IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 January 2024, accepted 31 January 2024, date of publication 5 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3362633

RESEARCH ARTICLE

A Bus Privacy Preserving Decentralized Power Flow Algorithm Considering Neighbor Partial Derivative Information

SI-KAI TAN[1], JIAN-CHUN PENG[1], (Senior Member, IEEE), MING-HUAN WU[1], AND HUI JIANG[2]

[1]College of Mechatronic and Control Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China
[2]College of Physical and Optoelectronic Engineering, Shenzhen University, Shenzhen, Guangdong 518060, China

Corresponding author: Hui Jiang (huijiang@szu.edu.cn)

This work was supported by the Shenzhen Science and Technology Innovation Foundation under Grant GJHZ20180928160212241 and Grant JCYJ20190808165201648.

ABSTRACT To ensure the privacy of power system users while maximizing the efficiency of decentralized computational resources, this paper presents a novel decentralized power flow algorithm. Its core focus lies in preserving bus privacy through the integration of neighboring partial derivative information. Initially, the algorithm utilizes partial derivative data from adjacent buses to formulate a bus-level voltage iterative function, rooted in the bus power balance equation. Each bus in the power network is treated as an individual computational unit managed by an agent. This process involves scanning the network's buses, computing estimated bus voltage values using a designated iteration formula, and iterating until convergence is achieved. Subsequent utilization of the latest neighbor information accelerates this iterative process. To fortify the algorithm's reliability, a strategy assessing local convergence, utilizing the Jacobian matrix at the solution's convergence, is proposed. The simulation results across various IEEE test systems unequivocally validate the proposed algorithm's efficacy in power flow computation, showcasing accelerated speed and robust anti-interference capabilities. Compared to the traditional decentralized power flow method at the bus level, this algorithm significantly optimizes the iterative process, slashing the number of iterations by a minimum of 1.36 times. Notably, this efficiency boost amplifies in larger system scales, boasting an impressive reduction of up to 9.21 times fewer iterations.

INDEX TERMS Decentralized power flow, privacy preservation, voltage iterative function, local convergence.

NOMENCLATURE

ABBREVIATIONS

PF	Power flow.
NR	Newton-Raphson method.
GS	Gauss-Seidel method.
ADMM	Alternating directions method of multipliers.
ACPF	Alternating current power flow.
SA	Straightforward Approximation neighbor bus voltage method.

SINGLE VARIABLE

i and j	Numbers of the i-th and j-th bus.
P_i	Active power injection of the i-th bus.
Q_i	Reactive power injection of the i-th bus.
V_i and θ_i	Magnitude and phase of the i-th bus voltage.
θ_{ij}	Phase angle difference $\theta_i - \theta_j$
G_{ij}	Real parts of the ij-th element of the admittance matrix.
B_{ij}	Imaginary parts of the ij-th element of the admittance matrix.
$\mathcal{N}_i$	Set of adjacent buses of the i-th bus.
Bus_Type	Bus type: PQ, PV or Slack bus.
P_load	Active load of the bus.

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

Q_load	Reactive load of the bus.
*G*_parallel	Compensated conductance of the bus.
*B*_parallel	Compensated Susceptance of the bus
P_Gen	Active power generation of the bus.
Q_Gen	Reactive power generation of the bus.
Statue_Gen	Generator running status.
R and *X*	Resistance and Reactance of Lines.
B	Equivalent capacitance of π-type line.
Ratio	Transformer ratio in the line.
Statue_Line	Line running status.
ε	Maximum convergence error.
ΔS	Maximum power unbalance value.

VECTOR VARIABLE

$\boldsymbol{\theta}$	The vector consisting of θ_i.
$\boldsymbol{V}$	The vector consisting of V_i.
$\boldsymbol{J}$	The Jacobian matrix.
$\boldsymbol{D_i}$	The partial derivative of i-th bus with respect to the voltage of i-th bus.
$\boldsymbol{U_{ij}}$	The partial derivative of i-th bus with respect to the voltage of j-th bus.

FUNCTION VARIABLE

$f_{Pi}(\boldsymbol{\theta}, \boldsymbol{V})$	The Active power function of i-th bus.
$f_{Qi}(\boldsymbol{\theta}, \boldsymbol{V})$	The reactive power function of i-th bus.
$o_2(\boldsymbol{\theta}_0, \boldsymbol{V}_0)$	The higher order errors of Taylor expansions.
$\boldsymbol{g}(\boldsymbol{\theta}_k, \boldsymbol{V}_k)$	The overall iterative function of system.
$\boldsymbol{g'}(\boldsymbol{\theta}_*, \boldsymbol{V}_*)$	The Partial derivative of system iteration function at convergence point.
$\rho\left(\boldsymbol{g'}(\boldsymbol{\theta}_*, \boldsymbol{V}_*)\right)$	The Spectral radius of partial derivative of system iteration function at convergence point.

I. INTRODUCTION

A. RESEARCH MOTIVATION

Power flow (PF) is a highly influential computational tool in power system analysis, and it is now widely used in power system planning, operation and control [1], [2], [3], [4]. Essentially, PF is the solution of a set of high-dimensional nonlinear equations for bus voltage, bus injected power, and bus load [5]. It is widely known that the commonly used approach for solving nonlinear equations is Newton-Raphson's method (NR) [6]. As shown in figure 1.(a), the traditional PF mode is that the power system control center collects the injected power and load information of all buses, and then uses the NR method to solve the steady-state voltage value of each bus. Clearly, traditional PF are performed in a centralized manner.

In the future, a large number of power load units, distributed generation, distributed energy storage devices and plug-and-play devices will be added to the power system [7], [8], [24]. At that time, the structure of the power system will become increasingly complex and PF will construct a set of nonlinear equations with a huge computational load. It not only requires the control center of the traditional centralized approach to have an ultra-high-performance solver, but also requires ensuring that the communication lines connecting each bus to the control center are stable and reliable. In addition, more and more generators/consumers are concerned about the leakage of their private information (such as bus injection power and load). It is difficult for a single control center to ensure that their private information will not be leaked [9], [10]. A bus-oriented decentralized PF approach can avoid the above problems. As shown in figure 1.(b), the decentralized approach disperses a large number of computations into each bus, which eliminates the computational burden of the control center and makes full use of the computing resources distributed in the buses. Since there is no control center, there is no need to worry about privacy leaks. The motivation of this paper is to develop a reliable and efficient approach in the direction of bus-level decentralized PF.

B. LITERATURE REVIEW

In the past, the PF has been highly developed. The fast decomposition method [11], [12] improves the calculation speed by modifying the Jacobian matrix to decouple the active PF and reactive PF. the powerful PF [5], [13] transforms the power flow problem into an optimization problem and makes the PF of the ill-conditioned system converge. the Newton-Krylov method [14], [15] uses the least squares idea to approximately solve the inverse of the Jacobian matrix, which reduces the compu-tational complexity of large systems. the higher-order Newton's method [16], [17] constructs a special iteration formula that makes each iteration closer to the correct solution, which improves the computational efficiency of large systems.

In recent years, distributed PF algorithms have also been greatly developed. Numerous academics have worked on developing distributed algorithms at the subsystem level. Their main idea is to decompose the PF of the entire network into centralized PF of several subsystems and iterate repeatedly between the subsystems until the boundary bus PF variables are consistent. References [18] and [19] solve the subsystem PF with a centralized approach, which not only enjoys the fast convergence of a centralized approach, but also prevents the high computational cost brought by the overall operation. References [20] and [21] utilize state estimation to obtain the voltage information of the boundary bus, and set an agent in each subsystem to perform PF. When the voltages of all boundary buses tend to be stable, the agents no longer exchange data and the distributed PF is completed. This method enables real-time distributed

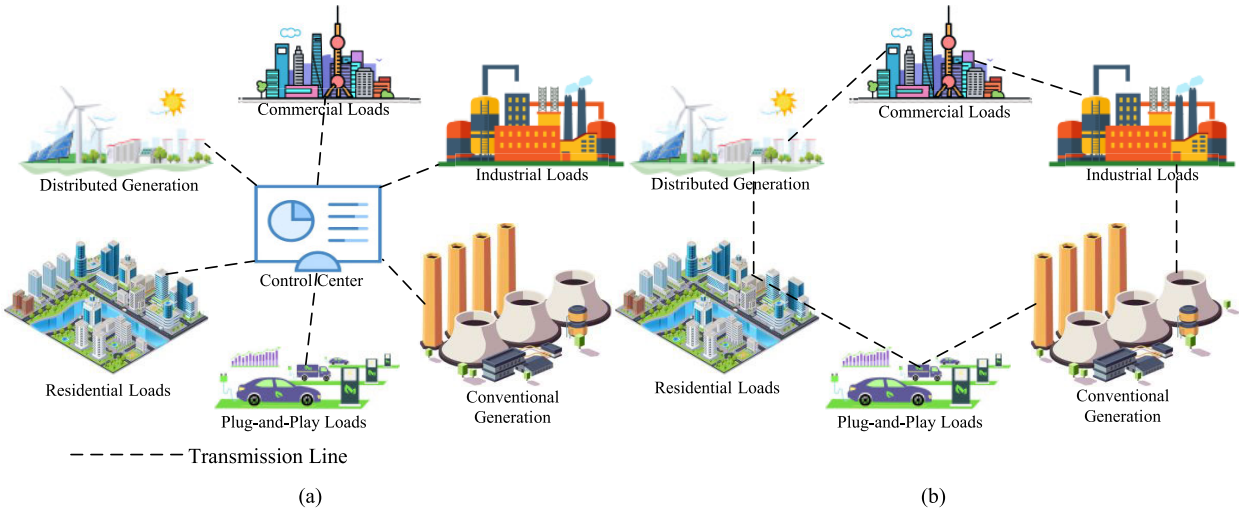


FIGURE 1. Different approaches for power grids. (a) centralized approach; (b) decentralized approach (bus-to-bus).

