Abiezer A. FERNANDES, Francisco D. FREITAS, João Y. ISHIHARA

Department of Electrical Engineering, University of Brasilia, Brasilia – DF, Brazil

doi:10.15199/48.2015.09.63

Performance assessment of iterative linear methods for the computation of the power flow problem solution

Abstract. *This paper investigates the use of iterative linear methods applied to the power flow problem (PFP) solution in power systems. Preconditioning techniques are studied and incorporated to the methodology to solve a linear system. As application to the PFP, four iterative linear techniques and a direct method are evaluated and their performances are compared. Numerical experiments demonstrate the effectiveness of the proposed methodology, suggesting the good performance of the Bi-conjugate Gradient Stabilized (BiCGStab) as an adequate iterative linear method for the applicability in the solution of a PFP.*

Streszczenie. *W artykule analizowano metodę iteracyjną zastosowaną do kontroli przepływów mocy. Zbadano cztery liniowe techniki iteracyjne i metodę bezpośrednią Najlepsze właściwości wykazała technika Bi-conjugated Gradient Stabilized BiCGStab.* ***Oszacowanie możliwości liniowych metod iteracyjnych do kontroli przepływów mocy***

Keywords: power flow, iterative linear systems, BiCGStab, preconditioning.
Słowa kluczowe: przepływy mocy, systemy iteracyjne.

Introduction

Iterative solvers [1] have been usually used as a feasible option when the solution of a large-scale linear system by employing a direct method (based on full Gaussian elimination or full LU factorization) is computationally intensive. The power flow analysis in modern power systems is an example of this type of application. In most cases, the linear sub-problem derived from a nonlinear equation system derived from problems as power flow problem (PFP) is solved by using the traditional direct method [2]. However, according to the network complexity [3], or even the condition of the system of equations [4], the solution can be efficiently determined by iterative linear methods.

There exist several techniques to determine the solution of an iterative linear system. But, depending on the system, the convergence to a stationary result can fail. Hence, iterative solvers usually need to be combined with appropriate preconditioning matrices in order to provide a good convergence rate [1]. Determination of a suitable preconditioner is not a trivial task since it depends on the problem at hand. A preconditioner which is proper for a specific problem often performs poorly in other systems; and, the investigation of general procedures to guide how to obtain a good preconditioner for each class of systems is subject of intense research [3]. In particular, for some tough problems, the direct methods are the unique viable option because or iterative solvers do not converge or, even though convergent, they are not competitive since obtaining an adequate preconditioner can be computationally more expensive than the use of a direct method.

Iterative linear methods have been used to solve the power flower problem (PFP) [3],[5]. In this specific problem, the use of a preconditioner is mandatory, mainly because the Jacobian matrix associated to the problem is non-symmetric. In general, in this problem the preference is for the solver Generalized Minimal Residual (GMRES). In this paper, we demonstrate and justify our preference for the method Bi-conjugate Gradient Stabilized (BiCGStab) [1].

This paper provides a systematic way to search for preconditioning matrices in the solution of hard sparse non-symmetric systems derived from the power flow equations. Experiments on the power flow problem in power systems are used to illustrate the performance of the technique when it is applied to solve the set of nonlinear equations. The experiments are performed in four test-systems, including a Polish power system equivalent.

Background Foundation

A linear system

$$\mathbf{A}x = b \quad \text{with} \quad \mathbf{A} \in \mathbb{R}^{n \times n},\ x \in \mathbb{R}^{n}, \text{ and } b \in \mathbb{R}^{n}, \tag{1}$$

can be solved for the variable x by direct and iterative methods. The use of direct methods (Gaussian elimination) is the straightforward way of solving the system. However, iterative methods are more appropriate to large systems, especially if only an inexact solution is sought [3]. For non-symmetric matrices, usually the iterative methods have poor convergence and preconditioning is necessary [1]. For each iterative solver one can consider left, right or split preconditioning. The left or right preconditioning (matrices $\mathbf{P}_L$ and $\mathbf{P}_R$) mean, respectively, that it is iteratively solved the linear systems

$$\mathbf{P}_L^{-1}\mathbf{A}x = \mathbf{P}_L^{-1}b \quad \text{or} \quad \mathbf{A}\mathbf{P}_R^{-1}y = b\ ,\ x = \mathbf{P}_R^{-1}y\ . \tag{2}$$

