A Jacobian-Free Newton-GMRES(m) Method with Adaptive Preconditioner and Its Application for Power Flow Calculations

Ying Chen and Chen Shen, *Member, IEEE*

***Abstract*—In this paper, an adaptive preconditioner is constructed for Jacobian-free Newton-GMRES(m) [JFNG(m)] methods, which is devised for solving coordination equations in distributed simulations of power systems. The preconditioner is updated during both Newton iterations and GMRES iterations by means of a rank-one update algorithm. The proposed preconditioned JFNG(m) is applied to power flow calculations for test. The results show that the adaptive preconditioner can enhance convergence of Newton-GMRES(m) iteration schemes greatly and has stronger robustness compared with other precondition methods. Moreover, the proposed method has strong parallelism and scalability, which makes it feasible to solve distributed simulation problems of power systems.**

***Index Terms*—Newton-GMRES(m), precondition, power flow.**

I. Introduction

When studying distributed simulations of power systems, we notice that in heterogeneous computing environments, it is hard to require all local simulation programs to provide the same data interface due to the legacy problem. Meanwhile, power companies joining in distributed simulations would not like to share too much information with others. It is ideal for distributed simulations to require only simple and fundamental information such as states of boundary nodes. Thus, we model the coordination side in distributed simulations as a set of implicit nonlinear equations, which regards voltages of boundary nodes as inputs and power mismatches of these nodes as outputs. Then a general nonlinear solver, which can find roots of nonlinear equations without exact system models, is required to tackle such boundary equations. Jacobian-free Newton-GMRES(m) [JFNG(m)] methods are ideal candidates, which are well developed to solve PDEs and ODEs. As summarized in [1], these methods are synergistic combinations of Newton-type methods for super-linear convergent solution of nonlinear equations and GMRES methods [2] for solving the Newton correction equations. By using the finite difference technique, the Jacobian-vector products are approximated without forming and storing Jacobian matrix elements.

Manuscript received July 18, 2005; revised January 6, 2006. This work was supported in part by the National Basic Research Program of China under Grant 2004CB217903 and in part by the Fundamental Research Funds of Tsinghua University under Grant Jc2003021. Paper no. TPWRS-00441-2005.

The authors are with Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: chenying97@mails.tsinghua.edu.cn; shenchen@mail.tsinghua.edu.cn).

Digital Object Identifier 10.1109/TPWRS.2006.876696

To reduce the number of GMRES iterations and speed up the JFNG(m) methods, various precondition techniques have been developed. Some of these works build up effective preconditioners based on ILU decomposition of a fixed Jacobian matrix obtained at the beginning of the Newton iteration or some detailed mathematical models of the nonlinear equations [3]–[6]. Some other works concern constructing preconditioners adaptively inside the Arnoldi process. These works can be split into two main families, depending on whether the scheme adaptively updates the preconditioner [7], [8] or enlarges the generated Krylov space [9], [10]. However, almost all of them require eigenvalue or eigenvector estimations during preconditioning, which increases both complexities and costs of algorithms.

In this paper, we focus on building an adaptive preconditioner that does not need estimations of the Jacobian or its eigenvalues. The idea is to utilize the projections on the Krylov subspace produced by the Arnoldi process directly to perform corrections of preconditioners. The intrinsic relations between these projections guarantee that the preconditioner can approximate the inverse of the Jacobian matrix effectively. The light is coming from [11], in which a low-order Broyden correction method is used to update the preconditioner for each of the subsequent Newton iterations. Here, we impose similar updates into GMRES iterations that make use of Krylov subspace projections to perform Broyden corrections. A simple but totally "Jacobian-free" right preconditioned Newton-GMRES(m) method has been forged based on this adaptive update scheme. This totally "Jacobian-free" feature is quite meaningful when the Jacobian matrix is too complex to be obtained explicitly or is too expensive to be generated completely, especially for some distributed or parallel computing applications.

Since it is developed to solve boundary equations in distributed simulations, which are actually power balance equations of boundary nodes, the proposed method is applied to power flow calculations to test its convergence. Different from [3]–[5], we focus on Jacobian-free features and make comparisons only between iterative methods with such features. As we are limited by space, the distributed power flow model will not be detailed in this paper.

The content of this paper will be organized as follows. In Section II, some preliminaries such as the standard right preconditioned Newton-GMRES(m) method, rank-one update methods, and the work of [11] are introduced briefly. Section III details the inner preconditioner updates contributed by this paper. Section IV describes applications of the proposed preconditioned JFNG(m) method to power flow calculations,

including the comparison results with other preconditioned JFNG(m) methods. Section V contains conclusions.

II. Preliminaries

The general form of the JFNG(m) method for nonlinear equations is given for the reference as shown below [12].

Algorithm 2.1: The JFNG(m) method with modified Gram–Schmidt orthogonalization and Givens rotations for the nonlinear equations $f(x)=0, x\in R^n, f\in R^n$

1) Set $k=-1$ and choose an initial approximation x_0.
2) $k=k+1$, repeat until $\|f(x_k)\|_2 < errtol_N$.
3) Enter the GMRES(m) iteration to solve the correction function: $f'(x_k)\Delta x_k = -f(x_k)$
   - $r_0=-f(x_k), l=-1, \rho=\|r_0\|_2, v_0=r_0\, errtol_G=\epsilon\|r_0\|_2>0$.
   - While $\rho > errtol_G$ and $l<m$, do

$$l=l+1$$
$$z_l=Mv_l$$
$$v_{l+1}=f'(x_k)z_l \cong \frac{f(x_k+\omega z_l)-f(x_k)}{\omega}.$$

     Orthogonalize $V_{l+1}=[v_1,v_2,\ldots,v_{l+1}]$ and get Hessenberg matrix $H=(h_{i,j})$. Apply and create Givens rotations, $\rho=\|(g)_{l+1}\|$.
   - Set

$$\begin{cases} r_{i,j}=h_{i,j}, & \text{for } 1\le i,j\le k \\ (\omega)_i=(g)_i, & \text{for } 1\le i\le k. \end{cases}$$

     Solve $Ry_l=\omega$ to obtain $\Delta x_k = MV_l y_l$.
4) Update. Compute the new approximation of $x: x_{k+1}=x_k+\Delta x_k$.

