TOWARD IMPROVED USES OF THE CONJUGATE GRADIENT METHOD FOR POWER SYSTEM APPLICATIONS

Hasan Dağ **Fernando L. Alvarado**
Department of Electrical and Computer Engineering
The University of Wisconsin – Madison

Abstract

The conjugate gradient method has been suggested as a better alternative to direct methods for the solution of certain large sparse linear systems $A\,\mathbf{x} = \mathbf{b}$, where A is symmetric and positive definite. Efficiency considerations often require that the conjugate gradient method be accelerated by preconditioning (a linear transformation of A). One of the most widely used preconditioners is based on the incomplete LU factors of A. Positive definite preconditioner matrices assure convergence. However, the incomplete factorization for a symmetric and positive definite matrix is not necessarily positive definite. This paper provides significant theoretical insights into the conjugate gradient method for matrices arising from several classes of power systems problems. The paper also presents a new preconditioner (based on a one-time complete factorization) that is guaranteed to be positive definite.

Keywords Linear equations, iterative methods, conjugate gradient method, preconditioner.

1 Introduction

The solution of the linear equations (1) is required as part of the solution of the set of nonlinear power flow, the state estimation problem, the security analysis problem, and also during transient stability and electro-magnetic transient analysis. The conjugate gradient (CG) method of Hestenes and Stiefel [9] has been suggested for the solution of sets of large sparse linear equations of the form:

$$A\,\mathbf{x} = \mathbf{b}, \tag{1}$$

where matrix A is symmetric and positive definite (A is positive definite if $\mathbf{x}^T A\,\mathbf{x} > 0\ \forall\ \mathbf{x} \neq 0$) [4, 7, 16, 19].

For power systems problems, solution times for direct methods grow faster than linearly (but usually less than quadratically) with matrix dimension [1]. The growth of the solution times for the conjugate gradient (CG) method can be slower. However, solution times for the conjugate gradient are not predictable a-priori. Solution times depend mainly on the condition number of the coefficient matrix A. Furthermore, unless A is positive definite, convergence cannot be assured. The time complexity of the CG method is roughly proportional to $\mathcal{O}(\tau(A)\,\kappa(A))$, and the space complexity is $\mathcal{O}(\tau(A))$ where $\tau(A)$ is the total number of nonzeros in A and $\kappa(A)$ is the condition number (see Shewchuk [21]).

The CG method has been used by others to solve linear equations associated with power system analysis in serial environments. Galiana et al. [7] consider the application of the PCG method for security analysis using the DC load flow approximation and for power flow analysis using the fast decoupled load flow (FDLF) [22]. They conclude that the PCG method can be faster than direct method by a factor of up to 60. Decker at al. [4, 5] also use the PCG for the dynamic simulation of transient analysis problem. Nieplocha and Carroll [17] evaluate the effectiveness of iterative methods for the Weighted Least Squares (WLS) state estimation problem on vector and parallel computers. They conclude that the PCG method is well suited to parallel and vector processing. Pai et al. [19, 20] also use the CG method for the parallel solution of transient stability analysis.

The computational complexity of direct methods is critically dependent on the ordering of the matrix A. For the CG method, ordering is unimportant. However, for certain classes of preconditioners, ordering becomes, once again, important [3]. Also, taking advantage of the physical structure of the network can lead to better preconditioners [11].

Section 2 of the paper provides some theoretical results for specific classes of power system matrices to show when the PCG method can be considered as a viable alternative to direct methods. Section 3 describes and tests a new method (the XD method) to obtain preconditioners for symmetric positive definite problems. This new method has the property that it leads to positive definite ILU factors, whereas ordinary incomplete LU factorization fails to always produce positive definite matrices (see appendix B for details).

2 Theoretical Developments

One of the basic requirements for the CG method to converge is that matrix A be symmetric and positive definite. The use of the CG method for the solution of linear systems (1) usually results in a high number of iterations. A widely used practice is to apply a linear transformation called *preconditioning* to the linear sys-

96 SM 563-7 PWRS A paper recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the 1996 IEEE/PES Summer Meeting, July 28 - August 1, 1996, in Denver, Colorado. Manuscript submitted January 2, 1996; made available for printing June 25, 1996.

tem at hand [15, 18]. The CG method applied to a linearly transformed system is called *preconditioned conjugate gradient (PCG)*. When the preconditioner matrix is also positive definite, convergence can be assured. Furthermore, solving a preconditioned problem usually results in a great decrease in the number of required iterations to attain any desired tolerance, resulting in much faster solution times than for the case without preconditioners. Although the CG method may converge with a non-positive definite preconditioner matrix, a positive definite preconditioner can guarantee convergence.

Subject to some mild assumptions, the coefficient matrix B of the DC load flow is symmetric and positive definite. The coefficient matrices B' and B'' of the FDLF method are also symmetric and positive definite (the former in the absence of phase shifter transformers, which destroys the the symmetry of B'.) However, since we are dealing with the approximations, the off diagonal terms corresponding to phase shifters can be made equal in the formulation.

