A right preconditioner for the LSMR method

Afsaneh Hasanpour$^1$ and Maryam Mojarrab$^{1,2}$

$^1$Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran

$^2$Email: ma_mojarrab@math.usb.ac.ir

Abstract. The LSMR (Least Squares Minimal Residual) method is an absorbing solver that can solve linear system $Ax = b$ and least squares problem $\min \| Ax - b \|$ where $A$ is a sparse and large matrix. This method is based on the Golub-Kahan bidiagonalization process and sometimes it may converge slowly like other methods. In order to prevent this event, a right preconditioner for LSMR method is presented to solve large and sparse linear system which used for LSQR (Least Squares with QR factorization) method before. Numerical examples and comparing the preconditioned LSMR method to unpreconditioned LSMR method would show the effectiveness of the preconditioner. It is obtained from this paper that PLSMR (Preconditioned LSMR) method has a better performance in reducing the number of iterations and relative residual norm in comparing with the original LSMR method.

1. Introduction

Many large linear systems can be solved by iterative methods, especially those ones that are based on a Krylov subspace [1, 2]. The GMRES (Generalized Minimal Residual) and CMRH (Changing Minimal Residual method based on the Hessenberg process) methods [3, 4] are examples of the Krylov subspace methods. The LSQR introduced by Paige and Saunders is another method in this group [5]. The application of these methods are in scientific and engineering fields such as linear programming, geodetic survey problems, augmented Lagrange methods for computational fluid dynamics and natural factor method in structural engineering analysis [6-11].

The LSMR method is a Krylov subspace iterative method that was presented by Fong and Saunders for solving linear system $Ax = b$ [12]. This method is based on the procedure of Golub-Kahan bidiagonalization [13]. There is a mathematical equivalency between the LSMR method and the MINRES (Minimal Residual) method [14] applying to the normal equation $A^T Ax = A^T b$, while the LSQR method is equivalent to the CG (Conjugate Gradient) method [15]. Comparing to the LSQR that only monotonically decreases $\| r_k \| = \| b - Ax_k \|$, the LSMR method monotonically reduces both $\| r_k \|$ and $\| A^T r_k \|$ where $\| \cdot \|$ is the Euclidian norm.

The LSMR method generates two sets of vectors $v_1, v_2, ..., v_k$ and $u_1, u_2, ..., u_k$, made by Golub-Kahan bidiagonalization process, which form an orthogonal basis for Krylov subspaces $\kappa_k(A^T A, v_1)$ and $\kappa_k(A A^T, u_1)$ respectively where

$$\kappa_k(A^T A, v_1) = \text{span}\{v_1, A^T A v_1, ..., (A^T A)^{k-1} v_1\},$$

$$\kappa_k(A A^T, u_1) = \text{span}\{u_1, A A^T u_1, ..., (A A^T)^{k-1} u_1\}.$$ 

The performance of Krylov subspace methods can be approved by using an appropriate preconditioner or with efficient matrix splitting techniques [16-18]. A preconditioner for the LSQR method was introduced by Karimi et al [19] in which the incomplete inverse factor $R$ of $A^T A$ that is
used as a right preconditioner for the LSQR algorithm to solve the linear system $Ax = b$. If we apply the LSMR algorithm to the transformed system $ARy = b, x = Ry$

where $R$ is the inverse factor of the upper-lower factorization $(A^T A)^{-1} = RR^T$, then we will reach to the exact solution of the original system after one step. Because of this fact

$(AR)^T AR = R^T A^T AR = I$

where $I$ is the identity matrix of order $n$. So an incomplete inverse upper-lower factorization $(A^T A)^{-1} = \tilde{R}\tilde{R}^T$ could be reached, then $\tilde{R}$ can be used as a right preconditioner for the LSMR algorithm.

The LSMR method sometimes converge slowly like other solvers. In this study a right preconditioner is proposed which is based on $C$-orthogonalization, where $C$ is a symmetric positive definite matrix. The preconditioned least squares algorithm is presented. The preconditioned and unpreconditioned methods are applied to solve different linear systems and the results are compared together.

This paper is structured as follows. There is a brief describe about LSMR method in section 2. In Section 3 the right preconditioned LSMR method is presented. Some numerical experiments are given in Section 4 to illustrate the robustness of the new preconditioned method. Finally, conclusions of this study are summarized in Section 5.