PF computation. To prevent the private information in the transmission process from being intercepted and deduced by reverse reasoning, [11], [22] adds encryption and decryption algorithms when the subsystem boundary agent transmits data. Reference [23] considers the distributed PF of an ill-conditioned system, transforms the PF model into a convex model whose minimum is the PF solution, and then solves for the minimum of the model via the ADMM method.

The subsystem-level distributed PF algorithm has shown great improvement. However, the subsystem distributed approach is not sufficient to make full use of the computing resources distributed at the generation/load units and does not ensure that each generation/load unit independently participates in the electricity market. Bus-level decentralized algorithms can handle the above problems well. In fact, the traditional Gauss-Seidel method (GS) is a natural bus-level decentralized algorithm [24], but the convergence speed is slow, and the convergence time increases exponentially as the scale of the system grows. Reference [25] propose to set up an agent at each bus to perform backward/forward sweep iterative computation of PF, which requires the PF to be executed in a specific direction, resulting in slower PF computation. Actually, if the subsystem-level distributed PF algorithm sets the subsystem as a bus, its computational form is similar to this approach. Reference [26] proposes to approximately linearize the ACPF equation and then solve the approximate linear equation by bus-level decentralized iteration. This method is able to solve the bus-level PF quickly, but the linear approximation will bring inherent errors to the PF that cannot be eliminated by iteration. Reference [27] constructed the power equation of the bus into two circle equations with the bus voltage as the variable, an intersection of the circles is an approximate solution, and all buses repeatedly solve the intersection until the system converges. Since the iteration is an approximate process, there is no guarantee that each bus will have an intersection at each iteration, and the method will be greatly affected when there are multiple occurrences of two circles that do not intersect. Reference [28] derived the bus-level voltage iterative update formula from a single-bus perspective, which enables bus-level PF computation, but the computation speed is similar to that of GS method, which is difficult to apply in general.

C. ARTICLE CONTRIBUTION

In general, the main contributions of this paper are in the following three aspects:

1. We develop a bus-level decentralized PF algorithm. This algorithm protects the private information of the buses, prevents the central computer from taking all the computational burden, and enables each bus to participate in the PF computation.
2. Using the latest neighbor bus information, we speed up the computation of the proposed algorithm.
3. We present a strategy for judging the local convergence of the proposed algorithm. This strategy requires only the value of the Jacobian at the convergence point.

D. ARTICLE ORGANIZATION

The rest of this article is organized as follows. Section II describes the mathematical model for PF and the existing solutions to the PF problem. Section III presents a bus-level decentralized PF algorithm considering neighbor first-order information. Section IV presents a strategy for analyzing the local convergence of the proposed algorithm. Section V presents the numerical results to show the performance of the proposed algorithm to solve the PF problem, and Section VI presents the conclusions.

II. POWER FLOW PROBLEM

A. MATHEMATICAL MODEL FOR POWER FLOW

For the convenience of description, we assume that the power system has n buses, where the n-th bus is a slack bus and

the rest of the buses are PQ buses. According to Kirchhoff's Current Law and Tellegen's Theorem, the power balance equation of PF in polar coordinates can be as follows

$$\begin{cases} f_{Pi}(\boldsymbol{\theta}, \boldsymbol{V}) = P_i - V_i \sum_{j=1}^{j=n} V_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right) = 0 \\ f_{Qi}(\boldsymbol{\theta}, \boldsymbol{V}) = Q_i - V_i \sum_{j=1}^{j=n} V_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right) = 0 \\ (i = 1, 2, \cdots, n-1) \end{cases} \tag{1}$$

B. CENTRALIZED NEWTON-RAPHSON METHOD FOR POWER FLOW

As mentioned in the introduction, the NR method is currently considered one of the most dominant approaches for PF problems. Its general iterative form for solving PF mathematical equations is as follows.

$$\begin{pmatrix} \boldsymbol{\theta}_{k+1} \\ \boldsymbol{V}_{k+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\theta}_k \\ \boldsymbol{V}_k \end{pmatrix} - \left[\boldsymbol{J}(\boldsymbol{\theta}_k, \mathbf{V}_k)\right]^{-1} \begin{bmatrix} \boldsymbol{f_{Pi}}(\boldsymbol{\theta_k}, \mathbf{V_k}) \\ \boldsymbol{f_{Qi}}(\boldsymbol{\theta_k}, \boldsymbol{V_k}) \end{bmatrix} \tag{2}$$

where $\boldsymbol{J}$ is called the Jacobian matrix, which is the partial derivative of the system of equations (1) with respect to $(\boldsymbol{\theta}_k, \boldsymbol{V}_k)$.

III. BUS-LEVEL DECENTRALIZED POWER FLOW ALGORITHM CONSIDERING NEIGHBOR FIRST-ORDER INFORMATION

The traditional NR method must solve the inverse of the Jacobian, which requires a centralized solution. How to avoid computing the inverse of the Jacobian when solving the PF problem? We believe that this requires deriving a new PF algorithm from the bottom layer (independent bus) of the PF problem object.

A. GENERAL DESCRIPTION OF THE PROPOSED ALGORITHM

Starting from the power balance equation of a single bus. For i-th bus, take out the active and reactive power balance equations based on the i-th bus from formula (1).

$$\begin{cases} f_{Pi}(\boldsymbol{\theta}, \boldsymbol{V}) = P_i - V_i \sum_{j=1}^{j=n} V_j \left(G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}\right) = 0 \\ f_{Qi}(\boldsymbol{\theta}, \boldsymbol{V}) = Q_i - V_i \sum_{j=1}^{j=n} V_j \left(G_{ij}\sin\theta_{ij} - B_{ij}\cos\theta_{ij}\right) = 0 \end{cases} \tag{3}$$

At the initial approximate solution $(\boldsymbol{\theta}_0, \boldsymbol{V}_0)$, there is a first-order Taylor expansion.

$$\begin{cases} f_{Pi}(\boldsymbol{\theta}, \boldsymbol{V}) = f_{Pi}(\boldsymbol{\theta}_0, \boldsymbol{V}_0) + \frac{\partial f_{Pi}}{\partial \theta_{i,0}}\left(\theta_i - \theta_{i,0}\right) \\ + \frac{\partial f_{Pi}}{\partial V_{i,0}}\left(V_i - V_{i,0}\right) + \sum_{j\in\mathcal{N}_i} \frac{\partial f_{Pi}}{\partial \theta_{j,0}}\left(\theta_j - \theta_{j,0}\right) \\ + \sum_{j\in\mathcal{N}_i} \frac{\partial f_{Pi}}{\partial V_{j,0}}\left(V_j - V_{j,0}\right) + o_2(\boldsymbol{\theta}_0, \boldsymbol{V}_0) = 0 \\ f_{Qi}(\boldsymbol{\theta}, \boldsymbol{V}) = f_{Qi}(\boldsymbol{\theta}_0, \boldsymbol{V}_0) + \frac{\partial f_{Qi}}{\partial \theta_{i,0}}\left(\theta_i - \theta_{i,0}\right) \\ + \frac{\partial f_{Qi}}{\partial V_{i,0}}\left(V_i - V_{i,0}\right) + \sum_{j\in\mathcal{N}_i} \frac{\partial f_{Qi}}{\partial \theta_{j,0}}\left(\theta_j - \theta_{j,0}\right) \\ + \sum_{j\in\mathcal{N}_i} \frac{\partial f_{Qi}}{\partial V_{j,0}}\left(V_j - V_{j,0}\right) + o_2(\boldsymbol{\theta}_0, \boldsymbol{V}_0) = 0 \end{cases} \tag{4}$$

Ignore the error $o_2(\boldsymbol{\theta}_0, \boldsymbol{V}_0)$ caused by the first-order Taylor approximation, and rewrite formula (4) into matrix form.

$$\begin{bmatrix} f_{Pi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \\ f_{Qi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{i,0}} & \frac{\partial f_{Pi}}{\partial V_{i,0}} \\ \frac{\partial f_{Qi}}{\partial \theta_{i,0}} & \frac{\partial f_{Qi}}{\partial V_{i,0}} \end{bmatrix} \left(\begin{bmatrix} \theta_i \\ V_i \end{bmatrix} - \begin{bmatrix} \theta_{i,0} \\ V_{i,0} \end{bmatrix} \right) + \sum_{j\in\mathcal{N}_i} \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{j,0}} & \frac{\partial f_{Pi}}{\partial V_{j,0}} \\ \frac{\partial f_{Qi}}{\partial \theta_{j,0}} & \frac{\partial f_{Qi}}{\partial V_{j,0}} \end{bmatrix} \left(\begin{bmatrix} \theta_j \\ V_j \end{bmatrix} - \begin{bmatrix} \theta_{j,0} \\ V_{j,0} \end{bmatrix} \right) = 0 \tag{5}$$

