Journal of Computational Mathematics, Vol.17, No.1, 1999, 89–96.

ITERATIVE METHODS WITH PRECONDITIONERS FOR INDEFINITE SYSTEMS[*1)]

Wei-qing Ren   Jin-xi Zhao
(*Department of Mathematics, Nanjing University, Nanjing 210008, China*)

Abstract

For the sparse linear equations $Kx = b$, where K arising from optimization and discretization of some PDEs is symmetric and indefinite, it is shown that the $L\overline{L}^T$ factorization can be used to provide an "exact" preconditioner for SYMMLQ and UZAWA algorithms. "Inexact" preconditioner derived from approximate factorization is used in the numerical experiments.

Key words: Generalized condition number, Indefinite systems, Factorization method

1. Introduction

Symmetric indefinite systems of linear equations arise in many areas of scientific computation. In this paper, we will discuss the solution of sparse indefinite system of the form

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}, \tag{1}$$

where $A \in R^{n\times n}$ is a symmetric positive definite matrix, $B \in R^{m\times n}$ has full row rank $m \le n$, $C \in R^{m\times m}$ is symmetric positive semidefinte, $f \in R^n$ and $g \in R^m$. In this case, the linear equations has the unique solution[8−10]. For simplicity, we denote the equations as $Kx = b$.

Discretizations of the Stokes equations or other PDEs produce the linear equations as (1). In optimization, when barrier or interior-point methods are applied to some linear or nonlinear programs, the Karush-Kuhn-Tucker optimality conditions also lead to a set of equations as (1). The system often need not to be solved exactly, therefore it is appropriate to consider iterative methods and preconditioners for the indefinite matrix K.

Our main aim is to present a simple result that shows how to use the $L\overline{L}^T$ factorization of K[8] to construct a preconditioner for iterative methods. The iterative methods to be discussed are the Paige-Saunders algorithm named as SYMMLQ[7] and the UZAWA method[1].

The rest of the paper is organized as follows. In section 2, we derive the exact preconditioner from the $L\overline{L}^T$ factorization and take inexact preconditioner from approximate factorization into account. In section 3, two iterative methods, SYMMLQ

* Received August 13, 1996
1)Project supported by the 863-plan of national High Technology of China

and UZAWA algorithms with preconditioners, are presented. In section 4, we present the numerical results and show the effectiveness of the preconditioners.

2. Preconditioning Indefinite System Using $L\overline{L}^T$ Factorization

The indefinite system $Kx = b$ arising from optimization and PDEs is often ill-conditioned. It is appropriate to take a positive definite matrix $M = CC^T$ as preconditioner for K so that $C^{-1}KC^{-T}$ has lower condition number or better eigenvalue distribution.

The following theorem presents the $L\overline{L}^T$ factorization of K. For more detail, see [8].

Theorem 2.1. *Given any symmetric indefinite matrix*

$$K = \begin{pmatrix} A & B^T \\ B & -C \end{pmatrix}, \tag{2}$$

where A, B and C are the same as that defined in (1). Then we have

$$K = L\overline{L}^T, \tag{3}$$

$$L = \begin{pmatrix} l_{11} & \\ l_{21} & l_{22} \end{pmatrix}, \quad \overline{L}^T = \begin{pmatrix} l_{11}^T & l_{21}^T \\ & -l_{22}^T \end{pmatrix}, \tag{4}$$

where $l_{11} \in R^{n\times n}$ and $l_{22} \in R^{m\times m}$ are lower triangular matrices, $l_{21} \in R^{m\times n}$.

The matrices l_{11}, l_{21} and l_{22} can be easily calculated from the following matrix equations:

$$A = l_{11}l_{11}^T, \tag{5}$$

$$B = l_{21}l_{11}^T, \tag{6}$$

$$C + l_{21}l_{21}^T = l_{22}l_{22}^T. \tag{7}$$

If we take LL^T as the preconditioner of K, it is easily verified that

$$\overline{K} = L^{-1}KL^{-T} = \begin{pmatrix} I_{11} & \\ & -I_{22} \end{pmatrix} \equiv J, \tag{8}$$

where $I_{11} \in R^{n\times n}$ and $I_{22} \in R^{m\times m}$ are identity matrices. This means the "perfect" preconditioner for K is the matrix

$$M = LL^T, \tag{9}$$

since the preconditioned matrix $\overline{K}$ has at most two distinct eigenvalues and the Paige-Saunders algorithm converges in at most two iterations[2]. The matrix LL^T is named as the exact preconditioner for K.

