage limits with tap changers action. Here, flat start is applied and other limits are ignored.

Table IV shows that voltage-tap changer ratio switching logic leads more iterations for all these algorithms. However, both Algorithm I and the two-stage algorithm can successfully achieve the solutions for well- and ill-conditioned systems, while Newton's method diverges when solving 'case3375wp'. Besides, the proposed algorithms keep a high convergence rate and computational efficiency both in cases with local control and remote control.

VI. Conclusion

This paper proposes a globally convergent method for PFC. Further, by combining it with Newton's method, a robust and efficient two-stage algorithm for PFC of large-scale systems is proposed. Through extensive demonstration in many cases, the following observations can be obtained:

- The selection of damping factor expression in Algorithm I effectively reduces iterations compared with some previous research achievements. This enhances the convergence rate and efficiency of Algorithm I. The proposed Algorithm I has powerful robustness, even in extremely ill-conditioned systems, operational modes and inappropriate initial values.
- The proposed two-stage algorithm by combining Algorithm I with Newton's method further improves the convergence rate and efficiency. It has universality for well-and ill-conditioned systems as well as cases under ill-conditioned operational modes, heavy loads and inappropriate initial values. Particularly, for the large-scale systems, such as 'case6515rte', 'case9241pegase' and 'case13659pegase', the two-stage algorithm achieves accurate solutions within 1s and 10 iterations from the flat start, and the time consumption is only 10%$\sim$55% of other methods in previous research.
- The proposed Algorithm I and the two-stage algorithm can successfully achieve solutions considering reactive power limits and voltage limits, and they are also well-behaved when remote transformers considered in the systems.

However, there are still some potential drawbacks in the proposed algorithms: i) Due to the nonlinear and nonconvex nature of AC power flow model, it is difficult to theoretically prove that Algorithm I can finally converge to the solution corresponding to real-world system operation, instead of an undesirable solution, such as a saddle point, or a local minimizer of least-square minimization problem which is not the solution of power flow equations. This may further affect the performance of Newton's method if Algorithm I swaps to Stage II; ii) Numerous empirical parameters exist in the proposed algorithms, and tuning them is a big challenge.

Nonetheless, the situation that converging to an undesirable solution did not appear in the numerical tests in this paper, which indicates that the proposed algorithms are reasonably well-behaved, even in large-scale ill-conditioned systems. Besides, based on previous research achievements and numerical experiments, the values of the empirical parameters used in this paper can bring out satisfying performance.

The proposed robust and efficient two-stage algorithm can significantly benefit real practice. Future work will extend it to solve other problems related to PFC, such as contingency analysis, risk assessment, probabilistic load flow, etc.

Acknowledgment

The authors are grateful for the support of Zhejiang University-Wanke New Energy Joint Research Center, Hangzhou, China.

References

[1] F. Milano, "Continuous Newton's method for power flow analysis," *IEEE Trans. Power Syst.*, vol. 24, no. 1, pp. 50–57, Feb. 2009.

[2] W. F. Tinney and C. E. Hart, "Power flow solution by Newton's method," *IEEE Trans. Power App. Syst.*, vol. PAS-86, no. 11, pp. 1449–1460, Nov. 1967.

[3] C. A. Ferreira and V. M. D. Costa, "A second order power flow based on current injection equations," *Int. J. Elect. Power Energy Syst.*, vol. 27, no. 4, pp. 254–263, May 2005.

[4] S. Y. Derakhshandeh and R. Pourbagher, "Application of high-order Newton-like methods to solve power flow equations," *IET Gener., Trans. Distrib.*, vol. 10, no. 8, pp. 1853–1859, May 2016.

[5] Y. Chen and C. Shen, "A Jacobian-free Newton-GMRES(m) method with adaptive preconditioner and its application for power flow calculations," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1096–1103, Aug. 2006.

[6] H. Yang, F. Wen, and L. Wang, "Newton-Raphson on power flow algorithm and Broyden method in the distribution system," in *Proc. 2nd Int. Power Energy Conf.*, 2008, pp. 1613–1618.

[7] R. Pourbagher and S. Y. Derakhshandeh, "A powerful method for solving the power flow problem in the ill-conditioned systems," *Int. J. Elect. Power Energy Syst.*, vol. 94, pp. 88–96, Jan. 2018.

[8] S. Iwamoto and Y. Tamura, "A load flow calculation method for III-conditioned power systems," *IEEE Power Eng. Rev.*, vol. PER-1, no. 4, pp. 44–45, Apr. 1981.

[9] P. J. Lagace, M. H. Vuong, and I. Kamwa, "Improving power flow convergence by Newton Raphson with a Levenberg-Marquardt method," in *Proc. IEEE Power Energy Soc. Gen. Meeting—Convers. Delivery Elect. Energy 21st Century*, 2008, pp. 1–6.

[10] P. J. Lagace, "Power flow methods for improving convergence," in *Proc. 38th Annu. Conf. IEEE Ind. Electron. Society*, 2012, pp. 1387–1392.