For simplicity of description, we set

$$\boldsymbol{D_{i,0}} = \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{i,0}} & \frac{\partial f_{Pi}}{\partial V_{i,0}} \\ \frac{\partial f_{Qi}}{\partial \theta_{i,0}} & \frac{\partial f_{Qi}}{\partial V_{i,0}} \end{bmatrix} \leftarrow \boldsymbol{D_{i,k}} = \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{i,k}} & \frac{\partial f_{Pi}}{\partial V_{i,k}} \\ \frac{\partial f_{Qi}}{\partial \theta_{i,k}} & \frac{\partial f_{Qi}}{\partial V_{i,k}} \end{bmatrix} \tag{6}$$

$$\boldsymbol{U_{ij,0}} = \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{j,0}} & \frac{\partial f_{Pi}}{\partial V_{j,0}} \\ \frac{\partial f_{Qi}}{\partial \theta_{j,0}} & \frac{\partial f_{Qi}}{\partial V_{j,0}} \end{bmatrix} \leftarrow \boldsymbol{U_{ij,k}} = \begin{bmatrix} \frac{\partial f_{Pi}}{\partial \theta_{j,k}} & \frac{\partial f_{Pi}}{\partial V_{j,k}} \\ \frac{\partial f_{Qi}}{\partial \theta_{j,k}} & \frac{\partial f_{Qi}}{\partial V_{j,k}} \end{bmatrix} \tag{7}$$

Formula (5) becomes the following simplified form

$$\begin{bmatrix} f_{Pi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \\ f_{Qi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \end{bmatrix} + \boldsymbol{D_{i,0}} \left(\begin{bmatrix} \theta_i \\ V_i \end{bmatrix} - \begin{bmatrix} \theta_{i,0} \\ V_{i,0} \end{bmatrix} \right) + \sum_{j\in\mathcal{N}_i} \boldsymbol{U_{ij,0}} \left(\begin{bmatrix} \theta_j \\ V_j \end{bmatrix} - \begin{bmatrix} \theta_{j,0} \\ V_{j,0} \end{bmatrix} \right) = 0 \tag{8}$$

Move the variables to be solved $((\theta_i, V_i)$ and $(\theta_j, V_j))$ in (8) to the left side of the formula.

$$\boldsymbol{D_{i,0}} \begin{bmatrix} \theta_i \\ V_i \end{bmatrix} + \sum_{j\in\mathcal{N}_i} \boldsymbol{U_{ij,0}} \left(\begin{bmatrix} \theta_j \\ V_j \end{bmatrix} \right) = \boldsymbol{D_{i,0}} \begin{bmatrix} \theta_{i,0} \\ V_{i,0} \end{bmatrix} + \sum_{j\in\mathcal{N}_i} \boldsymbol{U_{ij,0}} \left(\begin{bmatrix} \theta_{j,0} \\ V_{j,0} \end{bmatrix} \right) - \begin{bmatrix} f_{Pi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \\ f_{Qi}(\boldsymbol{\theta_0}, \boldsymbol{V_0}) \end{bmatrix} \tag{9}$$

Next, we denote the known initial value as the k-th step approximate solution, and denote the value to be solved as the $k+1$-th step approximate solution, then formula (9) becomes the following iterative form.

$$\boldsymbol{D_{i,k}}\begin{bmatrix}\theta_{i,k+1}\\ V_{i,k+1}\end{bmatrix}+\sum_{j\in\mathcal{N}_i}\boldsymbol{U_{ij,k}}\left(\begin{bmatrix}\theta_{j,k+1}\\ V_{j,k+1}\end{bmatrix}\right)=\boldsymbol{D_{i,k}}\begin{bmatrix}\theta_{i,k}\\ V_{i,k}\end{bmatrix}+\sum_{j\in\mathcal{N}_i}\boldsymbol{U_{ij,k}}\left(\begin{bmatrix}\theta_{j,k}\\ V_{j,k}\end{bmatrix}\right)-\begin{bmatrix}f_{Pi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\\ f_{Qi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\end{bmatrix}\quad(10)$$

Before each iteration, the subscript containing k is the known variable, and the subscript containing $k+1$ is the variable waiting to be solved. Clearly, formula (10) has only two equations; However, there are more than two variables subscripted by $k+1$. These variables are the voltage magnitude and phase of the i-th bus $\left(\theta_{i,k+1}, V_{i,k+1}\right)$, and the voltage magnitude and phase of the j-th bus $\left(\theta_{j,k+1}, V_{j,k+1}\right)$, respectively.

Since iteration is a process of approximating the exact point from the initial point, the solution at each step in iteration is an estimated solution. In order to solve the problem of more unknown variables than equations, we intend to take an approximation to convert the voltage variable of the neighbor bus $\left(\theta_{j,k+1}, V_{j,k+1}\right)$ to a known estimate.

B. STRAIGHTFORWARD APPROXIMATION

The simplest and most straightforward approximation is to replace the current value with the previous value of the neighbor bus. Namely, the neighbor bus voltage variable whose subscript contains $k+1$ $\left(\theta_{j,k+1}, V_{j,k+1}\right)$ in formula (10) is directly approximated to the known approximation $\left(\theta_{j,k}, V_{j,k}\right)$ obtained from the previous iteration.

$$\begin{bmatrix}\theta_{i,k+1}\\ V_{i,k+1}\end{bmatrix}\approx\begin{bmatrix}\theta_{i,k}\\ V_{i,k}\end{bmatrix}\quad(11)$$

Neglecting the error caused by the estimation and substituting formula (11) to formula (10)

$$\begin{bmatrix}\theta_{i,k+1}\\ V_{i,k+1}\end{bmatrix}=\begin{bmatrix}\theta_{i,k}\\ V_{i,k}\end{bmatrix}-\boldsymbol{D_{i,k}^{-1}}\begin{bmatrix}f_{Pi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\\ f_{Qi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\end{bmatrix}\quad(12)$$

Formula (12) is the iterative formula for the i-th bus voltage derived by straightforwardly estimating the neighbor bus. At this time, the variables to be solved in the iterative formula are consistent with the number of equations, and when all buses update the voltage through the formula (12) to meet the convergence condition, the PF calculation ends.

Although the straightforward approximation method is very simple, it will slow down iterations due to too much information being ignored in the estimation. Next, we preserve the first-order information of the neighbors during estimation to speed up the iteration.

C. CONSIDER FIRST-ORDER NEIGHBOR PARTIAL DERIVATIVE INFORMATION APPROXIMATION

The straightforward approximation is available for all buses in the system. If we regard formula (11) as the iterative formula of the j-th bus, we can get

$$\begin{bmatrix}\theta_{j,k+1}\\ V_{j,k+1}\end{bmatrix}=\begin{bmatrix}\theta_{j,k}\\ V_{j,k}\end{bmatrix}-\boldsymbol{D_{j,k}^{-1}}\begin{bmatrix}f_{Pj}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\\ f_{Qj}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\end{bmatrix}\quad(13)$$

Let us return to the question of how to approximate the neighbor bus voltage. Certainly, when the j-th bus is a neighbor bus, formula (13) can be considered as an approximation of the neighbor bus. Substituting formula (13) into formula (10) can obtain a fresh iterative formula for updating the i-th bus voltage.

$$\begin{bmatrix}\theta_{i,k+1}\\ V_{i,k+1}\end{bmatrix}=\begin{bmatrix}\theta_{i,k}\\ V_{i,k}\end{bmatrix}+\sum_{j\in\mathcal{N}_1}\boldsymbol{D_{i,k}^{-1}U_{ij,k}D_{j,k}^{-1}}\begin{bmatrix}f_{Pj}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\\ f_{Qj}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\end{bmatrix}-\boldsymbol{D_{i,k}^{-1}}\begin{bmatrix}f_{Pi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\\ f_{Qi}\left(\boldsymbol{\theta_k},\boldsymbol{V_k}\right)\end{bmatrix}\quad(14)$$

According to formula (6) and (7), it can be known that $\boldsymbol{D_{j,k}}$ and $\boldsymbol{U_{ij,k}}$ are the first-order information of neighbors, and they can provide a more accurate correction direction for the iterative formula of the i-th bus voltage. Therefore, (14) is a more accurate iterative formula for the i-th bus voltage.

From the iterative formula (14), it can be seen that the iterative formula for solving the latest voltage of the i-th bus only includes the variables in the i-th bus ($\theta_{i,k}, V_{i,k}, \boldsymbol{D_{i,k}}$ and $\boldsymbol{U_{ij,k}}$) and the non-private variables of the neighbor bus ($\boldsymbol{\theta_k}, \boldsymbol{V_k}$ and $\boldsymbol{D_{j,k}}$). Thus, the voltage iterative formula considering the first-order information of neighbors is a simple iterative algorithm that can preserve privacy.

It is worth mentioning that it is feasible to take formula (14) as the iterative formula of the neighbor bus and then substitute it into (10) to obtain a more accurate and faster iterative formula. However, the obtained voltage update formula contains the information of the non-neighbor bus, which not only increases the communication burden, but also makes the decentralized algorithms become more complex. We choose to keep formula (14) and use other ways to speed it up.

D. ITERATIVE ACCELERATION

Noting the formula (14), it can be found that the subscript of the neighboring bus is always the k-th step. Although decentralized decoupling computation has been achieved, all buses require synchronous computation, which poses a huge challenge to communication and computing speed.