JFNG(m) methods can be viewed as combinations of inexact Newton strategies (using k as iteration index) and GMRES(m) methods (using l as iteration index). For GMRES(m) methods, preconditioners can be used to enhance their convergence. In above algorithm, steps such as $z_l=Mv_l$ and $\Delta x_k=MV_l y_l$ are actually performing right preconditioning, which means that instead of solving the equation $f'(x_k)\Delta x_k=-f(x_k)$ directly, equations such as

$$f'(x_k)My=-f(x_k), My=\Delta x_k$$

are solved. The matrix M is called a preconditioner, which is designed to cluster eigenvalues of $f'(x_k)$. It is easy to see that the more accurately the M approximates $f'(x_k)^{-1}$, the faster GMRES converges. Further, if the preconditioner matrix series used during each Newton iteration such as $M_i, i=1,\ldots,m$ satisfies that

$$\lim_{i\to\infty} M_i = \alpha f'(x_k)^{-1}, \text{ where } \alpha\in R \text{ and } \alpha\neq 0 \qquad (1)$$

the convergence of GMRES iterations would be better enhanced more than the normal fixed preconditioner strategy. Different approaches have been developed for constructing the series M_i, such as [7] and [8]. In this paper, a rank-one update method is adopted to achieve this goal without concerning system models and eigenvalue estimations of Jacobian matrices.

The original rank-one update methods [13] are designed to approximate the destination matrix or its inverse iteratively. There are many classes of rank-one update methods. The Broyden methods are focused on in our paper, which are suitable for solving nonsymmetrical linear systems. In its most general form, the Broyden method is given by the following.

Algorithm 2.2: Broyden's method [14]

1) x_0, H_0 arbitrary, $r_0=b-Ax_0$
2) For $k=0,1,\cdots$, until convergence
   - $p_k=H_k r_k$
   - $q_k=Ap_k$
   - $x_{k+1}=x_k+\alpha_k p_k$
   - $r_{k+1}=r_k-\alpha_k q_k$
   - $H_{k+1}=H_k+((p_k-H_k q_k)f_k^T)/(f_k^T q_k)$

where H_k is an approximation of A^{-1} and needs to be chosen in such a way that $f_k^T q_k\neq 0$. The best-known member of the Broyden methods is so-called "Broyden's good" method with $f_k=H_k^T p_k$. As discussed in [11], the Broyden's good method can be used to update the preconditioner for each of the subsequent Newton iterations.

Algorithm 2.3: Preconditioner update based on Broyden's good method [11]

1) Suppose a preconditioner is obtained as M_*.
2) Compute $f'(x_k)\Delta x_k=-f(x_k)$.
   Perform GMRES iterations to solve linear equations with fixed right preconditioner M_* and obtain solutions as x_{k+1}.
3) Set $s_0=x_{k+1}-x_0, \omega_0=f(x_{k+1})-f(x_k)$.
4) $M_*=M_*+((s_0-M_*\omega_0)s_0^T M_*)/(s_0^T M_*\omega_0)$, $k=k+1$; then, go to step 2).

This update strategy can make the preconditioner matrix tracing the inverse of the real Jacobian matrix during the Newton iterations. It is denoted as the **outer preconditioner update** in this paper. However, it still starts from an already known preconditioner matrix M_* constructed at the beginning of the Newton iteration, which is usually the inverse of the Jacobian. Enlightened by its update strategy, we propose a similar **inner preconditioner update** into the GMRES iterations, which makes use of projections on Krylov subspaces generated by the Arnoldi process to update the preconditioner continuously.

III. Adaptive Preconditioners Based on Outer and Inner Update

It can be observed that in the step of Algorithm 2.1 as

$$v_{l+1}=f'(x_k)z_l \cong \frac{f(x_k+\omega z_l)-f(x_k)}{\omega} \qquad (2)$$

where the finite difference is performed to approximate the Jacobian vector product, $f(x)$ is evaluated once, and a group of

$$\begin{cases} \Delta f = f(x_k+\omega z_l)-f(x_k) \\ \Delta x = x_k+\omega z_l - x_k = \omega z_l \end{cases} \qquad (3)$$

is obtained. Obviously this group of Δf and Δx renders a chance to perform one Broyden correction to approximate the

inverse matrix of Jacobian. So, by utilizing the secant direction produced during the finite difference, we can update the preconditioner continuously in GMRES iterations. This update procedure is denoted as the **inner preconditioner update** here.

Together with the outer preconditioner update, the JFNG(m) with full outer and inner preconditioner updates is outlined as follows.

Algorithm 3.1: JFNG(m) method with inner and outer preconditioner update for the nonlinear equations $f(x) = 0, x \in R^n, f \in R^n$

1) Set $k = -1$ and choose an initial approximation x_0, select a nonsingular matrix M_0 as the preconditioner, such as $M_0 = \eta I$, where $I = \mathrm{diag}(1, \ldots, 1) \in R^{n\times n}$, and η is a real constant value.
2) $k = k + 1$; repeat until $\|f(x_k)\|_2 < errtol_N$.
3) Enter the GMRES(m) iteration to solve the correction function: $f'(x_k)\Delta x_k = -f(x_k)$.
   - $r_0 = -f(x_k), l = -1, \rho = \|r_0\|_2, v_0 = r_0 errtol_G = \epsilon\|r_0\|_2 > 0$ $M_{\mathrm{in}}^0 = M_{\mathrm{k}}$.
   - While $\rho < errtol_G$ and $l < m$, do
     — $l = l + 1$
     — $z_l = M_k v_l$
     — $\Delta f_{\mathrm{in}}^l = f(x_k + \omega z_l) - f(x_k), \Delta x_{\mathrm{in}}^l = \omega z_l, v_{l+1} = \Delta f_{\mathrm{in}}^l/\omega$
     — $\surd$Update the preconditioner matrix M_{in} as $M_{\mathrm{in}}^{l+1} = M_{\mathrm{in}}^l + ((\Delta x_{\mathrm{in}}^l - M_{\mathrm{in}}^l \Delta f_{\mathrm{in}}^l)\Delta x_{\mathrm{in}}^{l\,T} M_{\mathrm{in}}^l)/(\Delta x_{\mathrm{in}}^l M_{\mathrm{in}}^l \Delta f_{\mathrm{in}}^l)$.
     — Orthogonalize $V_{l+1} = [v_1, v_2, \ldots, v_{l+1}]$ and get Hessenberg' matrix $H = (h_{i,j})$.
     — Apply and create Givens rotations, $\rho = \|(g)_{l+1}\|$.
   - Set

$$\begin{cases} r_{i,j} = h_{i,j}, & \text{for } 1 \le i, j \le k \\ (\omega)_i = (g)_i, & \text{for} 1 \le i \le k. \end{cases}$$

   Solve $Ry_l = \omega$ to obtain $\Delta x_k = M_k[v_1, v_2, \ldots, v_l]y_l$. Set $M_k = M_{\mathrm{in}}^{l+1}$.
4) Compute the new approximation of $x : x_{k+1} = x_k + \Delta x_k$, compute the residual $f(x_{k+1})$, and $\Delta f_k = f(x_{k+1}) - f(x_k)$.
5) $\surd$Update the preconditioner matrix M_k as

$$M_{k+1} = M_k + \frac{(\Delta x_k - M_k \Delta f_k)(M_k{}^T \Delta x_k)^T}{(M_k{}^T \Delta x_k)^T \Delta f_k}.$$

Steps with $\surd$ are inner and outer preconditioner update procedures, respectively.