If the preconditioners are chosen as the factors **L** and **U** of a matrix **P** whose product is **P=LU,** the problem consists in solving the two-sided preconditioned system [1]

$$\mathbf{L}^{-1}\mathbf{A}\mathbf{U}^{-1}y = \mathbf{L}^{-1}b \quad \text{and} \quad x = \mathbf{U}^{-1}y\ . \tag{3}$$

For fast convergence, the preconditioner should resemble the coefficient matrix **A** and its inverse should be obtained without great complexity [1]. It is then usually chosen a target matrix **P** and constructed either a LU or an incomplete LU (ILU) factorization [1]. In the case of a full LU factorization, this leads to a preconditioner **P=LU≈A**. On the other hand, with the ILU factorization, the target matrix **P=LU** only resembles **A**. In this work, the ILU is preferred against the traditional LU, since it is intended a cheap iterative solver that can compete with the direct solver. A way to construct a target matrix **P** is by assigning to it sub-matrices of **A**, dropping those sub-matrices defined as less dominant. Fortunately, some classes of problem have such peculiar characteristic and so the motivation of this paper will be lead for this kind of application.

A target matrix **P** can be generated by zeroing the less coupled sub-matrices in **A**. This matrix **P** has less nonzero elements than the original matrix **A**. However, according to our experience better target matrix will not be necessarily that with less nonzero elements but whose **L** and **U** factors present less nonzero elements.

The performance (assessed by number of iterations and CPU time) of iterative methods preconditioned with ILU factors can be further improved if a reordering pre-processing on **A** and b is carried out before performing the resolution of the linear system (3). The procedure consists in permutation operations in rows and columns of **A** [6]

$$\mathbf{R}_L\mathbf{A}\mathbf{R}_R y = \mathbf{R}_L b \quad \text{and} \quad x = \mathbf{R}_R y \tag{4}$$

where the transformation matrices $\mathbf{R}_L$ and $\mathbf{R}_R$ are defined from the corresponding row and column permutations of an identity matrix of order n. Depending on the particular application, the best result is obtained by using one of the several reordering algorithms [1]. In this work, the influence of Reverse Cuthill-McKee (RCM) and Approximate Minimum Degree (AMD) methods [6] is evaluated.

Based on the above considerations, the proposed basic procedure to solve iteratively a linear system is

- ✓ evaluate dominance of the sub-matrices of some matrix **A** partitions;
- ✓ select a target matrix **P** among the matrices obtained after truncation of the less dominant sub-matrices of **A**;
- ✓ obtain the reordering matrices $\mathbf{R}_L$ and $\mathbf{R}_R$ of **A** according to (4);
- ✓ obtain the ILU factors, **L** and **U**, for the reordered matrix $\mathbf{R}_L\mathbf{P}\mathbf{R}_R$;
- ✓ apply an iterative solver with the **L** and **U** factors as preconditioner to the reordered system (4) to compute the solution x of the linear system (1).

Iterative Linear Methods

An iterative linear method applied to (1) search a solution starting from an initial estimate $x^{(0)}$. As the technique is iterative, the result is said an inexact solution. We accept an inexact solution $\mathbf{x}\approx\mathbf{x}^{(k)}$, if a given accuracy is reached for a defined numerical quality criterion. A stop criterion based on the relative residue at the iteration k, $\delta=||\mathbf{A}x^{(k)}-b|| / ||b|| < 1\times10^{-6}$ is adopted in this paper.

Methods exploiting characteristics based on Krylov subspace have been used in several applications, especially when the matrix **A** is non-symmetric [1]. A Krylov subspace is generated by a sequence of vectors forming a base defined as

$$K_k(\mathbf{A}, r^{(0)}) := span\{r^{(0)}, \mathbf{A}r^{(0)}, \mathbf{A}^2 r^{(0)}, \cdots, \mathbf{A}^{k-1} r^{(0)}\}, \tag{4.1}$$

where $r^{(0)}$ is the residue calculated as $r^{(0)}=b-\mathbf{A}x^{(0)}$.

