Iranian Journal of Numerical Analysis and Optimization
Vol. 5, No. 2, (2015), pp 11-28

The block LSMR algorithm for solving linear systems with multiple right-hand sides

F. Toutounian* and M. Mojarrab

Abstract

LSMR (Least Squares Minimal Residual) is an iterative method for the solution of the linear system of equations and least-squares problems. This paper presents a block version of the LSMR algorithm for solving linear systems with multiple right-hand sides. The new algorithm is based on the block bidiagonalization and derived by minimizing the Frobenius norm of the residual matrix of normal equations. In addition, the convergence of the proposed algorithm is discussed. In practice, it is also observed that the Frobenius norm of the residual matrix decreases monotonically. Finally, numerical experiments from real applications are employed to verify the effectiveness of the presented method.

Keywords: LSMR method; Bidiagonalization; Block methods; Iterative methods; Multiple right-hand sides.

1 Introduction

This paper is concerned with the solution of linear system of the form

$$AX = B, \quad A \in \mathbb{R}^{n\times n}, \quad B \in \mathbb{R}^{n\times s}, \quad s \ll n. \tag{1}$$

If A is large and sparse or sometimes not readily available, then iterative solvers may become the only choice. These solvers are categorized to the following three classes:

*Corresponding author
Received 23 April 2014; revised 6 August 2014; accepted 18 March 2015
F. Toutounian
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran.
The Center of Excellence on Modelling and Control Systems, Ferdowsi University of Mashhad, Iran. e-mail: toutouni@math.um.ac.ir

M. Mojarrab
Department of Applied Mathematics, Faculty of Mathematical Sciences, Ferdowsi University of Mashhad, Iran.
Department of Mathematics, University of Sistan and Baluchestan, Zahedan, Iran. email: mojarrab@stu-mail.um.ac.ir

The first class is the global methods. The term global is due to Saad [34] and has been further expanded by Jbilou et al. [21] with the global FOM and GMRES algorithms for matrix equations. These methods are based on the use of a global projection process onto a matrix Krylov subspace. References on this class include [2, 7, 8, 12, 13, 13, 21–23, 25–27, 32, 33].

The second class is the seed methods. The main idea of this kind of methods is briefed below. We first select a single system as the seed system and generate the corresponding Krylov subspace. Then we project all the residuals of the other linear systems onto the same Krylov subspace to find new approximate solutions as initial approximations. See [3, 5, 7, 18, 20, 30, 35] for details.

The last class is the block methods which are more suitable for dense systems with preconditioner. The first block solvers are the block conjugate gradient (Bl-CG) algorithm and the block biconjugate gradient (Bl-BCG) algorithm proposed in [28]. Variable Bl-CG algorithms for symmetric positive definite problems are implemented on parallel computers [19, 29]. If the matrix is symmetric, an adaptive block Lanczos algorithm and a block version of Minres method are devised in [17]. For nonsymmetric problems, the Bl-BCG algorithm [6, 28], the block generalized minimal residual (Bl-GMRES) algorithm [1, 1, 4, 7, 9–11, 36, 37], the block quasi minimum residual (Bl-QMR) algorithm [14], the block BiCGStab (Bl-BICGSTAB) algorithm [31], the block Lanczos method [34] and the block least squares (Bl-LSQR) algorithm [15] have been developed.

In this paper, we present a block version of LSMR algorithm [4] for solving the problem (1). Our algorithm is based on the block bidiagonalization [9]. We construct a simple recurrence formula for generating the sequences of approximations $\{X_k\}$ such that the Frobenius norm of $A^T R_k$ decreases monotonically, where $R_k = B - AX_k$.

Throughout this paper, we use the following notations. For two $n \times s$ matrices X and Y, we define the following inner product: $\langle X, Y\rangle = \mathrm{tr}(X^T Y)$, where $\mathrm{tr}(Z)$ denoted the trace of the square matrix Z. The associated norm is the Frobenius norm denoted by $\|\cdot\|_F$. We will use the notation $\langle\cdot,\cdot\rangle_2$ for the usual inner product in $\mathbb{R}^n$ and the associated norm denoted by $\|\cdot\|_2$. Finally, 0_s and I_s will denote the zero and the identity matrices in $\mathbb{R}^{s\times s}$.

The remainder of this paper is organized as follows. In Section 2, we give a sketch of the LSMR method and its properties. In Section 3, we present the block version of the LSMR algorithm. In Section 4, the convergence of the presented algorithm is considered. In Section 5, some numerical experiments on test matrices from the University of Florida Sparse Matrix Collection(Davis [7]) are presented to show the efficiency of the method. Finally, we make some concluding remarks in Section 6.

2 The LSMR algorithm

In this section, we present a brief of the LSMR algorithm [4], which is an iterative method for solving real linear system of the form

$$Ax = b,$$

where A is a matrix of order n and $x, b \in \mathbb{R}^n$.

LSMR algorithm uses an algorithm of Golub and Kahan [10], which is stated as procedure Bidiag 1 in [32] to reduce the augmented matrix $[b\ A]$ to the upper-diagonal form $[\beta_1 e_1\ B_k]$, where e_1 denotes the first column of the identity matrix. The procedure Bidiag 1 can be described as follows.

Bidiag 1 (Starting vector b; reduction to lower bidiagonal form)

$$\begin{aligned} &\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1, \\ &\left.\begin{aligned} \beta_{i+1} u_{i+1} &= A v_i - \alpha_i u_i, \\ \alpha_{i+1} v_{i+1} &= A^T u_{i+1} - \beta_{i+1} v_i, \end{aligned}\right\} \qquad i = 1, 2, \ldots \end{aligned} \tag{2}$$