A. Difference Between the Outer and Inner Preconditioner Updates

The outer update utilizes information produced by Newton iterations. When the Newton strategy is used to search the solution of $f(x)$, the Jacobian matrix actually indicates the tangent direction at certain point on the curve of $f(x)$. The Jacobian matrix $f'(x)$ varies as the searching point approaches the real solution. The preconditioner for GMRES iterations is the approximation of $f'(x)^{-1}$. The outer update, which uses the secant direction to approximate the tangent direction, enables the preconditioner to follow the changes of the real $f'(x)^{-1}$.

The inner update constructs the preconditioner based on projections on the Krylov subspace generated by GMRES iterations. It can be proven that in GMRES iterations, each update will pull one eigenvalue of $f'(x_k)M$ to the neighborhood of the point $(1, 0)$ on the complex plane.

During inner updates of Algorithm 3.1, as $\Delta x_{\mathrm{in}}^l = \omega M_k v_l$, $\Delta f_{\mathrm{in}}^l = \omega v_{l+1} = f'(x_k)\Delta x_{\mathrm{in}}^l$, from the update equation

$$M_{\mathrm{in}}^{l+1} = M_{\mathrm{in}}^l + \frac{(\Delta x_{\mathrm{in}}^l - M_{\mathrm{in}}^l \Delta f_{\mathrm{in}}^l)(M_{\mathrm{in}}^{l\,T} \Delta x_{\mathrm{in}}^l)^T}{(M_{\mathrm{in}}^{l\,T} \Delta x_{\mathrm{in}}^l)^T \Delta f_{\mathrm{in}}^l} \tag{4}$$

we can have

$$f'(x_k) M_{\mathrm{in}}^{l+1} \Delta f_{\mathrm{in}}^l = f'(x)\Delta x_{\mathrm{in}}^l = \Delta f_{\mathrm{in}}^l \tag{5}$$

$$f'(x_k) M_{\mathrm{in}}^{l+1} v_{l+1} = v_{l+1}. \tag{6}$$

Obviously, $f'(x_k)M_{\mathrm{in}}^{l+1}$ has an eigenvalue as 1, and v_{l+1} is the eigenvector corresponding to it. Actually, not all the eigenvalues of $f'(x_k)M_{\mathrm{in}}^{l+1}$ can be pulled to (1,0) exactly but to a small neighborhood of it, due to influences of the outer updates. These can be seen in the later test results.

In a word, the outer preconditioner update will trace the Newton search direction, while the inner update procedure will utilize projections on the Krylov subspace to rearrange the spectral structure of $f'(x)M$. It should be noted that as M is started from an almost arbitrary matrix ηI, Jacobian-free preconditioners really could be constructed.

B. Modifications for Efficiency Improvement

The rank-one update (4) involves matrix vector productions and vector multiplies. As the rank of M increases, it will take great time and memories to perform such updates. These costs sometimes are even larger than those for GMRES iterations. Two modifications are proposed to improve the efficiency of both inner and outer updates.

1) Generate M_k implicitly: Note that only the matrix vector products such as $z_l = M_k v_l$ and $\Delta x_k = M_k V_l y_l$ are required in the Algorithm 3.1. So there is no need to generate the M explicitly.

As M_0 is known, we collect all Δx and Δf used in steps with $\surd$ of Algorithm 3.1 together. Suppose that when $M_k v_l$ is wanted, there are already n_{used} number of Δx and Δf obtained, which can be expressed as $\Delta x_i, \Delta f_i, i = 1, \cdots, n_{\mathrm{used}}$. Then referring to [13], $z_l = M_k v_l$ can be calculated through the following scheme.

Algorithm 3.2: Calculation of $M_k v_l$

1) $z_l = M_0 v_l$
2) for $i = 1, \cdots, n_{\mathrm{used}}$, $z_l = z_l + p_i^T z_l s_i$, where p_i and s_i are calculated as
3) $t_i = M_0 \Delta f_i$
4) for $j = 1, \ldots, i - 1$, $t_i = t_i + p_{j-1}^T t_i s_{j-1}$
5) $s_i = \Delta x_i - t_i$
6) $\zeta_i = \Delta x_i^T t_i, p_i = \Delta x_i/\zeta_i$

The overall procedure above could be viewed as two parts as that $3) \rightarrow 6)$ preparing vectors for preconditioner updates and $1) \rightarrow 2)$ calculating the desired matrix vector products recursively using the latest preconditioner implicitly. It can be seen that besides M_0, only two vectors such as p_i and s_i need to be stored for preconditioner updates actually, which saves storage effectively.

2) Truncated Vectors for Update: As Newton and GMRES iterations are carried out, more and more p_i and s_i are produced and stored. This increases number of the vectors used for steps 2) and 4) of Algorithm 3.2, which would consume more time and storage. Therefore, a simple strategy is adopted to overcome this shortcoming, which limits the number of vectors for update and precondition as shown in the following.