A stronger property than positive definite is when a matrix is also an M-matrix[1]. This section also shows that these power flow matrices are, in fact, M-matrices. The reason this is important is because the incomplete LU factorization of an M-matrix remains positive definite (see appendix B for more details on the incomplete LU factorization). Hence, the PCG method can be used along with the incomplete LU factors of matrix A as an alternative to direct methods for the solution without fear of convergence problems of linear equations arising from load flow problem using the FDLF method. It can also be used for the linear sets of equations arising from contingency analysis using the DC load flow approximation.

The CG method can also be used for state estimation problem to solve linear equations of the form:

$$H^T H \mathbf{x} = H^T \mathbf{b} \tag{2}$$

where the gain matrix $H^T H$ is symmetric and positive definite by construction. However, the gain matrix is not necessarily an M-matrix or an H-matrix[2]. Thus, the incomplete factorization of a state estimation gain matrix is not necessarily positive definite [12]. Hence, incomplete LU factorization may not be a good preconditioner for such sets of linear equations. An alternative method to obtain incomplete LU factors, which is guaranteed to be positive definite, is proposed and tested in section 3.

Proposition 1 *If a network of (all positive) series reactances is considered, the coefficient matrix B associated with the DC load flow for this network is symmetric and positive definite. (*Note: *Any series line compensation is assumed to be strictly less than 100%.)*

[1]A real square matrix $B = (b)_{ij}$ with $b_{ij} \leq 0$ for all $i \neq j$, is an *M-matrix* if B is nonsingular and $B^{-1} \geq 0$, i.e., every element of B^{-1} is non-negative.

[2]Matrix $B = (b)_{ij}$ is said to be an *H-matrix* if the matrix $A = (a)_{ij}$ with $a_{ii} = b_{ii}$ and $a_{ij} = -|b_{ij}|$ for all $i \neq j$ is an M-matrix. Thus, an H-matrix is a matrix that can be reduced to an M-matrix.

Proof 1 *(Extension to that of Galiana et al. [7]): Construct an analogous purely resistive network, with the property that the solution to this analogous network is the same as the solution to the original network. Let inductance x_{ij} be analogous to a resistance r_{ij}, the vector of power injections be analogous to a vector of current injections, and the vector of phase angles be analogous to a vector of nodal voltages for a grounded resistive[3] network, as shown in Figure 1. The matrix B of the DC load flow becomes analogous to the conductance matrix G of the resistive network. Using the energy conservation principle of a grounded resistive network, one can show that G is positive definite. That is, the losses in the network can be given as:*

$$P_{Losses} = V^T G V. \tag{3}$$

Because the losses are positive in a purely resistive grounded network $P_{Losses} > 0$, G must be positive definite. As a result, matrix B is also positive definite. □

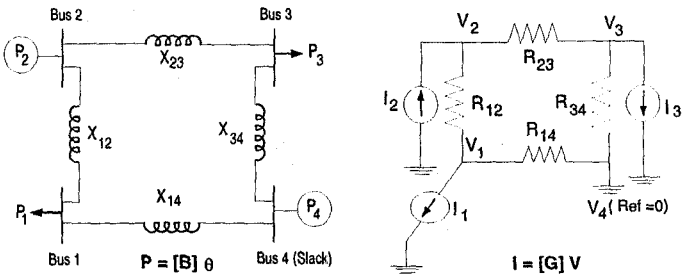

Fig. 1: *Grounded resistive network analog of lossless transmission network.*

Meijerink and van der Vorst [15] proved that any incomplete LU factors of an M-matrix are positive definite. Manteuffel [14] generalized this proof to the case of H-matrices. Some matrices arising from power system problems are, in fact, M-matrices. Hence, their incomplete LU factors can be used as preconditioners for the CG method.

Proposition 2 *The coefficient matrix B associated with the DC load flow of a connected network is irreducible[4].*

Proof 2 *Assume B is reducible. Then there exists a nonsingular permutation matrix P such that:*

$$P B P^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix} \tag{4}$$

[3]A resistive network means a network that has only positive resistances, which implicitly assumes that the series capacitive compensation of the series branches of the original network is strictly below 100%.

[4]A square matrix $B = (b)_{ij}$ is *reducible* if there exists a nonsingular permutation matrix P such that:

$$P B P^T = \begin{pmatrix} B_{11} & B_{12} \\ 0 & B_{22} \end{pmatrix},$$

where B_{11} is an $r \times r$ matrix, B_{22} is a $(n-r) \times (n-r)$ matrix, $1 \leq r < n$ and n is the dimension of B. Matrix B is *irreducible* if no such permutation matrix exists.

where B_{11} is an $r \times r$ matrix, B_{22} is a $(n-r) \times (n-r)$ matrix, $1 \leq r < n$ and n is the dimension of B. By construction, matrix B is a symmetric matrix, which means that B_{12} in (5) is zero.

$$B = P^T \begin{pmatrix} B_{11} & 0 \\ B_{12} & B_{22} \end{pmatrix} P \tag{5}$$

This leads to a decoupled network, which is a contradiction to the connected network assumption. □

Proposition 3 *If A is a* Stieltjes *matrix*[5], *then it is also an M-matrix. If A is in addition irreducible, then $A^{-1} > 0$ (all nonzero entries of A^{-1} are positive).*