The scalars $\alpha_i \geq 0$ and $\beta_i \geq 0$ are chosen so that $\|u_i\|_2 = \|v_i\|_2 = 1$. With the definitions

$$U_k \equiv [u_1, u_2, \ \ldots\ u_k], \quad V_k \equiv [v_1, v_2, \ \ldots\ , v_k], \quad B_k \equiv \begin{bmatrix} \alpha_1 & & & \\ \beta_2 & \alpha_2 & & \\ & \ddots & \ddots & \\ & & \beta_k & \alpha_k \\ & & & \beta_{k+1} \end{bmatrix},$$

$$L_{k+1} = [B_k \quad \alpha_{k+1} e_{k+1}], \quad V_{k+1} = \begin{bmatrix} V_k & v_{k+1} \end{bmatrix},$$

the recurrence relations (2) may be rewritten as

$$\begin{aligned} &U_{k+1}(\beta_1 e_1) = b, \\ &A V_k = U_{k+1} B_k, \\ &A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T = V_{k+1} L_{k+1}^T. \\ &A^T A V_k = A^T U_{k+1} B_k = V_{k+1} L_{k+1}^T B_k = V_{k+1} \begin{bmatrix} B_k^T \\ \alpha_{k+1} e_{k+1}^T \end{bmatrix} B_k, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\quad = V_{k+1} \begin{bmatrix} B_k^T B_k \\ \alpha_{k+1} \beta_{k+1} e_k^T \end{bmatrix}. \end{aligned}$$

This is equivalent to what would be generated by the symmetric Lanczos process with matrix $A^T A$ and starting vector $A^T b$. As we observe the procedure Bidiag1 will be stop if $A v_i - \alpha_i u_i = 0$ or $A^T u_{i+1} - \beta_{i+1} v_i = 0$, for some i. In exact arithmetic, we have $U_{k+1}^T U_{k+1} = I$ and $V_k^T V_k = I$, where I is the identity matrix.

Hence using procedure Bidiag 1 the LSMR method constructs an approximation solution of the form $x_k = V_k y_k$ which solves the least-squares problem $\min_{y_k} \|A^T r_k\|$, where $r_k = b - Ax_k$. The main steps of the LSMR algorithm can be summarized as follows.

Algorithm 1 LSMR algorithm

Set $\beta_1 u_1 = b$, $\alpha_1 v_1 = A^T u_1$, $\overline{\alpha}_1 = \alpha_1$, $\overline{\zeta}_1 = \alpha_1\beta_1$, $\rho_0 = 1$, $\overline{\rho}_0 = 1$, $\overline{c}_0 = 1$, $\overline{s}_0 = 0$, $h_1 = v_1$, $\overline{\mathrm{h}}_0 = 0$, $x_0 = 0$,

For $k = 1, 2, \ldots$, until convergence Do:

$\beta_{k+1} u_{k+1} = Av_k - \alpha_k u_k$,
$\alpha_{k+1} v_{k+1} = A^T u_{k+1} - \beta_{k+1} v_k$,
$\rho_k = (\overline{\alpha}_k^2 + \beta_{k+1}^2)^{\frac{1}{2}}$,
$c_k = \overline{\alpha}_k / \rho_k$,
$s_k = \beta_{k+1} / \rho_k$,
$\theta_{k+1} = s_k \alpha_{k+1}$,
$\overline{\alpha}_{k+1} = c_k \alpha_{k+1}$,
$\overline{\theta}_k = \overline{s}_{k-1} \rho_k$,
$\overline{\rho}_k = ((\overline{c}_{k-1}\rho_k)^2 + \theta_{k+1}^2)^{\frac{1}{2}}$,
$\overline{c}_k = \overline{c}_{k-1}\rho_k / \overline{\rho}_k$,
$\overline{s}_k = \theta_{k+1} / \overline{\rho}_k$,
$\zeta_k = \overline{c}_k \overline{\zeta}_k$,
$\overline{\zeta}_{k+1} = -\overline{s}_k \overline{\zeta}_k$,
$\overline{h}_k = h_k - (\overline{\theta}_k \rho_k / (\rho_{k-1}\overline{\rho}_{k-1}))\overline{h}_{k-1}$,
$x_k = x_{k-1} + (\zeta_k / (\rho_k \overline{\rho}_k))\overline{h}_k$,
$h_{k+1} = v_{k+1} - (\theta_{k+1}/\rho_k) h_k$,
If $|\overline{\zeta}_{k+1}|$ is small enough then stop,

End Do.

More details about the LSMR algorithm can be found in [4].

3 The block LSMR method

We first recall the block Bidiag 1 algorithm [9]. This algorithm is the basis for our block LSMR method.

The block Bidiag 1 procedure constructs the sets of the $n \times s$ block vectors $V_1, V_2, \ldots$ and $U_1, U_2, \ldots$ such that $V_i^T V_j = 0_s$, $U_i^T U_j = 0_s$, for $i \neq j$, and $V_i^T V_i = I_s$, $U_i^T U_i = I_s$; and they form the orthonormal basis of $\mathbb{R}^{n \times ks}$.

Block Bidiag 1 (Starting matrix B; reduction to block lower bidiagonal form)

$$
\begin{aligned}
&U_1 B_1 = B, \quad V_1 A_1 = A^T U_1, \\
&\left.\begin{aligned} U_{i+1} B_{i+1} &= A V_i - U_i A_i^T, \\ V_{i+1} A_{i+1} &= A^T U_{i+1} - V_i B_{i+1}^T, \end{aligned}\right\} \qquad i = 1, 2, \ldots, k,
\end{aligned} \tag{3}
$$

where $U_i, V_i \in \mathbb{R}^{n \times s}; B_i, A_i \in \mathbb{R}^{s \times s}$, and $U_1 B_1, V_1 A_1, U_{i+1} B_{i+1}, V_{i+1} A_{i+1}$ are thin QR decompositions of the matrices $B, A^T U_1, A V_i - U_i A_i^T, A^T U_{i+1} - V_i B_{i+1}^T$, respectively. With the definitions

$$
\overline{U}_k \equiv [U_1,\ U_2,\ \ldots\ ,\ U_k]\,,\ \overline{V}_k \equiv [V_1,\ V_2,\ \ldots\ ,\ V_k]\,,\ T_k \equiv \begin{bmatrix} A_1^T & & & \\ B_2 & A_2^T & & \\ & \ddots & \ddots & \\ & & B_k & A_k^T \\ & & & B_{k+1} \end{bmatrix},
$$

the recurrence relations (3) may be rewritten as:

$$
\begin{aligned}
&\overline{U}_{k+1} E_1 B_1 = B, \\
&A \overline{V}_k = \overline{U}_{k+1} T_k, \\
&A^T \overline{U}_{k+1} = \overline{V}_k T_k^T + V_{k+1} A_{k+1} E_{k+1}^T,
\end{aligned}
$$