- Changes in Algorithm 3.2 step 2) for $i = \max(1, n_{\text{used}} - n_{\text{limitP}}), \cdots, n_{\text{used}}\ z_l = z_l + p_i^T z_l s_i$
- Changes in Algorithm 3.2 step 4) for $j = \max(1, n_{\text{used}} - n_{\text{limitU}}), \cdots, i-1\ t_i = t_i + p_{j-1}^T t_i s_{j-1}$

The n_{limitU} and n_{limitP} are limitations for preconditioner update and precondition, respectively. In our numeric tests, these two are selected heuristically as

$$n_{\text{limitU}} = n_{\text{limitP}} = 3\text{rank}(M)/4 + 1. \tag{7}$$

Remarks

Remark 3.1: Destination of eigenvalue shifting: Let us review the equation (4). If a parameter α that $\alpha \in R$ and $\alpha \neq 0$ is added before the first Δx_{in}^l, the equation (6) will become as

$$(f^{'}(x_k) M_{\text{in}}^{l+1}) v_{l+1} = f^{'}(x_k) M_{\text{in}}^{l} \alpha v_l = \alpha v_{l+1} \tag{8}$$

which means that $f^{'}(x_k) M_{\text{in}}^{l+1}$ has an eigenvalue as α, and v_{l+1} is the corresponding eigenvector. That is, by selecting α, the eigenvalues of $f^{'}(x_k) M_{\text{in}}^{l+1}$ can be shifted to any desired point on the complex plane during inner updates. The normal choice is to shift all the eigenvalues to point (1,0), which is adopted by later application actually. However, as implied in equation (1), the destination of shifting can be other points on the horizontal axis. Then a problem is arisen as which point is the best choice that needs minimal efforts to shift all the eigenvalues of $f^{'}(x_k) M_{\text{in}}^{l+1}$ to it? Some researches have been done on this problem. The initial results show that the effects of preconditioner M is not only decided by the value of the smallest eigenvalues but also affected by distributions of all the eigenvalues of $f^{'}(x) M$.

Remark 3.2: Choice of M_0: Notice that in the above preconditioner update scheme, the initial preconditioner matrix is chosen almost arbitrary. The outer and inner updates can make it approximate $f^{'}(x_k)^{-1}$ effectively. This is its best merit that makes it suitable to JFNG(m) methods. However, these update strategies could also be used to improve other preconditioners easily. The choice of preconditioner M_0 could be some ILU decomposition results of Jacobian matrices or some preconditioners produced before. This is very useful in applications for dynamic simulations of power systems, where nonlinear equations are solved for continuous time steps (see [6]).

Remark 3.3: Parallelization abilities: One of the advantages of JFNG(m) methods is that they are easy to be deployed for parallel computing. The prototype of our preconditioned JFNG(m) method as Algorithm 3.1 has good scalability for parallel applications. The most simple parallel strategy might be performing the inner preconditioner updates simultaneously with other GMRES iteration procedures.

As discussed in [13], the modifications for efficiency improvement would make the preconditioner updates hard to be parallelized. However, the Jacobian-free feature of preconditioners creates possibilities for parallelization. In our researches, a distributed computing scheme has been developed successfully based on the proposed method for distributed power flow of power systems.

IV. Application to Power Flow Calculation

As reported in [3], Newton–Krylov methods could be applied to solve different power system problems, such as power flow calculations and dynamic simulations, especially for large power systems. In [4]–[6], Newton-GMRES methods are proposed for power flow calculations. Some popular precondition technologies are adopted in these works, especially the ILU method, which decomposes an approximation of the Jacobian matrix to produce the preconditioner. Obviously, the initial Jacobian matrix is needed. In the recent work of [15], a preconditioner obtained from a linear combination of matrix-valued Chebyshev polynomials is used for the conjugate gradient method to solve power flow problems. This kind of preconditioner requires eigenvalue estimations of Jacobian matrices and historical matrix-valued Chebyshev polynomials to perform updates. These works only make use of GMRES methods to solve linear equations as $Ax = b$ without concerning the Jacobian-free feature of Newton-GMRES methods.

Combined with the presented adaptively updated preconditioner, a fully Jacobian-free Newton-GMRES(m) method has been developed in the above sections. It is easy to deploy this method for the power flow problems. What we need to do is just to write out the power balance equations of the power network as

$$f(x) = 0, x = [\theta_1, \ldots, \theta_n, v_1, \ldots, v_n]^T \tag{9}$$

and substitute them into Algorithm 3.1.

Since Jacobian matrices of power flow equations are almost pivot dominated, the M_0 in Algorithm 3.1 is selected as $(\text{Diag}(\text{f}^{'}(\text{x}_0)))^{-1}$, which is the inverse of diagonal parts of the initial Jacobian matrix and could be obtained implicitly as

$$\text{Diag}(f^{'}(x_0)) = \text{diag}([J_i, J_2, \ldots, J_n]).$$

For $i = 1, 2, \ldots, n$

$$J_i = \frac{f_i(x_0 + \omega e_i) - f_i(x_0)}{\omega} \tag{10}$$

where f_i is the ith equation of power balance equations; $e_i \in R^{n \times 1}, e_i(j) = 0, j = 1, 2, \ldots, i-1, i+1, \ldots, n, e_i(i) = 1$, and $\omega > 0$ is a small constant. Actually, the computation cost of the

above procedure is equal to one evaluation of overall equations (9).

A. Test Cases

All simulations were carried out on the platform of MATLAB version 5.3. Three systems are tested for the proposed method, which are IEEE 39- and 118-bus systems and a real network (1167 buses) of Hebei Province in China.

For the small system such as the IEEE 39-bus system, tests performed are mainly concerned with effects of preconditioner updates. The M_0 is chosen as I. Preconditioner matrices M_k are produced explicitly. Matrices of $f^{'}(x_k)^{-1}$, to which M_k is made to approximate, are also generated. Spectral structures of $f^{'}(x_k)M_k$ are studied to evaluate effects of preconditioner updates. For medium and large systems such as IEEE 118-bus and Hebei systems, the efficiency of the proposed method is tested. The number of both Newton iterations and GMRES iterations is recorded. Comparison between the unpreconditioned Newton-GMRES(m) method and our preconditioned version is made to illustrate effects and efficiency of preconditioners.

As the number of boundary nodes is limited, the size of boundary equations is equal to that of power flow equations of small or middle scale systems. Thus, IEEE 39- and 118-bus systems are used in comparisons between different Jacobian-free preconditioners.

For Newton-GMRES methods, the number of matrix vector products is one of the important indexes of the convergence rate. Moreover, as finite difference technologies are adopted, one Jacobian matrix vector product will require one evaluation of $f(x)$.

B. Test Results

1) Test Results of IEEE 39-Bus System: For the IEEE 39-bus system, parameters for Algorithm 3.1 are selected as

$$errtol_N = 10^{-5}, errtol_G = 0.1\|r_0\|_2, m = 40.$$

The power flow calculation begins from flat start and converges after four Newton iterations. The relation between $\log_{10}\|f(x_k)\|_2$ obtained at each Newton iteration and the times of evaluation of $f(x)$ can be viewed as shown in Fig. 1.