Proof 3 *Corollary 3 of Theorem 3.11, Varga [24].* □

Proposition 4 *The matrix B associated with the DC load flow of a connected network is an M-Matrix.*

Proof 4 *By construction $b_{ij} \leq 0$ and B is symmetric. Furthermore, proposition 1 proved that B is positive definite. These three properties define a Stieltjes matrix. By proposition 3 we conclude that B is an M-matrix.* □

The B' and B'' matrices of the FDLF are formed in a manner similar to that of the DC load flow. Using similar arguments, it can be shown that these matrices are also M-matrices, hence their incomplete factors remain positive definite [2]. Moreover, the pivots of incomplete factors are larger than those of full factors [14], thus the convergence of incomplete factorization is assured. The *gain* matrices of weighted least square (WLS) are not necessarily M-matrices. As a result, their incomplete factors are not always positive definite as shown in Tables 2 and 3.

3 The XD method

The incomplete Cholesky factors of an ordinary symmetric positive definite matrix are not necessarily positive definite. Diagonal entries L_{ii} can become zero or negative, in which case the Cholesky decomposition fails if the LL^T form is used. This is illustrated by the following example of Kershaw [12]. The zero level incomplete $L\,D\,L^T$ factorization of symmetric positive definite matrix A below has a negative entry on its diagonal.

$$A = \begin{pmatrix} 3 & -2 & 0 & 2 \\ -2 & 3 & -2 & 0 \\ 0 & -2 & 3 & -2 \\ 2 & 0 & -2 & 3 \end{pmatrix} \quad D = \begin{pmatrix} 3 & & & \\ & 1.67 & & \\ & & 0.6 & \\ & & & -5 \end{pmatrix}$$

The PCG method may not converge since it requires the preconditioner to be positive definite [18]. One solution to this problem, suggested by Kershaw [12], is to replace a negative L_{ii} with a positive number and continue the decomposition. The positive number can either be the previous diagonal entry or the sum of absolute values of nonzeros in the *i-th* row. Since incomplete factorization itself is an approximate factorization, this process works for some cases but not all. This approach was tested on power system matrices and results were not encouraging.

This section describes an alternative way of computing ILU entries. This method is not a *computationally* competitive alternative to ILU for the cases where only a single solution of a set of linear equations is required. Rather, the method is offered for two reasons: The main reason is that the use of the method can be justified for specific problems, such as for the solution of linear equations as part of the solution of nonlinear problems, slowly varying time-dependent systems and linear systems with multiple right hand sides. The second reason for offering the XD method is to demonstrate that some modifications to an ordinary ILU algorithm can result in significantly better numerical performance.

The XD ("eXact then Discard") algorithm is simple:

Algorithm 1 *XD*

- *Order the matrix to reduce factorization fills,*
- *Perform a complete ordinary LDU factorization,*
- *Discard those entries in the factored matrix that do not correspond to nonzero entries in the original matrix (or to level one fills for the ILU_1 method, or to level two fills for the ILU_2 method etc.).*

This can also be written as:

$$ILU_m = L\,D\,U + P(m) \tag{6}$$

where P is a perturbation matrix and m is an integer defining level of fills. For example, ILU_0 means keep those entries in $L\,D\,U$ that correspond to original nonzero positions and ILU_1 means keep nonzeros in $L\,D\,U$ corresponding to original nonzero and level one fill positions. Thus, smaller m indicates a larger perturbation P. When $m = n$, matrix dimension, $P(m) = 0$, i.e., no perturbation is done to $L\,D\,U$.

When solving linear equations associated with the aforementioned problems using the PCG method, the XD method can be used to obtain a 'good' ILU preconditioner once and be used for all the successive linearizations of WLS state estimation problem, for different iterations of nonlinear solver and/or for the later time steps of slowly varying systems. The preconditioner can be updated only when structural changes in the network take place or it can no longer result in fewer number of PCG iterations. Thus, compensating for the initial relatively costly setup.

3.1 Testing the XD method

This method is employed for solving the set of linear equations where the coefficient matrices are the *gain matrices*, which are symmetric and positive definite, of the WLS state estimation problem. The effect of the XD preconditioner is that convergence is attained for all cases, even for those cases that do not converge with ordinary ILU preconditioners. The cases where PCG

[5] A real square matrix $B = (b)_{ij}$ with $b_{ij} \leq 0$ for all $i \neq j$ is a *Stieltjes* matrix if B is symmetric and positive definite.

failed to converge corresponds to non-positive definite preconditioners, such as ILU_2. The statistics for the test matrices are presented in Table 1 and Table 2 summarizes the test results. It is interesting to note that even though ILU_0 and ILU_1 are positive definite, ILU_2 is not for some of the *gain* matrices.

Table 1: Statistics for the gain matrices ($G = H^T H$).

Bus number	14	57	68	118	300	414
n (G)	27	113	135	235	599	827
Rows (H)	27	113	135	235	599	827
Columns (H)	63	289	343	634	1518	2020
Non-zeros (H)	271	1347	1545	3283	7273	9631

Table 2: Number of iterations without ordering. τ is the number of nonzeros and κ is the condition number. ILU_i refers to level i incomplete LU preconditioner.