In practice, we will use "inexact" preconditioner, which is derived from the $L\overline{L}^T$ factorization of an approximation to K. For the inexact preconditioner, we have the following results. Let $\lambda_{\max}(K)$ denote the maximum eigenvalue of K, $\lambda_{\min}(K)$ the minimum eigenvalue. $\lambda_1(K)$, $\lambda_2(K)$ is the maximum and minimum of $|\lambda(K)|$ respectively. The generalized condition number of K is defined by $\kappa(K) = |\lambda_1(K)/\lambda_2(K)|$.

Theorem 2.2. *Let K, L and $\overline{L}$ be as those in Theorem 2.1 and let $M = LL^T$. Then for any symmetric positive definite matrix C, we have*

$$\kappa(C^{-1}K) \le \kappa(C^{-1}M). \tag{10}$$

Proof. In the following, $A \sim B$ means A is similar to B.

$$C^{-1}K = C^{-1}(LJL^T) \sim (L^TC^{-1}L)J = (L_1L_1^T)J \sim L_1^TJL_1, \tag{11}$$

where $L_1L_1^T$ is the Cholesky factorization of the poisitive definite matrix $L^TC^{-1}L$. Let

$$L_1 = \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix},$$

where $\tilde{l}_{11} \in R^{n\times n}$, $\tilde{l}_{22} \in R^{m\times m}$ and $\tilde{l}_{21} \in R^{m\times n}$. Then

$$\begin{aligned} L_1^TJL_1 &= \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} - \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} 0 & \\ & 2I_{22} \end{pmatrix} \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} \\ &= \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} - 2\begin{pmatrix} \tilde{l}_{21}^T\tilde{l}_{21} & \tilde{l}_{21}^T\tilde{l}_{22} \\ \tilde{l}_{22}^T\tilde{l}_{21} & \tilde{l}_{22}^T\tilde{l}_{22} \end{pmatrix} \equiv L_1^TL_1 - H_1. \end{aligned}$$

A simple calculation shows that H_1 is a positive semidefinite matrix. It is followed from Weyl theorem[4] that

$$\lambda_{\max}(L_1^TJL_1) \le \lambda_{\max}(L_1^TL_1). \tag{12}$$

On the other hand,

$$\begin{aligned} L_1^TJL_1 &= -\begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} + 2\begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} I_{11} & \\ & 0 \end{pmatrix} \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} \\ &= -\begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix}^T \begin{pmatrix} \tilde{l}_{11} & \\ \tilde{l}_{21} & \tilde{l}_{22} \end{pmatrix} + 2\begin{pmatrix} \tilde{l}_{11}^T\tilde{l}_{11} & \\ & 0 \end{pmatrix} \equiv -L_1^TL_1 + H_2, \end{aligned}$$

here H_2 is positive semidefinite, then

$$\lambda_{\min}(L_1^TJL_1) \ge \lambda_{\min}(-L_1^TL_1). \tag{13}$$

From (11), (12) and (13), we have

$$\lambda_1(C^{-1}K) = \lambda_1(L_1^TJL_1) \le \lambda_{\max}(L_1^TL_1) = \lambda_{\max}(C^{-1}M). \tag{14}$$

Similarly,

$$K^{-1}C = (L^{-T}JL^{-1})C \sim (L^{-1}CL^{-T})J = (L_2L_2^T)J \sim L_2^TJL_2, \tag{15}$$

where $L_2L_2^T$ is the Cholesky factorization of $L^{-1}CL^{-T}$. Let

$$L_2 = \begin{pmatrix} \hat{l}_{11} & \\ \hat{l}_{21} & \hat{l}_{22} \end{pmatrix},$$

where $\hat{l}_{11} \in R^{n\times n}$, $\hat{l}_{22} \in R^{m\times m}$ and $\hat{l}_{21} \in R^{m\times n}$. Then we have

$$\lambda_{\max}(K^{-1}C) = \lambda_{\max}(L_2^T J L_2) \leq \lambda_{\max}(L_2^T L_2) = \lambda_{\max}(M^{-1}C). \tag{16}$$

and

$$\lambda_{\min}(K^{-1}C) = \lambda_{\min}(L_2^T J L_2) \geq \lambda_{\min}(-L_2^T L_2) = \lambda_{\min}(-M^{-1}C). \tag{17}$$