In fact, when a bus performs formula (14), its neighbor bus may have acquired the latest iteration value (subscript value greater than k). Since the iteration is a process of approximating the exact solution, the iterative value greater than the k-th step is closer to the exact solution. If we put the newly obtained iterative solution of the neighbor bus into the local iteration, we will get a voltage iteration formula with better convergence. In other words, as soon as the neighbor bus gets the latest voltage value, the local bus applies this latest value to the iteration. Simulating in the laboratory, this idea is that when $j < i$, the neighbor bus j takes the value of the $k+1$-th step, otherwise it takes the value of the kth step. In the following, we apply this accelerated idea to give

an accelerated voltage iteration formula.

$$\begin{bmatrix}\theta_{i,k+1}\\ V_{i,k+1}\end{bmatrix}=\begin{bmatrix}\theta_{i,k}\\ V_{i,k}\end{bmatrix}+\sum_{j\in\mathcal{N}_1}\boldsymbol{D}_{i,\tilde{k}}^{-1}\boldsymbol{U}_{ij,\tilde{k}}\boldsymbol{D}_{j,\tilde{k}}^{-1}\begin{bmatrix}f_{Pj}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)\\ f_{Qj}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)\end{bmatrix}-\boldsymbol{D}_{i,\tilde{k}}^{-1}\begin{bmatrix}f_{Pi}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)\\ f_{Qi}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)\end{bmatrix}\tag{15}$$

where

$$\tilde{k}=\begin{cases}k+1 & for\ j<i\\ k & for\ j>i\end{cases}\tag{16}$$

In the previous description, for the sake of uniformity, we assumed that there is no PV bus in the power network. In fact, the voltage iterative derivation process for the PV bus is similar to the PQ bus. Below, we present the iterative formulation of the PV bus.

$$\theta_{i,k+1}=\theta_{i,k}+\sum_{j\in\mathcal{N}_1}D_{i,\tilde{k}}^{-1}\tilde{\boldsymbol{U}}_{\boldsymbol{ij},\tilde{\boldsymbol{k}}}\tilde{\boldsymbol{D}}_{\boldsymbol{j},\tilde{\boldsymbol{k}}}^{-1}\tilde{\boldsymbol{f}}_{\boldsymbol{j}}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)-D_{i,\tilde{k}}^{-1}f_{Pi}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)\tag{17}$$

where

$$D_{i,\tilde{k}}=\frac{\partial f_{Pi}}{\partial\theta_{i,\tilde{k}}}\tag{18}$$

$$\tilde{\boldsymbol{U}}_{\boldsymbol{ij},\tilde{\boldsymbol{k}}}=\begin{cases}\begin{bmatrix}\frac{\partial f_{Pi}}{\partial\theta_{j,\tilde{k}}} & \frac{\partial f_{Pi}}{\partial V_{j,\tilde{k}}}\end{bmatrix} & \text{for the } j\text{th bus is PQ bus}\\ \frac{\partial f_{Pj}}{\partial\theta_{j,\tilde{k}}} & \text{for the } j\text{th bus is PV bus}\end{cases}\tag{19}$$

$$\tilde{\boldsymbol{D}}_{\boldsymbol{j},\tilde{\boldsymbol{k}}}=\begin{cases}\begin{bmatrix}\frac{\partial f_{Pj}}{\partial\theta_{j,\tilde{k}}} & \frac{\partial f_{Pj}}{\partial V_{j,\tilde{k}}}\\ \frac{\partial f_{Qj}}{\partial\theta_{j,\tilde{k}}} & \frac{\partial f_{Qj}}{\partial V_{j,\tilde{k}}}\end{bmatrix} & \text{for the } j\text{th bus is PQ bus}\\ \frac{\partial f_{Pj}}{\partial\theta_{j,\tilde{k}}} & \text{for the } j\text{th bus is PV bus}\end{cases}\tag{20}$$

$$\tilde{\boldsymbol{f}}_{\boldsymbol{j}}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)=\begin{cases}\begin{bmatrix}f_{Pj}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)\\ f_{Qj}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)\end{bmatrix} & \text{for the } j\text{th bus is PQ bus}\\ f_{Pj}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right) & \text{for the } j\text{th bus is PV bus}\end{cases}\tag{21}$$

E. ALGORITHM OPERATION PROCESS

In figure 2, we give a diagram of the iterative process of bus-level PF calculation for a five-bus system. It should be noted that five buses need to be configured with the necessary computing resources. If a local bus cannot provide any computing resources, it can seek assistance from neighbor buses as agent. The agent can establish an independent computing unit for the local bus to join the decentralized PF calculation, or it can combine their information to perform the decentralized PF calculation.

Returning to figure 2, we can see that when all buses complete calculations according to formula (14), the entire system completes one iteration. Repeat the iteration until the convergence condition $\{|f_{P1}|,\cdots|f_{P(n-1)}|,|f_{Q1}|\cdots|f_{Q(n-1)}|\}<1e-6$ is satisfied (equivalent to the maximum error of the bus power under the actual value is less than 0.1kW). Currently, the solution is considered as the final PF solution.

Algorithm 1 Bus-Privacy-Preserving Decentralized Power Flow Algorithm Considering Neighbor Partial Derivative Information

Bus Data: *Bus_Type*, *P_load*, *Q_load*, *G*_parallel and *B*_parallel

1 **Reading**
Generator Data: *P_Gen*, *Q_Gen*, *Statue_Gen*
Line Data: *R*, *X*, B, *Ratio*, *Statue_Line*

2 **Pre-compute** G_{ij}, B_{ij}, P_i and Q_i for all buses based on reading data

3 **Initializing** $\theta_{i,0}=0, V_{i,0}=1$ for all buses, set $\varepsilon=10^{-6}$, $\Delta S=1$ and $k=1$

4 **While** $(\Delta S>\varepsilon)$ **do**

5 **For** $i=1,\ldots,n-1$

6 $f_{Pi}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)$ and $f_{Qi}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)$ ← Solve by (3) and (16)

7 **If** *Bus_Type is PQ*

8 $\boldsymbol{D}_{\boldsymbol{i},\tilde{\boldsymbol{k}}}$, $\boldsymbol{D}_{\boldsymbol{j},\tilde{\boldsymbol{k}}}$ and $\boldsymbol{U}_{\boldsymbol{ij},\tilde{\boldsymbol{k}}}$ ← Solve by (6), (7) and (16)

9 $f_{Pj}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)$ and $f_{Qj}\left(\boldsymbol{\theta}_{\tilde{k}},\boldsymbol{V}_{\tilde{k}}\right)$ ← Solve by (3) and (16)

10 $\theta_{i,k}, V_{i,k}$ ← Solve by (15)

11 **Else if** *Bus_Type* **is** *PV*

12 $D_{i,\tilde{k}}$, $\tilde{\boldsymbol{U}}_{\boldsymbol{ij},\tilde{\boldsymbol{k}}}$ and $\tilde{\boldsymbol{D}}_{\boldsymbol{j},\tilde{\boldsymbol{k}}}$ ← Solve by (16) and (18)-(20)

13 $\tilde{\boldsymbol{f}}_{\boldsymbol{j}}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}}\right)$ ← Solve by (3), (16) and (21)

14 $\theta_{i,k}$ ← Solve by (17)

15 **End If**

16 $k=k+1$

17 **End For**

18 $\Delta S=\mathrm{Max}(\boldsymbol{f}_{\boldsymbol{P}}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}+\boldsymbol{1}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}+\boldsymbol{1}}\right),\boldsymbol{f}_{\boldsymbol{Q}}\left(\boldsymbol{\theta}_{\tilde{\boldsymbol{k}}+\boldsymbol{1}},\boldsymbol{V}_{\tilde{\boldsymbol{k}}+\boldsymbol{1}}\right))$

19 **End while**

20 **Return** $V_i, \theta_i \forall$ buses $i=1,\ldots,n-1$;

IV. CONVERGENCE ANALYSIS

In the field of mathematical analysis, research on the convergence of nonlinear equations focuses on local convergence and semi-local convergence. Due to the uncertainty in the iterative solution of nonlinear equations, global convergence is difficult to confirm. As far as we know, for PF with nonlinear equation properties, except for the NR method with a special iterative form that can verify the semi-local convergence property from the initial point, other methods can only verify local convergence.

The proposed algorithm can verify local convergence. Local convergence is the analysis of why the algorithm can achieve iterative convergence when the solution to the system of equations is known. If the algorithm has local convergence in a known system, the algorithm is relatively reliable. For unknown systems, it is likely to be able to find solutions as well.

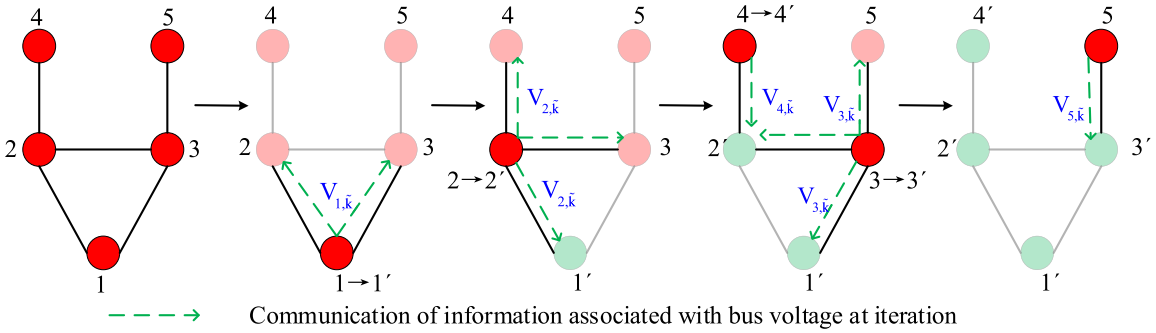


FIGURE 2. A five-bus system demonstrates the proposed voltage iteration algorithm.