where E_i is the $(k+1)s \times s$ matrix which is zero except for the rows i to $i+s$, which are the $s \times s$ identity matrix. We have also $\overline{V}_k^T \overline{V}_k = I_{ks}$ and $\overline{U}_{k+1}^T \overline{U}_{k+1} = I_{(k+1)s}$, where I_l is the $l \times l$ identity matrix. We define

$$
\overline{L}_{k+1} \equiv \left[\, T_k \ E_{k+1} A_{k+1}^T \,\right],
$$

then

$$
\begin{aligned}
A^T \overline{U}_{k+1} &= \overline{V}_{k+1} \overline{L}_{k+1}^T, \\
A^T A \overline{V}_k &= A^T \overline{U}_{k+1} T_k = \overline{V}_{k+1} \overline{L}_{k+1}^T T_k = \overline{V}_{k+1} \begin{bmatrix} T_k^T \\ A_{k+1} E_{k+1}^T \end{bmatrix} T_k \\
&= \overline{V}_{k+1} \begin{bmatrix} T_k^T T_k \\ A_{k+1} E_{k+1}^T T_k \end{bmatrix}.
\end{aligned} \tag{4}
$$

At iteration k we seek an approximate solution X_k of the form

$$
X_k = \overline{V}_k Y_k, \tag{5}
$$

where Y_k is an $ks \times s$ matrix. Let $\overline{B}_k \equiv A_k B_k$ for all k. Since

$$
\begin{aligned}
A^T R_k &= A^T B - A^T A X_k \\
&= V_1 A_1 B_1 - A^T A \overline{V}_k Y_k,
\end{aligned}
$$

we have

$$A^T R_k = V_1 \overline{B}_1 - \overline{V}_{k+1} \begin{bmatrix} T_k^T T_k \\ A_{k+1} E_{k+1}^T T_k \end{bmatrix} Y_k$$
$$= \overline{V}_{k+1}(E_1 \overline{B}_1 - \begin{bmatrix} T_k^T T_k \\ \overline{B}_{k+1} \overline{E}_k^T \end{bmatrix} Y_k), \quad (6)$$

where $\overline{E}_k$ is the $ks \times s$ matrix, which is zero except for kth s rows, which are the $s \times s$ identity matrix.

In the block LSMR algorithm, we would like to choose $Y_k \in \mathbb{R}^{ks \times s}$ which minimizes the Frobenius norm of $A^T R_k$. From (6), $A^T R_k$ can be written as

$$A^T R_k = \overline{V}_{k+1} \begin{bmatrix} \widetilde{E}_1 \overline{B}_1 - T_k^T T_k Y_k \\ -\overline{B}_{k+1} \overline{E}_k^T Y_k \end{bmatrix}, \quad (7)$$

where $\widetilde{E}_1$ is the matrix obtained from E_1 by deleting its last block row. But since the columns of the matrix $\overline{V}_{k+1}$ are orthonormal, it follows that:

$$\|A^T R_k\|_F^2 = \left\| \begin{bmatrix} \widetilde{E}_1 \overline{B}_1 - T_k^T T_k Y_k \\ -\overline{B}_{k+1} \overline{E}_k^T Y_k \end{bmatrix} \right\|_F^2 = \|\widetilde{E}_1 \overline{B}_1 - T_k^T T_k Y_k\|_F^2 + \|\overline{B}_{k+1} \overline{E}_k^T Y_k\|_F^2. \quad (8)$$

We now define the linear operators χ_k and ψ_k as follows.

For $Y \in \mathbb{R}^{ks \times s}$

$\chi_k(Y) = T_k^T T_k Y,$

and

$\psi_k(Y) = \overline{B}_{k+1} \overline{E}_k^T Y.$

Then the relation (8) can be expressed as

$$\|A^T R_k\|_F^2 = \|\chi_k(Y_k) - \widetilde{E}_1 \overline{B}_1\|_F^2 + \|\psi_k(Y_k)\|_F^2. \quad (9)$$

Therefore, Y_k minimizes the Frobenius norm of the quantity $A^T R_k$ if and only if it satisfies the following linear matrix equation

$$\chi_k^T(\chi_k(Y_k) - \widetilde{E}_1 \overline{B}_1) + \psi_k^T(\psi_k(Y_k)) = 0_s, \quad (10)$$

where the linear operators χ_k^T and ψ_k^T are the transpose of the operators χ_k and ψ_k, respectively. Therefore, (10) is also written as the following

$$(T_k^T T_k)^T (T_k^T T_k Y_k - \widetilde{E}_1 \overline{B}_1) + (\overline{B}_{k+1} \overline{E}_k^T)^T (\overline{B}_{k+1} \overline{E}_k^T Y_k) = 0_s. \quad (11)$$

Hence, Y_k is given by

$$Y_k = \widehat{T}_k^{-1} F_k,$$

where

$$\widehat{T}_k = (T_k^T T_k)^2 + \overline{E}_k \overline{B}_{k+1}^T \overline{B}_{k+1} \overline{E}_k^T, \qquad F_k = T_k^T T_k \widetilde{E}_1 \overline{B}_1. \tag{12}$$

We define the matrix $\overline{T}_k$ as follows:

$$\overline{T}_k = \begin{bmatrix} T_k^T T_k \\ \overline{B}_{k+1} \overline{E}_k^T \end{bmatrix} = \begin{bmatrix} \overline{A}_1 & \overline{B}_2^T & & \\ \overline{B}_2 & \overline{A}_2 & \ddots & \\ & \ddots & \ddots & \overline{B}_k^T \\ & & \overline{B}_k & \overline{A}_k \\ & & & \overline{B}_{k+1} \end{bmatrix},$$

where $\overline{A}_i = A_i A_i^T + B_{i+1}^T B_{i+1}$, for $i = 1, 2, \ldots, k$. Therefore

$$\widehat{T}_k = \overline{T}_k^T \overline{T}_k, \quad F_k = [(\overline{A}_1 \overline{B}_1)^T (\overline{B}_2 \overline{B}_1)^T 0_s \ldots 0_s]^T, \tag{13}$$

and the approximate solution of the system (1) is given by

$$X_k = \overline{V}_k \widehat{T}_k^{-1} F_k.$$

Suppose that using the QR decomposition [11], we obtain a unitary matrix $\overline{Q}_k$ such that