2. A review of the LSMR method

In this section a brief discussion about the LSMR method is presented [12]. The LSMR method is an iterative method that is appropriate to solve linear systems of the form $Ax = b$, where $A$ is a sparse and large matrix of order $n$ and $x, b \in \mathbb{R}^n$.

The matrix $A$ can be transformed to the lower bidiagonal form by using Golub-Kahan bidiagonalization process [13]. In the following a brief description of the method is presented.

The Golub-Kahan bidiagonalization procedure uses $b$ as a starting vector and reduces matrix $A$ to the lower bidiagonal form

$$
\begin{align*}
\beta_1 u_1 &= b, & \alpha_1 v_1 &= A^T u_1, \\
\beta_{i+1} u_{i+1} &= Av_i - \alpha_i u_i, & \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_i v_i, & i &= 1, 2, \ldots
\end{align*}
$$

where $u_i, v_i \in \mathbb{R}^n$ and the scalars $\alpha_i, \beta_i \geq 0$ are chosen such that $\|u_i\| = \|v_i\| = 1$. By setting

$$
U_k = [u_1, u_2, \ldots, u_k], \quad V_k = [v_1, v_2, \ldots, v_k],
$$

$$
B_k = \begin{pmatrix}
\alpha_1 & & & \\
\beta_1 & \alpha_2 & & \\
& \ddots & \ddots & \\
& & \beta_k & \alpha_k \\
& & & \beta_{k+1}
\end{pmatrix},
$$

So the recurrence relations (1) can be rewritten as follows.

$$
U_{k+1}(\beta_1 e_1) = b,
$$

$$
AV_k = U_{k+1} B_k,
$$

$$
A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T.
$$

By using the procedure Bidiagonalization, the method constructs an approximation solution $x_k = V_k y_k$, where $y_k \in \mathbb{R}^k$ and solves the least-squares problem $\min \|A^T r_k\|$ where $r_k = b - Ax_k$. For more detail about the LSMR algorithm see [12].

3. Construction of the preconditioner for LSMR method

In many cases the iterative methods have the sufficient accuracy, but not always. The LSMR is one of these methods and sometimes may have a low convergence rate. In this situation an acceleration technique could speed-up to the convergence rate. One way is using preconditioners to approve the method. Accordingly, in this section it is tried to present an appropriate preconditioner by using [19-21].
The right preconditioner for the LSMR method is defined as an upper triangular matrix $R$ which is based on the C-matrix product that is defined as follows.

$$(x, y)_C = y^T C x,$$

where $C$ is a symmetric positive definite matrix and $x, y \in \mathbb{R}^n$.

Unit basis vectors $e_1, e_2, ..., e_n \in \mathbb{R}^n$ can construct a set of C-orthogonal vectors $z_1, z_2, ..., z_n \in \mathbb{R}^n$ by using conjugate Gram-Schmidt according to C-matrix product. The algorithm of C-orthogonalization can start by $z_j = e_j, j = 1, 2, ..., n$. Then the following nested loop will be performed

$$z_i \leftarrow z_i - ((z_j, z_i)_C / (z_j, z_j)_C) z_j.$$

Since $A$ is a full rank matrix, $A^T A$ is a symmetric positive definite matrix, so $C = A^T A$ can be used. By using $Z = (z_1, z_2, ..., z_n)$ and $D = Z^T C Z$ the inverse upper-lower factorization $(A^T A)^{-1} = ZD^{-1} Z^T$ will be obtained. The matrix $R$ can be defined by using SPD (symmetric positive definite) matrix $D$ as $R = ZD^{-1/2}$ and the inverse upper-lower factorization $R R^T$ is approximately equal to $(A^T A)^{-1}$. The entries of $z_i$ are scanned in each update by using a dropping tolerance $0 < \tau < 1$, if they are smaller than $\tau$ they will discard. After discarding the new sparsified $\tilde{z}_i$ will be defined.

$$\tilde{Z} = (\tilde{z}_1, \tilde{z}_2, ..., \tilde{z}_n), \quad \tilde{D} = \text{diag} (\tilde{d}_1, \tilde{d}_2, ..., \tilde{d}_n), \quad d_j = \|A z_j\|^2,$$

and $\tilde{R} = \tilde{Z} \tilde{D}^{-1/2}$ will be the incomplete inverse factor of $A^T A$.

The LSMR method searches for the approximation of solution in Krylov subspaces $\kappa_k(A^T A, v_1)$. So the incomplete inverse factor $\tilde{R}$ can be used as a right preconditioner for this method.