Matrix	n	$\tau(A)$	$\kappa(A)$	PCG Iterations		
				ILU_0	ILU_1	ILU_2
G14	27	345	$1.3 \cdot 10^4$	10	5	2
G57	113	1903	$4.8 \cdot 10^4$	44	13	13
G68	135	2517	$8.3 \cdot 10^5$	51	20	8
G118	235	5029	$5.2 \cdot 10^5$	123	21	*
G300	599	11387	$2.6 \cdot 10^8$	333	99	*
G414	827	13867	$5.0 \cdot 10^7$	309	62	13

The CG method does not converge with ILU_2 because ILU_2 is not positive definite.
* : Did not converge to a prescribed precision of 10^{-5}.

Table 3: PCG iterations using level-based ILU on the normal equations. Solution tolerance is 10^{-5}.

n	ILU_0			ILU_1			ILU_2		
	md	**mmd**	**gps**	**md**	**mmd**	**gps**	**md**	**mmd**	**gps**
27	4	4	8	0**	0	4	0	0	0
113	19	24	48	11	10	9	7	8	*
135	18	20	68	10	10	12	5	6	*
235	103	97	72	16	14	24	6	7	*
599	210	293	400	27	35	75	*	*	*
827	28	35	339	11	6	40	3	2	31

The CG method does not converge with ILU_2 because ILU_2 is not positive definite.
** : All fills added, equivalent to a direct solver.
* : Did not converge to a prescribed precision of 10^{-5}.

The effect of ordering the equations when performing ILU can be significant [3, 6][6]. The effect of ordering on the convergence characteristics of the ILU preconditioner when applied to gain matrices is illustrated in Table 3. Three ordering algorithms are used: the minimum degree (**md**) [23], the multiple minimum degree (**mmd**) ordering [13] and the Gibbs-Poole-Stockmeyer method (**gps**) [8]. The first two are considered because they are the most widely used algorithms for direct solvers. The third one is included because it offers a greater degree of localization than the other two, and localization has been mentioned to favor iterative solvers [6]. Table 4 shows the effect of ordering on the XD method.

Table 4: Number of iterations using the XD based ILU preconditioner on the normal equations. Solution tolerance is 10^{-5}.

n	ILU_0			ILU_1			ILU_2		
	md	**mmd**	**gps**	**md**	**mmd**	**gps**	**md**	**mmd**	**gps**
27	5	5	8	0	0	4	0	0	0
113	20	21	27	8	7	7	4	6	9
135	13	18	23	9	8	8	5	4	4
235	33	38	49	9	10	15	4	5	7
599	108	75	383	14	18	30	7	7	9
827	19	19	152	5	6	18	2	2	3

Though expensive to compute, the reliable convergence behavior and the reduction in PCG iterations makes the XD based ILU preconditioner a better alternative to level-fills based ILU preconditioners for ill conditioned problems.

3.2 Theoretical basis for the XD method

It is of interest to explore whether factors the obtained using the XD method always lead to positive definite preconditioners. Consider some facts from linear algebra.

Fact 1: The eigenvalues of a matrix are continuous functions of its entries [10].

Fact 2: The eigenvalues of a diagonal matrix are the diagonal entries.

Fact 3: Let us write matrix A as $A = D + B$, where $D = \text{diag}\{a_{11}, a_{22}, \cdots, a_{nn}\}$ and all diagonal entries of B are zero. Define:

$$A_\varepsilon = D + \varepsilon B, \quad \varepsilon \in [0, 1]. \tag{7}$$

Observe that $A_0 = D$ and $A_1 = D + B = A$. If matrix A is diagonally dominant, then B is small. Thus, one can infer that eigenvalues, which are equal to diagonal entries with $\varepsilon = 0$, do not change too much as ε tends to one for a fixed matrix B.

These facts can be used to obtain a bound on changes in the eigenvalues for ILU_m in (6) so that the ILU is still positive definite. For $m = n$, the matrix is positive definite. As m becomes smaller, however, the ILU

[6]Ordering should be used with caution, since the amount of work required to perform the ordering can overshadow the amount of work necessary to carry out the rest of the computation.

may no longer be be positive definite. Let us pose the problem as follows. Let,

$$A = A(\varepsilon) \quad \varepsilon \in [0,1] \tag{8}$$

where $A(0)$ is assumed to be a symmetric and positive definite matrix with a simple smallest eigenvalue $\lambda(0)$. What is the amount of change in this simple smallest eigenvalue as ε varies so that $A(\varepsilon)$ is still positive definite? The answer to this question is explained in Section 3.2.1.