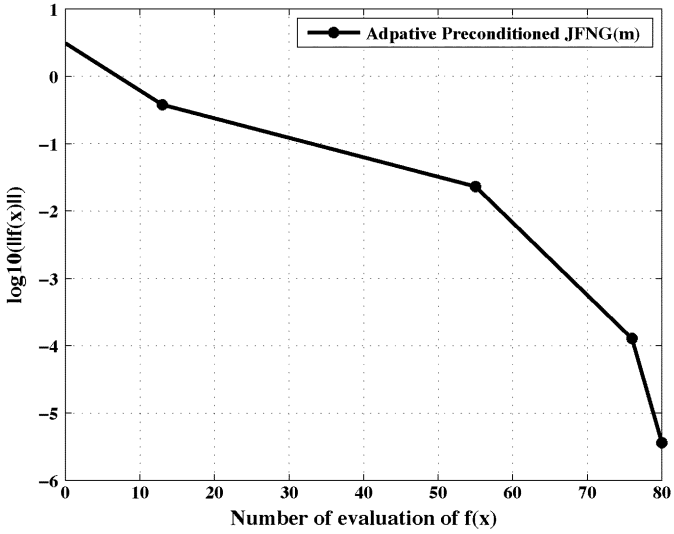


Fig. 1 Convergence procedure of IEEE 39-bus system.

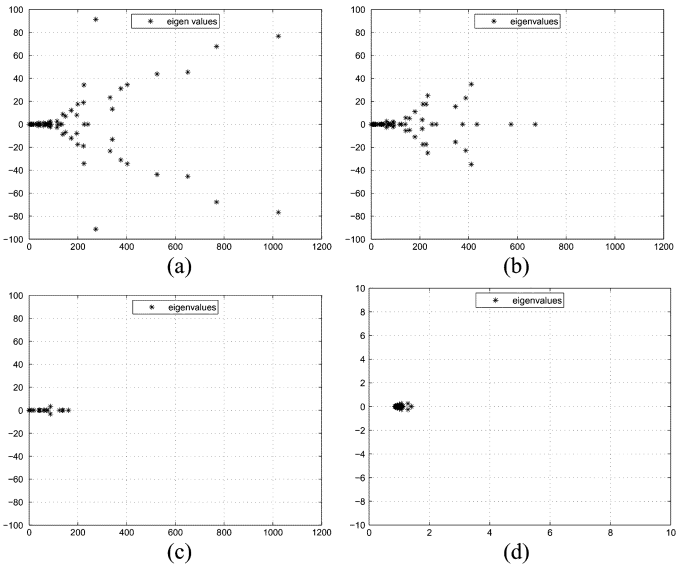


Fig. 2 Spectral structure of $f^{'}(x_k)M_k$ changing with preconditioner updates. (a) Without updates. (b) After 12 updates. (c) After 53 updates. (d) After 76 updates.

In our method, each evaluation of $f(x)$ would activate one preconditioner update. As shown in Fig. 2, after updates are carried out, the eigenvalues of $f^{'}(x_k)M_k$ are pulled to the point of $(1, 0)$ on the complex plane, which indicates that M_k approaches $f^{'}(x_k)^{-1}$ gradually.

2) Test Results of IEEE 118-Bus System: For the IEEE 118-bus system, parameters are selected as $errtol_N = 10^{-5}, errtol_G = 0.1\|r_0\|_2, M_0 = (\mathrm{Diag}(f^{'}(x_0)))^{-1}$. From flat start, normal Newton-GMRES(m) (without precondition) and the proposed preconditioned version are both applied to solve the flow problem. Two tests are performed for $m = 40$ and $m = 100$.

Fig. 3 shows that due to preconditions, the proposed method converges faster than the unpreconditioned Newton-GMRES(m) method. It also can be seen that the parameter m affects convergence of unpreconditioned JFNG(m) greatly, while it makes little influence to the JFNG(m) with the adaptive preconditioner. This is because with inner updates, the minimum eigenvalue of preconditioned Jacobian matrix $f^{'}(x_k)M_k$ could be shifted to $(1, 0)$ after a certain number of updates. After that, the GMRES recurrence could converge quickly in a few iterations. This feature makes our proposed method more robust to different systems.

It can also be observed that during the first several Newton iterations, two methods have almost the same convergence rate. This phenomenon can be explained as that because the updates of preconditioner always follow the projections in the Krylov subspace generated by GMRES methods, the preconditioner will not work on the new projection vectors. In the first several Newton iterations, although the x_k steps forward substantially, the projections made by GMRES are almost on different directions in the Krylov subspace. Therefore, the preconditioner could only accelerate the iterations after a certain number of directions in the Krylov subspace have been searched.

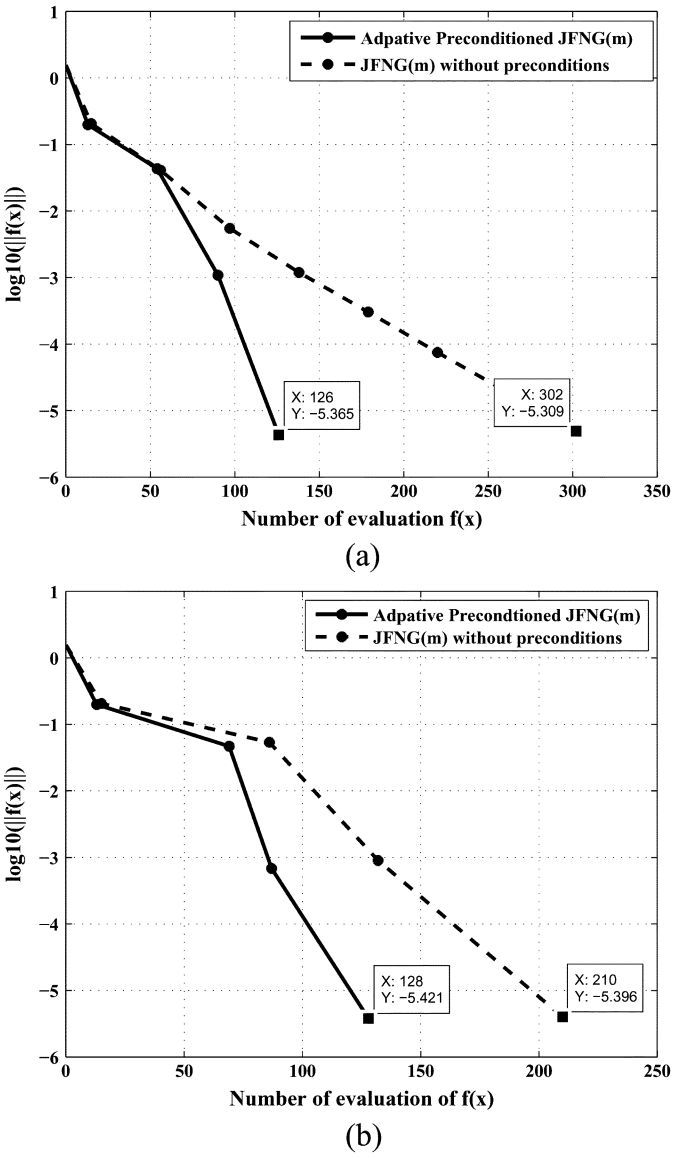


Fig. 3 Convergence procedure of IEEE 118-bus system.