Next, we give a local convergence proof for the decentralized PF algorithm proposed in this paper. Convergence is a characteristic of the whole system, so it is necessary to analyze its convergence based on the iterative equation of the whole system. The following is a $2n-2$ equation system formed by formula (15) when $i = 1, 2, \cdots, n-1$ to analyze the convergence of the proposed algorithm.

$$(\boldsymbol{\theta}_k, \boldsymbol{V}_k) = \begin{pmatrix} \theta_{1,k+1} \\ V_{1,k+1} \\ \vdots \\ \theta_{i,k+1} \\ V_{i,k+1} \\ \vdots \\ \theta_{n-1,k+1} \\ V_{n-1,k+1} \end{pmatrix} = \begin{pmatrix} \theta_{1,k} \\ V_{1,k} \\ \vdots \\ \theta_{i,k} \\ V_{i,k} \\ \vdots \\ \theta_{n-1,k} \\ V_{n-1,k} \end{pmatrix} - \left(\boldsymbol{D}_k^{-1}\boldsymbol{U}_k + \boldsymbol{I}\right)\boldsymbol{D}_k^{-1} \begin{bmatrix} f_{P1}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \\ f_{Q1}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \\ \vdots \\ f_{Pi}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \\ f_{Qi}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \\ \vdots \\ f_{P(n-1)}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \\ f_{P(n-1)}(\boldsymbol{\theta}_k, \boldsymbol{V}_k) \end{bmatrix} \tag{22}$$

where $\boldsymbol{D}_k$ is a diagonal matrix whose elements are the matrix $\boldsymbol{D}_{i,k}$ in formula (6).

$$\boldsymbol{D}_k = \begin{bmatrix} \boldsymbol{D}_{1,k} & & & & \\ & \ddots & & & \\ & & \boldsymbol{D}_{i,k} & & \\ & & & \ddots & \\ & & & & \boldsymbol{D}_{n-1,k} \end{bmatrix}_{(n-1)\times(n-1)} \tag{23}$$

$\boldsymbol{U}_k$ satisfies the following formula

$$\boldsymbol{U}_k = \boldsymbol{D}_k - \boldsymbol{J} \tag{24}$$

where $\boldsymbol{J}$ is the Jacobian matrix in formula (2).

According to Ostrowski's theorem [29], at the exact solution $(\boldsymbol{\theta}_*, \boldsymbol{V}_*)^T$, if the spectral radius of the first-order partial derivative matrix of the iterative function $\boldsymbol{g}(\boldsymbol{\theta}_k, \boldsymbol{V}_k)$ is less than 1, the iterative function has local convergence. According to formula (22), for the proposed algorithm in this paper, the first-order partial derivative of the iterative function at the exact solution is:

$$\boldsymbol{g}'(\boldsymbol{\theta}_*, \boldsymbol{V}_*) = \boldsymbol{I} - \left(\boldsymbol{D}_*^{-1}\boldsymbol{U}_* + \boldsymbol{I}\right)\boldsymbol{D}_*^{-1}\boldsymbol{J}_* \tag{25}$$

Now, if we solve the spectral radius $\rho\left(\boldsymbol{g}'(\boldsymbol{\theta}_*, \boldsymbol{V}_*)\right) < 1$, we can infer that the algorithm has local convergence. Next, we look for an equivalent solution to $\rho\left(\boldsymbol{g}'(\boldsymbol{\theta}_*, \boldsymbol{V}_*)\right) < 1$.

First, construct an $\boldsymbol{H}$ matrix

$$\boldsymbol{H}_* = \boldsymbol{D}_*^{-1}\boldsymbol{U}_* = \boldsymbol{D}_*^{-1}(\boldsymbol{D}_* - \boldsymbol{J}_*) = \boldsymbol{I} - \boldsymbol{D}_*^{-1}\boldsymbol{J}_* \tag{26}$$

By (26) we can get

$$\boldsymbol{D}_*^{-1}\boldsymbol{J}_* = \boldsymbol{I} - \boldsymbol{H}_* \tag{27}$$

Substitute equation (27) into equation (25)

$$\boldsymbol{g}'(\boldsymbol{\theta}_*, \boldsymbol{V}_*) = \boldsymbol{H}_*^2 \tag{28}$$

Furthermore, it is worth mentioning that the spectral radius is the largest eigenvalue of the matrix, so there is the following relationship.

$$\rho\left(\boldsymbol{H}_*^2\right) < \rho(\boldsymbol{H}_*) < 1 \tag{29}$$

To sum up, if $\rho(\boldsymbol{H}_*) < 1$, it can be proved that the algorithm proposed in this paper has local convergence. We give $\rho(\boldsymbol{H}_*)$ of some test cases in the appendix, and it can be found that they all satisfy the condition of $\rho(\boldsymbol{H}_*) < 1$.

V. CASE STUDIES

In this section, we conduct simulation analysis to test the speed performance and anti-disturbance performance of the pro-posed algorithm (PA). We performed simulations using IEEE test cases from the matpower database [30], where these systems included the common test systems (14, 30, 39, 57bus systems), the radiation test systems (33, 69 bus systems) and the larger test systems (118, 300, 1354pegase, 2383wp case system). We also show the results calculated by the Gauss-Seidel method (GS) [24] and the Straightforward

Approximation neighbor bus voltage method (SA) for the above test system. All simulations have been done under Windows 10 on a 3.20GHz Intel Core i5-6500 CPU personal computer (4-GB RAM). It is worth mentioning that, in order to avoid the influence of other computing activities, the results reported for the larger test system are obtained as the average of 100 simulations, and the results reported for the remaining system are obtained as the average of 1000 simulations.

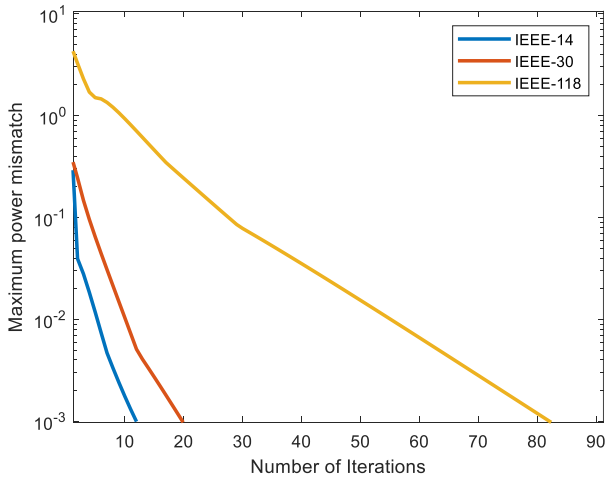


FIGURE 3. Semi-log plot of the maximum power mismatch for the standard 14-, 30-, 118- IEEE bus systems under PA operation.

A. SPEED PERFORMANCE OF PROPOSED ALGORITHM

First, we study the convergence speed of the proposed algorithm at low precision ($\varepsilon = 10^{-3}$) on standard IEEE-14, −30, −118 systems. As shown in figure 3, the proposed algorithm completes convergence within dozens of iterations, where the computation time is 0.005s, 0.018s and 0.355s for IEEE-14, −30- and −118 systems, respectively. Clearly, when computing commonly used test systems, the proposed algorithm is able to complete the computation quickly.

Next, at high precision ($\varepsilon = 10^{-6}$), we compare the number of iterations and computation time between the PA and other approaches. It needs to be acknowledged that the traditional centralized NR method has fewer iterations and shorter calculating time than the PA. Its excellent iterative effect is achieved by sacrificing the privacy information of all buses and letting the central control bear all the computational burden. The starting point of our study is precisely to avoid these shortcomings.

For comparison with bus-level decentralized PF approaches, we show in Table 1 the number of iterations and the ratio of PA reductions for PA, GS, and SA. Comparing the number of iterations, it is easy to see that PA has considerably fewer iterations than GS and SA, regardless of the system size. The reason why PA has fewer iterations is that it considers first-order neighbor bus information further than existing methods. The first-order neighbor bus information reflects the tendency of the bus voltage to change toward a steady state.Taking them into account during the iteration will enable the voltage iteration to be updated in a more accurate direction. Thus, PA can reduce the number of iterations by a large amount.

By analyzing the ratio of PA reductions, it can be found that larger test systems have larger reduction ratios, while radiation test systems have smaller reduction ratios. This implies that PA can better reflect its speed advantage in calculations of large-scale systems. For the radiation test system, the reduction is relatively minor due to its special structure (the system structure is not closed loop and each bus is only conn- ected to adjacent numbered buses), which makes it difficult to reflect the acceleration effect of the decentralized algorithm.

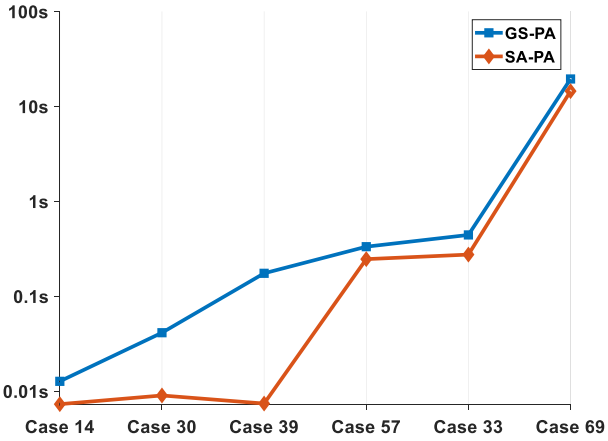


FIGURE 4. Curves with less computation time for GS and SA compared to PA.