The methods based on Krylov subspace to solve (1) consists in finding an approximate solution $x^{(k)}$ which is in a subspace formed by K_k. Assuming this premise, the solution must satisfy an error minimum norm condition, such that the original problem is into the optimization problem

$$\min\{||\ x - x^{(k)}\ ||_2 : x \in x_0 + K_k\}. \tag{4.2}$$

Among several methods based on Krylov subspace, we can cite the GMRES, the Bi-conjugate Gradient (BiCG), the BiCGStab and the Conjugate Gradient Squared (CGS). These methods are adequate for systems with non-symmetric matrix **A**, however each one has its intrinsic characteristics of convergence and robustness.

When an iterative method is employed to solve the linear sub-problem of a classical nonlinear PFP (see e.g. [3],[5]), usually the preference is given to the GMRES method. The Algorithm 1 is a procedure illustrating how the solution x by the GMRES can be computed for (1). In this algorithm, if the approximate solution $x^{(m)}$ does not satisfy the relative residue δ, the algorithm can be restarted by fixing $x^{(0)}$ as the value of x of the last iteration, i.e. $x^{(0)}=x^{(m)}$ and continue until the convergence for the desired stop criterion. In some cases, even with restarting, the process is divergent for a pre-specified number of restarting. For this situation, a remedy is to expand the Krylov subspace or/and augment the number of restarting. But, a reordering and preconditioning process is the best option to obtain a fast convergence. In this paper we propose to use the method BiCGStab as an alternative to the use of the GMRES. As will be illustrated later, the BiCGStab when used for computation of inexact solution may be an adequate technique to solve iterative problems. We present the Algorithm 2 highlighting the main points which are necessary on the iterative process of BiCGStab.

Algorithm 1 – GMRES used to solve $\mathbf{A}x=b$ [1]

INPUT: **A**, b and initial estimate $x^{(0)}$
OUTPUT: Solution $x^{(m)}$

1. Compute $r^{(0)}=b-\mathbf{A}x^{(0)}$, $\beta=||r^{(0)}||_2$ and $v_1=r^{(0)}/\beta$
2. Do j=1,2, …, while j ≤ m
3. Calculate $w_j=\mathbf{A}v_j$
4. Do i=1,2, …, while i ≤ j
5. Compute $h_{ij}=<w_j,v_i>$, where $<w_j,v_i>=w_j^T v_i$ and update w_j as $w_j:=w_j+h_{ij}v_i$
6. End Do i
7. $h_{j+1,j}=||w_j||_2$. If $h_{j+1,j}=0$, do m=j and go to step 10
8. $v_{j+1}=w_j / h_{j+1,j}$
9. End Do j
10. Define a Hessemberg matrix $\mathbf{H}_m$ of dimension $(m+1)\times m$, with entries $H_m(i,j)=h_{ij}$ and a matrix $\mathbf{V}_m=[v_1, v_2,\ldots,v_m]$
11. Calculate y_m which minimize $||\beta e_1-\mathbf{H}_m y_m||_2$ and obtain $x^{(m)}=x^{(0)}+\mathbf{V}_m y_m$, where $e_1=[1, 0, \ldots, 0]^T \in \mathbb{R}^{(m+1)}$

Algorithm 2 – BiCGStab used to solve $\mathbf{A}x=b$ [1]

INPUT: **A**, b and initial estimate $x^{(0)}$
OUTPUT: Solution x

1. Compute $r^{(0)}=b-\mathbf{A}x^{(0)}$, define $p_0:=r^{(0)}$
2. Do j=0,1,2, …, until convergence
3. half iteration $\alpha_j=<r^{(j)},r^{(0)}>/<\mathbf{A}p_j,r^{(0)}>$, $s_j:=r^{(j)}+\alpha_j\mathbf{A}p_j$, $w_j=<\mathbf{A}s_j,s_j>/<\mathbf{A}s_j,\mathbf{A}s_j>$
4. complement of the iteration $x^{(j+1)}=x^{(j)}+\alpha_j p_j+w_j s_j$, $r^{(j+1)}=s_j-w_j\mathbf{A}s_j$, $\beta_j=[<r^{(j+1)},r^{(0)}>/<r^{(j)},r^{(0)}>]\times[\alpha_j / w_j]$
5. $p_{j+1}=r^{(j+1)}+\beta_j\times[p_j-w_j\mathbf{A}p_j]$
6. End Do j
7. $x=x^{(j+1)}$

In this work we show that the *a priori* preference for the GMRES as linear iterative solver, as it is done in the most of the literature, is not justified. Other solvers besides GMRES can also be competitive with the direct method and even perform better. So, besides the GMRES and BiCGStab, we study two other techniques BiCG and CGS. For details about these methods, see for example [1].