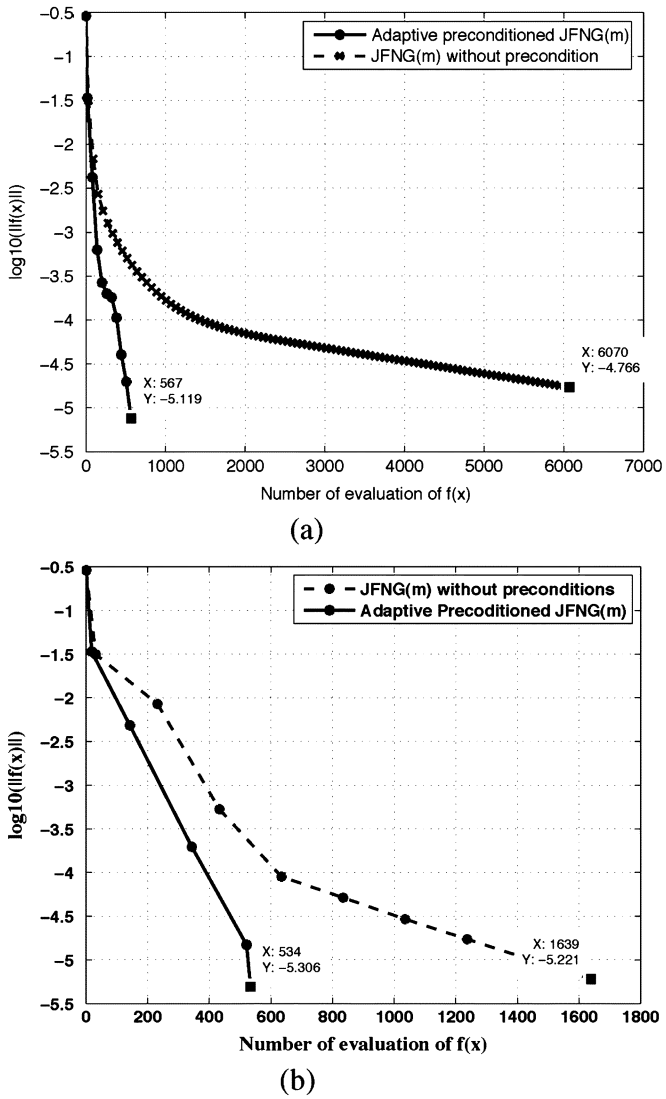


Fig. 4 Convergence procedure of Hebei system. (a) m=60. (b) m=200.

3) Test Results of Hebei System: For the Hebei system, parameters are selected as $errtol_N = 10^{-5}, errtol_G = 0.1\|r_0\|_2, M_0 = (\mathrm{Diag}(f^{'}(x_0)))^{-1}$ and $m = 60, 200$. Starting from flat start, normal Newton-GMRES(m) and the proposed preconditioned version are compared as in Fig. 4.

From the above figures, it is easy to obtain the same results as those got from test results of the IEEE 118-bus system. However, to larger systems, the proposed method shows even more advantages in convergence.

4) Comparisons Between Different Jacobian-Free Preconditioners: As introduced before, several Jacobian-free precondition techniques have been developed for GMRES methods, such as those in [7]–[10]. Here convergence rates of Newton-GMRES methods with these techniques are compared with that of the proposed preconditioned method in this paper, under the situation that the size of the Krylov subspace changes greatly.

Among these works, [7] and [8] are derived from the same idea as our method, which builds preconditioners through adaptive updates. In [7], Choquet constructs preconditioners directly from the Krylov subspace and Hessenberg matrices generated by Arnoldi processes. Reference [8] utilizes a similar formula to build preconditioners in an approximated invariant subspace of the coefficient matrix. Reference [8] is proposed for GMRES to solve linear equations, while [7] has full considerations about nonlinear situations. Therefore, comparisons are only made between Choquet's preconditioned method and our method. As illustrated in Fig. 5, for the IEEE 39-bus system, as the size of the Krylov subspace increased, the Choquet's method has the same convergence rate as the proposed method, while for the IEEE 118-bus system, although the Choquet's method is faster than the unpreconditioned Newton-GMRES method, it is notably slower than the method introduced in this paper. It also can be observed that the presented method is more stable than the Choquet's method, whose convergence rate varies greatly as the size of the Krylov subspace changing.

In [9] and [10], GMRES methods are accelerated by deflated and augmented Krylov subspace techniques. Reference [9] presents the famous GMRES-E(m,k) method, which augments the Krylov subspace with the approximated eigenvector of the coefficient matrix. This method has been adopted by [6] to