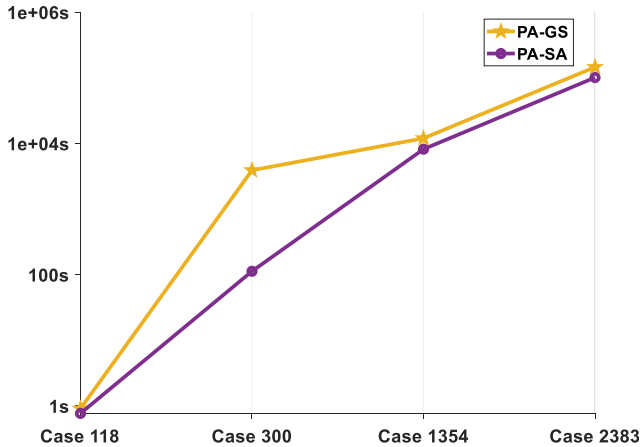


FIGURE 5. Curves with less computation time for PA compared to GS and SA.

Then, in figure 4 and 5, we show the difference between the PA and GS, SA computation times. For common test systems and radiation test systems, figure 4 shows that the conven-tional approach takes less time to compute these systems. Executing the simulation on a single computer cannot distribute the computational burden of the first-order

TABLE 1. The number of iterations calculated by GS, SA and PA under different test systems and the ratio of PA reductions.

System		Number of iterations			Ratio of PA reductions	
		GS	SA	PA	GS/PA	SA/PA
common test systems	Case 14	179	176	55	3.25	3.2
	Case 30	473	448	86	5.5	5.21
	Case 39	1759	1789	353	4.98	5.07
	Case 57	590	588	338	1.75	1.74
radiation test systems	Case 33	1317	1383	763	1.73	1.81
	Case 69	23711	23751	17449	1.36	1.36
larger test systems	Case 118	2046	2009	463	4.42	4.34
	Case 300	22512	22194	2717	8.29	8.17
	Case 1354	77975	75034	8462	9.21	8.87
	Case 2383	323891	313100	54452	5.95	5.75

information of the neighbor bus, so the calculation time of PA is longer than that of the existing method. For large test systems, figure 5 shows that PA computes these systems in less time, and the time reduction is greater as the system size increases. Although PA requires additional calculations for first-order information, it reduces the number of iterations too much, which makes the entire process calculation time faster than existing methods.

B. ACCURACY ANALYSIS OF PROPOSED ALGORITHM

In order to verify that PA can accurately calculate the PF solution, we use the solution of NR method as a stable PF solution, and subtract the voltage solution calculated by PA. If the deviation value is within the acceptable range, the calculation result of PA can be considered to be accurate enough.

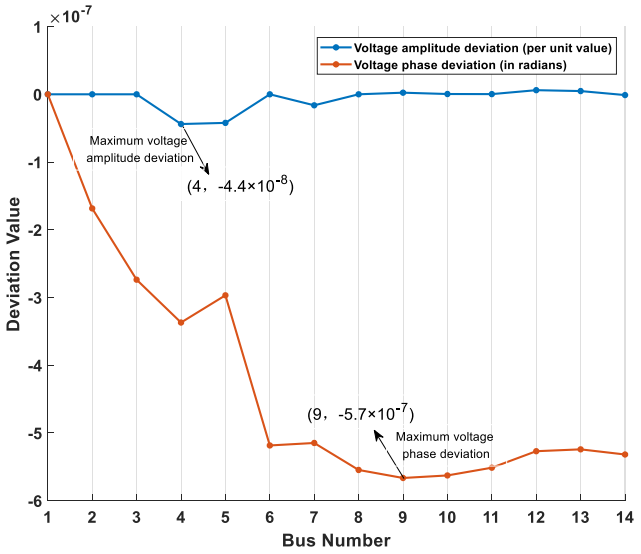


FIGURE 6. The deviation values of PA and NR when calculating the IEEE-14 bus system.

Firstly, taking the IEEE-14 bus system as an example, calculate the PF solutions for all buses using NR and PA, and then subtract the two solutions. figure 6 shows the deviation values of PA and NR when calculating the IEEE-14 bus system.The maximum voltage amplitude and phase deviation in figure 6 are 4.4×10^{-8} and 5.7×10^{-7}, respectively, which is equivalent to an error of 4.4W and 57W in practice. For a MW-level power grid, the error is very small. Therefore, the calculation results of the PA and NR methods can be considered to be equal.

Furthermore, we followed a similar strategy to obtain the maximum deviation values for other IEEE testing systems, as shown in Table 2.

TABLE 2. The $\rho(H_*)$ value of test system.

IEEE Test Systems	maximum voltage amplitude deviation	maximum voltage phase deviation
Case 30	4.5×10^{-8}	6.9×10^{-7}
Case 39	3.8×10^{-9}	1.3×10^{-7}
Case 57	1.1×10^{-7}	4.3×10^{-7}
Case 33	3.6×10^{-7}	2.2×10^{-7}
Case 69	2.5×10^{-7}	6.4×10^{-7}
Case 118	1.0×10^{-8}	7.0×10^{-7}
Case 300	6.7×10^{-7}	3.4×10^{-7}
Case 1354	3.3×10^{-6}	6.6×10^{-6}
Case 2383	1.4×10^{-6}	9.6×10^{-6}

In Table 2, the largest error value is 9.6×10^{-6}, which is obviously small enough. Considering various types of systems, the calculation results of the PA method and the NR method are the same. Therefore, it can be inferred that the calculation of the PA is sufficiently accurate.

C. ANTI-DISTURBANCE PERFORMANCE OF PROPOSED ALGORITHM

The actual power flow calculation process may be affected by the instability of renewable energy generation, uncertainty of electric vehicle loads, and the impact of extreme weather on the power grid. These negative effects can lead to fluctuations in generation/load or changes in the power network topology. Whereas the centralized PF approach requires re-computation to solve such problems, the bus-level decentralized PF algorithm can overcome the disturbance during computation and iterate each bus voltage in a new steady-state direction. As depicted in figure 7, We imposed some actual possible situations in the IEEE-14 bus system: adding a new bus, outage of a branch with a load, and changing the load demand of some buses. It should be noted that to reflect the perturbation in the calculation process,

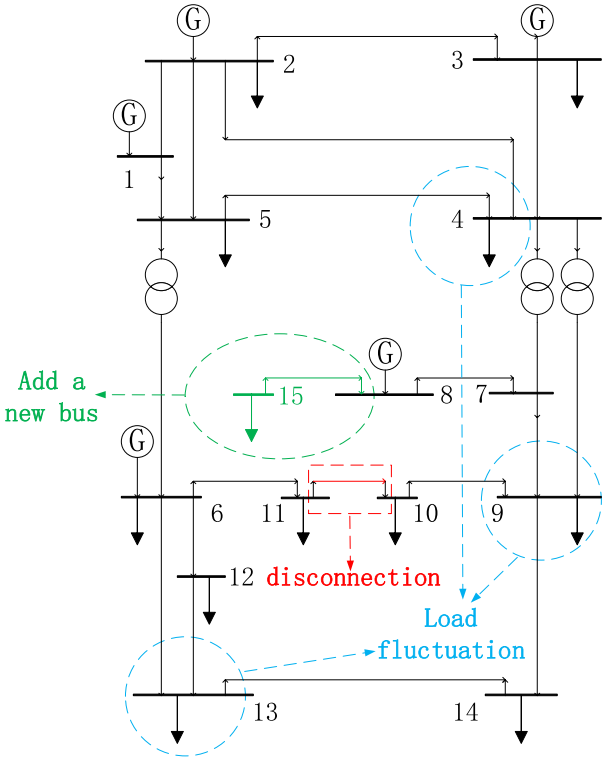


FIGURE 7. Wiring diagram of IEEE-14 bus system with disturbance.

we implemented the perturbation at the 40th iteration (a standard IEEE-14 bus test system requires 55 iterations under the PA algorithm operation)

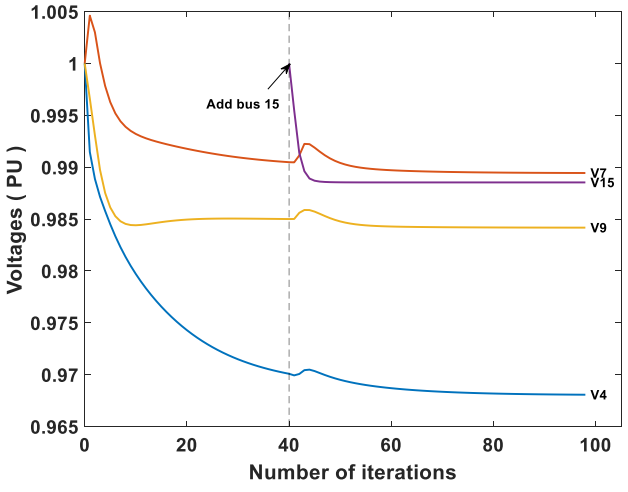


FIGURE 8. Dynamic graph of voltage amplitude in IEEE-14 system with bus added under PA operation.

1) ADDING A NEW BUS

We added a PQ bus with P = 3.5MW, Q = 1.8MVAR to the IEEE-14 bus system and monitored the voltage changes of the bus to verify the ability of the PA to handle the changes in the power network structure caused by the addition of the bus. Figures 8 and 9 take some buses as examples, from which we can see that all buses respond quickly when the 15th bus is added. In the first few iterations, the bus undergoes relatively large corrections to deal with interference. In subsequent iterations, the bus is repeatedly modified to achieve converged accuracy. Judging from the resolution of the interference process, PA can resolve problems such as sudden increase of buses.