$$\overline{T}_k = \overline{Q}_k \begin{bmatrix} \overline{R}_k \\ 0_{s \times ks} \end{bmatrix}, \qquad \overline{R}_k = \begin{bmatrix} \overline{\alpha}_1 & \overline{\beta}_2 & \overline{\theta}_3 & & & & \\ & \overline{\alpha}_2 & \overline{\beta}_3 & \overline{\theta}_4 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \overline{\alpha}_{k-2} & \overline{\beta}_{k-1} & \overline{\theta}_k \\ & & & & \overline{\alpha}_{k-1} & \overline{\beta}_k \\ & & & & & \overline{\alpha}_k \end{bmatrix}, \tag{14}$$

where $\overline{R}_k$ is upper triangular as shown and $\overline{\alpha}_i$, $\overline{\beta}_i$, $\overline{\theta}_i$ are the $s \times s$ matrices. So,

$$X_k = \overline{V}_k (\overline{R}_k^T \overline{R}_k)^{-1} F_k.$$

By setting

$$\overline{P}_k = \overline{V}_k \overline{R}_k^{-1} \equiv \begin{bmatrix} P_1 & P_2 & \ldots & P_k \end{bmatrix},$$

and

$$\overline{F}_k = \overline{R}_k^{-T} F_k \equiv \begin{bmatrix} \varphi_1^T & \varphi_2^T & \ldots & \varphi_k^T \end{bmatrix}^T,$$

we have

$$\begin{aligned} P_k &= (V_k - P_{k-2} \overline{\theta}_k - P_{k-1} \overline{\beta}_k) \overline{\alpha}_k^{-1}, \\ X_k &= X_{k-1} + P_k \varphi_k. \end{aligned} \tag{15}$$

From (15) the residual R_k is given by

$$R_k = R_{k-1} - AP_k\varphi_k, \tag{16}$$

where AP_k can be computed from the previous AP_k's and AV_k by the simple update

$$AP_k = (AV_k - AP_{k-2}\overline{\theta}_k - AP_{k-1}\overline{\beta}_k)\overline{\alpha}_k^{-1}.$$

In addition, as [4], we show that the $\|R_k\|_F$ can be estimated by a simple formula. By transforming T_k to block upper-bidiagonal form using a QR factorization: $\begin{bmatrix} \widehat{R}_k \\ 0 \end{bmatrix} = \widehat{Q}_{k+1}T_k$ with $\widehat{Q}_{k+1} = \widehat{P}_k \dots \widehat{P}_1$, we have

$$\begin{aligned} R_k &= B - AX_k \\ &= U_1B_1 - A\overline{V}_kY_k \\ &= \overline{U}_{k+1}(E_1B_1 - T_kY_k) \\ &= \check{U}_{k+1}\widehat{Q}_{k+1}^T(\widehat{Q}_{k+1}E_1B_1 - \begin{bmatrix} \widehat{R}_k \\ 0 \end{bmatrix} Y_k). \end{aligned}$$

Since the columns of the matrices $\widehat{Q}_{k+1}$ and $\overline{U}_{k+1}$ are orthonormal, we have

$$\|R_k\|_F = \|\widehat{Q}_{k+1}E_1B_1 - \begin{bmatrix} \widehat{R}_k \\ 0 \end{bmatrix} Y_k\|_F. \tag{17}$$

With definitions

$$\widehat{Q}_{k+1}E_1B_1 = [\widetilde{\beta}_1^T \ \dots \ \widetilde{\beta}_{k-1}^T \ \dot{\beta}_k^T \ \ddot{\beta}_{k+1}^T]^T, \qquad \widehat{R}_kY = [\widetilde{\tau}_1^T \ \dots \ \widetilde{\tau}_{k-1}^T \ \dot{\tau}_k^T]^T, \tag{18}$$

the following Lemma shows that we can estimate $\|R_k\|_F$ from just the last two blocks of $\widehat{Q}_{k+1}E_1B_1$ and the last block of $\widehat{R}_kY_k$.

Lemma 1. *In* (17) *and* (18), $\widetilde{\beta}_i = \widetilde{\tau}_i$ *for* $i = 1, 2, \dots, k-1$.

Proof. The proof is similar to that of Lemma 3.1 in [4] (see [28]). □

For the Frobenius norm of A^TR_k, by using Theorem 1 (in section 4), we can also obtain the following simple formula:

$$\|A^TR_k\|_F^2 = \|A^TR_{k-1}\|_F^2 - \|\varphi_k\|_F^2, \qquad \text{with } \|A^TR_0\|_F = \|\overline{B}_1\|_F = \|\varphi_0\|_F.$$

Now we can summarize the above descriptions as the following algorithm.

Algorithm 2 Algorithm (Bl-LSMR)