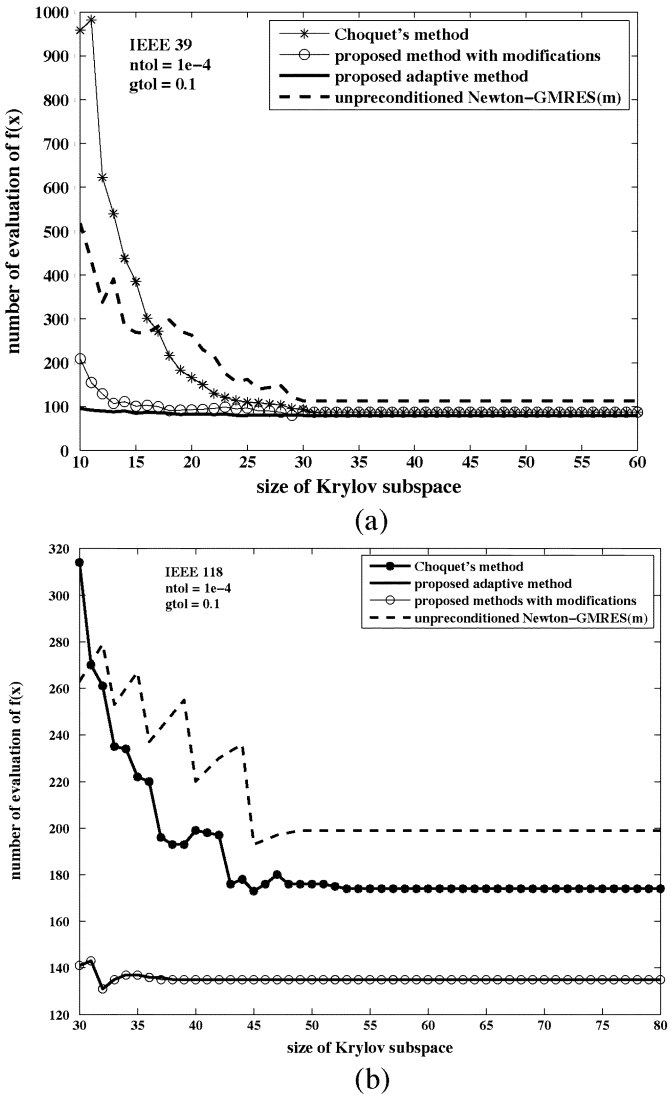


Fig. 5 Comparisons with Choquet's method in [7]. (a) Test results from the IEEE 39-bus system. (b) Test results from the IEEE 118-bus system.

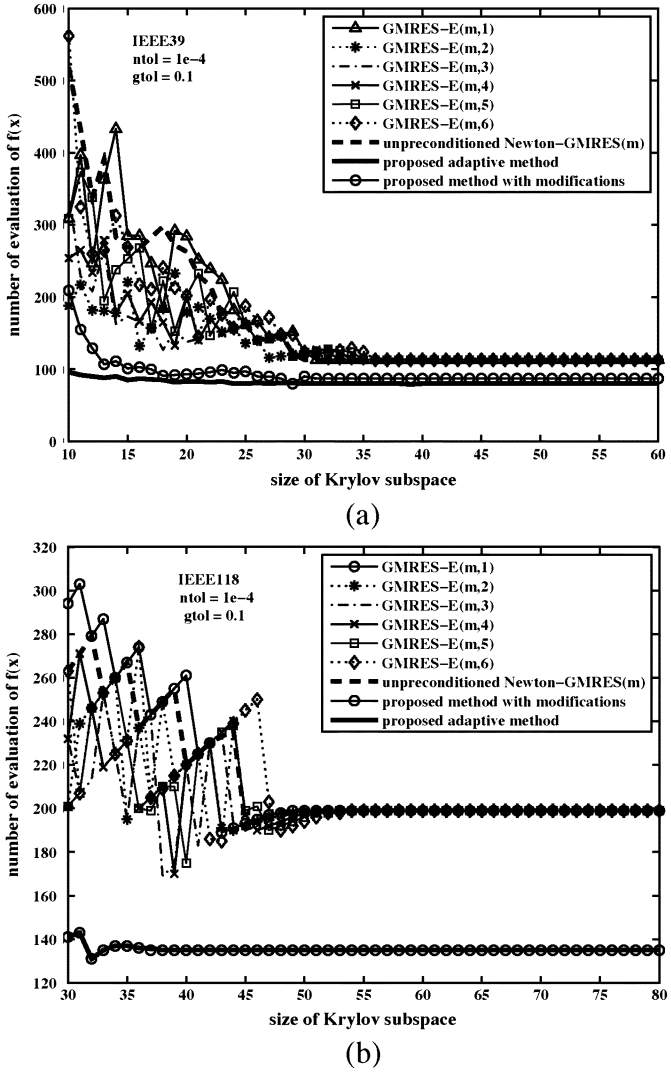


Fig. 6 Comparisons with GMRES-E(m,k) methods. (a) Test results from the IEEE 39-bus system. (b) Test results from the IEEE 118-bus system.

solve power flow problems. It should be mentioned that in [6], GMRES-E(m,k) method is used with ILUT preconditioners, which are not Jacobian-free. Therefore, it should not be compared with the introduced Jacobian-free method. Only if pure GMRES-E(m,k) methods are applied to power flows could fair comparisons be made.

From Fig. 6, it can be seen that the proposed adaptively preconditioned method is faster and more stable than GMRES-E(m,k) methods distinctly.

In the work of [10], DEFLGMRES(m,l) is introduced, which constructs preconditioners from an invariant subspace of the Hessenberg matrix generated through Schur decomposition. During building preconditioners, this method needs to evaluate the matrix vector product of the coefficient matrix and the base vectors of the invariant subspace. Thus, as more base vectors of the invariant subspace are used to forge preconditioners, additional matrix vector products as well as evaluations of $f(x)$ are needed. This can be observed in Fig. 7, especially from those test results from the IEEE 118-bus system.

Obviously, the presented method wins advantages over DEFLGMRES(m,l) methods on the convergence rate and robustness.

V. Conclusion

In this paper, a new adaptive preconditioner is introduced for Jacobian-free Newton-GMRES(m) methods. The preconditioner is constructed simply based on rank-one matrix updates inside and outside GMRES iterations. After theoretical analysis and tests, we believe that the preconditioner has several advantages as shown in the following.

- The proposed update scheme needs no approximation of Jacobian matrices and their eigenvalue evaluations. Based on it, a totally "Jacobian-free" Newton-GMRES(m) method can be formed.
- As mentioned in Section III, the inner updates make the preconditioner approximate the inverse of the Jacobian matrix effectively. Therefore, as proven in tests, the proposed preconditioner can enhance the convergence of JFNG(m) greatly.