It is followed from (16) and (17) that

$$\frac{1}{\lambda_2(C^{-1}K)} = \lambda_1(K^{-1}C) \leq \lambda_{\max}(M^{-1}C) = \frac{1}{\lambda_{\min}(C^{-1}M)}. \tag{18}$$

The result is derived from (14) and (18) immediately:

$$\kappa(C^{-1}K) = \frac{\lambda_1(C^{-1}K)}{\lambda_2(C^{-1}K)} \leq \frac{\lambda_{\max}(C^{-1}M)}{\lambda_{\min}(C^{-1}M)} = \kappa(C^{-1}M).$$

Theorem 2.3. *Let M and C be positive definite such that $\|I - C^{-1}M\|_2 \leq \epsilon$, where $\epsilon < 1$. Then*

$$\kappa(C^{-1}M) \leq \frac{1+\epsilon}{1-\epsilon}. \tag{19}$$

Proof. see [5].

The following result can be easily derived from (10) and (19).

Corollary 2.4. *Let K be the same as that in theorem 2.1. LJL^T is the $L\overline{L}^T$ factrization of K, $M = LL^T$, and the positive definite matrix C is an approximation to M satisfying $\|I - C^{-1}M\|_2 \leq \epsilon < 1$. Then*

$$\kappa(C^{-1}K) \leq \frac{1+\epsilon}{1-\epsilon}. \tag{20}$$

3. Iterative Methods with Preconditioners

In this section, we present two algorithms with preconditioners for the indefinite system (1).

3.1 The Preconditioned Paige-Saunders Method

This algorithm known as SYMMLQ is a conjugate-gradient-like method which can be applied to indefinite system. To solve $Kx = b$ with preconditioner $M = LL^T$, we apply SYMMLQ to the system

$$L^{-1}KL^{-T}y = L^{-1}b,$$

accumulating approximations to the solution $x = L^{-T}y$. In fact, it need not approximate, using the transformation$x = L^{-T}y$, we can have the following implementation.

Algorithm 3.1. (SYMMLQ)

$\beta_0 = \|L^{-1}b\|_2, q_0 = 0, q_1 = M^{-1}b/\beta_0,$

$s_{-1} = s_0 = 0, c_{-1} = c_0 = -1,$

$z_{-1} = z_0 = 1, \overline{w}_1 = q_1, x_0 = 0, j = 1.$

step 1. $\alpha_j = q_j^T K q_j$,
$r_j = (M^{-1}K - \alpha_j I)q_j - \beta_{j-1}q_{j-1}$,
$\beta_j = \|L^T r_j\|_2$,
if $\beta_j = 0$ then stop.
$q_{j+1} = r_j/\beta_j$;
step 2. $\epsilon_j = s_{j-2}/\beta_{j-1}$,
$\delta_j = -c_{j-2}c_{j-1}\beta_{j-1} + s_{j-1}\alpha_j$,
$\overline{\gamma}_j = -c_{j-2}s_{j-1}\beta_{j-1} - c_{j-1}\alpha_j$,
$\gamma_j = (\overline{\gamma}_j^2 + \beta_j^2)^{\frac{1}{2}}$;
step 3. $c_j = \overline{\gamma}_j/\gamma_j$,
$s_j = \beta_j/\gamma_j$;
step 4. $z_j = -(\epsilon_j z_{j-2} + \delta_j z_{j-1})/\gamma_j$,
$(w_j, \overline{w}_{j+1}) = (\overline{w}_j, q_{j+1})\begin{pmatrix} c_j & s_j \\ s_j & -c_j \end{pmatrix}$,
$x_j = x_{j-1} + z_j w_j$;
step 5. if $j = m + n$ then stop,
else $j = j + 1$, goto step 1.

To improve the convergence of SYMMLQ, the transformed matrices $\overline{K} = L^{-1}KL^{-T}$ should have a better condition than K, or a more favorable distribution of eigenvalues (clustered ± 1). In the next section, we will present the numerical results.

3.2 The UZAWA Method

The algorithm using preconditioner Q is presented in the following, which starts with an arbitray guess p_0.