**Remark.** Assume that $\tilde{R}$ is the inverse factor of $A^T A$. Then $(AR)^T AR \approx I$ [22].

C-orthogonalization algorithm can be written as follows.

**Algorithm 1.** C-orthogonalization algorithm

1. Let $z_j = e_j, j = 1, 2, ..., n$
2. For $j = 1, 2, ..., n$ Do
   1. For $i = j + 1, ..., n$ Do
      1. $z_i = z_i - ((z_j, z_i)_C / (z_j, z_j)_C) z_j$
   End Do.
3. Use a dropping strategy for the vector $z_i$
4. End Do.

Use $d_j = \|A z_j\|^2$.

**Algorithm 2.** Preconditioned LSMR algorithm

1. Set $x_0 = 0$
2. $\beta_1 = \|b\|$, $u_1 = b / \beta_1$, $q_1 = A^T u_1$, $\alpha_1 = \|R^T q_1\|$, $\nu_1 = \beta_1 \alpha_1$
3. Set $\tilde{\xi}_1 = \alpha_1 \beta_1$, $\tilde{\alpha}_1 = \alpha_1$, $\rho_0 = 1$, $\rho_1 = 1$, $\tilde{c}_0 = 1$, $\tilde{s}_0 = 0$, $h_0 = \nu_1$, $\tilde{h}_0 = 0$
4. For $i=1,2,\ldots$ until convergence, Do
   1. $d_i = \tilde{R} v_i$
   2. $w_{i+1} = A d_i - \alpha_i u_i$
   3. $\alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_i v_i$
   4. $\rho_i = (\tilde{\alpha}_i^2 + \beta_{i+1}^{-2})^{1/2}$
   5. $c_i = \rho_i$
   6. $s_i = \beta_{i+1}$
   7. $\theta_{i+1} = s_i (\tilde{\alpha}_{i+1}^{-1})$
   8. $\tilde{\alpha}_{i+1} = c_i (\tilde{\alpha}_{i+1}^{-1})$
   9. $\tilde{\theta}_i = -s_i \rho_i$
   10. $\tilde{\beta}_i = (\tilde{\xi}_{i-1} \rho_i)^2 + \theta_{i+1}^{-2}$
5. End Do.
\[ \tilde{c}_i = \tilde{c}_{i-1} \tilde{\rho}_i \]
\[ \tilde{s}_i = \frac{\tilde{s}_{i+1}}{\tilde{\rho}_i} \]
\[ \tilde{\xi}_i = c_i \tilde{s}_i \]
\[ \tilde{\xi}_{i+1} = -\tilde{s}_i \tilde{\xi}_i \]
\[ \tilde{h}_i = h_i - (\tilde{\xi}_i / (\rho_i \tilde{\rho}_{i-1}) \tilde{h}_{i-1} \]
\[ x_i = x_{i-1} - (\tilde{\xi}_i / (\rho_i \tilde{\rho}_i)) \tilde{h}_i \]
\[ h_{i+1} = v_{i+1} - (\tilde{\xi}_{i+1} / (\rho_i \tilde{\rho}_i)) \tilde{h}_i \]

If \(|\tilde{\xi}_{i+1}|\) is small enough then stop
End Do

4. Numerical examples

In this section some numerical experiments are used to show the effectiveness of the preconditioned LSMR method. In all the examples the initial guess for the method is considered as \(x_0 = 0\). The right-hand side vector is chosen such that the exact solution of linear system \(Ax = b\) will be a vector that all the entries are equal to 1. The performance of the preconditioned method is compared to the unpreconditioned LSMR method. The dropping tolerance of the C-orthogonalization is \(10^{-2}\). The stopping criterion \(\|A^T r_k\| < 10^{-10}\) is used and a maximum of 10000 iterations is allowed. All the examples were executed in double precision in MATLAB R2014a. The machine have been used is an Intel Core i5, CPU 3.1 GHz.

Example 1. The coefficient matrices which are used in this example are given from Florida Sparse Matrix Collection \([23]\). Table 1 contains the properties of these matrices where "nnz" denotes the number of nonzero elements of the matrix. Order shows the dimension of matrix and cond demonstrates the condition number of matrix with this definition \(\|A\| \cdot \|A^{-1}\|\). Table 2 denotes the number of iterations for PLSMR (P-Itr) and the LSMR method (Unp-Itr) and relative residual norm for both preconditioned and unpreconditioned LSMR methods are contained in this table. CPU time for preconditioned LSMR method (P-time) and the LSMR method (Unp-time) are computed in Table 2.