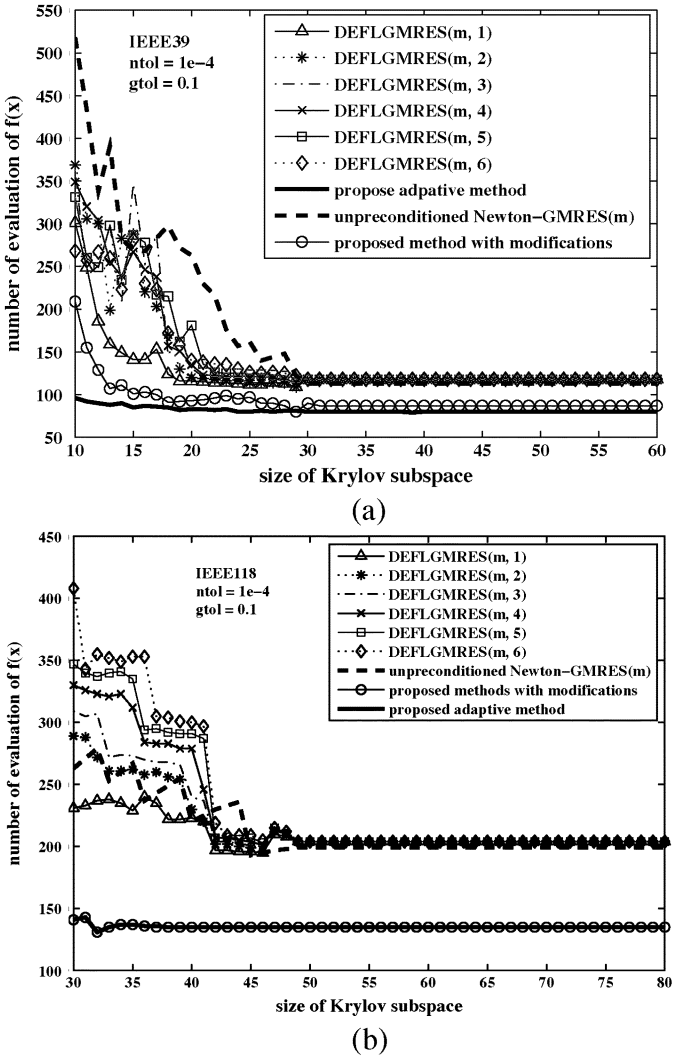
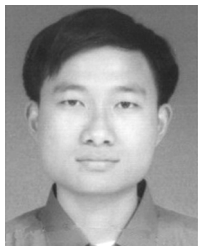


Fig. 7 Comparisons with DEFLGMRES(m,l) methods. (a) Test results from the IEEE 39-bus system. (b) Test results from the IEEE 118-bus system.

- Because it is designed for solving general nonlinear equations, our JFNG(m) method can be easily deployed for many different applications. In this paper, this method has been applied to power flow calculations. The numeric results show that it is comparable to conventional Newton–Raphson methods.
- The update scheme can be used with other precondition technologies. It can be used to enhance the performance of preconditioners obtained by other methods.
- The proposed method has been proven to have high convergence rate and strong robustness by comparisons with other preconditioned Jacobian-free iterative methods referred to in this paper.
- As a fully Jacobian-free method, our method has strong parallelism compared with traditional methods based on direct linear solvers, which means it can be easily deployed into centralized parallel computers or distributed computing environments.

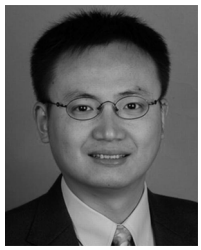
REFERENCES

[1] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton-krylov methods: A survey of approaches and applications," *J. Comput. Phys.*, vol. 193, pp. 357–397, 2004.
[2] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual method for solving nonsymmetric linear systems," *SIAM J. Sci. Comput.*, vol. 6, pp. 856–869, 1986.
[3] M. A. Pai and H. Dag, "Iterative solver techniques in large scale power system computation," in *Proc. 36th Conf. Decision Control*, San Diego, CA, Dec. 1997, pp. 3861–3866.
[4] A. Semlyen, "Fundamental concepts of a Krylov subspace power flow methodology," *IEEE Trans. Power Syst.*, vol. 11, no. 3, pp. 1528–1534, Aug. 1996.
[5] A. J. Flueck and H. D. Chiang, "Solving the nonlinear power flow equations with an inexact newton method using GMRES," in *Proc. IEEE Power Eng. Soc. Winter Meeting*, Tampa, FL, Feb. 1998, pp. 1–5.
[6] M. A. Pai, "A new preconditioning technique for the GMRES algorithm in power flow and P-V curve calculations," *Int. J. Elect. Power Energy Syst.*, vol. 25, pp. 239–245, 2003.
[7] R. Choquet, A matrix-free preconditioner applied to CFD 2605, Jun. 1995, INRIA, Institut National de Recherche en Informatique et en Automatique.
[8] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel, "Adaptively preconditioned GMRES algorithms," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 243–269, 1999.
[9] R. B Morgan, "A restarted GMRES method augmented with eigenvectors," *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 4, pp. 1154–1171, 1995.
[10] J. Erhel, K. Burrage, and B. Pohl, "Restarted GMRES preconditioned by deflation," *J. Comput. Appl. Math.*, vol. 69, pp. 303–318, 1996.
[11] A. Soulaimani, N. B. Salah, and Y. Saad, "Enhanced GMRES acceleration techniques for some CFD problems," *Int. J. Comput. Fluid Dynam.*, vol. 16, no. 1, pp. 1–20, 2002.
[12] P. N. Brown, "A local convergence theory for combined inexact-newton/finite difference projection methods," *SIAM J. Numer. Anal.*, vol. 24, pp. 407–434, 1987.
[13] U. M. Yang and K. A. Gallivan, "A new family of preconditioned iterative solvers for nonsymmetric linear systems," *Appl. Numer. Math.*, vol. 19, pp. 287–317, 1995.
[14] C. Broyden, "A new method of solving nonlinear simultaneous equations," *Comput. J.*, vol. 12, pp. 94–99, 1969.
[15] H. Dag and A. Semlyen, "A new preconditioned conjugate gradient power flow," *IEEE Trans. Power Syst.*, vol. 18, no. 4, pp. 1248–1255, Nov. 2003.



Ying Chen received the B.Eng. degree in 2001 from the Electrical Engineering Department, Tsinghua University, Beijing, China, where he is currently pursuing the Ph.D. degree in electrical engineering.

His research interests are in the areas of power system dynamics and simulation, parallel computing, and grid computing.



Chen Shen (M'98) received the B.S., M.S., and Ph.D. degrees, all in electrical engineering, from Tsinghua University, Beijing, China, in 1993, 1995, and 1998, respectively.

From 1998 to 2001, he was a Postdoctoral Research Fellow with the Electrical and Computer Engineering Department of the University of Missouri–Rolla. In 2001, he joined ISO New England as a Senior Engineer. Currently, he is an Associate Professor with the Department of Electrical Engineering and Applied Electronic Technology, Tsinghua University. His areas of research include power system analysis and control as well as distributed computing in power systems.