3.2.1 Effect of perturbations on the smallest eigenvalue

Assume that matrix $A(\varepsilon)$ is differentiable and that any perturbation to its entries is symmetric. (Discarding nonzero entries for ILU is a symmetric operation.) Let $\mathbf{v}_0$ and $\mathbf{w}_0$ be right and left eigenvectors corresponding to $\lambda(0)$ respectively. The vectors $\mathbf{v}(\varepsilon)$ and $\mathbf{w}(\varepsilon)$ can be chosen such that $\mathbf{v}(\varepsilon)^T\,\mathbf{w}(\varepsilon) = 1$ for any ε. The objective is to find a first order approximation to $\lambda(\varepsilon)$ about $\varepsilon = 0$ from (9).

$$\begin{aligned} \lambda(\varepsilon)\mathbf{w}(\varepsilon)^T\,\mathbf{v}(\varepsilon) &= \mathbf{w}(\varepsilon)^T\,A(\varepsilon)\,\mathbf{v}(\varepsilon) \\ \lambda(\varepsilon) &= \mathbf{w}(\varepsilon)^T\,A(\varepsilon)\,\mathbf{v}(\varepsilon) \end{aligned} \tag{9}$$

Differentiating (9) with respect to ε [10]:

$$\begin{aligned} \tfrac{d\lambda(\varepsilon)}{d\varepsilon}|_{\varepsilon=0} &= \tfrac{\partial \mathbf{w}(\varepsilon)^T}{\partial \varepsilon} A_0\,\mathbf{v}_0 + \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 + \mathbf{w}_0^T\,A_0\,\tfrac{\partial \mathbf{v}(\varepsilon)}{\partial \varepsilon} \\ &= \tfrac{\partial \mathbf{w}(\varepsilon)^T}{\partial \varepsilon}\,\lambda_0\,\mathbf{v}_0 + \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 + \lambda_0\,\mathbf{w}_0^T\,\tfrac{\partial \mathbf{v}(\varepsilon)}{\partial \varepsilon} \\ &= \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 + \lambda_0\{\tfrac{\partial \mathbf{w}(\varepsilon)^T}{\partial \varepsilon}\,\mathbf{v}_0 + \mathbf{w}_0^T\,\tfrac{\partial \mathbf{v}(\varepsilon)}{\partial \varepsilon}\} \\ &= \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 + \lambda_0\,\tfrac{d}{d\varepsilon}\{\mathbf{w}(\varepsilon)^T\,\mathbf{v}(\varepsilon)\} \\ &= \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 + \lambda_0\,\tfrac{d\{1\}}{d\varepsilon} \\ &= \mathbf{w}_0^T\,\tfrac{\partial A(\varepsilon)}{\partial \varepsilon}\,\mathbf{v}_0 \end{aligned} \tag{10}$$

Thus, it is possible to bound the perturbations to the eigenvalues of A. If this bound is smaller than the smallest negative eigenvalue, the perturbation will preserve the positive definite property the matrix. The following discussion shows a much stronger result.

3.2.2 Can the eigenvalues become negative?

The eigenvalues of a symmetric positive definite matrix A are all positive and real. That is, the characteristic polynomial of A, $det(\lambda I - A) = 0$, has all positive and real roots. The smallest root is the closest one to the origin. Matrix A can be decomposed as $A = L\,D\,L^T$ where D is a diagonal matrix with positive entries. If A is perturbed symmetrically (discarding entries of L and L^T) none of the eigenvalues can become complex since the matrix is still symmetric.

Symmetric perturbations to entries of L and L^T will perturb the characteristic polynomial for A. Assume that $\widetilde{L}$ and $\widetilde{L}^T$ are the perturbed matrices and that $\widetilde{A} = \widetilde{L}\,D\,\widetilde{L}^T$. The roots of the characteristic polynomial of $\widetilde{A}$ must remain real but need not be all positive. In the extreme case when all entries of L and L^T are discarded, $\widetilde{A}$ becomes equal to D, which has all positive real eigenvalues. By assumption, the original matrix A has all real positive eigenvalues. As the entries of L and L^T are discarded, in essence, the matrix $\widetilde{A}$ becomes more diagonally dominant. The eigenvalues of a diagonally dominant matrix are influenced mostly by the diagonal entries. This means that as more entries are discarded the eigenvalues move (or make small jumps since the perturbations are discrete) from their original values towards the diagonal entries of D, which are the final values for the eigenvalues of the perturbed matrix. The question that may arise from this conclusion is whether the eigenvalues (roots of characteristic polynomial) could jump to the left hand plane (i.e., become negative) and come back to right hand plane as ε tends to its final value. According to the following proposition, this can never happen. Assume that matrix A is decomposed as

$$A = L\,D\,L^T, \tag{11}$$

where D is diagonal and positive definite. Obtaining the incomplete LU factors from $L\,D\,L^T$ factors can be formulated as:

$$ILU_m = L(m)\,D\,L(m)^T,\ m \in [0,n], \tag{12}$$

where $L(m)$ is a unit lower triangular matrix whose nonzero topology (Top) can be determined by allowing up to level m fills, e.g., Top(ILU_0) =Top($L(0)\,D\,L(0)^T$) = Top(A).