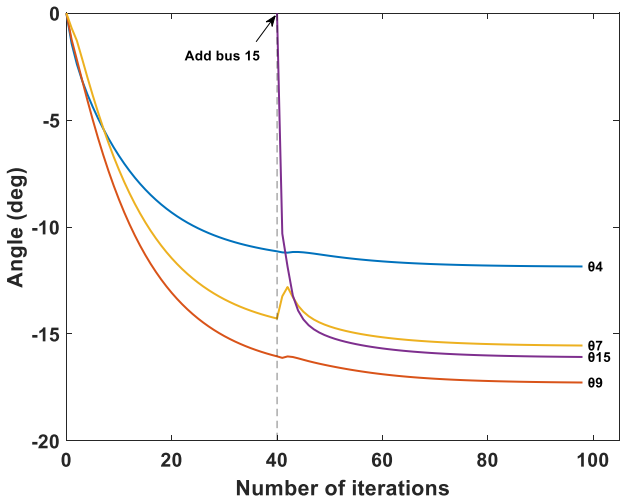


FIGURE 9. Dynamic graph of voltage angle in IEEE-14 system with bus added under PA operation.

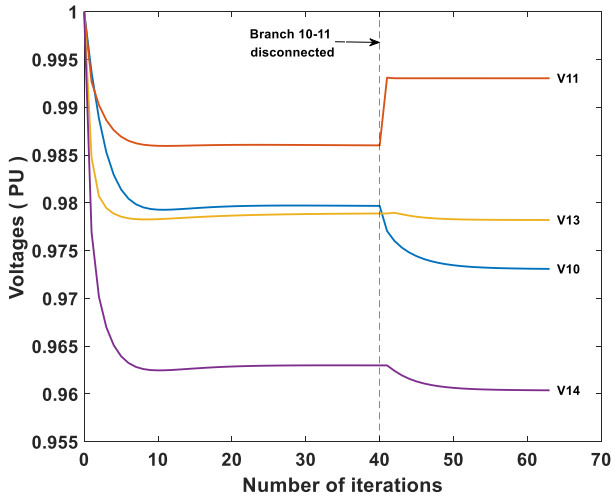


FIGURE 10. Dynamic graph of voltage amplitude in IEEE-14 system with load branch outage under PA operation.

2) OUTAGE A BRANCH WITH LOAD

Under certain extreme conditions, the power system suffers a power outage on a loaded line. We simulate the occurrence of faults by disconnecting the 10-11 branches of the IEEE-14 system during the PF calculation. Taking the 10th, 11th, 13th, and 14th buses as an example, we can observe their voltage changes in figures 10 and 11, where we can find that the voltage of the 10th and 11th buses changes a lot due to the disconnection of the connected branches. The remaining

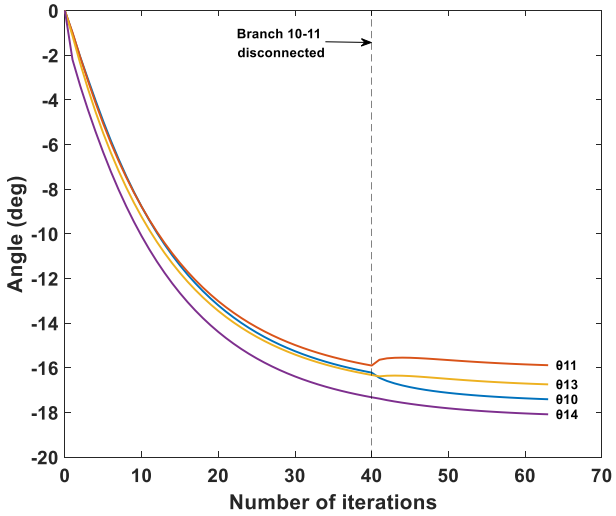


FIGURE 11. Dynamic graph of voltage angle in IEEE-14 system with load branch outage under PA operation.

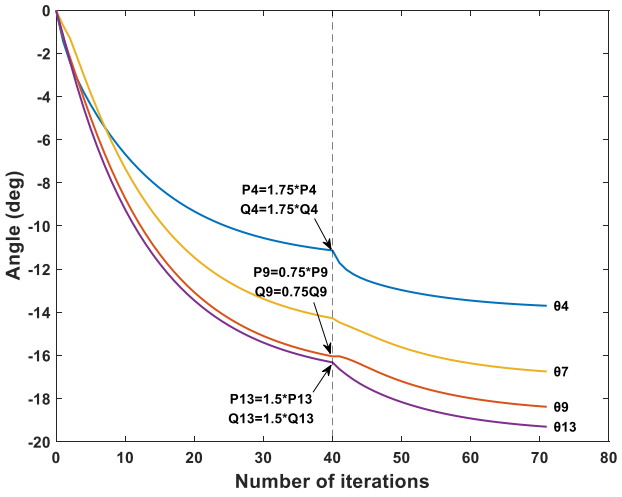


FIGURE 13. Dynamic graph of voltage angle in IEEE-14 system with load fluctuation under PA operation.

bus voltages will also be corrected accordingly in a small range to handle the occurrence of interference. After several iterations to correct the voltage to a new steady-state value, the bus power tends to balance. obviously, the PA algorithm success-fully resolves the interference caused by the line disconnection.

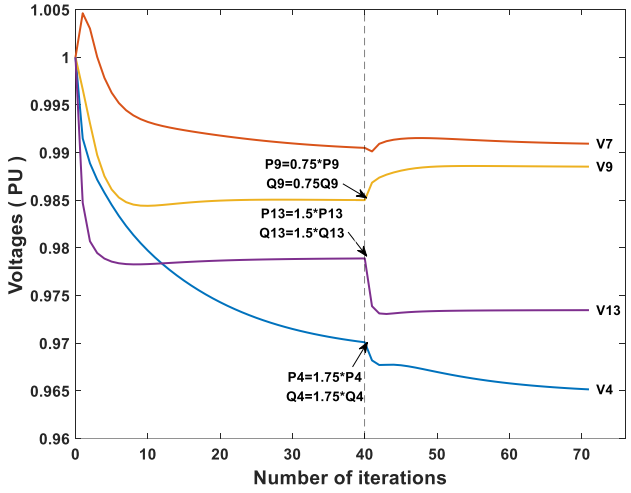


FIGURE 12. Dynamic graph of voltage amplitude in IEEE-14 system with load fluctuation under PA operation.

3) LOAD FLUCTUATION

During the operation of the power system, the load demand of the power users changes frequently. A mature and stable PF algorithm should have the ability to resist such interference. Next, we select three disconnected PQ buses on the IEEE-14 test system to apply fluctuating load for testing, where they are bus 4 with load demand becoming 1.75 times, bus 9 with load demand becoming 0.75 times, and bus 13 with load demand becoming 1.5 times. From figures 12 and 13, we can observe that buses 4, 9, and 13 substantially adjust the voltage to a new steady-state value within a few iterations when load fluctuations occur. The load demand of bus 7 does not change, but since it is a neighbor of buses 4 and 9 with varying loads, it also needs to be retuned in the direction of the voltage iteration to bring the power of the buses to equilibrium. After 17 more iterations than planned, the PA reached a new equilibrium under disturbance, therefore, the PA was able to cope with the disturbance of load fluctuations.

VI. CONCLUSION

This paper proposes a bus-level decentralized power flow calculation algorithm considering the first-order information of neighbors. The proposed algorithm has a simple iterative structure and can protect the privacy of all buses involved in power network power flow calculations. Since the constructed iterative format takes into account the first-order information value of the neighbor bus, the proposed algorithm reduces a large number of iterations compared to the existing methods, and as the system scale increases, the number of iterations decreases more.

The bus-level decentralized power flow algorithm can solve the privacy leakage problem caused by the integration of large-scale renewable energy sources (such as wind energy and solar energy distributed power supplies) and uncon-trollable loads (such as electric vehicles) into the power grid. In addition, it will also make full use of low-performance computing resources scattered on the load side, which makes the overall power flow calculation workload dispersed. However, there is still room for improvement in the algorithm we developed, such as considering higher-order neighbor information to speed up iteration and how to prove the semi-local convergence of iteration. In the future, the proposed algorithm will be continuously improved in application.

TABLE 3. The $\rho(H_*)$ value of test system.

IEEE Test Systems	$\rho(H_*)$
Case 14	0.9630
Case 30	0.9873
Case 39	0.9963
Case 57	0.9894
Case 33	0.9968
Case 69	0.9998
Case 118	0.9968
Case 300	0.9997
Case 1354	0.9999
Case 2383	0.9999

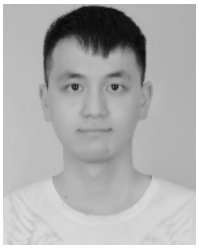
APPENDIX

The $\rho(H_*)$ value of the test system is given in table 3. As demonstrated above, when $\rho(H_*) < 1$, the algorithm has local convergence on the given test system. Evidently, the $\rho(H_*)$ of our given test systems are all less than 1, but they are close to 1. According to [31], we think that $\rho(H_*)$ can reflect the iterative efficiency of the algorithm, the smaller the $\rho(H_*)$, the higher the iterative efficiency of the algorithm. Since the decentralized algorithm preserves privacy among the buses, it sacrifices a part of the iteration efficiency. So $\boldsymbol{\rho}(\boldsymbol{H}_*)$ in Tab. 2 is relatively close to 1.