Algorithm 3.2. (UZAWA)

for $i = 0$ untill convergence,do
$u_{i+1} = A^{-1}(f - B^T p_i)$
$p_{i+1} = p_i + \alpha Q^{-1}(Bu_{i+1} - Cp_i - g)$
enddo

here, α is a scalar parameter that must be determined prior to the iteration.

The convergence factor of the algorithm is $\rho(I - \alpha Q^{-1}(BA^{-1}B^T + C))$, which has the smallest value $(\kappa - 1)/(\kappa + 1)$ for the choice $\alpha = 2/(\lambda_1 + \lambda_2)$, where κ, λ_1 and λ_2 denote the generalized condition number, maximum and minimum eigenvalues of $Q^{-1}(BA^{-1}B^T + C)$ respectively[1]. From (5), (6) and (7), we can derive $BA^{-1}B^T + C = l_{22}l_{22}^T$, so if $l_{22}l_{22}^T$ is taken as preconditioner Q and $\alpha = 2/(\lambda_1 + \lambda_2)$, the UZAWA algorithm will converge one step.

In practice, once have got an approximative factorization of K, we can take $Q_A^{-1} \equiv (l_{11}l_{11}^T)^{-1}$ to replace A^{-1} in the algorithm. Then, we get the following "inexact" version of the UZAWA algorithm, which starts with $u_0 \equiv 0$ and an arbitrary initial guess p_0:

for $i = 0$ until convergence,do
$u_{i+1} = u_i + Q_A^{-1}(f - (Au_i + B^T p_i))$
$p_{i+1} = p_i + \alpha Q^{-1}(Bu_{i+1} - Cp_i - g)$
enddo

In the next section, we will use the "inexact" version of UZAWA algorithm.

4. Numerical Examples

Here we investigate the effectiveness of the preconditioner described in section 2. For test purposes, we have used MATLAB in PC486 to implement the SYMMLQ and UZAWA algorithms.

A special type of matrix K is considered, where $n = 200$ and $m = 100$. A is pentadiagonal and the nonzero entries are given by

$a_{i,i+1} = a_{i+1,i} = -1 \quad$ for $i = 1 : n-1$,
$a_{i,i+10} = a_{i+10,i} = -1 \quad$ for $i = 1 : n-10$,

the diagonals are random between 4 and 50. Entries of A that are not defined should be replaced by zeros. The matrices B and C are chosen with all entries random between 0 and 1 except that the diagonals of C are between 0 and 30.

We have computed all the eigenvalues of K, $M_0^{-1}K$ and $M_3^{-1}K$. The eigenvalues of K range from -103.75 to 63.23, and those of $M_0^{-1}K$, $M_3^{-1}K$ cluster around ± 1 as we have expected. It is evident that the tranformed systems have more favorable eigenvalue distributions than K. The generalized condition number is given in the following:

$$\kappa(K) = 1.0375e + 7,$$
$$\kappa(M_0^{-1}K) = 1.1654,$$
$$\kappa(M_3^{-1}K) = 1.0073.$$

Fig.1 and Fig.2 illustrate the behaviors of SYMMLQ and UZAWA on the preconditioned systems respectively. It is evident that for the preconditioned systems, less number of SYMMLQ and UZAWA iterations is required to reach a certain precision.

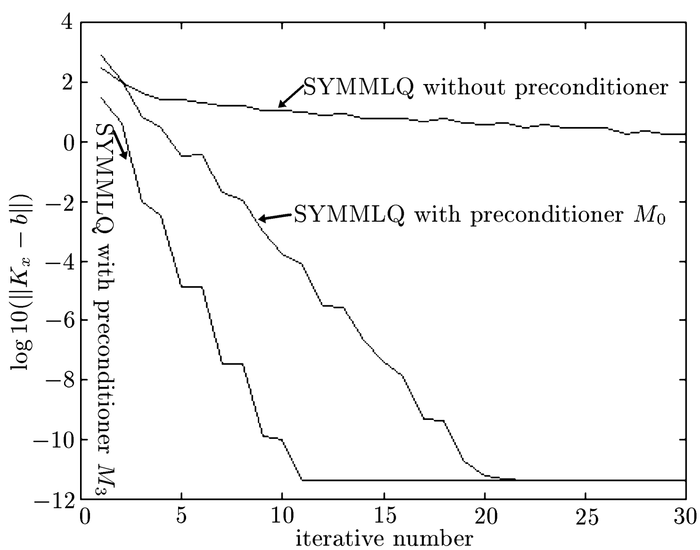

Fig.1. SYMML Q method

In the process of $L\overline{L}^T$ factorazition of K, two different incomplete cholesky factorizations introduced in [6] are applied to A. The corresponding preconditioners are denoted as $M_0 = L_0L_0^T$ and $M_3 = L_3L_3^T$, respectively.