Power System Application

To analyze the applicability of the proposed methodology, it is considered the power flow problem in power systems. A detailed modeling of the mathematical and physical problem formulation can be found in [4]. Basically, this is a nonlinear problem which consists on determining the solution of the set of nonlinear equations

$$\Delta P_k(V,\theta) = P_k^{sp} - P_k(V,\theta) = 0,\ \ k = 1,2,\cdots,n1 \tag{5}$$

$$\Delta Q_j(V,\theta) = Q_j^{sp} - Q_j(V,\theta) = 0,\ \ j = 1,2,\cdots,n2 \tag{6}$$

(7) $$P_k(V,\theta) = V_k \sum_{m\in\Omega_k} V_m[G_{km}\cos(\theta_{km}) + B_{km}\sin(\theta_{km})]$$

(8) $$Q_j(V,\theta) = V_j \sum_{m\in\Omega_j} V_m[G_{jm}\sin(\theta_{jm}) - B_{jm}\cos(\theta_{jm})]$$

where $\theta=[\theta_1, \theta_2,\cdots, \theta_{n1}]^T \in \mathbb{R}^{n1}$ and $V=[V_1, V_2,\cdots, V_{n2}]^T \in \mathbb{R}^{n2}$ are the unknow variables in the set of equations composed of (5) and (6). The variables V_i and θ_i are the magnitude and phase angle of the voltage at bus i, respectively. The specified active power at bus of type PQ and PV and specified reactive power at bus PQ are P_k^{sp} and Q_k^{sp}, respectively. Consequently, $n1$ is the number of PQ and PV bus, while $n2$ is the number of only PQ bus. θ_{km} stands for the deviation $\theta_{km}=\theta_k-\theta_m$. G_{km} and B_{km} are, respectively, the real and imaginary entries of the bus nodal admittance matrix, $\mathbf{Y}_{bus}=\mathbf{G}+j\mathbf{B}$. The symbol Ω_i means a subset of bus with physical connection at bus i. The quantities ΔP_k and ΔQ_j are known as mismatches of the equations (5) and (6), respectively. Considering that this nonlinear system is solved iteratively, an acceptable solution is found if the maximum between $\max|\Delta P_k|$, $k=1,2,\cdots,n1$ and $\max|\Delta Q_j|$, $j=1,2,\cdots,n2$ reaches a value smaller than a given tolerance $\varepsilon>0$ sufficiently small. This characteristics give to the problem a nature of inexact solution.

Given an initial estimates $\theta^{(0)}$ and $V^{(0)}$, an iteration q of the Newton-Raphson method applied to (5)-(6), starting with $q=0$, consists on to compute the solution $\Delta\theta^{(q)}$ and $\Delta V^{(q)}$ of a linear sub-problem of the type

(9) $$\begin{bmatrix} \mathbf{H}^{(q)} & \mathbf{N}^{(q)} \\ \mathbf{M}^{(q)} & \mathbf{L}^{(q)} \end{bmatrix} \begin{bmatrix} \Delta\theta^{(q)} \\ \Delta V^{(q)} \end{bmatrix} = \begin{bmatrix} \Delta P^{(q)} \\ \Delta Q^{(q)} \end{bmatrix}$$

where

(10) $$\mathbf{H} = \frac{\partial P}{\partial \theta}, \quad \mathbf{N} = \frac{\partial P}{\partial V}, \quad \mathbf{M} = \frac{\partial Q}{\partial \theta}, \quad \mathbf{L} = \frac{\partial Q}{\partial V}.$$

After computing the deviations $\Delta\theta^{(q)}$ and $\Delta V^{(q)}$, then they are used to update the solution $\theta^{(q+1)}= \theta^{(q)}+\Delta\theta^{(q)}$ and $V^{(q+1)}= V^{(q)}+\Delta V^{(q)}$, until convergence (or not) of the Newton-Raphson method for a maximum number of iterations. In case of convergence, other quantities are computed by using the values of the states (magnitude and phase angle of bus voltages). These other quantities, such as power at the swing bus, reactive power at PV bus, are calculated with computational burden much smaller than that for obtaining the states. Then, the interest to efficiently solve the linear sub-problem at each iteration is a key point on the solution of the PFP.