Proposition 5 *Assume D is composed of strictly positive diagonal entries. Then $ILU_m \geq 0$ for any choice of $L(m)$.*

Proof 5 *The inequality*

$$\mathbf{x}^T\,ILU_n\,\mathbf{x} = \mathbf{x}^T\,L(n)\,D\,L(n)^T\,\mathbf{x} \tag{13}$$

holds for any nonzero $\mathbf{x}$ since A is assumed to be positive definite. Then,

$$\begin{aligned} \mathbf{x}^T\,ILU_m\,\mathbf{x} &= \mathbf{x}^T\,L(m)\,D\,L(m)^T\,\mathbf{x} \\ &= (L(m)^T\,\mathbf{x})^T\,D\,(L(m)^T\,\mathbf{x}) \\ &= \mathbf{y}^T\,D\,\mathbf{y} \end{aligned} \tag{14}$$

where $\mathbf{x} \neq 0$ and $\mathbf{y} = L(m)^T\,\mathbf{x}$. If $\mathbf{x}$ lies in the nullspace of $L(m)^T$ then $\mathbf{y}$ will be zero and $\mathbf{y}^T\,D\,\mathbf{y} = 0$. Otherwise $\mathbf{y}^T\,D\,\mathbf{y} > 0$ since D is positive definite. Hence, in the worst case $\mathbf{x}^T\,ILU(m)\,\mathbf{x}$ becomes positive semi definite. □

Corollary 1 *If the perturbations to $L(m)$ and $L(m)^T$ are restricted to the off-diagonal entries, then ILU_m will always be positive definite.*

4 Conclusions

The matrix B of the DC load flow and the matrices B' (in the absence of phase shifters) and B'' of the FDLF method are M-matrices. Thus, the incomplete factors of these matrices remain positive definite and can be used as preconditioners for the CG method.

A method has been proposed to obtain positive definite incomplete LU factors of a positive definite matrix which is not an M-matrix, such as the often ill-conditioned gain matrices that arise from state estimation. The method has been tested and the test results indicate that it has reliable convergence characteristics.

Acknowledgements

The authors acknowledge support from NSF grant ECS-9216308 and thank Dr. Harmohan Singh for his assistance with data cases for the gain matrices.

REFERENCES

[1] F. L. Alvarado. Computational complexity in power systems. *IEEE Transactions on Power Apparatus and Systems*, 95(4):1028–1037, July/August 1976.

[2] H. Dağ. *Iterative Methods and Parallel Computation for Power Systems*. PhD thesis, Electrical and Computer Engineering, The University of Wisconsin, 1996.

[3] H. Dağ and F. L. Alvarado. The effect of ordering on the preconditioned conjugate gradient method for power system applications. In *Proceedings of the North American Power Symposium*, Manhattan, KS, September 1994.

[4] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz. Parallel implementation of a power system dynamic simulation methodology using the conjugate gradient method. In *Power Industry Computer Applications Conference (PICA)*, pages 245–252, May 1991.

[5] I. C. Decker, D. M. Falcão, and E. Kaszkurewicz. Conjugate gradient methods for power system dynamic simulation on parallel computers. In *IEEE/PES Summer Meeting*, July 1995. Paper number 95 SM 506-6 PWRS.

[6] I. S. Duff and G. A. Meurant. The effect of ordering on preconditioned conjugate gradients. *BIT*, 29:635–657, 1989.

[7] F. D. Galiana, H. Javidi, and S. McFee. On the applications of a pre-conditioned conjugate gradient algorithm to power network analysis. In *Power Industry Computer Applications Conference (PICA)*, pages 404–410, April 1993.

[8] N. E. Gibbs, W. G. Poole, and P. K. Stockmeyer. An algorithm for reducing the bandwidth and profile of a sparse matrix. *SIAM Journal of Numerical Analysis*, 13:236–250, 1976.

[9] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.*, 49:409–436, 1952.

[10] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[11] H. Javidi, S. McFee, and F. D. Galiana. Investigation of eigenvalue clustering by modified incomplete Cholesky decomposition in power network matrices. In *Power System Computation Conference*, August 1993.

[12] D. Kershaw. The incomplete Cholesky-conjugate gradient method for the iterative solution of systems of linear equations. *Journal of Computational Physics*, 26:43–65, 1978.

[13] J. W. H. Liu. Modification of the minimum degree algorithm by multiple elimination. *ACM Transactions on Mathematical Software*, 11:141–153, 1985.

[14] T. Manteuffel. An incomplete factorization technique for positive definite linear systems. *Mathematics of Computation*, 34:473–497, April 1980.

[15] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. *Mathematics of Computation*, 31(137):148–162, January 1977.

[16] H. Mori, J. Kanno, and S. Tsuzuki. A sparsity-oriented technique for power system small signal stability analysis with a precondition conjugate residual method. *IEEE Transactions on Power Systems*, 8(3):1150–1158, August 1993.

[17] J. Nieplocha and C. C. Carroll. Iterative methods for the WLS state estimation on RISC, vector and parallel computers. In *Proceedings of the North American Power Symposium*, pages 355–363, Washington DC, October 1993.

[18] J. M. Ortega. *Introduction to Parallel and Vector Solution of Linear Systems*. Plenum Press, 1988.

[19] M. A. Pai, P. W. Sauer, and A. Y. Kulkarni. Conjugate gradient approach to parallel processing in dynamic simulation of power systems. In *American Control Conference*, pages 1644–1647, June 1992.