In fact, in the description in [32], for the iterative type $\boldsymbol{G(x)} = \boldsymbol{x} - \boldsymbol{f(x)}/\boldsymbol{A'(x)}$(iterative form of the algorithm in this paper), $\boldsymbol{\rho}\left(\boldsymbol{G}(\boldsymbol{x}_*)\right)$ is slightly larger than 1, and the algorithm can also convergence. Therefore, $\boldsymbol{\rho}\left(\boldsymbol{G}(\boldsymbol{x}_*)\right) < 1$ is a sufficient but not necessary condition for the algorithm to have local convergence. In this paper, it is proved that $\boldsymbol{\rho}\left(\boldsymbol{G}(\boldsymbol{x}_*)\right) = \rho\left(\boldsymbol{H}_*^2\right) < \boldsymbol{\rho}(\boldsymbol{H}_*)$. The algorithm may converge even $\boldsymbol{\rho}(\boldsymbol{H}_*)$ is larger than 1, so $\boldsymbol{\rho}(\boldsymbol{H}_*)$ close to 1 is not directly related to the local convergence of the algorithm.

REFERENCES

[1] J. J. Grainger, *Power System Analysis*. New York, NY, USA: McGraw-Hill, 1999.
[2] F. Gao, G. Wu, S. Guo, W. Dai, and F. Shuang, "Solving DC power flow problems using quantum and hybrid algorithms," *Appl. Soft Comput.*, vol. 137, Apr. 2023, Art. no. 110147.
[3] Y. Chen, C. Wu, and J. Qi, "Data-driven power flow method based on exact linear regression equations," *J. Modern Power Syst. Clean Energy*, vol. 10, no. 3, pp. 800–804, May 2022.
[4] Y. Liu, Z. Li, and Y. Zhou, "Data-driven-aided linear three-phase power flow model for distribution power systems," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2783–2795, Jul. 2022.
[5] M. Tostado-Véliz, S. Kamel, and F. Jurado, "A powerful power-flow method based on composite Newton–Cotes formula for ill-conditioned power systems," *Int. J. Electr. Power Energy Syst.*, vol. 116, Mar. 2020, Art. no. 105558.
[6] M. Aristizabal, J. L. Hernández-Estrada, M. Garcia, and H. Millwater, "Solution and sensitivity analysis of nonlinear equations using a hypercomplex-variable Newton–Raphson method," *Appl. Math. Comput.*, vol. 451, Aug. 2023, Art. no. 127981.
[7] X. Meng, Y. Li, D. Shi, S. Hu, and F. Zhao, "A method of power flow database generation base on weighted sample elimination algorithm," *Frontiers Energy Res.*, vol. 10, May 2022, Art. no. 919842.
[8] H. Pourbabak, J. Luo, T. Chen, and W. Su, "A novel consensus-based distributed algorithm for economic dispatch based on local estimation of power mismatch," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 5930–5942, Nov. 2018.
[9] V. Dvorkin, F. Fioretto, P. Van Hentenryck, P. Pinson, and J. Kazempour, "Differentially private optimal power flow for distribution grids," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2186–2196, May 2021.
[10] M. Ryu and K. Kim, "A privacy-preserving distributed control of optimal power flow," *IEEE Trans. Power Syst.*, vol. 37, no. 3, pp. 2042–2051, May 2022.
[11] B. Stott and O. Alsac, "Fast decoupled load flow," *IEEE Trans. Power App. Syst.*, vol. PAS-93, no. 3, pp. 859–869, May 1974.
[12] R. A. M. van Amerongen, "A general-purpose version of the fast decoupled load flow," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 760–770, May 1989.
[13] R. Pires, G. Chagas, and L. Mili, "Enhanced power flow solution in complex plane," *Int. J. Electr. Power Energy Syst.*, vol. 135, Feb. 2022, Art. no. 107501.
[14] E. F. Yetkin and O. Ceylan, "Recycling Newton–Krylov algorithm for efficient solution of large scale power systems," *Int. J. Electr. Power Energy Syst.*, vol. 144, Jan. 2023, Art. no. 108559.
[15] C. Zhai, "Jacobian-free Newton–Krylov method for risk identification," in *Control and Optimization Methods for Complex System Resilience*. Singapore: Springer, 2023, pp. 37–58.
[16] 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.
[17] T. Alharbi, M. Tostado-Véliz, O. Alrumayh, and F. Jurado, "On various high-order Newton-like power flow methods for well and ill-conditioned cases," *Mathematics*, vol. 9, no. 17, p. 2019, Aug. 2021.
[18] R. S. Kumar and E. Chandrasekharan, "A parallel distributed computing framework for Newton–Raphson load flow analysis of large interconnected power systems," *Int. J. Electr. Power Energy Syst.*, vol. 73, pp. 1–6, Dec. 2015.
[19] H. Zhang, B. Zhang, A. Bose, and H. Sun, "A distributed multi-control-center dynamic power flow algorithm based on asynchronous iteration scheme," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1716–1724, Mar. 2018.
[20] X. Zhu, X. Han, M. Yang, Y. Xu, D. Sun, and W. Li, "A scalable distributed online algorithm for optimal power flow in distribution system," *Int. J. Electr. Power Energy Syst.*, vol. 129, Jul. 2021, Art. no. 106559.
[21] X. Zhu, X. Han, M. Yang, Y. Xu, and S. Wang, "Distributed online optimal power flow for distribution system," *Int. J. Electr. Power Energy Syst.*, vol. 120, Sep. 2020, Art. no. 105970.
[22] T. Wu, C. Zhao, and Y. A. Zhang, "Privacy-preserving distributed optimal power flow with partially homomorphic encryption," *IEEE Trans. Smart Grid*, vol. 12, no. 5, pp. 4506–4521, Sep. 2021.
[23] S. D. Manshadi, G. Liu, M. E. Khodayar, J. Wang, and R. Dai, "A distributed convex relaxation approach to solve the power flow problem," *IEEE Syst. J.*, vol. 14, no. 1, pp. 803–812, Mar. 2020.
[24] 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.
[25] F. Hameed, M. Al Hosani, M. F. Shaaban, A. Saber, H. H. Zeineldin, and K. Meng, "A meshed backward/forward sweep load flow method for islanded meshed microgrids," *Int. Trans. Electr. Energy Syst.*, vol. 31, no. 11, p. e13127, Nov. 2021.
[26] H. Pourbabak, A. Ajao, T. Chen, and W. Su, "Fully distributed AC power flow (ACPF) algorithm for distribution systems," *IET Smart Grid*, vol. 2, no. 2, pp. 155–162, Jun. 2019.
[27] K. P. Guddanti, Y. Weng, and B. Zhang, "A matrix-inversion-free fixed-point method for distributed power flow analysis," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 653–665, Jan. 2022, doi: 10.1109/TPWRS.2021.3098479.
[28] S. Tan, M. Wu, J. Peng, and H. Jiang, "A node-level distributed power flow based on Newton–Raphson method," in *Proc. 5th Int. Conf. Renew. Energy Power Eng. (REPE)*, Beijing, China, 2022, pp. 422–425.
[29] A. Cuyt and L. Wuytack, *Nonlinear Methods in Numerical Analysis*. Amsterdam, The Netherlands: Elsevier, 1987.
[30] *Matpower's Download Site*. Accessed: Dec. 19, 2023. [Online]. Available: https://matpower.org/download/
[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2003.
[32] O. Sevaioğlu, "Convergence of the fast decoupled load flow for systems with heavy reactive power compensation," *Electric Power Syst. Res.*, vol. 25, no. 2, pp. 91–99, Nov. 1992.



SI-KAI TAN received the B.S. degree in building electricity and intelligence from Anhui Jianzhu University, Anhui, China, in 2020. He is currently pursuing the M.E. degree in control engineering with Shenzhen University, Guangdong, China.

His research interest includes the development of distributed algorithms for complicated nonlinear systems using advanced nonlinear numerical methods.

JIAN-CHUN PENG (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Chongqing University, Chongqing, China, in 1986 and 1989, respectively, and the Ph.D. degree in control science and engineering from Hunan University, Hunan, China, in 1998.

From 1989 to 2008, he was a Teacher with the College of Electrical and Information Engineering, Hunan University. He has been an Associate Professor, since 1996. He has been a Professor in electrical engineering, since 2001. From November 2002 to November 2003, he was a Visiting Researcher with Arizona State University, Tempe, AZ, USA. From May 2006 to August 2006, he was a Visiting Researcher with Brunel University, London, U.K. Since November 2008, he has been a Professor with the College of Mechatronics and Control Engineering, Shenzhen University, Guangdong, China. His research interests include power system optimal operation and control, distributed optimization methods, integrated energy systems, and electricity markets.

MING-HUAN WU received the B.S. degree in electrical engineering from China Three Gorges University, Hubei, China, in 2020. She is currently pursuing the M.S. degree in control science and engineering with Shenzhen University, Guangdong, China.

Her research interests include the development of optimization algorithms for power system operation and complex system control using consensus dynamics.

HUI JIANG received the B.S. degree in electrical engineering from Chongqing University, Chongqing, China, in 1990, and the M.S. and Ph.D. degrees in electrical engineering from Hunan University, Hunan, China, in 1999 and 2005, respectively.

From 1990 to 2009, she was a Teacher with the College of Electrical and Information Engineering, Hunan University. She has been an Associate Professor, since 2003. She has been a Professor in electrical engineering, since 2008. From October 2005 to October 2006, she was a Visiting Researcher with Brunel University, London, U.K. Since June 2009, she has been a Professor with the College of Physical and Optoelectronic Engineering, Shenzhen University, Guangdong, China. Her research interests include power system economics, complex system optimization, and power quality analysis and control.