The linear sub-problem (9) will be rewritten as

(11) $$\begin{bmatrix} \mathbf{A}_1 & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad \Rightarrow \quad \mathbf{A}x = b$$

where $\Delta\theta$ and ΔV are the unknown variables to be computed, i.e the variable $x=[\Delta\theta^T, \Delta V^T]^T$ in (1), while $b=[\Delta P^T, \Delta Q^T]^T$ represents the independent vector. Also, $\mathbf{A}_1=\mathbf{H}$, $\mathbf{B}=\mathbf{N}$, $\mathbf{C}=\mathbf{M}$ and $\mathbf{D}=\mathbf{L}$. It is noteworthy to mention in (11) that all sub-matrices are non-symmetric, although **B** and **C** have symmetrical structure. The physical knowledge of the power flow problem lead us to the study of the relative dominance among the matrices on the (ΔP, $\Delta\theta$) and (ΔQ, ΔV) sub-problems.

Strategy for Computing a Preconditioner

For the iterative linear sub-problems, we propose to use a preconditioner computed just along the first iteration of the nonlinear iterative process of the Newton-Raphson method.

To determine a preconditioner, we exploit the fact that the active power has stronger coupling with phase angle than with magnitude of bus voltage. Also, it is considered the fact that the reactive power is more sensitive to changes in the magnitude than phase angle of bus voltages. These physical characteristics of the power flow gave place to variants of the Newton method applied to this problem. Based on decoupling according to the aforementioned sensitivity characteristics, modifications known as decoupled Newton and fast decoupled Newton methods were proposed [3]. These insights suggest the study of the following structures of target matrix for preconditioner:

(12) $$\mathbf{P}_1 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}, \quad \mathbf{P}_3 = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}$$

The target matrix $\mathbf{P}_1$ has structure whose block-diagonal sub-matrices are fully decoupled with relation to the off-diagonal sub-matrices. In this sense, the preconditioners $\mathbf{P}_2$ and $\mathbf{P}_3$ are partially decoupled. Another fourth option is to employ the own **A** as preconditioner. However, this option will be generate factors **L** and **U** with greater number of nonzero entries than those structures proposed in (12).

The Algorithm 3 summarizes the schema for computing a solution of the power flow problem when an iterative linear solver is employed. In this algorithm, the stop condition is in the step 4, case convergence be reached; or in the step 6, if the maximum number of iterations for the nonlinear loops be surpassed. We are assuming that the algorithm is convergent. Then, at the end of it the states θ and V are calculated. Our focus in this paper is to evaluate the performance of some iterative linear methods to compute these states and compare against the performance of the traditional direct method.

Numerical Experiments and Results

To analyze the applicability of the proposed methodology numerical experiments were carried out on test-systems of four power systems. The smallest system is the well known IEEE 118-bus system. The biggest is a Polish equivalent system representing the Polish 400 kV, 220 kV and 110 kV networks during the winter 2007-2008 evening peak conditions (for details of this test-system, see the file `case3375wp.m` obtained from [7]). The system has 3374 buses, with 2982 load and 392 generation buses. Two other intermediate-sized systems are the IEEE 300-bus and a 319-bus Brazilian equivalent network related to the year 1987.

Algorithm 3 – Iterative linear method for the PFP

INPUT:	Network data, $x^{(0)}$, **H**, **N**, **M**, **L** and ε
OUTPUT:	States θ and V

1. Compute $\Delta P^{(0)}$, $\Delta Q^{(0)}$, $\mathbf{H}^{(0)}$, $\mathbf{N}^{(0)}$, $\mathbf{M}^{(0)}$ and $\mathbf{L}^{(0)}$; perform reordering in **A**, as in (4) to determine $\mathbf{R}_L$ and $\mathbf{R}_R$; reorder **A** and b in (4) to form $\mathbf{A}_r=\mathbf{R}_L\mathbf{A}\mathbf{R}_R$, $b_r=\mathbf{R}_L b$, and the linear system $\mathbf{A}_r x_r=b_r$. Compute $x=\mathbf{R}_R x_r$.
2. Define **P** as in (12); reorder **P** with the same matrices $\mathbf{R}_L$ and $\mathbf{R}_R$ computed in the step 1 and calculate $\mathbf{P}_r= \mathbf{R}_L\mathbf{P}\mathbf{R}_R$. Calculate the ILU factors **L** and **U** of $\mathbf{P}_r$.
3. Solve iteratively, as in (3), the linear system $\mathbf{A}_r x_r=b_r$ with the factors **L** and **U** obtained in the step 2. After, compute the original solution x and do $q:=1$.
4. Compute $\Delta P^{(q)}$, $\Delta Q^{(q)}$, and test for convergence of the Newton-Raphson method. In case of convergence, do $x=[\theta^T, V^T]^T$ and stop; else, continue.
5. Compute $\mathbf{A}_1=\mathbf{H}^{(q)}$, $\mathbf{B}=\mathbf{N}^{(q)}$, $\mathbf{C}=\mathbf{M}^{(q)}$, $\mathbf{D}=\mathbf{L}^{(q)}$; form **A** and b according to (11), reorder them with the permutations