[20] M. A. Pai, P. W. Sauer, and A. Y. Kulkarni. A preconditioned iterative solver for dynamic simulation of power systems. In *International Symposium on Circuits and Systems*, pages 1279–1281, Seattle,WA, April/May 1995.

[21] J. R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, 1994.

[22] B. Stott and O. Alsaç. Fast decoupled load flow. *IEEE Transactions on Power Apparatus and Systems*, PAS-93(3):859–869, May/June 1974.

[23] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, 55(11):1801–1809, November 1967.

[24] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, 1962.

Appendix A The PCG algorithm [18]

Initialize
  Select $\mathbf{x}^0$
  Let $\mathbf{r}^0 = \mathbf{b} - A\mathbf{x}^0$
  Solve $M\tilde{\mathbf{r}}^0 \leftarrow \mathbf{r}^0$
  Let $\mathbf{p}^0 \leftarrow \tilde{\mathbf{r}}^0$
while $(\tilde{\mathbf{r}}^k, \mathbf{r}^k) \geq \epsilon$ *do*
  $\alpha_k \leftarrow -(\tilde{\mathbf{r}}^k, \mathbf{r}^k)/(\mathbf{p}^k, A\,\mathbf{p}^k)$
  $\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \alpha_k \mathbf{p}^k$
  $\mathbf{r}^{k+1} \leftarrow \mathbf{r}^k + \alpha_k A\,\mathbf{p}^k$
  Solve $M\tilde{\mathbf{r}}^{k+1} = \mathbf{r}^{k+1}$
  $\beta_k \leftarrow (\tilde{\mathbf{r}}^{k+1}, \mathbf{r}^{k+1})/(\tilde{\mathbf{r}}^k, \mathbf{r}^k)$
  $\mathbf{p}^{k+1} \leftarrow \tilde{\mathbf{r}}^{k+1} + \beta_k \mathbf{p}^k$
  $k \leftarrow k+1$
End

The expression $(\mathbf{x}, \mathbf{y})$ defines an inner (dot) product, i.e., $(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\,\mathbf{y}$. Bold face lowercase letters represent vectors, upper case letters represent matrices and Greek letters represent scalars.

The preconditioning steps are indicated. Matrix M is *a preconditioner matrix* that approximates matrix A. Matrix M is chosen such that the condition number of $M^{-1}A$ is improved relative to that of A. The simplest choice for matrix M is a diagonal matrix whose entries are the diagonal elements of A, but this choice does not improve the condition number of $M^{-1}A$ enough. A widely used preconditioner is described below.

The PCG method is dominated by matrix vector products. The number of iterations required for obtaining a solution to some tolerance is a function of both the condition number [18, 21] and of the quality of the preconditioner.

Appendix B Conventional ILU preconditioner

A popular preconditioner is based on the approximate (or incomplete) $L\,L^T$ or $L\,D\,L^T$ factorization of the matrix A. Due to fill-in, L can be considerably less sparse than A. Instead of using L, an approximation to L (denoted by $\widetilde{L}$) can be used:

$$A = \widetilde{L}\widetilde{L}^T + E \tag{15}$$

where $E \neq 0$ is an error matrix, and $\widetilde{L}$ is a lower triangular matrix, which is more sparse than L. This matrix implicitly defines $M = \widetilde{L}\widetilde{L}^T$ and it is called an *incomplete factorization* of A [18].

One of several possible ways of constructing $\widetilde{L}$ is to construct L and then discard those entries within L that correspond to zero positions in A. This approach is inefficient in that it requires the computation of the entire L matrix, which is often a costly step.

A more efficient way to obtain ILU factors is to perform an ordinary factorization of a matrix, but preclude the creation of any new non-zeros. This simple departure from ordinary LU factorization is the "level 0" ILU algorithm [14, 15].

The numerical performance of an ILU preconditioner can usually be improved upon if some fill-in are permitted to occur. The simplest possibility is to permit the occurrence of fills that involve original matrix entries, but preclude the creation of fill entries that depend on prior fills. This is the "level 1" ILU algorithm. Further levels of fills based on prior fills may be permitted, defining higher level incomplete factorization algorithms. The more levels that are included, the closer $\widetilde{L}$ can be expected to be to L. However, more accuracy also implies greater density. It has been observed that the number of iterations of the conjugate gradient method does not depend heavily on the number of nonzeros (fills) precluded, rather it depends on the norm of the error matrix E [6].

Biographies

Hasan Dağ (S'90) obtained the BS(Hon) degree in electrical engineering from Istanbul Technical University, Turkey, the MS and Ph.D. degrees from the University of Wisconsin-Madison in the Department of Electrical and Computer Engineering. His main interests are in the application of iterative methods to large sparse power systems problems and parallel solution of these problems. He will be a faculty in Istanbul Technical University, Turkey starting July 1, 1996.

Fernando L. Alvarado (F'93) obtained the BS degree from the National University of Engineering in Lima, Peru, the MS degree from Clarkson University, and a Ph.D. from the University of Michigan. He is currently a Professor at the University of Wisconsin-Madison in the Department of Electrical and Computer Engineering. His main interests are in computer applications to power systems and sparse matrix problems.