Set $X_0 = 0_{n\times s}$,
Set $\overline{a}_0 = 0_s$, $\overline{b}_{-1} = 0_s$, $\overline{b}_0 = I_s$, $\overline{c}_0 = 0_s$, $\overline{d}_{-1} = 0_s$, $\overline{d}_0 = I_s$,
Set $P_{-1} = P_0 = 0_{n\times s}$,
Compute $U_1 B_1 = B$, $V_1 A_1 = A^T U_1$ (QR decomposition of B and $A^T U_1$),
Set $\overline{B}_1 = A_1 B_1$,
Set $\varphi_{-1} = 0_s$, $\varphi_0 = -\overline{B}_1$,
Set $\|A^T R_0\|_F = \|\varphi_0\|_F$,
For $k = 1, 2, \ldots,$ until convergence Do:
$\overline{W}_k = AV_k - U_k A_k^T$,
$U_{k+1} B_{k+1} = \overline{W}_k$ (QR decomposition of $\overline{W}_k$),
$\overline{A}_k = A_k A_k^T + B_{k+1}^T B_{k+1}$,
$\overline{S}_k = A^T U_{k+1} - V_k B_{k+1}^T$,
$V_{k+1} A_{k+1} = \overline{S}_k$ (QR decomposition of $\overline{S}_k$),
$\overline{B}_{k+1} = A_{k+1} B_{k+1}$,
$\dot{\beta}_k = \overline{d}_{k-2} \overline{B}_k^T$,
$\dot{\alpha}_k = \overline{c}_{k-1} \dot{\beta}_k + \overline{d}_{k-1} \overline{A}_k$,
$\overline{\beta}_k = \overline{a}_{k-1} \dot{\beta}_k + \overline{b}_{k-1} \overline{A}_k$,
$\overline{\theta}_k = \overline{b}_{k-2} \overline{B}_k^T$,
Compute an unitary matrix $\overline{Q}(\overline{a}_k, \overline{b}_k, \overline{c}_k, \overline{d}_k)$ such that
$\begin{bmatrix} \overline{a}_k & \overline{b}_k \\ \overline{c}_k & \overline{d}_k \end{bmatrix} \begin{bmatrix} \dot{\alpha}_k \\ \overline{B}_{k+1} \end{bmatrix} = \begin{bmatrix} \overline{\alpha}_k \\ 0 \end{bmatrix}$,
$\varphi_k = -\overline{\alpha}_k^{-T} (\overline{\theta}_k^T \varphi_{k-2} + \overline{\beta}_k^T \varphi_{k-1})$,
$P_k = (V_k - P_{k-2} \overline{\theta}_k - P_{k-1} \overline{\beta}_k) \overline{\alpha}_k^{-1}$,
$X_k = X_{k-1} + P_k \varphi_k$,
$R_k = R_{k-1} - A P_k \varphi_k$,
$\|A^T R_k\|_F^2 = \|A^T R_{k-1}\|_F^2 - \|\varphi_k\|_F^2$,
If $\|A^T R_k\|_F$ is small enough then stop,
End Do.

The Bl-LSMR algorithm will be break down at step k, if $\overline{\alpha}_k$ is singular. This happens when the matrix $\begin{bmatrix} \dot{\alpha}_k \\ \overline{B}_{k+1} \end{bmatrix}$ is not full rank. So the Bl-LSMR algorithm will not break down at step k, if $\overline{B}_{k+1}$ is nonsingular. We will not treat the problem of breakdown in this paper and we also assume that the matrices $\overline{B}_k$'s produced by the Bl-LSMR algorithm are nonsingular.

We mention that, we can use the Bl-LSMR algorithm for computing a matrix solution X to the problem

$$\mathrm{minimize}\|\mathrm{AX} - \mathrm{B}\|_\mathrm{F}, \quad \mathrm{A} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}, \quad \mathrm{B} \in \mathbb{R}^{\mathrm{m}\times\mathrm{s}}, \quad \mathrm{s} \ll \min\{\mathrm{m}, \mathrm{n}\},$$

where $m \geq n$ or $m \leq n$. In Section 5, we present the results of the Bl-LSMR algorithm for this kind of problems.

4 The convergence of the Bl-LSMR algorithm

In this section, we aim at studying the convergence behavior of the Bl-LSMR method. We first give the following lemmas.

Lemma 2. *Let $P_i, i = 1, 2, \dots, k,$ be the $n \times s$ auxiliary matrices produced by the Bl-LSMR algorithm and R_k be the residual matrix associated with the approximate solution X_k of the matrix equation(1). Then, we have*

$$(A^T A P_k)^T A^T R_k = 0_s.$$

Proof. Using $\overline{P}_k = \overline{V}_k \overline{R}_k^{-1}$ and equation(4), we have

$$\begin{aligned} A^T A P_k &= A^T A \overline{P}_k \overline{E}_k \\ &= A^T A \overline{V}_k \overline{R}_k^{-1} \overline{E}_k \\ &= \overline{V}_{k+1} \begin{bmatrix} T_k^T T_k \\ \overline{B}_{k+1} \overline{E}_k^T \end{bmatrix} \overline{R}_k^{-1} \overline{E}_k. \end{aligned} \tag{19}$$

From (19), and (7), we have

$$\begin{aligned} (A^T A P_k)^T (A^T R_k) &= \overline{E}_k^T \overline{R}_k^{-T} \left[T_k^T T_k, (\overline{B}_{k+1} \overline{E}_k^T)^T \right] \overline{V}_{k+1}^T \overline{V}_{k+1} \begin{bmatrix} \widetilde{E}_1 \overline{B}_1 - T_k^T T_k Y_k \\ -\overline{B}_{k+1} \overline{E}_k^T Y_k \end{bmatrix} \\ &= \overline{E}_k^T \overline{R}_k^{-T} (T_k^T T_k (\widetilde{E}_1 \overline{B}_1 - T_k^T T_k Y_k) - (\overline{B}_{k+1} \overline{E}_k^T)^T \overline{B}_{k+1} \overline{E}_k^T Y_k) \\ &= 0_s. \qquad \text{(from } (11)) \end{aligned}$$

We note that $\overline{V}_{k+1}$ is orthonormal, thus $\overline{V}_{k+1}^T \overline{V}_{k+1} = I_{(k+1)s}$. □

Lemma 3. *Let $P_i, i = 1, 2, \dots, k,$ be the $n \times s$ auxiliary matrices produced by the Bl-LSMR algorithm. Then we have the following property*

$$P_i^T A^T A A^T A P_i = I_s.$$

Proof. Using (19), (12), (13) and (14), we have

$$\begin{aligned}
(A^TAP_i)^T(A^TAP_i) &= (\overline{V}_{i+1}\begin{bmatrix} T_i^TT_i \\ \overline{B}_{i+1}\overline{E}_i^T \end{bmatrix}\overline{R}_i^{-1}\overline{E}_i)^T(\overline{V}_{i+1}\begin{bmatrix} T_i^TT_i \\ \overline{B}_{i+1}\overline{E}_i^T \end{bmatrix}\overline{R}_i^{-1}\overline{E}_i) \\
&= \overline{E}_i^T\overline{R}_i^{-T}\begin{bmatrix} T_i^TT_i & \overline{B}_{i+1}^T\overline{E}_i \end{bmatrix}\begin{bmatrix} T_i^TT_i \\ \overline{B}_{i+1}\overline{E}_i^T \end{bmatrix}\overline{R}_i^{-1}\overline{E}_i \\
&= \overline{E}_i^T\overline{R}_i^{-T}\overline{T}_i^T\overline{T}_i\overline{R}_i^{-T}\overline{E}_i \\
&= \overline{E}_i^T\overline{R}_i^{-T}\begin{bmatrix} \overline{R}_i^T & 0_{ks\times s} \end{bmatrix}\overline{Q}_i^T\overline{Q}_i\begin{bmatrix} \overline{R}_i \\ 0_{s\times ks} \end{bmatrix}\overline{R}_i^{-1}\overline{E}_i \\
&= \overline{E}_i^T\begin{bmatrix} I_{ks} & 0_{ks\times s} \end{bmatrix}\begin{bmatrix} I_{ks} \\ 0_{s\times ks} \end{bmatrix}\overline{E}_i \\
&= \overline{E}_i^T\overline{E}_i = I_s.
\end{aligned}$$

□

Theorem 1. *Let X_k be the approximate solution of* (1)*, obtained from the Bl-LSMR algorithm. Then*

$$\|A^TR_k\|_F \leq \|A^TR_{k-1}\|_F,$$

where $R_k = B - AX_k$.