matrices obtained in the step 1, and then use the factors **L** and **U** of the step 2 to solve iteratively the updated linear system $\mathbf{A}_r x_r = b_r$ and calculate x.

6. Do $q:=q+1$ and return to the step 4, while q do not surpass a pre-specified number of iterations.

In the first part, the studies aimed to determine the best preconditioner associated to the PFP. We propose to compute **P** and its ILU factors **L** and **U** only for the first iteration of the Newton-Raphson process. Then, these factors are kept and used in the subsequent iterations as preconditioner for solving the linear systems. Due to brevity of space in this paper, just results concerning the biggest test-system will be presented. In the second part, performance of iterative methods for computing the solution of the PFP (the nonlinear problem) are evaluated and compared with the performance of the direct method.

All simulations were accomplished in an Intel (R) core™ 2Duo processor notebook, 32 bits, clock of 1.66 GHz and 2 GB of RAM. The experiments were carried out by using the software Matlab R2009b. We have used the own Matlab's linear solver scripts to compute the solution related to the direct and iterative methods.

Preconditioner studies

The matrix **A** in (11), as well as the vector b for the Polish test-system was computed for the first iteration of the Newton-Raphson process. The linear system formed this way was utilized to study the influence of a preconditioner when iterative methods are employed. The matrix **A** has dimension 6355×6355 and 40704 nonzero entries.

Besides the performance of some iterative methods, the influence of three types of preconditioner as presented in (12) were investigated.

For all simulations, it is assumed initial value $x^{(0)}=0$ and stop criterion δ less than 10^{-6}. The CPU times in this work are the average of the values obtained in 500 repeated experiments using `tic` and `toc` MATLAB commands. As first experiment, it was performed a direct application (no preconditioner) of the iterative methods GMRES, CGS, BiCG and BiCGstab [1]. None provided a satisfactory convergence behavior. Then, we considered the natural partition of the matrix **A** in (11), and performed a relative dominance analysis by dropping submatrices **B** or (and) **C**.

Applying each target matrix (**P**$_1$, **P**$_2$ or **P**$_3$) directly as left or right preconditioner as in (12), the system converges but, all iterative methods are much slower than the direct method. Next we considered the two-sided preconditioning with ILU factorization, as in (3). The ILU factorization depends on a rule to form **L** and **U** and a parameter known as drop tolerance τ [8]. The drop tolerance is a quantifier which gives a metric to approximate a nonzero entry by a null value. This way, **L** and **U** can be constructed in such way to fully preserve the same sparse structure of **A**. This situation is known as ILU(0) and it is characterized by factors **L** and **U** whose number of nonzero entries is the smallest. The extreme case occurs when **L** and **U** are the same matrices obtained from a full LU factorization of **A**. The factorization with such pattern is known as ILU(n). In practice, these two extreme situations are of little interest, since ILU(0) provides poor quality of convergence, while ILU(n) is similar to the classical Gaussian elimination technique. Then an intermediate strategy must be preferred. But, this intermediate candidate, called ILU(k) is dependent of τ. In view of these aspects, we investigate the influence of the drop tolerance, when a kind of ILU(k), called ILUT(τ) [8] is applied. The study also considered the reordering of x. Numeric experiments were performed for several values of τ. According to [1], a range of values of τ from 10^{-5} to 10^{-4} is suggested. Typical results obtained for $\tau=6.5\times10^{-5}$ and considering experiments with AMD and RCM ordering are shown in Tab. 1. The method performance with no reordering is much worst and so the results were omitted in the table.