[11] 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.

[12] 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., Trans. Distrib.*, vol. 10, no. 12, pp. 3017–3022, Sept. 2016.

[13] S. Y. Derakhshandeh, R. Pourbagher, and A. Kargar, "A novel fuzzy logic Levenberg-Marquardt method to solve the ill-conditioned power flow problem" *Int. J. Elect. Power Energy Syst.*, vol. 99, pp. 299–308, Jul. 2018.

[14] J. Cao, X. Xu, Z. Yan, Y. Li, Z. Yang, and N. Feng, "Application of several levenberg-marquardt methods with higher-order convergence in power flow calculation," *Power Syst. Technol.*, vol. 41, no. 4, pp. 1181–1187, Apr. 2017.

[15] M. Tostado-Véliz, S. Kamel, and F. Jurado, "Development of combined Runge–Kutta Broyden's load flow approach for well- and ill-conditioned power systems," *IET Gener., Trans. Distrib.*, vol. 12, no. 21, pp. 5723–5729, Nov. 2018.

[16] A. Gómez-Expósito and C. Gómez-Quiles, "Factorized load flow," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4607–4614, Nov. 2013.

[17] C. Gómez-Quiles, A. Gómez-Expósito, and W. Vargas, "Computation of maximum loading points via the factored load flow," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 4128–4134, Sep. 2016.

[18] J. Fan and Y. Yuan, "On the convergence of a new Levenberg–Marquardt method," *Mathematica Numerica Sinica*, vol. 27, no. 2, pp. 55–62, Feb. 2005.

[19] N. Yamashita and M. Fukushima, "On the rate of convergence of the Levenberg-Marquardt method," *Computing*, vol. 15, pp. 239–249, 2001.

[20] 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.

[21] J. Fan, "A Shamanskii-like Levenberg-Marquardt method for nonlinear equations," *Comput. Optim. Appl.*, vol. 56, no. 1, pp. 63–80, Oct. 2018.

[22] K. Amini and F. Rostami, "A modified two steps Levenberg–Marquardt method for nonlinear equations," *J. Comput. Appl. Math.*, vol. 288, pp. 341–350, Nov. 2015.

[23] J. Fan and J. Zeng, "A Levenberg–Marquardt algorithm with correction for singular system of nonlinear equations," *Appl. Math. Comput.*, vol. 219, no. 17, pp. 9438–9446, May 2013.

[24] J. Fan and J. Pan, "A note on the Levenberg–Marquardt parameter," *Appl. Math. Comput.*, vol. 207, no. 2, pp. 351–359, Jan. 2009.

[25] M. Ahookhosh and K. Amini, "An efficient nonmonotone trust-region method for unconstrained optimization," *Numer. Algorithms*, vol. 59, no. 4, pp. 523–540, Apr. 2012.

[26] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust Region Methods*, Philadelphia, PA, USA: SIAM, 2000, ch. 2, sec. 6, pp. 113–166; ch. 5, sec. 17, pp. 779–798.

[27] K. R. C. Mamandur and G. J. Berg, "Automatic adjustment of generator voltages in Newton-Raphson method of power flow solutions," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1400–1409, Jun. 1982.

[28] N. M. Peterson and W. S. Meyer, "Automatic adjustment of transformer and phase-shifter taps in the Newton power flow," *IEEE Trans. Power App. Syst.*, vol. PAS-90, no. 1, pp. 103–108, Jan. 1971.

[29] G. A. Maria, A. H. Yuen, and J. A. Findlay, "Control variable adjustment in load flows," *IEEE Trans. Power Syst.*, vol. 3, no. 3, pp. 858–864, Aug. 1988.

Kunjie Tang (S'18) received the B.E degree from the College of Electrical Engineering, Zhejiang University, Hangzhou, China, in 2017, where he is currently working toward the Ph.D. degree.

His research interests include transmission-distribution coordinated optimization and high-performance computing in power systems.

Shufeng Dong received the B.E. and Ph.D. degrees from Tsinghua University, Beijing China, in 2004 and 2009, respectively.

He is currently an Assistant Professor and Qiushi Young Scholar with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. His research interests include power system state estimation, cloud computing, and high-performance computing in power systems.

Jie Shen, photograph and biography not available at the time of publication.

Chengzhi Zhu, photograph and biography not available at the time of publication.

Yonghua Song (M'90–SM'94–F'08) received the B.Eng. degree from the Chengdu University of Science and Technology, Chengdu, China, in 1984, and the Ph.D. degree from the China Electric Power Research Institute, Beijing, China, in 1989.

He is currently a Rector with the University of Macau, Macau, China. He is also a Professor with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macau, China; an Adjunct Professor with the College of Electrical Engineering, Zhejiang University, Hangzhou, China; and an Adjunct Professor with the Department of Electrical Engineering, Tsinghua University, Beijing, China. His research interests include smart grid, electricity economics, and operation and control of power systems.