Proof. From(16), we have

$$A^TR_{k-1} = A^TR_k + A^TAP_k\varphi_k.$$

Using Lemma 2, since A^TR_k and A^TAP_k are orthogonal, we have

$$\|A^TR_{k-1}\|_F^2 = \|A^TR_k\|_F^2 + \|A^TAP_k\varphi_k\|_F^2.$$

Thus

$$\|A^TR_k\|_F^2 = \|A^TR_{k-1}\|_F^2 - \|A^TAP_k\varphi_k\|_F^2.$$

Using Lemma 3, we have

$$\begin{aligned}
&\|A^TR_k\|_F^2 = \|A^TR_{k-1}\|_F^2 - \|\varphi_k\|_F^2, \\
&\|A^TR_k\|_F \leq \|A^TR_{k-1}\|_F.
\end{aligned}$$

□

Theorem 1 is helpful in showing that if $\|\varphi_k\|_F$ is not very small in each iteration of the Bl-LSMR algorithm, then the Bl-LSMR algorithm will be stopped after a finite number of iterations. Otherwise, it is possible to occur stagnation. In this case, we can apply a reliable preconditioner for the block linear system of equations (1).

5 Numerical examples

In this section, we consider the system $AX = B$, where $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times s}$, $X \in \mathbb{R}^{n\times s}$, and we present numerical results for several matrices taken from the University of Florida Sparse Matrix Collection (Davis [7]). These matrices with their properties are shown in Table 1. Our implementation is done on MATLAB version 07 on a PC machine with 4 GB RAM. Moreover, for the initial guess $X_0 = 0_{n\times s}$ and $B = \mathrm{rand}(m, s)$, where the function rand creates an $m \times s$ random matrix with the coefficients uniformly distributed in $[0, 1]$. The stopping criteria is set to $\|A^T R_k\|_F/\|R_k\|_F \leq 10^{-10} \times \|A\|_F$.

Diagonal scaling was applied to the columns of $[A,\ B]$ to give a scaled problem $AX = B$, in which the columns of $[A,\ B]$ have unit 2-norm. By scaling, the number of iterations of Bl-LSMR for convergence reduced satisfactorily.

In Table 2, we give the ratio $t(s)/t(1)$, for $s = 5, 10, 20$, and 30, where $t(s)$ is the CPU time for Bl-LSMR algorithm and $t(1)$ is the CPU time obtained when applying LSMR for one right-hand side linear system. Note that the time obtained by LSMR for one right-hand side depends on which right-hand was used. So, $t(1)$ is the average of the times needed for the s right-hand sides using LSMR. The results of Table 2 show that the Bl-LSMR algorithm is effective and less expensive than the LSMR algorithm, because the indicator $t(s)/t(1)$ is less than s.

To show that the Frobenius norm of residual matrix decreases monotonically, we display the convergence history in Figure 1 for the systems corresponding to the matrices of Table 2 and Bl-LSMR algorithm. In this figure, the vertical axis and horizontal axis are the logarithm in base 10 of the Frobenius norm of residual matrix and the number of iterations to convergence, respectively. We observe that for all matrices the Frobenius norm of residual matrix decreases monotonically.

We display the convergence history of Bl-LSMR and Bl-LSQR in Figure 2 for the system corresponding to the matrix LPnetlib/lp_pilot. Figure 3 (left and right) shows both solvers reducing $\|A^T R_k\|_F/\|R_k\|_F$ and $\|R_k\|_F$ monotonically and similarly.

Table 1: Test problems information

Matrix\Property	rows	columns	sym	nnz	id	Discipline
Hamm/add32	4960	4960	no	19848	540	Electronic circuit design
Simon/appu	14000	14000	no	1853104	811	Random sparse matrix used in the APP BENCHMARK
HB/fs6801	680	680	no	2184	149	Chemical kinetics
HB/gre115	115	115	no	421	161	Simulation studies in computer systems
HB/gr-30-30	900	900	yes	7744	159	Partial differential equations
LPnetlib/lpadlittle	56	138	no	424	596	Linear programming problem
LPnetlib/lp_maros	846	1966	no	10137	642	Linear programming problem
LPnetlib/lp_pilot	1441	4860	no	44375	654	Linear programming problem
LPnetlib/lp_sc205	205	317	no	665	665	Linear programming problem
Bai/pde2961	2961	2961	no	14585	324	Partial differential equations
Bai/pde900	900	900	no	4380	325	Partial differential equations
Bai/rdb3200l	3200	3200	no	18880	1633	Chemical engineering
HB/sherman4	1104	1104	no	3786	245	Oil reservoir modeling

Table 2: Effectiveness of Bl-LSMR algorithm measured $t(s)/t(1)$

Matrix\s	5	10	20	30
Hamm/add32	0.47	0.95	3.07	5.39
Simon/appu	1.24	1.89	3.21	5.13
HB/fs6801	0.27	0.38	0.97	1.19
HB/gre115	0.99	0.51	3.41	8.57
HB/gr-30-30	1.55	1.72	2.05	2.53
LPnetlib/lpadlittle	0.37	0.42	1.63	12.54
LPnetlib/lp_maros	2.92	3.75	6.79	12.36
LPnetlib/lp_pilot	2.40	4.95	15.90	22.92
LPnetlib/lp_sc205	0.70	1.30	2.11	4.70
Bai/pde2961	0.33	0.52	0.98	1.14
Bai/pde900	0.49	0.72	1.10	1.47
Bai/rdb3200l	0.30	0.39	0.38	0.76
HB/sherman4	0.37	0.50	0.54	1.03