Table 1. Number of iterations and timing, in seconds

Reord. Type	*Method*	*Iterations*			*Timing*		
		P$_1$	**P**$_2$	**P**$_3$	**P**$_1$	**P**$_2$	**P**$_3$
RCM	GMRES	3/1	6/5	2/3	0.044	0.118	**0.036**
	BiCGstab	8	13	4.5	0.031	0.057	**0.024**
	BiCG	13	16	9	0.068	0.095	**0.060**
	CGS	8	12	5	0.031	0.056	**0.023**
AMD	GMRES	2/5	2/3	2/1	0.034	0.028	**0.023**
	BiCGstab	8.5	5	3.5	0.031	0.020	**0.015**
	BiCG	12	9	8	0.054	**0.043**	0.044
	CGS	9	7	4	0.032	0.027	**0.017**

In all cases, we used the ILUTP method with diagonal pivoting and row sum preservation [8]. In the Tab. 1, at 'Iterations' columns, the results are the number of iterations required when **L** and **U** factors of ILU of **P**$_1$, **P**$_2$ or **P**$_3$ are used. The notation *k/m* in the GMRES column refers to number of external loop iterations (m) and number of restarts k. The BiCGstab is counted by half iterations.

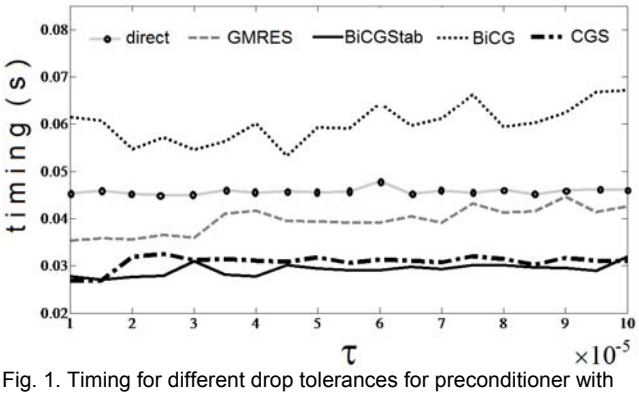

Fig. 1. Timing for different drop tolerances for preconditioner with target matrix **P**$_3$

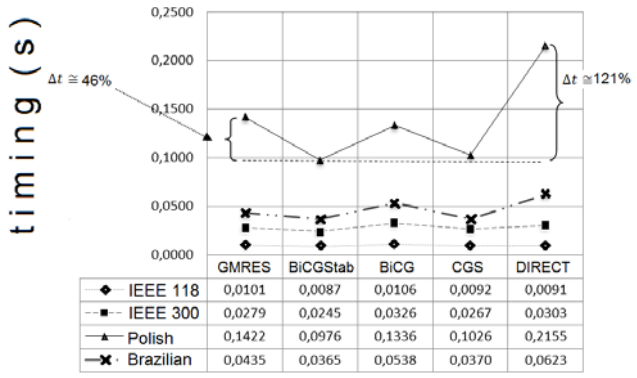

Fig. 2. Timing for solving the PFP

Using only ILU (without reordering), the iterative methods spent less CPU time than in the previous experiments with pure left or right preconditioning. However, all of them remained much slower than the direct method, which demands 0.046 s of CPU time. Applying reordering, the results are more robust against variations on τ. Its influence for each **P**$_i$, i=1,2,3 were investigated. Fig. 1

illustrates the total CPU time against τ when a preconditioner is based on a target matrix $\mathbf{P}_3$. Only AMD reordering is used. One can see that the tuning of the drop tolerance becomes noncritical on the performance of each iterative method. Our experiments showed that, with $\mathbf{P}_1$ or $\mathbf{P}_2$, no iterative method is preferable over the others since none was systematically faster in all τ intervals. The AMD reordering performed better than the RCM for all target matrices. Consistently to this fact, experiments showed lower number of nonzero elements of the **L** and **U** factors of the reordered matrix when AMD is used. The lowest CPU time consumption (0.015 s) is obtained for the BiCGstab method using ILU factors of $\mathbf{P}_3$ as preconditioner with AMD reordering (see Tab. 1). This result is approximately 33% of the CPU time of the direct method with reordering (0.046 s). In the literature, it is very common to use GMRES and $\mathbf{P}_1$[2]. Our methodology enabled us to find a less natural target matrix, $\mathbf{P}_3$, and iterative solver, BiCGstab, with better results than the direct method and GMRES with $\mathbf{P}_1$.