Discussion

M. A. Pai (Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL): This is an important paper in the growing literature on the use of iterative solvers for $Ax = b$. The key contribution of the paper lies in ensuring that the pre-conditioner is also positive definite while using the C-G method. As the authors point out this is not a serious problem in load flow using Stott's method except when phase shifters are present. Hence iterative methods offer a good alternative to LU factorization technique in load flow.

The XD method (algorithm 1) is not computationally efficient since it requires a LU factorization (Step 2) followed by discarding of certain entries (Step 3). In the WLS state estimation problem it is proven to be advantageous. In this way a good ILU(s) ($s = 0, 1, 2, \ldots$) pre-conditioners which is p.d is obtained. Tables 1-3 show failure to converge when ILU(2) is used whereas this is rectified when proper ordering is done (Table 4).

Can the authors comment on the following:

1. What are the computation times using ILU(0), ILU(1) and ILU(2) in Table 1-4?
2. Have the authors considered the ILU(τ) method where τ corresponds to numerical dropping instead of level fills [A]. In this technique the fill-in elements of the pre-conditioner are dropped when their numerical value is below a certain threshold.

Our experience in using iterative methods are when A is unsymmetric. This is when the dynamic response of a DAE system is computed using the simultaneous implicit method [B]. Reliable convergence is a problem while using ILU(s) method as our experience indicates but the present paper certainly offers some fresh ideas to tackle the problem.

References

[A] K. Gallivan, A. Sameh, Z. Zlatev, "A parallel hybrid sparse linear system solver," Computing Systems in Engineering, 1, 1990, pp. 183–185.

[B] A. Y. Kulkarni, M. A. Pai and P. W. Sauer, "Power system dynamic response calculation using Krylov subspace methods," 12th Power system Computation Conference, Dresden, Germany, Aug. 1996.

Manuscript received August 2, 1996.

H. Dağ*, F. L. Alvarado (*Istanbul Technical University, Turkey, University of Wisconsin–Madison): The authors appreciate the interest demonstrated by Prof. Pai in their paper.

Prof. Pai rightly observes that the proposed method is not computationally efficient in a general case. However, the use of the method can be justified for the following cases. If the linear system is a part of a nonlinear solver (such as the Newton-Raphson method for the load flow problem, the WLS state estimation problem, the transient stability problem) ILU preconditioner can be computed once and used for the subsequent nonlinear solver cycles. When a structural change takes place in the system or when linear solver part takes many iterations, the preconditioner is recomputed. This is what Kulkarni et al. [B] call a "dishonest preconditioner". In the WLS state estimation problem one has to deal with ill-conditioned linear systems and it is well-known that not only can the computed solution be wrong, the computed solution is also sensitive to changes in the data. Thus, a traditional ILU may not be a good preconditioner even if it is positive definite. The suggested method guarantees both a positive definite preconditioner and a numerically less corrupted approximate factors of matrix A.

As Prof. Pai points out, the failure to converge in Tables 1-3 is rectified in Table 4. However, the rectification is not due to ordering, it is due to the use of the XD method.

The computation time for the proposed XD method is approximately equivalent to that of a full factorization. Assuming matrices are ordered according to the multiple minimum degree algorithm of Liu [13], Table A1 presents cpu time for ILU preconditioners and the XD method. The cpu times are times obtained on a 486 processor based PC with a unix-like operating system. The PC has 8 MB of memory.

The cpu times for the XD method and the ILU_2 preconditioner are almost the same. This is because the majority of the factorization fills occur when up to level 2 fills are allowed. This can be seen in Table A2.

As for the second question, we have considered $ILU(\tau)$ preconditioner previously, however, with little success for most of the cases we have tried. The method does not behave consistently and fails for ill-conditioned problems. It sometimes results in failure to converge. For load flow matrices if system is stable, (the system is not near a collapse point), it is a good preconditioner. However, as the system approaches to a collapse point

Table A1: *The cpu (milliseconds) times for preconditioners*

Matrix	n	ILU_0	ILU_1	ILU_2	XD
G14	27	10	20	20	20
G57	113	90	180	200	250
G68	135	120	230	310	270
G118	235	270	450	510	530
G300	599	570	1120	1320	1460
G414	827	620	870	970	1010

Table A2: *Factorization fills for preconditioners*

Matrix	n	ILU_0	ILU_1	ILU_2	XD
G14	27	0	16	16	16
G57	113	0	762	1002	1294
G68	135	0	832	1346	1704
G118	235	0	1312	1798	1940
G300	599	0	3536	4974	5792
G414	827	0	1742	2486	2718

(the Jacobian becomes ill-conditioned), the preconditioner does not reduce the number of iterations greatly. We think that instead of randomly choosing τ, a relationship between an induced matrix norm and τ may be a better numerical dropping tolerance criterion. Javidi et al.'s [C1] experience also indicates that this is a better alternative.

We again thank Prof. Pai for his interest in out paper.

[C1] H. Javidi and S. McFee and F. D. Galiana, *Investigation of Eigenvalue Clustering by Modified Incomplete Cholesky Decomposition in Power Network Matrices*, Power System Computation Conference, Avignon, France, August 1993.

Manuscript received October 15, 1996.