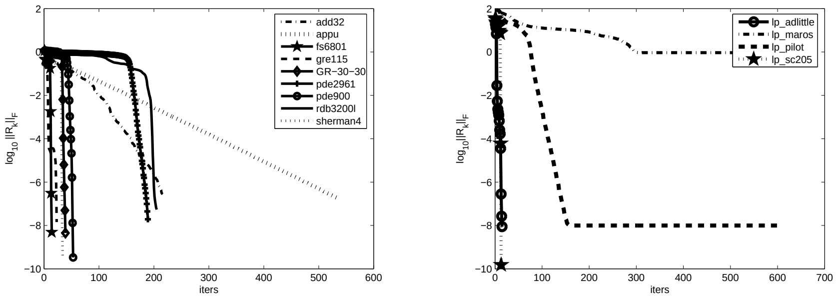

Figure 1: Convergence history of the Bl-LSMR algorithm with s=20

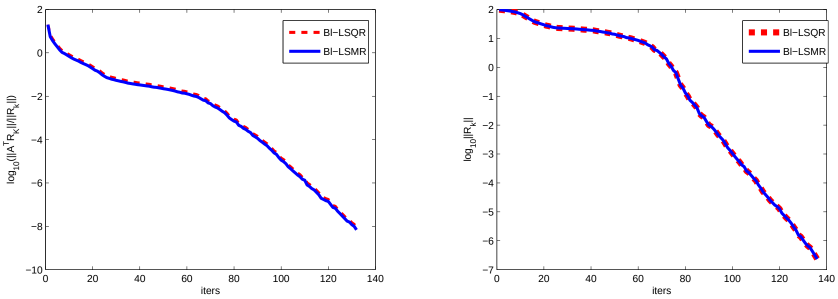

Figure 2: Bl-LSMR and Bl-LSQR solving a linear system AX = B with s = 20: problem LPnetlib/lp_pilot

6 Conclusion

In this paper, we have presented a block version of LSMR algorithm for solving linear systems with multiple right-hand sides. We derived a simple recurrence formula for generating the sequence of approximate solutions $\{X_k\}$ such that the Frobenius norm of the quantity $A^T R_k$ decreases monotonically. In addition, we studied the convergence of the presented method. Besides, we showed that in absence of the break down condition, the presented algorithm always converges. Numerical results have shown that the new algorithm obtains the results which are effective and less expensive than the LSMR algorithm applied to each right-hand side.

Acknowledgements
We would like to thank the referees for their valuable remarks and helpful suggestions.

References

1. Abdel-Rehim, A. M., Morgan, R. B. and Wilcox, W. *Improved seed methods for symmetric positive definite linear equations with multiple right-hand sides*, 2008, Arxiv preprint arXiv:0810.0330v1.

2. Bellalij, M., Jbilou, K. and Sadok, H.*New convergence results on the global GMRES method for diagonalizable matrices*, J. Comput. Appl. Math. 219 (2008) 350-358.

3. Chan,T. F. and Wang, W.*Analysis of projection methods for solving linear systems with multiple right-hand sides*, SIAM J. Sci. Comput. 18 (1997) 1698-1721.

4. Chin-Lung Fong, D. and Saunders, M. *LSMR: An iterative algorithm for sparse least-squares problems*, SIAM J. Sci. Comput. 33 (2011) 2950-2971.

5. Dai, H. *Two algorithms for symmetric linear systems with multiple right-hand sides*, Numer. Math. J. Chin. Univ. (Engl. Ser.) 9 (2000) 91-110.

6. Darnell, D., Morgan, R. B. and Wilcox, W. *Deflated GMRES for systems with multiple shifts and multiple right-hand sides*, Linear Algebra Appl. 429 (2008) 2415-2434.

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

8. Freund, R. and Malhotra, M. *A Block-QMR algorithm for non-hermitian linear systems with multiple right-hand sides*, Linear Algebra Appl. 254 (1997) 119-157.

9. Golub, G. H., Luk, F. T. and Overton, M. L. *A Block Lanczos method for computing the singular values and corresponding singular vectors of the matrix*, ACM Trans. Math. Software 7 (1981) 149-169.

10. Golub, G. H. and Kahan, W. *Calculating the singular values and pseudoinverse of a matrix*, SIAM J. Numer. Anal, 2 (1965) 205-224.

11. Golub, G. H. and Van Loan, C. F. *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

12. Gu, G. and Cao, Z.*A block GMRES method augmented with eigenvectors*, Appl. Math. Comput. 121 (2001) 271-289.

13. Gu, G. and Qian, H. *Skew-symmetric methods for solving nonsymmetric linear systems with multiple right-hand sides*, J. Comput. Appl. Math. 223 (2009) 567-577.

14. Gu, C. and Yang, Z. *Global SCD algorithm for real positive definite linear systems with multiple right-hand sides*, Appl. Math. Comput. 189 (2007) 59-67.

15. Guennouni, A. El., Jbilou, K. and Sadok, H. *A block version of BICGSTAB for linear systems with multiple right-hand sides*, Elec. Trans. Numer. Anal. 16 (2003) 129-142.

16. Guennouni, A. El., Jbilou, K. and Sadok, H. *The block Lanczos method for linear systems with multiple right-hand sides*, Appl. Numer. Math. 51 (2004) 243-256.

17. Gutknecht, M. H. *Block Krylov space methods for linear systems with multiple right-hand sides: an introduction, in: A. H. Siddiqi, I. S. Duff, O. Christensen (Eds.), Modern Mathematical Models, Methods and Algorithms for Real Word Systems*, Anamaya Publishers, New Delhi, India, 2007, 420-447.

18. Haase, G. and Reitzinger, S. *Cache issues of algebraic multigrid methods for linear systems with multiple right-hand sides*, SIAM J. Sci. Comput. 27 (2005) 1-18.

19. Heyouni, M.*The global Hessenberg and global CMRH methods for linear systems with multiple right-hand sides*, Numer. Algorithms 26 (2001) 317-332.

20. Heyouni, M. and Essai, A. *Matrix Krylov subspace methods for linear systems with multiple right-hand sides*, Numer. Algorithms 40 (2005) 137-156.