Power Flow Experiments

In this section, we present results about experiments in four systems, the iterative linear methods and the direct method. A base case of the PFP was studied with reordering AMD and using the preconditioner $\mathbf{P}_3$. A $\tau=2.5\times10^{-5}$ and $\varepsilon=10^{-4}$ were used. For each system a mean of four iterations were necessary to convergence. The CPU time for each system and method is shown in Fig. 2. We observe that for the two smaller systems the performances of all methods are similar. But, there is difference of performance among the methods for the two biggest systems. The best performance for the biggest system (Polish system) is attributed to the method BiCGStab. The Method CGS has presented similar performance. The worst performance was observed for the direct method, being required about 0.21 s of CPU time, against only 0.098 s of the method BiCGStab. This result represents a difference Δt of about 121%. The method GMRES, usually used in many applications, demanded 0.14 s, a Δt about 46% higher in relation to the BiCGStab.

Based on these results, the larger the system, the greater the difference of CPU time between the method BiCGStab and the direct method.

Conclusions

This paper presented a method to determine a solution of the power flow equations based on the use of technique for iterative linear systems. The methodology requires the computation of a preconditioner whose results have demonstrated that the best option occurs when the decoupling of the reactive power with the phase angle of the voltage bus is considered. It was studied four iterative methods and compared their performance with the direct method (based on Gaussian elimination). It was verified that the best performance is observed when a reordering process of AMD type is employed jointly with the method BiCGStab. We also demonstrate through simulations that for inexact solution of linear systems, as required at each iteration of the Newton-Raphson process, the method BiCGStab has better performance than the GMRES.

The proposed approach applied to the linear sub-problem associated to the power flow can furnish a significantly faster iterative solver than the usual ones. The method is general, has low computational cost and might be used as a viable candidate for tough large systems where the usual direct and iterative methods are limited.

REFERENCES

[1] Saad, Y., Iterative methods for sparse linear system, SIAM, Philadelphia, (2000)

[2] Zadeh, A. K., Zeynal, H., Nor, K. M., Shaaban, M., Performance Evaluation of SuperLU and PARDISO in Power System Load Flow Calculations, *Przegląd Elektrotechniczny*, 87 (2011), nr 11, 290-294

[3] Idema, R., Lahaye, D. J., Vuik, C., Sluis, L. V. D., Scalable Newton-Krylov solver for very large power flow problems, *IEEE Trans. on Power Systems*, 27 (2012), nr 1, 390- 396

[4] Shahriari, A., Mokhlis, H., Bakar, A. H. A., Karimi, M., Illias, H. A., Application of State Space Search Method to find a Low Voltage Solution for Ill-Conditioned System, *Przegląd Elektrotechniczny*, 88 (2012), nr 12a, 247-250

[5] Flueck, A., Chiang, H., Solving the nonlinear power flow Equations with an inexact Newton method using GMRES, *IEEE Trans. on Power Systems*, 13 (1998), nr 2, 267-273

[6] Amestoy, P., Davis, T., and Duff, I., Johnson B., Pike G.E., An approximate minimum degree ordering algorithm, *SIAM Journal on Matrix Analysis and Applications*, 17 (1996), nr 4, 886-905

[7] Zimmerman, R. D., Sanchez, C. E., and Gan, D.: Kowalski J., MATPOWER, http://www.pserc.cornell.edu/matpower, accessed on Jan. 5th, 2015

[8] Saad, Y., ILUT: A dual threshold incomplete ILU preconditioner, Numerical Linear Algebra Application, 1 (1994), nr 4, 387-402

***Authors**: prof. dr Abiezer Amarilia Fernandes, UniCEUB,* Asa Norte, CEP:70000-900, *Brasilia, DF, Brazil, e-mail: abiezer.fernandes@uniceub.br; prof. dr Francisco Damasceno Freitas, e-mail: ffreitas@ene.unb.br, and prof. dr João Yoshiyuki Ishihara, e-mail: ishihara@ene.unb.br, Department of Electrical Engineering, University of Brasilia, CEP:70910-900, Brasilia, DF, Brazil.*

The correspondence address is:
E-mail: ffreitas@ene.unb.br