### Table 1. Test problems information.

| Matrix | Order | nnz | Cond   | Matrix Discipline                              |
|--------|-------|-----|--------|-----------------------------------------------|
| nos6   | 675   | 1965| 8e+06  | Linear equations in structural engineering    |
| fs7601 | 760   | 5976| 8.4e+03| Chemical kinetics                              |
| bfw782a| 782   | 7514| 1.7e+03| Electrical engineering                         |
| cde6   | 961   | 4681| 5e+02  | Computational fluid dynamics                  |
| pde900 | 900   | 4380| 2.9e+02| Partial differential equations                |
| pde2961| 2961  | 14585| 9.49e+02| Partial differential equations                |
| fidap037| 3565 | 67591| 2.26e+02| Finite element modelling                      |

### Table 2. Results for Example 1.

| Matrix | P-Itr | P-time | P-\|r_k\| | Unp-Itr | Unp-time | Unp-\|r_k\| |
|--------|-------|--------|------------|---------|----------|------------|
| nos6   | 4046  | 0.9713 | 9.40e-11   | 5764    | 1.3032   | 9.55e-11   |
| fs7601 | 1034  | 0.4634 | 6.21e-11   | 1885    | 0.5388   | 9.25e-11   |
| bfw782a| 1411  | 0.4559 | 9.14e-11   | 2739    | 1.0072   | 9.83e-11   |
| cde6   | 30    | 0.0240 | 3.87e-11   | 394     | 0.2592   | 8.63e-11   |
| pde900 | 113   | 0.1525 | 6.96e-11   | 592     | 0.3862   | 9.87e-11   |
| pde2961| 320   | 2.0645 | 8.93e-11   | 2462    | 14.887   | 9.95e-11   |
| fidap037| 436   | 4.0918 | 9.57e-11   | 799     | 6.7546   | 9.71e-11   |

Example 2. In this example some matrices of order \(n = 1000\) with special forms are used as coefficient matrix that can be defined as follows.
The first matrix is defined as follow.

\[
A_1 = \begin{pmatrix}
0 & -0.1 \\
0.1 & 1 & -0.1 \\
& 0.1 & 2 & -0.1 \\
& & & \ddots & \ddots \\
& & & & 0.1 & n-1
\end{pmatrix}
\]

The second matrix is \(A_2 = SDS^{-1}\) where \(S\) is an upper bidiagonal matrix with entries 1’s on the main diagonal and 0.9’s on the super diagonal, and \(D = diag(1, 2, \ldots, n)\) is a diagonal matrix. The last matrix can be defined as follow

\[
A_3 = \begin{pmatrix}
1 & 1 & 1 & \ldots & 1 & 1 \\
a_1 & 1 & 1 & \ldots & 1 & 1 \\
a_1 & a_2 & 1 & \ldots & 1 & 1 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
a_1 & a_2 & a_3 & \ldots & a_{n-1} & 1
\end{pmatrix}
\]

with \(a_i = 1 + \epsilon\) [24]. In this example \(\epsilon\) is considered equal to \(10^{-2}\). Figure 1 indicates the comparison between PLSMR and LSMR methods for matrices \(A_1, A_2, A_3\) in reducing relative residual norm. As we observe the preconditioned LSMR method reduces the residual norm of \(Ax = b\) and normal linear system \(A^TAx = A^Tb\) in less iterations.

Figure 1. Performance of the LSMR method and the PLSMR method in reducing relative residual norm for Example.
**Example 3.** In this example the following convection-diffusion equation is taken from [16].

\[
a u'' + b u' = 0 \quad 0 < x < 1
\]

\[
u(0) = 0, \quad u(1) = 1
\]

where \(a\) and \(b\) are constant and both of them are considered equal to 2. The backward difference formula for the first order derivative is used and makes the coefficient matrix \(A\) for linear system \(Ax = b\). Figure 2 illustrates the performance of the LSMR, preconditioned LSMR, LSQR and preconditioned LSQR method in reducing the residual norm of \(Ax = b\). It shows that the PLSMR method decreases \(\|r_k\|\) in fewer number of iterations as well as preconditioned LSQR method. The effectiveness of preconditioned method can be found from this figure.