21. Jbilou, K., Messaoudi, A. and Sadok, H. *Global FOM and GMRES algorithms for matrix equations*, Appl. Numer. Math. 31 (1999) 49-63.

22. Jbilou, K. and Sadok, H. *Global Lanczos-based methods with applications*, Technical Report LMA 42, Universiti du Littoral, Calais, France, 1997.

23. Jbilou, K, Sadok, H. and Tinzefte, A. *Oblique projection methods for linear systems with multiple right-hand sides*, Elec. Trans. Numer. Anal. 20 (2005)119-138.

24. Joly, P. *Resolution de Systems Lineaires Avec Plusieurs Second Members par la Methode du Gradient Conjugue*, Tech. Rep. R-91012, Publications du Laboratire d'Analyse Numerique, Universite Pierre et Marie Curie, Paris, 1991.

25. Karimi, S. and Toutounian, F. *The block least squares method for solving nonsymmetric linear systems with multiple right-hand sides*, Appl. Math. Comput. 177 (2006) 852-862.

26. Lin, Y. *Implicitly restarted global FOM and GMRES for nonsymmetric matrix equations and Sylvester equations*, Appl. Math. Comput. 167 (2005) 1004-1025.

27. Liu, H. and Zhong, B. *Simpler block GMRES for nonsymmetric systems with multiple right-hand sides*, Elec. Trans. Numer. Anal. 30 (2008) 1-9.

28. Mojarrab, M. *The Block least square methods for matrix equations*, PhD thesis, Ferdowsi University of Mashhad, Iran, 2014.

29. Morgan, R. B. *Restarted block-GMRES with deflation of eignvalues*, Appl. Numer. Math. 54 (2005) 222-236.

30. Nikishin, A. and Yeremin, A. *Variable block CG algorithms for solving large sparse symmetric positive definite linear systems on parallel computers I: general iterative scheme*, SIAM J. Matrix Anal. 16 (1995) 1135-1153.

31. ÓLeary, D. *The block conjugate gradient algorithm and related methods*, Linear Algebra Appl. 29 (1980) 293-322.

32. Paige, C. C. and Saunders, M. A. *LSQR: an algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software 8 (1982) 43-71.

33. Robbe, M. and Sadkane, M. *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra Appl. 419 (2006) 265-285.

34. Saad, Y. *Iterative methods for sparse linear systems*, SIAM, 2nd edn, 2003.

35. Saad, Y. *On the Lanczos method for solving symmetric linear systems with several right-hand sides*, Math. Comput. 48 (1987) 651-662.

36. Salkuyeh, D. K. *CG-type algorithms to solve symmetric matrix equations*, Appl. Math. Comput. 172 (2006) 985-999.

37. Simoncini, V. *A stabilized QMR version of block BICG*, SIAM J. Matrix Anal. Appl. 18 (1997) 419-434.

38. Simoncini, V. and Gallopoulos, E. *Convergence properties of block GMRES and matrix polynomials*, Linear Algebra Appl. 247 (1996) 97-119.

39. Simoncini, V. and Gallopoulos, E. *An iterative method for nonsymmetric systems with multiple right-hand sides*, SIAM J. Sci. Comput. 16 (1995) 917-933.

40. Smith, C., Peterson, A. and Mittra, R. *A conjugate gradient algorithm for treatment of multiple incident electromagnetic fields*, IEEE Trans. Antennas Propagation 37 (1989) 1490-1493.

41. Toutounian, F. and Karimi, S. *Global least squares method (Gl-LSQR) for solving general linear systems with several right-hand sides*, Appl. Math. Comput. 178 (2006) 452-460.

42. Van Der Vorst, H. *An iterative solution method for solving f(A) = b, using Krylov subspace information obtained for the symmetric positive definite matrix A*, J. Comput. Appl. 18 (1987) 249-263.

43. Vital, B. *Etude de Quelques Méthodes de Résolution de Problém̀s Linéaires de Grande Taille sur Multiprocesseur*, Ph.D. Thesis, Univérsité de Rennes, Rennes, France, 1990.

44. Zhang, J. and Dai, H. *Global CGS algorithm for linear systems with multiple right-hand sides*, Numer. Math. J. Chin. Univ. 30 (2008) 390-399 (in Chinese).

45. Zhang, J., Dai, H. and Zhao, J. *A new family of global methods for linear systems with multiple right-hand sides*, J. Comput. Appl. Math. 236 (2011) 1562-1575.

46. Zhang, J., Dai, H. and Zhao, J. *Generalized global conjugate gradient squared algorithm*, Appl. Math. Comput. 216 (2010) 3694-3706.

الگوریتم بلوکیLSMR برای حل دستگاه معادلات خطی با چند طرف ثانی

فائزه توتونیان [۲,۱] *و مریم مجرب* [۳,۱]

[۱] دانشگاه فردوسی مشهد، دانشکده علوم ریاضی، گروه ریاضی کاربردی
[۲] دانشگاه فردوسی مشهد، قطب علمی مدلسازی و کنترل دستگاه ها
[۳] دانشگاه سیستان و بلوچستان، گروه ریاضی

چکیده : LSMR (مانده مینیمال کمترین توان‌های دوم) یک روش تکراری برای حل دستگاه معادلات خطی و مسائل کمترین توان‌های دوم می‌باشد. این مقاله یک نسخه بلوکی از الگوریتم LSMR را برای حل دستگاه‌های خطی با چند طرف ثانی ارائه می‌دهد. الگوریتم جدید مبتنی بر دوقطری سازی بلوکی است و از مینیمم سازی نرم فربینیوس معادلات نرمال ماتریس مانده نتیجه می‌شود. به‌علاوه، همگرایی الگوریتم پیشنهادی مورد بحث قرار می‌گیرد. همچنین، در عمل ملاحظه می‌شود که نرم فربینیوس ماتریس مانده به طور یکنواخت کاهش می‌یابد. در نهایت، آزمایش‌های عددی که بر روی مسائل کاربردی واقعی پیاده سازی شده‌اند، کارایی روش ارائه شده را تایید خواهند کرد.

کلمات کلیدی : روش LSMR ؛ دوقطری سازی؛ روش‌های بلوکی؛ روش‌های تکراری؛ چند طرف ثانی.