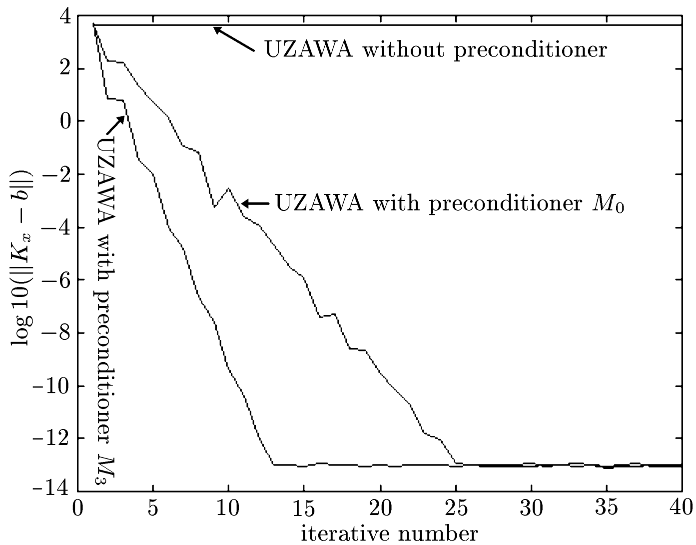

Fig.2. UZAWA method

The convergence factor of UZAWA without preconditioner($Q = I$) is close to 1, so the algorithm converges very slowly. The use of preconditioners M_0 and M_3 accelerates the convergence significiantly and the convergence factors are 0.07, 0.003 respectively.

Finally, we point out that in SYMMLQ algorithm, the LANCZOS vector will be computed. For ill-conditioned system, large number of iterations will lead to orthogonality loss and the reothogonalization is necessary. This makes the algorithm more complicated. For the preconditioned system, the algorithm converges after several iterations and the orthogonal loss is trivial, so the reorthogonalization is avoided.

The UZAWA method depends on the choice of parameter α. This makes it more difficult to implement the algorithm. For preconditioned system, we can take $\alpha = 1$ as an estimation.

5. Conclusion

For symmetric indefinite systems of linear equations of the form of (1), we have shown that the $L\overline{L}^T$ factorization can be used to provide a preconditioner for the Paige-Saunders algorithm SYMMLQ and UZAWA algorithm. The effect of the preconditioner is significiant in accelerating the convergence.

References

[1] Howard C.Elman, Multigrid and Krylov subspace methods for the discrete Stokes equations, Matrix Analysis and Parallel Computing, March 14 to 16 (1994), 151–164.

[2] Philip E.Gill, Walter Murray, Dulce B. Ponceleon, Michael A.Saunders, Preconditioners for indefinite systems arising in optimization, *SIAM J. Matrix Anal. Appl.,* **13**:1 (1992), 292–311.

[3] G.H. Golub, C.F. Van Loan, Matrix Computations, The Johns Hopkins University Press, 1983.

[4] Poger A.Horn, Charles A.Johnson, Matrix Analysis, Cambridge University Press, 1985.

[5] Hao Lu, Matrix compensation and diagonal compensation, *Journal of Computational and Applied Mathematics,* **63** (1995), 237–244.

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

[7] C.C. Paige, A. saunders, Solution of sparse indefinite linear equations, *SIAM J. Numer. Anal.,* **12** : 4 (1975), 617–629.

[8] Jin-xi Zhao, The generalized Cholesky factorization of symmetric indefinite matrix, *Mathematica Numerica Sinica,* **18** : 4 (1996), 442–448.

[9] Jin-xi Zhao, Matrix updating method for the constrained and weighted linear least squares problem, *Numerical Mathematics, A Journal of Chinese Universities,* **18** : 2 (1996), 97–103.

[10] Jin-xi Zhao, A modified conjugate direction method for computing the pseudoinverse, *J. Comput. Math.,* **12** : 2 (1994), 185–194.

[11] Jin-xi Zhao, A column recurrence method for solving linear least square problem, *J. Comput. Math.,* **14** : 4 (1996), 301–310.