![Figure 2](image.png)

**Figure 2.** Performance of LSMR, LSQR, PLSMR and PLSQR in reducing residual norm for Example 3.

5. **Conclusions**

In this paper a right preconditioner for the LSMR method is presented for solving large and sparse linear systems which is based on C-orthogonalization. The preconditioned method is compared with the LSMR method for number of iterations and reducing relative residual norm. The results demonstrate the effectiveness of the preconditioned method and it is found that the preconditioner improves the method specially for reducing the number of iterations.

**References**

[1] Freund R, Golub G H and Nachtigal N M 1992 Iterative solution of linear systems *Acta Numerica* **1** 57-100

[2] Wang M 2009 An iterative algorithm for a least squares solution of a matrix equation, *International Journal of Computer Mathematics* **87**(6) 1289-1298
[3] Morgan R B 2000 Implicitly Restarted GMRES and Arnoldi Methods for Nonsymmetric Systems of Equations, *SIAM Journal on Matrix Analysis and Applications* 21(4) 1112-1135

[4] Sadok H 1999 CMRH: a new method for solving nonsymmetric linear systems based on the Hessenberg reduction algorithm, *Numerical Algorithms* 20(4) 303-321

[5] Paige C C and Saunders M A 1982 LSQR: An algorithm for sparse linear equations and sparse least squares, *ACM Trans. Math. Software, 8* 43-71

[6] Bjorck A, Duff I S 1980 A direct method for the solution of sparse linear least squares problems. *Linear Algebra Appl.* 34 43-67

[7] Chen N Y 1975 Iterative methods for linear least-squares problems. Res. Rep. CS-75-04, Dep. of Computer Science, Univ. Waterloo, Waterloo, Ont., Canada

[8] Elfving T 1978 On the conjugate gradient method for solving linear least-squares problems. Res. Rep. L1TH-MAT-R-1978-3, Dep. Mathematics, Linkoping Univ., Link ping, Sweden

[9] Eorge A, Heath M T 1980 Solution of sparse linear least squares problems using Givens rotations. *Linear Algebra Appl.* 34 69-83

[10] Golub G H 1965 Numerical methods for solving linear least-squares problems. *Numer. Math.* 7 206-216

[11] Houshholder A S 1955 Terminating and non-terminating iterations for solving linear systems. *SIAM J. Appl. Math.* 3 67-72

[12] Fong D V L, Saunders M 2011 LSMR: An iterative algorithm For Sparse Least-Squares Problems, *SIAM J. Sci. Comput.* 33 2950-2971

[13] Golub G H and Kahan W 1965 Calculating the singular values and pseudo-inverse of a matrix, *J. Soc. Indust. Appl. Math. Ser. B Numer. Anal.* 2 205-224

[14] Paige C C and Saunders M A 1975 Solution of sparse indefinite systems of linear equations, *SIAM J. Numer. Anal.* 12 617-629

[15] Hestenes M R and Stiefel E 1952 Methods of conjugate gradients for solving linear systems, *J. Res. Nat. Bur. Standards* 49(6) 409-436

[16] Saad Y 2003 Iterative Methods for Sparse Linear Systems, SIAM

[17] Bai Z Z, Golub G H and Ng M. K 2003 Hermitian and skew-Hermitian splitting methods for non-Hermitian positive definite linear systems, *SIAM Journal on Matrix Analysis and Applications* 24(3) 603-626

[18] Bai Z Z, Golub G H, Lu L Z and Yin J F 2005 Block triangular and skew-Hermitian splitting methods for positive-definite linear systems, *SIAM Journal on Scientific Computing* 26(3) 844-863

[19] Karimi S, Toutounian F, Khojasteh Salkuyeh D 2008 A preconditioner for the LSQR algorithm, *Appl. Math. Comput.* 26 213-222

[20] Benzi M, Kouhia R, Tuma M 2001 Stabilized and block approximate inverse preconditioners for problems in solid and structural mechanics, *Comput. Method Appl. M.* 190 6533-6554

[21] Benzi M, Tuma M 2003 Robust preconditioner with low memory requirement for large sparse least squares problems, *SIAM J. Sci. Comput.* 25 499-512

[22] Karimi S and Zali B 2014 The block preconditioned LSQR and GL-LSQR algorithms for the block partitioned matrices, *Appl. Math. Comput.* 227 811-820

[23] Davis T A, University of Florida Sparse Matrix Collection, http://www.cise.ufl.edu/research/sparse/matrices

[24] Brown P N 1991 A theoretical comparison of the Arnoldi and GMRES algorithms, *SIAM Journal on Scientific and Statistical Computing* 12(1) 58-78